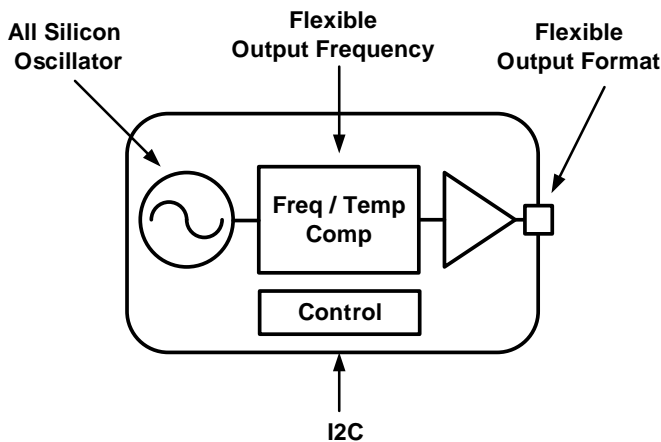


AS5003 Arcadium™ I²C Programmable Oscillator, 10 kHz to 350 MHz

The AS5003 Arcadium™ all-silicon oscillator utilizes proprietary frequency synthesis and sensor technologies to provide a quartz-free, MEMS-free, low jitter clock at any output frequency. The device is fully programmable to any frequency from 10 kHz to 350 MHz with < 0.026 ppb resolution and maintains low jitter across its operating range. The AS5003 uses on-chip temperature and strain sensors, and an advanced LC tank architecture to achieve excellent reliabilities even in high impact shock scenarios.

AS5003's on-chip power supply filtering provides industry-leading power supply noise rejection, simplifying the task of generating low jitter clocks in noisy systems that use switched-mode power supplies. Offered in a variety of industry-standard packages, the AS5003 has a dramatically simplified supply chain that enables Aeonsemi to ship samples shortly after receipt of order. The AS5003 is user-configurable with an I²C interface for a wide variety of user specifications, including frequency, output format, and DCO mode. Default configuration is factory programmed at time of shipment. It also guarantees 100% electrical testing of every device before shipment.



KEY FEATURES

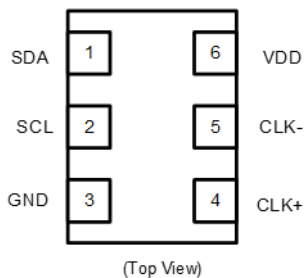
- Quartz-free and MEMS-free without mechanical moving parts
- I²C programmable to any frequency
 - Differential: 10 kHz to 350 MHz
 - LVCMOS: 10 kHz to 212.5 MHz
- Up to 0.026 ppb frequency tuning resolution
- I²C interface supports 100 kbps, 400 kbps and 1 Mbps (Fast Mode Plus)
- Low jitter: 350 fs Typ RMS (12 kHz – 20 MHz)
- Compliant to PCIe Gen 1/2/3/4/5 jitter requirements
- 50 ppm stability (-40 to 105°C)
- Integrated LDO for on-chip power supply noise filtering
- Support continuous 1.8V to 3.3V V_{DD} supply operation
- LVPECL, LVDS, CML, HCSL, CMOS, and Dual CMOS output options
- Industrial standard 3.2 x 5, 2.5 x 3.2 mm package footprints

APPLICATIONS

- 1G/10G/40G/100G Ethernet
- Servers, switches, storage, NICs, search acceleration
- Test and measurement
- Clock and data recovery
- FPGA/ASIC clocking

Pin Assignments

(LVPECL/LVDS/HCSL/CML/Dual CMOS)



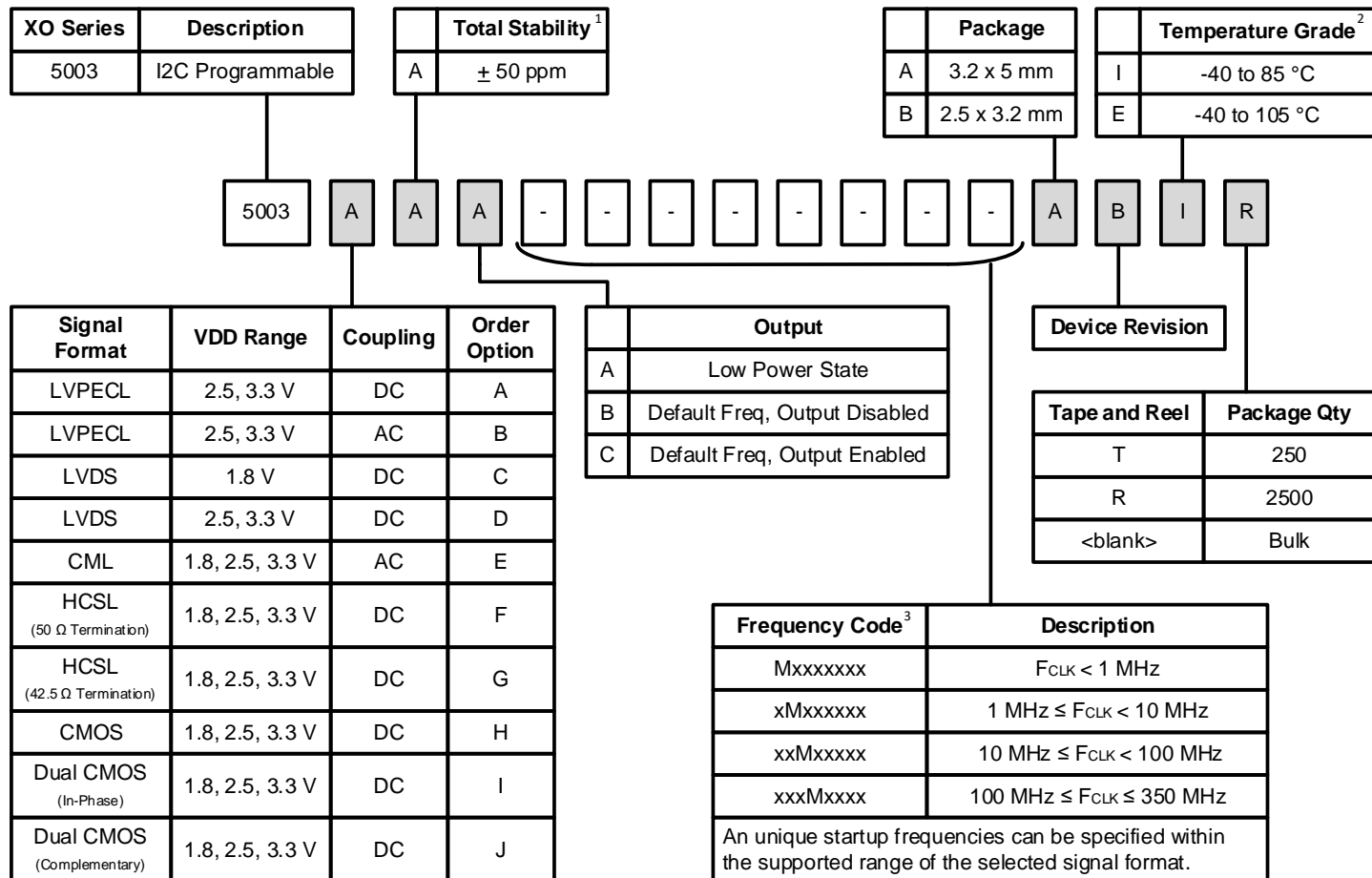
3.2 x 5 mm and 2.5 x 3.2 mm

| Pin # | Descriptions |
|-------|-------------------------------------|
| 1 | SDA = I ² C Serial Data |
| 2 | SCL = I ² C Serial Clock |
| 3 | GND = Ground |
| 4 | CLK+ = Clock output |
| 5 | CLK- = Complementary clock output |
| 6 | V _{DD} = Power supply |



1. Ordering Guide

The AS5003 Oscillator supports a variety of initial options including frequency, output format, as shown in the chart below. Configurations are changeable by users via I2C interface upon startup. Samples are available in 2 weeks.



Notes:

- Total stability includes temp stability, initial accuracy, load pulling, V_{DD} variation, and 10 years aging at 40 °C.
- Device supports extended industrial temperature range of -40 to 105°C only with $V_{DD} = 1.8\text{V}$ (+/- 5%).
- For example: 156.25 MHz = 156M2500; 25 MHz = 25M00000.

2. Electrical Specifications

Table 2.1 Electrical Specifications
 $V_{DD} = 1.8\text{ V}, 2.5\text{ or }3.3\text{ V} \pm 5\%$, $T_A = -40\text{ to }85\text{ }^\circ\text{C}$; $V_{DD} = 1.8\text{ V} \pm 5\%$, $T_A = -40\text{ to }105\text{ }^\circ\text{C}$

| Parameter | Symbol | Test Condition/Comment | Min | Typ | Max | Unit |
|---|------------|---|---------------------|------|---------------------|------------------|
| Temperature Range | T_A | | -40 | — | 105 | $^\circ\text{C}$ |
| Frequency Range | F_{CLK} | LVPECL, LVDS, CML, HCSL | 0.01 | — | 350 | MHz |
| | | CMOS, Dual CMOS | 0.01 | — | 212.5 | MHz |
| Supply Voltage | V_{DD} | | 1.71 | | 3.47 | V |
| Supply Current ($F_{CLK} = 50\text{ MHz}$) | I_{DD} | Tristate Hi-Z (output disabled) | — | 40 | 50 | mA |
| | | Ready State | — | 1 | 2 | mA |
| | | LVPECL (DC-Coupled) | — | 70 | 80 | mA |
| | | LVPECL (AC-Coupled) | — | 60 | 70 | mA |
| | | LVDS | — | 45 | 55 | mA |
| | | HCSL | — | 60 | 70 | mA |
| | | CML | — | 60 | 70 | mA |
| | | CMOS | — | 40 | 55 | mA |
| | | Dual CMOS | — | 50 | 60 | mA |
| Total Stability ¹ | F_{STAB} | Frequency stability | -50 | — | 50 | ppm |
| Rise/Fall Time (20% to 80% V_{PP}) | T_R/T_F | LVPECL/LVDS/CML | — | — | 350 | ps |
| | | CMOS / Dual CMOS ($C_L = 5\text{ pF}$) | — | 0.5 | 1.5 | ns |
| | | HCSL, $F_{CLK} > 50\text{ MHz}$ | — | — | 550 | ps |
| Duty Cycle | D_C | All formats | 45 | — | 55 | % |
| Output Enable (OE) ² | V_{IH} | | $0.7 \times V_{DD}$ | — | — | V |
| | V_{IL} | | — | — | $0.3 \times V_{DD}$ | V |
| | T_D | Output Disable Time, $F_{CLK} > 10\text{ MHz}$ | — | — | 0.5 | μs |
| | T_E | Output Enable Time, $F_{CLK} > 10\text{ MHz}$ | — | — | 0.5 | μs |
| Powerup Time | t_{OSC} | Time from $0.9 \times V_{DD}$ until output frequency (F_{CLK}) within spec | — | — | 4 | ms |
| LVPECL Output Option ³ (DC-Coupled) | V_{OC} | Mid-level | $V_{DD} - 1.55$ | — | $V_{DD} - 1.25$ | V |
| | V_O | Swing (diff) | 1.4 | — | 1.85 | V_{PP} |
| LVPECL Output Option ³ (AC-Coupled) | V_O | Swing (diff) | 1.4 | | 1.85 | V_{PP} |
| LVDS Output Option ⁴ (DC-Coupled) | V_{OC} | Mid-level (2.5 V, 3.3 V V_{DD}) | 1.125 | 1.20 | 1.275 | V |
| | | Mid-level (1.8 V V_{DD}) | 0.795 | 0.85 | 0.905 | V |
| | V_O | Swing (diff) | 0.5 | 0.82 | 0.96 | V_{PP} |

| Parameter | Symbol | Test Condition/Comment | Min | Typ | Max | Unit |
|--|-----------------|--|-----------------------------|-----|-----------------------------|-----------------|
| HCSL Output Option ⁴ ($R_{\text{term}} = 50 \Omega$; DC-Coupled) | V_{OH} | Output voltage high | 695 | 815 | 935 | mV |
| | V_{OL} | Output voltage low | 0 | 5 | 10 | mV |
| HCSL Output Option ⁵ ($R_{\text{term}} = 42.5 \Omega$; DC-Coupled) | V_{OH} | Output voltage high | 695 | 820 | 945 | mV |
| | V_{OL} | Output voltage low | 0 | 5 | 10 | mV |
| CML Output Option ⁴ (AC-Coupled) | V_{O} | Swing (diff) | 0.725 | 0.8 | 0.89 | V_{PP} |
| CMOS Output Option | V_{OH} | $I_{\text{OH}} = 8/6/4 \text{ mA for } 3.3/2.5/1.8V V_{\text{DD}}$ | $0.83 \times V_{\text{DD}}$ | — | — | V |
| | V_{OL} | $I_{\text{OL}} = 8/6/4 \text{ mA for } 3.3/2.5/1.8V V_{\text{DD}}$ | — | — | $0.17 \times V_{\text{DD}}$ | V |

