# ZL30264-ZL30267

## 2-APLL, 6- or 10-Output Any-to-Any Clock Multiplier and Frequency Synthesizer

Data Sheet

October 2018

## Features

- **Four Flexible Input Clocks**
  - One crystal/CMOS input
  - Two differential/CMOS inputs
  - One single-ended/CMOS input
  - Any input frequency from 9.72MHz to 1.25GHz (300MHz max for CMOS)
  - Activity monitors, automatic or manual switching
  - Glitchless clock switching by pin or register
- **6 or 10 Any-Frequency, Any-Format Outputs**
  - Any output frequency from 1Hz to 1045MHz
  - 2 fractional-N APLLs with 0ppm error
  - **Each APLL has a fractional divider and an integer divider to make a total of four independent frequency families**
  - Output jitter from integer multiply and dividers as low as 0.17ps RMS (12kHz-20MHz)
  - Output jitter from fractional dividers is typically < 1ps RMS, many frequencies <0.5ps RMS
  - Each output has an independent divider
  - Each output configurable as LVDS, LVPECL, HCSL, 2xCMOS or HSTL
  - In 2xCMOS mode, the P and N pins can be different frequencies (e.g. 125MHz and 25MHz)
  - Multiple output supply voltage banks with CMOS output voltages from 1.5V to 3.3V
  - Precise output alignment circuitry and per-output phase adjustment
  - Per-output enable/disable and glitchless start/stop (stop high or low)
- **General Features**
  - Automatic self-configuration at power-up from external (ZL30264 or 6) or internal (ZL30265 or 7) EEPROM; up to 8 configurations pin-selectable
  - External feedback for zero-delay applications
  - Numerically controlled oscillator mode
  - Spread-spectrum modulation mode
  - Generates PCIe 1, 2, 3, 4 compliant clocks
  - Easy-to-configure design requires no external VCXO or loop filter components
  - SPI or $I^2C$ processor Interface
  - Core supply voltage options: 2.5V only, 3.3V only, 1.8V+2.5V or 1.8V+3.3V
  - Space-saving 8x8mm QFN56 (0.5mm pitch)

**Ordering Information**

| | | | |
|---|---|---|---|
| ZL30264LDG1 | ext. EEPROM | 6 Outputs | Trays |
| ZL30264LDF1 | ext. EEPROM | 6 Outputs | Tape and Reel |
| ZL30265LDG1 | int. EEPROM | 6 Outputs | Trays |
| ZL30265LDF1 | int. EEPROM | 6 Outputs | Tape and Reel |
| ZL30266LDG1 | ext. EEPROM | 10 Outputs | Trays |
| ZL30266LDF1 | ext. EEPROM | 10 Outputs | Tape and Reel |
| ZL30267LDG1 | int. EEPROM | 10 Outputs | Trays |
| ZL30267LDF1 | int. EEPROM | 10 Outputs | Tape and Reel |

Matte Tin
Package size: 8 x 8 mm, 56 Pin QFN
**-40°C to +85°C**

## Applications

- Frequency conversion and frequency synthesis in a wide variety of equipment types

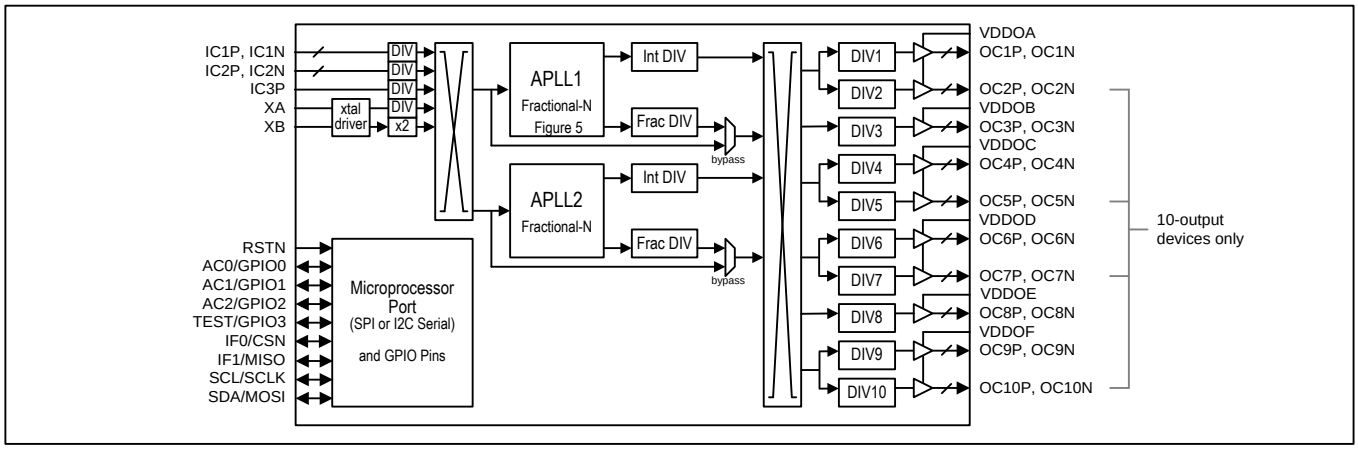



**Figure 1 - Functional Block Diagram**

## Table of Contents

## List of Figures



## List of Tables

# 1. Application Example

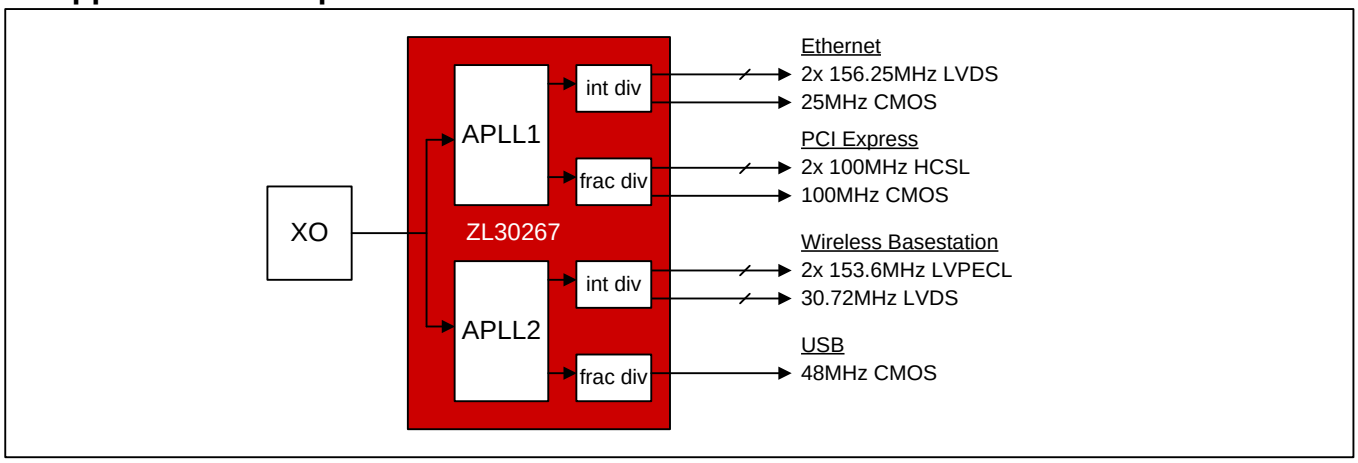



**Figure 2 - Application Example: ZL30267 Ethernet, PCIe, Wireless and USB Clocks Simultaneously**

# 2. Detailed Features

## 2.1 Input Clock Features

- Four input clocks: one crystal/CMOS, two differential/CMOS, one single-ended/CMOS
- Input clocks can be any frequency from 9.72MHz to 1250MHz (differential) or 300MHz (single-ended)
- Supported telecom frequencies include PDH, SDH, Synchronous Ethernet, OTN, wireless
- Activity monitor and glitchless input switching

## 2.2 APLL Features

- Two APLLs with very high-resolution fractional (i.e. non-integer) frequency multiplication
- Any-to-any frequency conversion with 0ppm error
- Two output dividers per APLL: one integer divider (4 to 15 plus half divides 4.5 to 7.5) and one fractional divider to make a total of four output frequency families
- Easy-to-configure, completely encapsulated design requires no external VCXO or loop filter
- Bypass mode supports system testing

## 2.3 Output Clock Features

- Six (ZL30264 or ZL30265) or ten (ZL30266 or ZL30267) low-jitter output clocks
- Each output can be one differential output or two CMOS outputs
- Output clocks can be any frequency from 1Hz to 1045MHz (250MHz max for HCSL, CMOS and HSTL)
- Output jitter from integer multiply and integer dividers as low as 0.17ps RMS (12kHz to 20MHz)
- Output jitter from fractional dividers is typically <1ps RMS, many frequencies <0.5ps RMS (12kHz to 20MHz)
- In CMOS mode, the OCxP and OCxN pins can be different divisors (Example 1: OC3P 125MHz, OC3N 25MHz; Example 2: OC3P 25MHz, OC3N 1Hz/1PPS)
- Outputs directly interface (DC coupled) with LVDS, LVPECL, HSTL, HCSL and CMOS components
- Supported telecom frequencies include PDH, SDH, Synchronous Ethernet, OTN
- Can produce clock frequencies for microprocessors, ASICs, FPGAs and other components
- Can produce PCIe-compliant clocks (PCIe 1, 2, 3 and 4)
- Sophisticated output-to-output phase alignment
- Per-output phase adjustment
- Per-output enable/disable
- Per-output glitchless start/stop (stop high or low)

## 2.4 General Features

- SPI or $I^2C$ serial microprocessor interface
- Automatic self-configuration at power-up; pin control to specify one of 8 stored configurations
  ZL30264 and ZL30266: preset configurations in ROM or user configurations in external EEPROM

ZL30265 and ZL30267: user configurations in internal EEPROM

- Numerically controlled oscillator (NCO) mode allows system software to steer DPLL frequency with resolution better than 0.01ppb (1ppt can be achieved with fractional output divider value >14.56)
- Spread-spectrum modulation mode (meets PCI Express requirements)
- Zero-delay buffer configuration using an external feedback path
- Four general-purpose I/O pins each with many possible status and control options
- Reference can be fundamental-mode crystal, low-cost XO or clock signal from elsewhere in the system

## 2.5 Evaluation Software

- Simple, intuitive Windows-based graphical user interface
- Supports all device features and register fields
- Makes lab evaluation of the ZL30264/5/6/7 quick and easy
- Generates configuration scripts to be stored in external (ZL30264/6) or internal (ZL30265/7) EEPROM
- Generates full or partial configuration scripts to be run on a system processor
- Works with or without an evaluation board

## 3. Pin Diagram

The device is packaged in a 8x8mm 56-pin QFN.

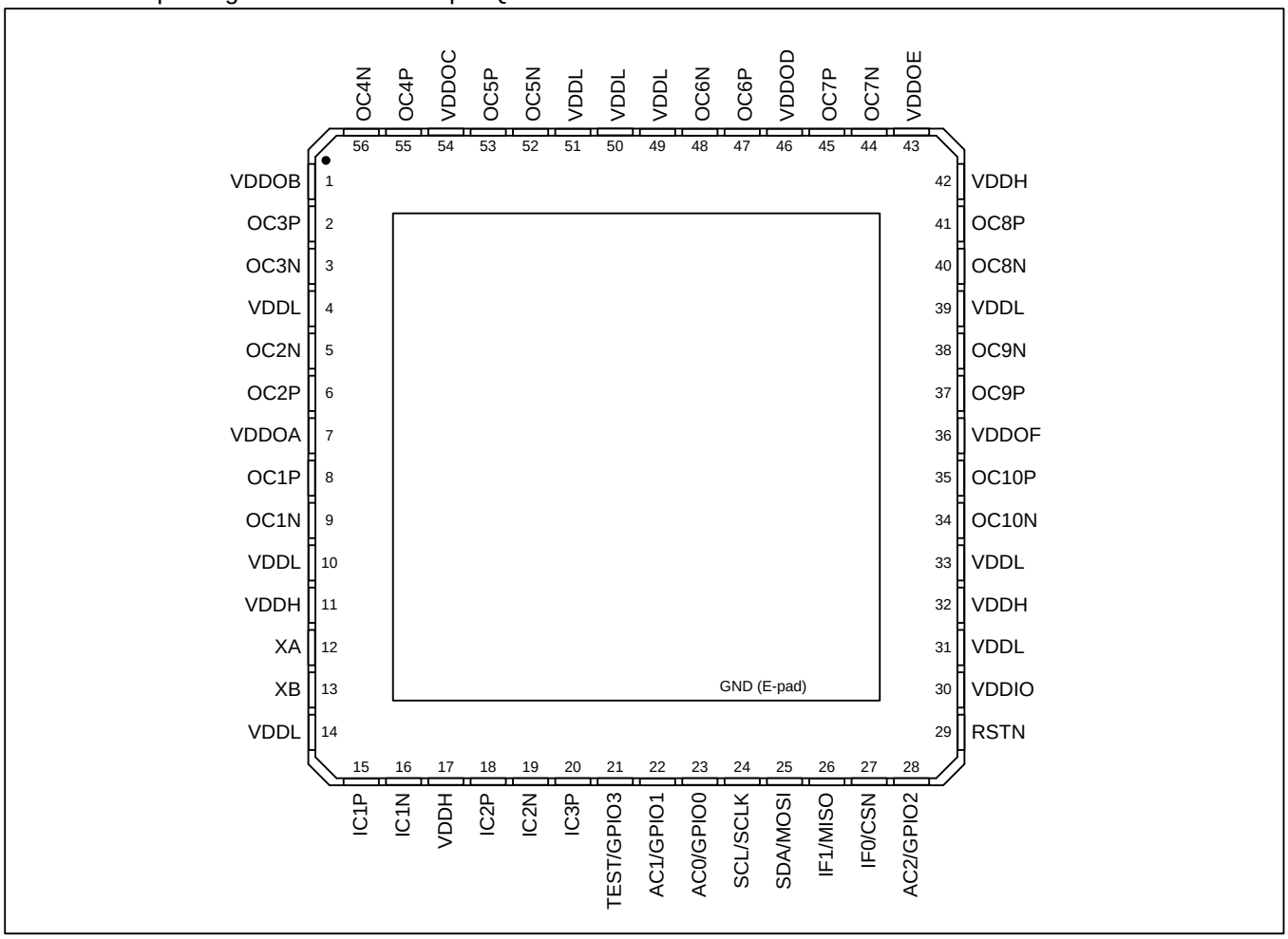



**Figure 3 - Pin Diagram**

## 4. Pin Descriptions

All device inputs and outputs are LVCMOS unless described otherwise. The Type column uses the following symbols: I – input, O – output, A – analog, P – power supply pin. All GPIO and SPI/I²C interface pins have Schmitt-trigger inputs and have output drivers that can be disabled (high impedance).

**Table 1 - Pin Descriptions**

| Pin # | Name | Type | Description |
|---|---|---|---|
| 15, 16<br>18, 19<br>20 | IC1P, IC1N<br>IC2P, IC2N<br>IC3P | I<br>I<br>I | **Input Clock Pins**<br>Differential or Single-ended signal format. Programmable frequency.<br><br>*Differential:* See Table 9 for electrical specifications, and see Figure 15 for recommended external circuitry for interfacing these differential inputs to LVDS, LVPECL, CML or HSCL output pins on neighboring devices.<br><br>*Single-ended:* For input signal amplitude >2.5V, connect the signal directly to ICxP pin. For input signal amplitude ≤2.5V, AC-coupling the signal to ICxP is recommended. Connect the N pin to a capacitor (0.1µF or 0.01µF) to VSS. As shown in Figure 15, the ICxP and ICxN pins are internally biased to approximately 1.3V. Treat the ICxN pin as a sensitive node; minimize stubs; do not connect to anything else including other ICxN pins.<br><br>*Unused:* Set ICEN.ICxEN=0. The ICxP and ICxN pins can be left floating.<br><br>Note that the IC3N pin is not bonded out. A differential signal can be connected to IC3P by AC-coupling the POS trace to IC3P and terminating the signal on the driver side of the coupling cap. |
| 12<br>13 | XA<br>XB | A / I | **Crystal or Input Clock Pins**<br>*Crystal:* MCR2.XAB=01. An on-chip crystal driver circuit is designed to work with an external crystal connected to the XA and XB pins. See section 5.3.2 for crystal characteristics and recommended external components.<br><br>*Input Clock:* MCR2.XAB=10. An external local oscillator or clock signal can be connected to the XA pin. The XB pin must be left unconnected. The signal on XA can be as large as 3.3V even when VDDH is only 2.5V. |
| 8, 9<br>6, 5<br>2, 3<br>55, 56<br>53, 52<br>47, 48<br>45, 44<br>41, 40<br>37, 38<br>35, 34 | OC1P, OC1N<br>OC2P, OC2N<br>OC3P, OC3N<br>OC4P, OC4N<br>OC5P, OC5N<br>OC6P, OC6N<br>OC7P, OC7N<br>OC8P, OC8N<br>OC9P, OC9N<br>OC10P, OC10N | O | **Output Clock Pins**<br>LVDS, programmable differential (which includes LVPECL), HCSL, HSTL or 1 or 2 CMOS. Programmable frequency. Programmable VCM and VOD in programmable differential mode. Programmable drive strength in CMOS and HSTL modes. See Figure 17 for example external interface circuitry.<br>See Table 10, Table 11 and Table 12 for electrical specifications for LVDS, LVPECL and HCSL, respectively.<br>See Table 13 for electrical specifications for interfacing to CMOS and HSTL inputs on neighboring devices.<br><br>Outputs OC2, OC5, OC7 and OC10 are not present on 6-output products. |
| 29 | RSTN | I | **Reset (Active Low)**<br>When this global asynchronous reset is pulled low, all internal circuitry is reset to default values. The device is held in reset as long as RSTN is low. Minimum low time is 1µs. See section 5.9. |
| 23<br>22<br>28 | AC0/GPIO0<br>AC1/GPIO1<br>AC2/GPIO2 | I/O | **Auto-Configure [2:0] / General Purpose I/O 0, 1 and 2**<br><br>*Auto Configure:* On the rising edge of RSTN these pins behave as AC[2:0] and specify one of the configurations stored in ROM or EEPROM. See section 5.2. |

| Pin # | Name | Type | Description |
|---|---|---|---|
| | | | *General-Purpose I/O:* After reset these pins are GPIO0, GPIO1 and GPIO2. GPIOCR1 and GPIOCR2.GPIO2C configure these pins. Their states are indicated in GPIOSR which has both real-time and latched status bits.<br><br>Note that when the power supply arrangement for the device has VDDL=1.8V, during the interval between VDDH ramping and VDDL ramping these pins can briefly behave as an output driving high. |
| 21 | TEST/GPIO3 | I/O | **Factory Test / General Purpose I/O 3**<br><br>*Factory Test:* On the rising edge of RSTN the pin behaves as TEST. Factory test mode is enabled when TEST is high. Typically TEST should be low on the rising edge of RSTN, but see section 5.2 for some options where TEST can be high on the rising edge of RSTN.<br><br>*General-Purpose I/O:* After reset this pin is GPIO3. GPIOCR2.GPIO3C configures the pin. It state is indicated in GPIOSR which has both real-time and latched status bits.<br><br>Note that when the power supply arrangement for the device has VDDL=1.8V, during the interval between VDDH ramping and VDDL ramping this pin can briefly behave as an output driving high. |
| 27 | IF0/CSN | I/O | **Interface Mode 0 / SPI Chip Select (Active Low)**<br><br>*Interface Mode:* On the rising edge of RSTN the pin behaves as IF0 and, together with IF1, specifies the interface mode for the device. See section 5.2.<br><br>*SPI Chip Select:* After reset this pin is CSN. When the device is configured as a SPI slave, an external SPI master must assert (low) CSN to access device registers. When the device is configured as a SPI master (ZL30264, ZL30266 only), the device asserts CSN to access an external SPI EEPROM during auto-configuration and then changes CSN to an input during normal operation. CSN should not be allowed to float. |
| 26 | IF1/MISO | I/O | **Interface Mode 1 / SPI Master-In-Slave-Out**<br><br>*Interface Mode:* On the rising edge of RSTN the pin behaves as IF1 and, together with IF0, specifies the interface mode for the device. See section 5.2.<br><br>*SPI MISO:* After reset this pin is MISO. When the device is configured as a SPI slave, the device outputs data to an external SPI master on MISO during SPI read transactions. When the device is configured as a SPI master (ZL30264, ZL30266 only), the device receives data on MISO from an external SPI EEPROM during auto-configuration. |
| 24 | SCL/SCLK | I/O | **$I^2C$ Clock / SPI Clock**<br><br>*$I^2C$ Clock:* When the device is configured as an $I^2C$ slave, an external $I^2C$ master must provide the $I^2C$ clock signal on the SCL pin. In $I^2C$ mode this pin should be externally pulled high by a 1kΩ to 5kΩ resistor.<br><br>*SPI Clock:* When the device is configured as a SPI slave, an external SPI master must provide the SPI clock signal on SCLK. When the device is configured as a SPI master (ZL30264, ZL30266 only), the device drives SCLK as an output to clock accesses to an external SPI EEPROM during auto-configuration. |
| 25 | SDA/MOSI | I/O | **$I^2C$ Data / SPI Master-Out-Slave-In**<br><br>*$I^2C$ Data:* When the device is configured as an $I^2C$ slave, SDA is the |

| Pin # | Name | Type | Description |
|---|---|---|---|
| | | | bidirectional data line between the device and an external I²C master. In I²C mode this pin should be externally pulled high by a 1kΩ to 5kΩ resistor.<br><br>*SPI MOSI:* When the device is configured as a SPI slave, an external SPI master sends commands, addresses and data to the device on MOSI. When the device is configured as a SPI master (ZL30264, ZL30266 only), the device sends commands, addresses and data on MOSI to an external SPI EEPROM during auto-configuration. |
| 11,17, 32,42 | VDDH | P | **Higher Core Power Supply.** 2.5V or 3.3V ±5%. When VDDH=3.3V the device has additional internal power supply regulators enabled. |
| 4,10, 14,31, 33,39, 49,50, 51 | VDDL | P | **Lower Core Power Supply.** 1.8V ±5% or same voltage as VDDH. |
| 30 | VDDIO | P | **Digital Power Supply for Non-Clock I/O Pins.** 1.8V to VDDH. |
| 7 | VDDOA | P | **Power Supply for OC1P/N and OC2P/N.** 1.5V to VDDH. |
| 1 | VDDOB | P | **Power Supply for OC3P/N.** 1.5V to VDDH. |
| 54 | VDDOC | P | **Power Supply for OC4P/N and OC5P/N.** 1.5V to VDDH. |
| 46 | VDDOD | P | **Power Supply for OC6P/N and OC7P/N.** 1.5V to VDDH. |
| 43 | VDDOE | P | **Power Supply for OC8P/N.** 1.5V to VDDH. |
| 36 | VDDOF | P | **Power Supply for OC9P/N and OC10P/N.** 1.5V to VDDH. |
| E-pad | VSS | P | **Ground.** 0 Volts. |

**Important Note**: The voltages on VDDL, VDDIO, and all VDDOx pins must not exceed VDDH. Not complying with this requirement may damage the device.

# 5. Functional Description

## 5.1 Device Identification

The 12-bit read-only ID field and the 4-bit revision field are found in the ID1 and ID2 registers. Contact the factory to interpret the revision value and determine the latest revision.

## 5.2 Pin-Controlled Automatic Configuration at Reset

The device configuration is determined at reset (i.e. on the rising edge of RSTN) by the signal levels on these device pins: TEST/GPIO3, AC2/GPIO2, AC1/GPIO1, AC0/GPIO0, IF1/MISO and IF0/CSN. For these pins, the first name (TEST, AC2, AC1, AC0, IF1, IF0) indicates their function when they are sampled by the rising edge of the RSTN pin. The second name refers to their function after reset. The values of these pins are latched into the CFGSR register when RSTN goes high. To ensure the device properly samples the reset values of these pins, the following guidelines should be followed:

1. Any pullup or pulldown resistors used to set the value of these pins at reset should be 1kΩ.
2. RSTN must be asserted at least as long as specified in section 5.9.

The hardware configuration pins are grouped into three sets:

1. TEST - Manufacturing test mode
2. IF[1:0] – Microprocessor interface mode and I²C address
3. AC[2:0] – Auto-config configuration number (0 to 7)

The TEST pin selects manufacturing test modes when TEST=1 (the AC[2:0] pins specify the test mode). For ZL30265 and ZL30267 (devices with internal EEPROM), TEST=1, AC[2:0]=000, IF[1:0]=11 configures the part so that production SPI EEPROM programmers can program the internal EEPROM (see section 5.11.2). TEST=1 and

AC[2:0]=011 causes the part to start normally except it does not auto-configure from EEPROM or ROM. For more information about auto-configuration from EEPROM or ROM see section 5.11.

For all of these pins Microsemi recommends that board designs include component sites for both pullup and pulldown resistors (only one or the other populated per pin).

### 5.2.1 ZL30264 and ZL30266—Internal ROM, External or No EEPROM

For these part numbers the IF[1:0] pins specify the processor interface mode, the I²C slave address and whether the device should auto-configure from internal ROM or external EEPROM. The AC[2:0] pins specify which device configuration in the ROM or EEPROM to execute after reset. Descriptions of the standard-product ROM configurations are available from Microsemi.

| IF1 | IF0 | Processor Interface | Configuration Memory to Use |
|---|---|---|---|
| 0 | 0 | I²C, slave address 11101 00 | Internal ROM |
| 0 | 1 | I²C, slave address 11101 01 | Internal ROM |
| 1 | 0 | SPI Slave | Internal ROM |
| 1 | 1 | SPI Master during auto-configuration then SPI Slave | External SPI EEPROM |

To configure the device as specified in the first three rows above but *without* auto-configuring from internal ROM, wire devices pins as follows: TEST=1 and AC[2:0]=011, as described in section 5.2.

| AC2 | AC1 | AC0 | Auto Configuration |
|---|---|---|---|
| 0 | 0 | 0 | Configuration 0 |
| 0 | 0 | 1 | Configuration 1 |
| 0 | 1 | 0 | Configuration 2 |
| 0 | 1 | 1 | Configuration 3 |
| 1 | 0 | 0 | Configuration 4 |
| 1 | 0 | 1 | Configuration 5 |
| 1 | 1 | 0 | Configuration 6 |
| 1 | 1 | 1 | Configuration 7 |

Notes about the device auto-configuring from external EEPROM:

1. The device's CSN pin should have a pull-up resistor to VDD to ensure its processor interface is inactive after auto-configuration is complete. The SCLK, MISO and MOSI pins should also have pull-up resistors to VDD to keep them from floating.
2. If a processor or similar device will access device registers after the device has auto-configured from external EEPROM, the SPI SCLK, MOSI and MISO wires can be connected directly to the processor, the device and the external EEPROM. The processor and device CSN pins can be wired together also. The EEPROM CSN signal must be controlled by the device's CSN pin during device auto-configuration and then held inactive when the processor accesses device registers.
3. The bits of the I²C address are as shown above by default but can be changed in the I2CA register.

### 5.2.2 ZL30265 and ZL30267—Internal EEPROM

For these part numbers the IF[1:0] pins specify the processor interface mode and the I²C slave address. The AC[2:0] pins specify which device configuration in the EEPROM to execute after reset.

| IF1 | IF0 | Processor Interface |
|---|---|---|
| 0 | 0 | I²C, slave address 11101 00 |
| 0 | 1 | I²C, slave address 11101 01 |
| 1 | 0 | I²C, slave address 11101 10 |
| 1 | 1 | SPI Slave |

| AC2 | AC1 | AC0 | Auto Configuration |
|---|---|---|---|
| 0 | 0 | 0 | Configuration 0 |
| 0 | 0 | 1 | Configuration 1 |

| AC2 | AC1 | AC0 | Auto Configuration |
|---|---|---|---|
| 0 | 1 | 0 | Configuration 2 |
| 0 | 1 | 1 | Configuration 3 |
| 1 | 0 | 0 | Configuration 4 |
| 1 | 0 | 1 | Configuration 5 |
| 1 | 1 | 0 | Configuration 6 |
| 1 | 1 | 1 | Configuration 7 |

Note: the bits of the I²C address are as shown above by default but can be changed in the I2CA register. A device's I²C slave address can be set to any value during auto-configuration at power-up by writing the I2CA register as part of the configuration script.

## 5.3 Local Oscillator or Crystal

Section 5.3.1 describes how to connect an external oscillator and the required characteristics of the oscillator. Section 5.3.2 describes how to connect an external crystal to the on-chip crystal driver circuit and the required characteristics of the crystal. The device does not require an external oscillator or crystal for operation.

### 5.3.1 External Oscillator

A signal from an external oscillator can be connected to the XA pin (XB must be left unconnected).

Table 8 specifies the range of possible frequencies for the XA input. To minimize jitter, the signal must be properly terminated and must have very short trace length. A poorly terminated single-ended signal can greatly increase output jitter, and long single-ended trace lengths are more susceptible to noise. When MCR2.XAB=10, XA is enabled as a single-ended input.

While the stability of the external oscillator can be important, its absolute frequency accuracy is less important because any known frequency inaccuracy of the oscillator can be compensated by adjusting the APLL's fractional feedback divider value (AFBDIV) by ppb or ppm.

The jitter on output clock signals depends on the phase noise and frequency of the external oscillator. For the device to operate with the lowest possible output jitter, the external oscillator should have the following characteristics:

- Phase Jitter: less than 0.1ps RMS over the 12kHz to 5MHz integration band
- Frequency: The higher the better, all else being equal

### 5.3.2 External Crystal and On-Chip Driver Circuit

The on-chip crystal driver circuit is designed to work with a fundamental mode, AT-cut crystal resonator. See Table 2 for recommended crystal specifications. To enable the crystal driver, set MCR2.XAB=01.

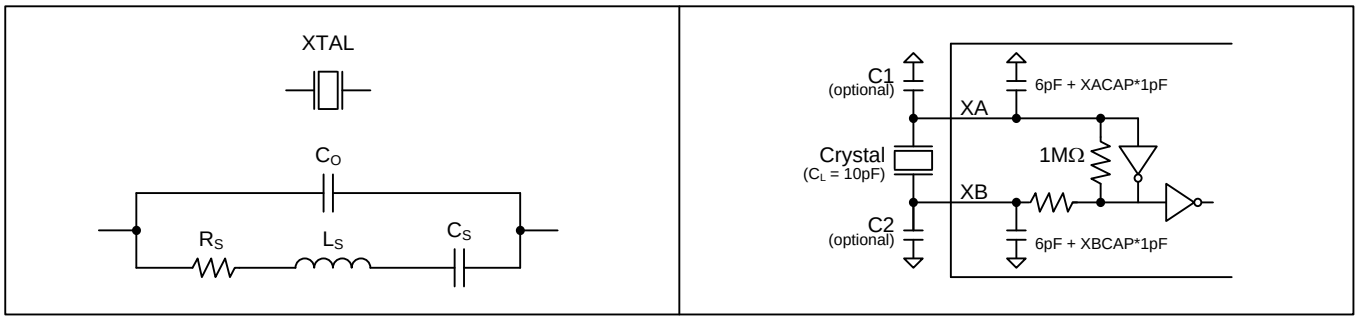



**Figure 4 - Crystal Equivalent Circuit / Recommended Crystal Circuit**

See Figure 4 for the crystal equivalent circuit and the recommended external component connections. The driver circuit design includes configurable internal load capacitors. For a 10pF crystal the total capacitance on each of XA and XB should be 2 x 10pF = 20pF. To achieve these loads without external capacitors, register field XACR3.XACAP should be set to 20pF minus actual XA external board trace capacitance minus XA's minimum

internal capacitance of 6pF. For example, if external trace capacitance is 2pF then XACAP should be set to 20pF – 2pF – 6pF = 12pF. Register field XACR3.XBCAP should be set in a similar manner for XB load capacitance. Crystals with nominal load capacitance other than 10pF usually can be supported with only internal load capacitance. If the XACAP and XBCAP fields do not have sufficient range for the application, capacitance can be increased by using external caps C1 and C2.

Users should also note that on-chip capacitors are not nearly as accurate as discrete capacitors (which can have 1% accuracy). If tight frequency accuracy is required for the crystal driver circuit then set XACAP and XBCAP both to 0 and choose appropriate C1 and C2 capacitors with 1% tolerance.

The crystal, traces, and two external capacitors sites (if included) should be placed on the board as close as possible to the XA and XB pins to reduce crosstalk of active signals into the oscillator. Also no active signals should be routed under the crystal circuitry.

Note: Crystals have temperature sensitivies that can cause frequency changes in response to ambient temperature changes. In applications where significant temperature changes are expected near the crystal, it is recommended that the crystal be covered with a thermal cap, or an external XO or TCXO should be used instead.

**Table 2 - Crystal Selection Parameters**

| Parameter | | Symbol | Min. | Typ. | Max. | Units |
|---|---|---|---|---|---|---|
| Crystal Oscillation Frequency[1] | | $f_{OSC}$ | 25 | | 60 | MHz |
| Shunt Capacitance | | $C_O$ | | 2 | 5 | pF |
| Load Capacitance[3] | | $C_L$ | 8 | 10 | 16 | pF |
| Equivalent Series Resistance (ESR)[2] | $f_{OSC}$ < 40MHz | $R_S$ | | | 60 | Ω |
| | $f_{OSC}$ > 40MHz | $R_S$ | | | 50 | Ω |
| Maximum Crystal Drive Level | | | 100 | 100, 200, 300 | | µW |

Note 1: Higher frequencies give lower output jitter, all else being equal.

