



# **MAX24605, MAX24610**

## **5- or 10-Output Any-to-Any Clock Multiplier / Jitter Attenuator ICs**

### **General Description**

The MAX24605 and MAX24610 are flexible, high-performance clock multiplier and jitter attenuator ICs that include a DPLL and two independent APLLs. When locked to one of two input clock signals, the device performs any-to-any frequency conversion. From any input clock frequency 2kHz to 750MHz the device can produce frequency-locked APLL output frequencies up to 750MHz and as many as 10 output clock signals that are integer divisors of the APLL frequencies. Input jitter can be attenuated by an internal low-bandwidth DPLL. The DPLL also provides glitchless switching between input clocks and numerically controlled oscillator capability. Input switching can be manual or automatic. Using only a low-cost crystal or oscillator, the device can also serve as a frequency synthesizer IC. Output jitter is typically 0.18 to 0.3ps RMS for an APLL-only integer multiply and 0.25 to 0.4ps RMS for APLL-only fractional multiply or DPLL+APLL operation.

### **Applications**

Jitter attenuation, frequency conversion and frequency synthesis applications in a wide variety of equipment types

### **Ordering Information**

PART	OUTPUTS	TEMP RANGE	PIN-PACKAGE
MAX24605EXG+	5	-40 to +85	81-CSBGA
MAX24610EXG+	10	-40 to +85	81-CSBGA

+Denotes a lead(Pb)-free/RoHS-compliant package.

**Block Diagram appears on page 6.**  
**Register Map appears on page 37.**

### **Features**

- ◆ **Input Clocks**
  - ◆ One Crystal Input
  - ◆ Two Differential or CMOS/TTL Inputs
  - ◆ Differential to 750MHz, CMOS/TTL to 160MHz
  - ◆ Continuous Input Clock Quality Monitoring
  - ◆ Automatic or Manual Clock Selection
  - ◆ Glitchless Reference Switching
- ◆ **Low-Bandwidth DPLL**
  - ◆ Programmable Bandwidth, 4Hz to 400Hz
  - ◆ Attenuates Input Jitter up to Several UI
  - ◆ Manual Phase Adjustment
- ◆ **Two APLLs Plus 5 or 10 Output Clocks**
  - ◆ APLLs Perform High Resolution Fractional-N Clock Multiplication
  - ◆ Any Output Frequency from <1Hz to 750MHz
  - ◆ Each Output Has an Independent Divider
  - ◆ Output Jitter Typically 0.18 to 0.3ps RMS for APLL-Only Integer Multiply and 0.25 to 0.4ps RMS for Other Modes (12kHz to 20MHz)
  - ◆ Outputs are CML or 2xCMOS, Can Interface to LVDS, LVPECL, HSTL, SSTL and HCSL
  - ◆ CMOS Output Voltage from 1.5V to 3.3V
- ◆ **General Features**
  - ◆ Automatic Self-Configuration at Power-Up from External EEPROM Memory
  - ◆ Uses External Crystal, Oscillator or Clock Signal As Master Clock
  - ◆ Internal Compensation for Local Oscillator Frequency Error
  - ◆ SPI Processor Interface
  - ◆ 1.8V + 3.3V Operation (5V Tolerant)
  - ◆ -40 to +85°C Operating Temp. Range
  - ◆ 10mm x 10mm CSBGA Package

**Table of Contents**

<b>1.</b>	<b>APPLICATION EXAMPLES .....</b>	<b>6</b>
<b>2.</b>	<b>BLOCK DIAGRAM .....</b>	<b>6</b>
<b>3.</b>	<b>DETAILED FEATURES.....</b>	<b>6</b>
3.1	INPUT BLOCK FEATURES.....	6
3.2	DPLL FEATURES.....	7
3.3	APLL FEATURES.....	7
3.4	OUTPUT CLOCK FEATURES.....	7
3.5	GENERAL FEATURES .....	7
<b>4.</b>	<b>PIN DESCRIPTIONS.....</b>	<b>8</b>
<b>5.</b>	<b>FUNCTIONAL DESCRIPTION .....</b>	<b>11</b>
5.1	DEVICE IDENTIFICATION AND PROTECTION .....	11
5.2	TOP-LEVEL CONFIGURATION.....	11
5.2.1	APLL-Only Mode.....	11
5.2.2	DPLL+APLL Mode .....	12
5.3	LOCAL OSCILLATOR AND MASTER CLOCK CONFIGURATION.....	14
5.3.1	External Oscillator .....	14
5.3.1.1	Oscillator Characteristics to Minimize Output Jitter .....	14
5.3.2	On-Chip Crystal Oscillator .....	15
5.3.3	Master Clock APLL Configuration.....	16
5.4	INPUT SIGNAL FORMAT CONFIGURATION.....	17
5.5	INPUT CLOCK DIVIDER, MONITOR AND SELECTOR .....	17
5.5.1	Input Clock Frequency Dividers, Scaling and Inversion .....	18
5.5.2	Input Clock Monitoring .....	18
5.5.2.1	Frequency Monitoring .....	18
5.5.2.2	Activity Monitoring.....	19
5.5.2.3	Selected Reference Fast Activity Monitoring .....	20
5.5.2.4	External Monitoring .....	20
5.5.3	Input Clock Priority, Selection and Switching .....	20
5.5.3.1	Priority Configuration .....	20
5.5.3.2	Automatic Selection .....	21
5.5.3.3	Forced Selection .....	21
5.5.3.4	Ultra-Fast Reference Switching .....	21
5.5.3.5	External Reference Switching Mode .....	21
5.5.3.6	Output Clock Phase Continuity During Reference Switching.....	22
5.6	DPLL ARCHITECTURE AND CONFIGURATION .....	22
5.6.1	DPLL State Machine .....	22
5.6.1.1	Free-Run State .....	23
5.6.1.2	Prelocked State.....	24
5.6.1.3	Locked State .....	24
5.6.1.4	Loss-of-Lock State.....	24
5.6.1.5	Prelocked 2 State .....	25
5.6.1.6	Digital Hold State .....	25
5.6.2	Bandwidth .....	25
5.6.3	Damping Factor.....	25
5.6.4	Phase Detectors.....	26
5.6.5	Loss of Phase Lock Detection .....	26
5.6.6	Manual Phase Adjustment.....	27
5.6.7	Frequency and Phase Measurement.....	27
5.6.8	Input Wander and Jitter Tolerance.....	27
5.6.9	Jitter and Wander Transfer .....	28

5.6.10	Output Jitter and Wander .....	28
5.6.11	$\pm 160$ ppm Tracking Range Mode .....	28
5.7	APLL CONFIGURATION .....	28
5.7.1	Input Selection and Frequency .....	28
5.7.1.1	APLL-Only Mode .....	28
5.7.1.2	DPLL+APLL Mode .....	29
5.7.2	Output Frequency .....	29
5.8	OUTPUT CLOCK CONFIGURATION .....	30
5.8.1	Enable, Signal Format, Voltage and Interfacing .....	30
5.8.2	Frequency Configuration .....	30
5.8.3	Phase Adjustment .....	31
5.9	MICROPROCESSOR INTERFACE .....	32
5.10	RESET LOGIC .....	34
5.11	POWER-SUPPLY CONSIDERATIONS .....	34
5.12	INITIALIZATION AND EEPROM CONFIGURATION MEMORY .....	34
<b>6.</b>	<b>REGISTER DESCRIPTIONS .....</b>	<b>35</b>
6.1	REGISTER TYPES .....	35
6.1.1	Status Bits .....	35
6.1.2	Configuration Fields .....	35
6.1.3	Bank-Switched Registers .....	35
6.1.4	Multiregister Fields .....	35
6.1.5	Input Clock Registers and DPLL Registers .....	36
6.2	REGISTER MAP .....	37
6.3	REGISTER DEFINITIONS .....	39
6.3.1	Global Registers .....	39
6.3.2	GPIO Registers .....	44
6.3.3	APLL Registers .....	47
6.3.4	Output Clock Registers .....	53
6.3.5	Input Clock Registers .....	57
6.3.6	DPLL Registers .....	68
6.3.7	DPLL and Input Block Status Registers .....	88
<b>7.</b>	<b>JTAG AND BOUNDARY SCAN .....</b>	<b>92</b>
7.1	JTAG DESCRIPTION .....	92
7.2	JTAG TAP CONTROLLER STATE MACHINE DESCRIPTION .....	93
7.3	JTAG INSTRUCTION REGISTER AND INSTRUCTIONS .....	95
7.4	JTAG TEST REGISTERS .....	96
<b>8.</b>	<b>ELECTRICAL CHARACTERISTICS .....</b>	<b>97</b>
<b>9.</b>	<b>PIN ASSIGNMENTS .....</b>	<b>107</b>
9.1	MAX24605 PIN ASSSIGNMENT .....	107
9.2	MAX24610 PIN ASSSIGNMENT .....	109
<b>10.</b>	<b>PACKAGE AND THERMAL INFORMATION .....</b>	<b>111</b>
10.1	PACKAGE TOP MARK FORMAT .....	111
10.2	THERMAL SPECIFICATIONS .....	112
<b>11.</b>	<b>ACRONYMS AND ABBREVIATIONS .....</b>	<b>113</b>
<b>12.</b>	<b>DATA SHEET REVISION HISTORY .....</b>	<b>114</b>

## **List of Figures**

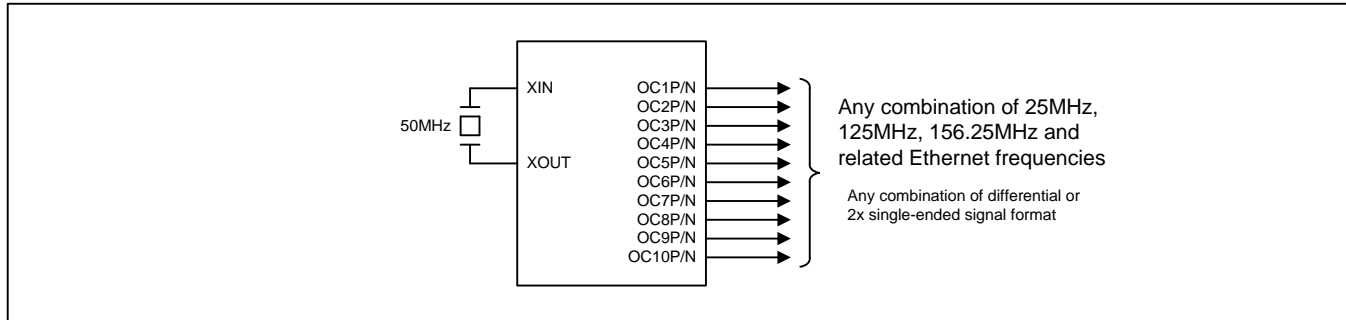
Figure 1-1. Frequency Multiplication and Fanout of Ethernet Clocks.....	6
Figure 2-1. Block Diagram .....	6
Figure 5-1. APLL-Only Mode: Clock Synthesis from a Crystal.....	11
Figure 5-2. APLL-Only Mode: Locked to One of Four Input Clocks .....	12
Figure 5-3. DPLL+APLL Mode: Method 1, Master Clock from High-Speed External Oscillator .....	13
Figure 5-4. DPLL+APLL Mode: Method 2a, Master Clock from Crystal Oscillator Multiplied by APLL2 .....	13
Figure 5-5. DPLL+APLL Mode: Method 2b, Master Clock from External Oscillator Multiplied by APLL2 .....	14
Figure 5-6. Crystal Equivalent Circuit / Crystal and Capacitor Connections .....	15
Figure 5-7. Input block Diagram .....	17
Figure 5-8. DPLL Block Diagram .....	22
Figure 5-9. DPLL State Transition Diagram .....	23
Figure 5-10. APLL Block Diagram .....	29
Figure 5-11. SPI Read Transaction Functional Timing.....	33
Figure 5-12. SPI Write Enable Transaction Functional Timing .....	33
Figure 5-13. SPI Write Transaction Functional Timing.....	33
Figure 7-1. JTAG Block Diagram.....	92
Figure 7-2. JTAG TAP Controller State Machine .....	94
Figure 8-1. Recommended External Components for Interfacing to Differential Inputs.....	99
Figure 8-2. Recommended External Components for Interfacing to CML Outputs.....	101
Figure 8-3. Recommended Configuration for Interfacing to HCSL Components .....	102
Figure 8-4. SPI Interface Timing Diagram .....	105
Figure 8-5. JTAG Timing Diagram.....	106
Figure 9-1. MAX24605 Pin Assignment Diagram.....	108
Figure 9-2. MAX24610 Pin Assignment Diagram.....	110
Figure 10-1. Device Top Mark .....	111

**List of Tables**

Table 4-1. Input Clock Pin Descriptions .....	8
Table 4-2. Output Clock Pin Descriptions .....	8
Table 4-3. Global Pin Descriptions .....	8
Table 4-4. SPI Interface Pin Descriptions .....	9
Table 4-5. External EEPROM SPI Interface Pin Descriptions .....	9
Table 4-6. JTAG Interface Pin Descriptions .....	9
Table 4-7. Power-Supply Pin Descriptions .....	10
Table 5-1. Crystal Selection Parameters .....	15
Table 5-2. Example Master Clock APLL Input Frequencies and Configurations .....	16
Table 5-3. Input Clock Capabilities .....	17
Table 5-4. Activity Monitoring, Missing Clock Cycles vs. Frequency .....	19
Table 5-5. Default Input Clock Priorities .....	20
Table 5-6. Damping Factors and Peak Jitter/Wander Gain .....	25
Table 6-1. Register Map .....	37
Table 7-1. JTAG Instruction Codes .....	95
Table 7-2. JTAG ID Code .....	96
Table 8-1. Recommended DC Operating Conditions .....	97
Table 8-2. Electrical Characteristics: Supply Currents .....	97
Table 8-3. Electrical Characteristics: Non-Clock CMOS/TTL Pins .....	98
Table 8-4. Electrical Characteristics: Clock Inputs .....	99
Table 8-5. Electrical Characteristics: CML Clock Outputs .....	100
Table 8-6. Electrical Characteristics: CMOS and HSTL (Class I) Clock Outputs .....	101
Table 8-7. Electrical Characteristics: Clock Output Timing .....	102
Table 8-8. Electrical Characteristics: Jitter Specifications .....	102
Table 8-9. Electrical Characteristics: Typical Output Jitter Performance, APLL Only .....	103
Table 8-10. Electrical Characteristics: Typical Output Jitter Performance, DPLL+APLL .....	103
Table 8-11. Electrical Characteristics: Typical Input-to-Output Clock Delay .....	104
Table 8-12. Electrical Characteristics: Typical Output-to-Output Clock Delay .....	104
Table 8-13. Electrical Characteristics: SPI Interface Timing .....	105
Table 8-14. Electrical Characteristics: External EEPROM SPI Interface Timing .....	105
Table 8-15. Electrical Characteristics: JTAG Interface Timing .....	106
Table 9-1. MAX24605 Pin Assignments Sorted by Signal Name .....	107
Table 9-2. MAX24610 Pin Assignments Sorted by Signal Name .....	109
Table 10-1. Package Top Mark Legend .....	111
Table 10-1. CSBGA Package Thermal Properties .....	112

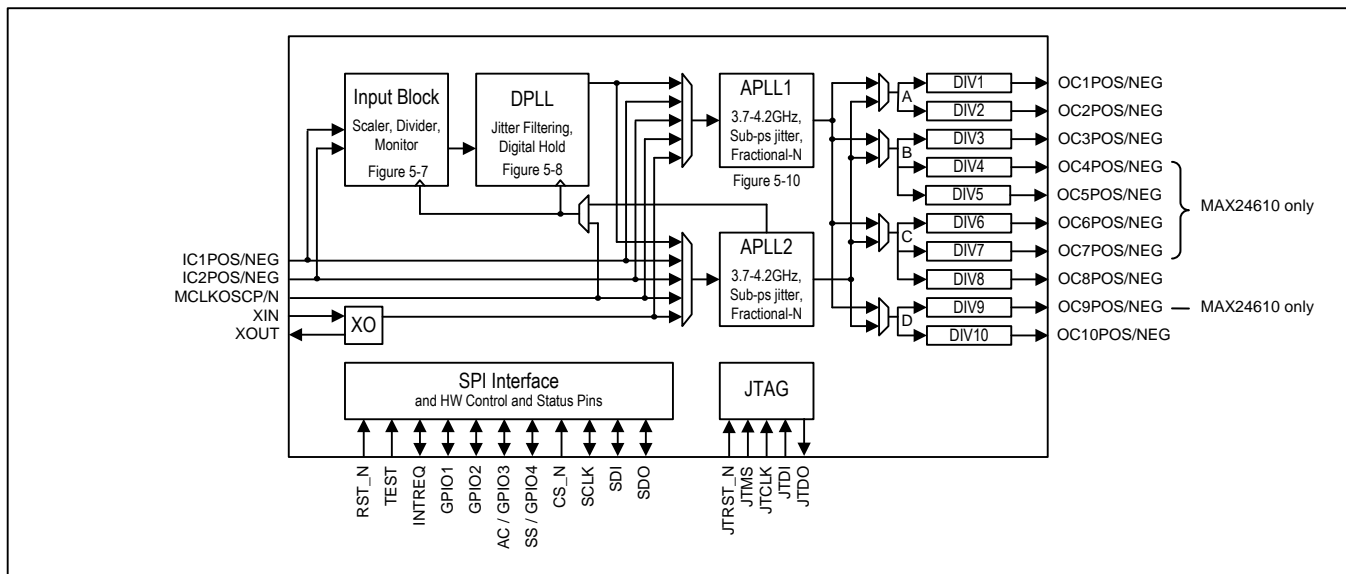
## 1. Application Examples

Figure 1-1. Frequency Multiplication and Fanout of Ethernet Clocks



## 2. Block Diagram

Figure 2-1. Block Diagram



## 3. Detailed Features

### 3.1 Input Block Features

- Two input clocks, differential or CMOS/TTL signal format
- Input clocks can be any frequency from 2kHz up to 750MHz
- Per-input fractional scaling (i.e. multiplying by  $N/D$  where  $N$  is a 16-bit integer and  $D$  is a 32-bit integer and  $N < D$ ) to undo 64B/66B and FEC scaling (e.g. 64/66, 238/255, 237/255, 236/255)
- All inputs constantly monitored by programmable activity monitors and frequency monitors
- Fast activity monitor can disqualify the selected reference after a few missing clock cycles
- Frequency measurement with 1.25ppm resolution
- Frequency monitor thresholds with 1.25ppm or 5ppb resolution

### **3.2 DPLL Features**

- Very high-resolution DPLL architecture
- Sophisticated state machine automatically transitions between free-run, locked, and digital hold states
- Revertive or nonrevertive reference selection algorithm
- Programmable bandwidth from 4Hz to 400Hz
- Separately configurable acquisition bandwidth and locked bandwidth
- Programmable damping factor to balance lock time with peaking: 1.2, 2.5, 5, 10 or 20
- Multiple phase detectors: phase/frequency and multicycle
- Phase/frequency locking ( $\pm 360^\circ$  capture) or nearest-edge phase locking ( $\pm 180^\circ$  capture)
- Multicycle phase detection and locking (up to  $\pm 8191$ UI) improves jitter tolerance and lock time
- Output phase adjustment up to  $\pm 200$ ns in 6ps steps with respect to selected input reference
- High-resolution frequency and phase measurement
- Numerically controlled oscillator (NCO) mode allows system software to steer DPLL frequency

### **3.3 APLL Features**

- Two independent APLLs simultaneously product two frequency families from the same reference clock or different reference clocks
- Very high-resolution fractional scaling (i.e. non-integer multiplication)
- Output jitter is typically 0.18 to 0.3ps RMS for APLL-only integer multiply and 0.25 to 0.4ps RMS for APLL-only fractional multiply or DPLL+APLL operation (12kHz to 20MHz integration band, for output frequencies  $>100$ MHz)
- Bypass mode for each APLL supports system testing and allows device to be used in fanout applications

### **3.4 Output Clock Features**

- Ten low-jitter output clocks
- Each output can be one differential output or two CMOS/TTL outputs
- Outputs easily interface with CML, LVDS, LVPECL, HSTL, SSTL, HCSL components
- Each output can be any integer divisor of either APLL output clock
- Can produce clock frequencies for microprocessors, ASICs, FPGAs and other components
- Can produce PCIe-compliant output clocks (PCIe gen. 1, 2 and 3)
- Per-output delay adjustment
- Per-output enable/disable

### **3.5 General Features**

- SPI serial microprocessor interface
- Optional automatic self-configuration at power-up from external EEPROM memory
- Four general-purpose I/O pins
- Register set can be write-protected
- Can operate as DPLL+APLL for jitter filtering or as APLL only
- Local oscillator can be nearly any frequency from 10MHz to 750MHz
- Internal compensation for local oscillator frequency error

## 4. Pin Descriptions

**Table 4-1. Input Clock Pin Descriptions**

PIN NAME	TYPE <sup>(1)</sup>	PIN DESCRIPTION
IC1POS, IC1NEG	I <sub>DIFF</sub>	<b>Input Clocks 1 and 2.</b> Differential or CMOS/TTL signal format. Programmable frequency. <i>Differential:</i> See <a href="#">Table 8-4</a> for electrical specifications, and see <a href="#">Figure 8-1</a> for recommended external circuitry for interfacing these differential inputs to LVDS, LVPECL or CML output pins on other devices. <i>CMOS/TTL:</i> Connect the single-ended signal to the POS pin. Connect the NEG pin to a capacitor (0.1µF or 0.01µF) to VSS_IO. As shown in <a href="#">Figure 8-1</a> , the NEG pin is internally biased to approximately 1.2V. Treat the NEG pin as a sensitive node; minimize stubs; do not connect to anything else including other NEG pins. <i>Unused:</i> The POS and NEG pins can be left floating. Set ICCR1.ICEN=0.
IC2POS, IC2NEG		
XIN	I	<b>Crystal Oscillator Input.</b> An on-chip XO circuit is designed to work with an external crystal connected to the XIN and XOUT pins. See section 5.3.2 for crystal characteristics and recommended external components. Alternately, the on-chip XO circuit can be disabled, and XIN can be used as a single-ended input clock pin that can accept a clock signal amplitude from 1.8V to 3.3V.
XOUT	O	<b>Crystal Oscillator Output.</b> See section 5.3.2 for crystal characteristics and recommended external components.
MCLKOSCP, MCLKOSCN	I <sub>DIFF</sub>	<b>Master Clock Oscillator.</b> These pins can be used to connect the device to a local oscillator (XO, TCXO, OCXO). The oscillator can be any of a range of frequencies. See section 5.3. <i>Differential:</i> See <a href="#">Table 8-4</a> for electrical specifications, and see <a href="#">Figure 8-1</a> for recommended external circuitry for interfacing these differential inputs to LVDS, LVPECL or CML output pins on other devices. <i>CMOS/TTL:</i> Connect the single-ended signal to the MCLKOSCP pin. Connect the MCLKOSCN pin to a capacitor (0.1µF or 0.01µF) to VSS_IO. As shown in <a href="#">Figure 8-1</a> , the MCLKOSCN pin is internally biased to approximately 1.2V. Treat MCLKOSCN as a sensitive node; minimize stubs; do not connect to anything else.

**Table 4-2. Output Clock Pin Descriptions**

PIN NAME	TYPE <sup>(1)</sup>	PIN DESCRIPTION
OC1POS, OC1NEG OC2POS, OC2NEG OC3POS, OC3NEG OC4POS, OC4NEG OC5POS, OC5NEG OC6POS, OC6NEG OC7POS, OC7NEG OC8POS, OC8NEG OC9POS, OC9NEG OC10POS, OC10NEG	O <sub>DIFF</sub>	<b>Differential Output Clocks 1 through 10.</b> CML, HSTL or 1 or 2 CMOS. Programmable frequency. See <a href="#">Table 8-5</a> and <a href="#">Figure 8-2</a> for electrical specifications and recommended external circuitry for interfacing to LVDS, LVPECL or CML input pins on other devices. See <a href="#">Table 8-6</a> for electrical specifications for interfacing to CMOS and HSTL inputs on other devices. See <a href="#">Figure 8-3</a> for recommended external circuitry for interfacing to HCSL inputs on other devices.

**Table 4-3. Global Pin Descriptions**

PIN NAME	TYPE <sup>(1)</sup>	PIN DESCRIPTION
RST_N	I <sub>PU</sub>	<b>Reset (Active Low).</b> 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 RST_N is low. RST_N should be held low for at least 100ns.
TEST	I <sub>PD</sub>	<b>Factory Test Mode Select.</b> Wire this pin to VSS for normal operation.
GPIO1	I/O <sub>PU</sub>	<b>General-Purpose I/O Pin 1.</b> <a href="#">GPCR.GPIO1C</a> configures this pin. Its state is indicated in <a href="#">GPSR.GPIO1</a> .



PIN NAME	TYPE <sup>(1)</sup>	PIN DESCRIPTION
GPIO2	I/O <sub>PD</sub>	<b>General-Purpose I/O Pin 2.</b> GPCR.GPIO2C configures this pin. Its state is indicated in GPSR.GPIO2.
AC / GPIO3	I/O <sub>PU</sub>	<b>Auto Configuration / General-Purpose I/O Pin 3.</b> If this pin is high when RST_N goes high the device automatically configures its registers based on the configuration script stored in external EEPROM memory. See section 5.12. After reset GPCR.GPIO3C configures this pin. Its state is indicated in GPSR.GPIO3.
SS / GPIO4	I/O <sub>PD</sub>	<b>Source Switch / General-Purpose I/O Pin 4.</b> When DPLLCR1.EXTSW=1 this pin behaves as SS, the source-switching control input for the input block and DPLL (see section 5.5.3.5). When APLLCR2.EXTSW=1 this pin behaves as SS, the sources-switching control input for one or both APLLs. When DPLLCR1.EXTSW=0 and APLLCR2.EXTSW=0 this pin behaves as GPIO4, it is configured by GPCR.GPIO4C, and its state is indicated in GPSR.GPIO4.

**Table 4-4. SPI Interface Pin Descriptions**

See section 5.9 for functional description and Table 8-13 for timing specifications.

PIN NAME	TYPE <sup>(1)</sup>	PIN DESCRIPTION
CS_N	I	<b>Chip Select.</b> The CS_N, SCLK, SDI and SDO pins together are a SPI slave port through which an external SPI master can communicate with the device. This pin must be asserted (low) to read or write internal registers.
SCLK	I	<b>Serial Clock.</b> SCLK is always driven by the SPI bus master.
SDI	I	<b>Serial Data Input.</b> The SPI bus master transmits data to the device on this pin.
SDO	O <sub>3</sub>	<b>Serial Data Output.</b> The device transmits data to the SPI bus master on this pin.

**Table 4-5. External EEPROM SPI Interface Pin Descriptions**

See section 5.12 for functional description and Table 8-14 for timing specifications.

ECS_N	O <sub>3</sub>	<b>External EEPROM Chip Select.</b> The ECS_N, ESCLK, ESDI and ESDO pins together are a SPI master port which can be connected to an external SPI EEPROM device. The device can automatically self-configure from data in the EEPROM at power-up and reset. When the device is reading configuration information from the EEPROM it asserts ECS_N. When the device's SPI master accesses the EEPROM (i.e. when the <b>EESEL</b> bit is set to 1 and CS_N is asserted), ECS_N is a buffered (delayed) version of CS_N.
ESCLK	O <sub>3</sub>	<b>External EEPROM Serial Clock.</b> This pin can be connected to the SCLK pin of an external SPI EEPROM. When the device is reading configuration information from the EEPROM it drives a clock signal on ESCLK. When the device's SPI master accesses the EEPROM (i.e. when the <b>EESEL</b> bit is set to 1 and CS_N is asserted), ESCLK is a buffered (delayed) version of SCLK.
ESDI	O <sub>3</sub>	<b>External EEPROM Serial Data Input.</b> This pin can be connected to the serial data input pin of an external SPI EEPROM. When the device is reading configuration information from the EEPROM it controls ESDI as needed. When the device's SPI master accesses the EEPROM (i.e. when the <b>EESEL</b> bit is set to 1 and CS_N is asserted), ESDI is a buffered (delayed) version of SDI.
ESDO	I	<b>External EEPROM Serial Data Output.</b> This pin can be connected to the serial data output of the external SPI EEPROM. When the device is reading configuration information from the EEPROM, the data is conveyed on the ESDO pin. When the device's SPI master reads the EEPROM (i.e. when the <b>EESEL</b> bit is set to 1 and CS_N is asserted), the SDO pin is a buffered (delayed) version of ESDO.

**Table 4-6. JTAG Interface Pin Descriptions**

See Section 7 for functional description and Table 8-15 for timing specifications.

PIN NAME	TYPE <sup>(1)</sup>	PIN DESCRIPTION
JTRST_N	I <sub>PU</sub>	<b>JTAG Test Reset (Active Low).</b> Asynchronously resets the test access port (TAP) controller. JTRST_N should be held low during device power-up. If not used, JTRST_N can be held low or high after power-up.
JTCLK	I	<b>JTAG Clock.</b> Shifts data into JTDI on the rising edge and out of JTDO on the falling edge. If not used, JTCLK can be held low or high.

PIN NAME	TYPE <sup>(1)</sup>	PIN DESCRIPTION
JTDI	I <sub>PU</sub>	<b>JTAG Test Data Input.</b> Test instructions and data are clocked in on this pin on the rising edge of JTCLK. If not used, JTDI can be held low or high.
JTDO	O <sub>3</sub>	<b>JTAG Test Data Output.</b> Test instructions and data are clocked out on this pin on the falling edge of JTCLK. If not used, leave floating.
JTMS	I <sub>PU</sub>	<b>JTAG Test Mode Select.</b> Sampled on the rising edge of JTCLK and is used to place the port into the various defined IEEE 1149.1 states. If not used connect to 3.3V or leave floating.

**Table 4-7. Power-Supply Pin Descriptions**

PIN NAME	TYPE <sup>(1)</sup>	PIN DESCRIPTION
VDD_18	P	<b>Digital I/O Power Supply.</b> 1.8V ±5%.
VDD_33	P	<b>Digital I/O Power Supply.</b> 3.3V ±5%.
VDD_APLL1_18	P	<b>APLL1 Power Supply.</b> 1.8V ±5%. Also supply for IC1 input.
VDD_APLL1_33	P	<b>APLL1 Power Supply.</b> 3.3V ±5%. Also supply for IC1 input.
VDD_APLL2_18	P	<b>APLL2 Power Supply.</b> 1.8V ±5%. Also supply for IC2 and MCLKOSC inputs.
VDD_APLL2_33	P	<b>APLL2 Power Supply.</b> 3.3V ±5%. Also supply for IC2 and MCLKOSC inputs.
VDD_DIG_18	P	<b>Core Digital Power Supply.</b> 1.8V ±5%.
VDD_OC_18	P	<b>Output Clock Power Supply.</b> 1.8V ±5%.
VDD_XO_18	P	<b>Crystal Oscillator Power Supply.</b> 1.8V ±5%.
VDD_XO_33		<b>Crystal Oscillator Power Supply.</b> 3.3V ±5%.
VDDO18A	P	<b>Output Clock Power Supply, Bank A (OC1, OC2).</b> 1.8V ±5%.
VDDO18B	P	<b>Output Clock Power Supply, Bank B (OC3–OC5).</b> 1.8V ±5%.
VDDO18C	P	<b>Output Clock Power Supply, Bank C (OC6–OC8).</b> 1.8V ±5%.
VDDO18D	P	<b>Output Clock Power Supply, Bank D (OC9, OC10).</b> 1.8V ±5%.
VDDOA	P	<b>Output Clock Power Supply, Bank A (OC1, OC2).</b> 1.5V to 3.3V ±5%.
VDDOB	P	<b>Output Clock Power Supply, Bank B (OC3–OC5).</b> 1.5V to 3.3V ±5%.
VDDOC	P	<b>Output Clock Power Supply, Bank C (OC6–OC8).</b> 1.5V to 3.3V ±5%.
VDDOD	P	<b>Output Clock Power Supply, Bank D (OC9, OC10).</b> 1.5V to 3.3V ±5%.
VSS_APLL1	P	<b>Return for VDD_APLL1 Supplies.</b>
VSS_APLL2	P	<b>Return for VDD_APLL2 Supplies.</b>
VSS_DIG	P	<b>Core Digital Return.</b>
VSS_OC	P	<b>Output Clock Return.</b>
VSS_XO	P	<b>Crystal Oscillator Return.</b>
VSSOA	P	<b>Return for VDDOA Supply.</b>
VSSOB	P	<b>Return for VDDOB Supply.</b>
VSSOC	P	<b>Return for VDDOC Supply.</b>
VSSOD	P	<b>Return for VDDOD Supply.</b>
VSUB	P	<b>Substrate Voltage.</b> Connect to board ground.

**Note 1:** All pins, except power and analog pins, are CMOS/TTL unless otherwise specified in the pin description.

#### PIN TYPES

I = input pin

I<sub>DIFF</sub> = differential input, can be interfaced to LVDS, LVPECL, CML, HSTL or CMOS/TTL signals

I<sub>PD</sub> = input pin with internal 50kΩ pulldown

I<sub>PU</sub> = input pin with internal 50kΩ pullup

I/O = input/output pin

IO<sub>PD</sub> = input/output pin with internal 50kΩ pulldown

IO<sub>PU</sub> = input/output pin with internal 50kΩ pullup

O = output pin

O<sub>3</sub> = output pin that can be tri-stated (i.e., placed in a high-impedance state)

O<sub>DIFF</sub> = differential output, CML format

P = power-supply pin

**Note 2:** All digital pins, except ICn and OCn, are I/O pins in JTAG mode. ICn and OCn pins do not have JTAG functionality.

## 5. Functional Description

### 5.1 Device Identification and Protection

The 16-bit read-only ID field in the [ID1](#) and [ID2](#) registers is set to 00C3h = 195 decimal. The device revision can be read from the [REV](#) register. Contact the factory to interpret this value and determine the latest revision. The register set can be protected from inadvertent writes using the [PROT](#) register.

### 5.2 Top-Level Configuration

MAX24605 and MAX24610 have two fundamental modes of operation: APLL-only and DPLL+APLL.

#### 5.2.1 APLL-Only Mode

In APLL-only mode, the input block and the DPLL are powered down, and APLL1 and/or APLL2 are available to produce two independent families of output clock frequencies. The input block and the DPLL are powered down by setting [MCR1.ICBEN](#)=0 and [MCR1.DPLLEN](#)=0, respectively. This reduces chip power consumption as shown in [Table 8-2](#).

The bandwidth of the APLLs is approximately 400kHz and therefore in APLL-only mode the device does not filter jitter. This means that in applications where output signals must have sub-ps jitter, the APLL input signal must have sub-ps jitter. In addition, features of the input block and the DPLL including activity monitoring and frequency monitoring are not available. APLL-only mode is enabled when the APLL input muxes are set to select an input other than the DPLL output (i.e. [APLLCR2.APLLMUX](#)=0xx).

APLL-only mode has two usage cases for each APLL. First, the APLLs can be locked to the on-chip crystal oscillator as shown in [Figure 5-1](#). Second, each APLL can be locked to any of the four input clock signals, as shown in [Figure 5-2](#).