Notes:

1. Total Stability includes temperature stability, initial accuracy, load pulling, V_{DD} variation, and aging for 10 years at 40°C.
2. The T_{D} and $T_{\text{E}} < 10 \text{ ns} + 3 * 1/F_{\text{CLK}}$ for all frequencies measured from the end of the I²C byte write to OE control registers.
3. $R_{\text{term}} = 50 \Omega$ to $V_{\text{DD}} - 2.0 \text{ V}$ (see Figure 4.1).
4. $R_{\text{term}} = 100 \Omega$ (differential) (see Figure 4.2).
5. $R_{\text{term}} = 42.5 \Omega$ or 50Ω to GND (see Figure 4.3).

Table 2.2: I²C Characteristics

$V_{\text{DD}} = 1.8 \text{ V}, 2.5 \text{ or } 3.3 \text{ V} \pm 5\%$, $T_{\text{A}} = -40 \text{ to } 85 \text{ }^\circ\text{C}$; $V_{\text{DD}} = 1.8 \text{ V} \pm 5\%$, $T_{\text{A}} = -40 \text{ to } 105 \text{ }^\circ\text{C}$

| Parameter | Symbol | Test Condition/Comment | Min | Typ | Max | Unit |
|---|------------------|---|------|-------|------|------|
| Frequency Reprogramming Resolution | V_{RES} | | — | 0.026 | — | ppb |
| Frequency Range for Small Frequency Change (Continuous Glitchless Output) | | From center frequency | -970 | | +970 | ppm |
| Settling Time for Small Frequency Change (DCXO Feature) | | $< \pm 970 \text{ ppm}$ from center frequency | | | 8 | us |
| Settling Time for Frequency Change (fUSER_FREQ Register) | | | | | 200 | us |

Table 2.3: Clock Output Phase Jitter and PSRR

$V_{DD} = 1.8\text{ V}$, 2.5 or 3.3 V $\pm 5\%$, $T_A = -40$ to 85 °C; $V_{DD} = 1.8\text{ V} \pm 5\%$, $T_A = -40$ to 105 °C

| Parameter | Symbol | Test Condition/Comment | Min | Typ | Max | Unit |
|--|----------|------------------------|-----|-----|-----|------|
| Phase Jitter (RMS, 12 kHz - 20 MHz) ^{1,2} $F_{CLK} \geq 10\text{ MHz}$ | ϕ_J | Differential Formats | — | 350 | 750 | fs |
| | | CMOS, Dual CMOS | — | 350 | — | fs |
| Phase Jitter (RMS, 50 kHz - 20 MHz) $F_{CLK} \geq 156.25\text{ MHz}$ | ϕ_J | Differential Formats | — | 150 | 250 | fs |
| | | CMOS, Dual CMOS | — | 100 | — | fs |
| Spurs Induced by External Power Supply Noise, 50 mVpp Ripple. LVDS 156.25 MHz Output $V_{DD} = 1.8\text{ V}$ | PSRR | 100 kHz sine wave | — | -76 | — | dBc |
| | | 200 kHz sine wave | — | -75 | — | |
| | | 500 kHz sine wave | — | -75 | — | |
| | | 1 MHz sine wave | — | -75 | — | |
| Spurs Induced by External Power Supply Noise, 50 mVpp Ripple. LVDS 156.25 MHz Output $V_{DD} = 2.5$ or 3.3 V | PSRR | 100 kHz sine wave | — | -83 | — | dBc |
| | | 200 kHz sine wave | — | -83 | — | |
| | | 500 kHz sine wave | — | -83 | — | |
| | | 1 MHz sine wave | — | -82 | — | |

Note:

1. Applies to output frequency: 50, 100, 156.25, 212.5, 350 MHz
2. Guaranteed by characterization. Jitter inclusive of any spurs

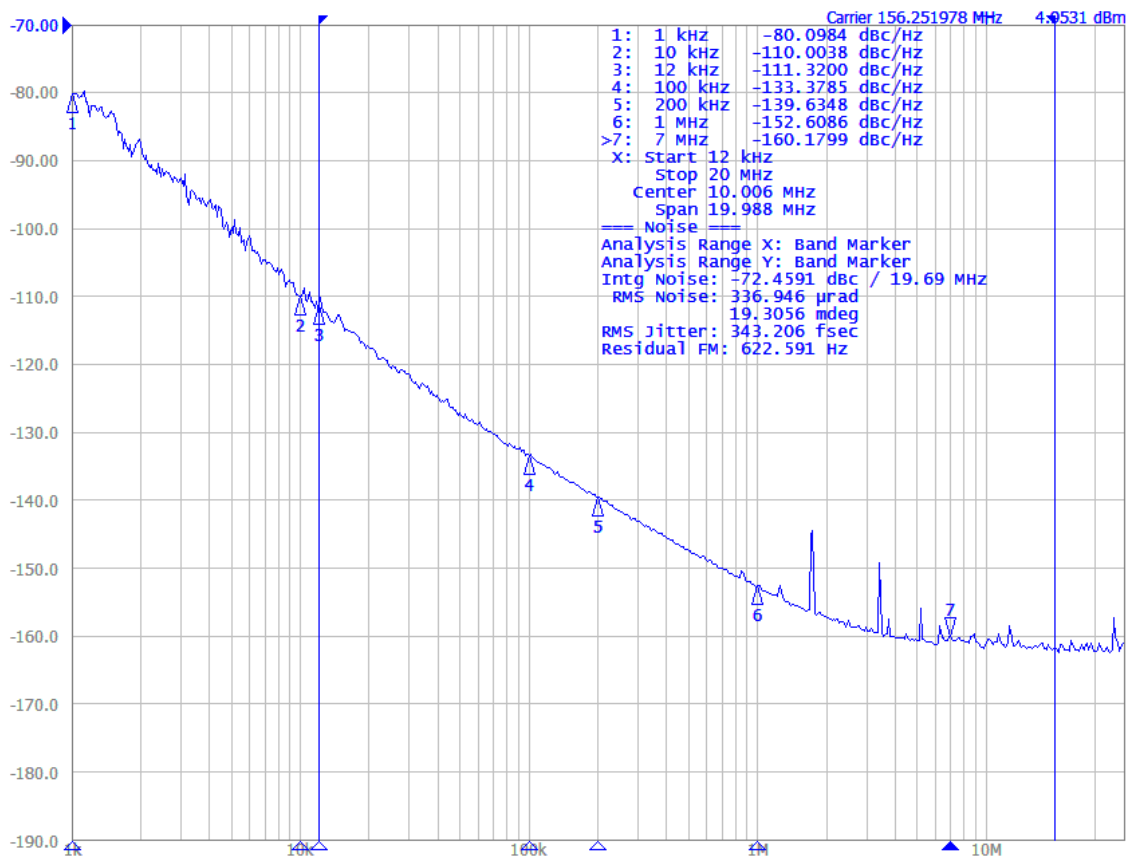
Figure 2.1: Phase Noise at 156.25 MHz

Table 2.4: Environmental Compliance and Package Information

| Parameter | Test Condition |
|---|----------------|
| Moisture Sensitivity Level | 1 |
| Note: For additional product information not listed in the data sheet (e.g. RoHS Certifications, MSDS data, qualification data, REACH Declarations, ECCN codes, etc.), contact aeonsemi.com/contact-us/ | |

Table 2.5: Thermal Conditions

| Package | Parameter | Symbol | Test Condition | Value | Unit |
|----------------------------|--|---------------|----------------|-------|------|
| 3.2 x 5 mm, 6-pin DFN | Thermal Resistance Junction to Ambient | Θ_{JA} | Still Air | 105 | °C/W |
| | Thermal Resistance Junction to Board | Θ_{JB} | Still Air | 81 | °C/W |
| | Max Junction Temperature | T_J | Still Air | 125 | °C |
| 2.5 x 3.2 mm, 6-pin DFN | Thermal Resistance Junction to Ambient | Θ_{JA} | Still Air | 108 | °C/W |
| | Thermal Resistance Junction to Board | Θ_{JB} | Still Air | 84 | °C/W |
| | Max Junction Temperature | T_J | Still Air | 125 | °C |

Table 2.6: Absolute Maximum Ratings¹

| Parameter | Symbol | Rating | Unit |
|--|------------|------------------------|------|
| Maximum Operating Temperature ² | T_{AMAX} | 105 | °C |
| Storage Temperature | T_S | -55 to 125 | °C |
| Supply Voltage | V_{DD} | -0.5 to 3.8 | V |
| Input Voltage | V_{IN} | -0.5 to $V_{DD} + 0.3$ | V |
| ESD HBM (JESD22-A114) | HBM | 4.0 | kV |
| Solder Temperature ³ | T_{PEAK} | 260 | °C |
| Solder Time at T_{PEAK} ³ | T_P | 20 - 40 | sec |

Notes:

- Stresses beyond those listed in this table may cause permanent damage to the device. Functional operation specification compliance is not implied at these conditions. Exposure to maximum rating conditions for extended periods may affect device reliability.
- For $V_{DD} = 1.8V$ only; otherwise 85°C.
- The device is compliant with JEDEC J-STD-020.

3. CMOS Buffer and Output Terminations

Dual CMOS output format ordering options support either complementary or in-phase signals for two identical frequency outputs. This feature enables replacement of multiple XOs with a single AS5003 device.