Note 2: These ESR limits are chosen to constrain crystal drive level to less than 100µW. If the crystal can tolerate a drive level greater than 100µW then proportionally higher ESR is acceptable.

Note 3: For crystals with 100µW max drive level: (a) $f_{OSC}$>55MHz and $C_L$≥12pF is not supported, and (b) $f_{OSC}$>45MHz and $C_L$≥16pF is not supported. Crystals with max drive level of 200µW or higher do not have these limitations.

| Parameter | Symbol | Min. | Typ. | Max. | Units |
|---|---|---|---|---|---|
| Crystal Frequency Stability vs. Power Supply | $f_{FVD}$ | | 0.2 | 0.5 | ppm per 10% Δ in VDD |

Any known frequency inaccuracy of the crystal can be compensated in the APLL by adjusting the APLL's fractional feedback divider value (AFBDIV) by ppb or ppm to compensate for crystal frequency error.

### 5.3.3 Clock Doublers

Figure 1 shows an optional clock doubler (“x2” block) following the crystal driver block. This XA doubler, which is enabled by setting MCR2.DBL=1, can be used to double the frequency of the internal crystal driver circuit or a 20MHz to 78.125MHz clock signal on the XA pin. For input clock frequencies from 25MHz to 78.125MHz the duty cycle of the signal can be anywhere in the 40% to 60% range. For input clock frequencies from 20MHz to 25MHz the duty cycle must be in the 45% to 55% range. Figure 1 also shows an optional doubler at the input of each APLL. This APLL input doubler, which is enabled by setting ACR1.INDBL=1 for APLL1 or A2CR1.INDBL=1 for APLL2, can be used to double the frequency of any of the inputs. The following table shows scenarios when the clock doubler can be used.

| Scenario | With Crystal | With XO or Clock Signal |
|---|---|---|
| APLL, Integer Multiply | Yes[1] | Maybe[1] |
| APLL, Fractional Multiply | Yes | Yes |
| NCO | Yes | Yes |
| Spread-Spectrum | Yes | Yes |
| APLL bypass path | No[2] | No[2] |

Note 1: For APLL integer multiplication, use of the doubler is application-dependent. On the positive side, use of the doubler reduces random jitter. On the negative side, the doubler causes a spur at the XA frequency (but this spur may be outside the band of interest for the application).

Note 2: The signal generated by the doubler has a very narrow and variable pulse width and therefore it is not recommended to connect the doubler signal directly to the OCx outputs using an APLL bypass path. The doubler signal is fine as an input to the APLL, which filters the duty cycle distortion and produces a 50% duty cycle output.

Note 3: Using both doublers in series to double the XA-doubled signal is not supported.

### 5.3.4 Ring Oscillator (for Auto-Configuration)

After reset the internal auto-configuration boot controller is clocked by an internal ring oscillator. After auto-configuration is complete (GLOBISR.BCDONE=1) the ring oscillator can be disabled by setting MCR1.ROSCD=1. The device's processor interface is asynchronous and does not require the ring oscillator.

## 5.4 Input Signal Format Configuration

Input clocks IC1, IC2 and IC3 are enabled by setting the enable bits in the ICEN register. The power consumed by a differential receiver is shown in Table 6. The electrical specifications for these inputs are listed in Table 9. Each input clock can be configured to accept nearly any differential signal format by using the proper set of external components (see Figure 15). To configure these differential inputs to accept single-ended CMOS signals, connect the single-ended signal to the ICxP pin, and connect the ICxN pin to a capacitor (0.1μF or 0.01μF) to VSS. Each ICxP and ICxN pin is internally biased to approximately 1.3V. If an input is not used, both ICxP and ICxN pins can be left floating. Note that the IC3N pin is not present. A differential signal can be connected to IC3P by AC-coupling the POS trace to IC3P and terminating the signal on the driver side of the coupling cap.

## 5.5 APLL Configuration

ZL30264 through ZL30267 have two independent APLLs. Each APLL has all of the functionality described in the subsections below. For brevity, only APLL1-specific registers are mentioned. APLL2 has equivalent registers.

### 5.5.1 APLL Input Frequency

The frequencies of all enabled input clocks (ICx and XA) associated with an APLL must divide to a common APLL phase-frequency detector (PFD) frequency from 9.72MHz to 156.25MHz. In this mode the input high-speed dividers (ICxCR1.HSDIV) can be used to divide the ICx frequencies by 1, 2, 4 or 8 and the XA divider (MCR2. XODIV2) can be used to divide the XA frequency by 1 or 2. The polarity of an ICx input signal can be inverted by setting ICxCR1.POL.

### 5.5.2 APLL Input Monitors

Each of the APLL's four inputs—IC1, IC2, IC3 and XA—have a simple activity monitor. If the monitor counts approximately four (eight if the input clock is doubled) APLL feedback clock cycles without seeing an input clock edge, the input is declared invalid and the corresponding ICxSR.ICV bit or XASR.ICV bit is set to 0. The input clock is declared valid, and the corresponding ICxSR.ICV bit or XASR.ICV bit is set to 1, when the input clock has no missing edges in an interval specified by the corresponding ICxCR1.VALTIME or XACR1.VALTIME field. The XASR and ICxSR registers provide real-time and latched status bits indicating the state of each input.

The feedback clock to use for each input monitor is specified by the MCR2.XAMCK and MCR2.ICxMCK bits.

### 5.5.3 APLL Input Selection

The APLL can lock to any of inputs IC1 through IC3, a clock signal on XA (optionally clock-doubled), or the crystal driver circuit (optionally clock-doubled) when a crystal is connected to XA and XB.

The input to the APLL can be controlled by a register field, a GPIO pin, or a simple input activity monitor. When ACR3.EXTSW=0 and ACR3.INMON=0, the ACR3.APLLMUX register field controls the APLL input mux.

When ACR3.EXTSW=1, a GPIO pin controls the APLL input mux. When the GPIO pin is low, the mux selects the input specified by ACR3.APLLMUX. When the GPIO pin is high, the mux selects the input specified by ACR3.ALTMUX. ACR1.EXTSS specifies which GPIO pin controls this behavior.

When ACR3.EXTSW=0 and ACR3.INMON=1, an input monitor (see section 5.5.2) controls the APLL input mux. When the monitor for the input specified by ACR3.APLLMUX says that input is valid (ICxSR.ICV=1 or

XASR.ICV=1), the mux selects the input specified by ACR3.APLLMUX otherwise the mux selects the input specified by ACR3.ALTMUX.

The APLL1SR.SELREF and APLL2SR.SELREF real-time status fields indicate the APLL's selected reference.

### 5.5.4 APLL Output Frequency

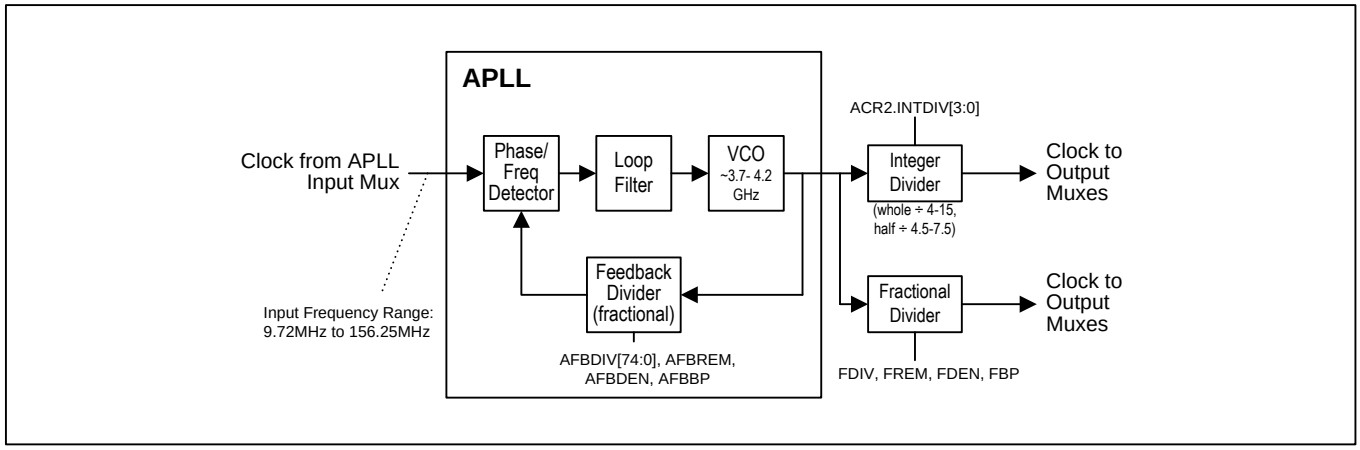



**Figure 5 - APLL Block Diagram**

The APLL is enabled when PLLEN.APLLEN=1. The APLL has a fractional-N architecture and therefore can produce output frequencies that are either integer or non-integer multiples of the input clock frequency. Figure 5 shows a block diagram of the APLL, which is built around an ultra-low-jitter multi-GHz VCO. Register fields AFBDIV, AFBREM, AFBDEN and AFBBP configure the frequency multiplication ratio of the APLL. The ACR2.INTDIV field specifies how the VCO frequency is divided down by the APLL's integer divider (which can also do some half divides). Dividing by 6 is the typical setting to produce 622.08MHz for SDH/SONET or 625MHz for Ethernet applications. The configuration registers for the APLL's fractional divider are described in section 5.5.5.

Internally, the exact APLL feedback divider value is expressed in the form AFBDIV + AFBREM / AFBDEN * $2^{-(33-AFBBP)}$. This feedback divider value must be chosen such that APLL_input_frequency * feedback_divider_value is in the operating range of the VCO (as specified in Table 14). The AFBDIV term is a fixed-point number with 9 integer bits and a configurable number of fractional bits (up to 33, as specified by AFBBP). Typically AFBBP is set to 9 to specify that AFBDIV has 33 – 9 = 24 fractional bits. Using more than 24 fractional bits does not yield a detectable benefit. Using less than 12 fractional bits is not recommended.

The following equations show how to calculate the feedback divider values for the situation where the APLL should multiply the APLL input frequency by integer M and also fractionally scale by the ratio of integers N / D. In other words, VCO_frequency = input_frequency * M * N / D. An example of this is multiplying 77.76MHz by M=48 and scaling by N / D = 255 / 237 for forward error correction applications.

$$afbdiv = trunc(M * N / D * 2^{24}) \quad (1)$$

$$lsb\_fraction = M * N / D * 2^{24} - afbdiv \quad (2)$$

$$AFBDEN = D \quad (3)$$

$$AFBREM = round(lsb\_fraction * AFBDEN) \quad (4)$$

$$AFBBP = 33 - 24 = 9 \quad (5)$$

$$AFBDIV[41:0] = afbdiv * 2^{AFBBP} \quad (6)$$

The trunc() function returns only the integer portion of the number. The round() function rounds the number to the nearest integer. In Equation (1), the temporary variable 'afbdiv' is set to the full-precision feedback divider value, M * N / D, truncated after the 24th fractional bit. In Equation (2) the temporary variable 'lsb_fraction' is the fraction that was truncated in Equation (1) and therefore is not represented in the afbdiv value. In Equation (3), AFBDEN is set to the denominator of the original M * N / D ratio. In Equation (4), AFBREM is calculated as the integer numerator of a fraction (with denominator AFBDEN) that equals the 'lsb_fraction' temporary variable. In Equation (5) AFBBP is set to 33 – 24 = 9 to correspond with AFBDIV having 24 fractional bits. Finally, in equation (6) the afbdiv bits are shifted into the proper position for the AFBDIV registers.

When a fractional scaling scenario involves multiplying an integer M times multiple scaling ratios $N_1 / D_1$ through $N_n / D_n$, the equations above can still be used if the numerators are multiplied together to get $N = N_1 \times N_2 \times \ldots \times N_n$ and the denominators are multiplied together to get $D = D_1 \times D_2 \times \ldots \times D_n$.

The easiest way to calculate the exact values to write to the APLL registers is to use the ZL3026x evaluation software, available on the Microsemi website. This software can be used even when no evaluation board is attached to the computer.

Note: After the APLL's feedback divider settings are configured in register fields AFBDIV, AFBREM, AFBDEN and AFBBP, the APLL enable bit PLLEN.APLLEN should be changed from 0 to 1 to cause the APLL to reacquire lock with the new settings. The real-time lock/unlock status of the APLL is indicated by APLL1SR.ALK.

### 5.5.5 Fractional Output Divider

As shown in Figure 1 and Figure 5, each APLL has a fractional output divider. This allows each APLL to be the source of two unrelated frequency families, one from the integer divider, and one from the fractional divider.

Configuration of the fractional output divider is very similar to configuration of the APLL's feedback divider. The fractional divider is enabled by setting ACR1.ENFDIV. Internally, the exact divider value is expressed in the form FDIV + FREM / FDEN * $2^{-(36-FBP)}$. The input clock to the fractional divider is APLL VCO frequency divided by 2 ($f_{VCO}/2$). The FDIV term is a fixed-point number with 4 integer bits and a configurable number of fractional bits (up to 36, as specified by FBP). Typically FBP is set to 12 to specify that FDIV has 36 – 12 = 24 fractional bits.

The output clock from the fractional divider has good phase noise on rising edges but worse phase noise on falling edges and can have non-50% duty cycle. Applications that only use clock rising edges can use the fractional divider's output clock directly. For applications that care about 50% duty cycle and/or the phase noise of both rising edges and falling edges, the fractional divider should be followed by an even medium-speed divider value (2, 4, 6, 8…). The low-speed divider can be used to further divide the output clock if needed.

The maximum output frequency for the fractional divider is $f_{VCO}/10$. This means the minimum fractional divider value is 5.0. Including the need for a divide-by-2 in the medium-speed divider, the maximum frequency for a 50% duty-cycle output clock signal is $f_{VCO}/20$. The minimum output frequency for the fractional divider is $f_{VCO}/32$ since the internal FDIV value has 4 integer bits. The combination FDIV=0, FREM=0, FDEN=1 specifies to divide by 16.0. The medium-speed and low-speed dividers can be configured to follow the fractional output divider to create output frequencies down to <1Hz.

The following equations show how to calculate the register values for the situation where the fractional divider should divide by the integer M and the ratio of integers N / D. In other words,

frac_div_output_freq = (VCO_freq / 2) / (M * N / D)

An example of this is starting with VCO_freq = 3750MHz (to get low-jitter Ethernet frequencies through the APLL's integer divider) and using the APLL's fractional divider to get 155.52MHz for SONET/SDH applications. In this example, M=12, N=15625, D=15552 are appropriate values to get 155.52MHz.

$$fdiv = trunc(M * N / D * 2^{24}) \quad (1)$$

$$lsb\_fraction = M * N / D * 2^{24} - fdiv \quad (2)$$

$$FDEN = D \quad (3)$$

$$FREM = round(lsb\_fraction * FDEN) \quad (4)$$

$$FBP = 36 - 24 = 12 \quad (5)$$

$$FDIV[39:0] = fdiv * 2^{FBP} \quad (6)$$

The trunc() function returns only the integer portion of the number. The round() function rounds the number to the nearest integer. In Equation (1), the temporary variable 'fdiv' is the full-precision feedback divider value, M * N / D, truncated after the 24th fractional bit. In Equation (2) the temporary variable 'lsb_fraction' is the fraction that was truncated in Equation (1) and therefore is not represented in the fdiv value. In Equation (3), FDEN is set to the denominator of the original M * N / D ratio. In Equation (4), FREM is calculated as the integer numerator of a

fraction (with denominator FDEN) that equals the 'lsb_fraction' temporary variable. In Equation (5) FBP is set to 36 – 24 =12 to correspond with FDIV having 24 fractional bits. Finally, in equation (6) the fdiv bits are shifted into the proper position for the FDIV registers.

When a fractional scaling scenario involves multiplying an integer M times multiple scaling ratios $N_1 / D_1$ through $N_n / D_n$, the equations above can still be used if the numerators are multiplied together to get $N = N_1 \times N_2 \times \ldots \times N_n$ and the denominators are multiplied together to get $D = D_1 \times D_2 \times \ldots \times D_n$.

The easiest way to calculate the exact values to write to the APLL's fractional output divider registers is to use the ZL3026x evaluation software, available on the Microsemi website. This software can be used even when no evaluation board is attached to the computer.

### 5.5.6 Numerically Controlled Oscillator (NCO) Mode

#### 5.5.6.1 Using the APLL's Feedback Divider

System software can steer output frequencies with high resolution by manipulating the APLL's AFBDIV value. The resolution can be better than 0.01ppb.

The nominal AFBDIV value, hereafter referred to as AFBDIV0, is the 0ppm nominal value for the desired device configuration.

(Fractional frequency offset (FFO) is defined as (actual_frequency – nominal_frequency) / nominal_frequency. FFO is a unitless number but is typically expressed in parts per billion (ppb), parts per million (ppm) or percent.)

To control the NCO, system software first reads the AFBDIV0 value from the device. Even though the AFBDIV register description describes AFBDIV as having 9 integer bits and 33 fractional bits, for the NCO calculations that follow, AFBDIV values should be treated as 42-bit unsigned integers.

To change the NCO frequency to a specific FFO (in ppm), system software calculates newAFBDIV (a 42-bit unsigned integer) as follows:

newAFBDIV = round(AFBDIV0 * (1 + FFO/1e6))

System software then writes the newAFBDIV value directly to the AFBDIV registers.

Note that any subsequent frequency changes are calculated using the same equation from the original AFBDIV0 value and are not a function of the previous newAFBDIV value. The value of newAFBDIV should be kept within ±1000ppm of AFBDIV0 and within ±500ppm of the previous newAFBDIV value to avoid causing the APLL to lose lock. If spread spectrum modulation is also in use, the total frequency change caused by spread spectrum modulation and NCO control should be kept within ±5000ppm of AFBDIV0 to avoid causing the APLL to lose lock.

During NCO operation using the APLL's feedback divider, AFBREM should be set to 0, AFBDEN should be set to 1 and AFBBP should be set to 0.

#### 5.5.6.2 Using the Fractional Output Divider

System software can steer output frequencies derived from the fractional output divider with high resolution by manipulating the divider's FDIV value. The resolution can be better than 0.01ppb. In the case of fractional output divider value >14.55192, the resolution is better than 1ppt.

The nominal FDIV value, hereafter referred to as FDIV0, is the 0ppm nominal value for the desired device configuration.

(Fractional frequency offset (FFO) is defined as (actual_frequency – nominal_frequency) / nominal_frequency. FFO is a unitless number but is typically expressed in parts per billion (ppb), parts per million (ppm) or percent.)

To control the NCO, system software first reads the FDIV0 value from the device. Even though the FDIV register description describes FDIV as having 4 integer bits and 36 fractional bits, for the NCO calculations that follow, FDIV values should be treated as 40-bit unsigned integers.

To change the NCO frequency to a specific FFO (in ppm), system software calculates newFDIV (a 40-bit unsigned integer) as follows:

newFDIV = round(FDIV0 / (1 + FFO/1e6))

System software then writes the newFDIV value directly to the FDIV registers.

Note that any subsequent frequency changes are calculated using the same equation from the original FDIV0 value and are not a function of the previous newFDIV value.

During NCO operation using the fractional output divider, FREM should be set to 0, FDEN should be set to 1 and FBP should be set to 0.

### 5.5.7 Frequency Increment and Decrement

When ACR1.USEFDIV=0 the APLL's feedback divider value can be incremented or decremented by values ranging from approximately 1ppb to 500ppm. The value AID[23:0] x $2^7$ is added to the APLL's feedback divider value each time the trigger specified by AIDCR.FISS changes state. AID[23:0] x $2^7$ is subtracted from the APLL's feedback divider value each time the trigger specified by AIDCR.FDSS changes state. The value to be written to AID[23:0] s as follows:

AID = round(AFBDIV0 * FFO/1e6 / $2^7$)

where FFO is the desired fractional frequency offset (FFO) per increment or decrement step in ppm

AFBDIV0 is the nominal AFBDIV value, obtained by reading AFBDIV when AFBDL.RDCUR=0

The current APLL feedback divider value (i.e. the value after increments and decrements) can be read from the AFBDIV registers when AFBDL.RDCUR=1. The original value of the AFBDIV registers before increments and decrements can be read from the AFBDIV registers when AFBDL.RDCUR=0. Incrementing and decrementing only occur when the APLL is locked (APLL1SR.ALK=1). If the APLL loses lock, when it locks again the APLL sets its feedback divider value back to the AFBDIV0 value. The system must be designed to ensure the current feedback divider value stays within ±1000ppm of the AFBDIV0 value to avoid causing the APLL to lose lock. The maximum increment/decrement is 500ppm. Frequency increment/decrement behavior is mutually exclusive with spread-spectrum modulation (see section 5.5.8) because both behaviors use the AID registers. Frequency increment/decrement is enabled when ACR1.ENFID=1. When ENFID is set to 0 the APLL feedback divider instantly changes to the AFBDIV0 value. Therefore, to avoid a frequency jump on output clocks, system software should increment or decrement back to the AFBDIV0 value before changing ENFID to 0.

### 5.5.8 Spread-Spectrum Modulation Mode

For EMI-sensitive applications such as PCI Express, the device can perform spread spectrum modulation (SSM). In SSM the frequency of the output clock is continually varied over a narrow frequency range to spread the energy of the signal and thereby reduce EMI. This mode is a special case of NCO mode.

Spread spectrum mode is enabled by the ASCR.ENSS bit or by a GPIO pin as specified by ASCR.SPRDSS.

#### 5.5.8.1 Using the APLL's Feedback Divider

When ACR1.USEFDIV=0 the device performs frequency modulation by modulating the APLL's feedback divider value starting from the nominal value specified in the AFBDIV registers.

For center-spread applications (ASCR.DWNEN=0), the frequency modulation is triangle-wave center-spread of up to ±0.5% deviation from the center frequency with modulation rate configurable from 25kHz to 55kHz. The spread deviation and modulation rate are controlled by specifying an increment/decrement step in the AID registers and the number of APLL input clock cycles to increment and decrement in the ASCNT registers. The values to be written to the device are calculated as follows:

$$ASCNT = round\left(\frac{F_{IN}}{4 * F_{MOD}}\right) - 2$$

$$AID = round\left(\frac{F_{VCO}}{F_{IN}} * \frac{S * 2^{32}}{ASCNT + 2}\right)$$

where $F_{VCO}$ is the APLL"s VCO frequency in Hz,

$F_{IN}$ is the input frequency at the APLL's PFD in Hz,

$F_{MOD}$ is the spread-spectrum modulation frequency in Hz,

and S is the peak-to-peak spread percentage expressed as a decimal (example ±0.5% → S=0.01)

For down-spread applications (ASCR.DWNEN=1), such as PCI Express Refclk, the frequency modulation is triangle-wave down-spread of up to -1% deviation from the nominal frequency with modulation rate configurable from 25kHz to 55kHz. The spread deviation and modulation rate are controlled by specifying an increment/decrement step in the AID registers and the number of APLL input clock cycles to increment and decrement in the ASCNT registers. The values to be written to the device are calculated as follows:

$$ASCNT = round\left(\frac{F_{IN}}{2 * F_{MOD}}\right) - 2$$

$$AID = round\left(\frac{F_{VCO}}{F_{IN}} * \frac{S * 2^{33}}{ASCNT + 2}\right)$$

All of the input parameters are the same as for center spread above. Note the small differences between these down-spread equations and the center-spread equations above. ASCNT here has 2 in the denominator while it has 4 for center-spread. AID has $2^{33}$ in the numerator for down-spread while it has $2^{32}$ for center-spread.

During spread-spectrum operation using the APLL's feedback divider, AFBREM should be set to 0, AFBDEN should be set to 1 and AFBBP should be set to 0. Spread-spectrum modulation only occurs when the APLL is locked (APLL1SR.ALK=1).

### 5.5.8.2 Using the Fractional Output Divider

When ACR1.USEFDIV=1 the device performs frequency modulation by modulating the fractional output divider value starting from the nominal value specified in the FDIV registers.

For center-spread applications (ASCR.DWNEN=0), the frequency modulation is triangle-wave center-spread up to ±5% deviation from the center frequency with modulation rate configurable from 10kHz to 100kHz. (Values outside of these ranges are often achievable as well.) The spread deviation and modulation rate are controlled by specifying an increment/decrement step in the AID registers and the number of APLL input clock cycles to increment and decrement in the ASCNT registers. The ASCR.CNTEN register field must also be set properly. The values to be written to the device are calculated as follows:

$$CNTEN = \begin{array}{l} 0 \text{ for } F_{IN} < 50MHz \\ 1 \text{ for } 50MHz \le F_{IN} < 100MHz \\ 2 \text{ for } F_{IN} \ge 100MHz \end{array}$$

$$ASCNT = round\left(\frac{F_{IN}}{4 * F_{MOD} * 2^{CNTEN}}\right) - 2$$

$$AID = round\left(\frac{F_{VCO}}{F_{OUT}} * \frac{S * 2^{34}}{(ASCNT + 2) * (1 - 0.5S)}\right)$$

where $F_{VCO}$ is the APLL"s VCO frequency in Hz,

$F_{IN}$ is the input frequency at the APLL's PFD in Hz,

$F_{OUT}$ is the fractional output divider block's output frequency in Hz,

$F_{MOD}$ is the spread-spectrum modulation frequency in Hz,

and S is the peak-to-peak spread percentage expressed as a decimal (example ±0.5% → S=0.01)

For down-spread applications (ASCR.DWNEN=1), such as PCI Express Refclk, the frequency modulation is triangle-wave down-spread of up to -10% deviation from the nominal frequency with modulation rate configurable from 10kHz to 100kHz. (Values outside of these ranges are often achievable as well.) The spread deviation and modulation rate are controlled by specifying an increment/decrement step in the AID registers and the number of

APLL input clock cycles to increment and decrement in the ASCNT registers. The ASCR.CNTEN register field must also be set propertly. The values to be written to the device are calculated as follows:

$$CNTEN = \begin{array}{l} 0 \text{ for } F_{IN} < 50MHz \\ 1 \text{ for } 50MHz \le F_{IN} < 100MHz \\ 2 \text{ for } F_{IN} \ge 100MHz \end{array}$$

$$ASCNT = round\left(\frac{F_{IN}}{2 * F_{MOD} * 2^{CNTEN}}\right) - 2$$

$$AID = round\left(\frac{F_{VCO}}{F_{OUT}} * \frac{S * 2^{35}}{(ASCNT + 2) * (1 - 0.5S)}\right)$$

All of the input parameters are the same as for center spread above. Note the small differences between these down-spread equations and the center-spread equations above. ASCNT here has 2 in the denominator while it has 4 for center-spread. AID has $2^{35}$ in the numerator for down-spread while it has $2^{34}$ for center-spread.

During spread-spectrum operation using the fractional output divider, FREM should be set to 0, FDEN should be set to 1 and FBP should be set to 0. The F1CR1.MODE field (F2CR1.MODE for APLL2) must be set to 1 when doing spread-spectrum modulation with the fractional output divider value. Spread-spectrum modulation only occurs when the APLL is locked (APLL1SR.ALK=1).

### 5.5.9 APLL Phase Adjustment

The phase of the APLL's output clock can be incremented or decremented. When the APLL's AFBDIV value is not an exact multiple of 0.5 then the phase adjustment step size is 1/8th of a VCO cycle. This phase step size is 30ps at maximum VCO frequency of 4180MHz and 33.7ps at minimum VCO frequency of 3715MHz. The ACR4.PDSS field specifies the phase decrement control signal, which can be the ACR4.DECPH bit or any of the four GPIOs. The ACR4.PISS field specifies the phase increment control signal, which can be the ACR4.INCPH bit or any of the four GPIOs. Phase is adjusted on every rising edge and every falling edge of the control signal. This phase adjustment affects the output of the APLL's output dividers (integer and fractional).

When the APLL's AFBDIV value is an exact multiple of 0.5 then the phase adjustment step size is one VCO cycle.

## 5.6 Output Clock Configuration

The ZL30264 and ZL30265 have six output clock signal pairs while the ZL30266 and ZL30267 have ten. Each output has individual divider, enable and signal format controls. In CMOS mode each signal pair can become two CMOS outputs, allowing the device to have up to 12 or 20 output clock signals. Also in CMOS mode, the OCxN pin can have an additional divider allowing the OCxN frequency to be an integer divisor of the OCxP frequency (example: OC3P 125MHz and OC3N 25MHz). The outputs can be aligned relative to each other, and the phases of output signals can be adjusted dynamically with high resolution.

### 5.6.1 Output Enable, Signal Format, Voltage and Interfacing

To use an output, the output driver must be enabled by setting OCxCR2.OCSF≠0, and the per-output dividers must be enabled by setting the appropriate bit in the OCEN register. The per-output dividers include the medium-speed divider, the low-speed divider and the associated phase adjustment/alignment circuitry and start/stop logic.

Using the OCxCR2.OCSF register field, each output pair can be disabled or configured as LVDS, LVPECL, HCSL, HSTL, or one or two CMOS outputs. When an output is disabled it is high impedance, and the output driver is in a low-power state. In CMOS mode, the OCxN pin can be disabled, in-phase or inverted vs. the OCxP pin. All of these options are specified by OCxCR2.OCSF. The clock to the output driver can inverted by setting OCxCR2.POL=1. The CMOS/HSTL output driver can be set to any of four drive strengths using OCxCR2.DRIVE.

When OCxCR2.OCSF=0001 the output driver is in LVDS mode. $V_{OD}$ is forced to 400mV and OCxDIFF.VOD is ignored. $V_{CM}$ can be configured in OCxDIFF.VCM, but the default value of 0000 is typically used to get $V_{CM}$=1.23V for LVDS.

When OCxCR2.OCSF=0010 the output driver is in programmable differential mode. In this mode the output swing ($V_{OD}$) can be set in OCxDIFF.VOD and the common-mode voltage can be set in OCxDIFF.VCM. Together these

fields allow the output signal to be customized to meet the requirements of the clock receiver and minimize the need for external components. By default, when OCSF=0010 the output is configured for LVPECL signal swing with a 1.23V common mode voltage. This gives a signal that can be AC-coupled (after a 100Ω termination resistor) to receivers that are LVPECL or that require a larger signal swing than LVDS. The output driver can also be configured for LVPECL output with standard 2.0V common-mode voltage by seting OCxDIFF.VCM for 2.0V and setting OCxREG.VREG appropriately.

In both LVDS mode and programmable differential mode the output driver requires a DC path through a 100Ω resistor between OCxP and OCxN for proper operation. This resistor is usually placed as close as possible to the receiver inputs to terminate the differential signal. If the receiver requires a common-mode voltage that cannot be matched by the output driver then the POS and NEG signals can be AC-coupled to the receiver after the 100Ω resistor.

HCSL mode requires a DC path through a 50Ω resistor to ground on each of OCxP and OCxN. Note that each of the OCxDIFF.VCM, OCxDIFF.VOD and OCxREG.VREG register fields has a particular setting required for HCSL signal format. See the descriptions of these fields for details.

Outputs are grouped into six power supply banks, VDDOA through VDDOF to allow CMOS or HSTL signal swing from 1.5V to 3.3V for glueless interfacing to neighboring components. 10-output products have outputs grouped into banks in a 2-1-2-2-1-2 arrangement, as shown in Figure 1. 6-output products have one output per bank. If OCSF is set to HSTL mode then a 1.5V power supply voltage should be used to get a standards-compliant HSTL output. Note that LVDS, LVPECL and HCSL signal formats must have a power supply of 2.5V or 3.3V. Also note that VDDO voltage must not exceed VDDH voltage.

### 5.6.2 Output Frequency Configuration

The frequency of each output is determined by the configuration of the APLL, the APLL's output dividers, and the per-output dividers. Each bank of outputs can be connected to either APLL's integer divider or fractional divider using the appropriate field in the OCMUX registers.

Each output has two output dividers, a 7-bit medium-speed divider (OCxCR1.MSDIV) and a 24-bit low-speed output divider (LSDIV field in the OCxDIV registers). These dividers are in series, medium-speed divider first then output divider. These dividers produce signals with 50% duty cycle for all divider values including odd numbers. The low-speed divider can only be used if the medium-speed divider is used (i.e. OCxCR1.MSDIV>0). The maxium input frequency to the medium-speed divider is 750MHz.

Since each output has its own independent dividers, the device can output families of related frequencies that have an APLL output frequency as a common multiple. For example, for Ethernet clocks, a 625MHz APLL output clock can be divided by four for one output to get 156.25MHz, divided by five for another output to get 125MHz, and divided by 25 for another output to get 25MHz. Similarly, for SDH/SONET clocks, a 622.08MHz APLL output clock can be divided by 4 to get 155.52MHz, by 8 to get 77.76MHz, by 16 to get 38.88MHz or by 32 to get 19.44MHz.