Figure 5-1. APLL-Only Mode: Clock Synthesis from a Crystal

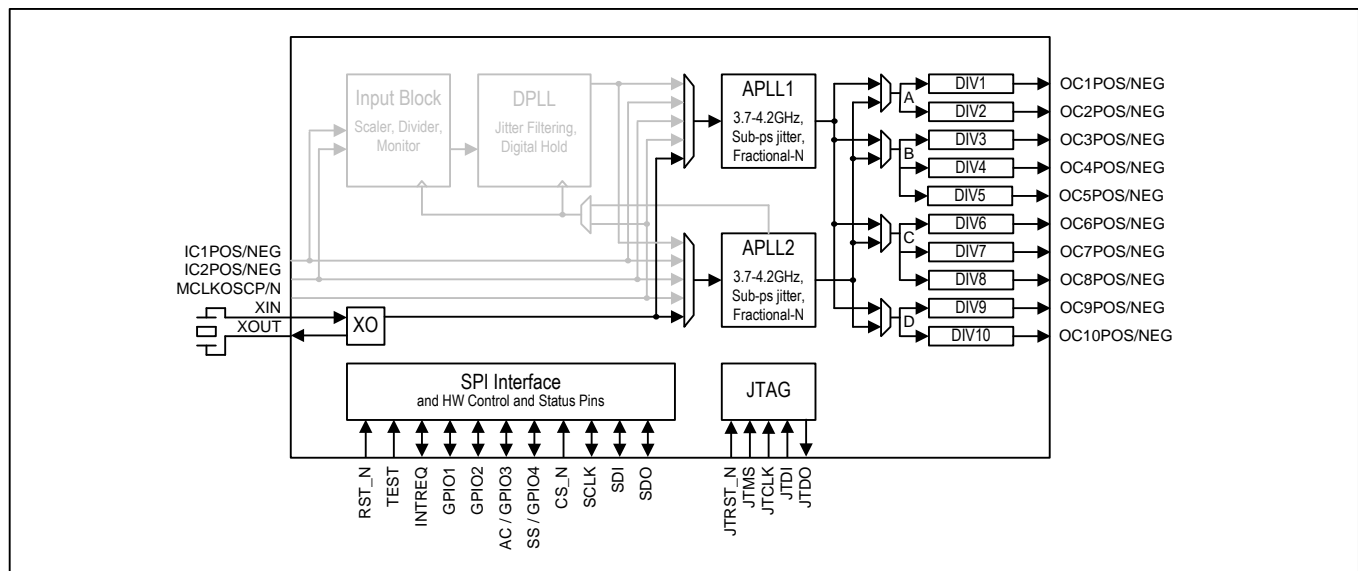
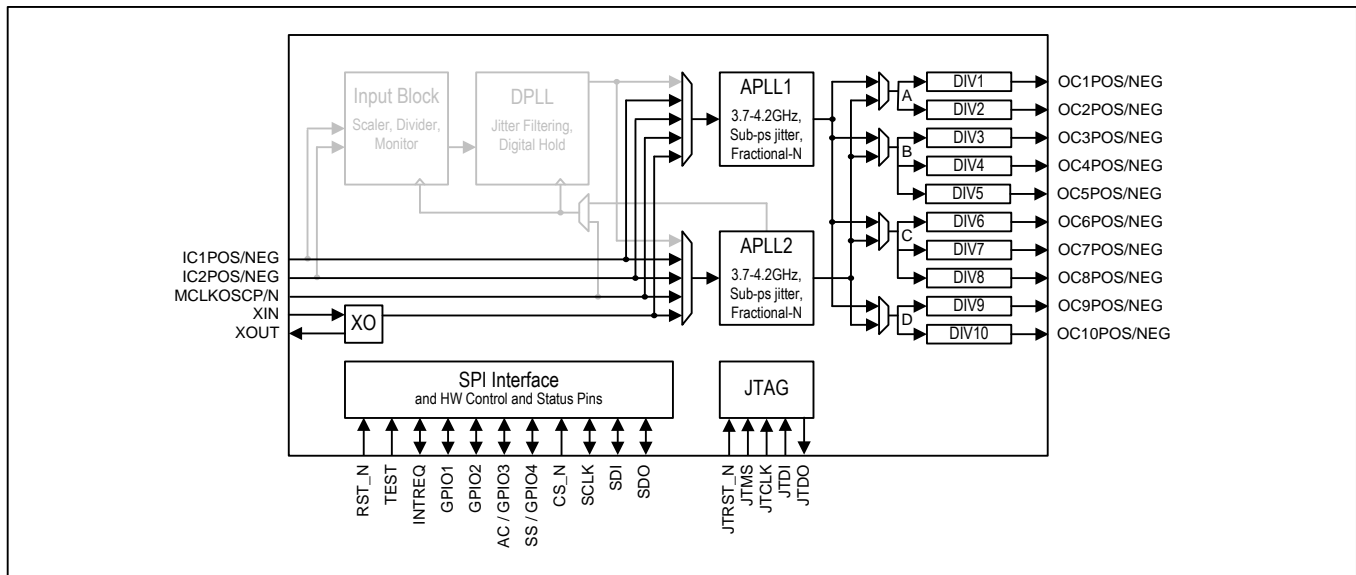


Figure 5-2. APLL-Only Mode: Locked to One of Four Input Clocks



### 5.2.2 DPLL+APLL Mode

In DPLL+APLL mode, the input block and DPLL are enabled and used. In this mode device power consumption is higher than APLL-only mode, but all input block features are available including activity monitoring, frequency monitoring and automatic reference switching. In addition, all DPLL features are available as well, including bandwidths low enough to filter jitter on the input clock signals.

DPLL+APLL mode is enabled when the APLL1 input mux is set to select the DPLL output (i.e. [APLLCR2.APLLMUX=100](#)) and the input block and DPLL are enabled using the enable bits in [MCR1](#).

In this mode the input block and the DPLL must operate from a master clock signal of approximately 200MHz. This master clock signal can be provided using either of two methods.

For method 1, a 190MHz to 208.333MHz local oscillator is connected directly to the MCLKOSCP/N pins, and the [MCR3.MCMUX](#) bit is set to 1 to connect this clock signal directly to the input block and the DPLL. This method, shown in [Figure 5-3](#), leaves APLL2 available to be synchronized to the DPLL and allows the device to make two families of output clock frequencies that are both synchronized to the DPLL's selected reference.

For method 2, APLL2 is configured to make the master clock signal from a lower frequency local oscillator connected to the MCLKOSCP/N pins. The APLL2 output frequency must be in the range 380MHz to 416.667MHz or the range 570MHz to 625MHz. APLL2's master clock divider ([MCR2.MCDIV](#)) is then configured to divide APLL2's output frequency by 2 or 3 to get a master clock frequency in the range 190MHz to 208.333MHz. The [MCR3.MCMUX](#) bit is set to 0 to connect the master clock signal from APLL2 to the input block and the DPLL. The APLL2 output clock frequency can also be provided to any of output banks A, B, C or D where it can be further divided to make output clock signals derived from the local oscillator.

Method 2 has two usage cases, 2a and 2b. For method 2a, APLL2 is locked to the on-chip crystal oscillator as shown in [Figure 5-4](#). This gives the lowest possible cost for the master clock reference, but the DPLL's frequency stability during digital hold is relatively poor due to the use of a non-temperature-compensated crystal. In some applications the DPLL is expected to always be locked to one of the two input clocks and rarely or never enter digital hold. For these applications DPLL stability during digital hold is not a requirement, and deriving the master clock from a crystal is appropriate.

For method 2b, APLL2 is locked to an external oscillator as shown in [Figure 5-5](#). This allows a more stable but more expensive reference for the master clock, such as a high-stability XO, a TCXO or even an OCXO.

Figure 5-3. DPLL+APLL Mode: Method 1, Master Clock from High-Speed External Oscillator

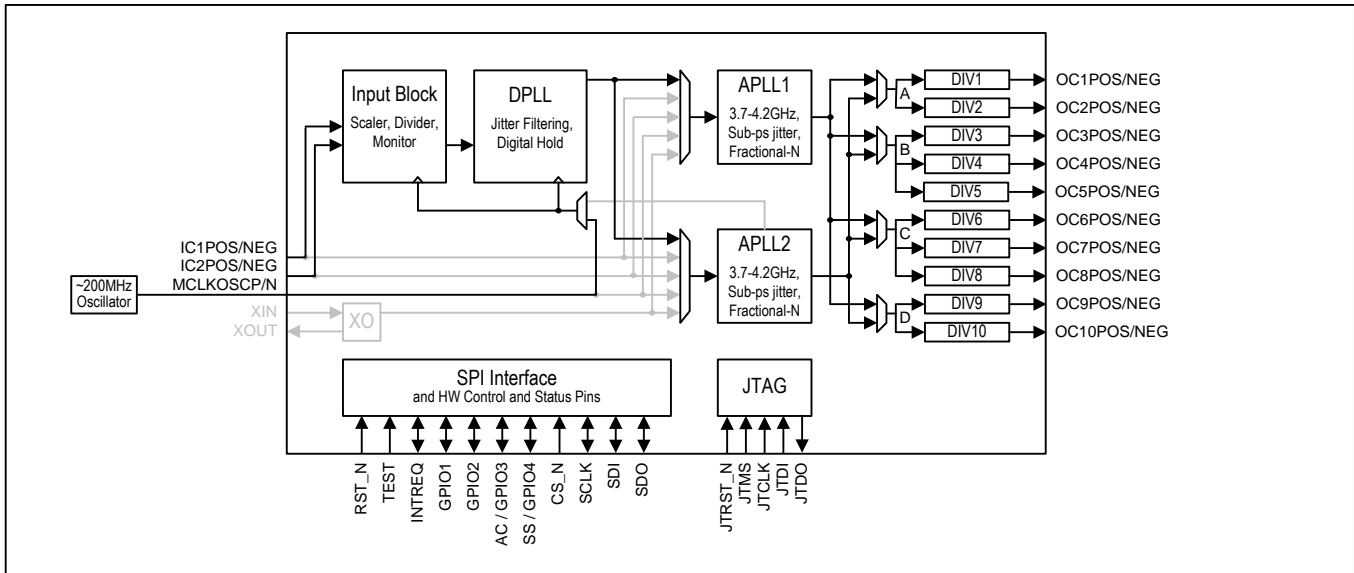


Figure 5-4. DPLL+APLL Mode: Method 2a, Master Clock from Crystal Oscillator Multiplied by APLL2

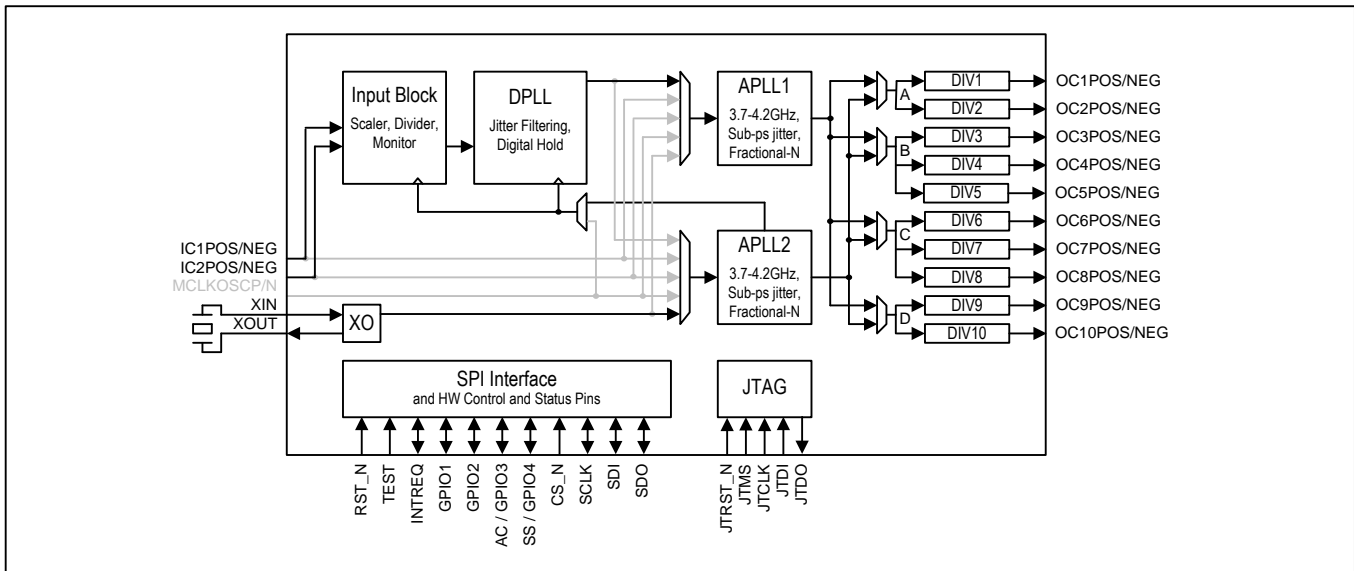
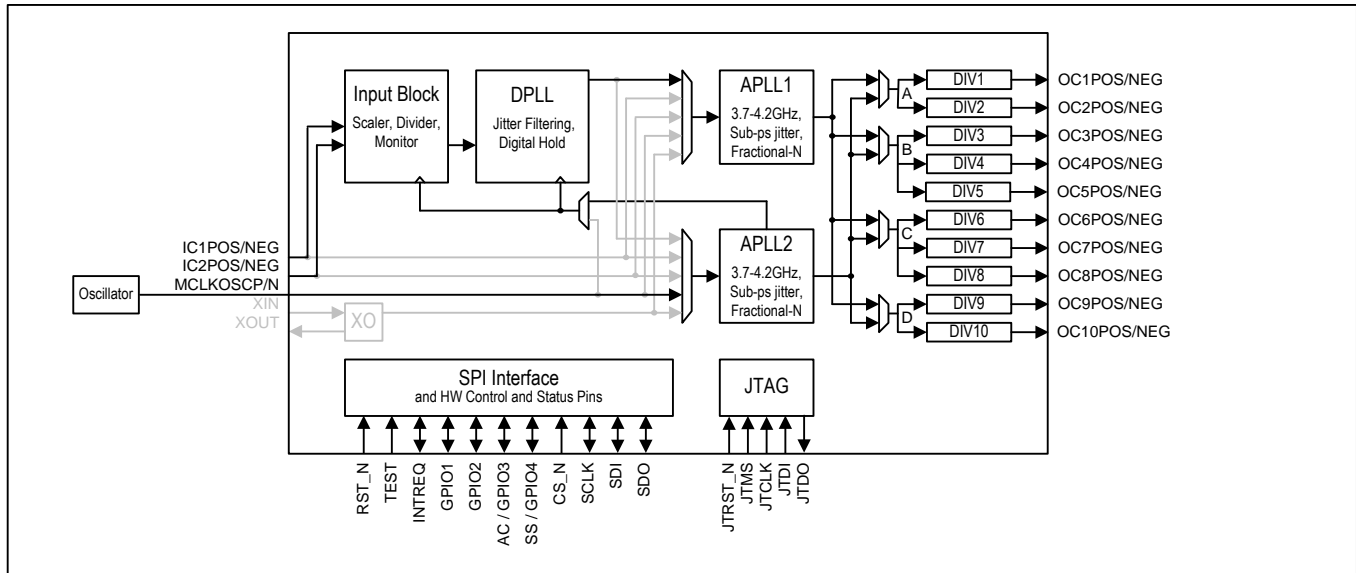


Figure 5-5. DPLL+APLL Mode: Method 2b, Master Clock from External Oscillator Multiplied by APLL2



### 5.3 Local Oscillator and Master Clock Configuration

Section 5.2 describes several device configurations that make use of either an external local oscillator (XO, TCXO, OCXO) or the on-chip crystal oscillator connected to an external 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 oscillator and the required characteristics of the crystal. Section 5.3.3 describes how to configure APLL2 to lock to either an external oscillator or the on-chip crystal oscillator and produce a suitable master clock for the input block and the DPLL.

#### 5.3.1 External Oscillator

A signal from an external oscillator can be connected to the MCLKOSCP/N pins. The external oscillator can be either differential or single-ended and any frequency from 9.72MHz to 750MHz (but see additional constraint for method 1 in section 5.2.2). See the MCLKOSCP/N pin description in Table 4-1 for additional details. For lowest output jitter, a differential signal is best. To minimize jitter when a single-ended signal is used, 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. If the oscillator is located more than 2cm away from the device, consider connecting the single-ended oscillator output to an LVDS driver IC (such as MAX9110) and sending a differential clock signal to the device pins.

When the DPLL master clock (see section 5.3.3) is derived from the oscillator signal applied to the MCLKOSCP/N pins, the stability of the DPLL in free-run or digital hold is equivalent to the stability of the oscillator. While many applications can make use of a simple crystal oscillator, some applications may require the stability of a TCXO or an OCXO. Contact Microsemi timing products technical support for recommended oscillator components.

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 in the DPLL or in the APLLs. When the device is configured for DPLL+APLL mode, the DPLL's MCFREQ field can be used to compensate for oscillator frequency error. When the device is configured for APLL-only mode, the APLLs' fractional feedback divider values (AFBDIV) can be adjusted by ppb or ppm to compensate for oscillator frequency error.

##### 5.3.1.1 Oscillator Characteristics to Minimize Output Jitter

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 Noise: Typical value of -148dBc/Hz or lower at 10kHz offset from the carrier.

- Frequency: The higher the better, all else being equal. Frequencies that are integer divisors of 4000MHz or 4096MHz are excellent choices, including 50MHz and 51.2MHz.

### 5.3.2 On-Chip Crystal Oscillator

The crystal oscillator is designed to drive a fundamental mode, AT-cut crystal resonator. See Table 5-1 for recommended crystal specifications. When a crystal is not connected between XIN and XOUT, the XIN pin can be used as a single-ended input to the APLLs.

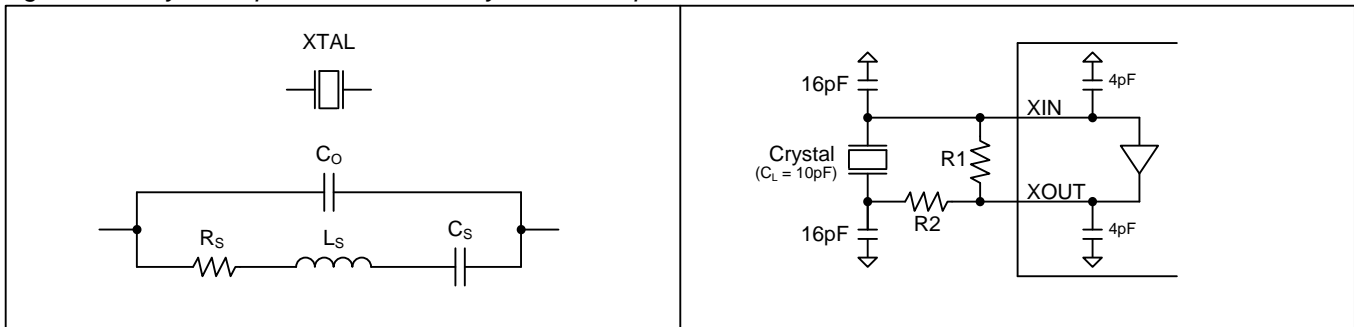
To use the crystal oscillator with an external crystal, set MCR2.XIEN=1 to enable the XIN pin logic and set MCR2.XOEN=1 to enable the XOUT pin so the XO can oscillate. To use the XIN pin as a single-ended input, set MCR2.XIEN=1 to enable the XIN pin and set MCR2.XOEN=0 to disable the XOUT pin to minimize power and noise. If the XIN pin is not used, set MCR2.XIEN=0 and MCR2.XOEN=0 to minimize power and noise.

See Figure 5-6 for the crystal equivalent circuit and the recommended external capacitor connections. To achieve a crystal load ( $C_L$ ) of 10pF, an external 16pF is placed in parallel with the 4pF internal capacitance of the XIN pin, and an external 16pF is placed in parallel with the 4pF internal capacitance of the XOUT pin. The crystal then sees a load of 20pF in series with 20pF, which is 10pF total load. Note that the 16pF capacitance values in Figure 5-6 include all capacitance on those nodes. If, for example, PCB trace capacitance between crystal pin and IC pin is 2pF then 14pF capacitors should be used to make 16pF total.

The crystal, traces, and two external capacitors should be placed on the board as close as possible to the XIN and XOUT 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 sensitivities that can cause crystal oscillator 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, TCXO or OCXO should be used instead.

Figure 5-6. Crystal Equivalent Circuit / Crystal and Capacitor Connections



Note 1:  $R_1=1\text{M}\Omega$ . The value of  $R_2$  is a function of crystal frequency, loading and maximum power rating. Contact the factory for guidance in choosing the right  $R_2$  resistor for a specific crystal.

Table 5-1. Crystal Selection Parameters

PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS
Crystal Oscillation Frequency	$f_{\text{OSC}}$	25	25, 50, 51.2 <sup>1</sup>	52	MHz
Shunt Capacitance	$C_0$		2	5	pF
Load Capacitance	$C_L$		10		pF
Equivalent Series Resistance (ESR) <sup>2</sup>	$f_{\text{OSC}} < 40\text{MHz}$	$R_S$		60	$\Omega$
	$f_{\text{OSC}} > 40\text{MHz}$	$R_S$		50	$\Omega$
Maximum Crystal Drive Level		100			$\mu\text{W}$

Note 1: Crystal frequencies of 49.152MHz, 50MHz and 51.2MHz are excellent choices for lowest output jitter.

Note 2: These ESR limits are chosen to constrain crystal drive level to less than 100 $\mu\text{W}$ . If the crystal can tolerate a drive level greater than 100 $\mu\text{W}$  then proportionally higher ESR is acceptable.



PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS
Crystal Oscillator Frequency Stability vs. Power Supply	$f_{FVD}$		0.2	0.5	ppm per 10% $\Delta$ in VDD

Any known frequency inaccuracy of the crystal can be compensated in the DPLL or in the APLLs. When the device is configured for DPLL+APLL mode, the DPLL's [MCFREQ](#) field can be used to compensate for crystal frequency error. When the device is configured for APLL-only mode, the APLLs' fractional feedback divider values ([AFBDIV](#)) can be adjusted by ppb or ppm to compensate for crystal oscillator frequency error.

### 5.3.3 Master Clock APLL Configuration

This section does not apply for APLL-only mode.

In DPLL+APLL mode method 2 (see section [5.2.2](#)) the main purpose of APLL2 is to provide the required master clock signal (typically 200MHz or 204.8MHz) to the input block and the DPLL. APLL2 accepts a clock signal from either the MCLKOSCP/N pins or from the on-chip crystal oscillator as specified by APLL2's [APLLCR2.APLLMUX](#) field. APLL2 can lock to any input clock frequency from 9.72MHz to 102.4MHz. The APLL's input divider, controlled by [APLLCR2.AIDIV](#), can be used to divide frequencies up to 750MHz down to the 9.72MHz to 102.4MHz range. To minimize output jitter, the APLL2 input frequency should be multiplied by an integer (i.e. APLL2's [AFBDIV](#) value should be an integer) to a VCO frequency that can be internally divided by APLL2's high-speed divider ([APLLCR1.HSDIV](#)) and then by the master clock divider ([MCR2.MCDIV](#)) to get a master clock frequency in the range of 190MHz to 208.333MHz. Higher APLL2 input frequencies give lower output jitter, all else being equal. Several possible APLL2 input clock frequencies are shown in [Table 5-2](#) below along with the corresponding APLL2 register settings and resulting master clock frequencies.

**Table 5-2. Example Master Clock APLL Input Frequencies and Configurations**

APLL2 Input Frequency <sup>2,3</sup>	Multiplier Value ( <a href="#">AFBDIV</a> )	APLL2 VCO Frequency	Divider Value ( <a href="#">APLLCR1.HSDIV</a> )	Divider Value ( <a href="#">MCR2.MCDIV</a> )	Master Clock Frequency
51.2MHz <sup>1</sup>	80	4096MHz	10	2	204.8MHz
50MHz <sup>1</sup>	80	4000MHz	10	2	200MHz
40MHz	100	4000MHz	10	2	200MHz
25.6MHz	160	4096MHz	10	2	204.8MHz
25MHz	160	4000MHz	10	2	200MHz
12.8MHz	320	4096MHz	10	2	204.8MHz
10MHz	400	4000MHz	10	2	200MHz

Note 1: Input frequencies of 98.304MHz, 50MHz and 51.2MHz are excellent choices for lowest output jitter.

Note 2: Many other input frequencies are possible.

Note 3: The APLL2 input frequency range is wider than the crystal oscillator frequency range.

By default the device assumes a master clock frequency of 204.8MHz. When the master clock frequency is different than 204.8MHz, the [MCDNOM](#), [MCINOM](#) and [MCAC](#) registers must be set correctly for proper operation of the input block and the DPLL.

The APLLs are self-oscillating, and therefore APLL2's output toggles even when the signal on the MCLKOSC pins or the output of the on-chip crystal oscillator is not toggling. This allows the device to continue to operate (although not in a standards-compliant manner) even during a complete oscillator failure. If the input clock to APLL2 is not toggling or is grossly off frequency, the device sets the [PLL1LSR.MCFAIL](#) latched status bit. This in turn can cause an interrupt if configured to do so.

The MCLKOSC input must be enabled before use by setting [MCR2.MCEN](#)=1. The master clock divider must be enabled before use by setting [MCR2.MCDIV](#) to a non-zero value.



## 5.4 Input Signal Format Configuration

Input clocks IC1 and IC2 are enabled by setting [MCR2.IC1EN=1](#) and [IC2EN=1](#), respectively. The power consumed by a differential receiver is shown in [Table 8-2](#). The electrical specifications for these inputs are listed in [Table 8-4](#). Each input clock can be configured to accept nearly any differential signal format by using the proper set of external components (see [Table 8-4](#) and [Figure 8-1](#)). To configure these differential inputs to accept single-ended CMOS or TTL signals, connect the single-ended signal to the POS pin, and connect the NEG pin to a capacitor (0.1μF or 0.01μF) to VSS\_IO. As shown in [Figure 8-1](#), the NEG pin is internally biased to approximately 1.2V. If a 1.2V bias is unsuitable, an external voltage divider can be used to set a different bias. If an input is not used, both POS and NEG pins can be left floating.

**Table 5-3. Input Clock Capabilities**

Input Clock	Signal Format	Frequency Range to the Input block (MHz)	Frequency Range to the APLLs (MHz)
IC1	Differential or CMOS/TTL	Differential: 2kHz to 750MHz	Differential: 9.72MHz to 750MHz
IC2		Single-ended: 2kHz to 160MHz <sup>(1)</sup>	Single-ended: 9.72MHz to 160MHz

**Note 1:** See sections [5.5.1](#) for details on frequency dividers, fractional scaling, and direct-lock frequencies supported by the DPLL.

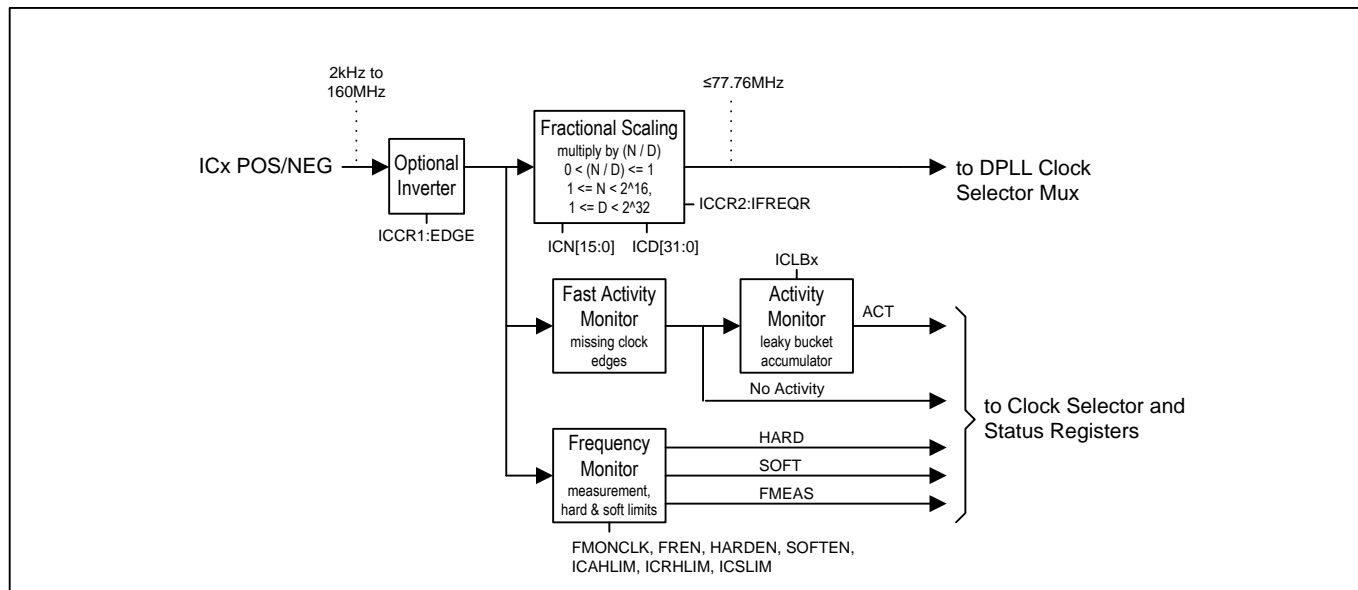
## 5.5 Input Clock Divider, Monitor and Selector

The input block performs the following functions:

- Frequency division (integer or fractional) to a frequency suitable for DPLL locking
- Activity monitoring
- Frequency monitoring
- DPLL input clock selection (automatic or manual)

[Figure 5-7](#) is a detailed block diagram of the input block. This block requires a master clock as described in section [5.2.2](#). To enable the input block set [MCR1.ICBEN=1](#). To enable APLL2, set [APLLSEL=2](#) and then set [APLLCR1.APLEN=1](#).

**Figure 5-7. Input block Diagram**



It is important to note that the input block provides its selector and divider services to the DPLL only. When the device is configured at the top level to connect an input signal to directly to one or both APLLs, the input block is bypassed as shown in the block diagram in [Figure 2-1](#). In this configuration the input block can still be used to monitor the input clock signals for activity and frequency accuracy.

### 5.5.1 Input Clock Frequency Dividers, Scaling and Inversion

The input block tolerates a wide range of duty cycles out to a minimum high time or minimum low time of 3ns or 30% of the clock period, whichever is smaller. The input clock registers are bank-selected by the [ICSEL](#) register (see section [6.1.3](#)).

As shown in [Figure 5-7](#), any frequency in the 2kHz to 750MHz range can be accepted by the input block as long as the frequency meets one of the following criteria:

1. A DPLL locking frequency listed in the [ICCR1.LKFREQ](#) register description
2. A frequency that can be divided by an unsigned integer ([ICD](#)+1) to produce a DPLL locking frequency listed in [ICCR1.LKFREQ](#)
3. A frequency that can be multiplied by the ratio of two integers ([ICN](#)+1) / ([ICD](#)+1) to produce a DPLL locking frequency  $\geq 1\text{MHz}$  listed in [ICCR1.LKFREQ](#)

Important notes about the input block:

- [ICCR1.POL](#) specifies the edge to which the DPLL will lock (by default, the falling edge).
- The frequency range field [ICCR1.IFREQR](#) must be set correctly for the actual frequency of the input clock.
- For fractional scaling, the input clock frequency must be  $\geq 1\text{MHz}$ , and [ICN](#) and [ICD](#) must be set to meet the requirement  $0 < (\text{ICN} + 1) / (\text{ICD} + 1) \leq 0.25$ .
- The frequency out of the scaling block must be a DPLL locking frequency listed in [ICCR1.LKFREQ](#).
- [ICN](#) and [ICD](#) are set to 0 by default to give no dividing or scaling. This setting is useful for rates that are DPLL locking frequencies (e.g. 1MHz and 25MHz)

### 5.5.2 Input Clock Monitoring

Each input clock (IC1, IC2) is continuously monitored for frequency accuracy and activity. Frequency monitoring is described in section [5.5.2.1](#), while activity monitoring is described in Sections [5.5.2.2](#) and [5.5.2.3](#). Any input clock that has a frequency out-of-band alarm or activity alarm is automatically declared invalid. The valid/invalid state of each input clock is reported in the corresponding real-time status bit in the [VALSR1](#) register. When the valid/invalid state of a clock changes, the corresponding latched status bit is set in the [ICLSR1](#) register, and an interrupt request occurs if the corresponding interrupt enable bit is set in the [ICIER1](#) register. Input clocks marked invalid cannot be automatically selected as the reference for the DPLL.

#### 5.5.2.1 Frequency Monitoring

The input block monitors the frequency of each input clock and invalidates any clock whose frequency is outside of specified limits. Measured frequency can be read from the [FMEAS](#) field. In addition, three frequency limits can be specified: a soft limit ([ICSLIM](#)), a rejection hard limit ([ICRHLIM](#)), and an acceptance hard limit ([IC AHLIM](#)). When the frequency of an input clock is greater than or equal to the soft limit, the corresponding [ISR.SOFT](#) alarm bit is set to 1. The soft limit is only for monitoring; triggering it does not invalidate the clock. When the frequency offset of an input clock is greater than or equal to the rejection hard limit, the corresponding [ISR.HARD](#) alarm bit is set to 1, and the clock is marked invalid in the [VALSR1](#) register. When the frequency offset of an input clock is less than the acceptance hard limit, the [ISR.HARD](#) alarm bit is cleared to 0. Together, the acceptance hard limit and the rejection hard limit allow hysteresis to be configured as required by Telcordia spec GR-1244-CORE.

Monitoring according to the hard and soft limits is enabled/disabled using the HARDEN and SOFTEN bits in the [ICCR2](#) register. Frequency monitoring is only done on an input clock when the clock does not have an activity alarm.

The frequency monitoring logic determines the nominal (ideal, zero-error) frequency of the input clock from the values in the [ICCR1.LKFREQ](#), [ICN](#), [ICD](#), and [ICCR1.IFREQR](#) fields. As must be done in any frequency measurement system, the frequency monitor counts the number of input clock cycles that occur in an interval of time equal to a specific number of reference clock periods. It then compares the actual count to the expected count to determine the fractional frequency offset of the input clock. The reference clock for the frequency monitor can be either the internal master clock (see section [5.3](#)) or the output of the DPLL, depending on the setting of [ICCR2.FMONCLK](#).

Frequency measurement time can be specified in the [ICCR3.FMONLEN](#) field. For any input clock there is a relationship among frequency measurement precision, measurement time (duration), and maximum input jitter amplitude as follows:

$$\text{freq\_meas\_time} \geq \text{max\_p-p\_jitter\_amplitude} / (0.5 * \text{freq\_meas\_precision})$$

When [ICCR2.FREN](#)=1 the input block performs gross frequency monitoring and invalidates any clock whose frequency is more than 10,000ppm away from nominal. This function is useful when hard limits are not enabled ([ICCR2.HARDEN](#)=0).

### 5.5.2.2 Activity Monitoring

The input block monitors each input clock for activity and proper behavior using a leaky bucket accumulator. A leaky bucket accumulator is similar to an analog integrator: the output amplitude increases in the presence of input events and gradually decays in the absence of events. When events occur infrequently, the accumulator value decays fully between events and no alarm is declared. When events occur close enough together, the accumulator increments faster than it can decay and eventually reaches the alarm threshold. After an alarm has been declared, if events occur infrequently enough, the accumulator can decay faster than it is incremented by new events and eventually reaches the alarm clear threshold. The leaky bucket events come from the fast activity monitor.

The leaky bucket accumulator for each input clock has programmable size, alarm declare threshold, alarm clear threshold, and decay rate, all of which are specified in the [ICLB](#) registers.

Activity monitoring is divided into 128ms intervals. The accumulator is incremented once for each 128ms interval in which the input clock is inactive for a few clock cycles (see [Table 5-4](#)). Thus the “fill” rate of the bucket is at most 1 unit per 128ms, or approximately 8 units/second. During each period of 1, 2, 4 or 8 intervals (programmable), the accumulator decrements if no irregularities occur. Thus the “leak” rate of the bucket is approximately 8, 4, 2, or 1 units/second. A leak is prevented when a fill event occurs in the same interval.

When the value of an accumulator reaches the alarm threshold ([ICLBU](#) register), the corresponding [ISR.ACT](#) alarm bit is set to 1, and the clock is marked invalid in the [VALSR1](#) register. When the value of an accumulator reaches the alarm clear threshold ([ICLBL](#) register), the activity alarm is cleared by clearing the clock’s ACT bit. The accumulator cannot increment past the size of the bucket specified in the [ICLBS](#) register. The decay rate of the accumulator is specified in the [ICLBD](#) register. The values stored in the leaky bucket configuration registers must have the following relationship at all times:  $\text{ICLBS} \geq \text{ICLBU} > \text{ICLBL}$ . If [ICLBS](#) is set to 00h, the leaky bucket count is set to 0, the leaky bucket is disabled, and [ISR.ACT](#) alarm bit is set to 0.

When the leaky bucket is empty, the minimum time to declare an activity alarm in seconds is  $\text{ICLBU} / 8$ . The minimum time to clear an activity alarm in seconds is  $2^{\text{ICLBD}} \times (\text{ICLBS} - \text{ICLBL}) / 8$ . As an example, assume  $\text{ICLBU} = 8$ ,  $\text{ICLBL} = 1$ ,  $\text{ICLBS} = 10$ , and  $\text{ICLBD} = 0$ . The minimum time to declare an activity alarm would be  $8 / 8 = 1$  second. The minimum time to clear the activity alarm would be  $2^0 \times (10 - 1) / 8 = 1.125$  seconds.

**Table 5-4. Activity Monitoring, Missing Clock Cycles vs. Frequency**

INPUT CLOCK FREQUENCY	NUMBER OF MISSING CLOCK CYCLES
<100 MHz	2
100 – 200 MHz	4
200 – 400 MHz	8
>400 MHz	16

### 5.5.2.3 Selected Reference Fast Activity Monitoring

The input clock that the DPLL is currently locked to is called the selected reference. The quality of the DPLL's selected reference is exceedingly important, since missing cycles and other anomalies on the selected reference can cause unwanted jitter, wander or frequency offset on the output clocks. When anomalies occur on the selected reference they must be detected as soon as possible to give the DPLL opportunity to temporarily disconnect from the reference until the reference is available again. By design, the regular input clock activity monitor (the leaky bucket accumulator described in section 5.5.2.2) is too slow to be suitable for monitoring the selected reference. Instead, the input block provides a fast activity monitor that detects inactivity after a few missing clock cycles (see Table 5-4).

When the fast activity monitor detects a no-activity event, the DPLL immediately enters a temporary digital hold state to isolate itself from the selected reference and sets the SRFAIL bit in PLL1LSR. The setting of the SRFAIL bit can cause an interrupt request if the corresponding enable bit is set in PLL1IER. By setting the appropriate GPIOSS register to xx001011b, a GPIO pin can be configured to follow the state of the SRFAIL status bit. Optionally, a no-activity event can also cause an ultra-fast reference switch (see Section 5.5.3.4). When DPLLCCR5.NALOL = 0 (default), the DPLL does not declare loss-of-lock during no-activity events. If the selected reference becomes available again before any alarms are declared by the activity monitor or frequency monitor, then the DPLL continues to track the selected reference using nearest-edge locking ( $\pm 180^\circ$ ) to avoid cycle slips. When NALOL = 1, the DPLL declares loss-of-lock during no-activity events. This causes the DPLL state machine to transition to the loss-of-lock state, which sets the STATE bit in PLL1LSR and causes an interrupt request if enabled. If the selected reference becomes available again before any alarms are declared by the activity monitor or frequency monitor, then the DPLL tracks the selected reference using phase/frequency locking ( $\pm 360^\circ$ ) until phase lock is reestablished.