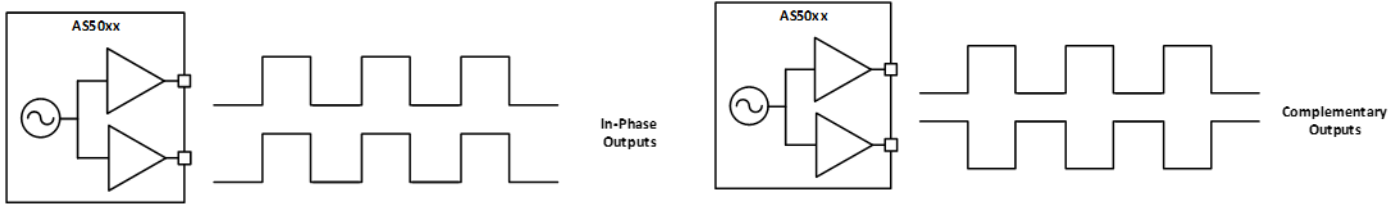


Figure 3.1: Integrated 1:2 CMOS Buffer Supports In-Phase or Complementary Outputs

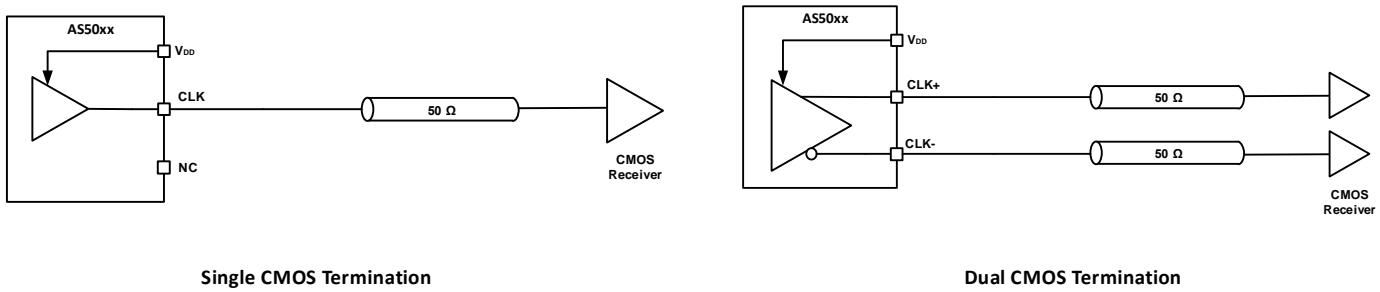
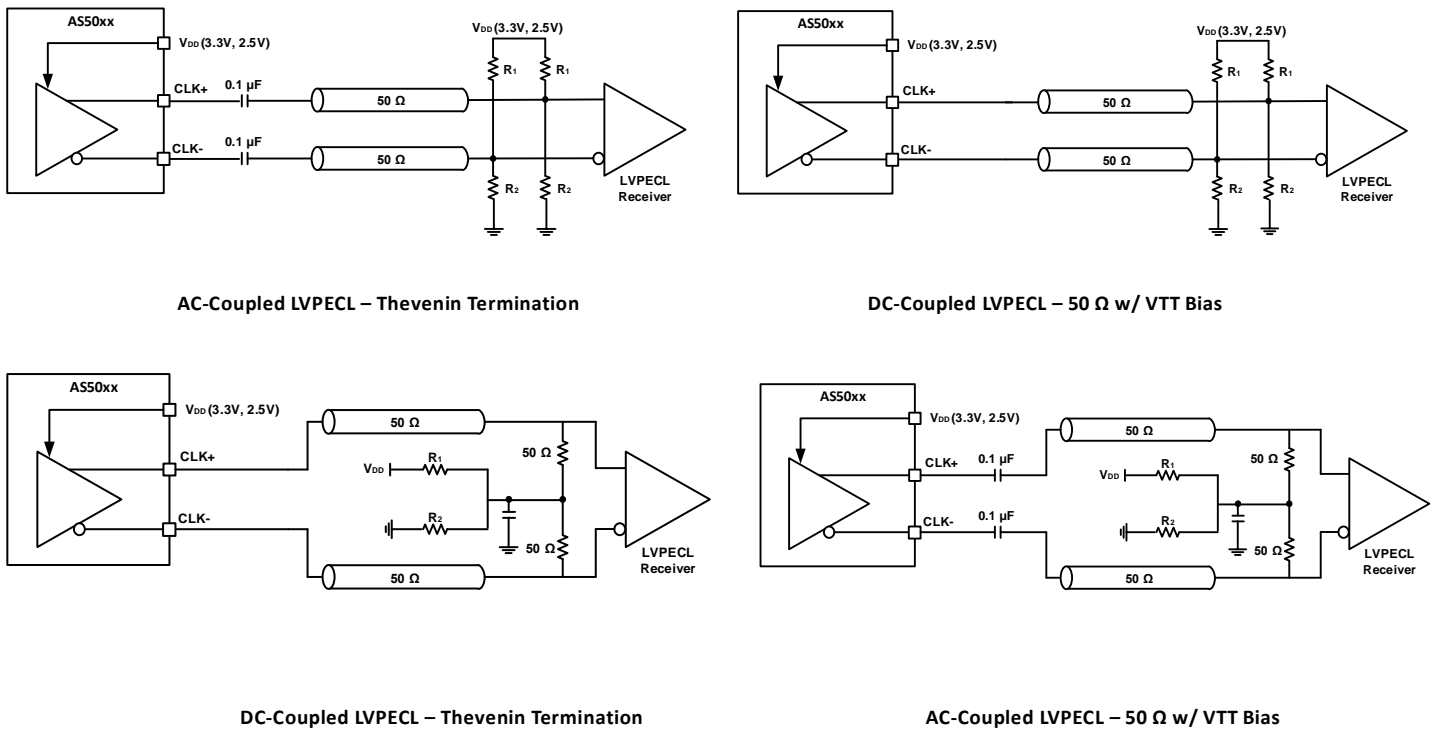


Figure 3.2: CMOS Output Terminations

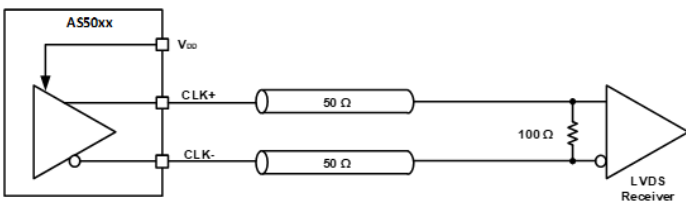
4. Recommended Output Terminations

The output drivers support AC-coupled or DC-coupled terminations as shown in figures below.

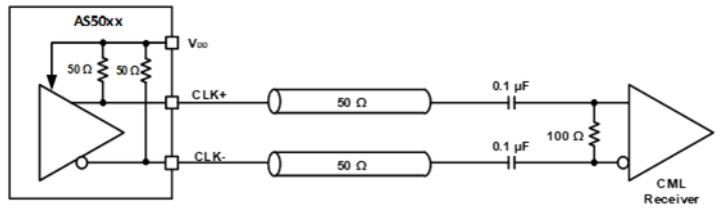


| Termination Resistor Values | | |
|-----------------------------|-------|--------|
| V _{DD} | R1 | R2 |
| 3.3 V | 127 Ω | 82.5 Ω |
| 2.5 V | 250 Ω | 62.5 Ω |

Figure 4.1: LVPECL Output Terminations

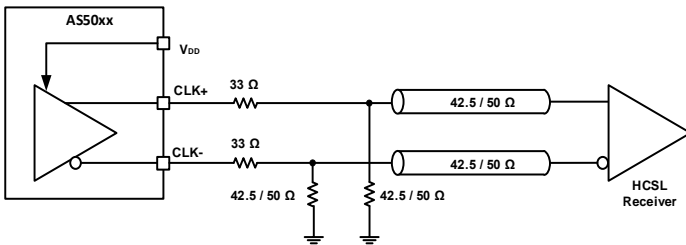


DC-Coupled LVDS

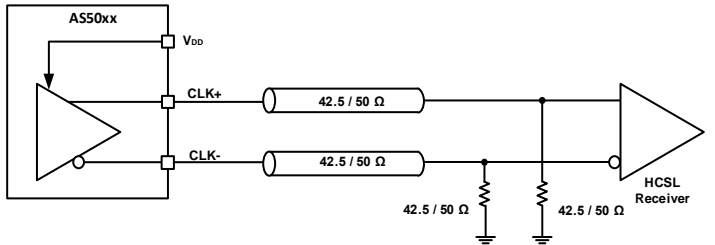


AC-Coupled CML

Figure 4.2: LVDS / CML Output Terminations



Source Terminated HCSL



Destination Terminated HCSL

Figure 4.3: HCSL Output Terminations

5. Configuring via I²C Interface

5.1 I²C Serial Interface

The I²C interface on the AS5003 is fully compatible with the “UM10204 I²C-bus specification and user manual, Rev. 6 - 4 April 2014” standard, as described in Table 5.1.

Table 5.1: I²C compatibility

| I ² C | Speed | Compliance | Notes |
|------------------|---------|------------|---|
| Standard | 100 kHz | Compliant | N/A |
| Fast | 400 kHz | Compatible | SDA falling edge can be faster than 20 ns depending on loading |
| Fast+ | 1 MHz | Compatible | SDA falling edge can be faster than 20 ns depending on loading. SDA max pull down current is 6 mA. |

5.2 I²C Register Write and Read Protocol

AS5003 implements an 8-bit I²C address space with 256 addressable byte register locations. Certain device register and bits are reserved, and they must not be changed from their default reset state.

In an I²C bus system, the AS5003 acts as a slave device connected to the I²C serial interface bus. It is accessed via a 7-bit factory programmed (per user specification) slave address. Allowed values of this device address are in the range from 16 to 119. Both write and read register transactions with register address autoincrement are enabled as shown in *Figure 5.1*.

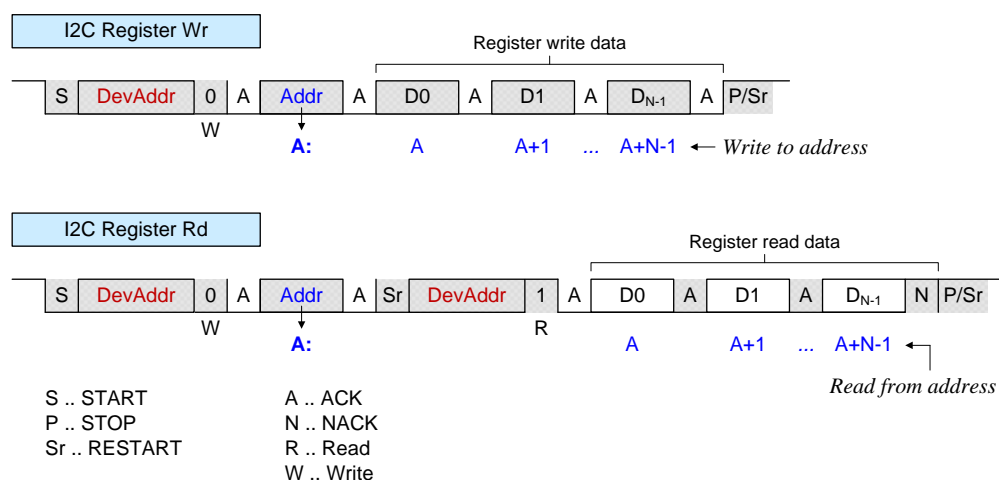


Figure 5.1: I²C write and read transactions

Write register transaction is an I²C write transaction with an 8-bit register address data byte stream. It is followed by one or more register data bytes. Read register sequence starts after a write transaction to set the read register address. It is followed by the I²C read transaction to read one or more data bytes.

The register address autoincrement is enabled upon power up. It is incremented till a maximum address 0xFF of the I²C register space is reached.

The register address autoincrement is disabled by writing register `bl2C_INC_DIS=1` and re-enabled again by writing `bl2C_INC_DIS=0`. During this process, all bytes in the I²C transactions are written to or read from a set address. Having the register autoincrement disabled is required for a DCXO streaming mode.

Data and address bytes appear on the SDA bus with the most significant bit (MSB) first per I²C standard. During I²C transactions, SCL clock bus is never stalled by the device.

5.3 Device Operation

After an initial power up sequence, the device operates in either Ready state or an Active state depending on customization in factory. The Ready state is a power down standby state when majority of internal circuitries are powered down. The **Active state** is the device active mode with all internal circuitries powered up. Writing to register bUSYS_CTRL can move the device between Ready state and Active state.