#### Two Different Frequencies in 2xCMOS Mode

When an output is in 2xCMOS mode it can be configured to have the frequency of the OCxN clock be an integer divisor of the frequency of the OCxP clock. Examples of where this can be useful:

- 125MHz on OCxP and 25MHz on OCxN for Ethernet applications
- 77.76MHz on OCxP and 19.44MHz on OCxN for SONET/SDH applications
- 25MHz on OCxP and 1Hz (i.e. 1PPS) on OCxN for telecom applications with Synchronous Ethernet and IEEE1588 timing

An output can be configured to operate like this by setting the LSDIV value in the OCxDIV registers to OCxP_freq / OCxN_freq - 1 and setting OCxCR3.LSSEL=0 and OCxCR3.NEGLSD=1. Here are some notes about this dual-frequency configuration option:

- In this mode only the medium speed divider is used to create the OCxP frequency. The low-speed divider is then used to divide the OCxP frequency down to the OCxN frequency. This means that the lowest OCxP frequency is the APLL divider output frequency divided by 128.

- An additional constraint is that the medium-speed divider must be configured to divide by 2 or more (i.e. must have OCxCR1.MSDIV≥1).

### 5.6.3 Output Duty Cycle Adjustment

For output frequencies less than or equal to 141.666MHz, the duty cycle of the output clock can be modified using the OCxDC.OCDC register field. This behavior is only available when MSDIV>0 and LSDIV > 1. When OCDC = 0 the output clock is 50%. Otherwise the clock signal is a pulse with a width of OCDC number of MSDIV output clock periods. The range of OCDC can create pulse widths of 1 to 255 MSDIV output clock periods. When OCxCR2.POL=0, the pulse is high and the signal is low the remainder of the cycle. When POL=1, the pulse is low and the signal is high the remainder of the cycle.

Note that duty cycle adjustment is done in the low-speed divider. Therefore when OCxCR3.LSSEL=0 the duty cycle of the output is not affected. Also, when a CMOS output is configured with OCxCR3.LSSEL=0 and OCxCR3.NEGLSD=1, the OCxN pin has duty cycle adjustment but the OCxP pin does not. This allows a higher-speed 50% duty cycle clock signal to be output on the OCxP pin and a lower-speed frame/phase/time pulse (e.g. 2kHz, 8kHz or 1PPS) to be output on the OCxN pin at the same time.

An output configured for CMOS or HSTL signal format should not be configured to have a duty cycle with high time shorter than 2ns or low time shorter than 2ns.

### 5.6.4 Output Phase Adjustment

The phase of an output signal can be shifted by 180° by setting OCxCR2.POL=1. In addition, the phase can be adjusted using the OCxPH.PHADJ register field. The adjustment is in units of bank source clock cycles. For example, if the bank source clock is 625MHz (from APLL1 for example) then one bank source clock cycle is 1.6ns, the smallest phase adjustment is 0.8ns, and the adjustment range is ±5.6ns.

### 5.6.5 Output-to-Output Phase Alignment

A 0-to-1 transition of the ACR1.DALIGN bit causes a simultaneous reset of the medium-speed dividers and low-speed dividers for all output clocks following APLL1 where OCxCR1.PHEN=1. After this reset, all PHEN=1 output clocks from the same APLL divider (IntDiv or FracDiv) are rising-edge aligned, with the phase of each output clock signal adjusted as specified by its OCxPH.PHADJ register field. Alignment of clocks from IntDiv with clocks from FracDiv is not supported. Similarly a 0-to-1 transition of the A2CR1.DALIGN bit aligns all output clocks following APLL2 where OCxCR1.PHEN=1. Alignment is not glitchess; i.e. it may cause a short high time or low time on participating output clock signals. A glitchless alignment can be accomplished by first stopping the clocks, then aligning them, then starting them. Output clock start and stop is described in section 5.6.7.

### 5.6.6 Output-to-Input Phase Alignment

The best approach for achieving output-to-input phase alignment is to use external feedback in which an OCx output is connected to an ICx input. To enable external feedback, set AFBDL.EXTFB=1, set AFBDL.FBSEL to specify the external feedback path, and provide the associated output-to-input wiring on the PCB. In this configuration the APLL, in a closed-loop manner, automatically phase-aligns all OCx outputs from an APLL to that APLL's selected reference. Any small error in this alignment due to wire delays can be compensated in the outputs' phase adjustment registers, OCxPH.PHADJ.

### 5.6.7 Output Clock Start and Stop

Output clocks can be stopped high or low or high-impedance. One use for this behavior is to ensure "glitchless" output clock operation while the output is reconfigured or phase aligned with some other signal.

Each output has an OCxSTOP register with fields to control this behavior. The OCxSTOP.MODE field specifies whether the output clock signal stops high, low, or high-impedance. The OCxSTOP.SRC field specifies the source of the stop signal. Options include control bits or one of the GPIO pins. When OCxSTOP.SRC=0001 the output clock is stopped when the corresponding bit is set in the STOPCR registers OR the MCR1.STOP bit is set.

When the stop mode is Stop High (OCxSTOP.MODE=x1) and the stop signal is asserted, the output clock is stopped after the next rising edge of the output clock. When the stop mode is Stop Low (OCxSTOP.MODE=x0) and the stop signal is asserted, the output clock is stopped after the next falling edge of the output clock. When the output is stopped, the output driver can optionally go high-impedance (OCxSTOP.MODE=1x). Internally the clock

signal continues to toggle while the output is stopped. When the stop signal is deasserted, the output clock resumes on the opposite edge that it stopped on. Low-speed output clocks can take long intervals before being stopped after the stop signal goes active. For example, a 1 Hz output could take up to 1 second to stop.

When OCxCR2.POL=1 the output stops on the opposite polarity that is specified by the OCxSTOP.MODE field.

Generally OCxCR1.MSDIV must be > 0 for this function to operate correctly since MSDIV=0 bypasses the start-stop circuits.

When MSDIV=0, OCxSTOP.MODE=11 (stop high then go high-impedance) can be used to make outputs high-impedance, but the action won't necessarily be glitchless. To use this behavior to get "stop *low* then go-impedance" behavior, OCxCR2.POL can be set to 1.

Note that when OCxCR3.NEGLSD=1 the start-stop logic is bypassed for the OCxN pin, and OCxN may not start/stop without glitches.

Each output has a status register (OCxSR) with several stop/start status bits. The STOPD bit is a real-time status bit indicating stopped or not stopped. The STOPL bit is a latched status bit that is set when the output clock has stopped. The STARTL bit is a latched status bit that is set when the output clock has started.

## 5.7 Microprocessor Interface

The device can communicate over a SPI interface or an $I^2C$ interface.

In SPI mode ZL3026x devices without internal EEPROM can be configured at reset to be a SPI slave to a processor master or a SPI master to an external EEPROM slave. (SPI master operation changes to SPI slave operation after auto-configuration from the external EEPROM is complete.) The ZL3026x devices with internal EEPROM can only be configured as a SPI slave to a processor master. All devices are always slaves on the $I^2C$ bus.

Section 5.2 describes reset pin settings required to configure the device for these interfaces.

### 5.7.1 SPI Slave

The device can present a SPI slave port on the CSN, SCLK, MOSI, and MISO pins. SPI is a widely used master/slave bus protocol that allows a master and one or more slaves to communicate over a serial bus. SPI masters are typically microprocessors, ASICs or FPGAs. Data transfers are always initiated by the master, which also generates the SCLK signal. The device receives serial data on the MOSI (Master Out Slave In) pin and transmits serial data on the MISO (Master In Slave Out) pin. MISO is high impedance except when the device is transmitting data to the bus master.

**Bit Order.** The register address and all data bytes are transmitted most significant bit first on both MOSI and MISO.

**Clock Polarity and Phase.** The device latches data on MOSI on the rising edge of SCLK and updates data on MISO on the falling edge of SCLK. SCLK does not have to toggle between accesses, i.e., when CSN is high.

**Device Selection.** Each SPI device has its own chip-select line. To select the device, the bus master drives its CSN pin low.

**Command and Address.** After driving CSN low, the bus master transmits an 8-bit command followed by a 16-bit register address. The available commands are shown below.

**Table 3 - SPI Commands**

| Command | Hex | Bit Order, Left to Right |
|---|---|---|
| Write Enable | 0x06 | 0000 0110 |
| Write | 0x02 | 0000 0010 |
| Read | 0x03 | 0000 0011 |
| Read Status | 0x05 | 0000 0101 |

**Read Transactions.** The device registers are accessible when EESEL=0. On ZL3026x devices with internal EEPROM, the EEPROM memory is accessible when the EESEL bit is 1. On ZL3026x devices without internal EEROM, the EESEL bit must be set to 0. After driving CSN low, the bus master transmits the read command followed by the 16-bit address. The device then responds with the requested data byte on MISO, increments its address counter, and prefetches the next data byte. If the bus master continues to demand data, the device continues to provide the data on MISO, increment its address counter, and prefetch the following byte. The read transaction is completed when the bus master drives CSN high. See Figure 6.

**Register Write Transactions.** The device registers are accessible when EESEL=0. After driving CSN low, the bus master transmits the write command followed by the 16-bit register address followed by the first data byte to be written. The device receives the first data byte on MOSI, writes it to the specified register, increments its internal address register, and prepares to receive the next data byte. If the master continues to transmit, the device continues to write the data received and increment its address counter. The write transaction is completed when the bus master drives CSN high. See Figure 8.

**EEPROM Writes (ZL30265, ZL30267 Only).** The internal EEPROM memory is accessible when the EESEL bit is 1. After driving CSN low, the bus master transmits the write enable command and then drives CSN high to set the internal write enable latch. The bus master then drives CSN low again and transmits the write command followed by the 16-bit address followed by the first data byte to be written. The device first copies the page to be written from EEPROM to its page buffer. The device then receives the first data byte on MOSI, writes it to its page buffer, increments its internal address register, and prepares to receive the next data byte. If the master continues to transmit, the device continues to write the data received to its page buffer and continues to increment its address counter. The address counter rolls over at the 32-byte page boundary (i.e. when the five least-significant address bits are 11111). When the bus master drives CSN high, the device transfers the data in the page buffer to the appropriate page in the EEPROM memory. See Figure 7 and Figure 8.

**EEPROM Read Status (ZL30265, ZL30267 Only)**. After the bus master drives CSN high to end an EEPROM write command, the EEPROM memory is not accessible for up to 5ms while the data is transferred from the page buffer. To determine when this transfer is complete, the bus master can use the Read Status command. After driving CSN low, the bus master transmits the Read Status command. The device then responds with the status byte on MISO. In this byte, the least significant bit is set to 1 if the transfer is still in progress and 0 if the transfer has completed.

**Early Termination of Bus Transactions.** The bus master can terminate SPI bus transactions at any time by pulling CSN high. In response to early terminations, the device resets its SPI interface logic and waits for the start of the next transaction. If a register write transaction is terminated prior to the SCLK edge that latches the least significant bit of a data byte, the data byte is not written. On devices with internal EEPROM, if an EEPROM write transaction is terminated prior to the SCLK edge that latches the least significant bit of a data byte, none of the bytes in that write transaction are written.

**Design Option: Wiring MOSI and MISO Together.** Because communication between the bus master and the device is half-duplex, the MOSI and MISO pins can be wired together externally to reduce wire count. To support this option, the bus master must not drive the MOSI/MISO line when the device is transmitting.

**AC Timing.** See Table 19 and Figure 18 for AC timing specifications for the SPI interface.

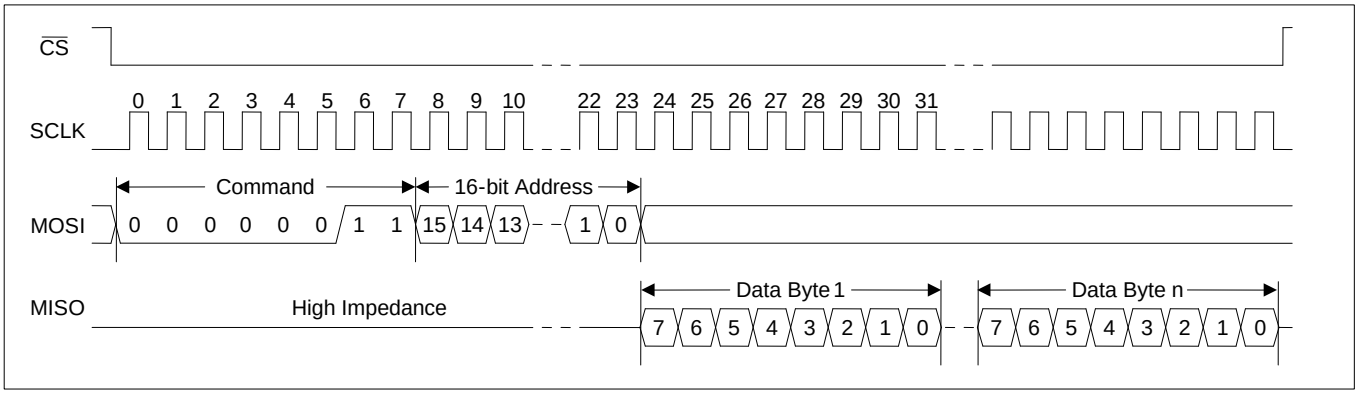



**Figure 6 - SPI Read Transaction Functional Timing**

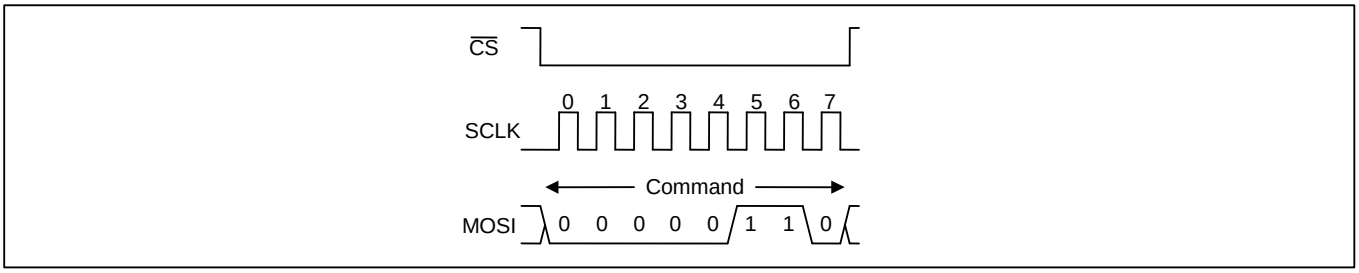



**Figure 7 - SPI Write Enable Transaction Functional Timing (ZL30265 and ZL30267 Only)**

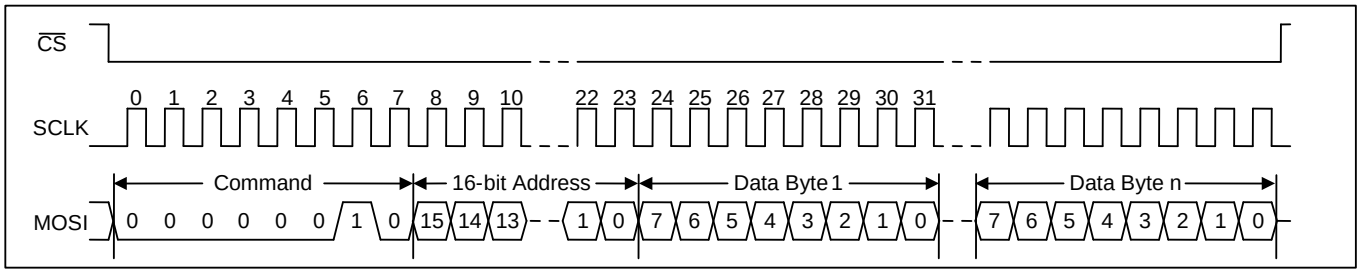



**Figure 8 - SPI Write Transaction Functional Timing**

### 5.7.2 SPI Master (ZL30264 and ZL30266 Only)

After reset these devices can present a SPI master port on the CSN, SCLK, MOSI, and MISO pins for auto-configuration using data read from an external SPI EEPROM. During auto-configuration the device is always the SPI master and generates the CSN and SCLK signals. The device transmits serial data on the the MOSI (Master Out Slave In) pin and receives serial data on the MISO (Master In Slave Out) pin.

**Bit Order.** The register address and all data bytes are transmitted most significant bit first on both MOSI and MISO.

**Clock Polarity and Phase.** The device latches data on MISO on the rising edge of SCLK and updates data on MOSI on the falling edge of SCLK.

**Device Selection.** Each SPI device has its own chip-select line. To select the external EEPROM, the device drives the CSN signal low.

**Command and Address.** After driving CSN low, the device transmits an 8-bit read command followed by a 16-bit register address. The read command is shown below.

| Command | Hex | Bit Order, Left to Right |
|---|---|---|
| Read | 0x03 | 0000 0011 |

**Read Transactions.** After driving CSN low, the device transmits the read command followed by the 16-bit register address. The external EEPROM then responds with the requested data byte on MISO, increments its address counter, and prefetches the next data byte. If the device continues to demand data, the EEPROM continues to provide the data on MISO, increment its address counter, and prefetch the following byte. The read transaction is completed when the device drives CSN high. See Figure 6.

**Writing the External EEPROM.** Due to the small package size and low pin count of the device, there is no way to use the ZL30264 or ZL30266 to write the external EEPROM. The auto-configuration data used by the ZL30264 or ZL30266 must be pre-programmed into the EEPROM by some other method, such as:

1. The EEPROM manufacturer can write the data to the EEPROM during production testing. This is a service they routinely provide.

2. A contract manufacturer or distributor can write the data to the EEPROM using a production EEPROM programmer before the EEPROM is mounted to the board.

### 5.7.3 I²C Slave

The device can present a fast-mode (400kbit/s) I²C slave port on the SCL and SDA pins. I²C is a widely used master/slave bus protocol that allows one or more masters and one or more slaves to communicate over a two-wire serial bus. I²C masters are typically microprocessors, ASICs or FPGAs. Data transfers are always initiated by the master, which also generates the SCL signal. The device is compliant with version 2.1 of the I²C specification.

The I²C interface on the device is a protocol translator from external I²C transactions to internal SPI transactions. This explains the slightly increased protocol complexity described in the paragraphs that follow.

**Read Transactions.** The device registers are accessible when the EESEL bit is 0. On ZL30265 and ZL30267 the internal EEPROM memory is accessible when the EESEL bit is 1. On ZL30264 and ZL30266 the EESEL bit must be set to 0. The bus master first does an I²C write to the device. In this transaction three bytes are written: the SPI Read command (see Table 3), the upper byte of the register address, and the lower byte of the register address. The bus master then does an I²C read. During each acknowledge (A) bit the device fetches data from the read address and then increments the read address. The device then transmits the data to the bus master during the next 8 SCL cycles. The bus master terminates the read with a not-acknowledge (NA) followed by a STOP condition (P). See Figure 9. After the I²C write there can be unlimited idle time on the bus before the I²C read, but the device cannot tolerate other I²C bus traffic between the I²C write and the I²C read. Care must be taken to ensure that the I²C read is the first command on the bus after the I²C write to ensure the two-part read transaction happens correctly.

**Register Write Transactions.** The device registers are accessible when the EESEL bit is 0. The bus master does an I²C write to the device. The first three bytes of this transaction are the SPI Write command (see Table 3), the upper byte of the register address, and the lower byte of the register address. Subsequent bytes are data bytes to be written. After each data byte is received, the device writes the byte to the write address and then increments the write address. The bus master terminates the write with a STOP condition (P). See Figure 10.

**EEPROM Writes (ZL30265 and ZL30267 Only).** The EEPROM memory is accessible when the EESEL bit is 1. The bus master first does an I²C write to transmit the SPI Write Enable command (see Table 3) to the device. The bus master then does an I²C write to transmit data to the device as described in the Register Write Transactions paragraph above. See Figure 11.

**EEPROM Read Status (ZL30265 and ZL3026 Only)**. The bus master first does an I²C write to transmit the SPI Read Status command (see Table 3) to the device. The bus master then does an I²C read to get the status byte. In this byte, the least significant bit is set to 1 if the transfer is still in progress and 0 if the transfer has completed. See Figure 12. Similar to read transactions described above, the I²C write and the I²C read cannot be separated by other I²C bus traffic.

**I²C Features Not Supported by the Device.** The I²C specification has several optional features that are not supported by the device. These are: 3.4Mbit/s high-speed mode (Hs-mode), 10-bit device addressing, general call address, software reset, and device ID. The device does not hold SCL low to force the master to wait.

**I²C Slave Address.** By default the upper 5 bits of the device's 7-bit slave address are fixed at 11101 and the lower 2 bits can be pin-configured for any of three values as shown in the table in section 5.2. For a device that can auto-configure from EEPROM at power-up, its I²C slave address can be set to any value during auto-configuration at power-up by writing the the I2CA register as part of the configuration script.

**Bit Order.** The I²C specification requires device address, register address and all data bytes to be transmitted most significant bit first on the SDA signal.

Note: as required by the I²C specification, when power is removed from the device, the SDA and SCL pins are left floating so they don't obstruct the bus lines.

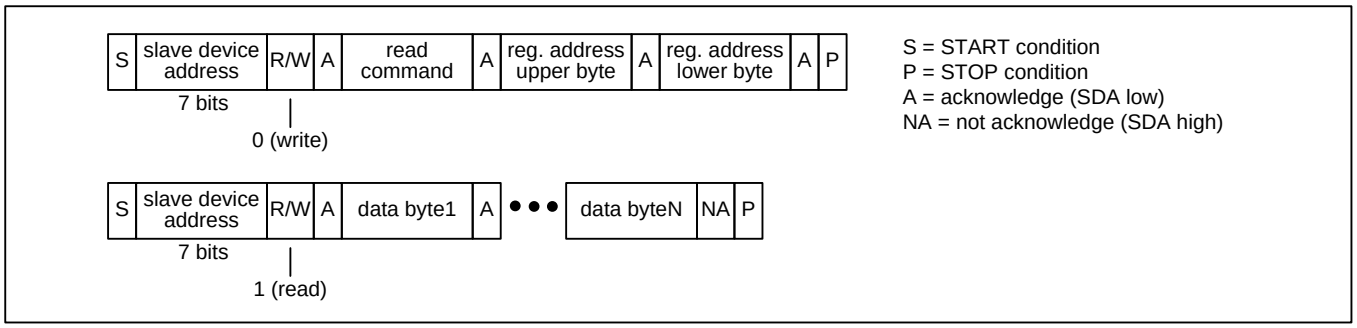



**Figure 9 - I²C Read Transaction Functional Timing**

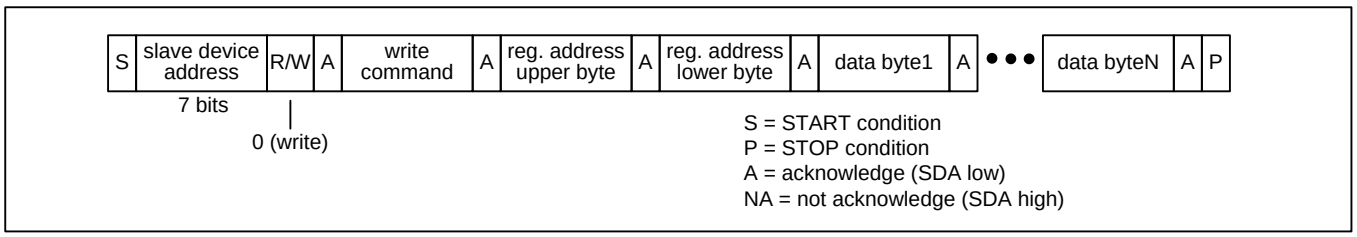



**Figure 10 - I²C Register Write Transaction Functional Timing**

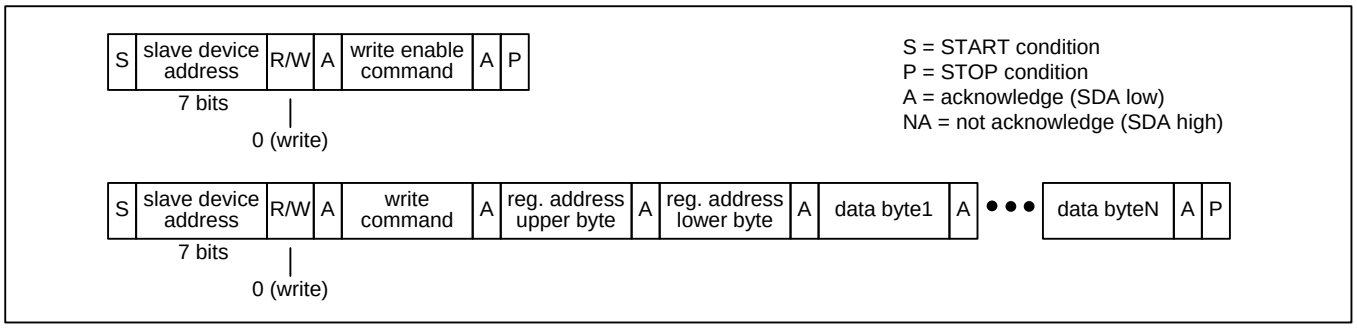



**Figure 11 - I²C EEPROM Write Transaction Functional Timing (ZL30265 and ZL30267 Only)**

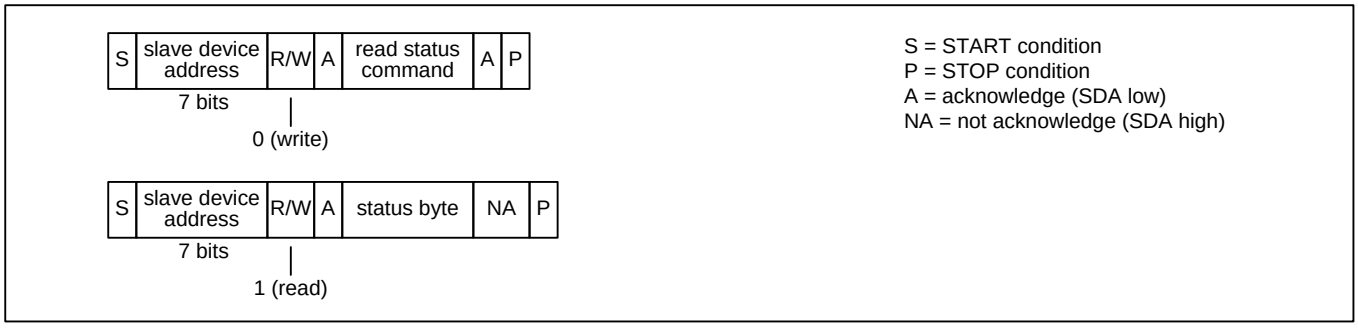



**Figure 12 - I²C EEPROM Read Status Transaction Functional Timing (ZL30265 and ZL30267 Only)**

Note: In Figure 9 through Figure 12, a STOP condition (P) immediately followed by a START condition (S) can be replaced by a repeated START condition (Sr) as described in the I²C specification.

## 5.8 Interrupt Logic

Any of the GPIO pins can be configured as an interrupt-request output by setting the appropriate GPIOxC field in the GPIOCR registers to one of the status output options (01xx) and configuring the appropriate GPIOxSS register to follow the INTSR.INT bit. If system software is written to poll rather than receive interrupt requests, then software can read the INTSR.INT bit first to determine if any interrupt requests are active in the device.

Many of the latched status bits in the device can be the source of an interrupt request if their corresponding interrupt enable bits are set. The device's interrupt logic is shown in Figure 13. See the register map (Table 4) and the status register descriptions in section 6.3.2 for descriptions of the register bits shown in the figure.

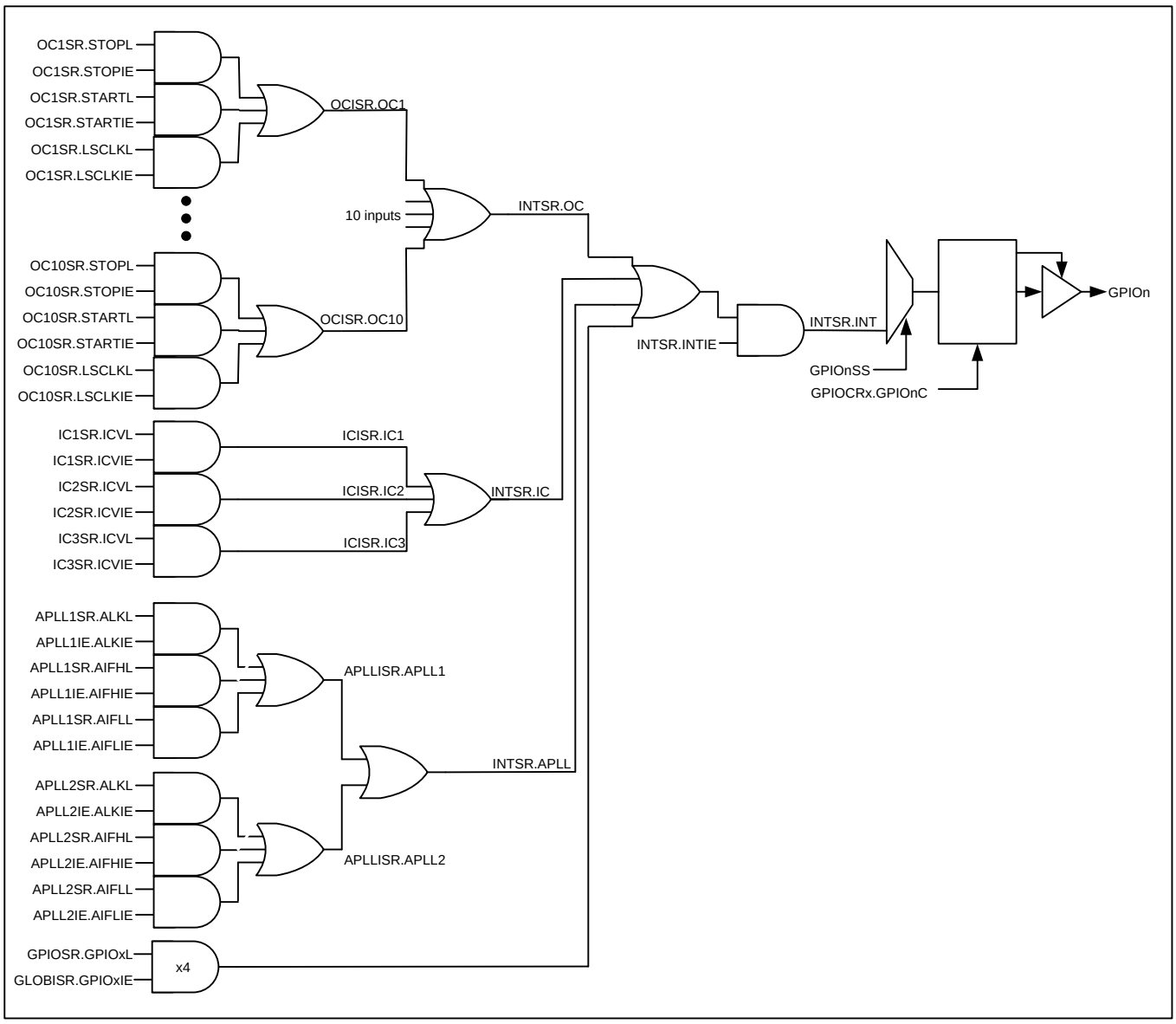



**Figure 13 - Interrupt Structure**

## 5.9 Reset Logic

The device has three reset controls: the RSTN pin, and the hard reset (HRST) and soft reset (SRST) bits in MCR1. The RSTN pin asynchronously resets the entire device. When the RSTN pin is low all internal registers are reset to their default values. When RSTN returns high the device's auto-configuration boot controller is started. **The RSTN pin must be asserted once after power-up.** Reset should be asserted for at least 1µs. See section 5.9.1 below for important details about using an external RC reset circuit with the RSTN pin.

Asserting the MCR1.HRST (hard reset) bit is functionally similar to asserting the RSTN pin. The HRST bit resets the entire device except for the microprocessor interface, the HRST bit itself, the I2CA register, and CFGSR.IF[1:0]. While HRST=1 the device accepts register writes so that HRST can be set back to 0, but register reads are not allowed. When HRST is set back to 0, the TEST and AC[2:0] pins are sampled as described in section 5.2, but, unlike when RSTN is deasserted, the IF[1:0] pins are not sampled so that the device remains in the same interface mode (SPI or $I^2C$) and maintains the same slave address when in $I^2C$ mode. When HRST is set back to 0, the device's auto-configuration boot controller is started after a 1 to 3µs delay.

The MCR1.SRST (soft reset) bit resets the entire device except for the microprocessor interface, the SRST bit itself, the MCR1.HRST bit, the I2CA register, and the CFGSR register. When the SRST bit is asserted the device's auto-configuration boot controller is **not** started.

Microsemi recommends holding RSTN low while the internal ring oscillator starts up and stabilizes. An incorrect reset condition could result if RSTN is released before the oscillator has started up completely.

**Important:** System software must wait at least 100µs after RSTN is deasserted and wait for GLOBISR.BCDONE=1 before configuring the device.

### 5.9.1 Design Considerations for Using an External RC Reset Circuit

When the power supply arrangement for the device has VDDH=VDDL (3.3V or 2.5V) an external RC reset circuit can be used to reset the device during power-up with no additional considerations.

When the power supply arrangement for the device has VDDL=1.8V then the board designer should choose one of two options: (a) a power-on-reset (POR) chip such as a Texas Instruments TPS3839 should be used instead of an external RC reset circuit, or (b) the device's VDDIO pin must be wired to VDDL. In option (a) during the interval between VDDH ramping and VDDL ramping the RSTN pin can briefly behave as an output driving high. Therefore a current-limiting series resistor should be used between the POR chip and the device RSTN pin.