### 5.5.2.4 External Monitoring

Some clock signals come from external components that can monitor the quality of a clock signal or the quality of a signal from which the clock signal is derived. One example is a BITS receiver, which receives a DS1, E1 or 2048kHz synchronization signal and recovers a clock from that signal. A BITS receiver monitors the incoming signal and can declare loss of signal (LOS), loss of frame alignment (LOF) and other defects in the incoming signal. Another example is a synchronous Ethernet PHY, which receives an Ethernet signal and recovers a clock from that signal and can declare loss of lock, loss of codeword alignment and other defects.

When a neighboring component can detect that the incoming signal or the clock recovered from the signal is somehow out of specification, a bad-clock signal from that component can be connected to a GPIO pin on the device. The device can then be configured to squelch the input clock when the bad-clock signal is high by setting ICCR2.GPIOSQ=1 for that input clock. IC1 is squelched when GPIO1 is high. IC2 is squelched when GPIO2 is high.

## 5.5.3 Input Clock Priority, Selection and Switching

### 5.5.3.1 Priority Configuration

During normal operation, the selected reference for the DPLL is chosen automatically based on the priority rankings assigned to the input clocks in the input priority register (IPR1). The default input clock priorities are shown in Table 5-5.

Any unused input clock should be given the priority value 0, which disables the clock and marks it as unavailable for selection. Priority 1 is highest while priority 15 is lowest.

**Table 5-5. Default Input Clock Priorities**

INPUT CLOCK	DPLL DEFAULT PRIORITY
IC1	1
IC2	2

### **5.5.3.2 Automatic Selection**

The reference selection algorithm for the DPLL chooses the highest-priority valid input clock to be the selected reference. The real-time valid/invalid state of each input clock is maintained in the [VALSR1](#) register (see section [5.5.2](#)). The priority of each input clock is set as described in section [5.5.3.1](#). To select the proper input clock based on these criteria, the selection algorithm maintains a priority table of valid inputs. The top entry in this priority table and the selected reference are displayed in the [PTAB1](#) register.

If two or more input clocks are given the same priority number then those inputs are prioritized among themselves using a fixed circular list. If one equal-priority clock is the selected reference but becomes invalid then the next equal-priority clock in the list becomes the selected reference. If an equal-priority clock that is not the selected reference becomes invalid, it is simply skipped over in the circular list. The selection among equal-priority inputs is inherently nonrevertive, and revertive switching mode (see next paragraph) has no effect in the case where multiple equal-priority inputs have the highest priority.

An important input to the selection algorithm is the REVERT bit in the [DPLLCR1](#) register. In revertive mode (REVERT = 1), if an input clock with a higher priority than the selected reference becomes valid, the higher priority reference immediately becomes the selected reference. In nonrevertive mode (REVERT = 0), the higher priority reference does not immediately become the selected reference but does become the highest priority reference in the priority table (REF1 field in the [PTAB1](#) register). (The selection algorithm always switches to the highest-priority valid input when the selected reference goes invalid, regardless of the state of the REVERT bit.) For many applications, nonrevertive mode is preferred because it minimizes disturbances on the output clocks due to reference switching.

In nonrevertive mode, planned switchover to a newly-valid higher priority input clock can be done manually under software control. The validation of the new higher priority clock sets the corresponding status bit in the [ICLSR](#) registers, which can drive an interrupt request if needed. System software can then respond to this change of state by briefly enabling revertive mode (toggling REVERT high then back low) to force the switchover to the higher priority clock.

### **5.5.3.3 Forced Selection**

The [DPLLCR1.FORCE](#) register field provides a way to force a specified input clock to be the selected reference for the DPLL. In this register field, 0 specifies normal operation with automatic reference selection. Nonzero values specify the input clock to be the forced selection. Internally, forcing is accomplished by giving the specified clock the highest priority (as specified in [PTAB1.REF1](#)). In revertive mode ([DPLLCR1.REVERT](#) = 1) the forced clock automatically becomes the selected reference (as specified in [PTAB1.SELREF](#)) as well. In nonrevertive mode the forced clock only becomes the selected reference when the existing selected reference is invalidated or made unavailable for selection.

### **5.5.3.4 Ultra-Fast Reference Switching**

By default, disqualification of the selected reference and switchover to another reference occurs when the activity monitor's inactivity alarm threshold has been crossed, a process that takes on the order of hundreds of milliseconds or seconds. However, an option for extremely fast disqualification and switchover is also available. When ultra-fast switching is enabled ([DPLLCR1.UFSW](#) = 1), if the fast activity monitor detects a few missing clock cycles (see [Table 5-4](#)) it declares the reference failed (by forcing the leaky bucket accumulator to its upper threshold, see Section [5.5.2.2](#)) and initiates reference switching. This is in addition to setting the SRFAIL bit and optionally generating an interrupt request, as described in Section [5.5.2.3](#). When ultra-fast switching occurs, the DPLL transitions to the prelocked 2 state, which allows switching to occur faster by bypassing the loss-of-lock state. The device should be in nonrevertive mode when ultra-fast switching is enabled. If the device is in revertive mode, ultra-fast switching could cause excessive reference switching when the highest priority input is intermittent.

### **5.5.3.5 External Reference Switching Mode**

In this mode the SS input pin controls reference switching between the IC1 and IC2 inputs. This mode is enabled by setting the EXTSW bit to 1 in the [DPLLCR1](#) register. In this mode, if the SS pin is high, the DPLL is forced to lock to input IC1 whether or not the selected input has a valid reference signal. If the SS pin is low the DPLL is forced to lock to input IC2 whether or not the selected input has a valid reference signal.

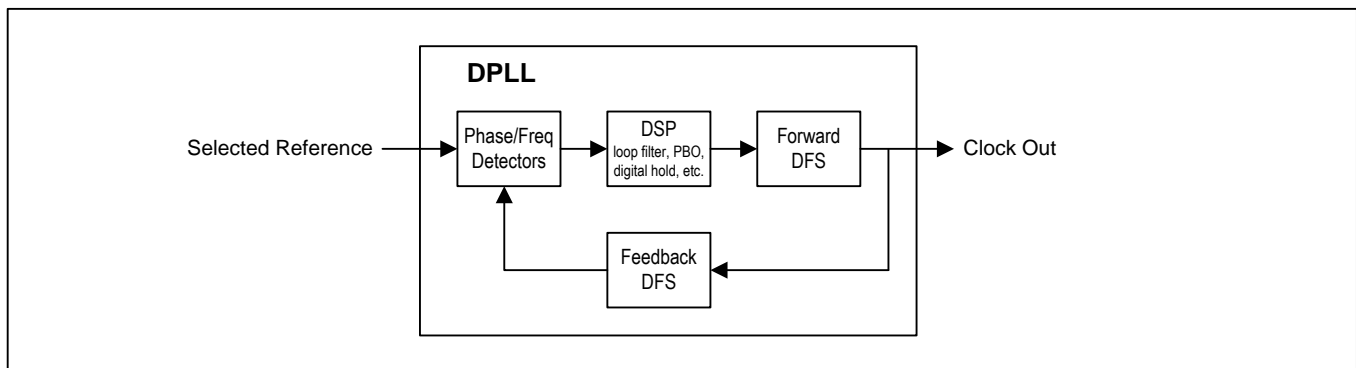
In external reference switching mode the input selector logic behaves as a simple 2:1 mux, and the DPLL is forced to try to lock to the selected reference whether it is valid or not. Unlike forced reference selection (Section 5.5.3.3) this mode controls the PTAB1.SELREF field directly and is, therefore, not affected by the state of the DPLLCR1.REVERT bit. During external reference switching mode, only PTAB1.SELREF is affected; the REF1 field continues to indicate the highest-priority valid input chosen by the automatic selection logic. The priorities of IC1 and IC2 in the IPR1 register must be non-zero for proper behavior in external reference switching mode.

### 5.5.3.6 Output Clock Phase Continuity During Reference Switching

If the DPLL frequency limit (HRDLIM) is set to less than  $\pm 30$ ppm, the device always complies with the GR-1244-CORE requirement that the rate of phase change must be less than 81ns per 1.326ms during reference switching.

## 5.6 DPLL Architecture and Configuration

Figure 5-8. DPLL Block Diagram



Digital PLLs have two key benefits: (1) stable, repeatable performance that is insensitive to process variations, temperature, and voltage; and (2) flexible behavior that is easily programmed via configuration registers. DPLLs use digital frequency synthesis (DFS) to generate various clocks. In DFS a high-speed master clock is multiplied up from the local oscillator clock applied to the MCLKOSC pins. This master clock is then digitally divided down to the desired output frequency. The DFS output clock has approximately 40ps RMS jitter.

An APLL can then be used to filter the jitter from the DPLL, reducing the output jitter to less than 1ps RMS, measured over 12kHz to 20MHz.

The DPLL in the device is configurable for many PLL parameters including bandwidth, damping factor, input frequency, pull-in/hold-in range, input-to-output phase offset, and more. No knowledge of loop equations or gain parameters is required to configure and operate the device. No external components are required for the DPLL except a local oscillator or crystal connected to the MCLKOSC pins.

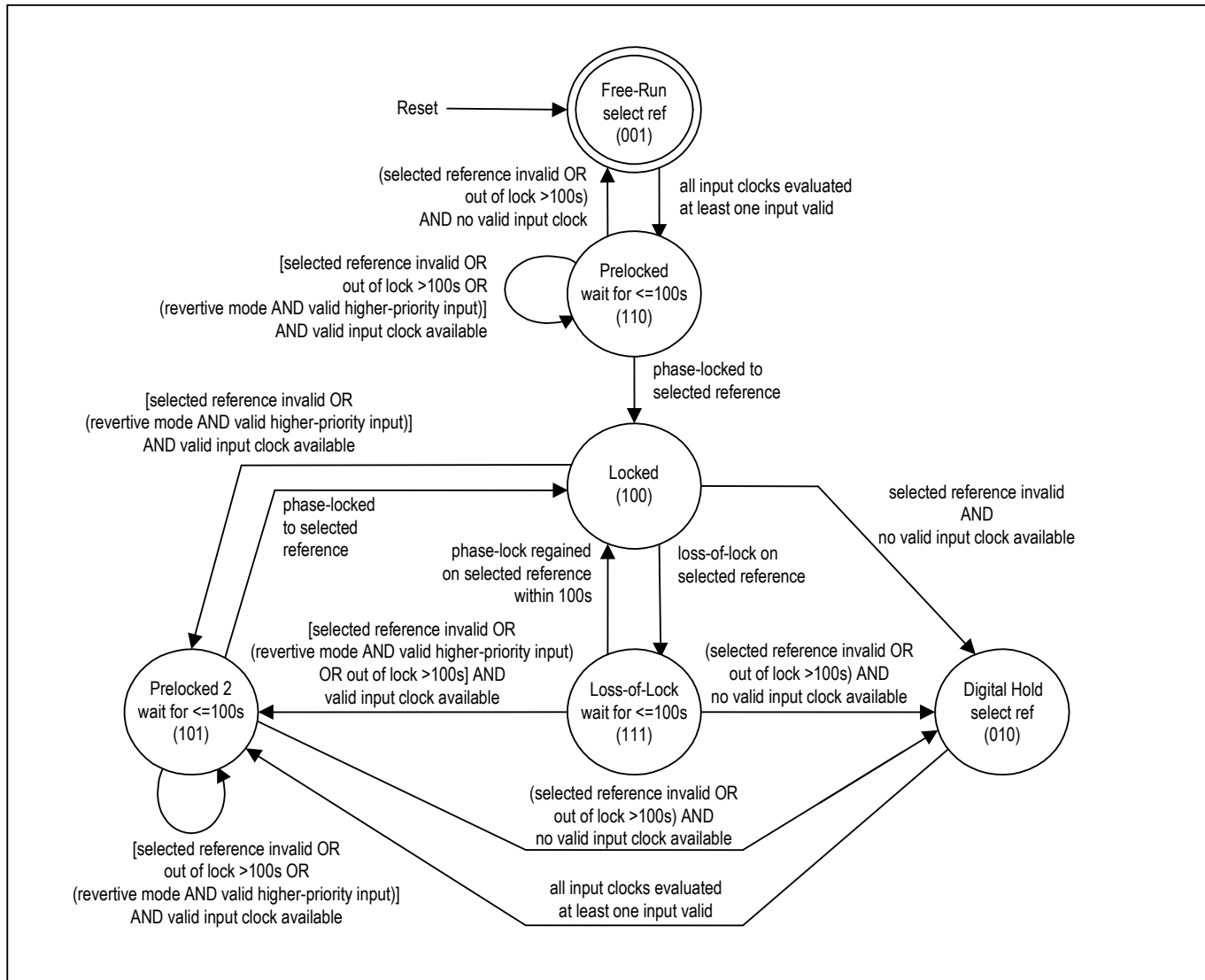
### 5.6.1 DPLL State Machine

The DPLL has three main timing modes: locked, digital hold and free-run. The control state machine for the DPLL has states for each timing mode as well as three temporary states: prelocked, prelocked 2 and loss-of-lock. The state transition diagram is shown in Figure 5-9. Descriptions of each state are given in the paragraphs below. During normal operation the state machine controls state transitions. When necessary, however, the state can be forced using the DPLLCR2.STATE configuration field.

Whenever the DPLL changes state, the STATE bit in PLL1LSR is set, which can cause an interrupt request if enabled. The current DPLL state can be read from the PLL1SR.STATE.



Figure 5-9. DPLL State Transition Diagram



Notes:

- An input clock is valid when it has no activity alarm, no frequency hard limit alarm, and no phase lock alarm (see the [VALSR1](#) register and the [ISR](#) register).
- All input clocks are continuously monitored for activity and frequency.
- Only the selected reference is monitored for loss of lock.
- Phase lock is declared internally when the DPLL has maintained phase lock continuously for approximately 1 to 2 seconds.
- To simplify the diagram, the phase-lock timeout period is always shown as 100s, which is the default value of the [PHLKTO](#) register. Longer or shorter timeout periods can be specified as needed by writing the appropriate value to the [PHLKTO](#) register.
- When the selected reference is invalid and the DPLL is not in free-run or digital hold, the DPLL is in a temporary digital hold state.

### 5.6.1.1 Free-Run State

Free-run is the reset default state. In free-run the DPLL output clock is derived from the local oscillator. The frequency of the output clock is a specific multiple of the local oscillator, and the frequency accuracy of the output clock is equal to the frequency accuracy of the master clock plus the frequency offset specified by the [MCFREQ](#) field (see Section 5.3). The state machine transitions from free-run to the prelocked state when a selected reference is available at the input of the DPLL.

### **5.6.1.2 Prelocked State**

The prelocked state provides a 100-second period (default value of **PHLKTO** register) for the DPLL to lock to the selected reference. If phase lock (see Section 5.6.5) is achieved for 2 seconds during this period then the state machine transitions to locked mode.

If the DPLL fails to lock to the selected reference within the phase-lock timeout period specified by **PHLKTO** then a phase lock alarm is raised (corresponding **LOCK** bit set in the **ISR** register), invalidating the input (**ICn** bit goes low in the **VALSR1** register). If the clock selector block determines that another input clock is valid then the DPLL state machine re-enters the prelocked state and tries to lock to the alternate input clock. If no other input clocks are valid for two seconds, then the state machine transitions back to the free-run state. Meanwhile, for the invalidated clock, the phase lock alarm can automatically timeout after an amount of time specified by the **LKATO** register (default 100 seconds) or can be cleared by software writing a 0 to the **LOCK** bit.

In revertive mode (**DPLLCR1.REVERT** = 1), if a higher priority input clock becomes valid during the phase-lock timeout period then the state machine re-enters the prelocked state and tries to lock the higher priority input.

If a phase-lock timeout period longer or shorter than 100 seconds is required for locking, then the **PHLKTO** register must be configured accordingly.

### **5.6.1.3 Locked State**

The DPLL state machine can reach the locked state from the prelocked, prelocked 2, or loss-of-lock states when the DPLL has locked to the selected reference for at least 2 seconds (see Section 5.6.5). In the locked state the output clocks track the phase and frequency of the selected reference.

While in the locked state, if the selected reference is so impaired that an activity alarm is raised (corresponding **ACT** bit set in the **ISR** register), then the selected reference is invalidated (**ICn** bit goes low in the **VALSR1** register), and the state machine immediately transitions to either the prelocked 2 state (if another valid input clock is available) or, after being invalid for 2 seconds, to the digital hold state (if no other input clock is valid).

If loss-of-lock (see Section 5.6.5) is declared while in the locked state then the state machine transitions to the loss-of-lock state.

Any of the GPIO pins can be configured to output a signal that is high when the DPLL is in the locked state and low when the DPLL is in any other state. See the **GPIOSS** registers for details.

### **5.6.1.4 Loss-of-Lock State**

When the loss-of-lock detectors (see Section 5.6.5) indicate loss of phase lock, the state machine immediately transitions from the locked state to the loss-of-lock state. In the loss-of-lock state the DPLL tries for 100 seconds (default value of **PHLKTO** register) to regain phase lock. If phase lock is regained during that period for more than 2 seconds, the state machine transitions back to the locked state.

If, during the phase-lock timeout period specified by **PHLKTO**, the selected reference is so impaired that an activity alarm or a hard frequency limit alarm is raised (corresponding **ACT** or **HARD** bit set in the **ISR** register), then the selected reference is invalidated (**ICn** bit goes low in the **VALSR1** register), and after being invalid for 2 seconds the state machine transitions to either the prelocked 2 state (if another valid input clock is available) or the digital hold state (if no other input clock is valid).

If phase lock cannot be regained by the end of the phase-lock timeout period then a phase lock alarm is raised (corresponding **LOCK** bit set in the **ISR** register), the selected reference is invalidated (**ICn** bit goes low in **VALSR** registers), and the state machine transitions to either the prelocked 2 state (if another valid input clock is available) or, after being invalid for 2 seconds, to the digital hold state (if no other input clock is valid). The phase lock alarm can automatically timeout after an amount of time specified by the **LKATO** register (default 100 seconds) or can be cleared by software writing a 0 to the **LOCK** bit.



Note that if [PHLKTO](#)[5:0]=0 then the phase lock timeout is disabled, and the DPLL can remain indefinitely in the loss-of-lock state. Also, if [LKATO](#)[5:0]=0, the lock alarm timeout is disabled, and any phase lock alarm remains active until cleared by software writing a 0 to the LOCK bit.

### 5.6.1.5 Prelocked 2 State

The prelocked and prelocked 2 states are similar. The prelocked 2 state provides a 100-second period (default value of [PHLKTO](#) register) for the DPLL to lock to the new selected reference. If phase lock (see Section 5.6.5) is achieved for more than 2 seconds during this period then the state machine transitions to locked mode.

If the DPLL fails to lock to the new selected reference within the phase-lock timeout period specified by [PHLKTO](#) then a phase lock alarm is raised (corresponding LOCK bit set in the [ISR](#) register), invalidating the input (ICn bit goes low in the [VALSR1](#) register). If the clock selector block determines that another input clock is valid then the state machine re-enters the prelocked 2 state and tries to lock to the alternate input clock. If no other input clocks are valid for 2 seconds, the state machine transitions to the digital hold state. Meanwhile, for the invalidated clock, the phase lock alarm can automatically timeout after an amount of time specified by the [LKATO](#) register (default 100 seconds) or can be cleared by software writing a 0 to the LOCK bit.

In revertive mode ([DPLLCR1](#).REVERT = 1), if a higher priority input clock becomes valid during the phase-lock timeout period then the state machine re-enters the prelocked 2 state and tries to lock to the higher priority input.

If a phase-lock timeout period longer or shorter than 100 seconds is required for locking, then the [PHLKTO](#) register must be configured accordingly.

### 5.6.1.6 Digital Hold State

The device reaches the digital hold state when it declares its selected reference invalid for 2 seconds and has no other valid input clocks available. During digital hold the DPLL is not phase locked to any input clock. Instead the frequency is set by the [HOFREQ](#) field.

For free-run operation the [HOFREQ](#) field can be set to zero. When this is done, the output frequency accuracy is generated with the accuracy of the external oscillator frequency, modified by the setting of the [MCFREQ](#) field.

For numerically controlled oscillator (NCO) operation, the [HOFREQ](#) field can be controlled by system software.

## 5.6.2 Bandwidth

The bandwidth of the DPLL is configured by the [DPLLCR3](#).ABW and [DPLLCR4](#).LBW fields for various values from 4Hz to 400Hz. The [DPLLCR6](#).AUTOBW bit controls automatic bandwidth selection. When AUTOBW = 1, the DPLL uses the ABW bandwidth during acquisition (not phase locked) and the LBW bandwidth when phase locked. When AUTOBW = 0 the DPLL uses the LBW bandwidth all the time, both during acquisition and when phase locked.

When [DPLLCR6](#).LIMINT = 1, the DPLL's integral path is limited (i.e., frozen) when the DPLL reaches minimum or maximum frequency. Setting LIMINT = 1 minimizes overshoot when the DPLL is pulling in.

### 5.6.3 Damping Factor

The damping factor for the DPLL is configured in the [DPLLCR3](#).ADAMP and [DPLLCR4](#).LDAMP fields. The reset default damping factor is chosen to give a maximum jitter/wander gain peak of approximately 0.1dB. Available settings are a function of DPLL bandwidth (section 5.6.2). See Table 5-6.

**Table 5-6. Damping Factors and Peak Jitter/Wander Gain**

BANDWIDTH (Hz)	DAMP[2:0] VALUE	DAMPING FACTOR	GAIN PEAK (dB)
0.1 to 4	1, 2, 3, 4, 5	5	0.1

BANDWIDTH (Hz)	DAMP[2:0] VALUE	DAMPING FACTOR	GAIN PEAK (dB)
8	1	2.5	0.2
	2, 3, 4, 5	5	0.1
18	1	1.2	0.4
	2	2.5	0.2
	3, 4, 5	5	0.1
35	1	1.2	0.4
	2	2.5	0.2
	3	5	0.1
	4, 5	10	0.06
70 to 400	1	1.2	0.4
	2	2.5	0.2
	3	5	0.1
	4	10	0.06
	5	20	0.03

#### 5.6.4 Phase Detectors

Phase detectors are used to compare the DPLL's feedback clock with its input clock. Two phase detectors are available in the DPLL:

Phase/frequency detector (PFD)

Multicycle phase detector (MCPD) for large input jitter tolerance and/or faster lock times

These detectors can be used in combination to give fine phase resolution combined with large jitter tolerance. As with the rest of the DPLL logic, the phase detectors operate at input frequencies up to 77.76MHz. The multicycle phase detector detects and remembers phase differences of many cycles (up to 8191UI). When locking to 8kHz or lower, the normal phase/frequency detector is always used.

The DPLL phase detectors can be configured for normal phase/frequency locking ( $\pm 360^\circ$  capture) or nearest-edge phase locking ( $\pm 180^\circ$  capture). With nearest-edge locking the phase detectors are immune to occasional missing clock cycles. The DPLL automatically switches to nearest-edge locking when the multicycle phase detector is disabled and the PFD determines that phase lock has been achieved. Setting [DPLLCR5.D180 = 1](#) disables nearest-edge locking and forces the DPLL to use phase/frequency locking.

The multicycle phase detector is enabled by setting [DPLLCR5.MCPDEN = 1](#). The range of the MCPD—from  $\pm 1$ UI up to  $\pm 8191$ UI—is configured in the [PHLIM.COARSELIM](#) field. The MCPD tracks phase position over many clock cycles, giving high jitter tolerance.

When [DPLLCR5.USEMCPD = 1](#), the MCPD is used in the DPLL loop, giving faster pull-in but more overshoot. In this mode the loop has behavior similar to a scenario where the input clock is divided down and the lock frequency is 8kHz or 2kHz. In both cases large phase differences contribute to the dynamics of the loop. When enabled by [MCPDEN = 1](#), the MCPD tracks the phase position whether or not it is used in the DPLL loop.

When the input clock is divided before being sent to the phase detector, the divider output clock edge gets aligned to the feedback clock edge before the DPLL starts to lock to a new input clock signal or after the input clock signal has a temporary signal loss. This helps ensure locking to the nearest input clock edge which reduces output transients and decreases lock times.

#### 5.6.5 Loss of Phase Lock Detection

Loss of phase lock can be triggered by any of the following:

- The fine phase limit
- The coarse limit
- Hard frequency limit
- Inactivity detector

The fine phase limit is enabled by setting **DPLLCR5.FLEN** = 1 and configured in the **PHLIM.FINELIM** field.

The coarse phase limit is enabled by setting **DPLLCR5.CLLEN** = 1 and configured in the **PHLIM.COARSELIM** field. This coarse phase limit is part of the multicycle phase detector (MCPD) described in Section 5.6.4. The **COARSELIM** field sets both the MCPD range and the coarse phase limit, since the two are equivalent. If loss of phase lock should not be declared for multiple-UI input jitter then the fine phase limit should be disabled and the coarse phase limit should be used instead.

The hard frequency limit detector is enabled by setting **DPLLCR5.FLLOL** = 1. The hard limit is configured in the **HRDLIM** field. When the DPLL frequency reaches the hard limit, loss-of-lock is declared. The DPLL also has a frequency soft limit specified in the **SOFTLIM** register. Exceeding the soft frequency limit causes the **SOFT** status bit in the **PLL1SR** register to be set but does not cause loss-of-lock to be declared.

The inactivity detector is enabled by setting **DPLLCR5.NALOL** = 1. When this detector is enabled the DPLL declares loss-of-lock after the selected reference has a few missing clock cycles (see Table 5-4).

When the DPLL declares loss of phase lock, the **PALARM** bit is set in **PLL1SR**, and the state machine immediately transitions to the loss-of-lock state, which sets the **STATE** bit in the **PLL1LSR** register and causes an interrupt request if enabled.

### **5.6.6 Manual Phase Adjustment**

The **OFFSET** field can be used to adjust the phase of the DPLL's output clock with respect to its input clock. Output phase offset can be adjusted over a  $\pm 200$ ns range in 6ps increments. This phase adjustment occurs in the feedback clock so that the output clocks are adjusted to compensate. The rate of change is therefore a function of DPLL bandwidth. Simply writing to the **OFFSET** registers causes a change in the input to output phase, which can be considered to be a delay adjustment. Changing the **OFFSET** adjustment while in free-run or digital hold state will not cause an output phase offset until the DPLL enters one of the locking states.

### **5.6.7 Frequency and Phase Measurement**

If the DPLL is otherwise unused, it can be employed as a high-resolution frequency and phase measurement system. As described in Section 5.5.2.1, the input clock frequency monitors report measured frequency with  $\sim 1.25$ ppm resolution. For higher resolution frequency measurement, the DPLL can be used. When the DPLL is locked to an input clock, the frequency of the DPLL, and therefore of the input clock, is reported in the **FREQ** field. This frequency measurement has a resolution of  $3.7427766 \times 10^{-8}$ ppm over a  $\pm 80$ ppm range. The value read from the **FREQ** field is the DPLL's integral path value, which is an averaged measurement with an averaging time inversely proportional to DPLL bandwidth. The reference for frequency measurements is the frequency of the master clock signal plus the frequency offset specified by the **MCFREQ** field.

DPLL phase measurements can be read from the **PHASE** field. This field indicates the phase difference between the input clock and the feedback clock. This phase measurement has a resolution of approximately 0.707 degrees and is internally averaged with a -3dB attenuation point of approximately 100Hz. Thus for low DPLL bandwidths the **PHASE** field gives input phase wander in the frequency band from the DPLL corner frequency up to 100Hz. This information could be used by software to compute a crude MTIE measurement.

### **5.6.8 Input Wander and Jitter Tolerance**

Wander is tolerated up to the point where wander causes an apparent long-term frequency offset larger than the limits specified in the **ICRHLIM** register. In such a situation the input clock would be declared invalid. When using the  $\pm 360^\circ/\pm 180^\circ$  phase/frequency detector, jitter can be tolerated up to the point of eye closure. The multicycle phase detector (see Section 5.6.4) should be used for high jitter tolerance.

### **5.6.9 Jitter and Wander Transfer**

The transfer of jitter and wander from the selected reference to the output clocks has a programmable transfer function that is determined by the DPLL bandwidth. (See section 5.6.2.) The 3dB corner frequency of the jitter transfer function can be set to any of a number of values from 4Hz to 400Hz.

During locked mode, the transfer of wander from the local oscillator clock (connected to the MCLKOSC pins) to the output clocks is not significant as long as the DPLL bandwidth is set high enough to allow the DPLL to quickly compensate for oscillator frequency changes. During free-run and digital hold states, local oscillator wander has a much more significant effect. See section 5.3.1.

### **5.6.10 Output Jitter and Wander**

Several factors contribute to jitter and wander on the output clocks, including:

- Jitter and wander amplitude on the selected reference (while in the locked state)
- The jitter/wander transfer characteristic of the device (while in the locked state)
- The jitter and wander on the local oscillator clock signal (especially wander while in the free-run or digital hold states)

The DPLL has programmable bandwidth (see Section 5.6.2). With respect to jitter and wander, the DPLL behaves as a low-pass filter with a programmable pole. The bandwidth of the DPLL is normally set low enough to strongly attenuate jitter. The wander and jitter attenuation depends on the DPLL bandwidth chosen.

Over time frequency changes in the local oscillator can cause a phase difference between the selected reference and the output clocks. This is especially true at lower frequency DPLL bandwidths because the DPLL's rate of change may be slower than the oscillator's rate of change. Oscillators with better stability will minimize this effect.

### **5.6.11 $\pm 160$ ppm Tracking Range Mode**

The DPLL has an optional mode where the resolution and range of all internal frequency offsets are scaled up by a factor of two. This mode is useful in systems where DPLL pull-in and hold-in range must be larger than the normal  $\pm 80$ ppm maximum. To enable this mode, set [DPLLCR1.PPM160](#).

When this mode is enabled the value of an lsb and the range of the following fields are doubled: [HRDLIM](#), [SOFTLIM](#), [HOFREQ](#) and [FREQ](#). In addition the DPLL bandwidths listed in [DPLLCR3.ABW](#) and [DPLLCR4.LBW](#) are doubled, and the damping factors listed in [DPLLCR3.ADAMP](#) and [DPLLCR4.LDAMP](#) are multiplied by the square root of 2.

## **5.7 APLL Configuration**

### **5.7.1 Input Selection and Frequency**

#### **5.7.1.1 APLL-Only Mode**

In APLL-Only mode ([APLLCR2.APLLMUX=0xx](#)) the APLLs lock to the crystal oscillator, the external oscillator connected to the MCLKOSCP/N pins, the IC1 clock signal or the IC2 clock signal. See section 5.2.1 for details and diagrams.

The input to each APLL can be controlled by the SS input pin or by the [APLLCR2.APLLMUX](#) register field. When [APLLCR2.EXTSW=0](#), the [APLLCR2.APLLMUX](#) register field controls the APLL input mux.

When [APLLCR2.EXTSW=1](#), the SS input pin controls the APLL input mux. When SS=0, the mux selects the input specified by [APLLCR2.APLLMUX](#). When SS=1, the mux selects the input specified by [APLLCR2.ALTMUX](#).

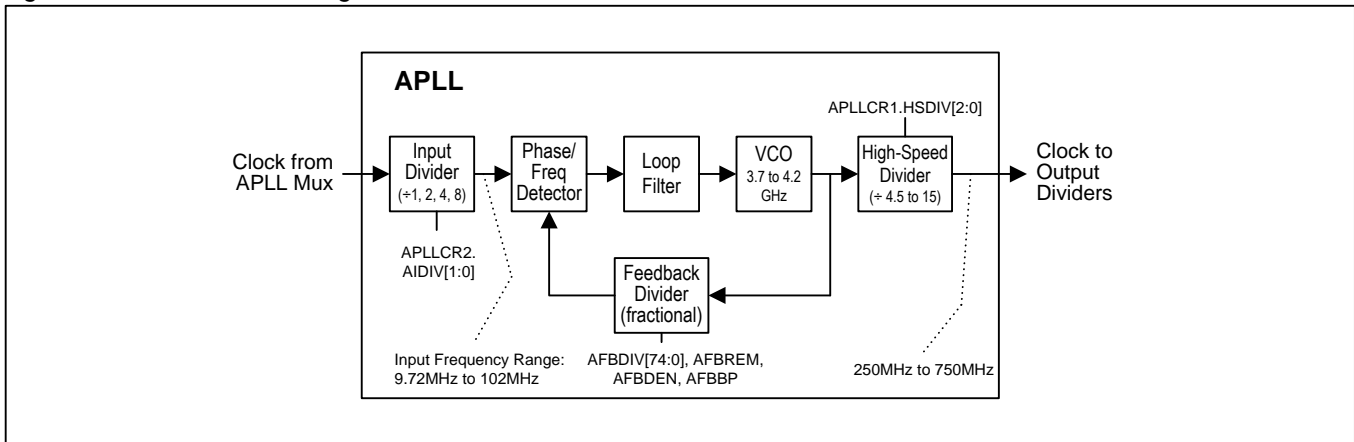
In APLL-Only mode the APLL input signal must be in the range 9.72MHz to 102.4MHz. For faster input frequencies, the APLL's input divider can be configured to divide the signal by 2, 4 or 8 ([APLLCR2.AIDIV](#)) to get a frequency in the APLL's locking range. Note the higher APLL input frequencies give lower output jitter, all else being equal.

### 5.7.1.2 DPLL+APLL Mode

In DPLL+APLL mode ([APLLCR2.APLLMUX=100](#)) APLL1 locks to the DPLL output clock signal while APLL2 synthesizes the master clock for the DPLL and the input block. The DPLL uses digital frequency synthesis (DFS) to synthesize its output clock. The DFS block has two modes of operation. When [DFSCR1.DFSFREQ≠1111](#), the DFS block synthesizes one of 15 common telecom, datacom or Nx10MHz frequencies. When [DFSFREQ=1111](#), the DFS block is configured for programmable DFS mode in which it can synthesize any multiple of 2kHz from 38.88MHz to 77.76MHz. The MAX24605/MAX24610 EV kit software makes configuration in programmable DFS mode easy.

### 5.7.2 Output Frequency

Figure 5-10. APLL Block Diagram



An APLL is enabled when [APLLCR1.APllen=1](#). The APLLs have a fractional-N architecture and therefore can produce output frequencies that are either integer or non-integer multiples of the input clock frequency. [Figure 5-10](#) 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 [APLLCR1.HSDIV](#) field specifies how the VCO frequency is divided down by the high-speed divider. Dividing by six is the typical setting to produce 622.08MHz for SDH/SONET or 625MHz for Ethernet applications. The HSDIV divider produces a clock signal with a 50% duty cycle for all divider values including odd numbers.

Internally, the exact APLL feedback divider value is expressed in the form  $\text{AFBDIV} + \text{AFBREM} / \text{AFBDEN} * 2^{-(66-\text{AFBBP})}$ . This feedback divider value must be chosen such that  $\text{APLL\_input\_frequency} * \text{feedback\_divider\_value}$  is in the operating range of the VCO (as specified in [Table 8-7](#)). The AFBDIV term is a fixed-point number with 9 integer bits and a configurable number of fractional bits (up to 66, as specified by [AFBBP](#)). Typically [AFBBP](#) is set to 42 to specify that [AFBDIV](#) has  $66 - 42 = 24$  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,  $\text{VCO\_frequency} = \text{input\_frequency} * M * N / D$ . An example of this is multiplying 77.76MHz from the DPLL by  $M=48$  and scaling by  $N / D = 255 / 237$  for forward error correction applications.

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

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

$$\text{AFBDEN} = D \quad (3)$$

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

$$\text{AFBBP} = 66 - 24 = 42 \quad (5)$$

The trunc() function returns only the integer portion of the number. The round() function rounds the number to the nearest integer. In Equation (1), **AFBDIV** is set to the full-precision feedback divider value,  $M * N / D$ , truncated after the 24<sup>th</sup> 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. Finally, in Equation (5) **AFBBP** is set to  $66 - 24 = 42$  to correspond with **AFBDIV** having 24 fractional bits.

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  $N = N_1 \times N_2 \times \dots \times N_n$  and the denominators are multiplied together to get  $D = D_1 \times D_2 \times \dots \times D_n$ .

Note that one easy way to calculate the exact values to write to the APLL registers is to use the MAX24605/MAX24610 evaluation board software, available on the MAX24605/MAX24610 page of Microsemi's 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 **APLLCR1.APPLEN** must be changed from 0 to 1 to cause the APLL to reacquire lock with the new settings.

## 5.8 Output Clock Configuration

The MAX24605 has five output clock signals. The MAX24610 has ten output clock signals. Each output has individual divider, enable and signal format controls.

### 5.8.1 Enable, Signal Format, Voltage and Interfacing

Using the **OCCR2.OCSF** register field, each output pair can be disabled or configured as a CML output, an HSTL output, 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 OCxNEG pin can be disabled, in phase or inverted vs. the OCxPOS pin. In CML mode the normal 800mV  $V_{OD}$  differential voltage is available as well as a lower-power 400mV  $V_{OD}$ . All of these options are specified by **OCCR2.OCSF**.