The following three power up options are available for factory configuration:

1. Ready state: The device is in the power down standby mode. Writing transaction to register bUSYS_CTRL is required to enable the device and enable the Active state.
2. Active state with bODIV_CTRL=0: The device generates frequency internally, but output is disabled. Writing transaction to register bODIV_CTRL is required to enable output driver to propagate output frequency.
3. Active state with bODIV_CTRL=1: The device is fully functional, and output is enabled. I²C intervention is not necessary.

In the state 1 and 2 above, user can write to I²C to change drive mode or central frequency from the factory set values prior to enabling the device output or before moving to Active state.

5.4 Changing Frequency and Output Driver Mode

After startup, the device is at the factory set frequency, output driver mode, and device state. The device is ready to be controlled by I²C register writes. All register writes can take immediate effect, except fUSER_FREQ and bDRV_MODE registers which require a follow on Apply command. User frequency register, "fUSER_FREQ", is a 4-byte big endian register representing frequency as a binary32 IEEE 754-2008 standard number, in [Hz] units. User driver mode register, "bDRV_MODE" is a single byte number representing driver mode described in Table 5.19.

Changing frequency and driver mode requires two steps:

1. Writing fUSER_FREQ and/or bDRV_MODE registers with the new desired central frequency and/or driver mode. Only the changing register needs to be written. Writing order in bytes is not essential. Writing these registers only records the new values, but does not invoke any internal processing.
2. Writing bUSYS_CTRL register with one of four Apply* commands. Once the Apply* command is accepted, the device uses the fUSER_FREQ and bDRV_MODE register values and invokes internal central frequency and/or driver mode.

Changing frequency can be completed by either disabling the output driver or keeping the output drive enabled. This is determined by applying different Apply* commands. The bUSYS_CTRL values and descriptions are shown in Table 5.2.

Table 5.2: System control register bUSYS_CTRL

| Value | Mnemonic | Description | stat_busy |
|--------|-------------------|---|-----------|
| 0 | UsysNop | No operation | -- |
| 1 | UsysReady | Ready standby power down state | -- |
| 2 | UsysActive | Active state | -- |
| 3 | <reserved> | No operation | -- |
| 4 | UsysRstSys | Invoke system reset restart | -- |
| 5 | UsysRstPor | Invoke power-on reset restart | -- |
| 6 | <reserved> | Do not use | -- |
| 7 | <reserved> | Do not use | -- |
| 8 | UsysApply | Apply: Disable output if active, update only changed frequency/driver mode | Check/Set |
| 9 | UsysApplyForce | Apply force: Disable output if active, force frequency/driver mode update | Check/Set |
| 10 | UsysApplyAct | Apply active: Keep output running if active, update only changed frequency/driver mode | Check/Set |
| 11 | UsysApplyActForce | Apply active force: Keep output running if active, force frequency/driver mode update | Check/Set |
| 12 | UsysRefresh | Refresh: Refresh fUSER_FREQ and bDRV_MODE register values to reflect the actual current device frequency and driver mode settings | Check/Set |
| Others | <reserved> | Do not use | -- |

Writing to “bUSYS_CTRL” register generates an acceptance request. When this command is accepted, the “bUSYS_CTRL” register is cleared. If a read from “bUSYS_CTRL” returns a non-zero value, it means that the writing command is still waiting to be accepted. A command acceptance happens in less than 5 us from the finishing of writing to the “bUSYS_CTRL” register. An accepted command can be either processed or ignored.

Commands other than “Apply*” and “Refresh” are always processed after the acceptance. It is possible to interrupt any “Apply*” or “Refresh” command under processing. A processing time depends on the command and its current state of the device, in most cases it is 1 ~ 5 us, but can be up to 160 us when the command is generated in the middle of the frequency change.

“Apply*” and “Refresh” commands are only processed when “bUSYS_STATS.stat_busy” = 0. When “bUSYS_STATS.stat_busy” = 1, although new “Apply*” and “Refresh” commands are accepted, they are ignored. The “bUSYS_STATS.stat_busy” = 1 status bit is the beginning of internal processing. The “bUSYS_STATS.stat_busy” = 0 occurs after this internal processing finishes. Changing frequency and output driver must be completed before invoking a new command.

Issuing a “Refresh” command reset the values of registers to match the actual state of the device. During a normal operation, the device keeps the values of fUSER_FREQ and bDRV_MODE synchronized with the internal state of the device. Reading these registers represent the actual frequency and output driver mode. However, the values could be different if writing occurs before applying the “Apply*” or “Refresh” command. If the “Apply*” command is not applied, then the registers value are the previous written values. Thus, issuing the “Refresh” command is necessary.

“fUSER_FREQ” and “bUSYS_STAT” registers are located back to back in I²C address space. As a result, the most used operation is done in a single I²C transaction as shown in Figure 5.2. This example shows changing the frequency to a 70 MHz, which corresponds to a 32-bit floating point number, 0x4c8583b0 in [Hz]. It is followed by a “Apply” command to invoke the internal frequency update, and assumed that the I²C register address autoincrement is enabled, which is a default.

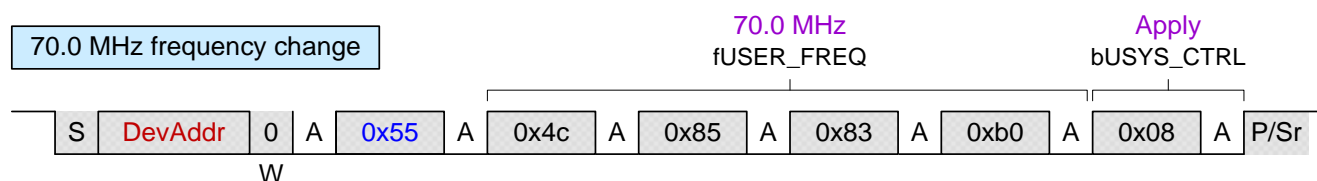


Figure 5.2: Frequency change I²C transaction

5.5 System Status

“bUSYS_STAT” is a read only system status register. It is readable any time and it reflects the current device status. All possible values of the register are listed in Table 5.3.

Table 5.3: System status bUSYS_STAT register values

| bUSYS_STAT | | | | | | | | Description |
|------------|-----|-----|------------|----------|-----------|-------------|------------|---------------------------------------|
| stat_busy | : | : | stat_trans | stat_rst | stat_tune | stat_active | stat_ready | |
| [7] | [6] | [5] | [4] | [3] | [2] | [1] | [0] | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | Start up |
| v | 0 | 0 | 0 | 0 | 0 | 0 | 1 | Ready state |
| v | 0 | 0 | 0 | 0 | 0 | 1 | 0 | Active state |
| v | 0 | 0 | 0 | 0 | 1 | 1 | 0 | Active state, tuning to new frequency |
| v | 0 | 0 | 1 | 0 | 0 | 0 | 1 | Ready → Active transition |
| v | 0 | 0 | 1 | 0 | 0 | 1 | 0 | Active → Ready transition |
| 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | Reset processing, shutting down |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | Reset accepted |

stat_busy bit [7] value v denotes value 0 or 1. The bit is 1 at the beginning of Apply* and Refresh commands internal processing and is 0 when the processing is done.

5.6 Output Drive Control

The output driver is powered up only in an **Active** state. CLKP and CLKM output signals are enabled or disabled by OE control register bits, "bUDRV_CTRL.udrv_oe_stop" and "bODIV_CTRL.odiv_oe_ena". The simplified control flow of output driver is shown in *Figure 5.3*.

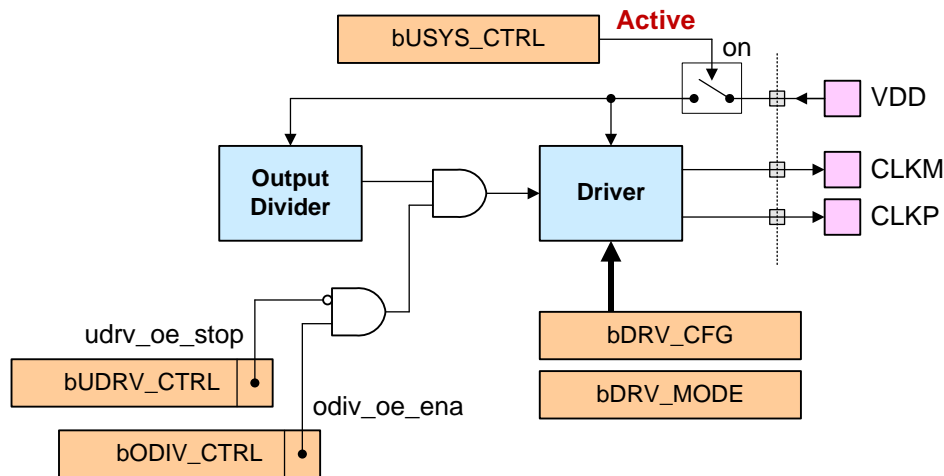


Figure 5.3: Driver control flow

Putting system to Ready state or disabling the driver with "bDRV_MODE" = 0 to issue one of the "Apply*" commands power down the output driver.

5.7 Center Frequency Control

Changing "fUSER_FREQ" register value controls the center frequency. The step does not apply to the DCXO feature.

Frequency change request shown in *Figure 5.4* and *Figure 5.5* marks the time when a "Apply*" command is written to a "bUSYS_CTRL" register which corresponds with the write of the last data bit to this "bUSYS_CTRL" register.

Figure 5.4 shows a timing of frequency change by using "Apply" command when a "fUSER_FREQ" register value is different from the current frequency; or using "ApplyForce" command with the output disabled during the frequency change.

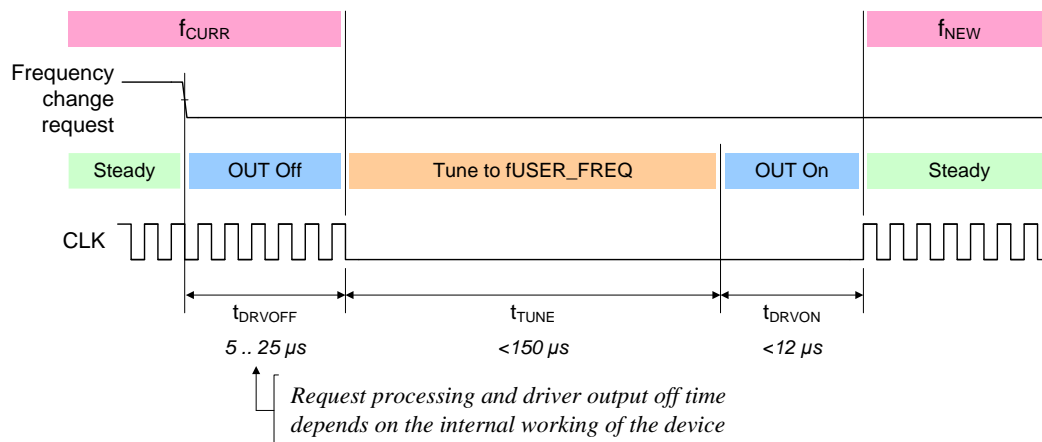


Figure 5.4: Frequency change by "Apply" command

Figure 5.5 shows either using an “ApplyAct” command when a “fUSER_FREQ” register has different value from the current frequency; or using an “ApplyActForce” command when the output keeps active during the frequency change. No signal glitches at the output occurs during the frequency change. However, there could be some periods in the output frequency corresponding when the current frequency and new frequency interval are not aligned. This behavior is device dependent and it also depends on the actual values and relationships in between the current and new frequencies.