The possible disadvantage of option (b) is that VDDIO, the power supply for all SPI/I2C pins and all GPIO pins, could be too low if neighboring devices operate at power supply voltages higher than VDDL. One exception to this disadvantage would be the I2C interface. Since I2C's logic-high voltage is set by pull-up resistors, those resistors can be externally wired to a voltage higher than VDDIO up to 3.3V. The SCL/SCLK and SDA/MOSI pins are 3.3V tolerant.

## 5.10 Power-Supply Considerations

Due to the multi-power-supply nature of the device, some I/Os have parasitic diodes between a lower-voltage supply and a higher-voltage supply. When ramping power supplies up or down, care must be taken to avoid forward-biasing these diodes because it could cause latchup. Two methods are available to prevent this. The first method is to place a Schottky diode external to the device between the lower-voltage supply and the higher-voltage supply to force the higher-voltage supply to be within one parasitic diode drop of the lower-voltage supply. The second method is to ramp up the higher-voltage supply first and then ramp up the lower-voltge supply.

**Important Note**: The voltages on VDDL, VDDIO, and all VDDOx pins must not exceed VDDH. Not complying with this requirement may damage the device.

## 5.11 Auto-Configuration from EEPROM or ROM

For ZL30264 and ZL30266, the device optionally can configure itself at reset from an internal ROM. The ROM stores eight configurations, known as configurations 0 through 7. As described in section 5.2.1, IF[1:0] must be 00, 01 or 10 at reset, and the device configuration to be used is specified by the values of the AC[2:0] pins at reset (0 through 7). Descriptions of the standard-product ROM configurations are available from Microsemi.

For ZL30264 and ZL30266, the device optionally can configure itself at reset from an external EEPROM connected to its SPI interface. The EEPROM can store up to eight configurations, known as configurations 0 through 7. As described in section 5.2.1, IF[1:0] must be 11 at reset, and the device configuration to be used is specified by the values of the AC[2:0] pins at reset (0 through 7).

For ZL30265 and ZL30267, the internal EEPROM memory can store up to eight device configurations, known as configurations 0 through 7. As described in section 5.2.2, the device configuration to be used is specified by the values of the AC[2:0] pins at reset.

### 5.11.1 Generating Device Configurations

Device configurations are most easily generated using the evaluation software. This is true for auto-configurations stored in internal or external EEPROM and for configurations that are written to the device by a system processor. See section 5.12 for guidance if device configurations must be developed without using the evaluation software.

### 5.11.2 Direct EEPROM Write Mode (ZL30265 and ZL30267 Only)

To simplify writing the device's internal EEPROM during manufacturing, the device has a test mode known as direct EEPROM write mode. The device enters this mode when TEST=1, AC[2:0]=000 and IF[1:0]=11 on the rising edge of RSTN. In this mode the EEPROM memory is mapped into the address map and can be written as needed to store configuration scripts in the device. Device registers are not accessible in this mode. The device exits this mode on the rising edge of RSTN. Note: the device drives the MISO pin continually during this mode. Therefore this mode cannot be used when MOSI and MISO are tied together as described in the **Design Option: Wiring MOSI and MISO Together** paragraph in section 5.7.1.

### 5.11.3 Holding Other Devices in Reset During Auto-Configuration

Using the appropriate GPIOCR and GPIO0SS registers, a GPIO pin can be configured to follow the GLOBISR.BCDONE status bit. This GPIO can then be used as a reset signal to hold other devices (device that use clocks from this device) in reset while the device configures itself. As an example, to configure GPIO0 to follow BCDONE with 0=reset add the following writes at the beginning of the configuration file: write 0x1F to GPIO0SS and write 0x04 to GPIOCR1.

## 5.12 Configuration Sequence

Device configurations are most easily generated using the evaluation software, which automatically generates configurations that follow Microsemi's suggested sequence. To develop device configurations manually (i.e. from device documentation rather than the evaluation software) see Application Note ZLAN-590 for Microsemi's suggested device configuration sequence.

## 5.13 Power Supply Decoupling and Layout Recommendations

Application Note ZLAN-592 describes recommended power supply decoupling and layout practices.

## 5.14 Choosing Among Core Power Supply Options

The device supports the following core supply voltage options:

| VDDH | VDDL |
|---|---|
| 3.3V | 3.3V |
| 3.3V | 1.8V |
| 2.5V | 2.5V |
| 2.5V | 1.8V |

Choosing the best option depends on several factors including supply voltages available on the board, willingness to use low-dropout (LDO) linear regulators to make local power supplies for the device, board power supply noise and mitigation strategies, target jitter performance, and how many device resources are enabled.

Starting with the VDDH=VDDL=3.3V option, the advantages of this option are (1) the device only requires a single power supply voltage (assuming all output driver VDDOx supplies are also 3.3V), and (2) internal regulation is used for the APLL, maximizing power supply noise rejection. The disadvantages are (1) power consumption is higher than other options, and (2) power dissipation can exceed the package limits at high temperatures if too many

blocks are enabled. See section 8 for some guidance on how many blocks can be enabled simultaneously, and see the evaluation software GUI for power calculations for any desired device configuration.

The VDDH=3.3V, VDDL=1.8V option does require two core power supply voltages, but internal regulation is used for the APLL, maximizing power supply noise rejection. Also this option does not have either of the two disadvantages of the VDDH=VDDL=3.3V option. *If the application can provide 3.3V and 1.8V supplies to the device, this option is highly recommended as a good balance of lower power consumption and better power supply noise rejection*.

The VDDH=VDDL=2.5 option is for applications that do not have a 3.3V power supply and do not want to provide an LDO to make a 3.3V supply. The advantages of this option are (1) the device only requires a single power supply voltage (assuming all output driver VDDOx supplies are also 2.5V), and (2) lower power consumption than the VDDH=VDDL=3.3V option. The disadvantage is that internal APLL regulators are bypassed and the APLL runs directly from the VDDH supply, which leaves the device more susceptible to power supply noise. This susceptibility can be mitigated using good power supply noise filtering and further mitigated with a dedicated LDO for the device.

The VDDH=2.5V, VDDL=1.8V option provides even lower power consumption than the VDDH=VDDL=2.5V option. The disadvantages are (1) it requires two core power supply voltages, and (2) just like the VDDH=VDDL=2.5V case, internal APLL regulators are bypassed and the APLL runs directly from the VDDH supply, which leaves the device more susceptible to power supply noise. This susceptibility can be mitigated using good power supply noise filtering and further mitigated with a dedicated LDO for the device.

See application note ZLAN-630 for power supply noise rejection performance.

# 6. Register Descriptions

Table 4 shows the register map. In each register, bit 7 is the MSb and bit 0 is the LSb. Register addresses not listed are reserved. Bits marked "—" are reserved and must be written with 0. Writing other values to these registers may put the device in a factory test mode resulting in undefined operation. Bits labeled "0" or "1" must be written with that value for proper operation. Register fields with underlined names are read-only fields; writes to these fields have no effect. All other fields are read-write. Register fields are described in detail in the register descriptions that follow Table 4.

## 6.1 Register Types

### 6.1.1 Status Bits

The device has two types of status bits. Real-time status bits are read-only and indicate the state of a signal at the time it is read. Latched status bits are set when a signal changes state (low-to-high, high-to-low, or both, depending on the bit) and cleared when written with a logic 1 value. Writing a 0 has no effect. When set, some latched status bits can cause an interrupt request if enabled to do so by corresponding interrupt enable bits. Status bits marked "—" are reserved and must be ignored.

### 6.1.2 Configuration Fields

Configuration fields are read-write. During reset, each configuration field reverts to the default value shown in the register definition. Configuration register bits marked "—" are reserved and must be written with 0.

### 6.1.3 Multiregister Fields

Multiregister fields—such as AFBDIV[41:0] in registers AFBDIV1 through AFBDIV6—must be handled carefully to ensure that the bytes of the field remain consistent during writes. A write access to a multiregister field is accomplished by writing all the registers of the field in order from smallest address to largest. Writes to registers other than the last register in the field (i.e. the register with the largest address) are stored in a transfer register. When the last register of the field is written, the entire multiregister field is updated simultaneously from the transfer register. If the last register of the field is not written, the field is not updated.

Read accesses from multiregister fields are the same as normal registers since the multiregister fields are not modified by the device.

An exception for AFBDIV happens when AFBDL.RDCUR=1. In this case reads come directly from the internal logic and don't affect the transfer register. Writes behave the same as above. A similar exception exists for A2FBDIV when A2FBDL.RDCUR=1.

The multiregister fields are:

| Field | Registers | Type |
|---|---|---|
| APLL1 AFBDIV[41:0] | AFBDIV1 to AFBDIV6 | Read/Write |
| APLL2 AFBDIV[41:0] | A2FBDIV1 to A2FBDIV6 | Read/Write |
| APLL1 FDIV[39:0] | F1DIV1 to F1DIV5 | Read/Write |
| APLL2 FDIV[39:0] | F2DIV1 to F2DIV5 | Read/Write |

### 6.1.4 Bank-Switched Registers (ZL30265 and ZL30267 Only)

The EESEL register is a bank-select control field that maps the device registers into the memory map at address 0x1 and above when the EESEL bit is 0 and maps the EEPROM memory into the memory map at address 0x1 and above when the EESEL bit is 1. The EESEL register itself is always in the memory map at address 0x0.

## 6.2 Register Map

**Table 4 - Register Map**

| ADDR | REGISTER | BIT 7 | BIT 6 | BIT 5 | BIT 4 | BIT 3 | BIT 2 | BIT 1 | BIT 0 |
|---|---|---|---|---|---|---|---|---|---|
| **Global Configuration Registers** | | | | | | | | | |
| 00h | EESEL | EESEL | — | — | — | — | — | — | — |
| 01 | MCR1 | SRST | HRST | STOP | — | ROSCD | AINCDIS | ODMISO | — |
| 02 | MCR2 | XAMCK | IC3MCK | IC2MCK | IC1MCK | — | DBL | XAB[1:0] | |
| 03 | PLLEN | — | — | — | — | — | — | APLL2EN | APLL1EN |
| 04 | ICEN | — | — | — | — | — | IC3EN | IC2EN | IC1EN |
| 05 | OCEN1 | OC8EN | OC7EN | OC6EN | OC5EN | OC4EN | OC3EN | OC2EN | OC1EN |
| 06 | OCEN2 | — | — | — | — | — | — | OC10EN | OC9EN |
| 07 | OCMUX1 | — | — | OCMUXC[1:0] | | OCMUXB[1:0] | | OCMUXA[1:0] | |
| 08 | OCMUX2 | — | — | OCMUXF[1:0] | | OCMUXE[1:0] | | OCMUXD[1:0] | |
| 09 | STOPCR1 | OC8 | OC7 | OC6 | OC5 | OC4 | OC3 | OC2 | OC1 |
| 0A | STOPCR2 | — | — | — | — | — | — | OC10 | OC9 |
| 0B | GPIOCR1 | GPIO1C[3:0] | | | | GPIO0C[3:0] | | | |
| 0C | GPIOCR2 | GPIO3C[3:0] | | | | GPIO2C[3:0] | | | |
| 0D | GPIO0SS | REG[4:0] | | | | | BIT[2:0] | | |
| 0E | GPIO1SS | REG[4:0] | | | | | BIT[2:0] | | |
| 0F | GPIO2SS | REG[4:0] | | | | | BIT[2:0] | | |
| 10 | GPIO3SS | REG[4:0] | | | | | BIT[2:0] | | |
| 11 | I2CA | — | I2CA[6:0] | | | | | | |
| **Status Registers** | | | | | | | | | |
| 30 | ID1 | IDU[7:0] | | | | | | | |
| 31 | ID2 | IDL[3:0] | | | | REV[3:0] | | | |
| 40 | CFGSR | CFGD | — | IF[1:0] | | TEST | AC[2:0] | | |
| 41 | GPIOSR | GPIO3L | GPIO2L | GPIO1L | GPIO0L | GPIO3 | GPIO2 | GPIO1 | GPIO0 |
| 42 | INTSR | — | — | OC | IC | — | APLL | INTIE | INT |
| 43 | GLOBISR | BCDONE | — | — | — | GPIO3IE | GPIO2IE | GPIO1IE | GPIO0IE |
| 44 | ICISR | — | — | — | — | XA | IC3 | IC2 | IC1 |
| 45 | OCISR1 | OC8 | OC7 | OC6 | OC5 | OC4 | OC3 | OC2 | OC1 |

| ADDR | REGISTER | BIT 7 | BIT 6 | BIT 5 | BIT 4 | BIT 3 | BIT 2 | BIT 1 | BIT 0 |
|---|---|---|---|---|---|---|---|---|---|
| 46 | OCISR2 | — | — | — | — | — | — | OC10 | OC9 |
| 47 | APLLISR | — | — | — | — | — | — | APLL2 | APLL1 |
| 48 | APLL1SR | AIFLL | AIFL | AIFHL | AIFH | ALKL | ALK | — | SELREF |
| 49 | APLL2SR | AIFLL | AIFL | AIFHL | AIFH | ALKL | ALK | — | SELREF |
| 4A | APLL1IE | AIFLIE | — | AIFHIE | — | ALKIE | — | — | — |
| 4B | APLL2IE | AIFLIE | — | AIFHIE | — | ALKIE | — | — | — |
| 4C | XASR | — | — | — | — | XAIE | XAVL | XAV | — |
| 4D | IC1SR | — | — | — | — | ICVIE | ICVL | ICV | — |
| 4E | IC2SR | — | — | — | — | ICVIE | ICVL | ICV | — |
| 4F | IC3SR | — | — | — | — | ICVIE | ICVL | ICV | — |
| 50 | OC1SR | LSCLKIE | LSCLKL | LSCLK | STARTIE | STARTL | STOPIE | STOPL | STOPD |
| 51 | OC2SR | LSCLKIE | LSCLKL | LSCLK | STARTIE | STARTL | STOPIE | STOPL | STOPD |
| 52 | OC3SR | LSCLKIE | LSCLKL | LSCLK | STARTIE | STARTL | STOPIE | STOPL | STOPD |
| 53 | OC4SR | LSCLKIE | LSCLKL | LSCLK | STARTIE | STARTL | STOPIE | STOPL | STOPD |
| 54 | OC5SR | LSCLKIE | LSCLKL | LSCLK | STARTIE | STARTL | STOPIE | STOPL | STOPD |
| 55 | OC6SR | LSCLKIE | LSCLKL | LSCLK | STARTIE | STARTL | STOPIE | STOPL | STOPD |
| 56 | OC7SR | LSCLKIE | LSCLKL | LSCLK | STARTIE | STARTL | STOPIE | STOPL | STOPD |
| 57 | OC8SR | LSCLKIE | LSCLKL | LSCLK | STARTIE | STARTL | STOPIE | STOPL | STOPD |
| 58 | OC9SR | LSCLKIE | LSCLKL | LSCLK | STARTIE | STARTL | STOPIE | STOPL | STOPD |
| 59 | OC10SR | LSCLKIE | LSCLKL | LSCLK | STARTIE | STARTL | STOPIE | STOPL | STOPD |
| **APLL Configuration Registers** | | | | | | | | | |
| | **APLL1 Registers** | | | | | | | | |
| 100 | ACR1 | ENFID | DALIGN | EXTSS[1:0] | | USEFDIV | ENFDIV | BYPASS | INDBL |
| 101 | ACR2 | — | — | — | — | INTDIV[3:0] | | | |
| 102 | ACR3 | INMON | EXTSW | ALTMUX[2:0] | | APLLMUX[2:0] | | | |
| 103 | ACR4 | DECPH | PDSS[2:0] | | | INCPH | PISS[2:0] | | |
| 105 | AFBDL | EXTFB | FBSEL[1:0] | | RDCUR | — | AFBDL[2:0] | | |
| 106 | AFBDIV1 | AFBDIV[7:0] | | | | | | | |
| 107 | AFBDIV2 | AFBDIV[15:8] | | | | | | | |
| 108 | AFBDIV3 | AFBDIV[23:16] | | | | | | | |
| 109 | AFBDIV4 | AFBDIV[31:24] | | | | | | | |
| 10A | AFBDIV5 | AFBDIV[39:32] | | | | | | | |
| 10B | AFBDIV6 | — | — | — | — | — | — | AFBDIV[41:40] | |
| 10C | AFBDEN1 | AFBDEN[7:0] | | | | | | | |
| 10D | AFBDEN2 | AFBDEN[15:8] | | | | | | | |
| 10E | AFBDEN3 | AFBDEN[23:16] | | | | | | | |
| 10F | AFBDEN4 | AFBDEN[31:24] | | | | | | | |
| 110 | AFBREM1 | AFBREM[7:0] | | | | | | | |
| 111 | AFBREM2 | AFBREM[15:8] | | | | | | | |
| 112 | AFBREM3 | AFBREM[23:16] | | | | | | | |
| 113 | AFBREM4 | AFBREM[31:24] | | | | | | | |
| 114 | AFBBP | AFBBP[7:0] | | | | | | | |
| 115 | AIDCR | DECF | FDSS[2:0] | | | INCF | FISS[2:0] | | |
| 116 | ASCR | — | CNTEN[1:0] | | DWNEN | ENSS | SPRDSS[2:0] | | |
| 117 | ASCNT1 | ASCNT[7:0] | | | | | | | |
| 118 | ASCNT2 | ASCNT[15:8] | | | | | | | |
| 119 | AID1 | AID[7:0] | | | | | | | |

| ADDR | REGISTER | BIT 7 | BIT 6 | BIT 5 | BIT 4 | BIT 3 | BIT 2 | BIT 1 | BIT 0 |
|---|---|---|---|---|---|---|---|---|---|
| 11A | AID2 | AID[15:8] | | | | | | | |
| 11B | AID3 | AID[23:16] | | | | | | | |
| 11C | AID4 | AID[31:24] | | | | | | | |
| | **APLL1 Fractional Divider Registers** | | | | | | | | |
| 140 | F1CR1 | MODE[3:0] | | | | — | FDL[2:0] | | |
| 141 | F1DIV1 | FDIV[7:0] | | | | | | | |
| 142 | F1DIV2 | FDIV[15:8] | | | | | | | |
| 143 | F1DIV3 | FDIV[23:16] | | | | | | | |
| 144 | F1DIV4 | FDIV[31:24] | | | | | | | |
| 145 | F1DIV5 | FDIV[39:32] | | | | | | | |
| 146 | F1DEN1 | FDEN[7:0] | | | | | | | |
| 147 | F1DEN2 | FDEN[15:8] | | | | | | | |
| 148 | F1DEN3 | FDEN[23:16] | | | | | | | |
| 149 | F1DEN4 | FDEN[31:24] | | | | | | | |
| 14A | F1REM1 | FREM[7:0] | | | | | | | |
| 14B | F1REM2 | FREM[15:8] | | | | | | | |
| 14C | F1REM3 | FREM[23:16] | | | | | | | |
| 14D | F1REM4 | FREM[31:24] | | | | | | | |
| 14E | F1BP | — | — | FBP[5:0] | | | | | |
| | **APLL2 Registers** | | | | | | | | |
| 180 | A2CR1 | same as APLL1 registers | | | | | | | |
| … | … | | | | | | | | |
| 186 | A2FBDIV1 | | | | | | | | |
| … | … | | | | | | | | |
| 19C | A2ID4 | | | | | | | | |
| | **APLL2 Fractional Divider Registers** | | | | | | | | |
| 1C0 | F2CR1 | same as APLL1 fractional divider registers | | | | | | | |
| … | … | | | | | | | | |
| 1CE | F2BP | | | | | | | | |
| **Output Clock Configuration Registers** | | | | | | | | | |
| | **OC1 Registers** | | | | | | | | |
| 200 | OC1CR1 | PHEN | MSDIV[6:0] | | | | | | |
| 201 | OC1CR2 | — | POL | DRIVE[1:0] | | OCSF[3:0] | | | |
| 202 | OC1DIFF | VCM[3:0] | | | | VOD[3:0] | | | |
| 203 | OC1REG | — | — | — | — | VREG[3:0] | | | |
| 204 | OC1CR3 | SRLSEN | — | NEGLSD | LSSEL | — | ASQUEL | — | LSDIV[24] |
| 205 | OC1DIV1 | LSDIV[7:0] | | | | | | | |
| 206 | OC1DIV2 | LSDIV[15:8] | | | | | | | |
| 207 | OC1DIV3 | LSDIV[23:16] | | | | | | | |
| 208 | OC1DC | OCDC[7:0] | | | | | | | |
| 209 | OC1PH | — | — | — | — | PHADJ[3:0] | | | |
| 20A | OC1STOP | — | SRC[3:0] | | | | NEGLSD | MODE[1:0] | |
| | **OC2 Registers** | | | | | | | | |
| 210 | OC2CR1 | same as OC1 registers | | | | | | | |
| … | … | | | | | | | | |
| 21A | OC2STOP | | | | | | | | |
| | **OC3 Registers** | | | | | | | | |
| 220 | OC3CR1 | same as OC1 registers | | | | | | | |

| ADDR | REGISTER | BIT 7 | BIT 6 | BIT 5 | BIT 4 | BIT 3 | BIT 2 | BIT 1 | BIT 0 |
|---|---|---|---|---|---|---|---|---|---|
| … | … | | | | | | | | |
| 22A | OC3STOP | | | | | | | | |
| | **OC4 Registers** | | | | | | | | |
| 230 | OC4CR1 | same as OC1 registers | | | | | | | |
| … | … | | | | | | | | |
| 23A | OC4STOP | | | | | | | | |
| | **OC5 Registers** | | | | | | | | |
| 240 | OC5CR1 | same as OC1 registers | | | | | | | |
| … | … | | | | | | | | |
| 24A | OC5STOP | | | | | | | | |
| | **OC6 Registers** | | | | | | | | |
| 250 | OC6CR1 | same as OC1 registers | | | | | | | |
| … | … | | | | | | | | |
| 25A | OC6STOP | | | | | | | | |
| | **OC7 Registers** | | | | | | | | |
| 260 | OC7CR1 | same as OC1 registers | | | | | | | |
| … | … | | | | | | | | |
| 26A | OC7STOP | | | | | | | | |
| | **OC8 Registers** | | | | | | | | |
| 270 | OC8CR1 | same as OC1 registers | | | | | | | |
| … | … | | | | | | | | |
| 27A | OC8STOP | | | | | | | | |
| | **OC9 Registers** | | | | | | | | |
| 280 | OC9CR1 | same as OC1 registers | | | | | | | |
| … | … | | | | | | | | |
| 28A | OC9STOP | | | | | | | | |
| | **OC10 Registers** | | | | | | | | |
| 290 | OC10CR1 | same as OC1 registers | | | | | | | |
| … | … | | | | | | | | |
| 29A | OC10STOP | | | | | | | | |
| **Input Clock Configuration** | | | | | | | | | |
| 300 | XACR1 | — | POL | DISMON | VALTIME[2:0] | | | HSDIV[1:0] | |
| 301 | XACR2 | XOAMP[7:0] | | | | | | | |
| 302 | XACR3 | XBCAP[3:0] | | | | XACAP[3:0] | | | |
| 303 | IC1CR1 | — | POL | DISMON | VALTIME[2:0] | | | HSDIV[1:0] | |
| 304 | IC2CR1 | — | POL | DISMON | VALTIME[2:0] | | | HSDIV[1:0] | |
| 305 | IC3CR1 | — | POL | DISMON | VALTIME[2:0] | | | HSDIV[1:0] | |

## 6.3 Register Definitions

### 6.3.1 Global Configuration Registers

**Register Name:** EESEL
**Register Description:** EEPROM Memory Selection Register
**Register Address:** 00h

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **Name** | EESEL | — | — | — | — | — | — | — |
| **Default** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 7: EEPROM Memory Select (EESEL).** This bit is a bank-select that specfies whether device register space or EEPROM memory is mapped into addresses 0x1 and above. This applies only to the ZL30265 and ZL30267. The ZL30264 and ZL30266 do not have internal EEPROM memory. Note that ROMSEL has priority over EESEL. See sections 5.7 and 6.1.4.

0 = Device registers
1= EEPROM memory

**Register Name:** MCR1
**Register Description:** Master Configuration Register 1
**Register Address:** 01h

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **Name** | SRST | HRST | STOP | — | ROSCD | AINCDIS | ODMISO | — |
| **Default** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 7: Soft Reset (SRST).** This bit resets the entire device except for the microprocessor interface, the SRST bit itself, the MCR1.HRST bit, the I2CA register, and CFGSR bits 5:0. When SRST is active, the register fields with pin-programmed defaults do not latch their values from the corresponding input pins. When the SRST bit is asserted the device's auto-configuration boot controller is **not** started. See section 5.9.

0 = Normal operation
1 = Reset

**Bit 6: Hard Reset (HRST).** Asserting this bit is functionally equivalent to asserting the RSTN pin. The HRST bit resets the entire device except for the microprocessor interface and the HRST bit itself. Register fields with pin-programmed defaults latch their values from the corresponding input pins, and the device's auto-configuration boot controller is started. See section 5.9.

0 = Normal operation
1 = Reset

**Bit 5: Output Clock Stop (STOP).** Asserting this bit stops all output clocks that are configured with OCxSTOP.SRC=0001. Note that this signal is ORed with the per-output stop control bit in the STOPCR registers to make each output's internal stop control signal. See section 5.6.7.

**Bit 3: Ring Oscillator Disable (ROSCD).** This bit disables the ring oscillator. It can be set to 1 when auto-configuration is complete. See section 5.3.4.

0 = Enable
1 = Disable (power-down)

**Bit 1: Open Drain MISO Enable (ODMISO).** This bit configures the MISO pin to be open-drain. When this bit is set, the MISO pin only drives low and must have an external pullup resistor.

0 = Disable (MISO drives 0 and 1, high-impedance when not driven)
1 = Enable (MISO drives 0 only, high-impedance all other times)

**Register Name:** MCR2
**Register Description:** Master Configuration Register 2
**Register Address:** 02h

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **Name** | XAMCK | IC3MCK | IC2MCK | IC1MCK | — | DBL | XAB[1:0] | |
| **Default** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 7: XA Monitor Clock (XAMCK).** This bit specifies which APLL's feedback clock to use for monitoring the XA input clock. The nominal frequency of the specified feedback clock must be the same as XA's nominal frequency for proper monitor operation. See section 5.5.2.

0 = APLL1
1 = APLL2

**Bit 6: IC3 Monitor Clock (IC3MCK).** This bit specifies which APLL's feedback clock to use for monitoring the IC3 input clock. The nominal frequency of the specified feedback clock must be the same as IC3's nominal frequency for proper monitor operation. See section 5.5.2.

0 = APLL1
1 = APLL2

**Bit 5: IC2 Monitor Clock (IC2MCK).** This bit specifies which APLL's feedback clock to use for monitoring the IC2 input clock. The nominal frequency of the specified feedback clock must be the same as IC2's nominal frequency for proper monitor operation. See section 5.5.2.

0 = APLL1
1 = APLL2

**Bit 4: IC1 Monitor Clock (IC1MCK).** This bit specifies which APLL's feedback clock to use for monitoring the IC1 input clock. The nominal frequency of the specified feedback clock must be the same as IC1's nominal frequency for proper monitor operation. See section 5.5.2.

0 = APLL1
1 = APLL2

**Bit 2: Clock Doubler Enable (DBL).** This bit enables the clock doubler for either the output of the crystal driver circuitry or the signal on the XA pin. During power-up, system software must wait at least 5ms for the crystal driver circuit to stabilize before enabling the clock doubler. See section 5.3.3.

0 = Disable (power down)
1 = Enable

**Bits 1 to 0: XA/XB Pin Mode (XAB[1:0]).** This field specifies the behavior of the XA and XB pins. See section 5.3.

00 = Crystal driver and input disabled / powered down
01 = Crystal driver and input enabled on XA/XB
10 = XA enabled as single-ended input for external oscillator signal; XB must be left floating
11 = {unused value}

**Register Name:** PLLEN
**Register Description:** APLL Enable Register
**Register Address:** 03h

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **Name** | — | — | — | — | — | — | APLL2EN | APLL1EN |
| **Default** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 1: APLL2 Enable (APLL2EN).** This bit enables or disables APLL2. See section 5.5.4.
0 = Disabled
1 = Enabled

**Bit 0: APLL1 Enable (APLL1EN).** This bit enables or disables APLL1. See section 5.5.4.
0 = Disabled
1 = Enabled

**Register Name:** ICEN
**Register Description:** Input Clock Enable Register
**Register Address:** 04h

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **Name** | — | — | — | — | — | IC3EN | IC2EN | IC1EN |
| **Default** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 2: Input Clock 3 Enable (IC3EN).** This bit enables and disables the input clock 3 differential receiver and input dividers. See section 5.4.
0 = Disabled
1 = Enabled

**Bit 1: Input Clock 2 Enable (IC2EN).** This bit enables and disables the input clock 2 differential receiver and input dividers. See section 5.4.
0 = Disabled
1 = Enabled

**Bit 0: Input Clock 1 Enable (IC1EN).** This bit enables and disables the input clock 1 differential receiver and input dividers. See section 5.4.
0 = Disabled
1 = Enabled

**Register Name:** OCEN1
**Register Description:** Output Clock Enable Register 1
**Register Address:** 05h

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **Name** | OC8EN | OC7EN | OC6EN | OC5EN | OC4EN | OC3EN | OC2EN | OC1EN |
| **Default** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 7 to 0: Output Clock x Enable (OCxEN).** Each of these bits enables and disables the corresponding output dividers, phase adjustment/alignment circuitry and start/stop circuitry. See section 5.6.1.

0 = Disabled
1 = Enabled

Note: On Rev A devices at least one OCxEN bit must be set for each of the six output banks for proper operation. These bits should be set at or near the beginning of the configuration sequence.

**Register Name:** OCEN2
**Register Description:** Output Clock Enable Register 2
**Register Address:** 06h

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **Name** | — | — | — | — | — | — | OC10EN | OC9EN |
| **Default** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

See the OCEN1 register description above.

**Register Name:** OCMUX1
**Register Description:** Output Clock Mux Register 1
**Register Address:** 07h

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **Name** | - | - | OCMUXC | | OCMUXB | | OCMUXA | |
| **Default** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 5 to 4: Output Clock Mux C (OCMUXC[1:0]).** Controls the high speed output mux for output group C (OC4 and OC5).

00 = APLL1 Integer divider
01 = APLL1 Fractional divider or bypass path (specified by ACR1.BYPASS)
10 = APLL2 Integer divider
11 = APLL2 Fractional divider or bypass path (specified by A2CR1.BYPASS)

**Bits 3 to 2: Output Clock Mux B (OCMUXB[1:0]).** Controls the high speed output mux for output group B (OC3).

00 = APLL1 Integer divider
01 = APLL1 Fractional divider or bypass path (specified by ACR1.BYPASS)
10 = APLL2 Integer divider
11 = APLL2 Fractional divider or bypass path (specified by A2CR1.BYPASS)

**Bits 1 to 0: Output Clock Mux A (OCMUXA[1:0]).** Controls the high speed output mux for output group A (OC1 and OC2).

00 = APLL1 Integer divider
01 = APLL1 Fractional divider or bypass path (specified by ACR1.BYPASS)
10 = APLL2 Integer divider
11 = APLL2 Fractional divider or bypass path (specified by A2CR1.BYPASS)

**Register Name:** OCMUX2
**Register Description:** Output Clock Mux Register 2
**Register Address:** 08h

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **Name** | - | - | OCMUXF | | OCMUXE | | OCMUXD | |
| **Default** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 5 to 4: Output Clock Mux F (OCMUXF[1:0]).** Controls the high speed output mux for output group F (OC9 and OC10).
00 = APLL1 Integer divider
01 = APLL1 Fractional divider or bypass path (specified by ACR1.BYPASS)
10 = APLL2 Integer divider
11 = APLL2 Fractional divider or bypass path (specified by A2CR1.BYPASS)

**Bits 3 to 2: Output Clock Mux E (OCMUXE[1:0]).** Controls the high speed output mux for output group E (OC8).
00 = APLL1 Integer divider
01 = APLL1 Fractional divider or bypass path (specified by ACR1.BYPASS)
10 = APLL2 Integer divider
11 = APLL2 Fractional divider or bypass path (specified by A2CR1.BYPASS)

**Bits 1 to 0: Output Clock Mux D (OCMUXD[1:0]).** Controls the high speed output mux for output group D (OC6 and OC7).
00 = APLL1 Integer divider
01 = APLL1 Fractional divider or bypass path (specified by ACR1.BYPASS)
10 = APLL2 Integer divider
11 = APLL2 Fractional divider or bypass path (specified by A2CR1.BYPASS)

**Register Name:** STOPCR1
**Register Description:** Output Clock Stop Control Register 1
**Register Address:** 09h

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **Name** | OC8STP | OC7STP | OC6STP | OC5STP | OC4STP | OC3STP | OC2STP | OC1STP |
| **Default** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 7: OC8 Stop Control (OC8STP).** When SRC=0001 in the OC8STOP register, setting this bit to 1 causes OC8 to stop. Note that this signal is ORed with MCR1.STOP to make OC8's internal stop control signal. See section 5.6.7.