Device clock outputs are grouped into four banks as shown below:

Bank	MAX24605 Outputs	MAX24610 Outputs
A	OC1, OC2	OC1, OC2
B	OC3	OC3, OC4, OC5
C	OC8	OC6, OC7, OC8
D	OC10	OC9, OC10

Each bank has its own power supply and ground pin to allow CMOS or HSTL signal swing from 1.5V to 3.3V for glueless interfacing to neighboring components. 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 differential (CML) outputs must have a bank power supply of 3.3V. If other outputs in that bank are configured for CMOS operation, the CMOS outputs will also have a 3.3V power supply. However, CMOS outputs from that bank can be externally attenuated using resistor divider networks if needed.

The differential outputs can be easily interfaced to LVDS, LVPECL, CML, HSTL and other differential inputs on neighboring ICs using a few external passive components. See [App Note HFAN-1.0](#) for details.

### 5.8.2 Frequency Configuration

The frequency of each output is determined by which APLL it is connected to, the configuration the APLL and the per-output dividers. Each bank of outputs can be connected to either APLL1 or APLL2. The register fields to control the bank muxes are AMUX, BMUX, CMUX and DMUX, respectively, in the **MCR1** register.



Each output has two output dividers, a 7-bit medium-speed divider (**OCCR1**.MSDIV) and a 24-bit output divider (**OCDIV** 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.

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 some outputs to get 156.25MHz, divided by five for other outputs to get 125MHz, and divided by 25 for other outputs 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.

Various divisors of the APLL output clock can be brought out on any combination of outputs. For the very lowest output jitter, however, frequencies such as 156.25MHz and 125MHz that are not integer divisors of one another should come from separate banks whenever possible.

### **5.8.3 Phase Adjustment**

The phase of an output signal can be shifted by 180° by setting **OCCR1**.POL=1. In addition, the phase can be adjusted using the **OCCR3**.PHADJ register field. The adjustment is in units of APLL output clock cycles. For example, if the APLL output frequency is 625MHz then one APLL output clock cycle is 1.6ns, the smallest phase adjustment is 0.8ns, and the adjustment range is ±5.6ns.

## 5.9 Microprocessor Interface

The device presents a SPI slave port on the CS\_N, SCLK, SDI, and SDO 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. The device is always a slave. 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 SDI pin and transmits serial data on the SDO pin. SDO 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 SDI and SDO.

**Clock Polarity and Phase.** The device latches data on SDI on the rising edge of SCLK and updates data on SDO on the falling edge of SCLK. SCLK does not have to toggle between accesses, i.e., when CS\_N is high.

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

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

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. The external EEPROM memory, if present, is accessible when **EESEL**=1. See section 6.1.3. After driving CS\_N low, the bus master transmits the read command followed by the 16-bit register address. The device then responds with the requested data byte on SDO, 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 SDO, increment its address counter, and prefetch the following byte. The read transaction is completed when the bus master drives CS\_N high. See Figure 5-11.

**Register Write Transactions.** The device registers are accessible when **EESEL**=0. After driving CS\_N 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 SDI, 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 CS\_N high. See Figure 5-13.

**EEPROM Writes.** The external EEPROM memory, if present, is accessible when **EESEL**=1. After driving CS\_N low, the bus master transmits the write enable command and then drives CS\_N high to set the internal write enable latch. The bus master then drives CS\_N low again and transmits the write command followed by the 16-bit register 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 SDI, 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 boundary (i.e. when the five least-significant address bits are 11111). When the bus master drives CS\_N high, the device transfers the data in the page buffer to the appropriate page in the EEPROM memory. See Figure 5-12 and Figure 5-13.

**EEPROM Read Status.** After the bus master drives CS\_N 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 CS\_N low, the bus master transmits the Read Status command. The device then responds with the status byte on SDO. 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 CS\_N 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. 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 SDI and SDO Together.** Because communication between the bus master and the device is half-duplex, the SDI and SDO pins can be wired together externally to reduce wire count. To support this option, the bus master must not drive the SDI/SDO line when the device is transmitting.

**AC Timing.** See [Table 8-13](#) and [Figure 8-4](#) for AC timing specifications for the SPI interface.

Figure 5-11. SPI Read Transaction Functional Timing

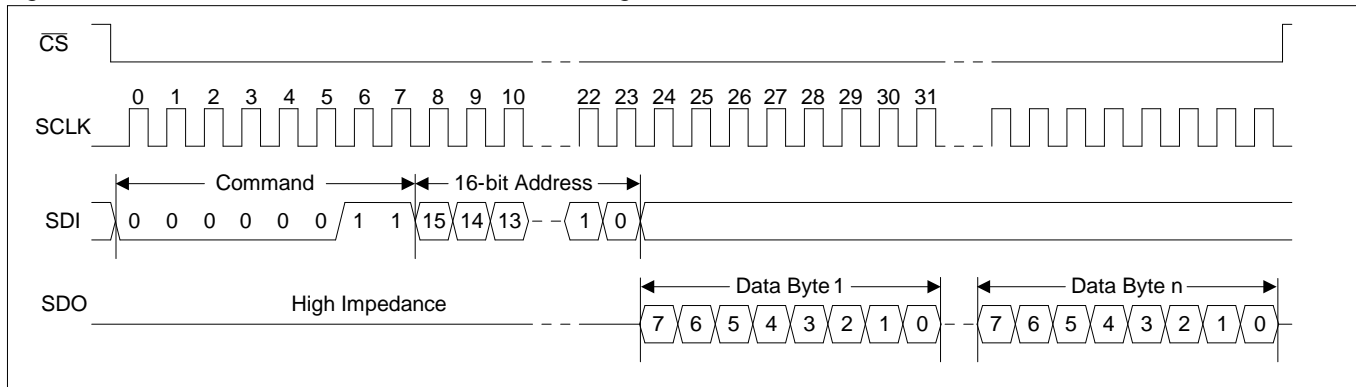


Figure 5-12. SPI Write Enable Transaction Functional Timing

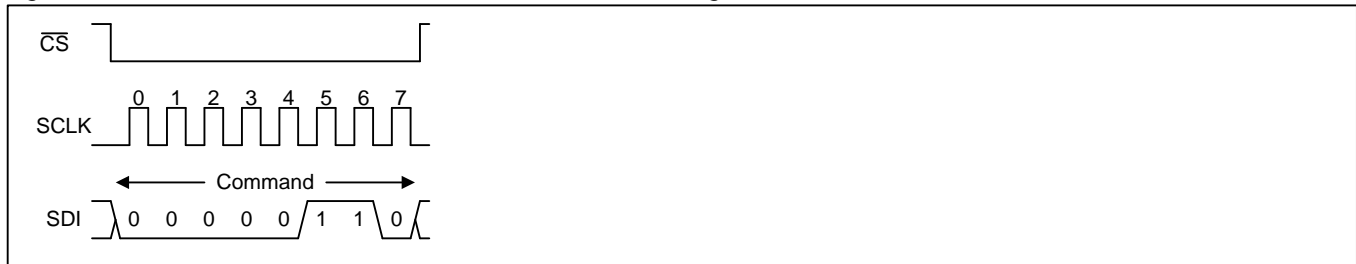
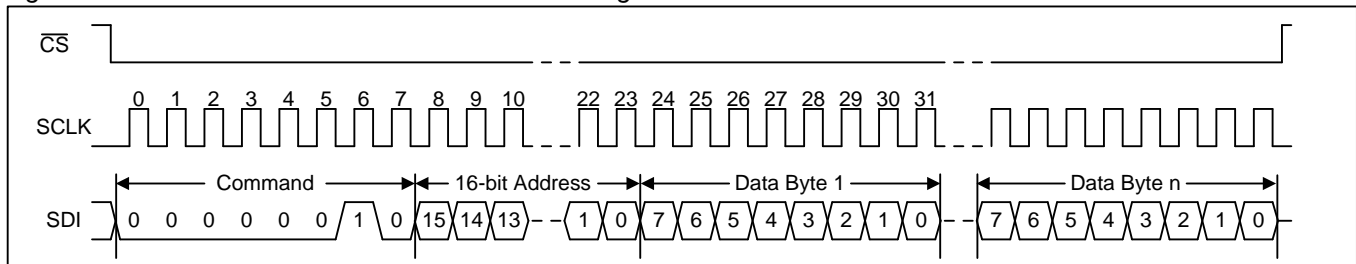


Figure 5-13. SPI Write Transaction Functional Timing



### **5.10 Reset Logic**

The device has three reset controls: the RST\_N pin, the RST bit in [MCR1](#), and the JTAG reset pin JTRST\_N. The RST\_N pin asynchronously resets the entire device, except for the JTAG logic. When the RST\_N pin is low all internal registers are reset to their default values, including those fields which latch their default values from, or based on, the states of configuration input pins when the RST\_N goes high. **The RST\_N pin must be asserted once after power-up while the external oscillator is stabilizing.** Reset should be asserted for at least 100ns.

The [MCR1](#).RST bit resets the entire device (except for the microprocessor interface, the JTAG logic and the RST bit itself), but when RST is active, the register fields with pin-programmed defaults do not latch their values from, or based on, the corresponding input pins. Instead these fields are reset to the default values that were latched when the RST\_N pin was last active.

Microsemi recommends holding RST\_N low while the external oscillator starts up and stabilizes. An incorrect reset condition could result if RST\_N is released before the oscillator has started up completely.

**Important:** System software must wait at least 100µs after reset (RST\_N pin or RST bit) is deasserted before initializing the device as described in section [5.12](#).

### **5.11 Power-Supply Considerations**

Due to the multi-power-supply nature of the device, some I/Os have parasitic diodes between a <3.3V supply and a 3.3V 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 <3.3V supply and the 3.3V supply to force the 3.3V supply to be within one parasitic diode drop of the <3.3V supply. The second method is to ramp up the 3.3V supply first and then ramp up the <3.3V supply.

### **5.12 Initialization and EEPROM Configuration Memory**

After power-up or reset, a series of writes must be done to the device to tune it for optimal performance. This series of writes is called the initialization script. Each die revision has a different initialization script. Download the latest initialization scripts from the the MAX24605/MAX24610 page of Microsemi's website or contact Microsemi timing products technical support. If an external EEPROM is used to store configuration information, the initialization script must be part of the configuration script stored in the EEPROM. The MAX24605/MAX24610 EV kit software automatically includes the correct initialization script in configuration scripts it creates.

The device reads the configuration script from the external EEPROM using a SPI interface consisting of the ECS\_N, ESCLK, ESDI and ESDO pins. When the EV kit software creates the configuration script, the software stores a value in EEPROM memory to specify the speed of the EEPROM's SPI interface. Later, when the device begins self-configuration, it first accesses the EEPROM using very slow timing (<1MHz) to read the speed value. Then, after learning the speed of the EEPROM, the device reads the EEPROM at the specified speed to minimize the self-configuration time.

The external EEPROM component must have a SPI interface, an industry-standard SPI EEPROM command set, and a 32-byte page size to be compatible with the MAX24605 and MAX24610. Compatible products are available from several vendors. Example part numbers are AT25160B from Atmel and 25LC160D from Microchip (both 16kbit). Minimum EEPROM memory size is 8kbit. Recommended EEPROM memory size is 16kbit.

Note that the following register bits must be changed from their power-up default values for proper operation of the device:

- [DPLLCR2](#) bit 6 must be changed from 0 to 1
- [DPLLCR2](#) bit 4 must be changed from 0 to 1
- [DPLLCR6](#) bit 5 must be changed from 1 to 0

## 6. Register Descriptions

The device has an overall address range from 000h to 1FFh. [Table 6-1](#) in [Section 6.2](#) shows the register map. In each register, bit 7 is the MSB and bit 0 is the LSB. Register addresses not listed and 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 6-1](#).

### 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. The LOCK bits in the [ISR](#) register are special-case latched status bits because they cannot create an interrupt request, and a “write 0” is needed to clear them.

#### 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 Bank-Switched Registers

To simplify the device’s register map and documentation, some registers are bank-switched, meaning banks of registers are switched in and out of the register map based on the value of a bank-select control field.

At the top level, The [ESEL](#) register is a bank-select control field that maps the device registers into the memory map at address 0x1 and above when [ESEL](#)=0 and maps the external EEPROM memory, if present, into the memory map at address 0x1 and above when [ESEL](#)=1. The [ESEL](#) register itself is always in the memory map at address 0x0 for both [ESEL](#)=0 and [ESEL](#)=1.

When [ESEL](#)=0 (device registers) the bank-switched sections of the memory map are: the input clock registers, the [APLL](#) registers, and the output clock registers.

The registers for the input clocks are bank-switched in the Input Clock Registers section of [Table 6-1](#). The [ICSEL](#) register is the bank-select control field for the input clock registers.

The registers for the [APLL](#)s are bank-switched in the [APLL](#) Registers section of [Table 6-1](#). The [APLLSEL](#) register is the bank-select control field for the [APLL](#) registers.

The registers for the output clocks are bank-switched in the Output Clock Registers section of [Table 6-1](#). The [OCSEL](#) register is the bank-select control field for the output clock registers.

#### 6.1.4 Multiregister Fields

Multiregister fields—such as [FREQ](#)[31:0] in registers [FREQ1](#) through [FREQ4](#)—must be handled carefully to ensure that the bytes of the field remain consistent. 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. Any reads to the multiregister field that occur during the middle of the multiregister write will read the existing value of the field not the new value in the transfer register.

A read access from a multiregister field is accomplished by reading the registers of the field in order from smallest address to largest. When the first register in the field (i.e. the register with the lowest address) is read, the entire multiregister field is copied to the transfer register. During subsequent reads from the other registers in the multiregister field, the data comes from the transfer register. Any writes to the multiregister field that occur during the middle of the multiregister read will overwrite values in the transfer register.

Each multiregister field has its own transfer register. The same transfer register is used for read and writes. For best results, system software should be organized such that only one software process accesses the device's registers. If two or more processes are allowed to make uncoordinated accesses to the device's registers, their accesses to multiregister fields could interrupt one another leading to incorrect writes and reads of the multiregister fields. The multiregister fields are:

FIELD	REGISTERS	TYPE
MCFREQ[15:0]	<a href="#">MCFREQ1</a> , <a href="#">MCFREQ2</a>	Read/Write
ICN[15:0]	<a href="#">ICN1</a> , <a href="#">ICN2</a>	Read/Write
ICD[15:0]	<a href="#">ICD1</a> , <a href="#">ICD2</a> , <a href="#">ICD3</a> , <a href="#">ICD4</a>	Read/Write
FMEAS[15:0]	<a href="#">FMEAS1</a> , <a href="#">FMEAS2</a>	Read-Only
HRDLIM[9:0]	<a href="#">HRDLIM1</a> , <a href="#">HRDLIM2</a>	Read/Write
OFFSET[15:0]	<a href="#">OFFSET1</a> , <a href="#">OFFSET2</a>	Read/Write
PHASE[15:0]	<a href="#">PHASE1</a> , <a href="#">PHASE2</a>	Read-Only
FREQ[23:0]	<a href="#">FREQ1</a> , <a href="#">FREQ2</a> , <a href="#">FREQ3</a> , <a href="#">FREQ4</a>	Read-Only
HOFREQ[23:0]	<a href="#">HOFREQ1</a> , <a href="#">HOFREQ2</a> , <a href="#">HOFREQ3</a> , <a href="#">HOFREQ4</a>	Read-Only

### **6.1.5 Input Clock Registers and DPLL Registers**

The input clock registers and DPLL registers at addresses 0x50 and above cannot be read or written unless a master clock is provided to the input block and the DPLL. See section [5.2.2](#).

## 6.2 Register Map

**Table 6-1. Register Map**

**Note:** Register names are hyperlinks to register definitions. Underlined fields are read-only.

ADDR	REGISTER	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Global Registers									
00h	EESEL	—	—	—	—	—	—	—	EESEL
01	ID1	ID[7:0]							
02	ID2	ID[15:8]							
03	REV	REV[7:0]							
04	PROT	PROT[7:0]							
05	MCR1	RST	ICBEN	DPLEN	—	AMUX	BMUX	CMUX	DMUX
06	MCR2	XIEN	XOEN	IC1EN	IC2EN	MCEN	—	MCDIV[1:0]	
387	MCR3	—	—	—	—	—	—	MCMUX	—
07	APLLSR	—	A2LKIE	A2LKL	A2LK	—	A1LKIE	A1LKL	A1LK
GPIO Registers									
08	GPCR	GPIO4C[1:0]		GPIO3C[1:0]		GPIO2C[1:0]		GPIO1C[1:0]	
09	GPSR	—	—	—	—	GPIO4	GPIO3	GPIO2	GPIO1
0A	GPIO1SS	POL	OD	REG[2:0]			BIT[2:0]		
0B	GPIO2SS	POL	OD	REG[2:0]			BIT[2:0]		
0C	GPIO3SS	POL	OD	REG[2:0]			BIT[2:0]		
0D	GPIO4SS	POL	OD	REG[2:0]			BIT[2:0]		
APLL Registers									
10	APLLSEL	—	—	—	—	—	—	APLLSEL[1:0]	
11	APLLCR1	APLLEN	APLLBYP	DALIGN	—	HSDIV[3:0]			
12	APLLCR2	AIDIV[1:0]		EXTSW	ALTMUX[1:0]		APLLMUX[2:0]		
22	AFBDIV1	AFBDIV[3:0]				—	—	—	—
23	AFBDIV2	AFBDIV[11:4]							
24	AFBDIV3	AFBDIV[19:12]							
25	AFBDIV4	AFBDIV[27:20]							
26	AFBDIV5	AFBDIV[35:28]							
27	AFBDIV6	AFBDIV[43:36]							
28	AFBDIV7	AFBDIV[51:44]							
29	AFBDIV8	AFBDIV[59:52]							
2A	AFBDIV9	AFBDIV[67:60]							
2B	AFBDIV10	—	AFBDIV[74:68]						
2C	AFBDEN1	AFBDEN[7:0]							
2D	AFBDEN2	AFBDEN[15:8]							
2E	AFBDEN3	AFBDEN[23:16]							
2F	AFBDEN4	AFBDEN[31:24]							
30	AFBREM1	AFBREM[7:0]							
31	AFBREM2	AFBREM[15:8]							
32	AFBREM3	AFBREM[23:16]							
33	AFBREM4	AFBREM[31:24]							
34	AFBBP	AFBBP[7:0]							
Output Clock Registers									
40	OCSEL	—	—	—	—	OCSEL[3:0]			
41	OCCR1	—	MSDIV[6:0]						

ADDR	REGISTER	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
42	<a href="#">OCCR2</a>	—	—	DRIVE[1:0]		OCSF[3:0]			
43	<a href="#">OCCR3</a>	PHADJ[3:0]				—	POL	ASQUEL	DALEN
44	<a href="#">OCDIV1</a>	OCDIV[7:0]							
45	<a href="#">OCDIV2</a>	OCDIV[15:8]							
46	<a href="#">OCDIV3</a>	OCDIV[23:16]							
Input Clock Registers									
50	<a href="#">ICSEL</a>	—	—	—	ICSEL[3:0]				
51	<a href="#">ICCR1</a>	ICEN	POL	IFREQR[1:0]		LKFREQ[3:0]			
52	<a href="#">ICCR2</a>	—	GPIO SQ	FMONCLK[1:0]		—	SOFTEN	HARDEN	FREN
53	<a href="#">ICCR3</a>	—	—	—	NSEN	FMONLEN[3:0]			
54	<a href="#">ICN1</a>	ICN[7:0]							
55	<a href="#">ICN2</a>	ICN[15:8]							
56	<a href="#">ICD1</a>	ICD[7:0]							
57	<a href="#">ICD2</a>	ICD[15:8]							
58	<a href="#">ICD3</a>	ICD[23:16]							
59	<a href="#">ICD4</a>	ICD[31:24]							
5A	<a href="#">ICLBU</a>	ICLBU[7:0]							
5B	<a href="#">ICLBL</a>	ICLBL[7:0]							
5C	<a href="#">ICLBS</a>	ICLBS[7:0]							
5D	<a href="#">ICLBD</a>	—	—	—	—	—	—	ICLBD[1:0]	
5E	<a href="#">ICAH LIM</a>	ICAH LIM[7:0]							
5F	<a href="#">ICRHLIM</a>	ICRHLIM[7:0]							
60	<a href="#">ICSLIM</a>	ICSLIM[7:0]							
61	<a href="#">FMEAS1</a>	FMEAS[7:0]							
62	<a href="#">FMEAS2</a>	FMEAS[15:8]							
63	<a href="#">ICCR4</a>	—	—	—	—	—	FMRES	—	—
DPLL Registers									
71	<a href="#">DPLL CR1</a>	EXTSW	UFSW	REVERT	PPM160	FORCE[3:0]			
72	<a href="#">DPLL CR2</a>	—	1	—	—	—	STATE[2:0]		
73	<a href="#">DPLL CR3</a>	ADAMP[2:0]			ABW[4:0]				
74	<a href="#">DPLL CR4</a>	LDAMP[2:0]			LBW[4:0]				
75	<a href="#">DPLL CR5</a>	NALOL	FLLOL	FLEN	CLEN	MCPDEN	USEMCPD	D180	PFD180
76	<a href="#">DPLL CR6</a>	AUTOBW	LIMINT	0	—	—	—	RDAVG[1:0]	
79	<a href="#">PHLIM</a>	—	FINELIM[2:0]			COARSELIM[3:0]			
7A	<a href="#">PHLKTO</a>	PHLKTOM[1:0]		PHLKTO[5:0]					
7B	<a href="#">LKATO</a>	LKATOM[1:0]		LKATO[5:0]					
7C	<a href="#">HRDLIM1</a>	HRDLIM[7:0]							
7D	<a href="#">HRDLIM2</a>	HRDLIM[15:8]							
7E	<a href="#">SOFTLIM</a>	SOFTLIM[7:0]							
80	<a href="#">OFFSET1</a>	OFFSET[7:0]							
81	<a href="#">OFFSET2</a>	OFFSET[15:8]							
82	<a href="#">VALCR1</a>	—	—	—	—	—	—	IC2	IC1
83	<a href="#">IPR1</a>	PRI2[3:0]				PRI1[3:0]			
85	<a href="#">PTAB1</a>	REF1[3:0]				SELREF[3:0]			
86	<a href="#">PTAB2</a>	—	—	—	—	REF2[3:0]			
87	<a href="#">PHASE1</a>	PHASE[7:0]							
88	<a href="#">PHASE2</a>	PHASE[15:8]							
89	<a href="#">FREQ1</a>	FREQ[7:0]							
8A	<a href="#">FREQ2</a>	FREQ[15:8]							

ADDR	REGISTER	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
8B	<a href="#">FREQ3</a>	<a href="#">FREQ[23:16]</a>							
8C	<a href="#">FREQ4</a>	<a href="#">FREQ[31:24]</a>							
8D	<a href="#">DFSCR1</a>	<a href="#">DFSFREQ[3:0]</a>				—	—	—	—
8E	<a href="#">MCFREQ1</a>	<a href="#">MCFREQ[7:0]</a>							
8F	<a href="#">MCFREQ2</a>	<a href="#">MCFREQ[15:8]</a>							
90	<a href="#">MCDNOM1</a>	<a href="#">MCDNOM[7:0]</a>							
91	<a href="#">MCDNOM2</a>	<a href="#">MCDNOM[15:8]</a>							
92	<a href="#">MCDNOM3</a>	<a href="#">MCDNOM[23:16]</a>							
93	<a href="#">MCDNOM4</a>	—	—	—	—	—	—	<a href="#">MCDNOM[25:24]</a>	
94	<a href="#">MCINOM1</a>	<a href="#">MCINOM[7:0]</a>							
95	<a href="#">MCINOM2</a>	<a href="#">MCINOM[15:8]</a>							
96	<a href="#">MCINOM3</a>	—	—	—	—	—	—	—	<a href="#">MCINOM16</a>
97	<a href="#">MCAC1</a>	<a href="#">MCAC[7:0]</a>							
98	<a href="#">MCAC2</a>	—	—	—	—	—	—	—	<a href="#">MCAC[8]</a>
9C	<a href="#">HOFREQ1</a>	<a href="#">HOFREQ[7:0]</a>							
9D	<a href="#">HOFREQ2</a>	<a href="#">HOFREQ[15:8]</a>							
9E	<a href="#">HOFREQ3</a>	<a href="#">HOFREQ[23:16]</a>							
9F	<a href="#">HOFREQ4</a>	<a href="#">HOFREQ[31:24]</a>							
DPLL and Input Block Status Registers and Interrupt Enables									
A0	<a href="#">PLL1SR</a>	—	—	—	<a href="#">PALARM</a>	<a href="#">SOFT</a>	<a href="#">STATE[2:0]</a>		
A1	<a href="#">PLL1LSR</a>	<a href="#">MCFAIL</a>	—	—	<a href="#">STATE</a>	<a href="#">SRFAIL</a>	<a href="#">NOIN</a>	—	—
A2	<a href="#">VALSR1</a>	—	—	—	—	—	—	<a href="#">IC2</a>	<a href="#">IC1</a>
A3	<a href="#">ICLSR1</a>	—	—	—	—	—	—	<a href="#">IC2</a>	<a href="#">IC1</a>
A4	<a href="#">ISR1</a>	<a href="#">SOFT2</a>	<a href="#">HARD2</a>	<a href="#">ACT2</a>	<a href="#">LOCK2</a>	<a href="#">SOFT1</a>	<a href="#">HARD1</a>	<a href="#">ACT1</a>	<a href="#">LOCK1</a>
A6	<a href="#">PLL1IER</a>	<a href="#">MCFAIL</a>	—	—	<a href="#">STATE</a>	<a href="#">SRFAIL</a>	<a href="#">NOIN</a>	—	—
A7	<a href="#">ICIER1</a>	—	—	—	—	—	—	<a href="#">IC2</a>	<a href="#">IC1</a>

## 6.3 Register Definitions

### 6.3.1 Global 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
<b>Name</b>	—	—	—	—	—	—	—	EESEL
<b>Default</b>	0	0	0	0	0	0	0	0

**Bit 0: EEPROM Memory Select (EESEL).** This bit is a bank-select that specifies whether device register space or external EEPROM memory is mapped into addresses 0x1 and above. See sections 5.9 and 6.1.3.

0 = Device registers

1 = EEPROM memory

**Register Name:** ID1  
**Register Description:** Device Identification Register, LSB  
**Register Address:** 01h

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Name	ID[7:0]							
Default	see below							

**Bits 7 to 0: Device ID (ID[7:0]).** The full 16-bit ID field spans this register and [ID2](#).

MAX24605: ID[15:0] = 0x00C8.

MAX24610: ID[15:0] = 0x00C9.

**Register Name:** ID2  
**Register Description:** Device Identification Register, MSB  
**Register Address:** 02h

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Name	ID[15:8]							
Default	0	0	0	0	0	0	0	0

**Bits 7 to 0: Device ID (ID[15:8]).** See the [ID1](#) register description.

**Register Name:** REV  
**Register Description:** Device Revision Register  
**Register Address:** 03h

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Name	REV[7:0]							
Default	0	0	0	0	0	0	0	0

**Bits 7 to 0: Device Revision (REV[7:0]).** Contact the factory to interpret this value and determine the latest revision.

**Register Name:** PROT  
**Register Description:** Protection Register  
**Register Address:** 04h

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Name	PROT[7:0]							
Default	1	0	0	0	0	1	0	1

**Bits 7 to 0: Protection Control (PROT[7:0]).** This field can be used to protect the rest of the register set from inadvertent writes. In protected mode writes to all other registers are ignored. In single unprotected mode, one register (other than PROT) can be written, but after that write the device reverts to protected mode (and the value of PROT is internally changed to 00h). In fully unprotected mode all register can be written without limitation. See section [5.1](#).

1000 0101 = Fully unprotected mode

1000 0110 = Single unprotected mode

All other values = Protected mode



**Register Name:** MCR1  
**Register Description:** Master Configuration Register 1  
**Register Address:** 05h

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>Name</b>	RST	ICBEN	DPLLEN	—	AMUX	BMUX	CMUX	DMUX
<b>Default</b>	0	0	0	0	0	0	0	0

**Bit 7: Device Reset (RST).** When this bit is high the entire device is held in reset, and all register fields, except the RST bit itself, are reset to their default states. When RST is active, the register fields with pin-programmed defaults do not latch their values from the corresponding input pins. Instead these fields are reset to the default values that were latched from the pins when the RST pin was last active. See section 5.10.

0 = Normal operation

1 = Reset

**Bit 6: Input block Enable (ICBEN).** This field enables or disables the input block. See section 5.2.1 and section 5.4. Note that APLL2 also must be enabled and properly configured to operate the input block.

0 = Disable (powered down)

1 = Enable

**Bit 5: DPLL Enable (DPLLEN).** This field enables or disables the DPLL. See section 5.2.1. Note that APLL2 also must be enabled and properly configured to operate the DPLL.

0 = Disable (powered down)

1 = Enable

**Bit 3: Bank A Mux Control (AMUX).** This field selects the source APLL for the bank A outputs. See the block diagram in Figure 2-1 and section 5.8.2.

0 = APLL1

1 = APLL2

**Bit 2: Bank B Mux Control (BMUX).** This field selects the source APLL for the bank B outputs. See the block diagram in Figure 2-1 and section 5.8.2.

0 = APLL1

1 = APLL2

**Bit 1: Bank C Mux Control (CMUX).** This field selects the source APLL for the bank C outputs. See the block diagram in Figure 2-1 and section 5.8.2.

0 = APLL1

1 = APLL2

**Bit 0: Bank D Mux Control (DMUX).** This field selects the source APLL for the bank D outputs. See the block diagram in Figure 2-1 and section 5.8.2.

0 = APLL1

1 = APLL2

**Register Name:** MCR2  
**Register Description:** Master Configuration Register 2  
**Register Address:** 06h

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>Name</b>	XIEN	XOEN	IC1EN	IC2EN	MCEN	—	MCDIV[1:0]	
<b>Default</b>	0	0	0	0	0	0	0	0

**Bit 7: XIN Enable (XIEN).** This field enables/disables the XIN pin and the XO analog circuitry. See section 5.3.2.

0 = Disable

1 = Enable

**Bit 6: XOUT Enable (XOEN).** This field enables and disables the XOUT pin driver. When XOUT is disabled the external crystal is not driven and the XO doesn't oscillate. See section 5.3.2.

0 = Disable (high impedance)

1 = Enable (XO amplifier drives external crystal)

**Bit 5: IC1POS/NEG Enable (IC1EN).** This field enables and disables the IC1POS/NEG differential receiver. The power consumption for the differential receiver is shown in Table 8-2. See section 5.4.

0 = Disable (power down)

1 = Enable

**Bit 4: IC2POS/NEG Enable (IC2EN).** This field enables and disables the IC2POS/NEG differential receiver. The power consumption for the differential receiver is shown in Table 8-2. See section 5.4.

0 = Disable (power down)

1 = Enable

**Bit 3: MCLKOSCP/N Enable (MCEN).** This field enables and disables the MCLKOSCP/N differential receiver. The power consumption for the differential receiver is shown in Table 8-2. See section 5.3.3.

0 = Disable (power down)

1 = Enable

**Bits 1 to 0: Master Clock Divider Value (MCDIV[1:0]).** This field specifies the setting for master clock divider. The master clock divider takes the APLL2 output frequency and divides it down to a master clock frequency in the range 190MHz to 208.333MHz for use by the input block and DPLL. The value MCDIV=0 disables the divider to reduce power consumption and noise generation. See section 5.3.3.

00 = Disabled, output low

01 = Divide by 2

10 = Divide by 3

11 = Divide by 4

**Register Name:** MCR3  
**Register Description:** Master Configuration Register 3  
**Register Address:** 387h

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>Name</b>	—	—	—	—	—	—	MCMUX	—
<b>Default</b>	0	0	0	0	0	1	0	0

**Bit 2:** When this bit is set to 1 the self-configuration controller's oscillator remains enabled after self-configuration is complete. This bit should be set to 0 at the end of the self-configuration script to minimize device output jitter.

**Bit 1: Master Clock Mux (MCMUX).** This bit controls the master clock mux. This mux, shown in Figure 2-1, selects between the master clock output of APLL2 and the signal on the MCLKOSCP/N pins. See section 5.2.2.

0 = APLL2 master clock output

1 = MCLKOSCP/N pins

**Register Name:** APLLSR  
**Register Description:** APLL Status Register  
**Register Address:** 07h

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>Name</b>	—	A2LKIE	A2LKL	<u>A2LK</u>	—	A1LKIE	A1LKL	<u>A1LK</u>
<b>Default</b>	0	0	0	0	0	0	0	0

**Bit 6: APLL2 Lock Interrupt Enable (A2LKIE).** This bit is an interrupt enable for the A2LKL bit.

0 = Mask the interrupt

1 = Enable the interrupt

**Bit 5: APLL2 Lock Latched Status (A2LKL).** This latched status bit is set to 1 when the A2LK status bit changes state (set or cleared). A2LKL is cleared when written with a 1. When A2LKL is set it can cause an interrupt request if the A2LKIE interrupt enable bit is set.

**Bit 4: APLL2 Lock Status (A2LK).** This real-time status bit indicates the lock status of APLL2.

0 = Not locked

1 = Locked

**Bit 2: APLL1 Lock Interrupt Enable (A1LKIE).** This bit is an interrupt enable for the A1LKL bit.

0 = Mask the interrupt

1 = Enable the interrupt

**Bit 1: APLL1 Lock Latched Status (A1LKL).** This latched status bit is set to 1 when the A1LK status bit changes state (set or cleared). A1LKL is cleared when written with a 1. When A1LKL is set it can cause an interrupt request if the A1LKIE interrupt enable bit is set.

**Bit 0: APLL1 Lock Status (A1LK).** This real-time status bit indicates the lock status of APLL1.

0 = Not locked

1 = Locked

### 6.3.2 GPIO Registers

**Register Name:** GPCR  
**Register Description:** GPIO Configuration Register  
**Register Address:** 08h

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Name	GPIO4C[1:0]		GPIO3C[1:0]		GPIO2C[1:0]		GPIO1C[1:0]	
Default	0	0	0	0	0	0	0	0

**Bits 7 to 6: GPIO4 Configuration (GPIO4C[1:0]).** When [DPLLCR1.EXTSW=0](#) and [APLLCR2.EXTSW=0](#), the SS/GPIO4 pin behaves as GPIO4, and this field configures the GPIO4 pin as a general-purpose input a general-purpose output driving low or high, or a status output. When GPIO4 is an input its current state can be read from [GPSR.GPIO4](#). When GPIO4 is a status output, the [GPIO4SS](#) register specifies which status bit is output. When [DPLLCR1.EXTSW=1](#) or [APLLCR2.EXTSW=1](#) the SS/GPIO4 pin behaves as SS and this field is ignored.

- 00 = General-purpose input
- 01 = Status output
- 10 = General-purpose output driving low
- 11 = General-purpose output driving high

**Bits 5 to 4: GPIO3 Configuration (GPIO3C[1:0]).** This field configures the GPIO3 pin as a general-purpose input, a general-purpose output driving low or high, or a status output. When GPIO3 is an input its current state can be read from [GPSR.GPIO3](#). When GPIO3 is a status output, the [GPIO3SS](#) register specifies which status bit is output.

- 00 = General-purpose input
- 01 = Status output
- 10 = General-purpose output driving low
- 11 = General-purpose output driving high

**Bits 3 to 2: GPIO2 Configuration (GPIO2C[1:0]).** This field configures the GPIO2 pin as a general-purpose input, a general-purpose output driving low or high, or a status output. When GPIO2 is an input its current state can be read from [GPSR.GPIO2](#). When GPIO2 is a status output, the [GPIO2SS](#) register specifies which status bit is output.

- 00 = General-purpose input
- 01 = Status output
- 10 = General-purpose output driving low
- 11 = General-purpose output driving high

**Bits 1 to 0: GPIO1 Configuration (GPIO1C[1:0]).** This field configures the GPIO1 pin as a general-purpose input, a general-purpose output driving low or high, or a status output. When GPIO1 is an input its current state can be read from [GPSR.GPIO1](#). When GPIO1 is a status output, the [GPIO1SS](#) register specifies which status bit is output.

- 00 = General-purpose input
- 01 = Status output
- 10 = General-purpose output driving low
- 11 = General-purpose output driving high

**Register Name:** GPSR  
**Register Description:** GPIO Status Register  
**Register Address:** 09h

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Name	—	—	—	—	<u>GPIO4</u>	<u>GPIO3</u>	<u>GPIO2</u>	<u>GPIO1</u>
Default	0	0	0	0	0	0	0	0

**Bit 3: GPIO4 State (GPIO4).** This bit indicates the current state of the GPIO4 pin.

0 = low

1 = high

**Bit 2: GPIO3 State (GPIO3).** This bit indicates the current state of the GPIO3 pin.

0 = low

1 = high

**Bit 1: GPIO2 State (GPIO2).** This bit indicates the current state of the GPIO2 pin.

0 = low

1 = high

**Bit 0: GPIO1 State (GPIO1).** This bit indicates the current state of the GPIO1 pin.

0 = low

1 = high

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

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

**Bit 7: Pin Polarity (POL).**

0 = Normal: GPIO pin has the same polarity as the status bit it follows

1 = Inverted: GPIO pin has inverted polarity vs. the status bit it follows

**Bit 6: Open-Drain Enable (OD).**

0 = Push-Pull: GPIO pin is driven in both inactive and active state

1 = Open-Drain: GPIO pin is driven in the active state but is high impedance in the inactive state

**Bits 5 to 3: Status Register (REG[2:0]).** When [GPCR.GPIO1C=01](#), this field specifies the register of the status bit that GPIO1 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 GPIO1 being driven low.