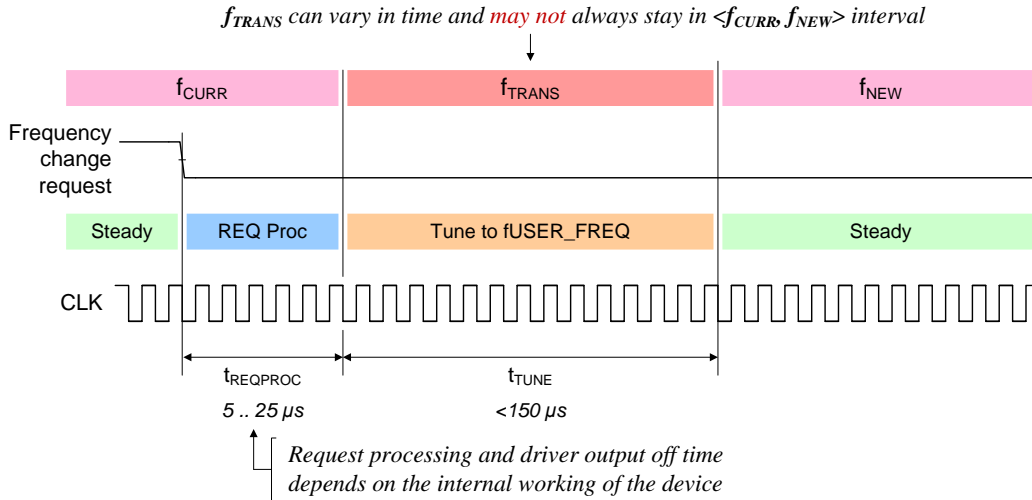


Figure 5.5: Frequency change by “ApplyAct” command

5.8 DCXO Control

The device incorporates a Digital Controlled Oscillator (DCXO) feature to allow seamless and fast setting of the frequency deviation from the center frequency. It implements a 32-bit signed “jDCXO_DATA” register in 11.21s signed bit format, which represents frequency offset from the current center frequency in “fUSER_FREQ” register in [0.9536743164 ppm] units. The [ppm] is a frequency step equal to $10^{-6} * f_{USER_FREQ}$ in [Hz]. The scaling factor is $0.9536743164 = 10^9/2^{20}$. It is possible to move the output frequency from the center frequency within the range ± 975 ppm.

The DCXO feature must be configured before writing to “bDCXO_CTRL” and “bDCXO_SHIFT” registers with an additional optional configuration, which is provided by “bVC_LPF_BW_DIR” and “bVC_LPF_ABS_SAT” registers. When a new frequency offset register “jDCXO_DATA” is written, the output signal changes to a new frequency seamlessly and glitchlessly. The settlement time of a new output frequency from the end of last data bit of the byte written to a “bDCXO_DATA” register is less than 8 us.

The DCXO data path configuration is schematically shown in Figure 5.6. Numerical values are shown in “NN.MM” notation where “NN” is the number of MSB bits representing integer part of the number, while MM represents number of LSB bits representing fractional part of the number. The letter ‘s’ means signed, and the letter ‘u’ is unsigned value. Signed values are represented in binary 2’s complement. The values are labeled in [ppm] in Figure 5.6 for simplicity. The numerical binary values must be multiplied by the factor of 0.9536743164 to represent the real [ppm] frequency offset value from the current center frequency.

Writing to a 32-bit “jDCXO_DATA” register normally requires an I²C transaction of 6 bytes. Every update of the DCXO value would take 6 byte writing on an I²C bus, which takes ~56 us if using an I²C Fast+ with 1 MHz speed, or ~140 us if using an I²C Fast 400 kHz speed. In some applications, faster update speed may be required while the full 32-bit frequency offset resolution is not needed. To accommodate faster DCXO offset updates using I²C, the DCXO has the following configurable features:

1. Data size: bDCXO_CTRL.dcxo_size = {0|1|2|3|4}
Select to write only 1, 2, 3, or all 4 bytes signed numbers, 0 means 4.
2. Left shift: bDCXO_SHIFT = <0, 24> range
Configurable shift of the entered input value to the left with up to 24 bits to scale small numbers to proper position within 32 bit number.
3. Data input:
 - Direct register mode: bDCXO_CTRL.dcxo_stream_mode = 0
Write an input value to the jDCXO_DATA bytes directly
 - Streaming mode: bDCXO_CTRL.dcxo_stream_mode = 1
Use only LSB byte of “jDCXO_DATA” as a streaming window. It requires to disable I²C register address autoincrement by writing “bI2C_INC_DIS” = 1 to allow DCXO offset updates in a very long I²C transaction, which is feeding only data bytes to the same register address.

4. Data application:

Absolute mode: bDCXO_CTRL.dcxo_rel_mode = 0

Apply the input offset value directly as a frequency offset.

Relative mode: bDCXO_CTRL.dcxo_rel_mode = 1

Apply the input offset value as a relative change from the current DCXO frequency offset in the internal 32-bit register, different from the "jDCXO_DATA" input register. This allows applying only incremental steps to move the internal frequency offset.

A DCXO register description in *Figure 5.6* and *t*

Table 5.10 provide more detailed descriptions.

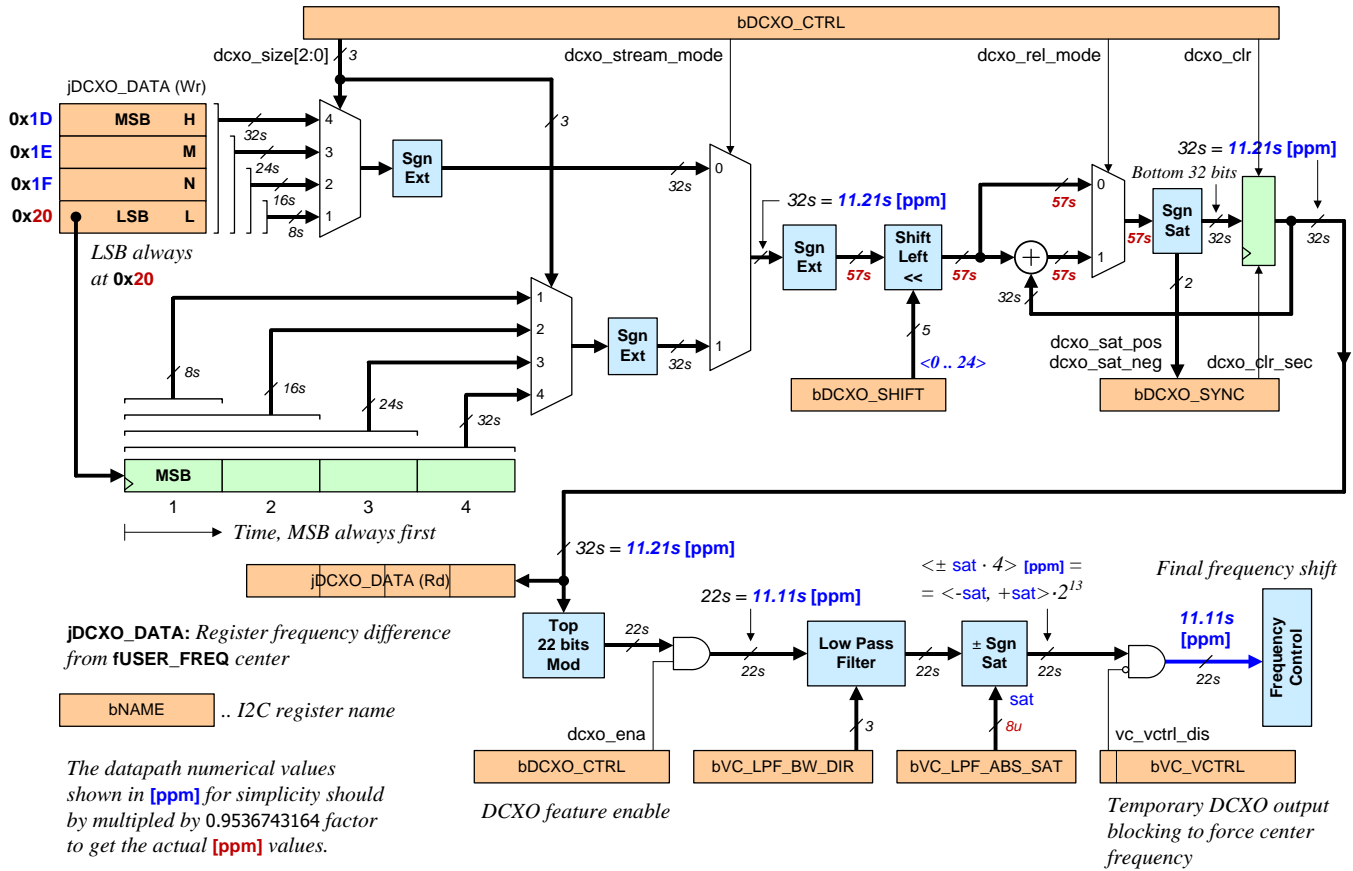


Figure 5.6: DCXO control and data processing

5.9 I²C Register Reference

The I²C interface is a byte-oriented interface. Registers are wider than 8 bits requiring to be split into multiple bytes located on subsequent register addresses. There are two types of multi byte registers:

1. Array of arrays, prefix **ab**, for example `abUSER_ID[4]` are organized in little endian fashion, byte `abUSER_ID[0]` is located at the lowest address, and the last byte of the array `abUSER_ID[3]` is located at the highest address.
2. An assigned 32-bit long integer, prefix **j**, and a 32-bit IEEE 754-2008 floating point number, prefix **f** are organized in big endian fashion, such that the most significant byte (MSB) is located at the lowest address.

Any unassigned byte registers have prefix **b**. Registers in the I²C address space `0x00..0xFF` without specifications in the register description table are reserved for factory use. All register bit fields labeled `<reserved>` are read only and are reserved for factory use. Read value should be ignored. Any register bits not specifically mentioned in “Bits” columns are unused. Writing to these registers is ignored and a read always returns a 0. Numerical values are either hexadecimal with `0x` prefix or decimal without any prefix.