**Bits 6 to 0:** These bits are similar to OC8STP above but for OC7 through OC1.

**Register Name:** STOPCR2
**Register Description:** Output Clock Stop Control Register 2
**Register Address:** 0Ah

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **Name** | — | — | — | — | — | — | OC10STP | OC9STP |
| **Default** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 1 to 0:** These bits are similar to STOPCR1.OC8STP but for OC10 and OC9.

**Register Name:** GPIOCR1
**Register Description:** GPIO Configuration Register 1
**Register Address:** 0Bh

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| Name | GPIO1C[3:0] | | | | GPIO0C[3:0] | | | |
| Default | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 7 to 4: GPIO1 Configuration (GPIO1C[3:0]).** This field configures the GPIO1 pin as a general-purpose input, a general-purpose output driving low or high, or a status output. The current state of the pin can be read from GPIOSR.GPIO1. When GPIO1 is a status output, the GPIO1SS register specifies which status bit is output.

0000 = General-purpose input
0001 = General-purpose input - inverted polarity
0010 = General-purpose output driving low
0011 = General-purpose output driving high
0100 = Status output – non-inverted polarity
0101 = Status output - inverted polarity of the status bit it follows
0110 = Status output – 0 drives low, 1 high impedance
0111 = Status output – 0 high impedance, 1 drives low
1000 to 1111 = {unused values}

**Bits 3 to 0: GPIO0 Configuration (GPIO0C[3:0]).** This field configures the GPIO0 pin as a general-purpose input, a general-purpose output driving low or high, or a status output. The current state of the pin can be read from GPIOSR.GPIO0. When GPIO0 is a status output, the GPIO0SS register specifies which status bit is output.

0000 = General-purpose input
0001 = General-purpose input - inverted polarity
0010 = General-purpose output driving low
0011 = General-purpose output driving high
0100 = Status output – non-inverted polarity
0101 = Status output - inverted polarity of the status bit it follows
0110 = Status output – 0 drives low, 1 high impedance
0111 = Status output – 0 high impedance, 1 drives low
1000 to 1111 = {unused values}

Note that the bits of the following registers cannot be internally connected to a GPIO configured as a status output: GPIOSR, APLL1IE and APLL2IE.

**Register Name:** GPIOCR2
**Register Description:** GPIO Configuration Register 2
**Register Address:** 0Ch

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| Name | GPIO3C[3:0] | | | | GPIO2C[3:0] | | | |
| Default | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

These fields are identical to those in GPIOCR1 except they control GPIO2 and GPIO3.

**Register Name:** GPIO0SS
**Register Description:** GPIO0 Status Select Register
**Register Address:** 0Dh

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| Name | REG[4:0] | | | | | BIT[2:0] | | |
| Default | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 7 to 3: Status Register (REG[4:0]).** When GPIOCR1.GPIO0C=01xx, this field specifies the register of the status bit that GPIO0 will follow while the BIT field below specifies the status bit within the register. Setting the combination of this field and the BIT field below to point to a bit that isn't implemented as a real-time or latched status register bit results in GPIO0 being driven low. The address of the status bit that GPIO0 follows is 0x40 + REG[4:0]

**Bits 2 to 0: Status Bit (BIT[2:0]).** When GPIOCR1.GPIO0C=01xx, the REG field above specifies the register of the status bit that GPIO0 will follow while this field specifies the status bit within the register. Setting the combination of the REG field and this field to point to a bit that isn't implemented as a real-time or latched status register bit results in GPIO1 being driven low. 000=bit 0 of the register. 111=bit 7 of the register.

Note: The device does not allow the GPIO status register bits in GPIOSR to be followed by a GPIO.

**Register Name:** GPIO1SS
**Register Description:** GPIO1 Status Select Register
**Register Address:** 0Eh

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| Name | REG[4:0] | | | | | BIT[2:0] | | |
| Default | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

These fields are identical to those in GPIO0SS except they control GPIO1.

**Register Name:** GPIO2SS
**Register Description:** GPIO2 Status Select Register
**Register Address:** 0Fh

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| Name | REG[4:0] | | | | | BIT[2:0] | | |
| Default | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

These fields are identical to those in GPIO0SS except they control GPIO2.

**Register Name:** GPIO3SS
**Register Description:** GPIO3 Status Select Register
**Register Address:** 10h

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| Name | REG[4:0] | | | | | BIT[2:0] | | |
| Default | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

These fields are identical to those in GPIO0SS except they control GPIO3.

**Register Name:** I2CA
**Register Description:** I2C Address register 1
**Register Address:** 11h

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **Name** | 0 | I2CA[6:0] | | | | | | |
| **Default** | 0 | 1 | 1 | 1 | 0 | 1 | See below | |

**Bits 6 to 0: I2C Address (I2CA[6:0]).** This field specifies the device's address on the $I^2C$ bus. At the assertion of the RSTN pin, bits 6:2 are set to the default values shown above, and bits 1:0 are set to the states of the IF1 and IF0 pins. The MCR1.HRST and MCR1.SRST bits have no effect on these bits. After reset these bits can be written by system software to change the device's $I^2C$ address as needed. Note: the value I2CA=0 is invalid.

### 6.3.2 Status Registers

**Register Name:** ID1
**Register Description:** Device Identification Register, MSB
**Register Address:** 30h

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **Name** | IDU[7:0] | | | | | | | |
| **Default** | 0 | 0 | 0 | 1 | 1 | 1 | see below | |

**Bits 7 to 0: Device ID Upper (IDU[7:0]).** This field is the upper eight bits of the device ID.

**Register Name:** ID2
**Register Description:** Device Identification Register, LSB and Revision
**Register Address:** 31h

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **Name** | IDL[3:0] | | | | REV[3:0] | | | |
| **Default** | see below | | | | contact factory | | | |

**Bits 7 to 4: Device ID Lower (IDL[3:0]).** This field is the lower four bits of the device ID.
ZL30264 = 0x1D8
ZL30265 = 0x1F8
ZL30266 = 0x1D9
ZL30267 = 0x1F9

**Bits 3 to 0: Device Revision (REV[3:0]).** These bits are the device hardware revision starting at 0.

**Register Name:** CFGSR
**Register Description:** Configuration Status Register
**Register Address:** 40h

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **Name** | CFGD | — | IF[1:0] | | TEST | AC[2:0] | | |
| **Default** | 0 | 0 | see below | | see below | see below | | |

**Bit 7: Configured (CFGD).** This read-only bit is cleared by assertion of RSTN, MCR1.HRST or MCR1.SRST and set when any register is written (by auto-configuration or through the processor interface). CFGD=1 indicates that the device register set is no longer in factory-default state, and, therefore, the device must be reset before a GUI-generated configuration script is executed.

**Bits 5 to 4: Interface Mode (IF[1:0]).** These read-only bits are the latched state of the IF1/MISO and IF0/CSN pins when the RSTN pin transitions high. See section 5.2.

**Bit 3: Test Mode (TEST).** This read-only bit is the latched state of the TEST/GPIO3 pin when the RSTN pin transitions high or the MCR1.HRST bit is deasserted. For proper operation it should be 0. See section 5.2.

**Bits 2 to 0: Auto-Configuration (AC[2:0]).** These bits are the latched state of the AC2/GPIO2, AC1/GPIO1 and AC0/GPIO0 pins when the RSTN pin transitions high or the MCR1.HRST bit is deasserted. See section 5.2.

**Register Name:** GPIOSR
**Register Description:** GPIO Status Register
**Register Address:** 41h

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **Name** | GPIO3L | GPIO2L | GPIO1L | GPIO0L | GPIO3 | GPIO2 | GPIO1 | GPIO0 |
| **Default** | 0 | 0 | 0 | 0 | pin state | pin state | pin state | pin state |

**Bit 7: GPIO3 Change Latched Status (GPIO3L).** This latched status bit is set to 1 when the GPIO3 status bit changes state, low-to-high or high-to-low. GPIO3L is cleared when written with a 1. When GPIO3L is set it can cause an interrupt request if the GPIO3IE interrupt enable bit is set.

**Bit 6: GPIO2 Change Latched Status (GPIO2L).** This latched status bit is set to 1 when the GPIO2 status bit changes state, low-to-high or high-to-low. GPIO2L is cleared when written with a 1. When GPIO2L is set it can cause an interrupt request if the GPIO2IE interrupt enable bit is set.

**Bit 5: GPIO1 Change Latched Status (GPIO1L).** This latched status bit is set to 1 when the GPIO1 status bit changes state, low-to-high or high-to-low. GPIO1L is cleared when written with a 1. When GPIO1L is set it can cause an interrupt request if the GPIO1IE interrupt enable bit is set.

**Bit 4: GPIO0 Change Latched Status (GPIO0L).** This latched status bit is set to 1 when the GPIO0 status bit changes state, low-to-high or high-to-low. GPIO0L is cleared when written with a 1. When GPIO0L is set it can cause an interrupt request if the GPIO0IE interrupt enable bit is set.

**Bit 3: GPIO3 State (GPIO3).** This real-time status bit indicates the current state of the GPIO3 pin, not influenced by any inversion that may be specified by GPIOCR2.GPIO3C.

0 = low
1 = high

**Bit 2: GPIO2 State (GPIO2).** This real-time status bit indicates the current state of the GPIO2 pin, not influenced by inversion that may be specified by GPIOCR2.GPIO2C.

0 = low
1 = high

**Bit 1: GPIO1 State (GPIO1).** This real-time status bit indicates the current state of the GPIO1 pin, not influenced by inversion that may be specified by GPIOCR1.GPIO1C.

0 = low
1 = high

**Bit 0: GPIO0 State (GPIO0).** This real-time status bit indicates the current state of the GPIO0 pin, not influenced by inversion that may be specified by GPIOCR1.GPIO0C.

0 = low
1 = high

**Register Name:** INTSR
**Register Description:** Interrupt Status Register
**Register Address:** 42h

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **Name** | — | — | OC | IC | — | APLL | INTIE | INT |
| **Default** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 5: Output Clock Interrupt Status (OC).** This read-only bit is set if any of the output clock interrupt status bits are set in the OCISR1 register. See section 5.8.

**Bit 4: Input Clock Interrupt Status (IC).** This read-only bit is set if any of the input clock interrupt status bits are set in the ICISR register. See section 5.8.

**Bit 2: APLL Interrupt Status (APLL).** This read-only bit is set if any of the APLL interrupt status bits are set in the APLLISR register. See section 5.8.

**Bit 1: Interrupt Enable Bit (INTIE).** This is the global interrupt enable bit. When this bit is 0 all interrupt sources are prevented from setting the INT global interrupt status bit (below). See section 5.8.

0 = Interrupts are disabled at the global level
1 = Interrupts are enabled at the global level

**Bit 0: Interrupt Status (INT).** This read-only bit is set when any of the IC, OC or APLL bits in this INTSR register are set and the INTIE bit is set. It is also set by GPIO latched status bits that have their corresponding interrupt enable bits set. This bit can cause an interrupt request when set by configuring one of the GPIO pins to follow it. See section 5.8.

0 = No interrupt
1 = An unmasked interrupt source is active

**Register Name:** GLOBISR
**Register Description:** Global Functions Interrupt Status Register
**Register Address:** 43h

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **Name** | BCDONE | — | — | — | GPIO3IE | GPIO2IE | GPIO1IE | GPIO0IE |
| **Default** | see below | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 7: Boot Controller Done (BCDONE).** This bit indicates the status of the on-chip boot controller, which performs auto-configuration from ROM or EEPROM. It is cleared when the device is reset and set after the boot controller finishes auto-configuration of the device. See section 5.11.

Note that BCDONE cannot be polled while the device is auto-configuring because the internal register bus is in use. The BCDONE bit was designed to be followed by a GPIO pin configured as a status output. To cause GPIO0, for example, to follow BCDONE, include the following settings at the beginning of the auto-configuration script: GPIOCR1=0x04 (configures GPIO0 as a non-inverted status output) and GPIO0SS=00011 111b (causes GPIO0 to follow the bit at register 0x43, bit 7, which is BCDONE).

Alternately, there is a way to poll the device to determine whether auto-configuration is complete. This involves choosing a writeable bit that (a) has a harmless effect, such as GLOBISR.GPIO3IE, and (b) is not set during auto-configuration. System software can then poll the device by writing the register to set the bit then reading the register to see if the bit is set. The bit cannot be set by system software while the device is auto-configuring. Therefore when it is found to be set auto-configuration must be complete.

**Bit 3: GPIO3 Change Interrupt Enable (GPIO3IE).** This bit enables the GPIOSR.GPIO3L latched status bit to send an interrupt request into the device's interrupt logic.

0 = Interrupt is disabled
1 = Interrupt is enabled

**Bit 2: GPIO2 Change Interrupt Enable (GPIO2IE).** This bit enables the GPIOSR.GPIO2L latched status bit to send an interrupt request into the device's interrupt logic.

0 = Interrupt is disabled
1 = Interrupt is enabled

**Bit 1: GPIO1 Change Interrupt Enable (GPIO1IE).** This bit enables the GPIOSR.GPIO1L latched status bit to send an interrupt request into the device's interrupt logic.

0 = Interrupt is disabled
1 = Interrupt is enabled

**Bit 0: GPIO0 Change Interrupt Enable (GPIO0IE).** This bit enables the GPIOSR.GPIO0L latched status bit to send an interrupt request into the device's interrupt logic.

0 = Interrupt is disabled
1 = Interrupt is enabled

**Register Name:** ICISR
**Register Description:** Input Clock Interrupt Status Register
**Register Address:** 44h

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **Name** | — | — | — | — | XA | IC3 | IC2 | IC1 |
| **Default** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 3: XA Input Interrupt Status (XA).** This bit indicates the current status of the interrupt sources for XA. This bit is set when latched status XASR.XAL is set and the associated interrupt enable bit is also set. See section 5.8.

**Bit 2 to 0: Input Clock x Interrupt Status (IC[3:1]).** Each bit indicates the current status of the interrupt sources for the corresponding input. It is set when latched status ICxSR.ICVL is set and the associated interrupt enable bit is also set. See section 5.8.

**Register Name:** OCISR1
**Register Description:** Output Clock Interrupt Status Register 1
**Register Address:** 45h

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **Name** | OC8 | OC7 | OC6 | OC5 | OC4 | OC3 | OC2 | OC1 |
| **Default** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 7 to 0: Output Clock x Interrupt Status (OC[8:1]).** Each bit indicates the current status of the interrupt sources for the corresponding output. It is set when any latched status bit in the OCxSR register is set and the associated interrupt enable bit is also set. See section 5.8.

**Register Name:** OCISR2
**Register Description:** Output Clock Interrupt Status Register 2
**Register Address:** 46h

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **Name** | — | — | — | — | — | — | OC10 | OC9 |
| **Default** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

See the OCISR1 register description above.

**Register Name:** APLLISR
**Register Description:** APLL Interrupt Status Register
**Register Address:** 47h

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **Name** | — | — | — | — | — | — | APLL2 | APLL1 |
| **Default** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 1: APLL2 Interrupt Status (APLL2).** This bit indicates the current status of the interrupt sources for APLL2. It is set when any latched status bit in the APLL2SR register is set and the associated interrupt enable bit is also set. See section 5.8.

**Bit 0: APLL1 Interrupt Status (APLL1).** This bit indicates the current status of the interrupt sources for APLL1. It is set when any latched status bit in the APLL1SR register is set and the associated interrupt enable bit is also set. See section 5.8.

**Register Name:** APLL1SR
**Register Description:** APLL1 Status Register
**Register Address:** 48h

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **Name** | AIFLL | AIFL | AIFHL | AIFH | ALKL | ALK | — | SELREF |
| **Default** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 7: APLL Input Frequency Low Latched Status (AIFLL).** This latched status bit is set to 1 when the AIFL status bit is set. AIFLL is cleared when written with a 1. When AIFLL is set it can cause an interrupt request if the AIFLIE interrupt enable bit is set.

**Bit 6: APLL Input Frequency Low Status (AIFL).** This real-time status bit indicates that the input frequency to the APLL is lower that expected.
0 = Input frequency ok
1 = Input frequency low

**Bit 5: APLL Input Frequency High Latched Status (AIFHL).** This latched status bit is set to 1 when the AIFH status bit is set. AIFHL is cleared when written with a 1. When AIFHL is set it can cause an interrupt request if the AIFHIE interrupt enable bit is set.

**Bit 4: APLL Input Frequency High Status (AIFH).** This real-time status bit indicates that the input frequency to the APLL is higher that expected.
0 = Input frequency ok
1 = Input frequency high

**Bit 3: APLL Lock Latched Status (ALKL).** This latched status bit is set to 1 when the ALK status bit changes state (set or cleared). ALKL is cleared when written with a 1. When ALKL is set it can cause an interrupt request if the ALKIE interrupt enable bit is set.

**Bit 2: APLL Lock Status (ALK).** This real-time status bit indicates the lock status of the APLL. See section 5.5.
0 = Not locked
1 = Locked

**Bit 0: Selected Reference (SELREF).** This real-time status field indicates the APLL's selected reference. See section 5.5.3.
0 = The input specified by ACR3.APLLMUX
1 = The input specified by ACR3.ALTMUX

**Register Name:** APLL2SR
**Register Description:** APLL2 Status Register
**Register Address:** 49h

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **Name** | AIFLL | AIFL | AIFHL | AIFH | ALKL | ALK | — | SELREF |
| **Default** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

The fields in the register are the same as the fields in the APLL1SR register except they apply to APLL2.

**Register Name:** APLL1IE
**Register Description:** APLL1 Interrupt Enable Register
**Register Address:** 4Ah

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **Name** | AIFLIE | — | AIFHIE | — | ALKIE | — | — | — |
| **Default** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 7: APLL Input Frequency Low Interrupt Enable (AIFLIE).** This bit enables the AIFLL latched status bit to send an interrupt request into the device's interrupt logic.

0 = Interrupt is disabled
1 = Interrupt is enabled

**Bit 5: APLL Input Frequency High Interrupt Enable (AIFHIE).** This bit enables the AIFHL latched status bit to send an interrupt request into the device's interrupt logic.

0 = Interrupt is disabled
1 = Interrupt is enabled

**Bit 3: APLL Lock Interrupt Enable (ALKIE).** This bit enables the ALKL latched status bit to send an interrupt request into the device's interrupt logic.

0 = Interrupt is disabled
1 = Interrupt is enabled

**Register Name:** APLL2IE
**Register Description:** APLL2 Interrupt Enable Register
**Register Address:** 4Bh

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **Name** | AIFLIE | — | AIFHIE | — | ALKIE | — | — | — |
| **Default** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

The fields in the register are the same as the fields in the APLL1IE register except they apply to APLL2.

**Register Name:** XASR
**Register Description:** XA Input Status Register
**Register Address:** 4Ch

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **Name** | — | — | — | — | XAVIE | XAVL | XAV | — |
| **Default** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

The fields in the register are the same as the fields in the ICxSR registers except they apply to the XA input.

**Register Name:** ICxSR
**Register Description:** Input Clock Status Register
**Register Address:** IC1: 4Dh, IC2: 4Eh, IC3: 4Fh

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **Name** | — | — | — | — | ICVIE | ICVL | ICV | — |
| **Default** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 3: Input Clock Valid Interrupt Enable (ICVIE).** This bit enables the ICxSR.ICVL latched status bit to send an interrupt request into device's interrupt logic.

0 = Interrupt is disabled
1 = Interrupt is enabled

**Bit 2: Input Clock Valid Latched (ICVL).** This latched status bit is set to 1 when the ICxSR.ICV status bit changes state (set or cleared). This bit is cleared when written with a 1 and not set again until the ICxSR.ICV bit changes state again. This bit can be the source of an interrupt request. See section 5.5.2.

0 = ICxSR.ICV bit has not changed state since last cleared
1 = ICxSR.ICV bit has changed state since last cleared

**Bit 1: Input Clock Valid (ICV).** This real-time status bit is high when the input clock monitor indicates the input is valid. See section 5.5.2.

0 = The input clock is not valid
1 = The input clock is valid

**Register Name:** OCxSR
**Register Description:** Output Clock x Status Register
**Register Address:** OC1: 50h, OC2: 51h, OC3: 52h, OC4: 53h, OC5: 54h
OC6: 55h, OC7: 56h, OC8: 57h, OC9: 58h, OC10: 59h

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **Name** | LSCLKIE | LSCLKL | LSCLK | STARTIE | STARTL | STOPIE | STOPL | STOPD |
| **Default** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 7: (LSCLKIE).** This bit enables the LSCLKL latched status bit to send an interrupt request into device's interrupt logic.
0 = Interrupt is disabled
1 = Interrupt is enabled

**Bit 6: (LSCLKL).** This latched status bit is set when the low-speed divider output clock transitions low-to-high. Writing a 1 to this bit clears it.
0 = Low speed output clock has not transitioned low to high
1 = Low speed output clock has transitioned low to high

**Bit 5: (LSCLK).** This real-time status bit follows the level of the low-speed divider output clock when the OCxCR3.SRLSEN bit is set.
0 = LSCLK is high
1 = LSCLK is low

**Bit 4: (STARTIE).** This bit enables the STARTL latched status bit to send an interrupt request into device's interrupt logic.
0 = Interrupt is disabled
1 = Interrupt is enabled

**Bit 3: (STARTL).** This latched status bit is set when the output clock signal has been started after being stopped. Writing a 1 to this bit clears it. See section 5.6.6.
0 = Output clock signal has not resumed from being stopped
1 = Output clock signal has resumed from being stopped

**Bit 2: (STOPIE).** This bit enables the STOPL latched status bit to send an interrupt request into device's interrupt logic.
0 = Interrupt is disabled
1 = Interrupt is enabled

**Bit 1: (STOPL).** This latched status bit is set when the output clock signal has been stopped. Writing a 1 to this bit clears it. See section 5.6.6.
0 = Output clock signal has not stopped
1 = Output clock signal has stopped

**Bit 0: (STOPD).** This real-time status bit is high when the output clock signal is stopped and low when the output clock is not stopped. See section 5.6.6.
0 = Output clock signal is not stopped
1 = Output clock signal is stopped

### 6.3.3 APLL Configuration Registers

**Register Name:** ACR1, A2CR1
**Register Description:** APLL Configuration Register 1
**Register Address:** APLL1: 100h, APLL2: 180h

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **Name** | ENFID | DALIGN | EXTSS[1:0] | | USEFDIV | ENFDIV | BYPASS | INDBL |
| **Default** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 7: Enable Frequency Increment/Decrement (ENFID).** This bit enables frequency increment/decrement behavior as described in section 5.5.7.

**Bit 6: Align Output Dividers (DALIGN).** A 0-to-1 transition on this bit causes a simultaneous reset of the medium-speed dividers and the low-speed dividers for all output clocks where OCxCR1.PHEN=1. After this reset all PHEN=1 output clocks with frequencies that are exactly integer multiples of one another will be rising-edge aligned as specified by their OCxPH registers. This bit should be set then cleared once during system startup. Setting this bit during normal system operation can cause phase jumps in the output clock signals.

**Bits 5 to 4: External Switch Source Select (EXTSS[1:0]).** This field selects the GPIO source for the external switch control signal. It is only valid when ACR3.EXTSW=1. See section 5.5.1.

00 = GPIO0
01 = GPIO1
10 = GPIO2
11 = GPIO3

**Bit 3: Use Fractional Divider (USEFDIV).** This bit controls which resource the spread spectrum logic controls. See section 5.5.8.2.

0 = APLL feedback divider
1 = fractional output divider

**Bit 2: Enable Fractional Divider (ENFDIV).** This bit is an enable/disable control for the APLL's fractional divider. When the fractional divider is disabled, device power consumption is reduced as shown in Table 6. The fractional divider is enabled when PLLEN.APLLEN=1, ENFDIV=1 and ACR1.BYPASS=0.

0 = Disable
1 = Enable

**Bit 1: Bypass APLL and Fractional Divider (BYPASS).** This bit controls selection between the APLL's fractional divider and the APLL's bypass path as shown in Figure 1. The mux that provides the bypass signal to this mux is controlled by APLLMUX and related fields in ACR3.

0 = No bypass
1 = Bypass

**Bit 0: APLL Input Doubler Enable (INDBL).** This bit enables a simple clock doubler at the input to the APLL. This feature allows signals from IC1, IC2 or IC3 to be doubled which often can result in lower output jitter. See section 5.3.3.

**Register Name:** ACR2, A2CR2
**Register Description:** APLL Configuration Register 2
**Register Address:** APLL1: 101h, APLL2: 181h

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **Name** | — | — | — | — | INTDIV[3:0] | | | |
| **Default** | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |

**Bits 3 to 0: APLL Integer Divider (INTDIV[3:0]).** This field controls the APLL's integer divider (see Figure 5). See section 5.5.4.

0000 = Divide by 4
0001 = Divide by 4.5
0010 = Divide by 5
0011 = Divide by 5.5
0100 = Divide by 6 (default)
0101 = Divide by 6.5
0110 = Divide by 7
0111 = Divide by 7.5
1000 = Divide by 8
1001 = Divide by 9
1010 = Divide by 10
1011 = Divide by 11
1100 = Divide by 12
1101 = Divide by 13
1110 = Divide by 14
1111 = Divide by 15

**Register Name:** ACR3, A2CR3
**Register Description:** APLL Configuration Register 3
**Register Address:** APLL1: 102h, APLL2: 182h

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **Name** | INMON | EXTSW | ALTMUX[2:0] | | | APLLMUX[2:0] | | |
| **Default** | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |

**Bit 7: APLL Input Monitor Switching Mode (INMON).** This bit enables the APLL input monitor loss of signal reference switching mode. In this mode, if the input specified by the APLLMUX field satisfies the loss-of-signal condition, the APLL input mux will switch to the input specified by the ALTMUX field. See section 5.5.3.

**Bit 6: APLL External Switching Mode (EXTSW).** This bit enables APLL external reference switching mode. In this mode, if the selected GPIO signal is low the APLL input mux is controlled by ACR3.APLLMUX. If the selected GPIO signal is high the APLL input mux is controlled by ACR3.ALTMUX. ACR1.EXTSS specifies which GPIO pin controls this behavior. See section 5.5.3.

**Bits 5 to 3: APLL Alternate Mux Control (ALTMUX[2:0]).** This field specifies the alternate APLL clock source for external switching (when EXTSW=1) and for automatic switching based on monitor status (when EXTSW=0 an INMON=1). See section 5.5.3.

000 = IC1 input (default)
001 = IC2 input
010 = IC3 input
011 = XA not doubled
100 = XA doubled (must have MCR2.DBL=1)
101-111 = {reserved values}

**Bits 2 to 0: APLL Mux Control (APLLMUX[2:0]).** By default this field controls the APLL input mux. It also specifies the primary APLL clock source for external switching (when EXTSW=1) and for automatic switching based on monitor status (when EXTSW=0 and INMON=1). See section 5.5.3.

000 = IC1 input
001 = IC2 input
010 = IC3 input
011 = XA not doubled (default)
100 = XA doubled (must have MCR2.DBL=1)
101-111 = {reserved values}

**Register Name:** ACR4, A2CR4
**Register Description:** APLL Configuration Register 4
**Register Address:** APLL1: 103h, APLL2: 183h

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **Name** | DECPH | PDSS[2:0] | | | INCPH | PISS[2:0] | | |
| **Default** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 7: Decrement Phase (DECPH).** When PDSS=000, this bit is the APLL phase decrement control signal. See section 5.5.9.

**Bits 6 to 4: Phase Decrement Source Select (PDSS[2:0]).** This field specifies the APLL phase decrement control signal. Every low-to-high transition and every high-to-low transition of the signal decrements the APLL's output phase. See section 5.5.9.

000 = DECPH bit
001 = GPIO0
010 = GPIO1
011 = GPIO2
100 = GPIO3
101 to 111 = {unused values}

**Bit 3: Increment Phase (INCPH).** When PISS=000, this bit is the APLL phase increment control signal. See section 5.5.9.

**Bits 2 to 0: Phase Increment Source Select (PISS[2:0]).** This field specifies the APLL phase increment control signal. Every low-to-high transition and every high-to-low transition of the signal increments the APLL's output phase. See section 5.5.9.

000 = INCPH bit
001 = GPIO0
010 = GPIO1
011 = GPIO2
100 = GPIO3
101 to 111 = {unused values}

**Register Name:** AFBDL, A2FBDL
**Register Description:** APLL Feedback Divider Write Length
**Register Address:** APLL1: 105h, APLL2: 185h

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **Name** | EXTFB | FBSEL[1:0] | | RDCUR | — | AFBDL[2:0] | | |
| **Default** | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |

**Bit 7: External Feedback Enable (EXTFB).**

0 = Internal feedback through the APLL's fractional feedback divider
1 = External feedback (typically for output vs. input phase alignment)

**Bits 6 to 5: External Feedback Select (FBSEL[1:0]).** When EXTFB=1, this field specifies the external feedback path.

00 = from any OCx output through external path to IC1 input to APLL's feedback input
01 = from any OCx output through external path to IC2 input to APLL's feedback input
10 = from any OCx output through external path to IC3 input to APLL's feedback input
11 = APLL1: from OC1 output through internal (on-die) path to APLL's feedback input
APLL2: from OC3 output through internal (on-die) path to APLL's feedback input

**Bit 4: Read Current AFBDIV Value (RDCUR).** See section 5.5.7. The system must not cause any frequency increments or decrements during the read of the current APLL feedback divider value; otherwise the bytes of the value read may not be coherent.

0 = Read the original APLL feedback divider value written to the AFBDIV registers by system software
1 = Read the current APLL feedback divider value after increments and decrements

**Bits 2 to 0: APLL Feedback Divider Write Length (AFBDL[2:0]).** This field indicates the last register to write of the AFBDIV multiregister field where "last register" is as described in section 6.1.3. The number of the last register to write is AFBDL+1. The default of 5 specifies AFBDIV6 as described in section 6.1.3. Changing this field from its default value can be useful in NCO mode to reduce SPI or I2C bus usage when NCO changes only affect the least significant bytes of AFBDIV.

**Register Name:** AFBDIV1, A2FBDIV1
**Register Description:** APLL Feedback Divider Register 1
**Register Address:** APLL1: 106h, APLL2: 186h

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **Name** | AFBDIV[7:0] | | | | | | | |
| **Default** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 7 to 0: APLL Feedback Divider Register (AFBDIV[7:0]).** The full 42-bit AFBDIV[41:0] field spans the AFBDIV1 through AFBDIV6 registers. AFBDIV is an unsigned number with 9 integer bits (AFBDIV[41:33]) and up to 33 fractional bits. AFBDIV specifies the fixed-point term of the APLL's fractional feedback divide value. The value AFBDIV=0 is undefined. Unused least significant bits must be written with 0. See section 5.5.4.

**Register Name:** AFBDIV2, A2FBDIV2
**Register Description:** APLL Feedback Divider Register 2
**Register Address:** APLL1: 107h, APLL2: 187h

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **Name** | AFBDIV[15:8] | | | | | | | |
| **Default** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 7 to 0: APLL Feedback Divider Register (AFBDIV[15:8]).** See the AFBDIV1 register description.

**Register Name:** AFBDIV3, A2FBDIV3
**Register Description:** APLL Feedback Divider Register 3
**Register Address:** APLL1: 108h, APLL2: 188h

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **Name** | AFBDIV[23:16] | | | | | | | |
| **Default** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 7 to 0: APLL Feedback Divider Register (AFBDIV[23:16]).** See the AFBDIV1 register description.

**Register Name:** AFBDIV4, A2FBDIV4
**Register Description:** APLL Feedback Divider Register 4
**Register Address:** APLL1: 109h, APLL2: 189h

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **Name** | AFBDIV[31:24] | | | | | | | |
| **Default** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 7 to 0: APLL Feedback Divider Register (AFBDIV[31:24]).** See the AFBDIV1 register description.

**Register Name:** AFBDIV5, A2FBDIV5
**Register Description:** APLL Feedback Divider Register 5
**Register Address:** APLL1: 10Ah, APLL2: 18Ah

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **Name** | AFBDIV[39:32] | | | | | | | |
| **Default** | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 |

**Bits 7 to 0: APLL Feedback Divider Register (AFBDIV[39:32]).** See the AFBDIV1 register description.

**Register Name:** AFBDIV6, A2FBDIV6
**Register Description:** APLL Feedback Divider Register 6
**Register Address:** APLL1: 10Bh, APLL2: 18Bh

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **Name** | — | — | — | — | — | — | AFBDIV[41:40] | |
| **Default** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 1 to 0: APLL Feedback Divider Register (AFBDIV[41:40]).** See the AFBDIV1 register description.

**Register Name:** AFBDEN1, A2FBDEN1
**Register Description:** APLL Feedback Divider Denominator Register 1
**Register Address:** APLL1: 10Ch, APLL2: 18Ch

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **Name** | AFBDEN[7:0] | | | | | | | |
| **Default** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

**Bits 7 to 0: APLL Feedback Divider Denominator Register (AFBDEN[7:0]).** The full 32-bit AFBDEN[31:0] field spans AFBDEN1 through AFBDEN4 registers. AFBDEN is an unsigned integer that specifies the denominator of the APLL's fractional feedback divide value. The value AFBDEN=0 is undefined. When AFBBP=0, AFBDEN must be set to 1. See section 5.5.4.