000 – 100 = The address of the status bit that GPIO1 follows is A0h + REG[2:0]

101 = APLL Lock. The address of the status bit that GPIO follows is 07h ([APLLSR](#) register)

110 = DPLL Lock Output: GPIO1 is active when [PLL1SR.STATE=Locked](#) (100b) and inactive otherwise

111 = Interrupt Output: GPIO1 is active when a latched status bit and its corresponding interrupt enable bit are both active. The POL and OD bits define pin behavior for the active and inactive states.

**Bits 2 to 0: Status Bit (BIT[2:0]).** When [GPCR.GPIO1C=01](#), the REG field above specifies the register of the status bit that GPIO1 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.

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

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

These fields are identical to those in [GPIO1SS](#) except they control GPIO2.

**Register Name:** GPIO3SS  
**Register Description:** GPIO3 Status Select Register  
**Register Address:** 0Ch

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

These fields are identical to those in [GPIO1SS](#) except they control GPIO3.

**Register Name:** GPIO4SS  
**Register Description:** GPIO4 Status Select Register  
**Register Address:** 0Dh

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

These fields are identical to those in [GPIO1SS](#) except they control GPIO4.

### 6.3.3 APLL Registers

**Register Name:** APLLSEL  
**Register Description:** APLL Select Register  
**Register Address:** 10h

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>Name</b>	—	—	—	—	—	—	APLLSEL[1:0]	
<b>Default</b>	0	0	0	0	0	0	0	1

**Bits 1 to 0: APLL Select (APLLSEL[1:0]).** This field is a bank-select control that specifies the APLL for which registers are mapped into the APLL Registers section of [Table 6-1](#). See [Section 6.1.3](#).

00 = {unused value}  
 01 = APLL1  
 10 = APLL2  
 11 = {unused value}

**Register Name:** APLLCR1  
**Register Description:** APLL Configuration Register 1  
**Register Address:** 11h

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>Name</b>	APLLEN	APLLBYP	DALIGN	—	HSDIV[3:0]			
<b>Default</b>	0	0	0	0	0	0	0	0

The APLL registers are bank-selected by the [APLLSEL](#) register. See [section 6.1.3](#).

**Bit 7: APLL Enable (APLLEN).** This bit enables and disables the APLL. When unused, the APLL should be disabled to reduce power consumption. See [section 5.7.2](#).

0 = Disabled  
 1 = Enabled

**Bit 6: APLL Bypass (APLLBYP).** This bit controls an internal bypass mux in the APLL.

0 = Normal APLL operation  
 1 = APLL bypass: the APLL input signal is routed directly to the APLL output

**Bit 5: Align Output Dividers (DALIGN).** A 0 to 1 transition on this bit causes a simultaneous reset of the medium-speed dividers and the output clock dividers for all output clocks where [OCCR1.DALEN](#)=1. After this reset all [DALEN](#)=1 output clocks with frequencies that are exactly integer multiples of one another will be falling-edge aligned. 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 3 to 0: APLL High-Speed Divider (HSDIV[3:0]).** This bit controls the high-speed divider block in the APLL (see [Figure 5-10](#)). See [section 5.7.2](#).

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



**Register Name:** APLLCR2  
**Register Description:** APLL Configuration Register 2  
**Register Address:** 12h

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>Name</b>	AIDIV[1:0]		EXTSW	ALTMUX[1:0]		APLLMUX[2:0]		
<b>Default</b>	0	0	0	0	0	0	0	0

The APLL registers are bank-selected by the [APLLSEL](#) register. See section [6.1.3](#).

**Bits 7 to 6: APLL Input Divider (AIDIV).** This field controls the APLL input divider. See [Figure 5-10](#).

- 00 = Divide by 1
- 01 = Divide by 2
- 10 = Divide by 4
- 11 = Divide by 8

**Bit 5: APLL External Switching Mode (EXTSW).** This bit enables APLL external reference switching mode. In this mode, if the SS pin is low the APLL input mux is controlled by [APLLCR2.APLLMUX](#). If the SS pin is high the APLL input mux is controlled by [APLLCR2.ALTMUX](#). See section [5.7.1.1](#).

**Bits 4 to 3: APLL Alternate Mux Control (ALTMUX[1:0]).** When [APLLCR2.EXTSW](#)=0 this field is ignored. When [APLLCR2.EXTSW](#)=1 and the SS pin is high this field controls the APLL input mux. See section [5.7.1.1](#).

- 00 = IC1 input
- 01 = IC2 input
- 10 = Crystal oscillator (XO) block if crystal is connected, otherwise XIN input
- 11 = MCLKOSCP/N pins

**Bits 2 to 0: APLL Mux Control (APLLMUX[2:0]).** By default this field controls the APLL input mux. See the block diagram in [Figure 2-1](#) for the location of this mux. When [APLLCR2.EXTSW](#)=1 and the SS pin is high, this field is ignored, and the APLL's clock source is specified by [APLLCR2.ALTMUX](#). See section [5.7.1.1](#). When [APLLMUX](#)≠100 for APLL1, the input block and DPLL are bypassed and can be powered down. See section [5.2](#).

- 000 = IC1 input
- 001 = IC2 input
- 010 = Crystal oscillator (XO) block if crystal is connected, otherwise XIN input
- 011 = MCLKOSCP/N pins
- 100 = DPLL output (when DPLL master clock comes from APLL2; this decode only valid for APLL1)
- 110 = DPLL output (when DPLL master clock comes from MCLKOSCP/N pins)
- 111 = DPLL output (only use in APLL bypass, i.e. when [APLLCR1.APLLBYP](#)=1; test/debug mode only)

**Register Name:** AFBDIV1  
**Register Description:** APLL Feedback Divider Register 1  
**Register Address:** 22h

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>Name</b>	AFBDIV[3:0]				—	—	—	—
<b>Default</b>	0	0	0	0	0	0	0	0

The APLL registers are bank-selected by the [APLLSEL](#) register. See section [6.1.3](#).

**Bits 7 to 4: APLL Feedback Divider Register (AFBDIV[3:0]).** The full 75 bit AFBDIV[74:0] field spans the AFBDIV1 through AFBDIV10 registers. AFBDIV is an unsigned number with 9 integer bits (AFBDIV[74:66]) and up to 66 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 [5.7.2](#).

**Register Name:** AFBDIV2  
**Register Description:** APLL Feedback Divider Register 2  
**Register Address:** 23h

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>Name</b>	AFBDIV[11:4]							
<b>Default</b>	0	0	0	0	0	0	0	0

**Bits 7 to 0: APLL Feedback Divider Register (AFBDIV[11:4]).** See the [AFBDIV1](#) register description.

**Register Name:** AFBDIV3  
**Register Description:** APLL Feedback Divider Register 3  
**Register Address:** 24h

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>Name</b>	AFBDIV[19:12]							
<b>Default</b>	0	0	0	0	0	0	0	0

**Bits 7 to 0: APLL Feedback Divider Register (AFBDIV[19:12]).** See the [AFBDIV1](#) register description.

**Register Name:** AFBDIV4  
**Register Description:** APLL Feedback Divider Register 4  
**Register Address:** 25h

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>Name</b>	AFBDIV[27:20]							
<b>Default</b>	0	0	0	0	0	0	0	0

**Bits 7 to 0: APLL Feedback Divider Register (AFBDIV[27:20]).** See the [AFBDIV1](#) register description.

**Register Name:** AFBDIV5  
**Register Description:** APLL Feedback Divider Register 5  
**Register Address:** 26h

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>Name</b>	AFBDIV[35:28]							
<b>Default</b>	0	0	0	0	0	0	0	0

**Bits 7 to 0: APLL Feedback Divider Register (AFBDIV[35:28]).** See the [AFBDIV1](#) register description.

**Register Name:** AFBDIV6  
**Register Description:** APLL Feedback Divider Register 6  
**Register Address:** 27h

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Name	AFBDIV[43:36]							
Default	0	0	0	0	0	0	0	0

**Bits 7 to 0: APLL Feedback Divider Register (AFBDIV[43:36]).** See the [AFBDIV1](#) register description.

**Register Name:** AFBDIV7  
**Register Description:** APLL Feedback Divider Register 7  
**Register Address:** 28h

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Name	AFBDIV[51:44]							
Default	0	0	0	0	0	0	0	0

**Bits 7 to 0: APLL Feedback Divider Register (AFBDIV[51:44]).** See the [AFBDIV1](#) register description.

**Register Name:** AFBDIV8  
**Register Description:** APLL Feedback Divider Register 8  
**Register Address:** 29h

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Name	AFBDIV[59:52]							
Default	0	0	0	0	0	0	0	0

**Bits 7 to 0: APLL Feedback Divider Register (AFBDIV[59:52]).** See the [AFBDIV1](#) register description.

**Register Name:** AFBDIV9  
**Register Description:** APLL Feedback Divider Register 9  
**Register Address:** 2Ah

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Name	AFBDIV[67:60]							
Default	0	0	0	0	0	0	0	0

**Bits 7 to 0: APLL Feedback Divider Register (AFBDIV[67:60]).** See the [AFBDIV1](#) register description.

**Register Name:** AFBDIV10  
**Register Description:** APLL Feedback Divider Register 10  
**Register Address:** 2Bh

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Name	—	AFBDIV[74:68]						
Default	0	0	0	0	0	0	0	0

**Bits 7 to 0: APLL Feedback Divider Register (AFBDIV[74:68]).** See the [AFBDIV1](#) register description.

**Register Name:** AFBDEN1  
**Register Description:** APLL Feedback Divider Denominator Register 1  
**Register Address:** 2Ch

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>Name</b>	AFBDEN[7:0]							
<b>Default</b>	0	0	0	0	0	0	0	1

The APLL registers are bank-selected by the [APLLSEL](#) register. See section [6.1.3](#).

**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 [5.7.2](#).

**Register Name:** AFBDEN2  
**Register Description:** APLL Feedback Divider Denominator Register 2  
**Register Address:** 2Dh

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>Name</b>	AFBDEN[15:8]							
<b>Default</b>	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  
**Register Description:** APLL Feedback Divider Denominator Register 3  
**Register Address:** 2Eh

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>Name</b>	AFBDEN[23:16]							
<b>Default</b>	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  
**Register Description:** APLL Feedback Divider Denominator Register 4  
**Register Address:** 2Fh

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>Name</b>	AFBDEN[31:24]							
<b>Default</b>	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  
**Register Description:** APLL Feedback Divider Remainder Register 1  
**Register Address:** 30h

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>Name</b>	AFBREM[7:0]							
<b>Default</b>	0	0	0	0	0	0	0	0

The APLL registers are bank-selected by the [APLLSEL](#) register. See section [6.1.3](#).

**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. When AFBBP=0, [AFBREM](#) must be set to 0. See section [5.7.2](#).

**Register Name:** AFBREM2  
**Register Description:** APLL Feedback Divider Remainder Register 2  
**Register Address:** 31h

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>Name</b>	AFBREM[15:8]							
<b>Default</b>	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  
**Register Description:** APLL Feedback Divider Remainder Register 3  
**Register Address:** 32h

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>Name</b>	AFBREM[23:16]							
<b>Default</b>	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  
**Register Description:** APLL Feedback Divider Remainder Register 4  
**Register Address:** 33h

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>Name</b>	AFBREM[31:24]							
<b>Default</b>	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  
**Register Description:** APLL Feedback Divider Truncate Bit Position  
**Register Address:** 34h

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>Name</b>	AFBBP[7:0]							
<b>Default</b>	0	0	0	0	0	0	0	0

The APLL registers are bank-selected by the [APLLSEL](#) register. See section [6.1.3](#).

**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 66 fractional bits in AFBDIV. The value in this AFBBP field specifies 66 – number\_of\_valid\_AFBDIV\_fractional\_bits. When AFBBP=0 all 66 AFBDIV fractional bits are valid. When AFBBP=42, the most significant 24 AFBDIV fractional bits are valid and the least significant 42 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 66 are invalid. When AFBBP=0, [AFBREM](#) must be set to 0 and [AFBDEN](#) must be set to 1. See section [5.7.2](#).

### 6.3.4 Output Clock Registers

**Register Name:** OCSEL  
**Register Description:** Output Clock Select Register  
**Register Address:** 40h

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>Name</b>	0	0	0	0	OCSEL[3:0]			
<b>Default</b>	0	0	0	0	0	0	0	1

**Bits 3 to 0: Output Clock Select (OCSEL[2:0]).** This field is a bank-select control that specifies the output clock for which registers are mapped into the Output Clock Registers section of [Table 6-1](#). See section [6.1.3](#).

0000 = {unused value}  
 0001 = Output clock 1  
 0010 = Output clock 2  
 0011 = Output clock 3  
 0100 = Output clock 4 (MAX24610 only)  
 0101 = Output clock 5 (MAX24610 only)  
 0110 = Output clock 6 (MAX24610 only)  
 0111 = Output clock 7 (MAX24610 only)  
 1000 = Output clock 8  
 1001 = Output clock 9 (MAX24610 only)  
 1010 = Output clock 10  
 1011 to 1111 = {unused value}

**Register Name:** OCCR1  
**Register Description:** Output Clock Configuration Register 1  
**Register Address:** 41h

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>Name</b>	—	MSDIV[6:0]						
<b>Default</b>	0	0	0	0	0	0	0	0

The output clock registers are bank-selected by the [OCSEL](#) register. See section [6.1.3](#).

**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 MSDIV must be set to a value that causes the output clock of the medium-speed divider to be 312.5MHz or less. See section [5.8.2](#).

**Register Name:** OCCR2  
**Register Description:** Output Clock Configuration Register 2  
**Register Address:** 42h

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>Name</b>	—	—	DRIVE[1:0]		OCSF[3:0]			
<b>Default</b>	0	0	0	0	0	0	0	0

The output clock registers are bank-selected by the [OCSEL](#) register. See section [6.1.3](#).

**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.

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

**Bits 3 to 0: Output Clock Signal Format (OCSF[3:0]).** See section [5.8.1](#).

0000 = Disabled (high-impedance, low power mode)  
 0001 = CML, standard swing ( $V_{OD}=800mV_{P-P}$  typical)  
 0010 = CML, narrow swing ( $V_{OD}=400mV_{P-P}$  typical)  
 0011 = {unused value}  
 0100 = One CMOS, OCxPOS enabled, OCxNEG high impedance  
 0101 = Two CMOS, OCxNEG in phase with OCxPOS  
 0110 = Two CMOS, OCxNEG inverted vs. OCxPOS  
 0111 = HSTL (Set [OCCR2.DRIVE](#)=11 (4x) to meet JESD8-6)



**Register Name:** OCCR3  
**Register Description:** Output Clock Configuration Register 3  
**Register Address:** 43h

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>Name</b>	PHADJ[3:0]				—	POL	ASQUEL	DALEN
<b>Default</b>	0	0	0	0	0	0	0	0

The output clock registers are bank-selected by the [OCSEL](#) register. See section [6.1.3](#).

**Bits 7 to 4: Output Clock Phase Adjustment (PHADJ[3:0]).** This field can be used to adjust the phase of output OCxPOS/NEG vs. the phase of other clock outputs. The adjustment is in units of APLL output clock cycles. For example, if the APLL output frequency is 625MHz then one APLL output clock cycle is 1.6ns, the smallest phase adjustment is 0.8ns, and the adjustment range is  $\pm 5.6$ ns. See section [5.8.3](#).

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

**Bit 2: Polarity (POL).** This bit specifies the polarity of the output clock signal. When [OCCR2.OCSF](#) configures the output for one of the 2x CMOS modes, POL=1 inverts both CMOS outputs vs. the polarity they have when POL=0. See section [5.8.3](#).

- 0 = Normal
- 1 = Inverted

**Bit 1: Auto-Squelch Enable (ASQUEL).** This bit enables automatic squelching of the output clock whenever the DPLL has no selected reference ([PTAB1.SELREF](#) = 0). When a CMOS output is squelched it is forced low. When a differential output is squelched, its POS pin is forced low and its NEG pin is forced high..

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

**Bit 0: Divider Align Enable (DALEN).** This bit enables alignment of the output clock's medium-speed divider and output clock divider when the [APLLCR1.DALIGN](#) bit is set to 1. For best results, this signal should be set to 1 for at least 2ms then set back to 0.

- 0 = Do not align the output clock dividers
- 1 = Align the output clock dividers

**Register Name:** OCDIV1  
**Register Description:** Output Clock Divider Register 1  
**Register Address:** 44h

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>Name</b>	OCDIV[7:0]							
<b>Default</b>	0	0	0	0	0	0	0	0

The output clock registers are bank-selected by the [OCSEL](#) register. See section [6.1.3](#).

**Bits 7 to 0: Output Clock Divider (OCDIV[7:0]).** The full 24-bit OCDIV[23:0] field spans this register, [OCDIV2](#) and [OCDIV3](#). OCDIV is an unsigned integer. The frequency of the clock from the medium-speed divider is divided by OCDIV+1 to make the output clock signal. See section [5.8.2](#).

**Register Name:** OCDIV2  
**Register Description:** Output Clock Divider Register 2  
**Register Address:** 45h

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>Name</b>	OCDIV[15:8]							
<b>Default</b>	0	0	0	0	0	0	0	0

The output clock registers are bank-selected by the [OCSEL](#) register. See section [6.1.3](#).

**Bits 7 to 0: Output Clock Divider (OCDIV[15:8]).** See the [OCDIV1](#) register description.

**Register Name:** OCDIV3  
**Register Description:** Output Clock Divider Register 3  
**Register Address:** 46h

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>Name</b>	OCDIV[23:16]							
<b>Default</b>	0	0	0	0	0	0	0	0

The output clock registers are bank-selected by the [OCSEL](#) register. See section [6.1.3](#).

**Bits 7 to 0: Output Clock Divider (OCDIV[23:16]).** See the [OCDIV1](#) register description.

### 6.3.5 Input Clock Registers

Note: The input clock registers cannot be read or written unless a master clock is provided to the input block and the DPLL. See section 5.2.2.

Note: When the input block is disabled ([MCR1.ICBEN=0](#)) all input clock register fields, except [ICCR1.ICEN](#), are ignored by the device and should be ignored by system software.

**Register Name:** ICSEL  
**Register Description:** Input Clock Select Register  
**Register Address:** 50h

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>Name</b>	0	0	0	0	ICSEL[3:0]			
<b>Default</b>	0	0	0	0	0	0	0	1

**Bits 3 to 0: Input Clock Select (ICSEL[3:0]).** This field is the bank-select control that specifies the input clock for which registers are mapped into the Input Clock Registers section of [Table 6-1](#). See section 6.1.3.

- 0000 = {unused value}
- 0001 = IC1 input
- 0010 = IC2 input
- 0011 to 1111 = {unused values}

**Register Name:** ICCR1  
**Register Description:** Input Clock Configuration Register 1  
**Register Address:** 51h

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>Name</b>	ICEN	POL	IFREQR[1:0]		LKFREQ[3:0]			
<b>Default</b>	0	0	0	0	0	1	1	1

The input clock registers are bank-selected by the [ICSEL](#) register. See section [6.1.3](#).

**Bit 7: Input Clock Enable (ICEN).** This field enables and disables the input clock's differential receiver. The power consumption numbers for the differential receiver and the crystal oscillator are shown in [Table 8-2](#). See section [5.4](#).

0 = Disable (power down)

1 = Enable

**Bit 6: Locking Polarity (POL).** This field specifies which input clock signal edge the DPLL will lock to. See section [5.5.1](#).

0 = Falling edge

1 = Rising edge

**Bits 5 to 4: Input Frequency Range (IFREQR[1:0]).** This field specifies the approximate frequency of the input clock at the device pins. This field must be set correctly for proper operation of the fractional scaling block. See section [5.5.1](#).

00 = Input clock frequency < 100MHz

01 = 100MHz <= input clock frequency < 200MHz

10 = 200MHz <= input clock frequency < 400MHz

11 = Input clock frequency >= 400MHz

**Bits 3 to 0: DPLL Lock Frequency (LKFREQ[3:0]).** The input clock frequency is optionally scaled by the ratio  $(ICN+1) / (ICD+1)$  before being presented to the DPLL. This field specifies the frequency at which the DPLL locks to the scaled signal. See section [5.5.1](#).

0000 = 2kHz\*

0001 = 8kHz\*

0010 = 64kHz\*

0011 = 1.544MHz

0100 = 2.048MHz

0101 = 6.312MHz

0110 = 6.48MHz

0111 = 19.44MHz

1000 = 25.92MHz

1001 = 1MHz

1010 = 2.5MHz

1011 = 25MHz

1100 = 31.25MHz

1101 = 10.24MHz

1110 to 1111 = {unused values}

\* Note lock frequencies of 2kHz, 8kHz and 64kHz should not be used with fractional scaling (i.e. when [ICN](#)>0) because the the fractional scaling block may generate wander.

**Register Name:** ICCR2  
**Register Description:** Input Clock Configuration Register 2  
**Register Address:** 52h

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>Name</b>	—	GPIOSQ	FMONCLK[1:0]	—	SOFTEN	HARDEN	FREN	
<b>Default</b>	0	0	0	0	0	1	1	

The input clock registers are bank-selected by the [ICSEL](#) register. See section [6.1.3](#).

**Bit 6: GPIO Squelch (GPIOSQ).** When this bit is high, the input clock is squelched in the input clock block when the associated GPIO pin is high. IC1 is squelched when GPIO1 is high. IC2 is squelched when GPIO2 is high. This bit has no effect on the input clock signal going to the APLL muxes.

0 = Disable

1 = Enable

**Bits 5 to 4: Frequency Monitor Clock Source (FMONCLK[1:0]).** This field specifies the reference clock source for the input clock frequency monitor. See section [5.5.2.1](#).

00 = Internal master clock

01 = DPLL output

10, 11 = {unused values}

**Bit 2: Soft Frequency Alarm Enable (SOFTEN).** This bit enables input clock frequency monitoring with the soft alarm limits set in the [ICSLIM](#) register. Soft alarms are reported in the SOFT status bits of the [ISR](#) register. See section [5.5.2.1](#).

0 = Disabled

1 = Enabled

**Bit 1: Hard Frequency Limit Enable (HARDEN).** This bit enables input clock frequency monitoring with the hard alarm limits set in the [ICAHLIM](#) and [ICRHLIM](#) registers. Hard alarms are reported in the HARD status bits of the [ISR](#) register. See section [5.5.2.1](#).

0 = Disabled

1 = Enabled

**Bit 0: Frequency Range Detect Enable (FREN).** When this bit is set to 1 the frequency of each input clock is measured and used to quickly declare the input inactive. See section [5.5.2.1](#).

0 = Frequency Range Detect disabled

1 = Frequency Range Detect enabled

**Register Name:** ICCR3  
**Register Description:** Input Clock Configuration Register 3  
**Register Address:** 53h

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>Name</b>	0	0	0	NSEN	FMONLEN[3:0]			
<b>Default</b>	0	0	0	0	0	0	1	0

The input clock registers are bank-selected by the [ICSEL](#) register. See section [6.1.3](#).

**Bits 4: Noise Shaping Enable (NSEN).** Setting this bit to one enables noise shaping circuitry in the input clock fractional scaling block. The effect of this noise shaping is to move the phase noise generated by the fractional scaling digital circuitry up to higher frequencies where it can be attenuated more by a downstream PLL. This feature is most beneficial when an APLL is locked directly to one of the input clock signals ([APLLCR2.APLLMUX=0xx](#)).

**Bits 3 to 0: Frequency Monitor Measurement Length (FMONLEN[3:0]).** This field specifies the length of time the input frequency monitor takes to measure the frequency of the input clock. The frequency measurement length specified by FMONLEN is a function of the measurement reference clock specified by [ICCR2.FMONCLK](#) as shown below. See section [5.5.2.1](#).

[ICCR4.FMRES=0 \(Standard Resolution\)](#)

[ICCR2.FMONCLK\[1:0\] = 00:](#)

0000 = 31ms  
 0001 = 62ms  
 0010 = 124ms  
 0011 = 250ms  
 0100 = 500ms  
 0101 = 1sec  
 0110 = 2sec  
 0111 = 4sec  
 1000 = 8sec  
 1001-1111 = {unused values}

[ICCR2.FMONCLK\[1:0\] = 01:](#)

0000 = 82ms  
 0001 = 164ms  
 0010 = 328ms  
 0011 = 656ms  
 0100 = 1.31sec  
 0101 = 2.62sec  
 0110 = 5.24sec  
 0111 = 10.5sec  
 1000 = 21sec  
 1001-1111 = {unused values}

[ICCR4.FMRES=1 \(High Resolution\)](#)

[ICCR2.FMONCLK\[1:0\] = 00:](#)

0000 = 3.982 sec  
 0001 = 7.962 sec  
 0010 = 15.926 sec  
 0011 = 31.850 sec  
 0100 = 62.700 sec  
 0101 = 127.402 sec  
 0110 = 254.804 sec  
 0111 = 509.608 sec  
 1000 = 1019.216 sec  
 1001 = 2038.432 sec  
 1010-1111 = {unused values}

[ICCR2.FMONCLK\[1:0\] = 01:](#)

0000 = 2.622 sec  
 0001 = 5.242 sec  
 0010 = 10.486 sec  
 0011 = 20.972 sec  
 0100 = 41.944 sec  
 0101 = 83.006 sec  
 0110 = 167.772 sec  
 0111 = 335.544 sec  
 1000 = 671.088 sec  
 1001 = 1342.178 sec  
 1010-1111 = {unused values}

**Register Name:** ICN1  
**Register Description:** Input Clock Fractional Scaling Numerator Register 1  
**Register Address:** 54h

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>Name</b>	ICN[7:0]							
<b>Default</b>	0	0	0	0	0	0	0	0

The input clock registers are bank-selected by the [ICSEL](#) register. See section [6.1.3](#).

The ICN1 and ICN2 registers must be read consecutively and written consecutively. See section [6.1.4](#).

**Bits 7 to 0: Input Clock Fractional Scaling Numerator (ICN[7:0]).** The full 16-bit ICN[15:0] field spans this register and [ICN2](#). ICN is an unsigned integer. The value ICN+1 is the numerator used for fractional scaling of the input clock frequency. See section [5.5.1](#).

**Register Name:** ICN2  
**Register Description:** Input Clock Fractional Scaling Numerator Register 2  
**Register Address:** 55h

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>Name</b>	ICN[15:8]							
<b>Default</b>	0	0	0	0	0	0	0	0

The input clock registers are bank-selected by the [ICSEL](#) register. See section [6.1.3](#).

The ICN1 and ICN2 registers must be read consecutively and written consecutively. See section [6.1.4](#).

**Bits 7 to 0: Input Clock Fractional Scaling Numerator (ICN[15:8]).** See the [ICN1](#) register description.



**Register Name:** ICD1  
**Register Description:** Input Clock Fractional Scaling Denominator Register 1  
**Register Address:** 56h

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>Name</b>	ICD[7:0]							
<b>Default</b>	0	0	0	0	0	0	0	0

The input clock registers are bank-selected by the [ICSEL](#) register. See section [6.1.3](#).

The ICD1 through ICD4 registers must be read consecutively and written consecutively. See section [6.1.4](#).

**Bits 7 to 0: Input Clock Fractional Scaling Denominator (ICD[7:0]).** The full 32-bit ICD[31:0] field spans this register, [ICD2](#), [ICD3](#) and [ICD4](#). ICD is an unsigned integer. The value ICD+1 is the denominator used for fractional scaling of the input clock frequency. See section [5.5.1](#).

**Register Name:** ICD2  
**Register Description:** Input Clock Fractional Scaling Denominator Register 2  
**Register Address:** 57h

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>Name</b>	ICD[15:8]							
<b>Default</b>	0	0	0	0	0	0	0	0

The input clock registers are bank-selected by the [ICSEL](#) register. See section [6.1.3](#).

The ICD1 through ICD4 registers must be read consecutively and written consecutively. See section [6.1.4](#).

**Bits 7 to 0: Input Clock Fractional Scaling Denominator (ICD[15:8]).** See the [ICD1](#) register description.

**Register Name:** ICD3  
**Register Description:** Input Clock Fractional Scaling Denominator Register 3  
**Register Address:** 58h

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>Name</b>	ICD[23:16]							
<b>Default</b>	0	0	0	0	0	0	0	0

The input clock registers are bank-selected by the [ICSEL](#) register. See section [6.1.3](#).

The ICD1 through ICD4 registers must be read consecutively and written consecutively. See section [6.1.4](#).

**Bits 7 to 0: Input Clock Fractional Scaling Denominator (ICD[23:16]).** See the [ICD1](#) register description.

**Register Name:** ICD4  
**Register Description:** Input Clock Fractional Scaling Denominator Register 4  
**Register Address:** 59h

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>Name</b>	ICD[31:24]							
<b>Default</b>	0	0	0	0	0	0	0	0

The input clock registers are bank-selected by the [ICSEL](#) register. See section [6.1.3](#).

The ICD1 through ICD4 registers must be read consecutively and written consecutively. See section [6.1.4](#).

**Bits 7 to 0: Input Clock Fractional Scaling Denominator (ICD[31:24]).** See the [ICD1](#) register description.

**Register Name:** ICLBU  
**Register Description:** Input Clock Leaky Bucket Upper Threshold  
**Register Address:** 5Ah

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>Name</b>	ICLBU[7:0]							
<b>Default</b>	0	0	0	0	0	1	1	0

The input clock registers are bank-selected by the [ICSEL](#) register. See section [6.1.3](#).

**Bits 7 to 0: Input Clock Leaky Bucket Upper Threshold (ICLBU[7:0]).** When the leaky bucket accumulator is equal to the value stored in this field, the activity monitor declares an activity alarm by setting the input clock's ACT bit in the [ISR](#) register. See section [5.5.2.2](#).

**Register Name:** ICLBL  
**Register Description:** Input Clock Leaky Bucket Lower Threshold  
**Register Address:** 5Bh

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>Name</b>	ICLBL[7:0]							
<b>Default</b>	0	0	0	0	0	1	0	0

The input clock registers are bank-selected by the [ICSEL](#) register. See section [6.1.3](#).

**Bits 7 to 0: Input Clock Leaky Bucket Lower Threshold (ICLBL[7:0]).** When the leaky bucket accumulator is equal to the value stored in this field, the activity monitoring logic clears the activity alarm (if previously declared) by clearing the input clock's ACT bit in the [ISR](#) register. See section [5.5.2.2](#).

**Register Name:** ICLBS  
**Register Description:** Input Clock Leaky Bucket Size  
**Register Address:** 5Ch

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>Name</b>	ICLBS[7:0]							
<b>Default</b>	0	0	0	0	1	0	0	0

The input clock registers are bank-selected by the [ICSEL](#) register. See section [6.1.3](#).

**Bits 7 to 0: Input Clock Leaky Bucket Size (ICLBS[7:0]).** This field specifies the maximum value of the leaky bucket accumulator. The accumulator cannot increment past this value. Setting this register to 00h disables activity monitoring and forces the ACT bit to 1 in the [ISR](#) register. See section [5.5.2.2](#).

**Register Name:** ICLBD  
**Register Description:** Input Clock Leaky Bucket Decay Rate  
**Register Address:** 5Dh

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>Name</b>	—	—	—	—	—	—	ICLBD[1:0]	
<b>Default</b>	0	0	0	0	0	0	0	1

The input clock registers are bank-selected by the [ICSEL](#) register. See section [6.1.3](#).

**Bits 1 to 0: Input Clock Leaky Bucket Decay Rate (ICLBD[1:0]).** This field specifies the decay or “leak” rate of the leaky bucket accumulator. For each period of 1, 2, 4, or 8 128ms intervals in which no irregularities are detected on the input clock, the accumulator decrements by 1. See section [5.5.2.2](#).

- 00 = decrement every 128ms (8 units/second)
- 01 = decrement every 256ms (4 units/second)
- 10 = decrement every 512ms (2 units/second)
- 11 = decrement every 1024ms (1 unit/second)

**Register Name:** ICAHLIM  
**Register Description:** Input Clock Frequency Acceptance Hard Limit  
**Register Address:** 5Eh

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>Name</b>	ICAHLIM[7:0]							
<b>Default</b>	0	0	0	0	1	0	0	0

The input clock registers are bank-selected by the [ICSEL](#) register. See section [6.1.3](#).

**Bits 7 to 0: Input Clock Frequency Acceptance Hard Limit (ICAHLIM[7:0]).** This field is an unsigned integer that specifies the hard frequency limit for accepting an input clock (i.e. the pull-in range for the input clock). When the fractional frequency offset of the input clock is less than this limit, the frequency monitor indicates the input clock has valid frequency by setting HARD = 0 in the [ISR](#) register.

When [ICCR4.FMRES](#)=0 (standard resolution), ICAHLIM can be set as high as  $\pm 320$ ppm and has  $\sim 1.25$ ppm resolution. The default limit is approximately  $\pm 10.05$ ppm. The limit in ppm is  $\text{ICAHLIM} \times 1.255867$ ppm.

When [ICCR4.FMRES](#)=1 (high resolution), ICAHLIM can be set as high as  $\pm 50$ ppm and has  $\sim 0.2$ ppm resolution. The limit in ppm is  $\pm \text{ICAHLIM} \times 0.19622928$ ppm.

The reference clock used to measure the frequency of the input clock is specified by [ICCR2.FMONCLK](#). The hard alarm is enabled for an input by setting [ICCR2.HARDEN](#) = 1. Set  $\text{ICRHLIM} \geq \text{ICAHLIM} \times 1.05$  to meet the hysteresis and rejection requirements of GR-1244 R3-30 [110] and R3-31 [111]. The value 00h is undefined. See section [5.5.2.1](#).

**Register Name:** ICRHLIM  
**Register Description:** Input Clock Frequency Rejection Hard Limit  
**Register Address:** 5Fh

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>Name</b>	ICRHLIM[7:0]							
<b>Default</b>	0	0	0	0	1	0	0	1

The input clock registers are bank-selected by the [ICSEL](#) register. See section [6.1.3](#).

**Bits 7 to 0: Input Clock Frequency Rejection Hard Limit (ICRHLIM[7:0]).** This field is an unsigned integer that specifies the hard frequency limit for rejecting an input clock. When the fractional frequency offset of the input clock is greater than or equal to this limit, the frequency monitor indicates hard frequency alarm by setting HARD = 1 in the [ISR](#) register, which immediately invalidates the clock.

When [ICCR4.FMRES](#)=0 (standard resolution), ICRHLIM can be set as high as  $\pm 320$ ppm and has  $\sim 1.25$ ppm resolution. The default limit is approximately  $\pm 11.3$ ppm. The limit in ppm is  $\pm \text{ICRHLIM} \times 1.255867$ ppm.

When [ICCR4.FMRES](#)=1 (high resolution), ICRHLIM can be set as high as  $\pm 50$ ppm and has  $\sim 0.2$ ppm resolution. The limit in ppm is  $\pm \text{ICRHLIM} \times 0.19622928$ ppm.

The reference clock used to measure the frequency of the input clock is specified by [ICCR2.FMONCLK](#). The hard alarm is enabled for an input by setting [ICCR2.HARDEN](#) = 1. Set  $\text{ICRHLIM} \geq \text{ICAHLIM} \times 1.05$  to meet the hysteresis and rejection requirements of GR-1244 R3-30 [110] and R3-31 [111]. The value 00h is undefined. See section [5.5.2.1](#).

**Register Name:** ICSLIM  
**Register Description:** Input Clock Frequency Soft Limit  
**Register Address:** 60h

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>Name</b>	ICSLIM[7:0]							
<b>Default</b>	0	0	0	0	0	1	1	0

The input clock registers are bank-selected by the [ICSEL](#) register. See section [6.1.3](#).

**Bits 7 to 0: Input Clock Frequency Soft Limit (ICSLIM[7:0]).** This field is an unsigned integer that specifies the soft frequency limit for an input clock. When the fractional frequency offset of the input clock is greater than or equal to this soft limit, the frequency monitor indicates soft frequency alarm by setting SOFT=1 in the appropriate [ISR](#) register. The soft alarm limit is only used for monitoring; soft alarms do not invalidate input clocks.

When [ICCR4.FMRES](#)=0 (standard resolution), ICSLIM can be set as high as  $\pm 320\text{ppm}$  and has  $\sim 1.25\text{ppm}$  resolution. The default limit is approximately  $\pm 7.5\text{ppm}$ . The limit in ppm is  $\pm \text{ICSLIM} \times 1.255867\text{ppm}$ .

When [ICCR4.FMRES](#)=1 (high resolution), ICSLIM can be set as high as  $\pm 50\text{ppm}$  and has  $\sim 0.2\text{ppm}$  resolution. The limit in ppm is  $\pm \text{ICSLIM} \times 0.19622928\text{ppm}$ .

The reference clock used to measure the frequency of the input clock is specified by [ICCR2.FMONCLK](#). The soft alarm is enabled for an input by setting [ICCR2.SOFTEN](#)=1. The value 00h is undefined. See section [5.5.2.1](#).