Descriptions of register table columns:

| Register | Description |
|------------|---|
| Addr | I ² C register address. |
| Register | Register name. |
| Bytes | Number of bytes in the register. Wider than single byte registers span several subsequent bytes in an address space. |
| Endian | Endian for registers wider than one byte. Not applicable for single byte registers. |
| B | Big endian. The most significant byte (MSB) of the register appears at the lowest address. |
| L | Little endian. The least significant byte (LSB) of the register appears at the lowest address. |
| Field | Field name within the register. |
| <reserved> | Reserved for factory use and readable only |
| Bits | Field bit locations within a register. When the field spans more than one bit, the [M:L] notate. Denoting locations of a most significant bit M and a least significant bit L within the register are used. |
| Type | Field access type. |
| R/W | Read/write field by I ² C. |
| R | Read only field by I ² C. |
| W1 | Writing a 1 triggers an associated event, while writing a 0 has no effect. A read always returns 0. |
| Rst | Field reset value. The value 'Factory' indicates that the value depends on the individual device factory configuration. |

Table 5.4: I²C control registers summary

| Register | Addr | Bytes | Endian | Field | Bits |
|------------------|------|-------|--------|--------------------|--------|
| bI2C_DEV_ID | 0x00 | 1 | -- | i2c_dev_id | [7:0] |
| bI2C_DERIV_ID | 0x01 | 1 | -- | i2c_deriv_id | [7:0] |
| bI2C_ANA_ID | 0x02 | 1 | -- | i2c_ana_id | [7:0] |
| bI2C_SYS_ID | 0x03 | 1 | -- | i2c_sys_id | [7:0] |
| bI2C_INC_DIS | 0x06 | 1 | -- | i2c_inc_dis | [0] |
| abUSER_REG[4] | 0x07 | 4 | L | user_reg | [31:0] |
| bUDRV_CTRL | 0x0b | 1 | -- | udrv_oe_stop | [0] |
| | | | | udrv_run | [1] |
| bDEV_LDIS | 0x0c | 1 | -- | <reserved> | [0] |
| | | | | <reserved> | [4] |
| | | | | dcxo_ldis | [6] |
| | | | | <reserved> | [7] |
| bDRV_LDIS | 0x0d | 1 | -- | drv_cmos_clkp_ldis | [0] |
| | | | | drv_cmos_clkm_ldis | [1] |
| | | | | drv_diff_ldis | [2] |
| bDCXO_SYNC | 0x1a | 1 | -- | dcxo_stream_sync | [0] |
| | | | | dcxo_clr_sec | [1] |
| | | | | dcxo_sat_neg | [6] |
| | | | | dcxo_sat_pos | [7] |
| bDCXO_SHIFT | 0x1b | 1 | -- | dcxo_shift | [4:0] |
| bDCXO_CTRL | 0x1c | 1 | -- | dcxo_size | [2:0] |
| | | | | dcxo_stream_mode | [4] |
| | | | | dcxo_rel_mode | [5] |
| | | | | dcxo_ena | [6] |
| | | | | dcxo_clr | [7] |
| jDCXO_DATA | 0x1d | 4 | B | dcxo_data | [31:0] |
| abFACT_ID[12] | 0x21 | 12 | L | fact_id | [95:0] |
| abCFG_ID[6] | 0x2d | 6 | L | cfg_id | [47:0] |
| abUSER_ID[4] | 0x33 | 4 | L | user_id | [31:0] |
| abUSER_INFO[2] | 0x37 | 2 | L | user_info | [15:0] |
| bDRV_CFG | 0x3b | 1 | -- | drv_stop_hiz | [0] |
| | | | | drv_cmos_high_str | [1] |
| | | | | drv_cmos_slew | [3:2] |
| | | | | drv_cmos_clkp_inv | [4] |
| | | | | drv_cmos_clkm_inv | [5] |
| bODIV_CTRL | 0x3c | 1 | -- | odiv_oe_ena | [0] |
| bVC_LPF_BW_DIR | 0x41 | 1 | -- | vc_lpf_bw_dir | [2:0] |
| bVC_LPF_ABS_SAT | 0x42 | 1 | -- | vc_lpf_abs_sat | [7:0] |
| bVC_VCTRL | 0x43 | 1 | -- | vc_vctrl_dis | [0] |
| bPADV_DIR_DATA | 0x44 | 1 | -- | padv_dir_data | [7:0] |
| bGPIO_PUP_ENA | 0x45 | 1 | -- | gpio_pup_ena | [5:0] |
| bGPIO_PDWN_ENA | 0x46 | 1 | -- | gpio_pdwn_ena | [5:0] |
| bGPIO_PULL5K_ENA | 0x47 | 1 | -- | gpio_pull5k_ena | [5:0] |
| bDRV_MODE | 0x54 | 1 | -- | drv_mode | [7:0] |
| fUSER_FREQ | 0x55 | 4 | B | user_freq | [31:0] |
| bUSYS_CTRL | 0x59 | 1 | -- | usys_ctrl | [7:0] |
| bUSYS_STAT | 0x5a | 1 | -- | stat_ready | [0] |
| | | | | stat_active | [1] |
| | | | | stat_tune | [2] |
| | | | | stat_rst | [3] |
| | | | | stat_tran | [4] |
| | | | | stat_busy | [7] |

Table 5.5: Factory device identification

| Register | Addr | Field | Bits | Type | Rst | Description |
|---------------|------|--------------|-------|------|---------|---------------------------|
| bI2C_DEV_ID | 0x00 | i2c_dev_id | [7:0] | R | 0x84 | Device identification |
| bI2C_DERIV_ID | 0x01 | i2c_deriv_id | [7:0] | R | Factory | Derivative identification |
| bI2C_ANA_ID | 0x02 | i2c_ana_id | [7:0] | R | Factory | Analog identification |
| bI2C_SYS_ID | 0x03 | i2c_sys_id | [7:0] | R | Factory | System identification |

Table 5.6: I²C address autoincrement control

| Register | Addr | Field | Bits | Type | Rst | Description |
|--------------|------|-------------|------|------|-----|---|
| bI2C_INC_DIS | 0x06 | i2c_inc_dis | [0] | R/W | 0 | Disable automatic I ² C register address increment. If this bit is set the automatic I ² C register address increment after each write or a read data is disabled. This is useful in situations when the user wants to repeatedly read a same address location. For example, monitoring some status bit or using the DCXO feature in a single I ² C transaction. |

Table 5.7: User scratch pad byte array

| Register | Addr | Field | Bits | Type | Rst | Description |
|---------------|------|----------|--------|------|-----|--|
| abUSER_REG[4] | 0x07 | user_reg | [31:0] | R/W | 0 | User scratch pad R/W byte array. Controls no hardware. |

Table 5.8: Output driver user control

| Register | Addr | Field | Bits | Type | Rst | Description |
|------------|------|--------------|------|------|-----|---|
| bUDRV_CTRL | 0x0b | udrv_oe_stop | [0] | R/W | 0 | Stop the output driver. This bit is an inverse of the enabling the output driver bit "bODIV_CTRL.odiv_oe_ena". An active high control bit disables the output driver. |
| | | udrv_run | [1] | R | 0 | Status bit of the output driver. Read only. If 1, then the device actively generates signal to the output driver. Output driver then can be enabled or disabled to propagate its input signal to outputs. |

Table 5.9: Feature lock or disable status

| Register | Addr | Field | Bits | Type | Rst | Description |
|-----------|------|--------------------|------|------|---------|--|
| bDEV_LDIS | 0x0c | <reserved> | [0] | R | Factory | Reserved for factory use. |
| | | <reserved> | [4] | R | Factory | Reserved for factory use. |
| | | dcxo_ldis | [6] | R | 0 | DCXO feature status. If the bit is 1 then the DCXO feature is factory blocked and cannot be enabled. |
| | | <reserved> | [7] | R | Factory | Reserved for factory use. |
| bDRV_LDIS | 0x0d | drv_cmos_clkp_ldis | [0] | R | 0 | CMOS CLK+ half of CMOS driver is factory disabled. |
| | | drv_cmos_clkm_ldis | [1] | R | 0 | CMOS CLK- half of CMOS driver is factory disabled. |
| | | drv_diff_ldis | [2] | R | 0 | Differential output driver is factory disabled. |

Table 5.10: DCXO control

| Register | Addr | Field | Bits | Type | Rst | Description |
|-------------|------|------------------|-------|------|-----|--|
| bDCXO_SYNC | 0x1a | dcxo_stream_sync | [0] | W1 | 0 | Writing a 1 to this bit clears the streaming mode synchronization logic. The next write to the bDCXO_DATAL register byte is treated as the MSB byte of the burst. This bit allows to forcibly synchronize the streaming operation when in doubt. Note that any write to bDCXO_CTRL register also clears the streaming mode synchronization logic. |
| | | dcxo_clr_sec | [1] | W1 | 0 | Secondary duplicated clear pulse bit for user convenience, same function as bDCXO_CTRL.dcxo_clr bit. See the bDCXO_CTRL description. The clear affects only the DCXO data processing and not the DCXO I ² C control registers. All the current DCXO settings as set in bDCXO_CTRL and bDCXO_SHIFT registers stay intact. |
| | | dcxo_sat_neg | [6] | R | 0 | If 1, then DCXO frequency deviation internal value has saturated to minimum negative value. Register bit updated every time when the 32-bit DCXO internal value is updated. The bit is not sticky. Cleared by bDCXO_SYNC.dcxo_clr_sec = 1 or bDCXO_CTRL.dcxo_clr = 1. |
| | | dcxo_sat_pos | [7] | R | 0 | If 1, then DCXO frequency deviation internal value has saturated to maximum positive value. Register bit updated every time the 32-bit DCXO internal value is updated. The bit is not sticky. Cleared by bDCXO_SYNC.dcxo_clr_sec = 1 or bDCXO_CTRL.dcxo_clr = 1. |
| bDCXO_SHIFT | 0x1b | dcxo_shift | [4:0] | R/W | 0 | Number of bits to shift the input jDCXO_DATA value to the left before internal processing. The left bit shift is always applied at the input DCXO value in all modes and configurations. The shift value is in the range of <0, 24> bits. If the shift number is greater than 24, then it is internally forced to 24. |
| bDCXO_CTRL | 0x1c | dcxo_size | [2:0] | R/W | 0 | Number of streamed bytes per input value or number of bottom jDCXO_DATA bytes to be used as input value in direct register access mode. The valid values are 0, 1, 2, 3, and 4 bytes. Default value 0 is an alias of value 4. The value greater than 4 is internally forced to 4. All values are internally sign-extended to 57 bits before the value is used. The bDCXO_SHIFT left shift is applied on the internally sign-extended 57-bit number. For streamed mode the number denotes the size of ordered byte burst in the I ² C transaction: 4 .. {H M N L} .. 32 bits, the last L byte is always LSB byte 3 .. {M N L} .. 24 bits 2 .. {N L} .. 16 bits 1 .. {L} .. 8 bits For direct register mode, this number denotes the number of LSB bytes from the jDCXO_DATA. It is used to form a number to be internally sign-extended to 57 bits. The number of bytes and their order is the same as in the streaming table above. The {H} denotes the MSB byte at byte address jDCXO_DATA while the {L} denotes the LSB byte at the byte address jDCXO_DATA + 3. When used as 4 bytes, the jDCXO_DATA is a 32-bit signed integer stored in a big endian fashion. MSB is at lower address. The bDCXO_DATAL byte at byte address jDCXO_DATA + 3 is always the LSB byte for all widths and could be viewed as the value alignment anchor. |
| | | dcxo_stream_mode | [4] | R/W | 0 | DCXO frequency offset value input mode. 0 .. regular direct register mode. For situations when I ² C address autoincrement is active (default mode), the jDCXO_DATA four byte array can be written any time. Whenever the bDCXO_DATAL location at the jDCXO_DATA + 3 byte address is written, the number of bytes is selected by bDCXO_CTRL.dcxo_size number is taken from the bottom of the jDCXO_DATA value, aligned towards bDCXO_DATAL location as LSB byte. Those selected bytes are sign-extended to 57 bits and left shifted by the bDCXO_SHIFT value. The resulting number is then either used to add to the existing internal DCXO value in 57 bit |