**Register Name:** AFBDEN2, A2FBDEN2
**Register Description:** APLL Feedback Divider Denominator Register 2
**Register Address:** APLL1: 10Dh, APLL2: 18Dh

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **Name** | AFBDEN[15:8] | | | | | | | |
| **Default** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 7 to 0: APLL Feedback Divider Denominator Register (AFBDEN[15:8]).** See the AFBDEN1 register description.

**Register Name:** AFBDEN3, A2FBDEN3
**Register Description:** APLL Feedback Divider Denominator Register 3
**Register Address:** APLL1: 10Eh, APLL2: 18Eh

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **Name** | AFBDEN[23:16] | | | | | | | |
| **Default** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 7 to 0: APLL Feedback Divider Denominator Register (AFBDEN[23:16]).** See the AFBDEN1 register description.

**Register Name:** AFBDEN4, A2FBDEN4
**Register Description:** APLL Feedback Divider Denominator Register 4
**Register Address:** APLL1: 10Fh, APLL2: 18Fh

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **Name** | AFBDEN[31:24] | | | | | | | |
| **Default** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 7 to 0: APLL Feedback Divider Denominator Register (AFBDEN[31:24]).** See the AFBDEN1 register description.

**Register Name:** AFBREM1, A2FBREM1
**Register Description:** APLL Feedback Divider Remainder Register 1
**Register Address:** APLL1: 110h, APLL2: 190h

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **Name** | AFBREM[7:0] | | | | | | | |
| **Default** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 7 to 0: APLL Feedback Divider Remainder Register (AFBREM[7:0]).** The full 32-bit AFBDEN[31:0] field spans AFBREM1 through AFBREM4 registers. AFBREM is an unsigned integer that specifies the remainder of the APLL's fractional feedback divider value. See section 5.5.4.

**Register Name:** AFBREM2, A2FBREM2
**Register Description:** APLL Feedback Divider Remainder Register 2
**Register Address:** APLL1: 111h, APLL2: 191h

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **Name** | AFBREM[15:8] | | | | | | | |
| **Default** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 7 to 0: APLL Feedback Divider Remainder Register (AFBREM[15:8]).** See the AFBREM1 register description.

**Register Name:** AFBREM3, A2FBREM3
**Register Description:** APLL Feedback Divider Remainder Register 3
**Register Address:** APLL1: 112h, APLL2: 192h

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **Name** | AFBREM[23:16] | | | | | | | |
| **Default** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 7 to 0: APLL Feedback Divider Remainder Register (AFBREM[23:16]).** See the AFBREM1 register description.

**Register Name:** AFBREM4, A2FBREM4
**Register Description:** APLL Feedback Divider Remainder Register 4
**Register Address:** APLL1: 113h, APLL2: 193h

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **Name** | AFBREM[31:24] | | | | | | | |
| **Default** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 7 to 0: APLL Feedback Divider Remainder Register (AFBREM[31:24]).** See the AFBREM1 register description.

**Register Name:** AFBBP, A2FBBP
**Register Description:** APLL Feedback Divider Truncate Bit Position
**Register Address:** APLL1: 114h, APLL2: 194h

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **Name** | AFBBP[7:0] | | | | | | | |
| **Default** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 7 to 0: APLL Feedback Divider Truncate Bit Position (AFBBP[7:0]).** This unsigned integer specifies the number of fractional bits that are valid in the AFBDIV value. There are 33 fractional bits in AFBDIV. The value in this AFBBP field specifies 33 – number_of_valid_AFBDIV_fractional_bits. When AFBBP=0 all 33 AFBDIV fractional bits are valid. When AFBBP=9, the most significant 24 AFBDIV fractional bits are valid and the least significant 9 bits must be set to 0. This register field is only used when the feedback divider value is expressed in the form AFBDIV + AFBREM / AFBDEN. AFBBP values greater than 33 are invalid. See section 5.5.4.

**Register Name:** AIDCR, A2IDCR
**Register Description:** APLL Frequency Increment-Decrement Configuration Register
**Register Address:** APLL1: 115h, APLL2: 195h

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **Name** | DECF | FDSS[2:0] | | | INCF | FISS[2:0] | | |
| **Default** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 7: Decrement Frequency (DECF).** When FDSS=000, this bit is the APLL frequency decrement control signal. See section 5.5.7.

**Bits 6 to 4: Frequency Decrement Source Select (FDSS[2:0]).** This field specifies the APLL frequency decrement control signal. Every low-to-high transition and every high-to-low transition of the signal decrements the APLL's output frequency by the amount specified by AID[23:0] x $2^7$. See section 5.5.7.

000 = DECF bit
001 = GPIO0
010 = GPIO1
011 = GPIO2
100 = GPIO3
101 to 111 = {unused values}

**Bit 3: Increment Frequency (INCF).** When FISS=000, this bit is the APLL frequency increment control signal. See section 5.5.7.

**Bits 2 to 0: Frequency Increment Source Select (FISS[2:0]).** This field specifies the APLL frequency increment control signal. Every low-to-high transition and every high-to-low transition of the signal increments the APLL's output frequency by the amount specified by AID[23:0] x $2^7$. See section 5.5.7.

000 = INCF bit
001 = GPIO0
010 = GPIO1
011 = GPIO2
100 = GPIO3
101 to 111 = {unused values}

**Register Name:** ASCR, A2SCR
**Register Description:** APLL Spread Spectrum Configuration Register 1
**Register Address:** APLL1: 116h, APLL2: 196h

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **Name** | — | CNTEN[1:0] | | DWNEN | SSEN | SPRDSS[2:0] | | |
| **Default** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 6 to 5: Count Enable (CNTEN[1:0]).** This field can be used to specify a count, or divide, value to divide the APLL feedback divider clock down as needed for clocking the spread-spectrum logic when doing spread-spectrum modulation in the fractional output divider. See section 5.5.8.2.

**Bit 4: Spread Spectrum Down Only (DWNEN).** Spread spectrum frequency modulation can be either center-spread or down-spread. See section 5.5.8.

0 = Center-spread
1 = Down-spread

**Bit 3: Spread Spectrum Enable (SSEN).** When SPRDSS (see below) is set to 000, this bit enables spread spectrum modulation in the APLL. See section 5.5.8.

**Bits 2 to 0: Spread Spectrum Enable Source Select (SPRDSS[2:0]).** This field specifies the APLL spread spectrum enable control signal. See section 5.5.8.

000 = SSEN bit
001 = GPIO0
010 = GPIO1
011 = GPIO2
100 = GPIO3
101 to 111 = {unused values}

**Register Name:** ASCNT1, A2SCNT1
**Register Description:** APLL Spread Spectrum Count Register 1
**Register Address:** APLL1: 117h, APLL2: 197h

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **Name** | ASCNT[7:0] | | | | | | | |
| **Default** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 7 to 0: APLL Spread-Spectrum Count (ASCNT[7:0]).** The full 16-bit ASCNT[15:0] register spans this register and ASCNT2. ASCNT is an unsigned integer that specifies the number of cycles of the APLL's input clock that the device increments the output frequency and then the number of cycles that the device decrements the output frequency. See section 5.5.8.

**Register Name:** ASCNT2, A2SCNT2
**Register Description:** APLL Spread Spectrum Count Register 1
**Register Address:** APLL1: 118h, APLL2: 198h

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **Name** | ASCNT[15:8] | | | | | | | |
| **Default** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 7 to 0: APLL Spread-Spectrum Count (ASCNT[15:8]).** See the ASCNT1 register description.

**Register Name:** AID1, A2ID1
**Register Description:** APLL Increment/Decrement Register 1
**Register Address:** APLL1: 119h, APLL2: 199h

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **Name** | AID[7:0] | | | | | | | |
| **Default** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 7 to 0: APLL Increment/Decrement (AID[7:0]).** The full 32-bit AID[31:0] register spans this register through AID4. It is an unsigned integer. There are two uses for this field: frequency increment/decrement, and spread spectrum modulation. The two uses are mutually-exclusive, i.e. only one can be in use at a time.

For frequency increment/decrement, AID[23:0] x $2^7$ is added to the APLL's feedback divider value each time the trigger specified by AIDCR.FISS changes state. AID[23:0] x $2^7$ is subtracted from the APLL's feedback divider value each time the trigger specified by AIDCR.FDSS changes state. AID[31:24] is ignored for frequency increment/decrement. ACR1.USEFDIV must be set to 0 for this behavior. See section 5.5.7.

For spread spectrum modulation, the full 32-bit AID[31:0] value is subtracted (decremented) and added (incremented) to the zero PPM value every APLL input clock cycle for the number of clock cycles specified by the ASCNT registers. The zero PPM value is the nominal AFBDIV register value (when ACR1.USEFDIV=0) or the nominal fractional output divider value (when ACR1.USEFDIV=1). See section 5.5.8.

**Register Name:** AID2, A2ID2
**Register Description:** APLL Increment/Decrement Register 2
**Register Address:** APLL1: 11Ah, APLL2: 19Ah

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **Name** | AID[15:8] | | | | | | | |
| **Default** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 7 to 0: APLL Increment/Decrement (AID[15:8]).** See the AID1 register description.

**Register Name:** AID3, A2ID3
**Register Description:** APLL Increment/Decrement Register 3
**Register Address:** APLL1: 11Bh, APLL2: 19Bh

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **Name** | AID[23:16] | | | | | | | |
| **Default** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 7 to 0: APLL Increment/Decrement (AID[23:16]).** See the AID1 register description.

**Register Name:** AID4, A2ID4
**Register Description:** APLL Increment/Decrement Register 4
**Register Address:** APLL1: 11Ch, APLL2: 19Ch

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **Name** | AID[31:24] | | | | | | | |
| **Default** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 7 to 0: APLL Increment/Decrement (AID[31:24]).** See the AID1 register description.

**Register Name:** F1CR1, F2CR1
**Register Description:** APLL Fractional Output Divider Control Register 1
**Register Address:** APLL1: 140h, APLL2: 1C0h

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **Name** | MODE[3:0] | | | | — | FDL[2:0] | | |
| **Default** | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |

**Bits 7 to 4: APLL Fractional Output Divider Mode (MODE[3:0]).**
0000 = Normal, divider-only operation
0001 = NCO or spread-spectrum operation
Other values reserved

**Bits 2 to 0: APLL Fractional Output Divider Write Length (FDL[2:0]).** This field indicates the last register to write of the FDIV multiregister field where “last register” is as described in section 6.1.3. The number of the last register to write is FDL+1. Changing this field from its default value can be useful in NCO mode to reduce SPI or I2C bus usage when NCO changes only affect the least significant bytes of FDIV.

**Register Name:** F1DIV1, F2DIV1
**Register Description:** APLL Fractional Output Divider Register 1
**Register Address:** APLL1: 141h, APLL2: 1C1h

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **Name** | FDIV[7:0] | | | | | | | |
| **Default** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 7 to 0: Fractional Output Divider Value (FDIV[7:0]).** The full 40-bit FDIV[39:0] field spans the F1DIV1 through F1DIV5 registers. FDIV is an unsigned number with 4 integer bits (FDIV[39:36]) and up to 36 fractional bits. FDIV specifies the fixed-point term of the fractional output divider value. The value FDIV=0 is undefined. Unused least significant bits must be written with 0. See section 5.5.5.

**Register Name:** F1DIV2, F2DIV2
**Register Description:** APLL Fractional Output Divider Register 2
**Register Address:** APLL1: 142h, APLL2: 1C2h

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **Name** | FDIV[15:8] | | | | | | | |
| **Default** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 7 to 0: Fractional Output Divider Value (FDIV[15:8]).** See the F1DIV1 register description.

**Register Name:** F1DIV3, F2DIV3
**Register Description:** APLL Fractional Output Divider Register 3
**Register Address:** APLL1: 143h, APLL2: 1C3h

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **Name** | FDIV[23:16] | | | | | | | |
| **Default** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 7 to 0: Fractional Output Divider Value (FDIV[23:16]).** See the F1DIV1 register description.

**Register Name:** F1DIV4, F2DIV4
**Register Description:** APLL Fractional Output Divider Register 4
**Register Address:** APLL1: 144h, APLL2: 1C4h

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **Name** | FDIV[31:24] | | | | | | | |
| **Default** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 7 to 0: Fractional Output Divider Value (FDIV[31:24]).** See the F1DIV1 register description.

**Register Name:** F1DIV5, F2DIV5
**Register Description:** APLL Fractional Output Divider Register 2
**Register Address:** APLL1: 145h, APLL2: 1C5h

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **Name** | FDIV[39:32] | | | | | | | |
| **Default** | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 |

**Bits 7 to 0: Fractional Output Divider Value (FDIV[39:32]).** See the F1DIV1 register description.

**Register Name:** F1DEN1, F2DEN1
**Register Description:** APLL Fractional Output Divider Denominator Register 1
**Register Address:** APLL1: 146h, APLL2: 1C6h

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **Name** | FDEN[7:0] | | | | | | | |
| **Default** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

**Bits 7 to 0: Fractional Output Divider Denominator (FDEN[7:0]).** The full 32-bit FDEN[31:0] field spans F1DEN1 through F1DEN4 registers. FDEN is an unsigned integer that specifies the denominator of the fractional output divider value. The value FDEN=0 is undefined. See section 5.5.5.

**Register Name:** F1DEN2, F2DEN2
**Register Description:** APLL Fractional Output Divider Denominator Register 2
**Register Address:** APLL1: 147h, APLL2: 1C7h

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **Name** | FDEN[15:8] | | | | | | | |
| **Default** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 7 to 0: Fractional Output Divider Denominator (FDEN [15:8]).** See the F1DEN1 register description.

**Register Name:** F1DEN3, F2DEN3
**Register Description:** APLL Fractional Output Divider Denominator Register 3
**Register Address:** APLL1: 148h, APLL2: 1C8h

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **Name** | FDEN[23:16] | | | | | | | |
| **Default** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 7 to 0: Fractional Output Divider Denominator (FDEN [23:16]).** See the F1DEN1 register description.

**Register Name:** F1DEN4, F2DEN4
**Register Description:** APLL Fractional Output Divider Denominator Register 4
**Register Address:** APLL1: 149h, APLL2: 1C9h

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **Name** | FDEN[31:24] | | | | | | | |
| **Default** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 7 to 0: Fractional Output Divider Denominator (FDEN [31:24]).** See the F1DEN1 register description.

**Register Name:** F1REM1, F2REM1
**Register Description:** APLL Fractional Output Divider Remainder Register 1
**Register Address:** APLL1: 14Ah, APLL2: 1CAh

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **Name** | FREM[7:0] | | | | | | | |
| **Default** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 7 to 0: Fractional Output Divider Remainder (FREM[7:0]).** The full 32-bit FREM[31:0] field spans F1REM1 through F1REM4 registers. FREM is an unsigned integer that specifies the remainder of the fractional output divider value. See section 5.5.5.

**Register Name:** F1REM2, F2REM2
**Register Description:** APLL Fractional Output Divider Remainder Register 2
**Register Address:** APLL1: 14Bh, APLL2: 1CBh

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **Name** | FREM[15:8] | | | | | | | |
| **Default** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 7 to 0: Fractional Output Divider Remainder (FREM [15:8]).** See the F1REM1 register description.

**Register Name:** F1REM3, F2REM3
**Register Description:** APLL Fractional Output Divider Remainder Register 3
**Register Address:** APLL1: 14Ch, APLL2: 1CCh

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **Name** | FREM[23:16] | | | | | | | |
| **Default** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 7 to 0: Fractional Output Divider Remainder (FREM [23:16]).** See the F1REM1 register description.

**Register Name:** F1REM4, F2REM4
**Register Description:** APLL Fractional Output Divider Remainder Register 4
**Register Address:** APLL1: 14Dh, APLL2: 1CDh

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **Name** | FREM[31:24] | | | | | | | |
| **Default** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 7 to 0: Fractional Output Divider Remainder (FREM [31:24]).** See the F1REM1 register description.

**Register Name:** F1BP, F2BP
**Register Description:** APLL Fractional Output Divider Truncate Bit Position
**Register Address:** APLL1: 14Eh, APLL2: 1CEh

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **Name** | — | — | FBP[5:0] | | | | | |
| **Default** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 5 to 0: APLL Fractional Output Divider Truncate Bit Position (FBP[5:0]).** This unsigned integer specifies the number of fractional bits that are valid in the fractional output divider value (FDIV). There are 36 fractional bits in FDIV. The value in this FBP field specifies 36 – number_of_valid_FDIV_fractional_bits. When FBP=0 all 36 FDIV fractional bits are valid. When FBP=12, the most significant 24 FDIV fractional bits are valid and the least significant 12 bits must be set to 0. This register field is only used when the feedback divider value is expressed in the form FDIV + FREM / FDEN. FBP values greater than 36 are invalid. See section 5.5.5.

### 6.3.4 Output Clock Configuration Registers

**Register Name:** OCxCR1
**Register Description:** Output Clock x Configuration Register 1
**Register Address:** OC1: 200h, OC2: 210h, OC3: 220h, OC4: 230h, OC5: 240h
OC6: 250h, OC7: 260h, OC8: 270h, OC9: 280h, OC10: 290h

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **Name** | PHEN | MSDIV[6:0] | | | | | | |
| **Default** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 7: Phase Alignment Enable (PHEN).** This bit enables this output to participate in phase alignment. See section 5.6.5.

0 = Phase alignment disabled for this output
1 = Phase alignment enabled for this output

**Bits 6 to 0: Medium-Speed Divider Value (MSDIV[6:0]).** This field specifies the setting for the output clock's medium-speed divider. The divisor is MSDIV+1. Note that if MSDIV is not set to 0 (bypass) then the maximum input clock frequency to the medium-speed divider is 750MHz and the maximum output clock frequency from the medium-speed divider is 375MHz. When MSDIV=0, the medium-speed divider, phase adjust, low-speed divider, start/stop and output duty cycle adjustment circuits are bypassed and the high-frequency clock signal is sent to the directly output driver. See section 5.6.2.

**Register Name:** OCxCR2
**Register Description:** Output Clock x Configuration Register 2
**Register Address:** OC1: 201h, OC2: 211h, OC3: 221h, OC4: 231h, OC5: 241h
OC6: 251h, OC7: 261h, OC8: 271h, OC9: 281h, OC10: 291h

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **Name** | — | POL | DRIVE[1:0] | | OCSF[3:0] | | | |
| **Default** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 6: Clock Path Polarity (POL).** The clock path to the output driver is inverted when this bit set. This does not invert the LSDIV path to the CMOS OCxN pin if that path is enabled. See section 5.6.1.

**Bits 5 to 4: CMOS/HSTL Output Drive Strength (DRIVE[1:0]).** The CMOS/HSTL output drivers have four equal sections that can be enabled or disabled to achieve four different drive strengths from 1x to 4x. When the output power supply VDDOx is 3.3V or 2.5V, the user should start with 1x and only increase drive strength if the output is highly loaded and signal transition time is unacceptable. When VDDOx is 1.8V or 1.5V the user should start with 4x and only decrease drive strength if the output signal has unacceptable overshoot. See section 5.6.1.

00 = 1x
01 = 2x
10 = 3x
11 = 4x

**Bits 3 to 0: Output Clock Signal Format (OCSF[3:0]).** See section 5.6.1.

0000 = Disabled (high-impedance, low power mode)
0001 = LVDS ($V_{OD}$ is forced to 400mV and OCxDIFF.VOD is ignored)
0010 = Differential (default is LVPECL with $V_{CM}$=1.2V, programmable using OCxDIFF fields)
0011 = HSTL (set OCxCR2.DRIVE=11 (4x) to meet JESD8-6)
0100 = Two CMOS: OCxN in phase with OCxP
0101 = One CMOS: OCxP enabled, OCxN high impedance
0110 = One CMOS: OCxP high impedance, OCxN enabled
0111 = Two CMOS: OCxN inverted vs. OCxP
1010 = HCSL

**Register Name:** OCxDIFF
**Register Description:** Output Clock x Start Stop Register
**Register Address:** OC1: 202h, OC2: 212h, OC3: 222h, OC4: 232h, OC5: 242h
OC6: 252h, OC7: 262h, OC8: 272h, OC9: 282h, OC10: 292h

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **Name** | VCM[3:0] | | | | VOD[3:0] | | | |
| **Default** | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |

**Bits 7 to 4: Differential Common-Mode Voltage (VCM[3:0]).** This field specifies the common-mode voltage for the differential output driver. See section 5.6.1.

0000 = 1.23V (default) – typical for LVDS and AC-coupled LVPECL
0011 = 1.0V
0100 = 1.1V
0101 = 1.3V
0110 = 1.4V
0111 = 1.5V
1000 = 1.6V
1001 = 1.7V
1010 = 1.8V
1011 = 1.9V
1100 = 2.0V – typical for DC-coupled LVPECL
1101 = 2.1V
1111 = Use this setting for HCSL signal format
All other values reserved

**Bits 3 to 0: Differential Swing Voltage (VOD[3:0]).** This field specifies the differential output voltage ($V_{OD}$) for the differential output driver. In the device this field actually controls driver output current. When the specified current is driven into the required external 100Ω termination resistor, the voltage across the termination resistor is the desired $V_{OD}$. See Figure 16 for the definition of $V_{OD}$. $V_{OD}$ is equivalent to the single-ended voltage swing of the OCxP pin or the OCxN pin. This field is ignored and $V_{OD}$ is set to 400mV when OCxCR2.OCSF=0001 (LVDS). See section 5.6.1.

0000 = 300mV (3mA driver current)
0001 = 400mV – typical for LVDS
0010 = 500mV
0011 = 600mV
0100 = 700mV
0101 = 800mV – default value, typical for LVPECL
0110 = 900mV (9mA driver current)
1010 = Use this setting for HCSL signal format
All other values reserved

**Register Name:** OCxREG
**Register Description:** Output Clock x Regulator Control Register
**Register Address:** OC1: 203h, OC2: 213h, OC3: 223h, OC4: 233h, OC5: 243h
OC6: 253h, OC7: 263h, OC8: 273h, OC9: 283h, OC10: 293h

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **Name** | — | — | — | — | VREG[3:0] | | | |
| **Default** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 3 to 0: Regulator Voltage (VREG[3:0]).** This field specifies the power supply regulator voltage for the differential output driver. Set this to at least VCM+VOD/2+0.5V. Max value is 2.2V when VDDOx is 2.5V, and max value is 2.9V when VDDOx is 3.3V. See section 5.6.1.

0000 = 2.2V (default) – typical for LVDS and AC-coupled LVPECL
0010 = 2.0V
0011 = 2.2V
0100 = 2.25V
0101 = 2.4V
0111 = 2.5V
1000 = 2.7V
1001 = 2.75V
1010 = 2.8V
1011 = 2.9V – typical for DC-coupled LVPECL
1100 = 3.0V
1111 = Use this setting for HCSL signal format
All other values reserved

**Register Name:** OCxCR3
**Register Description:** Output Clock x Configuration Register 3
**Register Address:** OC1: 204h, OC2: 214h, OC3: 224h, OC4: 234h, OC5: 244h
OC6: 254h, OC7: 264h, OC8: 274h, OC9: 284h, OC10: 294h

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **Name** | SRLSEN | — | NEGLSD | LSSEL | — | ASQUEL | — | LSDIV[24] |
| **Default** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 7: Enable LSDIV Statuses (SRLSEN).** This bit enables the OCxSR.LSCLK real-time status bit and its associated latched status bit OCxSR.LSCLKL.

0 = LSCLK status bit is not enabled (low)
1 = LSCLK status bit is enabled

**Bit 5: OCxN Low Speed Divider (NEGLSD).** This bit selects the source of the clock on the OCxN pin in CMOS mode. See section 5.6.2.

0 = Same as OCxP
1 = Output of the LSDIV divider

Note: NEGLSD should only be set to one in two-CMOS mode (OCxCR2.OCSF=100 or 111), when OCxCR2.POL=0, and when OCxCR3.LSSEL=0.

**Bit 4: LSDIV Select (LSSEL).** This bit selects the source of the output clock. When the MSDIV divider is selected (LSSEL=0) the LSDIV divider output can be independently selected as the source for the OCxN pin (in CMOS output mode) or monitored by the OCxSR.LSCLK status bit. This bit is only valid when OCxCR1.MSDIV > 0. See section 5.6.2.

0 = The output clock is sourced from the MSDIV divider.
1 = The output clock is sourced from the LSDIV divider.

**Bit 2: Auto-Squelch Enable (ASQUEL).** This bit enables automatic squelching of the output clock whenever (a) automatic input switching is enabled for the APLL to which the output is connected, and (b) the input monitors indicate that the inputs specified by that APLL's APLLMUX and ALTMUX fields are both invalid. OCxSTOP.MODE specifies the type of stop (high, low, high-impedance) and the output clock edge on which to stop. When a differential output or a CMOS complementary output is squelched, its OCxN pin is opposite polarity of its OCxP pin.

0 = Auto-squelch disabled
1 = Auto-squelch enabled

**Bit 0: Low-Speed Divider Value (LSDIV[24]).** See the OCxDIV1 register description.

**Register Name:** OCxDIV1
**Register Description:** Output Clock x Divider Register 1
**Register Address:** OC1: 205h, OC2: 215h, OC3: 225h, OC4: 235h, OC5: 245h
OC6: 255h, OC7: 265h, OC8: 275h, OC9: 285h, OC10: 295h

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **Name** | LSDIV[7:0] | | | | | | | |
| **Default** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 7 to 0: Low-Speed Divider Value (LSDIV[7:0]).** The full 25-bit LSDIV[24:0] field spans this register, OCxDIV2, OCxDIV3. and bit 0 of OCxCR3. LSDIV is an unsigned integer. The frequency of the clock from the medium-speed divider is divided by LSDIV+1. The OCxCR3.LSSEL and NEGLSD bits control when the output of the low-speed divider is present on the OCxP and OCxN output pins. OCxCR1.MSDIV must be > 0 for the low-speed divider to operate. See section 5.6.2.

**Register Name:** OCxDIV2
**Register Description:** Output Clock x Divider Register 2
**Register Address:** OC1: 206h, OC2: 216h, OC3: 226h, OC4: 236h, OC5: 246h
OC6: 256h, OC7: 266h, OC8: 276h, OC9: 286h, OC10: 296h

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **Name** | LSDIV[15:8] | | | | | | | |
| **Default** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 7 to 0: Low-Speed Divider Value (LSDIV[15:8]).** See the OCxDIV1 register description.

**Register Name:** OCxDIV3
**Register Description:** Output Clock x Divider Register 3
**Register Address:** OC1: 207h, OC2: 217h, OC3: 227h, OC4: 237h, OC5: 247h
OC6: 257h, OC7: 267h, OC8: 277h, OC9: 287h, OC10: 297h

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **Name** | LSDIV[23:16] | | | | | | | |
| **Default** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 7 to 0: Low-Speed Divider Value (LSDIV[23:16]).** See the OCxDIV1 register description.

**Register Name:** OCxDC
**Register Description:** Output Clock x Duty Cycle Register
**Register Address:** OC1: 208h, OC2: 218h, OC3: 228h, OC4: 238h, OC5: 248h
OC6: 258h, OC7: 268h, OC8: 278h, OC9: 288h, OC10: 298h

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **Name** | OCDC[7:0] | | | | | | | |
| **Default** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 7 to 0: Output Clock Duty Cycle (OCDC[7:0]).** This field controls the output clock signal duty cycle when MSDIV>0 and LSDIV>1. When OCDC = 0 the output clock is 50%. Otherwise the clock signal is a pulse with a width of OCDC number of MSDIV output clock periods. The range of OCDC can create pulse widths from 1 to 255 MSDIV output clock periods. When OCxCR2.POL=0, the pulse is high and the signal is low the remainder of the cycle. When POL=1, the pulse is low and the signal is high the remainder of the cycle. See section 5.6.3.

**Register Name:** OCxPH
**Register Description:** Output Clock x Phase Adjust Register
**Register Address:** OC1: 209h, OC2: 219h, OC3: 229h, OC4: 239h, OC5: 249h
OC6: 259h, OC7: 269h, OC8: 279h, OC9: 289h, OC10: 299h

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **Name** | — | — | — | — | PHADJ[3:0] | | | |
| **Default** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 3 to 0: Phase Adjust Value (PHADJ[3:0]).** This field can be used to adjust the phase of an output clock vs. the phase of other clock outputs. The adjustment is in units of bank source clock cycles. For example, if the bank source clock is 625MHz (from APLL1 for example) then one bank source clock cycle is 1.6ns, the smallest phase adjustment is 0.8ns, and the adjustment range is ±5.6ns. Negative values mean earlier in time (leading) and positive values mean later in time (lagging). See section 5.6.4.

0000 = 0 bank source clock cycles
0001 = 0.5
0010 = 1.0
0011 = 1.5
0100 = 2.0
0101 = 2.5
0110 = 3.0
0111 = 3.5
1000 = -1.0 bank source clock cycles
1001 = -0.5
1010 = -2.0
1011 = -1.5
1100 = -3.0
1101 = -2.5
1110 = -4.0
1111 = -3.5

**Register Name:** OCxSTOP
**Register Description:** Output Clock x Start Stop Register
**Register Address:** OC1: 20Ah, OC2: 21Ah, OC3: 22Ah, OC4: 23Ah, OC5: 24Ah
OC6: 25Ah, OC7: 26Ah, OC8: 27Ah, OC9: 28Ah, OC10: 29Ah

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **Name** | — | SRC[3:0] | | | | NEGLSD | MODE[1:0] | |
| **Default** | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |

**Bits 6 to 3: Output Clock Stop Source (SRC[3:0]).** This field specifies the source of the stop signal. See section 5.6.7.

0000 = Never stop
0001 = Logical OR of (the global MCR1.STOP bit) or (the OCx stop bit in the STOPCR registers)
0010 to 0111 = {unused values}
1000 = GPIO0
1001 = GPIO1
1010 = GPIO2
1011 = GPIO3
1100 to 1111 = {unused values}

**Bit 2: NEGLSD Stop Behavior (NEGLSD).** When an output pair is configured for two different frequencies in 2xCMOS mode (see section 5.6.2) this bit specifies the stop behavior for the pair. This field allows the user to trade off stop reaction time vs. possible short pulse on the NEG pin.

0 = Stop when higher-speed POS signal has the appropriate edge (see MODE field below)
1 = Stop when lower-speed NEG signal has the appropriate edge.

Setting this bit to 1 guarantees no short high/low time for the POS signal and for the NEG signal, but stopping can take a long time when the NEG pin is very low frequency, such as 2kHz or even 1Hz.

Setting this bit to 0 allows stopping to happen faster because it depends only on the frequency of the POS signal, but the NEG signal may have a short high or low time when it stops. For some applications, such as when NEG is

a 1 pulse per second (PPS) signal, a short high or low time when NEG stops may not matter because NEG is essentially a data signal (phase alignment or time alignment signal that is latched by a POS signal edge) rather than a true clock signal.

**Bits 1 to 0: Output Clock Stop Mode (MODE[1:0]).** This field selects the mode of the start-stop function. See section 5.6.6.

00 = Stop Low: stop after falling edge of output clock, start after rising edge of output clock
01 = Stop High: stop after rising edge of output clock, start after falling edge of output clock
10 = Stop Low then go high-impedance: stop after falling edge, start after rising edge
11 = Stop High then go high-impedance: stop after rising edge, start after falling edge

The following table shows which pin(s) stop high or low as specified above for each output signal format:

| Signal Format | OCxCR2.OCSF | Pin that Stops As Specified |
|---|---|---|
| LVDS, LVPECL, Programmable Differential | 001 or 010 | OCxP |
| HSTL | 011 | OCxP |
| Two CMOS, OCxP in phase with OCxN | 100 | OCxP and OCxN |
| One CMOS, OCxN enabled | 101 | OCxN |
| One CMOS, OCxP enabled | 110 | OCxP |
| Two CMOS, OCxN inverted vs. OCxP | 111 | OCxP |

### 6.3.5 Input Clock Configuration Registers

**Register Name:** XACR1
**Register Description:** XA Input Clock Configuration Register 1
**Register Address:** 300h

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **Name** | — | POL | DISMON | VALTIME[2:0] | | | HSDIV[1:0] | |
| **Default** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 6: Input Polarity (POL).** This field specifies which input clock edge the APLL will lock to. See section 5.5.1.
0 = Rising edge
1 = Falling edge

**Bit 5: Disable Signal Going to the Monitor (DISMON).**
0 = XA signal is provided to the XA monitor
1 = XA signal is not provided to the XA monitor

**Bits 4 to 2: Input Validation Time (VALTIME[2:0]).** The input clock monitor only declares the XA input clock valid if it has no missing edges in the interval specified by this field. See section 5.5.2.
000 = 1 cycle
001 = 4 cycles
010 = 16 cycles
011 = 64 cycles
100 = 256 cycles
101 = 1024 cycles
110 - 111 = {unused values}

**Bits 1 to 0: Input Clock High-Speed Divider (HSDIV[1:0]).** This field specifies the divide value for the XA input clock divider. This field should not be set to 01 at the same time MCR2.DBL=1. See section 5.5.1.
00 = Divide by 1
01 = Divide by 2
10, 11 = {unused values}

**Register Name:** XACR2
**Register Description:** XA Input Clock Configuration Register 2
**Register Address:** 301h

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **Name** | XOAMP[7:0] | | | | | | | |
| **Default** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 7 to 0: XO Amplifier Control (XOAMP[7:0]).** Set this value as follows for the recommended 10pF crystal (values in decimal). Contact Microsemi apps support for XOAMP values for crystals with other load capacitances.

| Crystal Frequency (MHz) | Max Crystal Drive | | |
|---|---|---|---|
| | 100µW | 200µW | 300µW |
| 25 | 0 | 80 | 152 |
| 30 | 0 | 72 | 136 |
| 35 | 0 | 72 | 136 |
| 40 | 8 | 80 | 136 |
| 45 | 8 | 80 | 136 |
| 50 | 16 | 88 | 136 |
| 55 | 16 | 88 | 136 |
| 60 | 24 | 96 | 136 |

**Register Name:** XACR3
**Register Description:** XA Input Clock Configuration Register 3
**Register Address:** 302h

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **Name** | XBCAP[3:0] | | | | XACAP[3:0] | | | |
| **Default** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 7 to 4: XB Internal Capacitor Selection (XBCAP[3:0]).** Actual internal capacitance on the XB pin in pF is approximately 6 + XBCAP. See section 5.3.2.