**Register Name:** FMEAS1  
**Register Description:** Input Clock Frequency Measurement Register 1  
**Register Address:** 61h

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>Name</b>	FMEAS[7:0]							
<b>Default</b>	0	0	0	0	0	0	0	0

The input clock registers are bank-selected by the [ICSEL](#) register. See section [6.1.3](#).

The FMEAS1 and FMEAS2 registers must be read consecutively. See section [6.1.4](#).

**Bits 7 to 0: Measured Frequency (FMEAS[7:0]).** The full 16-bit FMEAS[15:0] field spans this register and [FMEAS2](#). This read-only field indicates the measured frequency of the input clock. FMEAS is a two's-complement signed integer that expresses the fractional frequency offset of the input clock. When [ICCR4.FMRES](#)=0 (standard resolution) the measured frequency is FMEAS[15:0] x 0.156983ppm. When [ICCR4.FMRES](#)=1 (high resolution) the measured frequency is FMEAS[15:0] x 4.905732ppb. See section [5.5.2.1](#).

Note that if the DPLL's nominal master clock frequency ( $f_{MCLK}$ ) is not an integer multiple of 500Hz, the frequency reported by FMEAS will have a small offset error that can be calculated using the following equations:

$$N = \text{round}(f_{MCLK} / 500)$$

$$\text{offset error in ppm} = [ (500 * N) - f_{MCLK} ] / f_{MCLK} * 1,000,000$$

The worst possible offset error is 1.32ppm which occurs when  $f_{MCLK}$  ends in 250 and therefore  $f_{MCLK} / 500$  has a fractional part of exactly 0.5, for example  $f_{MCLK} = 190,000,250\text{Hz}$  and  $f_{MCLK} / 500 = 380,000.5$ . If the DPLL's master clock frequency is an integer multiple of 500Hz then the offset error is zero.

**Register Name:** FMEAS2  
**Register Description:** Input Clock Frequency Measurement Register 2  
**Register Address:** 62h

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>Name</b>	FMEAS[15:8]							
<b>Default</b>	0	0	0	0	0	0	0	0

The input clock registers are bank-selected by the [ICSEL](#) register. See section [6.1.3](#).

The FMEAS1 and FMEAS2 registers must be read consecutively. See section [6.1.4](#).

**Bits 7 to 0: Measured Frequency (FMEAS[15:8]).** See the [FMEAS1](#) register description.

**Register Name:** ICCR4  
**Register Description:** Input Clock Configuration Register 4  
**Register Address:** 63h

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>Name</b>	—	—	—	—	—	FMRES	—	—
<b>Default</b>	0	0	0	0	0	0	0	0

**Bit 2: Frequency Monitor Resolution (FMRES).** This bit specifies standard resolution or high resolution for the frequency monitor. See section [5.5.2.1](#).

0 = Standard resolution

1 = High resolution

Register	Standard Resolution	High Resolution
<a href="#">FMEAS</a>	0.156983ppm	4.905732ppb
<a href="#">ICAH LIM</a>	1.255867ppm	0.19622928ppm
<a href="#">ICRHLIM</a>		
<a href="#">ICSLIM</a>		

### 6.3.6 DPLL Registers

Note: The DPLL registers cannot be read or written unless a master clock is provided to the input block and the DPLL. See section 5.2.2.

Note: When the DPLL is disabled ([MCR1.DPLLEN=0](#)) all DPLL register fields are ignored by the device and should be ignored by system software.

**Register Name:** DPLLCR1  
**Register Description:** DPLL Configuration Register 1  
**Register Address:** 71h

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>Name</b>	EXTSW	UFSW	REVERT	PPM160	FORCE[3:0]			
<b>Default</b>	see below	0	0	0	0	0	0	0

**Bit 7: External Reference Switching Mode (EXTSW).** This bit enables the input block's external reference switching mode. In this mode, if the SS pin is high the DPLL is forced to lock to input IC1 whether or not the selected input has a valid reference signal. If the SS pin is low the DPLL is forced to lock to input IC2 whether or not the selected input has a valid reference signal. See section 5.5.3.5.

0 = Normal operation

1 = External switching mode

**Bit 6: Ultra-Fast Switching Mode (UFSW).** See section 5.5.3.4.

0 = Disabled

1 = Enabled. The current selected reference is disqualified after a few missing clock cycles (see [Table 5-4](#)).

**Bit 5: Revertive Mode (REVERT).** This bit configures the DPLL for revertive or nonrevertive operation. In revertive mode, if an input clock with a higher priority than the selected reference becomes valid, the higher priority reference immediately becomes the selected reference. In nonrevertive mode the higher priority reference does not immediately become the selected reference but does become the highest-priority reference in the priority table (REF1 field in the [PTAB1](#) register). See section 5.5.3.2.

**Bit 4: 160ppm Mode (PPM160).** This bit enables the DPLL's  $\pm 160$ ppm tracking range mode. See section 5.6.11.

0 = Disabled

1 = Enabled

**Bits 3 to 0: Force Selected Reference (FORCE[3:0]).** This field provides a way to force a specified input clock to be the selected reference for the DPLL. Internally this is accomplished by forcing the clock to have the highest priority (as specified in [PTAB1.REF1](#)). In revertive mode (REVERT=1) the forced clock automatically becomes the selected reference (as specified in [PTAB1.SELREF](#)) as well. In nonrevertive mode (REVERT=0) the forced clock only becomes the selected reference when the existing selected reference is invalidated or made unavailable for selection.

When a reference is forced, the frequency monitor and activity monitor for that input and the DPLL's loss-of-lock timeout logic all continue to operate and affect the relevant [ISLR](#), [VALSR](#) and [ICLSR](#) register bits. However, when the reference is declared invalid the DPLL is not allowed to switch to another input clock. The DPLL continues to respond to the fast activity monitor, transitioning to a temporary digital hold state in response to short-term events and to digital hold in response to longer events. This field has no effect when EXTSW=1. See section 5.5.3.3.

0000 = Automatic source selection (normal operation)

0001 = Force to IC1

0010 = Force to IC2

0011 to 1111 = {unused values}



**Register Name:** DPLLCR2  
**Register Description:** DPLL Configuration Register 2  
**Register Address:** 72h

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>Name</b>	—	1	—	—	—	STATE[2:0]		
<b>Default</b>	0	0	0	0	0	0	0	0

**Bit 6:** Set this bit to 1 for proper operation.

**Bits 2 to 0: DPLL State Control (STATE[2:0]).** This field can be used to force the DPLL state machine to a specified state. The state machine remains in the forced state, and therefore cannot react to alarms and other events, as long as STATE is not equal to 000. See section [5.6.1](#).

000 = Automatic (normal state machine operation)

001 = Free-run

010 = Digital hold

011 = {unused value}

100 = Locked

101 = Prelocked 2

110 = Prelocked

111 = Loss-of-lock

**Register Name:** DPLLCR3  
**Register Description:** DPLL Configuration Register 3  
**Register Address:** 73h

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>Name</b>	ADAMP[2:0]			ABW[4:0]				
<b>Default</b>	0	1	1	0	1	1	1	1

**Bits 7 to 5: Acquisition Damping Factor (ADAMP[2:0]).** This field configures the DPLL's damping factor when acquiring lock (i.e. pulling in). Acquisition damping factor is a function of both ADAMP and the acquisition DPLL bandwidth (ABW field below). The default value corresponds to a damping factor of 5 for all bandwidths. See section 5.6.3.

	≤ 4Hz	8Hz	18Hz	35Hz	≥ 70Hz
001 =	5	2.5	1.2	1.2	1.2
010 =	5	5	2.5	2.5	2.5
011 =	5	5	5	5	5
100 =	5	5	5	10	10
101 =	5	5	5	10	20
000, 110, and 111 =	{unused values}				

The gain peak for each damping factor is shown below:

DAMPING FACTOR	GAIN PEAK (dB)
1.2	0.4
2.5	0.2
5.0	0.1
10	0.06
20	0.03

**Bits 4 to 0: Acquisition Bandwidth (ABW[4:0]).** This field configures the bandwidth of the DPLL when acquiring lock (i.e. pulling in). When DPLLCR6.AUTOBW=0, DPLLCR4.LBW bandwidth is used for acquisition and for locked operation. When AUTOBW=1, ABW bandwidth is used for acquisition while LBW bandwidth is used for locked operation. See section 5.6.2.

01101 = 4 Hz  
 01110 = 8 Hz  
 01111 = 18 Hz (default)  
 10000 = 35 Hz  
 10001 = 70 Hz  
 10010 = 120Hz  
 10011 = 250Hz  
 10100 = 400Hz  
 10101 to 11111 = {unused values}

**Register Name:** DPLLCR4  
**Register Description:** DPLL Configuration Register 4  
**Register Address:** 74h

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>Name</b>	LDAMP[2:0]			LBW[4:0]				
<b>Default</b>	0	1	1	0	1	1	0	1

**Bits 7 to 5: Locked Damping Factor (LDAMP[2:0]).** This field configures the DPLL's damping factor when locked to an input clock. Locked damping factor is a function of both LDAMP and the locked DPLL bandwidth (LBW field below). The default value corresponds to a damping factor of 5 for all bandwidths. See section 5.6.3.

	≤ 4Hz	8Hz	18Hz	35Hz	≥ 70Hz
001 =	5	2.5	1.2	1.2	1.2
010 =	5	5	2.5	2.5	2.5
011 =	5	5	5	5	5
100 =	5	5	5	10	10
101 =	5	5	5	10	20
000, 110, and 111 =	{unused values}				

The gain peak for each damping factor is shown below:

DAMPING FACTOR	GAIN PEAK (dB)
1.2	0.4
2.5	0.2
5.0	0.1
10	0.06
20	0.03

**Bits 4 to 0: Locked Bandwidth (LBW[4:0]).** This field configures the bandwidth of the DPLL when locked to an input clock. When DPLLCR6.AUTOBW=0, the LBW bandwidth is used for acquisition and for locked operation. When AUTOBW=1, DPLLCR3.ABW bandwidth is used for acquisition while LBW bandwidth is used for locked operation. See section 5.6.2.

01101 = 4 Hz (default)  
 01110 = 8 Hz  
 01111 = 18 Hz  
 10000 = 35 Hz  
 10001 = 70 Hz  
 10010 = 120Hz  
 10011 = 250Hz  
 10100 = 400Hz  
 10101 to 11111 = {unused values}

**Register Name:** DPLLCR5  
**Register Description:** DPLL Configuration Register 5  
**Register Address:** 75h

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>Name</b>	NALOL	FLLOL	FLEN	CLEN	MCPDEN	USEMCPD	D180	PFD180
<b>Default</b>	0	1	1	1	0	0	0	0

**Bit 7: No-Activity Loss of Lock (NALOL).** The DPLL can detect that an input clock has no activity very quickly (within two clock cycles). When NALOL = 1, the DPLL internally declares loss-of-lock as soon as no activity is detected, and then switches to phase/frequency locking ( $\pm 360^\circ$ ). When NALOL = 0, loss-of-lock is not declared when clock cycles are missing, and nearest edge locking ( $\pm 180^\circ$ ) is used when the clock recovers. This gives tolerance to missing cycles. See sections 5.5.2.3 and 5.6.5.

- 0 = No activity does not trigger loss-of-lock
- 1 = No activity does trigger loss-of-lock

**Bit 6: Frequency Limit Loss of Lock (FLLOL).** When this bit is set to 1, the DPLL internally declares loss-of-lock when the DPLL's frequency exceeds the frequency hard limit specified in the [HRDLIM](#) registers. See section 5.6.5.

- 0 = DPLL does not declare loss-of-lock when the hard frequency limit is reached
- 1 = DPLL declares loss-of-lock when the hard frequency limit is reached

**Bit 5: Fine Phase Limit Enable (FLEN).** When this bit is set to 1, the DPLL internally declares loss-of-lock when the DPLL's phase (difference between output phase and input phase) exceeds the fine phase limit specified in the [PHLIM.FINELIM\[2:0\]](#) field. The fine limit must be disabled for multi-UI jitter tolerance. See section 5.6.5.

- 0 = Disabled
- 1 = Enabled

**Bit 4: Coarse Phase Limit Enable (CLEN).** When this bit is set to 1, the DPLL internally declares loss-of-lock when the DPLL's phase (difference between output phase and input phase) exceeds the coarse phase limit specified in the [PHLIM.COARSELIM\[3:0\]](#) field. See section 5.6.5.

- 0 = Disabled
- 1 = Enabled

**Bit 3: Multicycle Phase Detector Enable (MCPDEN).** This configuration bit enables the multicycle phase detector and allows the DPLL to tolerate large-amplitude jitter and wander. The range of the multicycle phase detector is the same as the coarse phase limit specified in the [PHLIM.COARSELIM\[3:0\]](#) field. See section 5.6.4.

- 0 = Disabled
- 1 = Enabled

**Bit 2: Use Multicycle Phase Detector in the DPLL Algorithm (USEMCPD).** This configuration bit enables the DPLL algorithm to use the multicycle phase detector so that a large phase measurement drives faster DPLL pull-in. When USEMCPD = 0, phase measurement is limited to  $\pm 360^\circ$ , giving slower pull-in at higher frequencies but with less overshoot. When USEMCPD = 1, phase measurement is set as specified in the [COARSELIM\[3:0\]](#) field, giving faster pull-in. MCPDEN should be set to 1 when USEMCPD = 1. See section 5.6.4.

- 0 = Disabled
- 1 = Enabled

**Bit 1: Disable 180 (D180).** When locking to a new reference, the DPLL first tries nearest-edge locking ( $\pm 180^\circ$ ) for the first two seconds. If unsuccessful it then tries full phase/frequency locking ( $\pm 360^\circ$ ). Disabling the nearest-edge locking can reduce lock time by up to two seconds but may cause an unnecessary phase shift (up to  $360^\circ$ ) when the new reference is close in frequency/phase to the old reference. See section 5.6.4.

- 0 = normal operation: try nearest-edge locking then phase/frequency locking
- 1 = phase/frequency locking only

**Bit 0: 180° PFD Enable (PFD180).** If D180 = 1, then PFD180 has no effect.

- 0 = Use 180° phase detector (nearest-edge locking mode)
- 1 = Use 180° phase-frequency detector

**Register Name:** DPLLCR6  
**Register Description:** DPLL Configuration Register 6  
**Register Address:** 76h

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>Name</b>	AUTOBW	LIMINT	0	—	—	—	RDAVG[1:0]	
<b>Default</b>	1	1	1	0	0	0	0	0

**Bit 7: Automatic Bandwidth Selection (AUTOBW).** See section 5.6.2.

0 = Use bandwidth specified in [DPLLCR4.LBW](#) during acquisition and while locked

1 = Use bandwidth specified in [DPLLCR3.ABW](#) during acquisition and use bandwidth specified in [DPLLCR4.LBW](#) while locked

**Bit 6: Limit Integral Path (LIMINT).** When this bit is set to 1, the DPLL's integral path is limited (i.e., frozen) when the DPLL reaches minimum or maximum frequency, as set in the [HRDLIM](#) registers. When the integral path is frozen, the current DPLL frequency in the [FREQ](#) registers is also frozen. Setting LIMINT = 1 minimizes overshoot when the DPLL is pulling in. See section 5.6.2.

0 = Do not freeze integral path at min/max frequency

1 = Freeze integral path at min/max frequency

**Bit 5: This bit defaults to 1 but must be set to 0 for proper operation.**

**Bits 1 to 0: Read Average (RDAVG[1:0]).** This field controls which value is accessed when reading the [FREQ](#) field: the DPLL's instantaneous frequency or average frequency.

00 = Read the instantaneous value

01 = Read the 1-second average

10 = {unused value}

11 = {unused value}

**Register Name:** PHLIM  
**Register Description:** DPLL Phase Limit Register  
**Register Address:** 79h

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>Name</b>	—	FINELIM[2:0]			COARSELIM[3:0]			
<b>Default</b>	0	0	1	0	0	1	0	1

**Bits 6 to 4: Fine Phase Limit (FINELIM[2:0]).** This field specifies the fine phase limit window, outside of which loss-of-lock is declared. The [DPLLCR5.FLEN](#) bit enables this feature. The phase of the input clock has to be inside the fine limit window for two seconds before phase lock is declared. Loss-of-lock is declared immediately if the phase of the input clock is outside the phase limit window. The default value of 010 is appropriate for most situations. See section [5.6.5](#).

- 000 = Always indicates loss of phase lock—do not use
- 001 = Small phase limit window,  $\pm 45$  to  $\pm 90^\circ$
- 010 = Normal phase limit window,  $\pm 90$  to  $\pm 180^\circ$  (default)
- 100, 101, 110, 111 = Proportionately larger phase limit window

**Bits 3 to 0: Coarse Phase Limit (COARSELIM[3:0]).** This field specifies the coarse phase limit and the tracking range of the multicycle phase detector. The [DPLLCR5.CLLEN](#) bit enables this feature. If jitter tolerance greater than 0.5UI is required and the input clock is a high frequency ( $\geq 10\text{MHz}$ ) signal then the DPLL can be configured to track phase errors over many UI using the multicycle phase detector. See section [5.6.4](#) and [5.6.5](#).

- 0000 =  $\pm 1\text{UI}$
- 0001 =  $\pm 3\text{UI}$
- 0010 =  $\pm 7\text{UI}$
- 0011 =  $\pm 15\text{UI}$
- 0100 =  $\pm 31\text{UI}$
- 0101 =  $\pm 63\text{UI}$
- 0110 =  $\pm 127\text{UI}$
- 0111 =  $\pm 255\text{UI}$
- 1000 =  $\pm 511\text{UI}$
- 1001 =  $\pm 1023\text{UI}$
- 1010 =  $\pm 2047\text{UI}$
- 1011 =  $\pm 4095\text{UI}$
- 1100 to 1111 =  $\pm 8191\text{UI}$

**Register Name:** PHLKTO  
**Register Description:** DPLL Phase Lock Timeout Register  
**Register Address:** 7Ah

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>Name</b>	PHLKTO[1:0]		PHLKTO[5:0]					
<b>Default</b>	0	0	1	1	0	0	1	0

**Bits 7 to 6: Phase Lock Timeout Multiplier (PHLKTO[1:0]).** This field is an unsigned integer that specifies the resolution of the PHLKTO field below.

- 00 = 2 seconds
- 01 = 4 seconds
- 10 = 8 seconds
- 11 = 16 seconds

**Bits 5 to 0: Phase Lock Timeout (PHLKTO[5:0]).** This field is an unsigned integer that, together with the PHLKTOM field above, specifies the length of time that the DPLL attempts to lock to an input clock before declaring a phase lock alarm (by setting the corresponding LOCK bit in the [ISR](#) register). The timeout period in seconds is  $\text{PHLKTO}[5:0] \times 2^{(\text{PHLKTO}[1:0]+1)}$ . When unable to declare lock, the DPLL remains in the prelocked, prelocked 2, or loss-of-lock states for the specified time before declaring a phase lock alarm on the selected input. When PHLKTO=0, the timeout is disabled, and the DPLL can remain indefinitely in the prelocked, prelocked 2 or loss-of-lock states. See section [5.6.1.4](#).

**Register Name:** LKATO  
**Register Description:** DPLL Lock Alarm Timeout Register  
**Register Address:** 7Bh

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>Name</b>	LKATOM[1:0]		LKATO[5:0]					
<b>Default</b>	0	0	1	1	0	0	1	0

**Bits 7 to 6: Lock Alarm Timeout Multiplier (LKATOM[1:0]).** This field is an unsigned integer that specifies the resolution of the LKATO field below.

- 00 = 2 seconds
- 01 = 4 seconds
- 10 = 8 seconds
- 11 = 16 seconds

**Bits 5 to 0: Lock Alarm Timeout (LKATO[5:0]).** This field is an unsigned integer that, together with the LKATOM field above, specifies the length of time that a phase lock alarm remains active before being automatically deasserted (by clearing the corresponding LOCK bit in the [ISR](#) register). The timeout period in seconds is  $\text{LKATO}[5:0] \times 2^{(\text{LKATOM}[1:0]+1)}$ . When LKATO=0, the timeout is disabled, and the phase lock alarm remains active until cleared by software writing a 0 to the LOCK bit. See section [5.6.1.4](#).



**Register Name:** HRDLIM1  
**Register Description:** DPLL Hard Frequency Limit Register 1  
**Register Address:** 7Ch

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>Name</b>	HRDLIM[7:0]							
<b>Default</b>	0	0	1	1	0	0	1	0

The HRDLIM1 and HRDLIM2 registers must be read consecutively and written consecutively. See section 6.1.4.

**Bits 7 to 0: DPLL Hard Frequency Limit (HRDLIM[7:0]).** The full 16-bit HRDLIM[15:0] field spans this register and HRDLIM2. HRDLIM is an unsigned integer that specifies the hard frequency limit or pull-in/hold-in range of the DPLL. This is a limit of the DPLL's integral path. HRDLIM can be set as high as  $\pm 80$ ppm and has  $\sim 1.2$ ppb resolution. The default limit is  $\pm 12$ ppm. When frequency limit detection is enabled by setting DPLLCR5.FLLOL = 1, if the DPLL frequency exceeds the hard limit the DPLL declares loss-of-lock. The hard frequency limit in ppb is equal to

$$\pm \text{HRDLIM}[15:0] \times R \times 1.226433036$$

where  $R = f_{\text{MCLK}} / 204.8\text{MHz}$  and  $f_{\text{MCLK}}$  is the nominal frequency of the DPLL's master clock (see section 5.3). If external reference switching mode is enabled during reset (see Section 5.5.3.5), the default value is configured to  $\pm 80$ ppm (FFFFh). The value 00h is undefined. See section 5.6.5.

**Register Name:** HRDLIM2  
**Register Description:** DPLL Hard Frequency Limit Register 2  
**Register Address:** 7Dh

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>Name</b>	HRDLIM[15:8]							
<b>Default</b>	0	0	1	0	0	1	1	0

The HRDLIM1 and HRDLIM2 registers must be read consecutively and written consecutively. See section 6.1.4.

**Bits 7 to 0: DPLL Hard Frequency Limit (HRDLIM[15:8]).** See the HRDLIM1 register description.

**Register Name:** SOFTLIM  
**Register Description:** DPLL Soft Frequency Limit Register  
**Register Address:** 7Eh

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>Name</b>	SOFTLIM[7:0]							
<b>Default</b>	0	0	0	1	1	0	1	0

**Bits 7 to 0: DPLL Soft Frequency Limit (SOFTLIM[7:0]).** This field is an unsigned integer that specifies the soft frequency limit for the DPLL. The soft limit is only used for monitoring; exceeding this limit does not cause loss-of-lock. The limit in ppm is equal to

$$\pm \text{SOFTLIM}[7:0] \times R \times 0.313966857.$$

where  $R = f_{\text{MCLK}} / 204.8\text{MHz}$  and  $f_{\text{MCLK}}$  is the nominal frequency of the DPLL's master clock (see section 5.3). The default value is approximately  $\pm 8.2$ ppm. When the DPLL frequency reaches the soft limit, the SOFT status bit is set in the PLL1SR register. The value 00h is undefined. See section 5.6.5.

**Register Name:** OFFSET1  
**Register Description:** DPLL Phase Offset Register 1  
**Register Address:** 80h

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>Name</b>	OFFSET[7:0]							
<b>Default</b>	0	0	0	0	0	0	0	0

The OFFSET1 and OFFSET2 registers must be read consecutively and written consecutively. See section 6.1.4.

**Bits 7 to 0: Phase Offset (OFFSET[7:0]).** The full 16-bit OFFSET[15:0] field spans this register and the [OFFSET2](#) register. OFFSET is a two's-complement signed integer that specifies the desired phase offset between the output of the DPLL and the selected input reference. The phase offset in picoseconds is equal to OFFSET[15:0] x actual\_internal\_clock\_period / 2<sup>11</sup>. If the internal clock is at its nominal frequency of 77.76MHz then the phase offset equation simplifies to OFFSET[15:0] x 6.279ps. If, however, the DPLL is locked to a reference whose frequency is +1ppm from ideal, for example, then the actual internal clock period is 1ppm shorter and the phase offset is 1ppm smaller. When the OFFSET field is written, the phase of the output clocks is automatically ramped to the new offset value to avoid loss of synchronization. The OFFSET field is ignored when the DPLL is not locked. See section 5.6.6.

Note: The DPLL cannot support a non-zero OFFSET value when transitioning to the Free-Run state. See the DPLL state diagram in [Figure 5-9](#) for the one state transition to the Free-Run state from the Prelocked state. To avoid this state transition when OFFSET≠0 do one of the following:

1. First step after device reset, with [MCR2.IC1EN](#) and [MCR2.IC2EN](#) both left at default values of 0, force the DPLL into the Digital Hold state ([DPLL2CR.STATE=010](#)) and then back to automatic state transitions ([DPLL2CR.STATE=000](#)). After reset the Digital Hold state behaves exactly the same as the Free-Run state (0ppm offset vs. the local oscillator).
2. Do not set the OFFSET field to a non-zero value until the DPLL is in one of these states: Locked, Loss-of-Lock, Digital Hold, Prelocked2 ([PLL1SR.STATE=010, 100, 101 or 111](#)). After the DPLL has reached one of these states it cannot return to the Free-Run state unless forced.

Also do not force the DPLL to the Free-Run state during operation when OFFSET≠0.

**Register Name:** OFFSET2  
**Register Description:** DPLL Phase Offset Register 2  
**Register Address:** 81h

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>Name</b>	OFFSET[15:8]							
<b>Default</b>	0	0	0	0	0	0	0	0

The OFFSET1 and OFFSET2 registers must be read consecutively and written consecutively. See section 6.1.4.

**Bits 7 to 0: Phase Offset (OFFSET[15:8]).** See the [OFFSET1](#) register description.

**Register Name:** VALCR1  
**Register Description:** Input Clock Valid Control Register 1  
**Register Address:** 82h

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>Name</b>	—	—	—	—	—	—	IC2	IC1
<b>Default</b>	1	1	1	1	1	1	1	1

**Bits 1 to 0: Input Clock Valid Control (IC2, IC1).** These control bits can be used to force input clocks to be considered invalid. If a clock is invalidated by one of these control bits it will not appear in the priority table in the [PTAB1](#) and [PTAB2](#) registers, even if the clock is otherwise valid. These bits are useful when system software needs to force clocks to be invalid in response to OAM commands. Note that setting a VALCR bit low has no effect on the corresponding bit in the [VALSR](#) register. See section [5.5.3.2](#).

0 = Force invalid

1 = Don't force invalid; determine validity normally

**Register Name:** IPR1  
**Register Description:** Input Priority Register 1  
**Register Address:** 83h

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>Name</b>	PRI2[3:0]				PRI1[3:0]			
<b>Default</b>	0	0	1	0	0	0	0	1

**Bits 7 to 4: Priority for Input Clock 2 (PRI2[3:0]).** This field specifies the priority of IC2. Priority 0001 is highest; priority 1111 is lowest. See section [5.5.3.1](#).

0000 = IC2 unavailable for selection.

0001–1111 = IC2 relative priority

**Bits 3 to 0: Priority for Input Clock 1 (PRI1[3:0]).** This field specifies the priority of IC1. Priority 0001 is highest; priority 1111 is lowest. See section [5.5.3.1](#).

0000 = IC1 unavailable for selection.

0001–1111 = IC1 relative priority

**Register Name:** PTAB1  
**Register Description:** Priority Table Register 1  
**Register Address:** 85h

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>Name</b>	REF1[3:0]				SELREF[3:0]			
<b>Default</b>	0	0	0	0	0	0	0	0

**Bits 7 to 4: Highest Priority Valid Reference (REF1[3:0]).** This real-time status field indicates the DPLL's highest-priority valid input reference. Note that an input reference cannot be indicated in this field if it has been marked invalid in the [VALCR1](#) register. When the DPLL is in nonrevertive mode ([DPLLCR1.REVERT](#) = 0) this field may not have the same value as the SELREF[3:0] field. See section [5.5.3.2](#).

0000 = No valid input reference available  
 0001 = IC1 input  
 0010 = IC2 input  
 0011 to 1111 = {unused values}

**Bits 3 to 0: Selected Reference (SELREF[3:0]).** This real-time status field indicates the DPLL's current selected reference. Note that an input clock cannot be indicated in this field if it has been marked invalid in the [VALCR1](#). When the DPLL is in nonrevertive mode ([DPLLCR1.REVERT](#) = 0) this field may not have the same value as the REF1[3:0] field. See section [5.5.3.2](#).

0000 = No valid input reference available  
 0001 = IC1 input  
 0010 = IC2 input  
 0011 to 1111 = {unused values}

**Register Name:** PTAB2  
**Register Description:** Priority Table Register 2  
**Register Address:** 86h

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>Name</b>	—	—	—	—	REF2[3:0]			
<b>Default</b>	0	0	0	0	0	0	0	0

**Bits 3 to 0: Second Highest Priority Valid Reference (REF2[3:0]).** This real-time status field indicates the DPLL's second highest priority validated input reference. Note that an input reference cannot be indicated in this field if it has been marked invalid in the [VALCR1](#) register. See section [5.5.3.2](#).

0000 = No valid input reference available  
 0001 = IC1 input  
 0010 = IC2 input  
 0011 to 1111 = {unused values}

**Register Name:** PHASE1  
**Register Description:** Phase Register 1  
**Register Address:** 87h

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>Name</b>	PHASE[7:0]							
<b>Default</b>	0	0	0	0	0	0	0	0

The PHASE1 and PHASE2 registers must be read consecutively. See section 6.1.4.

**Bits 7 to 0: Current DPLL Phase (PHASE[7:0]).** The full 16-bit PHASE[15:0] field spans this register and the PHASE2 register. PHASE is a two's-complement signed integer that indicates the current value of the phase detector (i.e. the phase difference between DPLL output and DPLL input). The value is the output of the phase averager. The averaged phase difference in degrees is equal to PHASE x 0.707. See section 5.6.7.

**Register Name:** PHASE2  
**Register Description:** Phase Register 2  
**Register Address:** 88h

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>Name</b>	PHASE[15:8]							
<b>Default</b>	0	0	0	0	0	0	0	0

The PHASE1 and PHASE2 registers must be read consecutively. See section 6.1.4.

**Bits 7 to 0: Current DPLL Phase (PHASE[15:8]).** See the PHASE1 register description.

**Register Name:** FREQ1  
**Register Description:** Frequency Register 1  
**Register Address:** 89h

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>Name</b>	FREQ[7:0]							
<b>Default</b>	0	0	0	0	0	0	0	0

The FREQ1 to FREQ4 registers must be read consecutively. See section 6.1.4.

**Bits 7 to 0: Current DPLL Frequency (FREQ[7:0]).** The full 32-bit FREQ[31:0] field spans this register, FREQ2, FREQ3 and FREQ4. This read-only field is a two's-complement signed integer that expresses the fractional frequency offset of the DPLL. The frequency in ppm is equal to

$$\text{FREQ}[31:0] \times R \times 3.7427766\text{E-}8$$

where  $R = f_{\text{MCLK}} / 204.8\text{MHz}$  and  $f_{\text{MCLK}}$  is the nominal frequency of the DPLL's master clock (see section 5.3). When DPLLCR6.RDAVG=0, the value in this field is derived from the DPLL integral path and can be considered a very short-term average frequency with a rate of change inversely proportional to the DPLL bandwidth. If DPLLCR6.LIMINT = 1, the value of FREQ freezes when the DPLL reaches its minimum or maximum frequency. When DPLLCR6.RDAVG≠0, the value in this field is one of the longer-term frequency averages computed by the DPLL. See section 5.6.1.6.

Note: After DPLLCR6.RDAVG is changed, system software must wait at least 50μs before reading the corresponding value from the FREQ field.

The reference clock for DPLL frequency measurement is the internal master clock (see section 5.3.3). This means the device counts the number of DPLL clock cycles that occur in an interval of time equal to a specific number of local oscillator clock periods. It then compares the actual count to the expected count to determine the fractional frequency offset of the DPLL vs. the fractional frequency offset of the local oscillator. Thus DPLL frequency measurements are relative. If the DPLL's input clock is known to have worse frequency accuracy than the local oscillator then the FREQ field can be assumed to indicate the fractional frequency offset of the input clock. If,

however, the DPLL's input clock is known to have much better frequency accuracy than the local oscillator then the FREQ field actually indicates the fractional frequency offset of the local oscillator.

**Register Name:** FREQ2  
**Register Description:** Frequency Register 2  
**Register Address:** 8Ah

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>Name</b>	FREQ[15:8]							
<b>Default</b>	0	0	0	0	0	0	0	0

The FREQ1 to FREQ4 registers must be read consecutively. See section [6.1.4](#).

**Bits 7 to 0: Current DPLL Frequency (FREQ[15:8]).** See the [FREQ1](#) register description.

**Register Name:** FREQ3  
**Register Description:** Frequency Register 3  
**Register Address:** 8Bh

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>Name</b>	FREQ[23:16]							
<b>Default</b>	0	0	0	0	0	0	0	0

The FREQ1 to FREQ4 registers must be read consecutively. See section [6.1.4](#).

**Bits 7 to 0: Current DPLL Frequency (FREQ[23:16]).** See the [FREQ1](#) register description.

**Register Name:** FREQ4  
**Register Description:** Frequency Register 4  
**Register Address:** 8Ch

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>Name</b>	FREQ[31:24]							
<b>Default</b>	0	0	0	0	0	0	0	0

The FREQ1 to FREQ4 registers must be read consecutively. See section [6.1.4](#).

**Bits 7 to 0: Current DPLL Frequency (FREQ[31:24]).** See the [FREQ1](#) register description.

**Register Name:** DFSCR1  
**Register Description:** DFS Configuration Register 1  
**Register Address:** 8Dh

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>Name</b>	DFSREQ[3:0]				—	—	—	—
<b>Default</b>	0	0	0	0	0	0	0	0

**Bits 7 to 4: DFS Frequency (DFSREQ[3:0]).** This field sets the frequency of the DPLL's output DFS block. See section 5.7.1.2.

When the DPLL's nominal master clock frequency is 204.8MHz, the following options are available:

- 0000 = Disabled (DFS output clock held low)
- 0001 = 77.760MHz (SONET/SDH)
- 0010 = 62.500MHz (Ethernet)
- 0011 = 49.152MHz (24 x E1)
- 0100 = 65.536MHz (32 x E1)
- 0101 = 74.112MHz (48 x DS1)
- 0110 = 68.736MHz (2 x E3)
- 0111 = 44.736MHz (DS3)
- 1000 = 50.496MHz (8 x 6312kHz)
- 1001 = 61.440MHz (2 x 30.72MHz, 6 x 10.24MHz)
- 1010 = 52.000MHz (4 x 13MHz)
- 1011 = 40.000MHz (4 x 10MHz)
- 1100 = 50.000MHz (2 x 25MHz)
- 1101 = 60.000MHz
- 1110 = 70.000MHz
- 1111 = Programmable DFS mode

When the DPLL's nominal master clock frequency is not 204.8MHz the following options are available:

- 0000 = Disabled (DFS output clock held low)
- 0010 = 62.500MHz (Ethernet)
- 0101 = 74.112MHz (48 x DS1)
- 1001 = 61.440MHz (2 x 30.72MHz, 6 x 10.24MHz)
- 1101 = 60.000MHz
- 1110 = 70.000MHz

Other values are not recommended.

**Register Name:** MCFREQ1  
**Register Description:** Master Clock Frequency Adjustment Register 1  
**Register Address:** 8Eh

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>Name</b>	MCFREQ[7:0]							
<b>Default</b>	0	0	0	0	0	0	0	0

The MCFREQ1 and MCFREQ2 registers must be read consecutively and written consecutively. See section 6.1.4.

**Bits 7 to 0: Master Clock Frequency Adjustment (MCFREQ[7:0]).** The full 16-bit MCFREQ[15:0] field spans this register and MCFREQ2. MCFREQ is an unsigned integer that tells the input block and the DPLL how to compensate for any known difference between the actual frequency of the signal on the MCLKOSCP/N pins and the nominal master clock frequency specified by [MCDNOM](#) and [MCINOM](#). The resolution of MCFREQ is ~2.5ppb. The range of MCFREQ values allows compensation for master clock oscillator frequencies up to ±80ppm.

Positive MCFREQ values effectively increase the frequency of the input block and the DPLL vs. the master clock. Negative MCFREQ values effectively decrease the frequency of the input block and the DPLL vs. the master clock. For example, if the MCLKOSCP/N signal has an offset of +1ppm, the adjustment should be -1ppm to correct the offset. The formulas below translate adjustments to register values and vice versa. The default register value of 32,768 corresponds to 0ppm. See section 5.3.

$$\begin{aligned} \text{MCFREQ}[23:0] &= \text{adjustment\_in\_ppm} / (R \times 0.002452866) + 32,768 \\ \text{adjustment\_in\_ppm} &= (\text{MCFREQ}[23:0] - 32,768) \times R \times 0.002452866 \end{aligned}$$

where  $R = f_{\text{MCLK}} / 204.8\text{MHz}$  and  $f_{\text{MCLK}}$  is the nominal frequency of the DPLL's master clock (see section 5.3).

Note that in APLL-only mode this field has no effect, but similar frequency adjustments (ppb or ppm) can be made in the APLLs' high-resolution fractional feedback divider value, [AFBDIV](#).

**Register Name:** MCFREQ2  
**Register Description:** Master Clock Frequency Adjustment Register 2  
**Register Address:** 8Fh

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>Name</b>	MCFREQ[15:8]							
<b>Default</b>	1	0	0	0	0	0	0	0

The MCFREQ1 and MCFREQ2 registers must be read consecutively and written consecutively. See section 6.1.4.