| Register | Addr | Field | Bits | Type | Rst | Description |
|------------|------|---------------|--------|------|-----|--|
| | | | | | | <p>addition, thus the sum is then sign saturated to bottom 32 bits to be used as a new DCXO value, or the value is sign saturated to 32 bits to be used directly as the new DCXO value.</p> <p>1 .. streaming mode. For situations when I²C register address autoincrement is disabled, user must write a register bI2C_INC_DIS = 1 to actively disable the I²C register address autoincrement. The setting is global and applies to all I²C register accesses, not only to the DCXO data registers.</p> <p>The user must choose to stream 4 {H M N L}, 3 {M N L}, 2 {N L}, or 1 {L} byte by setting the bDCXO_SIZE register before starting streaming operation. The stream byte order is from left to right and the LSB is always come last in the value burst. All data is written to bDCXO_DATAH register only. Writing to other jDCXO_DATA register bytes are ignored. The streamed byte value is then treated the same way as in the regular direct register mode described above.</p> <p>The streaming mode is useful for very fast DCXO value updates, possibly in a single and very long I²C transaction.</p> |
| | | dcxo_rel_mode | [5] | R/W | 0 | <p>DCXO frequency offset value usage mode.</p> <p>0 .. absolute mode. The user signed DCXO input value obtained using the current access mode is sign-extended to 57 bits and then left shifted by bDCXO_SHIFT bits. The shifted signed value is sign-saturated to bottom 32 bits and used as the internal DCXO frequency offset value.</p> <p>1 .. relative mode. The user signed DCXO input value obtained using the current access mode is sign-extended to 57 bits and then left shifted by bDCXO_SHIFT bits. The shifted signed value is added to the current internal 32-bit DCXO value resulting in 57-bit internal signed addition number. The signed 57-bit sum is sign saturated to bottom 32 bits and then used as the internal DCXO frequency offset value.</p> <p>If the bottom 32-bit saturation happened, the read only saturation flags bDCXO_SYNC.dcxo_sat_pos and bDCXO_SYNC.dcxo_sat_neg reflect the saturation operation of the last update of the internal DCXO frequency offset value.</p> |
| | | dcxo_ena | [6] | R/W | 0 | <p>Enable the DCXO operation.</p> <p>If this bit is 1, then the DCXO operation is enabled. If this bit is 0, then the DCXO operation is disabled. All control bits in bDCXO_CTRL register are ignored with an exception of bDCXO_CTRL.dcxo_clr, which can be written as 1 to clear DCXO processing module.</p> |
| | | dcxo_clr | [7] | W1 | 0 | <p>Writing 1 to this bit clears all the DCXO internal frequency offset value registers, all DCXO value holding registers, and streaming mode synchronization logic. All mentioned registers are cleared even if bDCXO_CTRL.dcxo_ena = 0. The clear does not affect bDCXO_SHIFT register.</p> <p>Note that the streaming mode synchronization logic is cleared every time the bDCXO_CTRL register is written, even if the bDCXO_CTRL.dcxo_ena = 0 or bDCXO_CTRL.dcxo_clr = 0. It is highly recommended that this bit is written as 1 when the DCXO is being enabled by setting bDCXO_CTRL.dcxo_ena = 1 to clear the internal processing registers.</p> |
| jDCXO_DATA | 0x1d | dcxo_data | [31:0] | R/W | 0 | <p>DCXO data register through which the actual signed 32 bit DCXO control data is applied. Increasing value increases the output frequency. The 32-bit input value represents 11.21s signed 2's complement number in [0.9536743164 ppm] units as deviation from current central frequency.</p> <p>The 4 register bytes can be viewed separately as bDCXO_DATAH, bDCXO_DATAM, bDCXO_DATAN, and bDCXO_DATAH data bytes and writing to them depends on the write mode selected.</p> |

| Register | Addr | Field | Bits | Type | Rst | Description |
|----------|------|-------|------|------|-----|--|
| | | | | | | <p>bDCXO_DATAH .. jDCXO_DATA address + 0 bDCXO_DATAM .. jDCXO_DATA address + 1 bDCXO_DATAN .. jDCXO_DATA address + 2 bDCXO_DATA[7:0] .. jDCXO_DATA address + 3 .. always LSB</p> <p>The bDCXO_DATA[7:0] LSB byte refers to the jDCXO_DATA[7:0] bits, which is the byte at the highest address (jDCXO_DATA address + 3) since the register value is represented in big endian. MSB is at lower address. There are holding registers for 3 MSB upper value bytes and the written value will get applied internally only if the bDCXO_DATA[7:0] byte is written.</p> <p>In a regular direct register access mode, bDCXO_CTRL.dcxo_stream_mode = 0, any byte in the 4-byte jDCXO_DATA register can be written at any time. The most convenient way is to write it as a burst of bytes with incrementing byte address after each byte write. The written value is applied internally only after the last bDCXO_DATA[7:0] byte is written. The user DCXO input value is formed as a signed selection of bottom N bytes (N = DCXO_CTRL.dcxo_size), where bDCXO_DATA[7:0] is always the LSB byte of the signed number.</p> <p>In a streaming access mode, bDCXO_CTRL.dcxo_stream_mode = 1, the bDCXO_DATA[7:0] byte is used to stream the data. Writing to the rest of the bytes in the jDCXO_DATA register is ignored in streaming mode. The user could decide to stream 4 {H M N L}, 3 {M N L}, 2 {N L}, or 1 {L} byte, all in big endian fashion. After the last byte of the value burst is written, the signed numerical value represented by 4, 3, 2, or 1 byte based on the streaming number configuration form the input user DCXO value.</p> <p>Reading from jDCXO_DATA register always returns current signed internal 32-bit DCXO value, which is applied to the input of the DCXO frequency control low pass filter in [0.9536743164 ppm] units as deviation from current central frequency. The read value does not depend on any write mode configuration and is always the current internal DCXO frequency offset value. This also applies in the streaming mode. The streaming mode applies only for value writes. To read the current 32-bit internal value even in streaming mode the user needs to read all 4 bytes of this register to get the 32-bit current internal value. See <i>Figure 5.6</i>.</p> |

Table 5.11: Factory identification

| Register | Addr | Field | Bits | Type | Rst | Description |
|---------------|------|---------|--------|------|---------|--|
| abFACT_ID[12] | 0x21 | fact_id | [95:0] | R | Factory | Factory device identification. |
| abCFG_ID[6] | 0x2d | cfg_id | [47:0] | R | Factory | Factory device configuration identification. |

Table 5.12: User identification

| Register | Addr | Field | Bits | Type | Rst | Description |
|----------------|------|-----------|--------|------|---------|--|
| abUSER_ID[4] | 0x33 | user_id | [31:0] | R | Factory | Factory user identification, optional. |
| abUSER_INFO[2] | 0x37 | user_info | [15:0] | R | Factory | User specified information, set in factory per user request for user specific device identification. Optional. Reset to 0 if not used. |

Table 5.13: Output driver configuration

| Register | Addr | Field | Bits | Type | Rst | Description |
|----------|------|-------------------|-------|------|-----|--|
| bDRV_CFG | 0x3b | drv_stop_hiz | [0] | R/W | 0 | Setting to 1, the driver is tri-stated when stopped. |
| | | drv_cmos_high_str | [1] | R/W | 0 | Enable CMOS driver to be 35 Ω output impedance rather than default 50 Ω. 0 .. 50 Ω 1 .. 35 Ω |
| | | drv_cmos_slew | [3:2] | R/W | 0 | CMOS driver output slew rate control. ($C_{L-Ext} = 10$ pF) 0 .. none 1 .. 9 ns 2 .. 6.4 ns 3 .. 4 ns |
| | | drv_cmos_clkp_inv | [4] | R/W | 0 | Invert clock polarity for CMOS CLK+ driver. It also determines polarity of the output when CMOS driver output is disabled. |
| | | drv_cmos_clkm_inv | [5] | R/W | 0 | Invert clock polarity for CMOS CLK- driver. It also determines polarity of the output when CMOS driver output is disabled. |

Table 5.14: User output divider enable

| Register | Addr | Field | Bits | Type | Rst | Description |
|------------|------|-------------|------|------|-----|---|
| bODIV_CTRL | 0x3c | odiv_oe_ena | [0] | R/W | 0x0 | Output divider and driver output enable. Enable the output driver by following condition: bODIV_CTRL.odiv_oe_ena = 1 and bUDRV_CTRL.udrv_oe_stop = 0. (If the condition is not met then the driver output is disabled.) The functionality of bODIV_CTRL.odiv_oe_ena and bUDRV_CTRL.udrv_oe_stop are inverse of each other for user convenience to use either an enable or a disable polarity. |

Table 5.15: DCXO low pass filter configuration

| Register | Addr | Field | Bits | Type | Rst | Description |
|-----------------|------|----------------|-------|------|-----|--|
| bVC_LPF_BW_DIR | 0x41 | vc_lpf_bw_dir | [2:0] | R/W | 7 | Frequency offset low pass filter bandwidth control. The LPF bandwidth formula: $BW = 1166 * 2^{vc_lpf_bw_dir} \text{ [Hz]}$ which corresponds to the following values: 0 .. 1.2 kHz 1 .. 2.3 kHz 2 .. 4.7 kHz 3 .. 9.3 kHz 4 .. 18.7 kHz 5 .. 37.3 kHz 6 .. 74.6 kHz 7 .. pass through .. default value |
| bVC_LPF_ABS_SAT | 0x42 | vc_lpf_abs_sat | [7:0] | R/W | 255 | An unsigned absolute saturation value of a frequency offset to user limits the maximum frequency offset swing. The unsigned 8-bit value is shifted left by 13 bits from 10.11 us unsigned absolute saturation value of the 11.11 s output. The 22-bit signed output will be saturated to fall in the value range $<-vc_lpf_abs_sat, +vc_lpf_abs_sat > * 2^{13}$. See <i>Figure 5.6</i> . The value 0 squelches the frequency offset and forces central frequency while value 255 allows full frequency offset range to be utilized. |
| bVC_VCTRL | 0x43 | vc_vctrl_dis | [0] | R/W | 0 | Disable the DCXO frequency control output and force the value to 0. This feature is to provide temporary disable for DCXO datapath to see how the device is doing while keeping the DCXO setting unchanged and the datapath active. See <i>Figure 5.6</i> . |

Table 5.16: Advanced configuration direct data

| Register | Addr | Field | Bits | Type | Rst | Description |
|----------------|------|---------------|-------|------|-----|---|
| bPADV_DIR_DATA | 0x44 | padv_dir_data | [7:0] | R/W | 0 | Control of factory programmed advanced configurations. Do not write unless specified by the manufacturer. |

Table 5.17: GPIO user configuration

| Register | Addr | Field | Bits | Type | Rst | Description |
|------------------|------|-----------------|-------|------|-----|--|
| bGPIO_PUP_ENA | 0x45 | gpio_pup_ena | [5:0] | R/W | 0 | GPIO pull up 5/50 k Ω resistor enable for all GPIO[5:0]. |
| bGPIO_PDWN_ENA | 0x46 | gpio_pdwn_ena | [5:0] | R/W | 0 | GPIO pull down 5/50 k Ω resistor enable for all GPIO[5:0]. |
| bGPIO_PULL5K_ENA | 0x47 | gpio_pull5k_ena | [5:0] | R/W | 0 | GPIO pull up/down 5 k Ω resistor select for all GPIO[5:0]. If set then the GPIO_PUP_ENA and GPIO_PDWN_ENA for the corresponding GPIO will use 5 k Ω resistors rather than the default 50 k Ω ones. This register chooses the pull up/down resistor value only. It does not turn the resistors on or off. The bGPIO_PUP_ENA and bGPIO_PDWN_ENA registers control the actual resistor on/off switches. |