**Bits 3 to 0: XA Internal Capacitor Selection (XACAP[3:0]).** Actual internal capacitance on the XA pin in pF is approximately 6 + XACAP. See section 5.3.2.

**Register Name:** ICxCR1
**Register Description:** Input Clock x Configuration Register 1
**Register Address:** IC1: 303h, IC2: 304h, IC3: 305h

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **Name** | — | POL | DISMON | VALTIME[2:0] | | | HSDIV[1:0] | |
| **Default** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 6: Input Polarity (POL).** This field specifies which input clock edge the APLL will lock to. See section 5.5.1.
0 = Rising edge
1 = Falling edge

**Bit 5: Disable Signal Going to the Monitor (DISMON).**
0 = ICx signal is provided to the ICx monitor
1 = ICx signal is not provided to the ICx monitor

**Bits 4 to 2: Input Validation Time (VALTIME[2:0]).** The input clock monitor only declares the input clock valid if it has no missing edges in the interval specified by this field. See section 5.5.2.
000 = 1 cycle
001 = 4 cycles
010 = 16 cycles
011 = 64 cycles
100 = 256 cycles
101 = 1024 cycles
110 - 111 = {unused values}

**Bits 1 to 0: Input Clock High-Speed Divider (HSDIV[1:0]).** This field specifies the divide value for the input clock high-speed divider. See section 5.5.1.
00 = Divide by 1
01 = Divide by 2
10 = Divide by 4
11 = Divide by 8

# 7. Electrical Characteristics

**Absolute Maximum Ratings**

| Parameter | Symbol | Min. | Max. | Units |
|---|---|---|---|---|
| Supply voltage, nominal 1.5V | VDD15 | -0.3 | 1.65 | V |
| Supply voltage, nominal 1.8V | VDD18 | -0.3 | 1.98 | V |
| Supply voltage, nominal 2.5V | VDD25 | -0.3 | 2.75 | V |
| Supply voltage, nominal 3.3V | VDD33 | -0.3 | 3.63 | V |
| Voltage on XA, any ICxP/N, any OCxP/N pin | VANAPIN | -0.3 | 3.63 | V |
| Voltage on any digital I/O pin | VDIGPIN | -0.3 | 3.63 | V |
| Storage Temperature Range | $T_{ST}$ | -55 | +125 | °C |

* Exceeding these values may cause permanent damage. Functional operation under these conditions is not implied.
* Voltages are with respect to ground (VSS) unless otherwise stated.

**Note 1:** The typical values listed in the tables of Section 7 are not production tested.
**Note 2:** Specifications to -40°C and 85°C are guaranteed by design or characterization and not production tested.

**Table 5 - Recommended DC Operating Conditions**

| Parameter | Symbol | Min. | Typ. | Max. | Units |
|---|---|---|---|---|---|
| Supply voltage, Higher Core (choose 1 row) | VDDH | 2.375 | 2.5 | 2.625 | V |
| | | 3.135 | 3.3 | 3.465 | |
| Supply voltage, Lower Core (choose 1 row) | VDDL | 1.71 | 1.8 | 1.89 | V |
| | | same as VDDH | | | |
| Supply voltage, Non-Clock I/O Pins (choose 1 row) | VDDIO | 1.71<br>2.375 | 1.8<br>2.5 | 1.89<br>2.625 | V |
| | | same as VDDH | | | |
| Supply voltage, OCx Outputs (x=A,B,C,D,E or F) (choose 1 row) | VDDOx | 1.425<br>1.71<br>2.375 | 1.5<br>1.8<br>2.5 | 1.575<br>1.89<br>2.625 | V |
| | | same as VDDH | | | |
| Operating temperature | $T_A$ | -40 | | +85 | °C |

**Table 6 - Electrical Characteristics: Supply Currents**

| Characteristics | Symbol | Min. | Typ.[1] | Max | Units | Notes |
|---|---|---|---|---|---|---|
| Total power, two ICx inputs, two APLLs and two LVDS outputs enabled, 1.8V+3.3V operation | $P_{DISS}$ | | 1.065 | | W | |
| 3.3V single-supply operation<br>Total current on 3.3V supply | $I_{DD33}$ | | 569 | 852 | mA | Note 2 |
| 2.5V single-supply operation<br>Total current on 2.5V supply | $I_{DD25}$ | | 550 | 843 | mA | Note 2 |
| 1.8V+3.3V operation<br>Total current on 3.3V supply<br>Total current on 1.8V supply | <br>$I_{DD33}$<br>$I_{DD18}$ | | <br>221<br>280 | <br>342<br>477 | mA | Note 2 |
| 1.8V+2.5V operation<br>Total current on 2.5V supply<br>Total current on 1.8V supply | <br>$I_{DD25}$<br>$I_{DD18}$ | | <br>212<br>279 | <br>339<br>479 | mA | Note 2 |
| VDDH supply current change from enabling or disabling the crystal driver circuit | $\Delta I_{DDXO}$ | | 13 | | mA | |
| VDDL supply current change from enabling or disabling an input clock | $\Delta I_{DDLIN}$ | | 12 | | mA | |
| VDDL supply current change from enabling or disabling the APLL's fractional output divider | $\Delta I_{DDLHSD}$ | | 38 | | mA | |
| VDDL supply current from enabling/disabling output divider for one OCx using OCEN.OCxEN | $\Delta I_{DDLDIV}$ | | 13 | | mA | |
| VDDL supply current change from enabling or | $\Delta I_{DDLD}$ | | 13 | | mA | |

| Characteristics | Symbol | Min. | Typ.[1] | Max | Units | Notes |
|---|---|---|---|---|---|---|
| disabling an output for LVDS, LVPECL or HCSL | | | | | | |
| VDDL supply current change from enabling or disabling an output for CMOS or HSTL | $\Delta I_{DDLC}$ | | 16 | | mA | |
| VDDOx supply current change from enabling or disabling an LVDS output | $\Delta I_{DDOD}$ | | 9 | | mA | |
| VDDOx supply current change from enabling or disabling an LVPECL output | $\Delta I_{DDOP}$ | | 15 | | mA | |
| VDDOx supply current change from enabling or disabling an HCSL output | $\Delta I_{DDOHC}$ | | 15 | | mA | Note 5 |
| VDDOx supply current change from enabling or disabling a CMOS output | $\Delta I_{DDOC}$ | | 6 | | mA | Note 3 |
| VDDOx supply current change from enabling or disabling an HSTL output | $\Delta I_{DDOHS}$ | | 6 | | mA | Note 4 |

**Note 1:** Typical values measured at nominal supply voltages and 25°C ambient temperature.

**Note 2:** Max $I_{DD}$ measurements made with all blocks enabled, 650MHz signals on IC1 and IC2 inputs, 187.5MHz signal on IC3, Crystal driver and doubler off , VCO frequency of 3750MHz, APLL output dividers dividing by 6, all MSDIV dividing by 2, all LSDIV dividing by 2, all outputs enabled as LVPECL outputs driving 156.25MHz signals, and all VDDO at same voltage as VDDH. Typical $I_{DD}$ measurements made with same setup as max $I_{DD}$ but APLL fractional dividers disabled and only six outputs enabled with LVDS signal format.

**Note 3:** VDDOx=3.3V, 1x drive strength, $f_O$=250MHz, 2pF load

**Note 4:** VDDOx=1.8V, 2x drive strength, $f_O$=100MHz, 100Ω differential termination.

**Note 5:** 50Ω to ground each on OCxP and OCxN.

**Table 7 - Electrical Characteristics: Non-Clock CMOS Pins**

| Characteristics | | Symbol | Min. | Typ. | Max. | Units | Notes |
|---|---|---|---|---|---|---|---|
| Input high voltage | | $V_{IH}$ | 0.7 x VDDIO | | | V | |
| Input low voltage | | $V_{IL}$ | | | 0.3 x VDDIO | V | |
| Input leakage current, all digital inputs | | $I_{IL}$ | -10 | | 10 | µA | Note 1 |
| Input capacitance | | $C_{IN}$ | | 3 | 10 | pF | |
| Input capacitance, SCL/SCLK, SDA/MOSI | | $C_{IN}$ | | 3 | 11 | pF | |
| Input hysteresis, SCL and SDA in $I^2C$ Bus Mode | | | 0.05 x VDDIO | | | mV | |
| Output leakage (when high impedance) | | $I_{LO}$ | -10 | | 10 | µA | Note 1 |
| Output high voltage | | $V_{OH}$ | 0.8 x VDDIO | | | V | $I_O$ = -3.0mA |
| Output low voltage | | $V_{OL}$ | | | 0.2 x VDDIO | V | $I_O$ = 3.0mA |
| Clock output on GPIO pin, frequency | | $f_{OUT}$ | | | 50 | MHz | Note 3 |
| Clock output on GPIO pin, rise/fall time | VDDIO=1.8V | $t_R$, $t_F$ | | 2.3 | | ns | Notes 3, 4 |
| | VDDIO=2.5V | | | 1.5 | | ns | Notes 3, 4 |
| | VDDIO=3.3V | | | 1.2 | | ns | Notes 3, 4 |

**Note 1:** 0V < $V_{IN}$ < VDDIO for all other non-clock inputs.

**Note 2:** $V_{OH}$ does not apply for SCL and SDA in $I^2C$ interface mode since they are open drain.

**Note 3:** To output a clock on a GPIO pin, an OCx output must be configured with NEGLSD=1 and SRLSEN=1 in OCxCR3 and the GPIO must be configured to as a status output following OCxSR.LSCLK (see the GPIOCR and GPIOxSS registers). Output jitter is not guaranteed for clock signals on GPIO pins but is typically 1 to 5ps rms 12kHz to 20MHz.

**Note 4:** 20%-80%, 15pF load.

**Table 8 - Electrical Characteristics: XA Clock Input**

This table covers the case when there is no external crystal connected and an external oscillator or clock signal is connected to the XA pin.

| Characteristics | Symbol | Min. | Typ. | Max. | Units | Notes |
|---|---|---|---|---|---|---|
| Input high voltage, XA | $V_{IH}$ | 1.2 | | VDDH | V | VDDH=2.5 or 3.3V |
| Input low voltage, XA | $V_{IL}$ | | | 0.8 | V | VDDH=2.5 or 3.3V |
| Input frequency, XA pin to APLL mux | $f_{IN}$ | 9.72 | | 156.25 | MHz | |
| Input frequency, XA pin to bypass mux | $f_{IN}$ | | | 156.25 | MHz | |
| Input leakage current | $I_{IL}$ | -10 | | 10 | µA | |
| Input duty cycle | | 40 | | 60 | % | Note 1 |

**Note 1:** 1.0V threshold. See section 5.3.3 for duty cycle restrictions when using the XA doubler with low input frequencies.

**Table 9 - Electrical Characteristics: Clock Inputs, ICxP/N**

| Characteristics | Symbol | Min. | Typ. | Max. | Units | Notes |
|---|---|---|---|---|---|---|
| Input voltage tolerance (each pin, single-ended) | $V_{TOL}$ | 0 | | VDDH | V | Note 1 |
| Input differential voltage | $V_{ID}$ | 0.1 | | 1.4 | V | Note 2 |
| Input DC bias voltage (internally biased) | $V_{CMI}$ | | 1.35 | | V | |
| Input frequency, ICx pins | $f_{IN}$ | 9.72 | | 1250 | MHz | Differential |
| | | 9.72 | | 300 | MHz | Single-ended |
| Minimum input clock high, low time, $f_{IN} \leq 250MHz$ | $t_H$, $t_L$ | | smaller of 3ns or 0.3 x 1 / $f_{IN}$ | | ns | |
| Minimum input clock high, low time, $250MHz \leq f_{IN} \leq 750MHz$ | $t_H$, $t_L$ | | 0.4*1 / $f_{IN}$ | | ns | |
| Minimum input clock high, low time, $f_{IN} \geq 750MHz$ | $t_H$, $t_L$ | | 0.45*1 / $f_{IN}$ | | ns | |
| Input resistance, single-ended to 1.8V, ICxP or ICxN | $R_{IN18}$ | | 50 | | kΩ | |
| Input resistance, single-ended to VSS, ICxP or ICxN | $R_{INVSS}$ | | 80 | | kΩ | |

**Note 1:** The device can tolerate voltages as specified in $V_{TOL}$ w.r.t. VSS on its ICxP and ICxN pins without being damaged. For differential input signals, proper operation of the input circuitry is only guaranteed when the other specifications in this table, including $V_{ID}$, are met.

**Note 2:** For inputs IC1P/N and IC2P/N $V_{ID}=|V_{ICxP} - V_{ICxN}|$. For input IC3P, $V_{ID}=|V_{IC3P} - V_{CMI}|$. The max $V_{ID}$ spec only applies when a differential signal is applied on ICxP/N; it does not apply when a single-ended signal is applied on ICxP.

**Note 3:** **Differential signals.** The differential inputs can easily be interfaced to neighboring ICs driving LVDS, LVPECL, CML, HCSL, HSTL or other differential signal formats using a few external passive components. In general, Microsemi recommends terminating the signal with the termination/load recommended in the neighboring component's data sheet and then AC-coupling the signal into the ICxP/ICxN pins. See Figure 15 for details. To connect a differential signal to IC3, AC-couple one side of the signal to IC3P and AC-couple the other side to VSS. For DC-coupling, treat the input as 1.8V CML.

**Note 4:** **Single-ended signals** can be connected to the ICxP pins. Signals with amplitude greater than 2.5V must be DC-coupled. For signals with amplitudes less than 2.5V Microsemi recommends AC-coupling but DC-coupling can also be used. When a single-ended signal is connected to ICxP, ICxN should be connected to a capacitor (0.1µF or 0.01µF) to VSS.

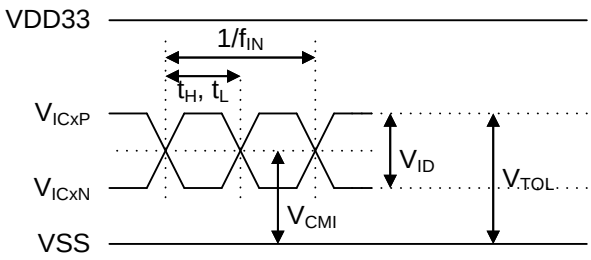



**Figure 14 - Electrical Characteristics: Clock Inputs**

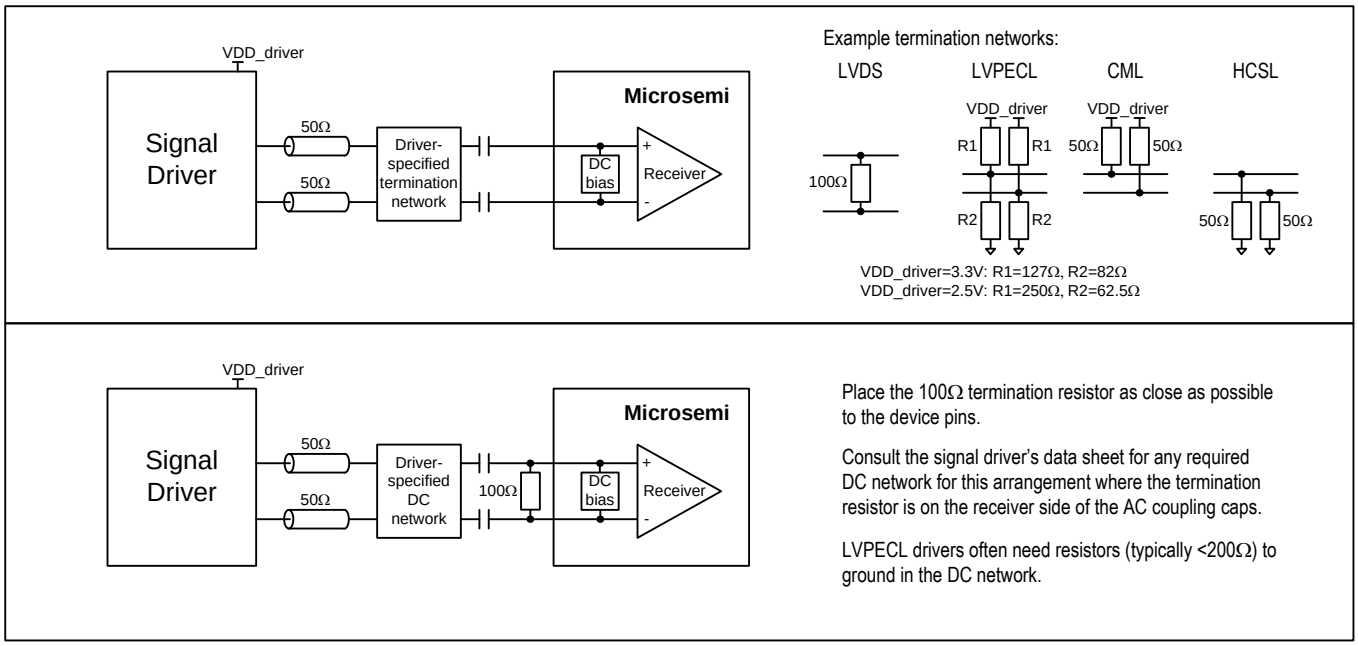



**Figure 15 - Example External Components for Differential Input Signals**

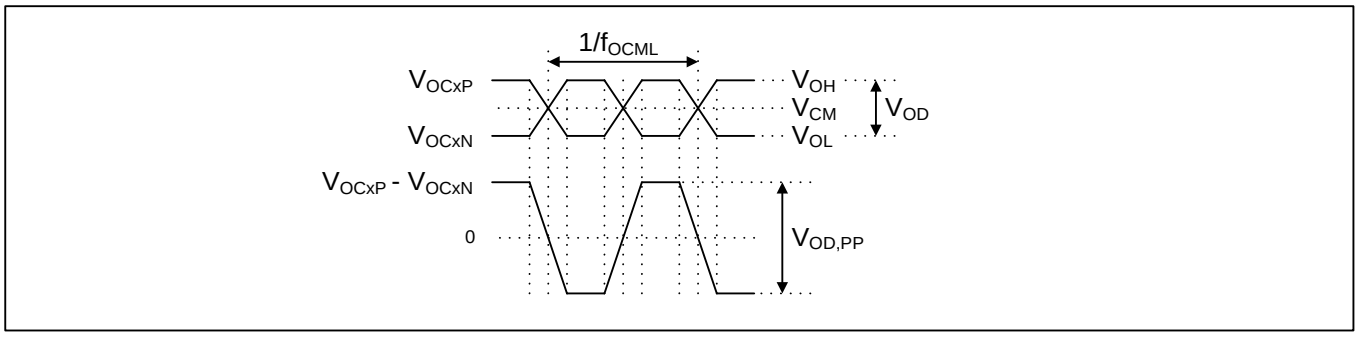



**Figure 16 - Electrical Characteristics: Differential Clock Outputs**

### Table 10 - Electrical Characteristics: LVDS Clock Outputs

VDDOx = 2.5V±5% or 3.3V±5% for LVDS operation.

| Characteristics | Symbol | Min. | Typ. | Max. | Units | Notes |
|---|---|---|---|---|---|---|
| Output frequency | $f_{OCD}$ | | | 1045 | MHz | |
| Output common-mode voltage | $V_{CM}$ | 1.13 | 1.2 | 1.37 | V | Note 1. See Figure 16 |
| Output differential voltage | $V_{OD}$ | 310 | 420 | 530 | mV | Note 1. See Figure 16 |
| Output differential swing, peak-to-peak | $V_{OD,PP}$ | 620 | 840 | 1060 | $mV_{PP}$ | Note 1. See Figure 16 |
| Output rise/fall time | $t_R$, $t_F$ | | 150 | | ps | 20%-80% |
| Output duty cycle | | 45 | 50 | 55 | % | |

**Note 1:** OCxCR2.OCSF=0001 (LVDS). Output must have 100Ω DC path between OUTxP and OUTxN to meet these $V_{OD}$ specs**.** See Figure 17 parts a) and b) for examples where this DC path is a 100Ω termination resistor at the receiver.

### Table 11 - Electrical Characteristics: LVPECL Clock Outputs

VDDOx = 2.5V±5% or 3.3V±5% for LVPECL operation.

| Characteristics | Symbol | Min. | Typ. | Max. | Units | Notes |
|---|---|---|---|---|---|---|
| Output frequency | $f_{OCD}$ | | | 1045 | MHz | |
| Output common-mode voltage, VDDOx=3.3V | $V_{CM}$ | 1.85 | 1.95 | 2.05 | V | Note 1. See Figure 16 |
| Output common-Mode voltage, VDDOx=2.5V | $V_{CM}$ | 1.13 | 1.23 | 1.33 | V | Note 1. See Figure 16 |
| Output differential voltage | $V_{OD}$ | 650 | 820 | 1050 | mV | Note 1. See Figure 16 |
| Output differential swing, peak-to-peak | $V_{OD}$ | 1300 | 1640 | 2100 | $mV_{PP}$ | Note 1. See Figure 16 |
| Output rise/fall time | $t_R$, $t_F$ | | 150 | | ps | 20%-80% |
| Output duty cycle | | 45 | 50 | 55 | % | |

**Note 1:** OCxCR2.OCSF=0010, OCxDIFF.VCM=1100, OCxDIFF.VOD=0101. Output must have 100Ω DC path between OUTxP and OUTxN to meet these $V_{OD}$ specs**.** See Figure 17 parts a) and b) for examples where this DC path is a 100Ω termination resistor at the receiver.

### Table 12 - Electrical Characteristics: HCSL Clock Outputs

VDDOx=VDDH=3.3V±5% or VDDOx=VDDH=2.5V±5% for HCSL operation.

| Characteristics | Symbol | Min. | Typ. | Max. | Units | Notes |
|---|---|---|---|---|---|---|
| Output frequency | $f_{OCHC}$ | | | 250 | MHz | |
| Output common-mode voltage, 2.5V | $V_{CM}$ | 0.315 | 0.355 | 0.395 | V | Note 1. See Figure 16 |
| Output common-mode voltage, 3.3V | $V_{CM}$ | 0.335 | 0.38 | 0.435 | V | Note 1. See Figure 16 |
| Output differential voltage | $V_{OD}$ | 0.62 | 0.75 | 0.86 | V | Note 1. See Figure 16 |
| Output rise/fall time | $t_R$, $t_F$ | | 250 | | ps | 20%-80% |
| Output duty cycle | | 45 | 50 | 55 | % | |

**Note 1:** Each of OCxP and OCxN with 50Ω termination resistor to ground.

### Table 13 - Electrical Characteristics: CMOS and HSTL (Class I) Clock Outputs

| Characteristics | Symbol | Min. | Typ. | Max. | Units | Notes |
|---|---|---|---|---|---|---|
| Output frequency | $f_{OCMOS}$ | <<1Hz | | 250 | MHz | Note 1 |
| Output high voltage | $V_{OH}$ | VDDOx –0.4 | | VDDOx | V | Notes 2, 3 |
| Output low voltage | $V_{OL}$ | 0 | | 0.4 | V | Notes 2, 3 |
| Output rise/fall time, VDDOx=1.8V, OCxCR2.DRIVE=4x | $t_R$, $t_F$ | | 0.4 | | ns | 2pF load |
| Output rise/fall time, VDDOx=1.8V, OCxCR2.DRIVE=4x | | | 1.2 | | ns | 15pF load |
| Output rise/fall time, VDDOx=3.3V, OCxCR2.DRIVE=1x | | | 0.7 | | ns | 2pF load |
| Output rise/fall time, VDDOx=3.3V, OCxCR2.DRIVE=1x | | | 2.2 | | ns | 15pF load |
| Output duty cycle | | 45 | 50 | 55 | % | Note 4, 6 |
| Output duty cycle | | 42 | 50 | 58 | % | Notes 5, 6 |
| Output duty cycle, OCxNEG single-ended | | | 50 | | % | |
| Output duty cycle, OCxPOS single-ended | | | 50 | | % | |
| Output current when output disabled | $I_{OH}$ | | 300 | | µA | OCxCR2.OCSF=0 |

**Note 1:** Minimum output frequency is a function of VCO frequency and output divider values and is guaranteed by design.

**Note 2:** For HSTL Class I, $V_{OH}$ and $V_{OL}$ apply for both unterminated loads and for symmetrically terminated loads, i.e. 50Ω to VDDOx/2.

**Note 3:** For VDDOx=3.3V and OCxCR2.DRIVE=1x, $I_O$=4mA. For VDDOx=1.5V and OCxCR2.DRIVE=4x, $I_O$=8mA.

**Note 4:** Output clock frequency ≤ 160MHz or VDDOx ≥ 1.8V.

**Note 5:** Output clock frequency > 160MHz and VDDOx < 1.8V.

**Note 6:** Measured differentially.

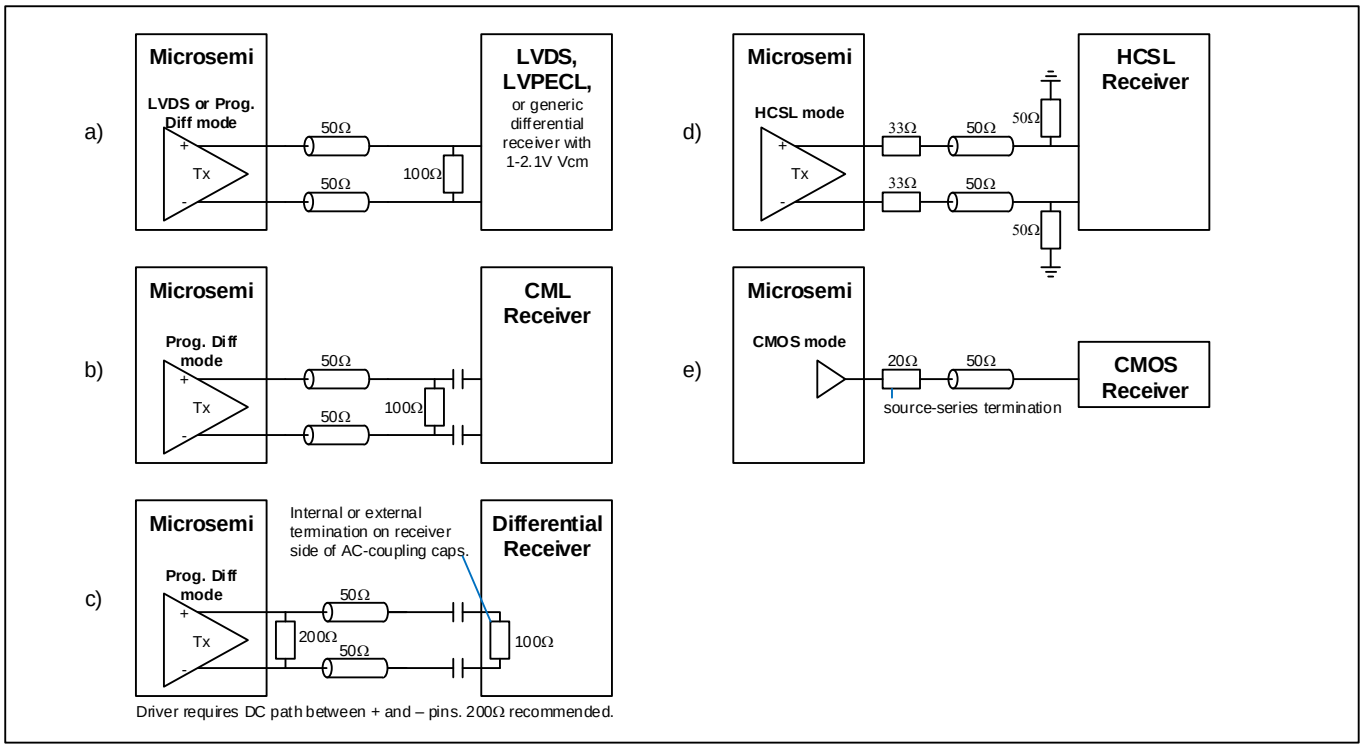



**Figure 17 - Example External Components for Output Signals**

Figure 17 part c) covers the case where an existing receiver has AC coupling caps followed by an internal or external 100Ω termination resistor. The driver requires a DC path between its POS and NEG pins even in this case. A 200Ω resistor is recommended. Since this arrangement attenuates the signal by one third, The OCxDIFF.VOD field should be set 50% higher to compensate.

**Table 14 - Electrical Characteristics: APLL Frequencies**

| Characteristics | Symbol | Min. | Typ. | Max. | Units | Notes |
|---|---|---|---|---|---|---|
| APLL VCO frequency range | $f_{VCO}$ | 3715 | | 4180 | MHz | |
| APLL PFD input frequency | $t_{PFD}$ | 9.72 | | 156.25 | MHz | |

**Table 15 - Electrical Characteristics: Jitter and Skew Specifications**

| Characteristics | Test Conditions | Min | Typ | Max | Units |
|---|---|---|---|---|---|
| APLL Jitter Transfer Bandwidth | | | 600 | | kHz |
| Output-to-Output Skew | Note 4 | | | 100 | ps |
| Input-to-Output Delay Variation, APLL1, external feedback, OC1 internal path | Note 12 | | 110 | 160 | ps |
| Input-to-Output Delay Variation, APLL2, external feedback, OC3 internal path | Note 12 | | 130 | 183 | ps |
| **APLL, Integer Output Divider** | | | | | |
| Phase Jitter, 156.25MHz | Notes 1, 3 | | 0.23 | 0.275 | ps RMS |
| | Notes 2, 3 | | 0.155 | 0.175 | ps RMS |
| PCI Express 1.1, Common Refclk Jitter | Total Jitter, Notes 6, 11 | | 5 | 25 | ps pk-pk |
| PCI Express 2.1, Common Refclk Jitter | 10kHz to 1.5MHz, Note 6 | | 0.1 | 0.3 | ps RMS |
| | 1.5MHz to 50MHz, Note 6 | | 1.6 | 2.1 | ps RMS |
| PCI Express 3.0, Common Refclk Jitter | Note 6 | | 0.08 | 0.10 | ps RMS |
| PCI Express 4.0, Common Refclk Jitter | Note 6 | | 0.08 | 0.10 | ps RMS |
| **APLL, Fractional Output Divider** | | | | | |
| Phase Jitter, 156.25MHz | Integer divisor, Notes 1, 8 | | 0.26 | 0.3 | ps RMS |
| | Frac. divisor, Note 1, 8 | | 0.35 | 0.425 | ps RMS |

| Characteristics | Test Conditions | Min | Typ | Max | Units |
|---|---|---|---|---|---|
| PCI Express 1.1, Common Refclk Jitter | Total Jitter, Notes 9, 11 | | 16 | 27 | ps pk-pk |
| PCI Express 2.1, Common Refclk Jitter | 10kHz to 1.5MHz, Note 9 | | 0.12 | 0.3 | ps RMS |
| | 1.5MHz to 50MHz, Note 9 | | 1.7 | 3.1 | ps RMS |
| PCI Express 3.0, Common Refclk Jitter | Note 8 | | 0.09 | 0.12 | ps RMS |
| PCI Express 4.0, Common Refclk Jitter | Note 8 | | 0.09 | 0.12 | ps RMS |
| Period Jitter, 100MHz | Notes 10, 11 | | 9-20 | 27 | ps pk-pk |
| Cycle-to-Cycle Jitter, 100MHz | Notes 10, 11 | | 8-18 | 26 | ps |
| Random Jitter, fractional output divider | Notes 1, 7 | | 0.26 | 0.3 | ps RMS |
| Deterministic Jitter, fractional output divider | Note 7 | | 0.3 | 6 | ps pk-pk |
| Total Jitter, fractional output divider 12kHz-20MHz | Note 5, 7 | | 3-5 | 10.25 | ps pk-pk |

**Note 1:** Jitter calculated from integrated phase noise from 12kHz to 20MHz. Phase jitter includes spurs (if present). Random jitter does not include spurs.

**Note 2:** Jitter calculated from integrated phase noise from 1.875MHz to 20MHz including spurs (if present).

**Note 3:** With 50MHz crystal doubled as APLL input, APLL VCO frequency 3750MHz, divide by 6 using APLL integer divider, then divide by 4 in output medium-speed divider.

**Note 4:** Requires phase alignment capability described in section 5.6.5. Only applies for outputs that have the same signal format, VDDO voltage. drive strength and loading/termination. Also, this skew spec doesn't apply to OCxN when an output pair is configured with OCxCR3.NEGLSD=1; in this configuration OCxN lags OCxP by up to 1ns.