**Bits 7 to 0: Master Clock Frequency Adjustment (MCFREQ[15:8]).** See the [MCFREQ1](#) register description.



**Register Name:** MCDNOM1  
**Register Description:** Master Clock DPLL Nominal Frequency Register 1  
**Register Address:** 90h

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>Name</b>	MCDNOM[7:0]							
<b>Default</b>	0	0	0	0	0	0	0	0

**Bits 7 to 0: Master Clock DPLL Nominal Frequency (MCDNOM[7:0]).** The full 26-bit MCDNOM[25:0] field spans this register through MCDNOM4. MCDNOM is a two's-complement signed integer that specifies to the DPLL the nominal frequency of the master clock. The nominal frequency must be between 190MHz and 208.333MHz. Typical nominal frequency values are 200.0MHz and 204.8MHz. See section [5.2.2](#).

The formulas below translate nominal\_frequency to MCDNOM register values and vice versa. The default register value of 0 corresponds to 204.8MHz.

$$\begin{aligned}
 \text{MCDNOM}[25:0] &= ((204,800,000 / \text{nominal\_frequency}) - 1) \times 1,000,000 / 0.002452866 \\
 \text{nominal\_frequency} &= 204,800,000 / (\text{MCDNOM}[25:0] \times 0.002452866 / 1,000,000 + 1)
 \end{aligned}$$

**Register Name:** MCDNOM2  
**Register Description:** Master Clock DPLL Nominal Frequency Register 2  
**Register Address:** 91h

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>Name</b>	MCDNOM[15:8]							
<b>Default</b>	0	0	0	0	0	0	0	0

**Bits 7 to 0: Master Clock DPLL Nominal Frequency (MCDNOM[15:8]).** See the [MCDNOM1](#) register description.

**Register Name:** MCDNOM3  
**Register Description:** Master Clock DPLL Nominal Frequency Register 3  
**Register Address:** 92h

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>Name</b>	MCDNOM[23:16]							
<b>Default</b>	0	0	0	0	0	0	0	0

**Bits 7 to 0: Master Clock DPLL Nominal Frequency (MCDNOM[23:16]).** See the [MCDNOM1](#) register description.

**Register Name:** MCDNOM4  
**Register Description:** Master Clock DPLL Nominal Frequency Register 4  
**Register Address:** 93h

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>Name</b>	—	—	—	—	—	—	MCDNOM[25:24]	
<b>Default</b>	0	0	0	0	0	0	0	0

**Bits 1 to 0: Master Clock DPLL Nominal Frequency (MCDNOM[25:24]).** See the [MCDNOM1](#) register description.

**Register Name:** MCINOM1  
**Register Description:** Master Clock Input-Block Nominal Frequency Register 1  
**Register Address:** 94h

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>Name</b>	MCINOM[7:0]							
<b>Default</b>	0	0	0	0	0	0	0	0

**Bits 7 to 0: Master Clock Input-Block Nominal Frequency (MCINOM[7:0]).** The full 17-bit MCINOM[16:0] field spans this register through [MCINOM3](#). MCINOM is a two's-complement signed integer that specifies to the input block the nominal frequency of the master clock. The nominal frequency must be between 190MHz and 208.333MHz. Typical nominal frequency values are 200.0MHz and 204.8MHz. See section [5.2.2](#).

The formulas below translate nominal\_frequency to MCINOM register values and vice versa. The default register value of 0 corresponds to 204.8MHz.

$$\begin{aligned}
 \text{MCINOM}[16:0] &= (204,800,000 / 500) - (\text{nominal\_frequency} / 500) \\
 \text{nominal\_frequency} &= 204,800,000 - 500 \times \text{MCINOM}[16:0]
 \end{aligned}$$

**Register Name:** MCINOM2  
**Register Description:** Master Clock Input-Block Nominal Frequency Register 2  
**Register Address:** 95h

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>Name</b>	MCINOM[15:8]							
<b>Default</b>	0	0	0	0	0	0	0	0

**Bits 7 to 0: Master Clock Input-Block Nominal Frequency (MCINOM[15:8]).** See the [MCINOM1](#) register description.

**Register Name:** MCINOM3  
**Register Description:** Master Clock Input-Block Nominal Frequency Register 3  
**Register Address:** 96h

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>Name</b>	—	—	—	—	—	—	—	MCINOM16
<b>Default</b>	0	0	0	0	0	0	0	0

**Bit 0: Master Clock Input-Block Nominal Frequency (MCINOM[16]).** See the [MCINOM1](#) register description.

**Register Name:** MCAC1  
**Register Description:** Master Clock Adjust Count Register 1  
**Register Address:** 97h

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>Name</b>	MCAC[7:0]							
<b>Default</b>	0	0	0	0	0	0	0	0

**Bits 7 to 0: Master Clock Adjust Count (MCAC[7:0]).** The full 9-bit MCAC[8:0] field spans this register through [MCAC2](#). MCAC is a two's-complement signed integer that must be set as shown below for proper operation of the input block. See section [5.3.3](#).

$N = \text{round}(f_{\text{MCLK}} / 500\text{Hz})$  where  $f_{\text{MCLK}}$  is the nominal frequency of the DPLL's master clock in Hz

if [ICCR4.FMRES](#) = 0

$\text{MCAC}[8:0] = \text{round}((2,000,000 / (0.002452866 * N) - 1991) / 16)$

if [ICCR4.FMRES](#) = 1

$\text{MCAC}[8:0] = \text{round}(2,000,000 / (0.002452866 * N) - 1991)$

**Register Name:** MCAC2  
**Register Description:** Master Clock Adjust Count Register 2  
**Register Address:** 98h

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>Name</b>	—	—	—	—	—	—	—	MCAC[8]
<b>Default</b>	0	0	0	0	0	0	0	0

**Bit 0: Master Clock Adjust Count (MCAC[8]).** See the [MCAC1](#) register description.

**Register Name:** HOFREQ1  
**Register Description:** Digital Hold Frequency Register 1  
**Register Address:** 9Ch

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>Name</b>	HOFREQ[7:0]							
<b>Default</b>	0	0	0	0	0	0	0	0

The HOFREQ1 to HOFREQ4 registers must be read consecutively and written consecutively. See section [6.1.4](#).

**Bits 7 to 0: Digital Hold Frequency (HOFREQ[7:0]).** The full 32-bit HOFREQ[31:0] field spans this register, [HOFREQ2](#), [HOFREQ3](#) and [HOFREQ4](#). HOFREQ is a two's-complement signed integer that specifies the digital hold frequency as a fractional frequency offset with respect to the nominal frequency. The HOFREQ field has the same size and format as the [FREQ](#) field to allow software to read FREQ, filter the value, and then write to HOFREQ. Digital hold offset in ppm is equal to

$$\text{HOFREQ}[31:0] \times R \times 3.7427766\text{E-}8.$$

where  $R = f_{\text{MCLK}} / 204.8\text{MHz}$  and  $f_{\text{MCLK}}$  is the nominal frequency of the DPLL's master clock (see section [5.3](#)). See section [5.6.1.6](#).

Note: bit 0 at address 205h must be set to 1 for HOFREQ to behave as described.

**Register Name:** HOFREQ2  
**Register Description:** Digital Hold Frequency Register 2  
**Register Address:** 9Dh

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>Name</b>	HOFREQ[15:8]							
<b>Default</b>	0	0	0	0	0	0	0	0

The HOFREQ1 to HOFREQ4 registers must be read consecutively and written consecutively. See section [6.1.4](#).

**Bits 7 to 0: Digital Hold Frequency (HOFREQ[15:8]).** See the [HOFREQ1](#) register description.

**Register Name:** HOFREQ3  
**Register Description:** Digital Hold Frequency Register 3  
**Register Address:** 9Eh

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>Name</b>	HOFREQ[23:16]							
<b>Default</b>	0	0	0	0	0	0	0	0

The HOFREQ1 to HOFREQ4 registers must be read consecutively and written consecutively. See section [6.1.4](#).

**Bits 7 to 0: Digital Hold Frequency (HOFREQ[23:16]).** See the [HOFREQ1](#) register description.

**Register Name:** HOFREQ4  
**Register Description:** Digital Hold Frequency Register 4  
**Register Address:** 9Fh

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>Name</b>	HOFREQ[31:24]							
<b>Default</b>	0	0	0	0	0	0	0	0

The HOFREQ1 to HOFREQ4 registers must be read consecutively and written consecutively. See section [6.1.4](#).

**Bits 7 to 0: Digital Hold Frequency (HOFREQ[31:24]).** See the [HOFREQ1](#) register description.

### 6.3.7 DPLL and Input Block Status Registers

**Register Name:** PLL1SR  
**Register Description:** DPLL Status Register  
**Register Address:** A0h

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>Name</b>	—	—	—	PALARM	SOFT	STATE[2:0]		
<b>Default</b>	0	0	0	0	0	0	0	1

**Bit 4: DPLL Phase Alarm (PALARM).** This real-time status bit indicates the state of the DPLL's phase lock detector. See section 5.6.5. (NOTE: This is not the same as STATE = Locked.)

0 = DPLL phase-lock parameters are met (as determined by DPLLCR5.NALOL, FLLOL, FLEN, CLEN)

1 = DPLL loss of phase lock

**Bit 3: DPLL Frequency Soft Alarm (SOFT).** This real-time status bit indicates whether or not the DPLL is tracking its reference within the soft alarm limits specified in the SOFTLIM register. See section 5.6.5.

0 = No alarm; frequency is within the soft alarm limits

1 = Soft alarm; frequency is outside the soft alarm limits

**Bits 2 to 0: DPLL Operating State (STATE[2:0]).** This real-time status field indicates the current state of the DPLL state machine. Values not listed below correspond to invalid (unused) states. See section 5.6.1.

001 = Free-run

010 = Digital hold

100 = Locked

101 = Prelocked 2

110 = Prelocked

111 = Loss-of-lock

**Register Name:** PLL1LSR  
**Register Description:** DPLL Latched Status Register  
**Register Address:** A1h

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Name	MCFAIL	—	—	STATE	SRFAIL	NOIN	—	—
Default	0	0	0	0	0	0	0	0

**Bit 7: MCLK Oscillator Failure (MCFAIL).** This latched status bit is set to 1 when the device detects that the MCLKOSC signal is not toggling or is grossly off frequency. MCFAIL is cleared when written with a 1. After being cleared, MCFAIL is not set again if the MCLK signal remains grossly off frequency, but it is set again if the MCLK signal is not toggling at all. When MCFAIL is set it can cause an interrupt request if the [PLL1IER.MCFAIL](#) interrupt enable bit is set. See section [5.3.3](#).

**Bit 4: DPLL State Change (STATE).** This latched status bit is set to 1 when the operating state of the DPLL changes. STATE is cleared when written with a 1 and not set again until the DPLL operating state changes again. When STATE is set it can cause an interrupt request if the [PLL1IER.STATE](#) interrupt enable bit is set. The current operating state can be read from [PLL1SR.STATE](#). See section [5.6.1](#).

**Bit 3: DPLL Selected Reference Failed (SRFAIL).** This latched status bit is set to 1 when the DPLL's selected reference fails, (i.e., no clock edges in a few clock cycles). SRFAIL is cleared when written with a 1. When SRFAIL is set it can cause an interrupt request if the [PLL1IER.SRFAIL](#) interrupt enable bit is set. SRFAIL is not set in free-run or digital hold states. See section [5.5.2.3](#).

**Bit 2: DPLL No Valid Inputs Alarm (NOIN).** This latched status bit is set to 1 when the DPLL has no valid inputs available. NOIN is cleared when written with a 1 unless the DPLL still has no valid inputs available. When NOIN is set it can cause an interrupt request if the [PLL1IER.NOIN](#) interrupt enable bit is set.

**Register Name:** VALSR1  
**Register Description:** Input Clock Valid Status Register 1  
**Register Address:** A2h

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Name	—	—	—	—	—	—	IC2	IC1
Default	0	0	0	0	0	0	0	0

**Bits 1 to 0: Input Clock Valid Status (IC2, IC1).** Each of these real-time status bits is set to 1 when the corresponding input clock is valid. An input is valid if it has no active alarms (HARD = 0, ACT = 0, LOCK = 0 in the [ISR1](#) register). See also the [ICLSR1](#) register and Section [5.5.2](#).

0 = Invalid

1 = Valid

**Register Name:** ICLSR1  
**Register Description:** Input Clock Latched Status Register 1  
**Register Address:** A3h

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Name	—	—	—	—	—	—	IC2	IC1
Default	1	1	1	1	1	1	1	1

**Bits 1 to 0: Input Clock Status Change (IC2, IC1).** Each of these latched status bits is set to 1 when the corresponding **VALSR1** status bit changes state (set or cleared). If soft frequency limit alarms are enabled (**ICCR2.SOFTEN** = 1), then each of these latched status bits is also set to 1 when the corresponding **ISR.SOFT** bit changes state (set or cleared). Each bit is cleared when written with a 1 and not set again until the **VALSR1** bit (or **SOFT** bit) changes state again. When one of these latched status bits is set it can cause an interrupt request if the corresponding interrupt enable bit is set in the **ICIER1** register. See section 5.5.2 for input clock validation/invalidation criteria.

**Register Name:** ISR1  
**Register Description:** Input Status Register 1  
**Register Address:** A4h

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Name	<u>SOFT2</u>	<u>HARD2</u>	<u>ACT2</u>	<u>LOCK2</u>	<u>SOFT1</u>	<u>HARD1</u>	<u>ACT1</u>	<u>LOCK1</u>
Default	0	1	1	0	0	1	1	0

**Bit 7: Soft Frequency Limit Alarm for Input Clock 2 (SOFT2).** This bit has the same behavior as the **SOFT1** bit but for the **IC2** input clock.

**Bit 6: Hard Frequency Limit Alarm for Input Clock 2 (HARD2).** This bit has the same behavior as the **HARD1** bit but for the **IC2** input clock.

**Bit 5: Activity Alarm for Input Clock 2 (ACT2).** This bit has the same behavior as the **ACT1** bit but for the **IC2** input clock.

**Bit 4: Phase Lock Alarm for Input Clock 2 (LOCK2).** This bit has the same behavior as the **LOCK1** bit but for the **IC2** input clock.

**Bit 3: Soft Frequency Limit Alarm for Input Clock 1 (SOFT1).** This real-time status bit indicates a soft frequency limit alarm for input clock 1. **SOFT1** is set to 1 when the frequency of **IC1** is greater than or equal to the soft limit set in the **ICSLIM** register. Soft alarms are disabled by default but can be enabled by setting **ICCR2.SOFTEN** = 1. A soft alarm does not invalidate an input clock. See section 5.5.2.1.

**Bit 2: Hard Frequency Limit Alarm for Input Clock 1 (HARD1).** This real-time status bit indicates a hard frequency limit alarm for input clock 1. **HARD1** is set to 1 when the frequency of **IC1** is greater than or equal to the rejection hard limit set in the **ICRHLIM** register. **HARD1** is set to 0 when the frequency of **IC1** is less than or equal to the acceptance hard limit set in the **IC AHLIM** register. Hard alarms are enabled by default but can be disabled by setting **ICCR2.HARDEN** = 0. A hard alarm clears the **IC1** status bit in the **VALSR1** register, invalidating the **IC1** clock. See section 5.5.2.1.

**Bit 1: Activity Alarm for Input Clock 1 (ACT1).** This real-time status bit is set to 1 when the leaky bucket accumulator for **IC1** reaches the alarm threshold specified in the **ICLBU** register. An activity alarm clears the **IC1** status bit in the **VALSR1** register, invalidating the **IC1** clock. See section 5.5.2.2.

**Bit 0: Phase Lock Alarm for Input Clock 1 (LOCK1).** This status bit is set to 1 if **IC1** is the selected reference for the **DPLL** and the **DPLL** cannot lock to it within the duration specified in the **PHLKTO** register (default = 100 seconds). A phase lock alarm clears the **IC1** status bit in **VALSR1**, invalidating the **IC1** clock. **LOCK1** can be automatically cleared after a programmable timeout period specified in the **LKATO** register (default = 100 seconds). System software can clear **LOCK1** by writing 0 to it, but writing 1 is ignored. See section 5.6.1.4.

**Register Name:** PLL1IER  
**Register Description:** DPLL Interrupt Enable Register  
**Register Address:** A6h

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>Name</b>	MCFAIL	—	—	STATE	SRFAIL	NOIN	—	—
<b>Default</b>	0	0	0	0	0	0	0	0

**Bit 7: Interrupt Enable for MCLK Oscillator Failure (MCFAIL).** This bit is an interrupt enable for the MCFAIL bit in the [PLL1LSR](#) register.

- 0 = Mask the interrupt
- 1 = Enable the interrupt

**Bit 4: Interrupt Enable for DPLL State Change (STATE).** This bit is an interrupt enable for the STATE bit in the [PLL1LSR](#) register.

- 0 = Mask the interrupt
- 1 = Enable the interrupt

**Bit 3: Interrupt Enable for DPLL Selected Reference Failed (SRFAIL).** This bit is an interrupt enable for the SRFAIL bit in the [PLL1LSR](#) register.

- 0 = Mask the interrupt
- 1 = Enable the interrupt

**Bit 2: Interrupt Enable for DPLL No Valid Inputs Alarm (NOIN).** This bit is an interrupt enable for the NOIN bit in the [PLL1LSR](#) register.

- 0 = Mask the interrupt
- 1 = Enable the interrupt

**Register Name:** ICIER1  
**Register Description:** Input Clock Interrupt Enable Register 1  
**Register Address:** A7h

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>Name</b>	—	—	—	—	—	—	IC2	IC1
<b>Default</b>	0	0	0	0	0	0	0	0

**Bits 1 to 0: Interrupt Enable for Input Clock Status Change (IC2, IC1).** Each of these bits is an interrupt enable control for the corresponding bit in the [ICLSR1](#) register.

- 0 = Mask the interrupt
- 1 = Enable the interrupt



## 7. JTAG and Boundary Scan

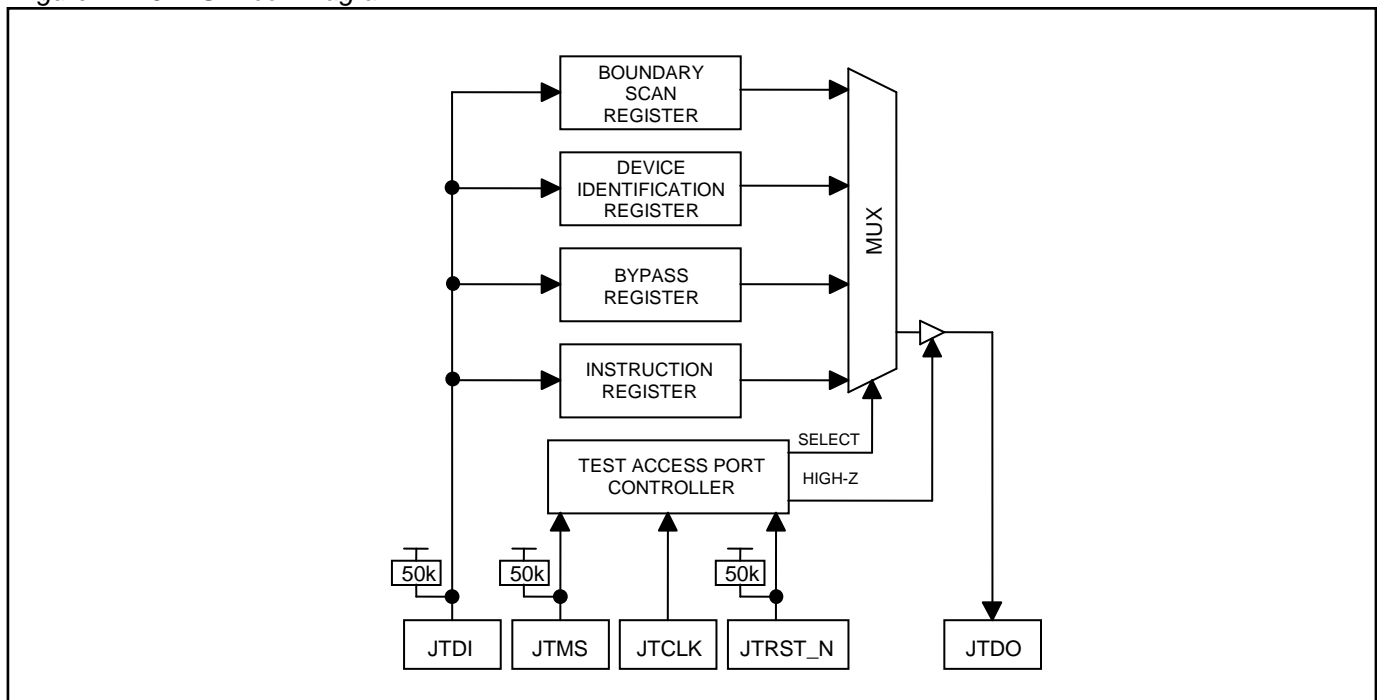
### 7.1 JTAG Description

The device supports the standard instruction codes SAMPLE/PRELOAD, BYPASS, and EXTEST. Optional public instructions included are HIGHZ, CLAMP, and IDCODE. Figure 7-1 shows a block diagram. The device contains the following items, which meet the requirements set by the IEEE 1149.1 Standard Test Access Port and Boundary Scan Architecture:

Test Access Port (TAP)	Bypass Register
TAP Controller	Boundary Scan Register
Instruction Register	Device Identification Register

The TAP has the necessary interface pins, namely JTCLK, JTRST\_N, JTDI, JTDO, and JTMS. Details on these pins can be found in Table 4-6. Details about the boundary scan architecture and the TAP can be found in IEEE 1149.1-1990, IEEE 1149.1a-1993, and IEEE 1149.1b-1994.

Figure 7-1. JTAG Block Diagram



## 7.2 JTAG TAP Controller State Machine Description

This section discusses the operation of the TAP controller state machine. The TAP controller is a finite state machine that responds to the logic level at JTMS on the rising edge of JTCLK. Each of the states denoted in [Figure 7-2](#) is described in the following paragraphs.

**Test-Logic-Reset.** Upon device power-up, the TAP controller starts in the Test-Logic-Reset state. The instruction register contains the IDCODE instruction. All system logic on the device operates normally.

**Run-Test-Idle.** Run-Test-Idle is used between scan operations or during specific tests. The instruction register and all test registers remain idle.

**Select-DR-Scan.** All test registers retain their previous state. With JTMS low, a rising edge of JTCLK moves the controller into the Capture-DR state and initiates a scan sequence. JTMS high moves the controller to the Select-IR-SCAN state.

**Capture-DR.** Data can be parallel-loaded into the test register selected by the current instruction. If the instruction does not call for a parallel load or the selected test register does not allow parallel loads, the register remains at its current value. On the rising edge of JTCLK, the controller goes to the Shift-DR state if JTMS is low or to the Exit1-DR state if JTMS is high.

**Shift-DR.** The test register selected by the current instruction is connected between JTDI and JTDO and data is shifted one stage toward the serial output on each rising edge of JTCLK. If a test register selected by the current instruction is not placed in the serial path, it maintains its previous state.

**Exit1-DR.** While in this state, a rising edge on JTCLK with JTMS high puts the controller in the Update-DR state, which terminates the scanning process. A rising edge on JTCLK with JTMS low puts the controller in the Pause-DR state.

**Pause-DR.** Shifting of the test registers is halted while in this state. All test registers selected by the current instruction retain their previous state. The controller remains in this state while JTMS is low. A rising edge on JTCLK with JTMS high puts the controller in the Exit2-DR state.

**Exit2-DR.** While in this state, a rising edge on JTCLK with JTMS high puts the controller in the Update-DR state and terminates the scanning process. A rising edge on JTCLK with JTMS low puts the controller in the Shift-DR state.

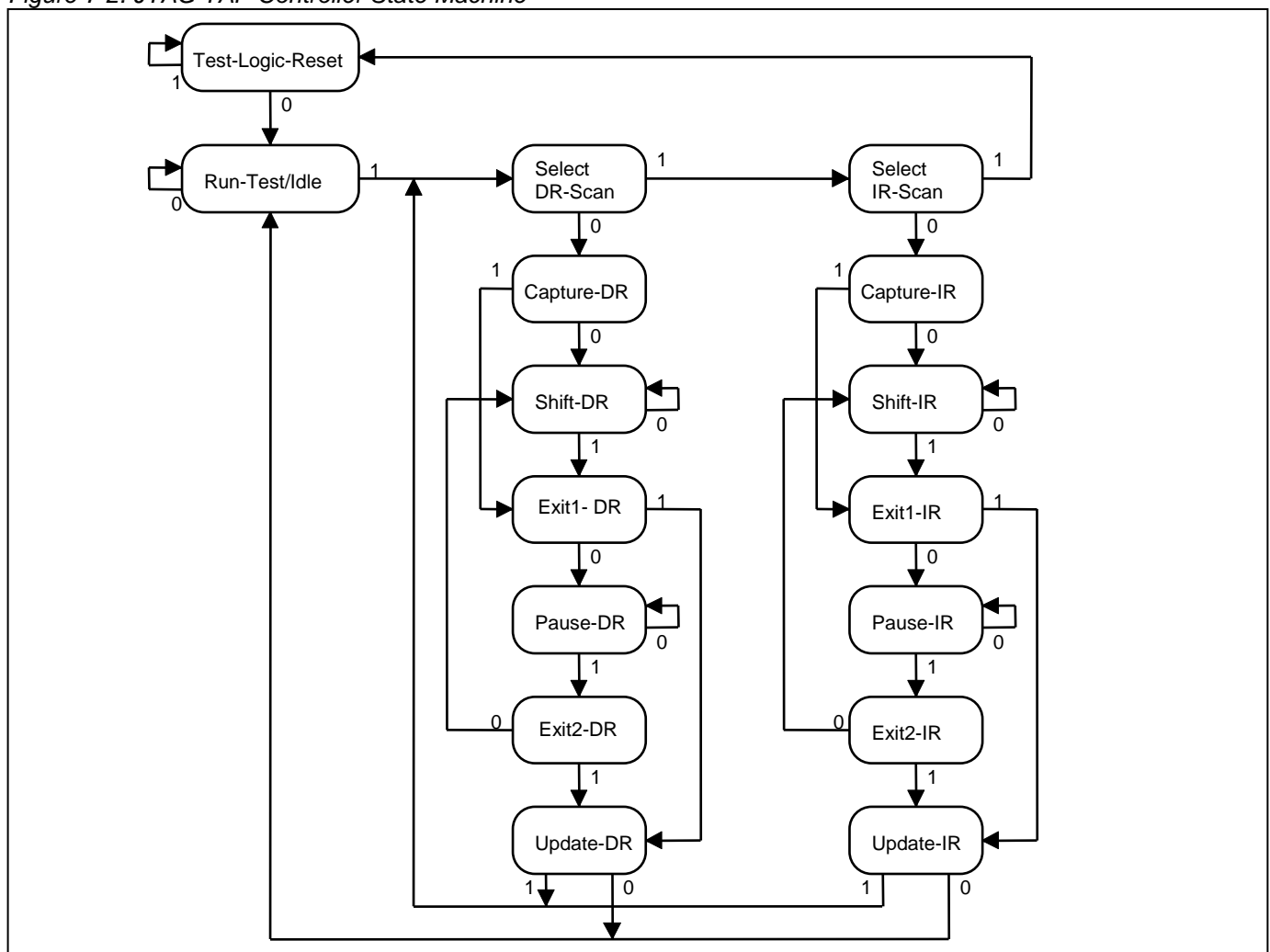
**Update-DR.** A falling edge on JTCLK while in the Update-DR state latches the data from the shift register path of the test registers into the data output latches. This prevents changes at the parallel output because of changes in the shift register. A rising edge on JTCLK with JTMS low puts the controller in the Run-Test-Idle state. With JTMS high, the controller enters the Select-DR-Scan state.

**Select-IR-Scan.** All test registers retain their previous state. The instruction register remains unchanged during this state. With JTMS low, a rising edge on JTCLK moves the controller into the Capture-IR state and initiates a scan sequence for the instruction register. JTMS high during a rising edge on JTCLK puts the controller back into the Test-Logic-Reset state.

**Capture-IR.** The Capture-IR state is used to load the shift register in the instruction register with a fixed value. This value is loaded on the rising edge of JTCLK. If JTMS is high on the rising edge of JTCLK, the controller enters the Exit1-IR state. If JTMS is low on the rising edge of JTCLK, the controller enters the Shift-IR state.

**Shift-IR.** In this state, the instruction register's shift register is connected between JTDI and JTDO and shifts data one stage for every rising edge of JTCLK toward the serial output. The parallel register and the test registers remain at their previous states. A rising edge on JTCLK with JTMS high moves the controller to the Exit1-IR state. A rising edge on JTCLK with JTMS low keeps the controller in the Shift-IR state, while moving data one stage through the instruction shift register.

**Update-IR.** The instruction shifted into the instruction shift register is latched into the parallel output on the falling edge of JTCLK as the controller enters this state. Once latched, this instruction becomes the current instruction. A rising edge on JTCLK with JTMS low puts the controller in the Run-Test-Idle state. With JTMS high, the controller enters the Select-DR-Scan state.



### 7.3 JTAG Instruction Register and Instructions

The instruction register contains a shift register as well as a latched parallel output and is 3 bits in length. When the TAP controller enters the Shift-IR state, the instruction shift register is connected between JTDI and JTDO. While in the Shift-IR state, a rising edge on JTCLK with JTMS low shifts data one stage toward the serial output at JTDO. A rising edge on JTCLK in the Exit1-IR state or the Exit2-IR state with JTMS high moves the controller to the Update-IR state. The falling edge of that same JTCLK latches the data in the instruction shift register to the instruction parallel output. [Table 7-1](#) shows the instructions supported and their respective operational binary codes.

**Table 7-1. JTAG Instruction Codes**

INSTRUCTIONS	SELECTED REGISTER	INSTRUCTION CODES
SAMPLE/PRELOAD	Boundary Scan	010
BYPASS	Bypass	111
EXTEST	Boundary Scan	000
CLAMP	Bypass	011
HIGHZ	Bypass	100
IDCODE	Device Identification	001

**SAMPLE/PRELOAD.** SAMPLe/PRELOAD is a mandatory instruction for the IEEE 1149.1 specification. This instruction supports two functions. First, the digital I/Os of the device can be sampled at the boundary scan register, using the Capture-DR state, without interfering with the device's normal operation. Second, data can be shifted into the boundary scan register through JTDI using the Shift-DR state.

**EXTEST.** EXTEST allows testing of the interconnections to the device. When the EXTEST instruction is latched in the instruction register, the following actions occur: (1) Once the EXTEST instruction is enabled through the Update-IR state, the parallel outputs of the digital output pins are driven. (2) The boundary scan register is connected between JTDI and JTDO. (3) The Capture-DR state samples all digital inputs into the boundary scan register.

**BYPASS.** When the BYPASS instruction is latched into the parallel instruction register, JTDI is connected to JTDO through the 1-bit bypass register. This allows data to pass from JTDI to JTDO without affecting the device's normal operation.

**IDCODE.** When the IDCODE instruction is latched into the parallel instruction register, the device identification register is selected. The device ID code is loaded into the device identification register on the rising edge of JTCLK, following entry into the Capture-DR state. Shift-DR can be used to shift the ID code out serially through JTDO. During Test-Logic-Reset, the ID code is forced into the instruction register's parallel output.

**HIGHZ.** All digital outputs are placed into a high-impedance state. The bypass register is connected between JTDI and JTDO.

**CLAMP.** All digital output pins output data from the boundary scan parallel output while connecting the bypass register between JTDI and JTDO. The outputs do not change during the CLAMP instruction.

## **7.4 JTAG Test Registers**

IEEE 1149.1 requires a minimum of two test registers—the bypass register and the boundary scan register. An optional test register, the identification register, has been included in the device design. It is used with the IDCODE instruction and the Test-Logic-Reset state of the TAP controller.

**Bypass Register.** This is a single 1-bit shift register used with the BYPASS, CLAMP, and HIGHZ instructions to provide a short path between JTDI and JTDO.

**Boundary Scan Register.** This register contains a shift register path and a latched parallel output for control cells and digital I/O cells. BSDL files are available on the MAX24605/MAX24610 page of Microsemi's website.

**Identification Register.** This register contains a 32-bit shift register and a 32-bit latched parallel output. It is selected during the IDCODE instruction and when the TAP controller is in the Test-Logic-Reset state. The device identification code for the MAX24605 and MAX24610 are shown in Table 7-2.

**Table 7-2. JTAG ID Code**

DEVICE	REVISION	DEVICE CODE	MANUFACTURER CODE	REQUIRED
MAX24605	Contact factory	0000 0000 1100 1000	000101000001	1
MAX24610	Contact factory	0000 0000 1100 1001	000101000001	1

## 8. Electrical Characteristics

### ABSOLUTE MAXIMUM RATINGS

Voltage Range on Any Pin with Respect to $V_{SS}$ (except Power Supply Pins)	-0.3V to +5.5V
Supply Voltage Range, Nominal 1.8V Supply with Respect to $V_{SS}$	-0.3V to +1.98V
Supply Voltage Range, Nominal 3.3V Supply with Respect to $V_{SS}$	-0.3V to +3.63V
Supply Voltage Range, VDDOx (x=A B C D) with Respect to $V_{SS}$	-0.3V to +3.63V
Ambient Operating Temperature Range	-40°C to +85°C
Junction Operating Temperature Range	-40°C to +125°C
Storage Temperature Range	-55°C to +125°C
Soldering Temperature (reflow)	
Lead (Pb) free	+260°C
Containing lead (Pb)	+240°C

Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of the specifications is not implied. Exposure to the absolute maximum rating conditions for extended periods may affect device. Ambient operating temperature range when device is mounted on a four-layer JEDEC test board with no airflow.

**Note 1:** The typical values listed in the tables of Section 8 are not production tested.

**Note 2:** Specifications to -40°C are guaranteed by design and not production tested.

**Table 8-1. Recommended DC Operating Conditions**

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Supply Voltage, Nominal 1.8V	VDD18		1.71	1.8	1.89	V
Supply Voltage, Nominal 3.3V	VDD33		3.135	3.3	3.465	V
Supply Voltage, VDDOx (x=A B C D)	VDDOx		1.425	1.5, 1.8, 2.5, 3.3	3.465	V
Ambient Temperature Range	$T_A$		-40		+85	°C
Junction Temperature Range	$T_J$		-40		+125	°C

**Table 8-2. Electrical Characteristics: Supply Currents**

(1.8V Supplies: 1.8V  $\pm 5\%$ ; 3.3V Supplies: 3.3V  $\pm 5\%$ ,  $T_A = -40^\circ\text{C}$  to  $+85^\circ\text{C}$ )

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP <sup>2</sup>	MAX	UNITS
MAX24605 Total Current, All 1.8V Supply Pins	$I_{DD18}$	Note 1		359	440	mA
MAX24605 Total Current, All 3.3V Supply Pins	$I_{DD33}$	Note 1		249	306	mA
MAX24610 Total Current, All 1.8V Supply Pins	$I_{DD18}$	Note 1		467	575	mA
MAX24610 Total Current, All 3.3V Supply Pins	$I_{DD33}$	Note 1		333	408	mA
1.8V Supply Current Change from Enabling or Disabling APLL2	$\Delta I_{DD18APLL}$			50		mA
3.3V Supply Current Change from Enabling or Disabling APLL2	$\Delta I_{DD33APLL}$			75		mA
1.8V Supply Current Change from Enabling or Disabling the Input Block	$\Delta I_{DD18ICB}$			14		mA
1.8V Supply Current Change from Enabling or Disabling the DPLL	$\Delta I_{DD18DPLL}$			90		mA
1.8V Supply Current Change from Enabling or Disabling a CML Output, Standard Swing	$\Delta I_{DD18CML}$			22		mA
3.3V Supply Current Change from Enabling or Disabling a CML Output, Standard Swing	$\Delta I_{DD33CML}$			16		mA
1.8V Supply Current Change from Enabling or Disabling a CML Output, Narrow Swing	$\Delta I_{DD18CMLN}$			22		mA
3.3V Supply Current Change from Enabling or Disabling a CML Output, Narrow Swing	$\Delta I_{DD33CMLN}$			8		mA
VDDO18x Supply Current Change from Enabling or Disabling a Pair of Single-Ended Outputs	$\Delta I_{DD18CMOS}$			8		mA
VDDOx Supply Current Change from Enabling or Disabling a Pair of Single-Ended Outputs	$\Delta I_{DD33CMOS}$			6		mA

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP <sup>2</sup>	MAX	UNITS
1.8V Supply Current Change from Enabling or Disabling an Input Clock	$\Delta I_{DD18IN}$			6		mA
1.8V Supply Current Change from Enabling or Disabling the Crystal Oscillator	$\Delta I_{DD18DFS}$			4		mA

**Note 1:** Max  $I_{DD}$  measurements made with all blocks enabled, 750MHz signals on both inputs, and all outputs enabled as CML outputs driving 750MHz signals.