Table 5.18: User system control

| Register | Addr | Field | Bits | Type | Rst | Description |
|------------|------|-------------|--------|------|-----|---|
| bDRV_MODE | 0x54 | drv_mode | [7:0] | R/W | 0 | Desired driver configuration. See Table 5.19 for encoding. I ² C write to bDRV_MODE has no immediate effect. The user is required to write a command to bUSYS_CTRL register to invoke the driver mode change process. |
| fUSER_FREQ | 0x55 | user_freq | [31:0] | R/W | 0 | Floating point binary32 IEEE 754-2008 number in big endian representation with the most significant byte at the lowest address. It represents user desired output center frequency in [Hz]. I ² C write to fUSER_FREQ has no immediate effect. The user is required to write a command to bUSYS_CTRL register to invoke the frequency change process. |
| bUSYS_CTRL | 0x59 | usys_ctrl | [7:0] | R/W | 0 | User system control. Command byte to be written whenever the frequency change, driver mode change, or special system action is required. See <i>After startup</i> , the device is at the factory set frequency, output driver mode, and device state. The device is ready to be controlled by I ² C register writes. All register writes can take immediate effect, except fUSER_FREQ and bDRV_MODE registers which require a follow on Apply command. User frequency register, “fUSER_FREQ”, is a 4-byte big endian register representing frequency as a binary32 IEEE 754-2008 standard number, in [Hz] units. User driver mode register, “bDRV_MODE” is a single byte number representing driver mode described in Table 5.19. Changing frequency and driver mode requires two steps: <ul style="list-style-type: none"> 3. Writing fUSER_FREQ and/or bDRV_MODE registers with the new desired central frequency and/or driver mode. Only the changing register needs to be written. Writing order in bytes is not essential. Writing these registers only records the new values, but does not invoke any internal processing. 4. Writing bUSYS_CTRL register with one of four Apply* commands. Once the Apply* command is accepted, the device uses the fUSER_FREQ and bDRV_MODE register values and invokes internal central frequency and/or driver mode. Changing frequency can be completed by either disabling the output driver or keeping the output drive enabled. This is determined by applying different Apply* commands. The bUSYS_CTRL values and descriptions are shown in Table 5.2. Table 5.2 for command values. |
| bUSYS_STAT | 0x5a | stat_ready | [0] | R | 0 | Ready state reached |
| | | stat_active | [1] | R | 0 | Active state reached |
| | | stat_tune | [2] | R | 0 | In Active tuning to a new frequency. If 1, then stat_active = 1. |
| | | stat_rst | [3] | R | 0 | Reset request accepted |
| | | stat_tran | [4] | R | 0 | In system state transition. Based on the initial state the device is transitioning from, bUSYS_STAT.stat_ready, bUSYS_STAT.stat_active, or bUSYS_STAT.stat_rst is set to 1. |
| | | stat_busy | [7] | R | 0 | Busy processing tuning to a new frequency and/or driver mode change. This bit is checked, set, and cleared only by Apply* and Refresh commands. |

Table 5.19: Driver mode bDRV_MODE value encoding

| Value | Description |
|----------|---|
| 0 | Driver disabled |
| 1 | CMOS single CLK+ |
| 2 | CMOS single CLK- |
| 3 | CMOS dual |
| 4 | LVDS 2.5 & 3.3 V |
| 5 | LVDS 1.8 V |
| 6 | Reserved |
| 7 | Reserved |
| 8 | HCSL 50 Ω external |
| 9 | Reserved |
| 10 | HCSL 42.5 Ω external |
| 11 | Reserved |
| 12 | Reserved |
| 13 | LVPECL AC-coupled low power |
| 14 | Reserved |
| 15 | CML 50 Ω internal |
| 16 | Reserved |
| 17 | Reserved |
| 18 .. 31 | Reserved, behaves as 0, driver will be disabled |
| Others | Unused, driver setting will stay unchanged |

6. Package Outline

6.1 Package Outline (3.2 x 5 mm)

The figure below illustrates the package details for the 3.2 x 5 mm AS5003. The table below lists the values for the dimensions shown in the illustration.

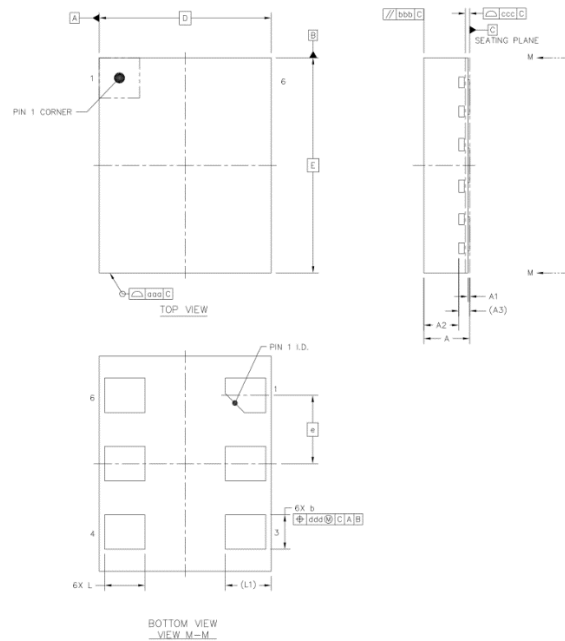


Figure 6.1: AS5003 (3.2 x 5 mm) Outline Diagram

Table 6.1: Package Diagram Dimensions (mm)

| Dimension | Min | Nom | Max |
|-----------|-----------|-------|------|
| A | 0.8 | 0.85 | 0.9 |
| A1 | 0 | 0.035 | 0.05 |
| A2 | -- | 0.65 | -- |
| A3 | 0.203 REF | | |
| b | 0.59 | 0.64 | 0.69 |
| D | 3.2 BSC | | |
| E | 4 BSC | | |
| e | 1.27 BSC | | |
| L | 0.7 | 0.75 | 0.8 |
| L1 | 0.85 REF | | |
| aaa | 0.1 | | |
| bbb | 0.1 | | |
| ccc | 0.08 | | |
| ddd | 0.1 | | |

Notes:

1. All dimensions shown are in millimeters (mm) unless otherwise noted.
2. Dimensioning and Tolerancing per ANSI Y14.5M-1994.

6.2 Package Outline (2.5 x 3.2 mm)

The figure below illustrates the package details for the 2.5 x 3.2 mm AS5003. The table below lists the values for the dimensions shown in the illustration.

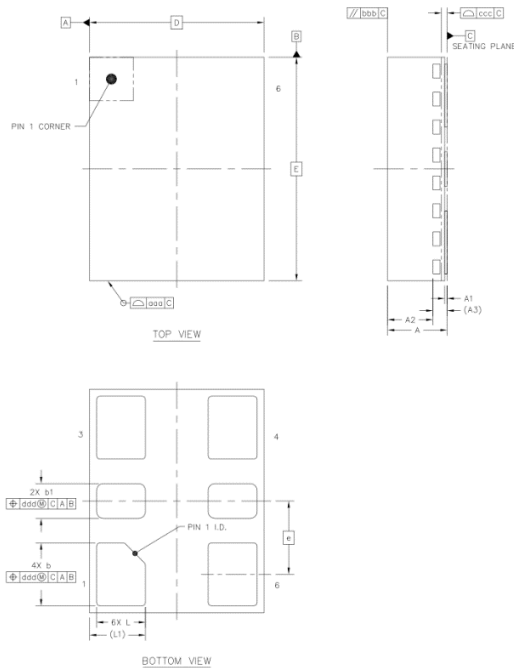


Figure 6.2: AS5003 (2.5 x 3.2 mm) Outline Diagram

Table 6.2: Package Diagram Dimensions (mm)

| Dimension | Min | Nom | Max |
|-----------|-----------|-------|------|
| A | 0.8 | 0.85 | 0.9 |
| A1 | 0 | 0.035 | 0.05 |
| A2 | --- | 0.65 | --- |
| A3 | 0.203 REF | | |
| b | 0.85 | 0.9 | 0.95 |
| b1 | 0.45 | 0.5 | 0.55 |
| D | 2.5 BSC | | |
| E | 3.2 BSC | | |
| e | 1.05 BSC | | |
| L | 0.65 | 0.7 | 0.75 |
| aaa | 0.1 | | |
| bbb | 0.1 | | |
| ccc | 0.08 | | |
| ddd | 0.1 | | |

Notes:

1. The dimensions in parentheses are reference.
2. All dimensions in millimeters (mm).
3. Dimensioning and Tolerancing per ANSI Y14.5M-1994.

7. PCB Land Pattern (3.2 x 5 mm and 2.5 x 3.2 mm)

The figure below illustrates the 3.2 x 5 mm PCB land pattern for the AS5003. The table below lists the values for the dimensions shown in the illustration.

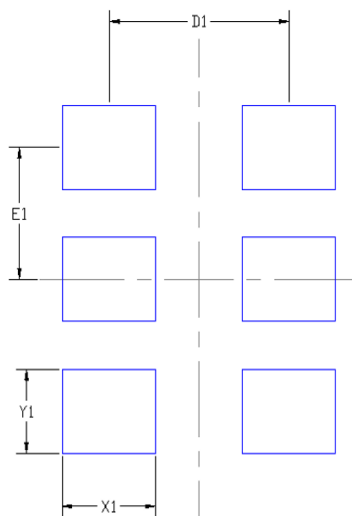


Figure 7.1: AS5003 (3.2 x 5 mm and 2.5 x 3.2 mm) PCB Land Pattern

Table 7.1: PCB Land Pattern Dimensions (mm)

| Dimension | Description | 3.2 x 5 mm Package Value (mm) | 2.5 x 3.2 mm Package Value (mm) |
|-----------|-------------------------------------|-------------------------------|---------------------------------|
| X1 | Width - leads on long sides | 0.75 | 0.7 |
| Y1 | Height - leads on long sides | 0.64 | 0.9, 0.5 |
| D1 | Pitch in X directions of XLY1 leads | 2.25 | 1.6 |
| E1 | Lead pitch XLY1 leads | 1.27 | 1.05 |

Notes: The following notes and stencil design are shared as recommendations only. A customer or user may find it necessary to use different parameters and fine-tune their SMT process as required for their application and tooling.

General

1. All dimensions shown are in millimeters (mm) unless otherwise noted.
2. Dimensioning and Tolerancing is per the ANSI Y14.5M-1994 specification.
3. This Land Pattern Design is based on the IPC-7351 guidelines.
4. All dimensions shown are at Maximum Material Condition (MMC). Least Material Condition (LMC) is calculated based on a Fabrication Allowance of 0.05 mm.

Solder Mask Design

1. All metal pads are to be non-solder mask defined (NSMD). Clearance between the solder mask and the metal pad is to be 60 μm minimum, all the way around the pad.

Stencil Design

1. A stainless steel, laser-cut and electro-polished stencil with trapezoidal walls should be used to assure good solder paste release.
2. The stencil thickness should be 0.125 mm (5 mils).
3. The ratio of stencil aperture to land pad size should be 0.8:1 for the pads.

Card Assembly

1. A No-Clean, Type-3 solder paste is recommended.
2. The recommended card reflow profile is per the JEDEC/IPC J-STD-020 specification for Small Body Components.

8. Top Marking (3.2 x 5 mm and 2.5 x 3.2 mm)

The figure below illustrates the mark specification for the AS5003. The table below lists the line information.

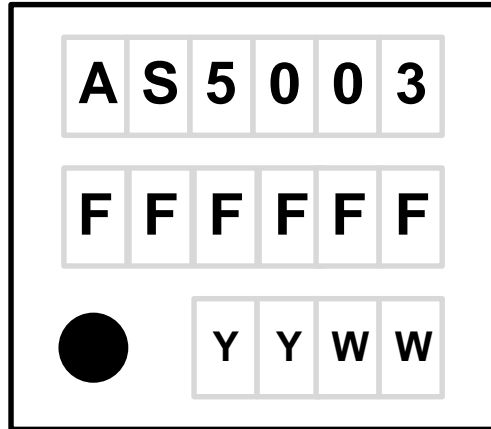


Figure 8.1: AS5003 Top Mark

Table 8.1: AS5003 Top Mark Description

| Line | Position | Description |
|------|-------------------|---|
| 1 | 1–6 | Device name |
| 2 | Trace Code | |
| | 1–6 | Field ID (Default = 000000) |
| 3 | Position 1 | Pin 1 orientation mark (dot) |
| | Position 2–3 | Year (last two digits of the year), to be assigned by assembly site (ex: 2017 = 17) |
| | Position 4–5 | Calendar Work Week number (1–53), to be assigned by assembly site |

9. Revision History

Revision 0.9

Sept 2020

- Initial release