**Note 5:** Total Jitter = Deterministic Jitter + 14.28 x Random Jitter.

**Note 6:** With 50MHz crystal doubled as APLL input. All output clocks 100MHz HCSL format. Jitter is from the PCIe jitter filter combination that produces the highest jitter.

**Note 7:** Output frequencies ≥ 25MHz.Tested at 156.25MHz.

**Note 8:** With 50MHz crystal doubled as APLL input. Integer divisor case tested with APLL VCO frequency 3750MHz. Fractional divisor case tested with VCO frequency 3993.6MHz. PCI Express 3.0 and 4.0 case tested with APLL VCO frequency 3750MHz.

**Note 9:** With 50MHz crystal doubled as APLL input and APLL VCO frequency 3993.6MHz. All output clocks 100MHz HCSL format. Jitter is from the PCIe jitter filter combination that produces the highest jitter. Applies (a) when spread spectrum modulation is disabled and (b) when spread spectrum modulation in the fractional output divider is 0.25% downspread modulated at 31.5kHz.

**Note 10:** Measured using Tektronix MSO71604C, Mixed Signal Oscilloscope with DPOJET software. Measured with 3750MHz VCO frequency, 200MHz out of the fractional divider, and 100MHz out of the medium-speed divider.

**Note 11:** N=10000

**Note 12:** AFBDL.FBSEL=11. Tested with 125MHz into IC1, APLL VCO frequency of 3750MHz, APLL integer divider set to 5, OCx medium-speed dividers set to 6, OCx 125MHz output frequency. Only applies for outputs that have the same signal format, output divider usage, VDDO voltage, drive strength and load/termination. Also, this delay spec doesn't apply to OCxN when an output pair is configured with OCxCR3.NEGLSD=1; in this configuration OCxN lags OCxP by up to 1ns.

**Table 16 - Electrical Characteristics: Typical Output Phase Jitter from the APLL Integer Divider**

| Output Frequency | Output Jitter, ps RMS 125MHz XO Reference[1] | Output Jitter, ps RMS 50MHz Crystal Reference[2] |
|---|---|---|
| 625MHz | 0.180 | 0.185 |
| 156.25MHz | 0.227 | 0.23 |
| 125MHz | 0.228 | 0.23 |
| 25MHz CMOS | 0.275 | 0.29 |
| 622.08MHz | 0.25 | 0.26 |
| 625MHz * 66/64 | 0.255 | 0.26 |
| 156.25MHz * 66/64 | 0.27 | 0.28 |
| 614.4MHz | 0.26 | 0.27 |
| 153.6MHz | 0.275 | 0.28 |

**Note 1:** APLL locked to external 125MHz XO (Vectron VCC1-1535-125M000).

**Note 2:** APLL locked to external 50MHz crystal (TXC 7M50070021), internal doubler enabled when multiplication is fractional.

**Note 3:** All signals are differential unless otherwise stated. Jitter is integrated 12kHz to 5MHz for 25MHz output frequency and 12kHz to 20MHz for all other output frequencies.

**Table 17 - Electrical Characteristics: Clock Buffer (APLL Bypass Path)**

| Characteristics | | Min. | Typ. | Max. | Units | Notes |
|---|---|---|---|---|---|---|
| Additive Jitter (Note 7) | 100MHz | | 0.198 | | ps RMS | Notes 1, 5 |
| | 125MHz | | 0.174 | | | |
| | 156.25MHz | | 0.155 | 0.175 | | |
| | 200MHz | | 0.141 | | | |
| | 400MHz | | 0.115 | | | |
| | 800MHz | | 0.094 | | | |
| Input-to-Output Propagation Delay, from IC1 or IC2 input | | 2 | 2.3 | 2.6 | ns | Note 3 |
| | | 3 | 3.4 | 3.8 | ns | Note 4 |
| Input-to-Output Propagation Delay, from IC3 input | | 1.9 | 2.3 | 2.6 | ns | Note 3 |
| | | 2.9 | 3.4 | 3.8 | ns | Note 4 |
| Input-to-Output Propagation Delay, from XA input | | 2.9 | 3.3 | 3.7 | ns | Note 3 |
| | | 3.9 | 4.4 | 4.9 | ns | Note 4 |
| Output-to-Output Skew | | | 60 | 100 | ps | Note 2 |
| Output Phase Jitter, 50MHz crystal, 50MHz output | | | 0.29 | | ps RMS | Notes 5, 6 |
| Output Period Jitter, 50MHz crystal, 50MHz output | | | 11 | | ps pk-pk | N=10000, Note 6 |
| Output Cycle-to-Cycle Jitter, 50MHz crystal, 50MHz output | | | 11 | | ps pk | N=10000, Note 6 |

**Note 1:** APLL bypass enabled, input frequency = output frequency, LVPECL output signal format.

**Note 2:** Only applies for outputs that have the same signal format, VDDO voltage, drive strength and load/termination. Also, this skew spec doesn't apply to OCxN when an output pair is configured with OCxCR3NEGLSD=1; in this configuration OCxN lags OCxP by up to 1ns.

**Note 3:** Measured at 125MHz. Differential outputs with 100Ω differential termination (LVDS, LVPECL, Programmable Differential).

**Note 4:** Measured at 125MHz. CMOS/HSTL outputs, 5pF load.

**Note 5:** Jitter calculated from integrated phase noise from 12kHz to 20MHz.

**Note 6:** Tested with 50MHz crystal TXC 7M50070021.

**Note 7:** Additive jitter contributes in a root-of-sum-of-squares manner. For example, a 156.25MHz input signal with 220fs of jitter will experience typical additive jitter of 155fs in the device, and the resulting output jitter will be $\sqrt{220^2+155^2}=269$fs.

**Table 18 - Electrical Characteristics: Typical Input-to-Output Clock Delay Through APLL**

| Mode | Delay, Input Clock Edge to Output Clock Edge |
|---|---|
| All Modes | Non-deterministic but constant as long as the APLL remains locked and output clock phases are not adjusted as described in section 5.6.4. |

**Table 19 - Electrical Characteristics: SPI Slave Interface Timing, Device Registers**

VDDIO = 3.3V±5% or 2.5V±5% or 1.8V±5%

| Characteristics (Notes 1 to 3) | Symbol | VDDIO 3.3V or 2.5V | | | VDDIO 1.8V | | | Units | Notes |
|---|---|---|---|---|---|---|---|---|---|
| | | Min. | Typ. | Max. | Min. | Typ. | Max. | | |
| SCLK frequency | $f_{BUS}$ | | | 23 | | | 15 | MHz | |
| SCLK cycle time | $t_{CYC}$ | 43.5 | | | 66 | | | ns | |
| CSN setup to first SCLK edge | $t_{SUC}$ | 10 | | | 10 | | | ns | |
| CSN hold time after last SCLK edge | $t_{HDC}$ | 10 | | | 10 | | | ns | |
| CSN high time | $t_{CSH}$ | 25 | | | 25 | | | ns | |
| SCLK high time | $t_{CLKH}$ | 10 | | | 33 | | | ns | |
| SCLK low time | $t_{CLKL}$ | 21.75 | | | 33 | | | ns | |
| MOSI data setup time | $t_{SUI}$ | 2 | | | 10 | | | ns | |
| MOSI data hold time | $t_{HDI}$ | 2 | | | 10 | | | ns | |
| MISO enable time from SCLK edge | $t_{EN}$ | 0 | | | 0 | | | ns | |
| MISO disable time from CSN high | $t_{DIS}$ | | | 80 | | | 80 | ns | |
| MISO data valid time | $t_{DV}$ | | | 20.5 | | | 32 | ns | |
| MISO data hold time from SCLK edge | $t_{HDO}$ | 0 | | | 0 | | | ns | |
| CSN, MOSI input rise time, fall time | $t_R$, $t_F$ | | | 10 | | | 10 | ns | |

**Note 1:** All timing is specified with 100pF load on all SPI pins.

**Note 2:** All parameters in this table are guaranteed by design or characterization.

**Note 3:** See timing diagram in Figure 18.

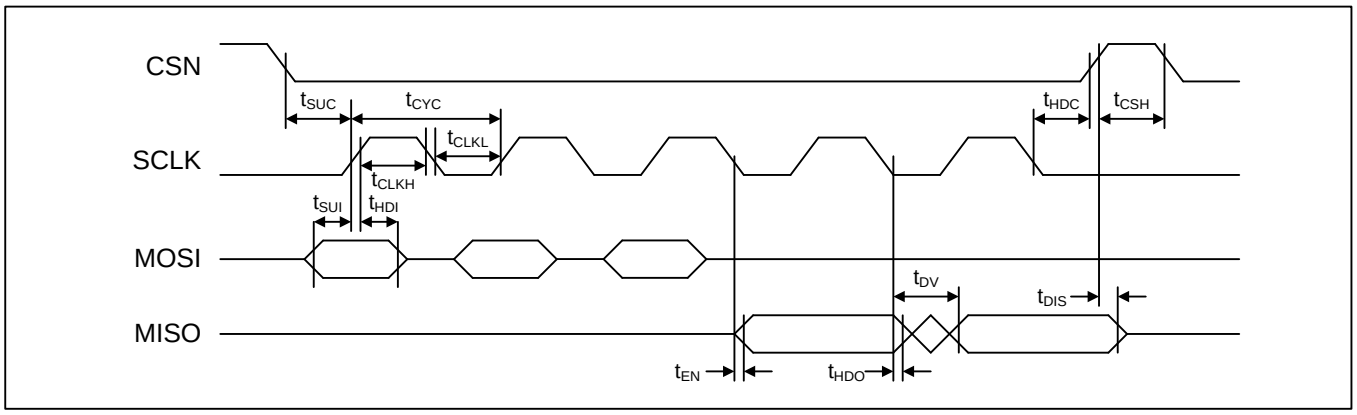



**Figure 18 - SPI Slave Interface Timing**

**Table 20 - Electrical Characteristics: SPI Slave Interface Timing, Internal EEPROM**
(ZL30265 and ZL30267 Only)
VDDIO = 3.3V±5% or 2.5V±5% or 1.8V±5%

| Characteristics (Notes 1 to 4) | Symb | VDDIO 3.3V | | VDDIO 2.5V | | VDDIO 1.8V | | Units |
|---|---|---|---|---|---|---|---|---|
| | | Min | Max | Min | Max | Min | Max | |
| SCLK frequency | $f_{BUS}$ | | 7.7 | | 4.5 | | 4 | MHz |
| SCLK cycle time | $t_{CYC}$ | 130 | | 220 | | 240 | | ns |
| CSN setup to first SCLK edge | $t_{SUC}$ | 50 | | 100 | | 100 | | ns |
| CSN hold time after last SCLK edge | $t_{HDC}$ | 53 | | 103 | | 105 | | ns |
| CSN high time | $t_{CSH}$ | 50 | | 100 | | 100 | | ns |
| SCLK high time | $t_{CLKH}$ | 40 | | 80 | | 80 | | ns |
| SCLK low time | $t_{CLKL}$ | 63 | | 107 | | 120 | | ns |
| MOSI data setup time | $t_{SUI}$ | 11 | | 21 | | 25 | | ns |
| MOSI data hold time | $t_{HDI}$ | 11 | | 21 | | 25 | | ns |
| MISO enable time from SCLK edge | $t_{EN}$ | 0 | | 0 | | 0 | | ns |
| MISO disable time from CSN high | $t_{DIS}$ | | 22 | | 22 | | 25 | ns |
| MISO data valid time | $t_{DV}$ | | 63 | | 107 | | 119 | ns |
| MISO data hold time from SCLK edge | $t_{HDO}$ | 0 | | 0 | | 0 | | ns |
| CSN, MOSI input rise time, fall time | $t_R$, $t_F$ | | 10 | | 10 | | 10 | ns |

**Note 1:** This timing applies (a) when EESEL=1 and (b) in direct EEPROM write mode (see section 5.11.2).

**Note 2:** All timing is specified with 100pF load on all SPI pins.

**Note 3:** All parameters in this table are guaranteed by design or characterization.

**Note 4:** See timing diagram in Figure 18.

**Table 21 - Electrical Characteristics: SPI Master Interface Timing (ZL30264 and ZL30266 Only)**

VDDIO = 3.3V±5% or 2.5V±5% or 1.8V±5%

| Characteristics (Notes 1 to 3) | Symbol | Min. | Typ. | Max. | Units | Notes |
|---|---|---|---|---|---|---|
| SCLK output frequency | $f_{BUS}$ | | | 5 | MHz | |
| SCLK output cycle time | $t_{CYC}$ | 200 | | | ns | |
| SCLK output duty cycle | $t_{CLKH}$/ $t_{CYC}$ | 45 | 50 | 55 | % | |
| CSN output setup to first SCLK rising edge | $t_{SUC}$ | 200 | | | ns | |
| CSN output hold after last SCLK falling edge | $t_{HDC}$ | 200 | | | ns | |
| CSN output high time | $t_{CSH}$ | 200 | | | ns | |
| MISO input setup time to SCLK rising edge | $t_{SU}$ | 15 | | | ns | |
| MISO input hold time from SCLK rising edge | $t_{HD}$ | 5 | | | ns | |
| MOSI output valid from SCLK falling edge | $t_{DV}$ | | | 10 | ns | |
| SCLK, CSN, MOSI output rise time, fall time | $t_R$, $t_F$ | | | 15 | ns | |
| MISO input rise time, fall time | $t_R$, $t_F$ | | | 10 | ns | |

**Note 1:** All timing is specified with 100pF load on all SPI pins.

**Note 2:** All parameters in this table are guaranteed by design or characterization.

**Note 3:** See timing diagram in Figure 19.

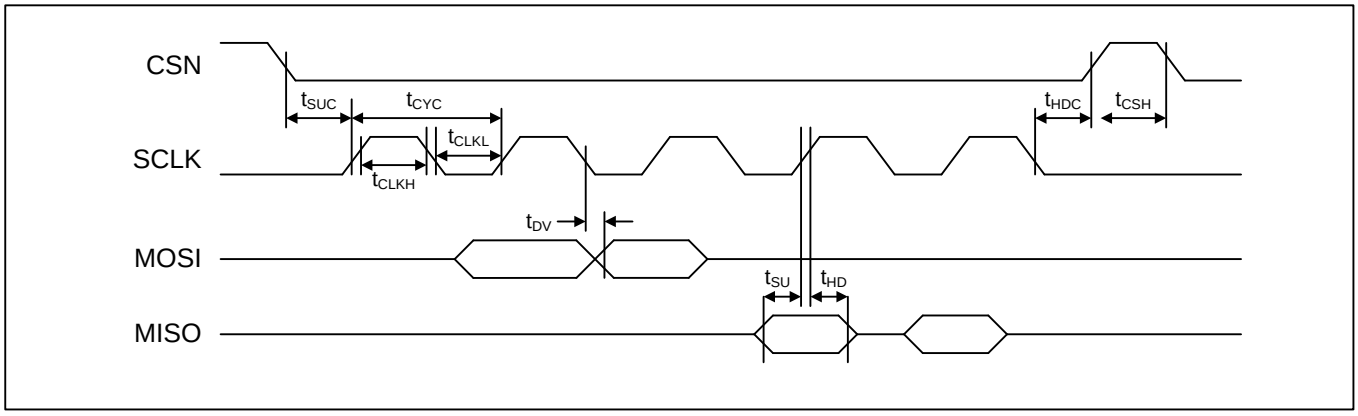



**Figure 19 - SPI Master Interface Timing**

**Table 22 - Electrical Characteristics: I²C Slave Interface Timing**

VDDIO = 3.3V±5% or 2.5V±5% or 1.8V±5%

| Characteristics | Symbol | Min. | Typ. | Max. | Units | Notes |
|---|---|---|---|---|---|---|
| SCL clock frequency | $f_{SCL}$ | | | 400 | kHz | Note 1 |
| Hold time, START condition | $t_{HD:STA}$ | 0.6 | | | µs | |
| Low time, SCL | $t_{LOW}$ | 1.3 | | | µs | |
| High time, SCL | $t_{HIGH}$ | 0.6 | | | µs | |
| Setup time, START condition | $t_{SU:STA}$ | 0.6 | | | µs | |
| Data hold time | $t_{HD:DAT}$ | 0 | | 0.9 | µs | Notes 2 and 3 |
| Data setup time | $t_{SU:DAT}$ | 100 | | | ns | |
| Rise time | $t_R$ | | | | ns | Note 4 |
| Fall time | $t_F$ | $20 + 0.1C_b$ | | 300 | ns | $C_b$ is cap. of one bus line |
| Setup time, STOP condition | $t_{SU:STO}$ | 0.6 | | | µs | |
| Bus free time between STOP/START | $t_{BUF}$ | 1.3 | | | µs | |
| Pulse width of spikes which must be suppressed by the input filter | $t_{SP}$ | 0 | | 50 | ns | |

**Note 1:** The timing parameters in this table are specifically for 400kbps Fast Mode. Fast Mode devices are downward-compatible with 100kbps Standard Mode I²C bus timing. All parameters in this table are guaranteed by design or characterization. All values referred to $V_{IHmin}$ and $V_{ILmax}$ levels (see Table 7).

**Note 2:** The device internally provides a hold time of at least 300ns for the SDA signal (referred to the $V_{IHmin}$ of the SCL signal) to bridge the undefined region of the falling edge of SCL. Other devices must provide this hold time as well per the I²C specification.

**Note 3:** The I²C specification indicates that the maximum $t_{HD:DAT}$ spec only has to be met if the device does not stretch the low period ($t_{LOW}$) of the SCL signal. The device does not stretch the low period of the SCL signal.

**Note 4:** Determined by choice of pull-up resistor.

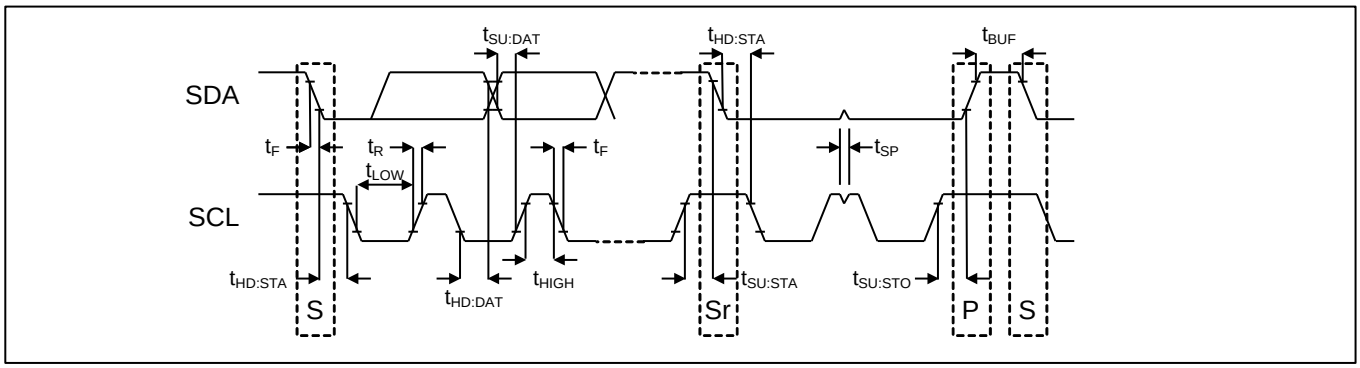



**Figure 20 - I²C Slave Interface Timing**

## 8. Package and Thermal Information

**Table 23 - 8x8mm QFN Package Thermal Properties**

| PARAMETER | SYMBOL | CONDITIONS | VALUE | UNITS |
|---|---|---|---|---|
| Maximum Ambient Temperature | $T_A$ | | 85 | °C |
| Maximum Junction Temperature | $T_{JMAX}$ | | 125 | °C |
| Junction to Ambient Thermal Resistance (Note 1) | $\theta_{JA}$ | still air | 15.1 | °C/W |
| | | 1m/s airflow | 12.4 | |
| | | 2.5m/s airflow | 10.6 | |
| Junction to Board Thermal Resistance | $\theta_{JB}$ | | 3.2 | °C/W |
| Junction to Case Thermal Resistance | $\theta_{JC}$ | | 7.3 | °C/W |
| Junction to Pad Thermal Resistance (Note 2) | $\theta_{JP}$ | Still air | 0.9 | °C/W |
| Junction to Top-Center Thermal Characterization Parameter | $\psi_{JT}$ | Still air | 0.1 | °C/W |

**Note 1:** Theta-JA ($\theta_{JA}$) is the thermal resistance from junction to ambient when the package is mounted on an 8-layer JEDEC standard test board and dissipating maximum power.

**Note 2:** Theta-JP ($\theta_{JP}$) is the thermal resistance from junction to the center exposed pad on the bottom of the package.

**Note 3:** For all numbers in the table, the exposed pad is connected to the ground plane with a 9x9 array of thermal vias; via diameter 0.33mm; via pitch 0.76mm.

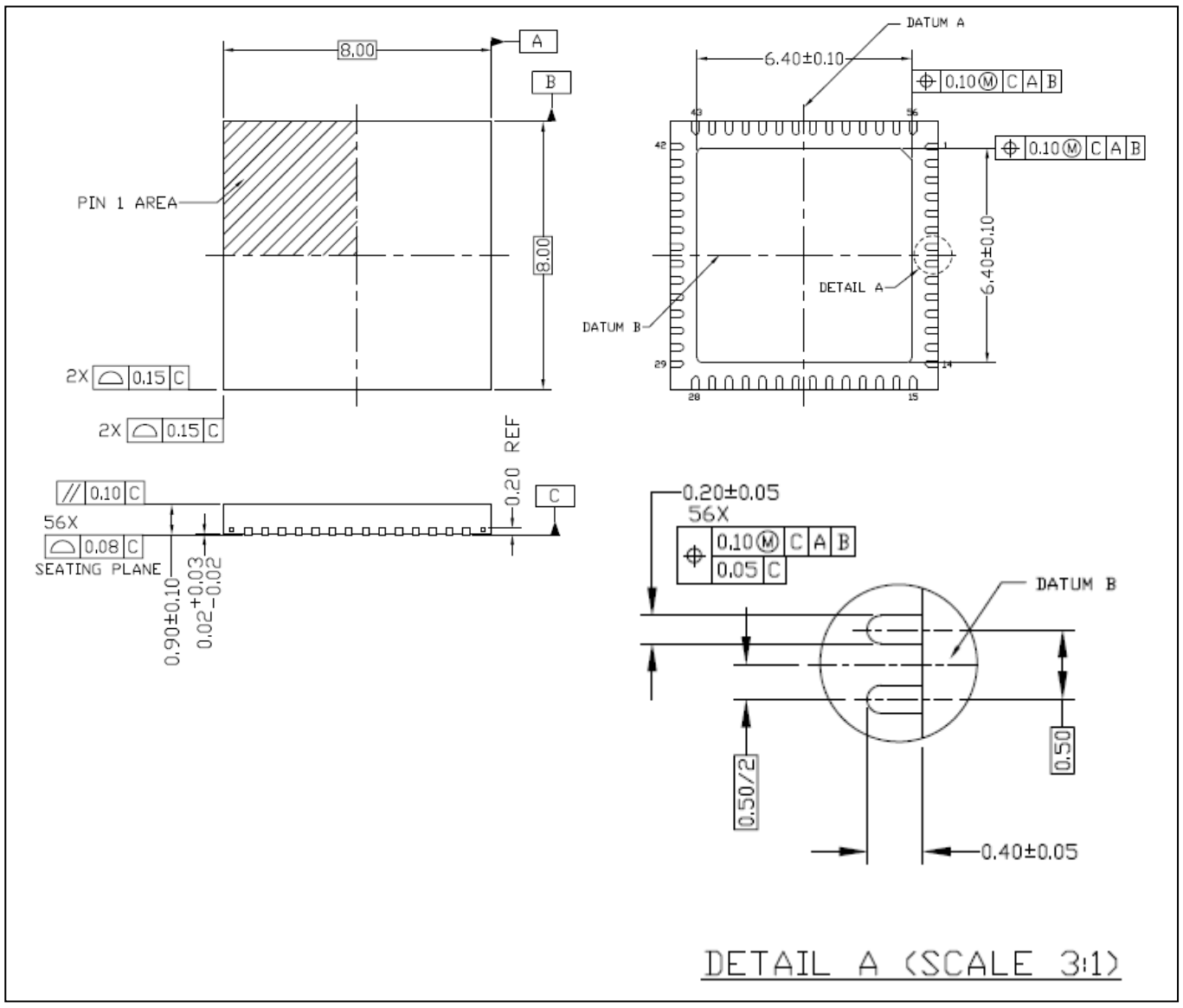
Devices can exceed $T_{JMAX}$ when ambient temperature at or near 85°C, the application has no airflow, the device is powered 3.3V-only (VDDH=VDDL=3.3V), both APLLs are enabled, and most or all other device blocks are enabled. The table below gives some guidance about the tradeoff between number of fractional dividers enabled and number of OCx outputs that can be enabled. See Table 6 for per-block IDD numbers. Devices have no thermal limitations when powered 2.5V-only, 1.8V+2.5V or 1.8V+3.3V.

**Table 24 - Number of OCx Outputs Available, No Airflow, 3.3V-Only Operation**

| Output Type | Output Frequencies ≤200MHz | | | Any Output Frequencies | | |
|---|---|---|---|---|---|---|
| | # Fractional Dividers Enabled | | | # Fractional Dividers Enabled | | |
| | 0 | 1 | 2 | 0 | 1 | 2 |
| LVDS | 10 | 9 | 7 | 8 | 7 | 6 |
| LVPECL | 9 | 8 | 7 | 8 | 7 | 6 |

**Note:** IC1 and IC2 inputs enabled, IC3 and XA disabled. Both APLLs enabled.

## 9. Mechanical Drawing



## 10. Acronyms and Abbreviations

| | |
|---|---|
| APLL | analog phase locked loop |
| CML | current mode logic |
| GbE | gigabit Ethernet |
| HCSL | high-speed current steering logic |
| HSTL | high-speed transceiver logic |
| I/O | input/output |
| LOS | loss of signal |
| LVDS | low-voltage differential signal |
| LVPECL | low-voltage positive emitter-coupled logic |
| PFD | phase/frequency detector |
| PLL | phase locked loop |
| ppb | parts per billion |
| ppm | parts per million |
| pk-pk | peak-to-peak |
| RMS | root-mean-square |
| RO | read-only |
| R/W | read/write |
| SS or SSM | spread spectrum modulation |
| TCXO | temperature-compensated crystal oscillator |
| UI | unit interval |
| $UI_{PP}$ or $UI_{P-P}$ | unit interval, peak to peak |
| XO | crystal oscillator |

## 11. Data Sheet Revision History

| Revision | Description |
|---|---|
| 26-Sep-2016 | First general release |
| 08-Dec-2016 | On page 1 added PCIe bullet.<br>In Table 1, OCx pin description, clarified what is programmable for each mode.<br>Added new Figure 17.<br>In Table 15 added max Input-to-Output Delay Variation numbers for external feedback using on-die paths.<br>In Table 17 added min, typ and max values for buffer Input-to-Output Propagation Delay. In footnotes 3 and 4 added “Measured at 125MHz”.<br>In Figure 17 added CMOS diagram. |
| 12-Jun-2017 | In section 2.4 and section 5.5.6.2 added note that NCO resolution of 1ppt is possible.<br>Corrected references to MCR1.XAB to MCR2.XAB.<br>Corrected four occurrences of VCCOx to VDDOx.<br>Added pullup recommendations to SCL/SCLK and SDA/MOSI pin descriptions.<br>Change ID2.REV default value to “contact factory”.<br>In OCxCR2.OCSF HCSL decode, deleted “VCM and VOD are ignored”. |
| 17-Jul-2017 | In Figure 17 removed the 50ohm to ground termination option from the CMOS diagram. |
| 05-Sep-2017 | In Table 1 TEST/GPIO3 pin description, changed “TEST must be low on the rising edge of RSTN” to “Typically TEST should be low on the rising edge of RSTN, but see section 5.2 for some options.”<br>In section 5.6.2 added the statement that maximum input frequency to the medium-speed divider is 750MHz. |

| Revision | Description |
|---|---|
| | In section 5.6.5 clarified that signals from the same IntDiv or FracDiv are aligned and that aligning clocks from IntDiv with clocks from FracDiv is not supported.<br>In section 5.9 added new subsection 5.9.1 to guide users in the use of external RC reset circuits. |
| 19-Jan-2018 | In the F1REM1 register description deleted “The value FREM=0 is undefined. When FBP=0, FREM must be set to 0.”<br>In the F1DEN1 register description deleted “When FBP=0, FDEN must be set to 1.”<br>In the F1BP register description deleted “When FBP=0, FREM must be set to 0 and FDEN must be set to 1.”<br>In the AFBREM1 register description deleted “When AFBBP=0, AFBREM must be set to 0.”<br>IIn the AFBBP register description deleted “When AFBBP=0, AFBREM must be set to 0 and AFBDEN must be set to 1.” |
| 29-May-2018 | On page 1 and in section 2.3 changed wording to indicate PCIe 1-4 compliance.<br>In OCxCR3.NEGLSD description added to note that OCxCR3.LSSEL must be 0 to set NEGLSD=1. |
| 14-Sep-2018 | In Table 17, corrected typo for "Input-to-Output Propagation Delay, from IC3 input" in the Note 4 row where the max of 2.8 was less than typical of 3.4. Corrected the max to 3.8.<br>In the OCxCR2.OCSF description, deleted "(must have VDDOx=VDDH=3.3V)" from the HCSL decode.<br>Updated Table 10 Note 1 and Table 11 Note 1 and added cross reference to Figure 17 parts a) and b).<br>Updated Figure 17 to include new part c) and added noted about part c) below the figure.<br>In section 5.9.1 second paragraph added need for current-limiting series resistor between source of reset signal and device RSTN pin.<br>In Table 1 pin descriptions for AC0/GPIO0, AC1/GPIO1, AC2/GPIO2 and TEST/GPIO3 added note. |
| 01-Oct-2018 | In Table 15:<br>• Reduced integer output divider PCIe 3.0 jitter to 0.08 typical, 0.10 max (ps RMS) and added new row for PCIe 4.0 jitter.<br>• Reduced fractional output divider PCIe 3.0 jitter to 0.09 typical, 0.12 max (ps RMS) and added new row for PCIe 4.0 jitter.<br>• Added PCIe sentence to Note 8 and deleted second sentence of Note 6. |

Microsemi Corporate Headquarters
One Enterprise
Aliso Viejo, CA 92656 USA

**Within the USA**: +1 (800) 713-4113
**Outside the USA**: +1 (949) 380-6100
**Sales**: +1 (949) 380-6136
**Fax**: +1 (949) 215-4996

**E-mail**: sales.support@microsemi.com



Microsemi Corporation (Nasdaq: MSCC) offers a comprehensive portfolio of semiconductor and system solutions for communications, defense & security, aerospace and industrial markets. Products include high-performance and radiation-hardened analog mixed-signal integrated circuits, FPGAs, SoCs and ASICs; power management products; timing and synchronization devices and precise time solutions, setting the world's standard for time; voice processing devices; RF solutions; discrete components; security technologies and scalable anti-tamper products; Power-over-Ethernet ICs and midspans; as well as custom design capabilities and services. Microsemi is headquartered in Aliso Viejo, Calif., and has approximately 3,400 employees globally. Learn more at **www.microsemi.com**.

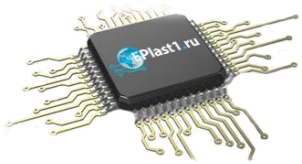
Компания «ЭлектроПласт» предлагает заключение долгосрочных отношений при поставках импортных электронных компонентов на взаимовыгодных условиях!

Наши преимущества:

- Оперативные поставки широкого спектра электронных компонентов отечественного и импортного производства напрямую от производителей и с крупнейших мировых складов;
- Поставка более 17-ти миллионов наименований электронных компонентов;
- Поставка сложных, дефицитных, либо снятых с производства позиций;
- Оперативные сроки поставки под заказ (от 5 рабочих дней);
- Экспресс доставка в любую точку России;
- Техническая поддержка проекта, помощь в подборе аналогов, поставка прототипов;
- Система менеджмента качества сертифицирована по Международному стандарту ISO 9001;
- Лицензия ФСБ на осуществление работ с использованием сведений, составляющих государственную тайну;
- Поставка специализированных компонентов (Xilinx, Altera, Analog Devices, Intersil, Interpoint, Microsemi, Aeroflex, Peregrine, Syfer, Eurofarad, Texas Instrument, Miteq, Cobham, E2V, MA-COM, Hittite, Mini-Circuits,General Dynamics и др.);

Помимо этого, одним из направлений компании «ЭлектроПласт» является направление «Источники питания». Мы предлагаем Вам помощь Конструкторского отдела:

- Подбор оптимального решения, техническое обоснование при выборе компонента;
- Подбор аналогов;
- Консультации по применению компонента;
- Поставка образцов и прототипов;
- Техническая поддержка проекта;
- Защита от снятия компонента с производства.




**Как с нами связаться**

**Телефон:** 8 (812) 309 58 32 (многоканальный)
**Факс:** 8 (812) 320-02-42
**Электронная почта:** org@eplast1.ru
**Адрес:** 198099, г. Санкт-Петербург, ул. Калинина, дом 2, корпус 4, литера А.