**Note 2:** Typical values measured at 1.80V and 3.30V supply voltages and 25°C ambient temperature.

**Table 8-3. Electrical Characteristics: Non-Clock CMOS/TTL Pins**

(1.8V Supplies: 1.8V  $\pm 5\%$ ; 3.3V Supplies: 3.3V  $\pm 5\%$ ,  $T_A = -40^\circ\text{C}$  to  $+85^\circ\text{C}$ )

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Input High Voltage	$V_{IH}$		2.0			V
Input Low Voltage	$V_{IL}$				0.8	V
Input Leakage	$I_{IL}$	Note 1	-10		10	$\mu\text{A}$
Input Leakage, Pins with Internal Pullup Resistor (50k $\Omega$ typ)	$I_{ILPU}$	Note 1	-85		10	$\mu\text{A}$
Input Leakage, Pins with Internal Pulldown Resistor (50k $\Omega$ typ)	$I_{ILPD}$	Note 1	-10		85	$\mu\text{A}$
Output Leakage (when High Impedance)	$I_{LO}$	Note 1	-10		10	$\mu\text{A}$
Output High Voltage	$V_{OH}$	$I_O = -4.0\text{mA}$	2.4			V
Output Low Voltage	$V_{OL}$	$I_O = 4.0\text{mA}$			0.4	V
Input Capacitance	$C_{IN}$			3		pF

**Note 1:**  $0\text{V} < V_{IN} < V_{DD33}$  for all other digital inputs.

**Table 8-4. Electrical Characteristics: Clock Inputs**

(1.8V Supplies: 1.8V  $\pm 5\%$ ; 3.3V Supplies: 3.3V  $\pm 5\%$ ,  $T_A = -40^\circ\text{C}$  to  $+85^\circ\text{C}$ )

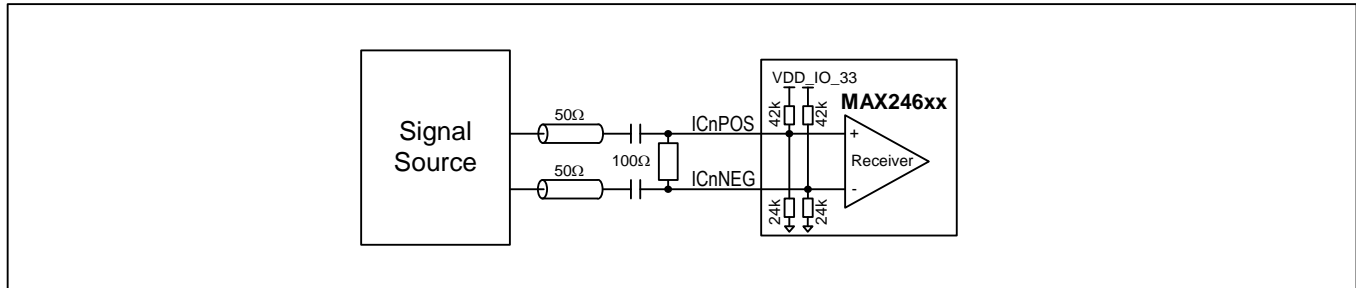
PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Input Voltage Tolerance (ICPOS or ICNEG, Single-Ended)	$V_{TOL}$	Note 1	0		$V_{DD33}$	V
Input Voltage Range, (ICPOS or ICNEG, Single-Ended)	$V_{IN}$	$ V_{ID}  = 100\text{mV}$	0		2.4	V
Input Bias Voltage	$V_{CMI}$	Note 2		1.2		V
Input Differential Voltage	$ V_{ID} $	Note 3	0.1		1.4	V
Input Frequency to Input block	$f_i$	Differential			750	MHz
Input Frequency to Input block	$f_i$	Single-Ended			160	MHz
Input Frequency to APLL Mux	$f_i$	Differential	9.72		750	MHz
Input Frequency to APLL Mux	$f_i$	Single-Ended	9.72		160	MHz
Minimum Input Clock High, Low Time	$t_H, t_L$			smaller of 3ns or $0.3 \times 1/f_i$		ns
Differential Input Capacitance	$C_{ID}$			1.5		pF

**Note 1:** The device can tolerate voltages as specified in  $V_{TOL}$  w.r.t. VSS on its ICxPOS and ICxNEG 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  $V_{IN}$ , are met. For single-ended signals, the input circuitry accepts signals that meet the  $V_{IH}$  and  $V_{IL}$  specifications in Table 8-3 above (but with  $V_{IH}$  max of  $V_{DD33}$ ).

**Note 2:** See internal resistors in Figure 8-1. Other common mode voltages can be set using external resistors.

**Note 3:**  $V_{ID} = V_{ICPOS} - V_{ICNEG}$

**Note 4:** The differential inputs can easily be interfaced to LVDS, LVPECL, and CML outputs on neighboring ICs using a few external passive components. See Figure 8-1 and App Note HFAN-1.0 for details.

**Figure 8-1. Recommended External Components for Interfacing to Differential Inputs**




**Table 8-5. Electrical Characteristics: CML Clock Outputs**

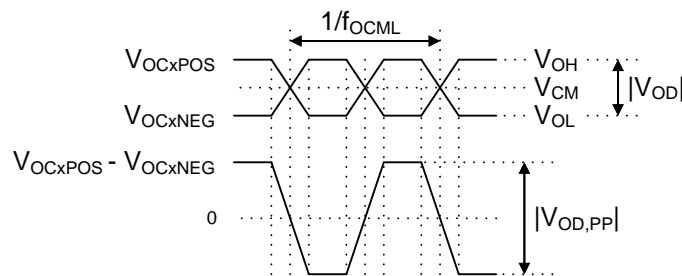
(1.8V Supplies: 1.8V  $\pm 5\%$ ; 3.3V Supplies: 3.3V  $\pm 5\%$ , VDDOx = 3.3V  $\pm 5\%$  (x=A|B|C|D); T<sub>A</sub> = -40°C to +85°C)

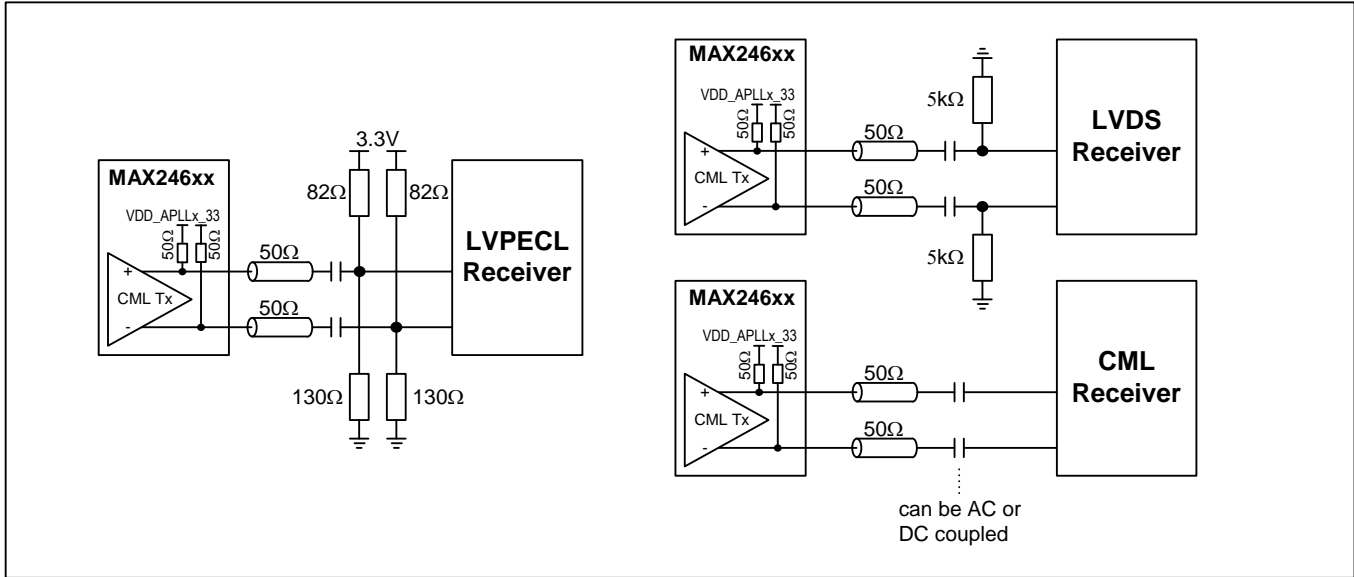
PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Output Frequency	f <sub>OCML</sub>				750	MHz
Output High Voltage (OCPOS or OCNEG, Singled-Ended)	V <sub>OH,S</sub>	Standard Swing (OCCR2.OCSEF=1), AC coupled to 50Ω termination		VDDOx – 0.2		V
Output Low Voltage (OCPOS or OCNEG, Singled-Ended)	V <sub>OL,S</sub>			VDDOx – 0.6		V
Output Common Mode Voltage	V <sub>CM,S</sub>			VDDOx – 0.4		V
Differential Output Voltage	V <sub>OD,S</sub>		320	400	500	mV
Differential Output Voltage Peak-to-Peak	V <sub>OD,S,PP</sub>		640	800	1000	mV <sub>P-P</sub>
Output High Voltage (OCPOS or OCNEG, Singled-Ended)	V <sub>OH,N</sub>	Narrow Swing (half the power) (OCCR2.OCSEF=2), AC coupled to 50Ω termination		VDDOx – 0.1		V
Output Low Voltage (OCPOS or OCNEG, Singled-Ended)	V <sub>OL,N</sub>			VDDOx – 0.3		V
Output Common Mode Voltage	V <sub>CM,N</sub>			VDDOx – 0.2		V
Differential Output Voltage	V <sub>OD,N</sub>		160	200	250	mV
Differential Output Voltage Peak-to-Peak	V <sub>OD,N,PP</sub>		320	400	500	mV <sub>P-P</sub>
Difference in Magnitude of Differential Voltage for Complementary States	V <sub>DOS</sub>				50	mV
Output Rise/Fall Time	t <sub>R</sub> , t <sub>F</sub>	20%-80%		150		ps
Output Duty Cycle		Notes 2	45	50	55	%
Output Duty Cycle		Notes 3	40		60	%
Output Impedance	R <sub>OUT</sub>	Single Ended, to VDDOx		50		Ω
Mismatch in a pair	ΔR <sub>OUT</sub>				10	%

**Note 1:** The differential CML outputs can easily be interfaced to LVDS, LVPECL, and CML outputs on neighboring ICs using a few external passive components. See [Figure 8-2](#) and [App Note HFAN-1.0](#) for details.

**Note 2:** For all HSDIV, MSDIV and OCDIV combinations other than those specified in Note 3.

**Note 3:** For the case when [APLLCR1.HSDIV](#) specifies a half divide and [OCCR1.MSDIV](#)=0 and [OCDIV](#)=0.



**Figure 8-2. Recommended External Components for Interfacing to CML Outputs**

**Table 8-6. Electrical Characteristics: CMOS and HSTL (Class I) Clock Outputs**

(1.8V Supplies: 1.8V  $\pm 5\%$ ; 3.3V Supplies: 3.3V  $\pm 5\%$ , VDDOx = 1.425V to 3.465V (x=A|B|C|D); T<sub>A</sub> = -40°C to +85°C)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Output Frequency	f <sub>OCML</sub>		<<1Hz <sup>1</sup>		160	MHz
Output High Voltage	V <sub>OH</sub>	Notes 3, 4	VDDOx -0.4		VDDOx	V
Output Low Voltage	V <sub>OL</sub>	Notes 3, 4	0		0.4	V
Output Rise/Fall Time, VDDOx=1.8V, <a href="#">OCCR2.DRIVE=4x</a>	t <sub>R</sub> , t <sub>F</sub>	2pF load		0.4		ns
Output Rise/Fall Time, VDDOx=1.8V, <a href="#">OCCR2.DRIVE=4x</a>	t <sub>R</sub> , t <sub>F</sub>	15pF load		1.2		ns
Output Rise/Fall Time, VDDOx=3.3V, <a href="#">OCCR2.DRIVE=1x</a>	t <sub>R</sub> , t <sub>F</sub>	2pF load		0.7		ns
Output Rise/Fall Time, VDDOx=3.3V, <a href="#">OCCR2.DRIVE=1x</a>	t <sub>R</sub> , t <sub>F</sub>	15pF load		2.2		ns
Output Duty-Cycle			45	50	55	%
Output Current When Output Disabled		<a href="#">OCCR2.OCSF=0</a>		10		μA

**Note 1:** Guaranteed by design.

**Note 2:** Measured with a series resistor of 33Ω and a 10pF load capacitance unless otherwise specified.

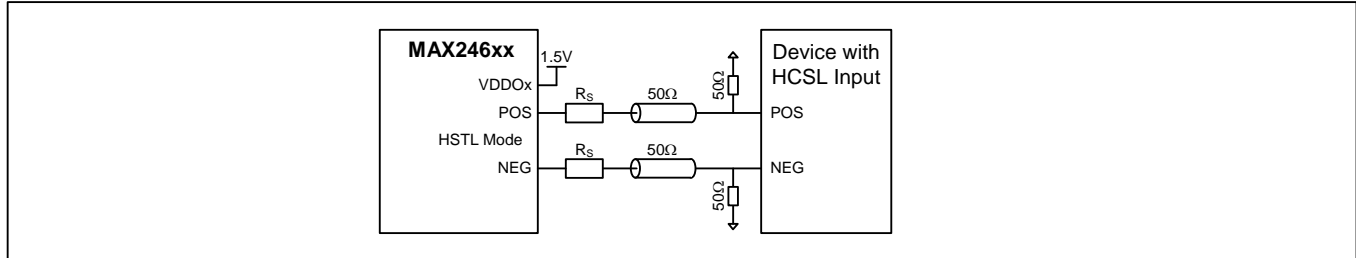
**Note 3:** For HSTL Class I, V<sub>OH</sub> and V<sub>OL</sub> apply for both unterminated loads and for symmetrically terminated loads, i.e. 50Ω to VDDOx/2.

**Note 4:** For VDDOx=3.3V and [OCCR2.DRIVE=1x](#), I<sub>O</sub>=4mA. For VDDOx=1.5V and [OCCR2.DRIVE=4x](#), I<sub>O</sub>=8mA..

## Interfacing to HCSL Components

Outputs in HSTL mode with  $V_{DDOx}=1.5V$  or  $V_{DDOx}=1.8V$  can provide an HCSL signal ( $V_{OH}$  typ. 0.75V) to a neighboring component when configured as shown in Figure 8-3 below. For  $V_{DDOx}=1.5V$  the value of  $R_s$  should be set to  $30\Omega$  and **OCCR2.DRIVE** should be set to 4x. For  $V_{DDOx}=1.8V$  the value of  $R_s$  should be set to  $20\Omega$  and **OCCR2.DRIVE** should be set to 2x.

Figure 8-3. Recommended Configuration for Interfacing to HCSL Components



**Table 8-7. Electrical Characteristics: Clock Output Timing**

(1.8V Supplies:  $1.8V \pm 5\%$ ; 3.3V Supplies:  $3.3V \pm 5\%$ ,  $T_A = -40^\circ C$  to  $+85^\circ C$ )

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
APLL VCO Frequency Range	$f_{VCO}$		3715		4180	MHz
APLL Phase-Frequency Detector Compare Frequency	$t_{PFD}$		9.72		102.4	MHz

**Table 8-8. Electrical Characteristics: Jitter Specifications**

(1.8V Supplies:  $1.8V \pm 5\%$ ; 3.3V Supplies:  $3.3V \pm 5\%$ ,  $T_A = -40^\circ C$  to  $+85^\circ C$ )

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Output Jitter, DPLL+APLL, 622.08MHz		Notes 1, 5		0.36	0.48	ps RMS
Output Jitter, APLL-Only, 622.08MHz		Notes 1, 6		0.19	0.35	ps RMS
Jitter Transfer Bandwidth, DPLL+APLL		Note 3	Programmable: 0.1 to 400			Hz
Jitter Transfer Bandwidth, APLL-Only		Note 4		400		kHz

**Note 1:** Jitter calculated from integrated phase noise from 12kHz to 20MHz.

**Note 2:** If DPLL is enabled and clocked from MCLKOSCP/N pins, the signal on MCLKOSCP/N has phase noise at 100kHz offset from the carrier  $\leq -150dBc/Hz$ .

**Note 3:** DPLL damping factor is also programmable. Other DPLL bandwidths also available. Contact the factory for details.

**Note 4:** APLL bandwidth and damping factor can be field configured over a limited range. Contact the factory for details.

**Note 5:** Tested with 51.2MHz MCLKOSC signal from production tester, 4096MHz APLL2 VCO frequency divided down to 204.8MHz DPLL master clock frequency. 77.76MHz DFS frequency to APLL1.

**Note 6:** Tested with 77.76MHz from production tester, 3732.48MHz VCO frequency.

**Table 8-9. Electrical Characteristics: Typical Output Jitter Performance, APLL Only**

APLL Locked to External 78.125MHz XO (Vectron VCC1-1540-78M12500), DPLL Disabled			
APLL1 Output Frequency	Output Jitter ps RMS	APLL2 Output Frequency	Output Jitter ps RMS
625MHz	0.18	APLL2 Disabled	
156.25MHz	0.23		
125MHz	0.27		
25MHz CMOS	0.34		
622.08MHz	0.28		
155.52MHz	0.35		
622.08MHz * 255/237	0.30		
155.52MHz * 255/237	0.36		
614.4MHz	0.29		
153.6MHz	0.33		
625MHz	0.19	622.08MHz	0.27
156.25MHz	0.24	155.52MHz	0.38
156.25MHz	0.23	156.25MHz * 66/64	0.38

**Table 8-10. Electrical Characteristics: Typical Output Jitter Performance, DPLL+APLL**

DPLL Locked to 25MHz Input on IC1, APLL1 Locked to DPLL			
98.304MHz XO (Vectron VCC1-1542-98M304) on MCLKOSCP/N to APLL2, 196.608MHz Master Clock from APLL2 to DPLL, 70MHz DFS frequency to APLL1.		20.48MHz Stratum 3 TCXO (Conner-Winfield MX602-20.48M) on MCLKOSCP/N to APLL2, 204.8MHz Master Clock from APLL2 to DPLL, 70MHz DFS frequency to APLL1.	
APLL1 Output Frequency	Output Jitter, ps RMS	APLL1 Output Frequency	Output Jitter, ps RMS
625MHz	0.33	625MHz	0.42
156.25MHz	0.39	156.25MHz	0.47
125MHz	0.37	125MHz	0.44
25MHz CMOS	0.45	25MHz CMOS	0.52
622.08MHz	0.32	622.08MHz	0.41
155.52MHz	0.39	155.52MHz	0.47
622.08MHz * 255/237	0.36	622.08MHz * 255/237	0.42
155.52MHz * 255/237	0.40	155.52MHz * 255/237	0.48
614.4MHz	0.33	614.4MHz	0.44
153.6MHz	0.38	153.6MHz	0.48

**Note:** All signals in Table 8-9 and Table 8-10 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 8-11. Electrical Characteristics: Typical Input-to-Output Clock Delay**

(1.8V Supplies: 1.8V  $\pm 5\%$ ; 3.3V Supplies: 3.3V  $\pm 5\%$ ,  $T_A = -40^\circ\text{C}$  to  $+85^\circ\text{C}$ )

MODE	DELAY, INPUT CLOCK EDGE TO OUTPUT CLOCK EDGE
DPLL+APLL Mode	$\pm 1$ UI of APLL Output Clock (Output of HSDIV) For example if APLL output clock is 625MHz, then delay is $\pm 1.6\text{ns}$ . Requires <a href="#">OFFSET</a> field set to -15 UI of the APLL output clock.  Delay can be tuned for all outputs traceable to the DPLL using the <a href="#">OFFSET</a> field. Delay for an individual output can be tuned using the <a href="#">OCCR3.PHADJ</a> field.
APLL-Only Mode	non-deterministic but constant as long as the APLL remains locked and alignment is not changed by the <a href="#">APLLCR1.DALIGN</a> and <a href="#">OCCR3.DALEN</a> bits.

**Table 8-12. Electrical Characteristics: Typical Output-to-Output Clock Delay**

(1.8V Supplies: 1.8V  $\pm 5\%$ ; 3.3V Supplies: 3.3V  $\pm 5\%$ ,  $T_A = -40^\circ\text{C}$  to  $+85^\circ\text{C}$ )

MODE	DELAY, OUTPUT CLOCK EDGE TO OUTPUT CLOCK EDGE
DPLL+APLL or APLL-Only	$< 100\text{ps}$  Requires use of <a href="#">APLLCR1.DALIGN</a> and <a href="#">OCCR3.DALEN</a> bits. See the register field descriptions for details.

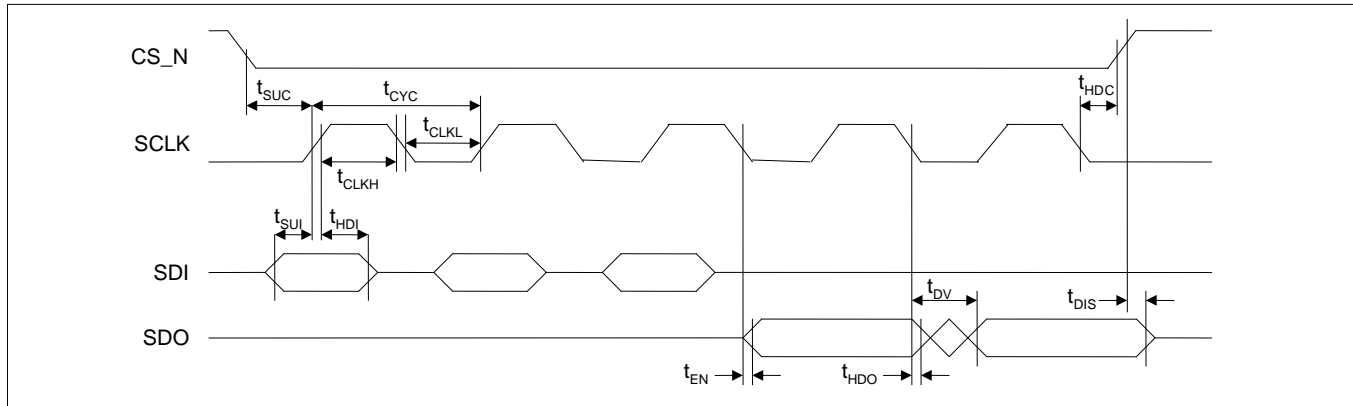
**Table 8-13. Electrical Characteristics: SPI Interface Timing**

(1.8V Supplies: 1.8V  $\pm 5\%$ ; 3.3V Supplies: 3.3V  $\pm 5\%$ ,  $T_A = -40^\circ\text{C}$  to  $+85^\circ\text{C}$ ) (See [Figure 8-4.](#))

PARAMETER (Note 1, 2)	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
SCLK Frequency	$f_{\text{BUS}}$				4	MHz
SCLK Cycle Time	$t_{\text{CYC}}$		250			ns
CS_N Setup to First SCLK Edge	$t_{\text{SUC}}$		125			ns
CS_N Hold Time After Last SCLK Edge	$t_{\text{HDC}}$		125			ns
SCLK High Time	$t_{\text{CLKH}}$		100			ns
SCLK Low Time	$t_{\text{CLKL}}$		100			ns
SDI Data Setup Time	$t_{\text{SUI}}$		30			ns
SDI Data Hold Time	$t_{\text{HDI}}$		40			ns
SDO Enable Time (High-Impedance to Output Active)	$t_{\text{EN}}$		0			ns
SDO Disable Time (Output Active to High-Impedance)	$t_{\text{DIS}}$				25	ns
SDO Data Valid Time	$t_{\text{DV}}$				100	ns
SDO Data Hold Time After Update SCLK Edge	$t_{\text{HDO}}$		5			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.

**Figure 8-4. SPI Interface Timing Diagram**

**Table 8-14. Electrical Characteristics: External EEPROM SPI Interface Timing**

(1.8V Supplies: 1.8V  $\pm 5\%$ ; 3.3V Supplies: 3.3V  $\pm 5\%$ ,  $T_A = -40^\circ\text{C}$  to  $+85^\circ\text{C}$ )

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
CS_N to ECS_N propagation delay	$t_{\text{PD\_CS}}$		6	9	15	ns
SCLK to ESCLK propagation delay	$t_{\text{PD\_SCLK}}$		6	9	15	ns
SDI to ESDI propagation delay	$t_{\text{PD\_SDI}}$		2	3.5	6	ns
ESDO to SDO propagation delay	$t_{\text{PD\_SDO}}$		2	3.5	6	ns

**Note 1:** All parameters in this table are guaranteed by design.

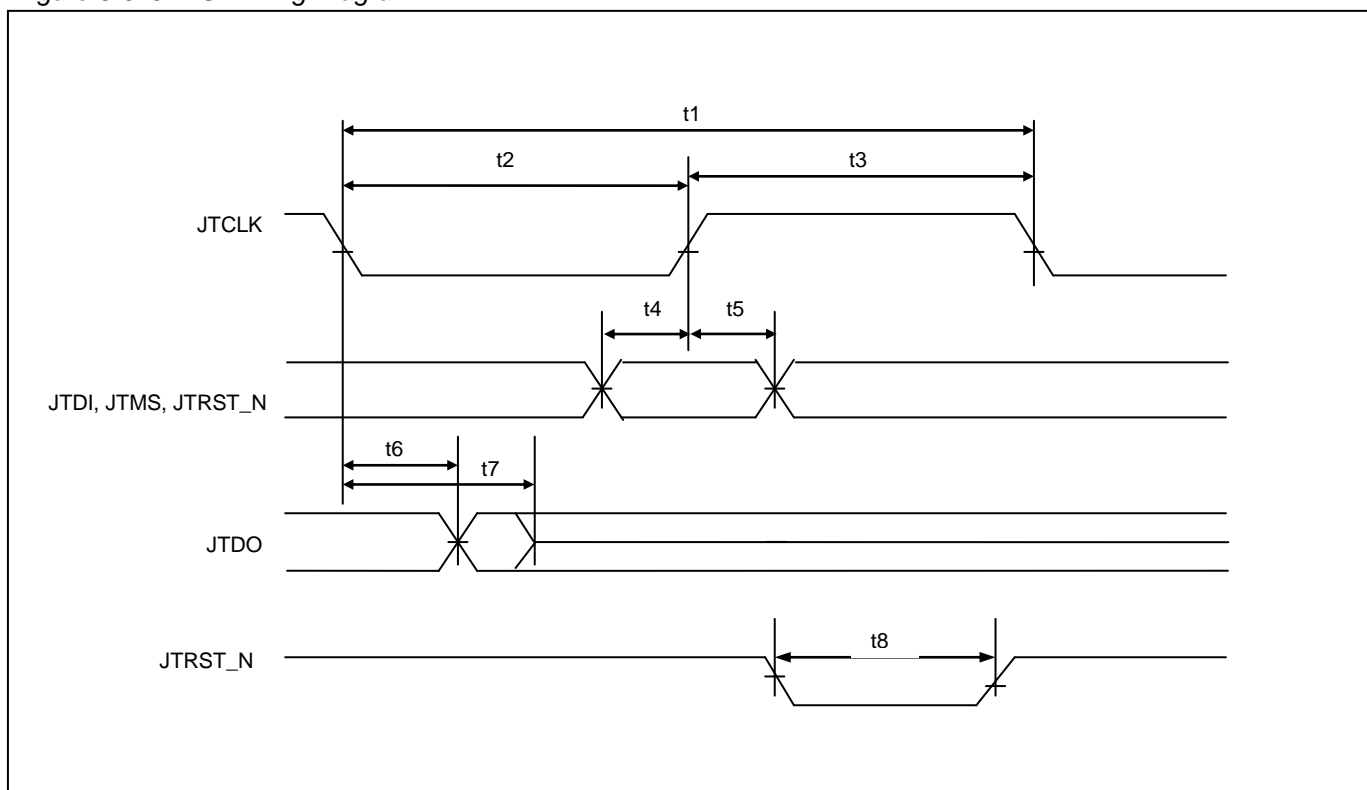
**Table 8-15. Electrical Characteristics: JTAG Interface Timing**

(1.8V Supplies: 1.8V  $\pm 5\%$ ; 3.3V Supplies: 3.3V  $\pm 5\%$ ,  $T_A = -40^\circ\text{C}$  to  $+85^\circ\text{C}$ ) (See [Figure 8-5.](#))

PARAMETER (Note 1)	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
JTCLK Clock Frequency	$f_{\text{JTAG}}$				15.625	MHz
JTCLK Clock Period	$t_1$		64			ns
JTCLK Clock High/Low Time	$t_2/t_3$	Note 2	32			ns
JTCLK to JTDI, JTMS Setup Time	$t_4$		16			ns
JTCLK to JTDI, JTMS Hold Time	$t_5$		16			ns
JTCLK to JTDO Delay	$t_6$		2		16	ns
JTCLK to JTDO High-Impedance Delay	$t_7$		2		16	ns
JTRST_N Width Low Time	$t_8$		100			ns

**Note 1:** All parameters in this table are guaranteed by design.

**Note 2:** Clock can be stopped high or low.

**Figure 8-5. JTAG Timing Diagram**


## 9. Pin Assignments

### 9.1 MAX24605 Pin Assignment

Table 9-1 below lists pin assignments sorted in alphabetical order by pin name. Figure 9-1 shows pin assignments arranged by pin number.















**Table 9-1. MAX24605 Pin Assignments Sorted by Signal Name**

PIN NAME	PIN NUMBERS	PIN NAME	PIN NUMBERS
CS_N	B7	SDO	A5
ECS_N	C3	TEST	C2
ESCLK	A4	VDD_18	D6
ESDI	B4	VDD_33	D7
ESDO	C4	VDD_APLL1_18	E6
GPIO1	A8	VDD_APLL1_33	E7
GPIO2	B8	VDD_APLL2_18	E4
GPIO3	A2	VDD_APLL2_33	E3
GPIO4	B2	VDD_DIG_18	D4, E5
IC1NEG	B9	VDD_OC_18	G3
IC1POS	A9	VDD_XO_18	G5
IC2NEG	B1	VDD_XO_33	G6
IC2POS	A1	VDDO18A	C9
JTCLK	B5	VDDO18B	H6
JTDI	C5	VDDO18C	H4
JTDO	B6	VDDO18D	C1
JTMS	C7	VDDOA	D8
JTRST_N	C6	VDDOB	G8
MCLKOSCP	A3	VDDOC	G2
MCLKOSCN	B3	VDDOD	D2
OC1NEG	E8	VSS_APLL1	F6, F7
OC1POS	E9	VSS_APLL2	F3, F4
OC2NEG	F8	VSS_DIG	D5, F5
OC2POS	F9	VSS_OC	G4
OC3NEG	H9	VSS_XO	G7
OC3POS	J9	VSSOA	D9
OC8NEG	H1	VSSOB	G9, J6
OC8POS	J1	VSSOC	G1, J4
OC10NEG	E2	VSSOD	D1
OC10POS	E1	VSUB	D3
RST_N	C8	XIN	H5
SCLK	A6	XOUT	J5
SDI	A7	N.C.	F1, F2, H2, H3, H7, H8, J2, J3, J7, J8



Figure 9-1. MAX24605 Pin Assignment Diagram

	1	2	3	4	5	6	7	8	9
A	IC2POS	GPIO3	MCLKOSCP	ESCLK	SDO	SCLK	SDI	GPIO1	IC1POS
B	IC2NEG	GPIO4	MCLKOSCN	ESDI	JTCLK	JTDO	CS_N	GPIO2	IC1NEG
C	VDDO18D	TEST	ECS_N	ESDO	JTDI	JTRST_N	JTMS	RST_N	VDDO18A
D	VSSOD	VDDOD	VSUB	VDD_DIG_18	VSS_DIG	VDD_18	VDD_33	VDDOA	VSSOA
E	OC10POS	OC10NEG	VDD_APLL2_33	VDD_APLL2_18	VDD_DIG_18	VDD_APLL1_18	VDD_APLL1_33	OC1NEG	OC1POS
F	N.C.	N.C.	VSS_APLL2	VSS_APLL2	VSS_DIG	VSS_APLL1	VSS_APLL1	OC2NEG	OC2POS
G	VSSOC	VDDOC	VDD_OC_18	VSS_OC	VDD_XO_18	VDD_XO_33	VSS_XO	VDDOB	VSSOB
H	OC8NEG	N.C.	N.C.	VDDO18C	XIN	VDDO18B	N.C.	N.C.	OC3NEG
J	OC8POS	N.C.	N.C.	VSSOC	XOUT	VSSOB	N.C.	N.C.	OC3POS

	Differential I/O (up to 750MHz)
	Low-Speed Digital I/O ( $\leq 10\text{MHz}$ )
	VDD 3.3V
	VDD 1.8V
	VSS
	APLL or XO VDD 3.3V
	APLL or XO VDD 1.8V
	APLL or XO VSS
	Output VDD 1.5-3.3V
	Output VDD 1.8V
	Output VSS
	Crystal I/O
	N.C. = No Connection. Lead is not connected to anything inside the device, or
	D.N.C. = Do Not Connect. Lead is internally connected. Do not connect anything to this lead.

## 9.2 MAX24610 Pin Assignment














Table 9-2 below lists pin assignments sorted in alphabetical order by pin name. Figure 9-2 shows pin assignments arranged by pin number.

**Table 9-2. MAX24610 Pin Assignments Sorted by Signal Name**

PIN NAME	PIN NUMBERS	PIN NAME	PIN NUMBERS
CS_N	B7	OC10NEG	E2
ECS_N	C3	OC10POS	E1
ESCLK	A4	RST_N	C8
ESDI	B4	SCLK	A6
ESDO	C4	SDI	A7
GPIO1	A8	SDO	A5
GPIO2	B8	TEST	C2
GPIO3	A2	VDD_18	D6
GPIO4	B2	VDD_33	D7
IC1NEG	B9	VDD_APLL1_18	E6
IC1POS	A9	VDD_APLL1_33	E7
IC2NEG	B1	VDD_APLL2_18	E4
IC2POS	A1	VDD_APLL2_33	E3
JTCLK	B5	VDD_DIG_18	D4, E5
JTDI	C5	VDD_OC_18	G3
JTDO	B6	VDD_XO_18	G5
JTMS	C7	VDD_XO_33	G6
JTRST_N	C6	VDDO18A	C9
MCLKOSCP	A3	VDDO18B	H6
MCLKOSCN	B3	VDDO18C	H4
OC1NEG	E8	VDDO18D	C1
OC1POS	E9	VDDOA	D8
OC2NEG	F8	VDDOB	G8
OC2POS	F9	VDDOC	G2
OC3NEG	H9	VDDOD	D2
OC3POS	J9	VSS_APLL1	F6, F7
OC4NEG	H8	VSS_APLL2	F3, F4
OC4POS	J8	VSS_DIG	D5, F5
OC5NEG	H7	VSS_OC	G4
OC5POS	J7	VSS_XO	G7
OC6NEG	H3	VSSOA	D9
OC6POS	J3	VSSOB	G9, J6
OC7NEG	H2	VSSOC	G1, J4
OC7POS	J2	VSSOD	D1
OC8NEG	H1	VSUB	D3
OC8POS	J1	XIN	H5
OC9NEG	F2	XOUT	J5
OC9POS	F1	N.C.	none

Figure 9-2. MAX24610 Pin Assignment Diagram

	1	2	3	4	5	6	7	8	9
A	IC2POS	GPIO3	MCLKOSCP	ESCLK	SDO	SCLK	SDI	GPIO1	IC1POS
B	IC2NEG	GPIO4	MCLKOSCN	ESDI	JTCLK	JTDO	CS_N	GPIO2	IC1NEG
C	VDDO18D	TEST	ECS_N	ESDO	JTDI	JTRST_N	JTMS	RST_N	VDDO18A
D	VSSOD	VDDOD	VSUB	VDD_DIG_18	VSS_DIG	VDD_18	VDD_33	VDDOA	VSSOA
E	OC10POS	OC10NEG	VDD_APLL2_33	VDD_APLL2_18	VDD_DIG_18	VDD_APLL1_18	VDD_APLL1_33	OC1NEG	OC1POS
F	OC9POS	OC9NEG	VSS_APLL2	VSS_APLL2	VSS_DIG	VSS_APLL1	VSS_APLL1	OC2NEG	OC2POS
G	VSSOC	VDDOC	VDD_OC_18	VSS_OC	VDD_XO_18	VDD_XO_33	VSS_XO	VDDOB	VSSOB
H	OC8NEG	OC7NEG	OC6NEG	VDDO18C	XIN	VDDO18B	OC5NEG	OC4NEG	OC3NEG
J	OC8POS	OC7POS	OC6POS	VSSOC	XOUT	VSSOB	OC5POS	OC4POS	OC3POS

	Differential I/O (up to 750MHz)
	Low-Speed Digital I/O ( $\leq 10\text{MHz}$ )
	VDD 3.3V
	VDD 1.8V
	VSS
	APLL or XO VDD 3.3V
	APLL or XO VDD 1.8V
	APLL or XO VSS
	Output VDD 1.5-3.3V
	Output VDD 1.8V
	Output VSS
	Crystal I/O
	N.C. = No Connection. Lead is not connected to anything inside the device, or D.N.C. = Do Not Connect. Lead is internally connected. Do not connect anything to this lead.

## 10. Package and Thermal Information

For the latest package outline information and land patterns contact Microsemi timing products technical support.

PACKAGE TYPE	PACKAGE CODE	OUTLINE NO.	LAND PATTERN
81 CSBGA	X8100M+4	<a href="#">21-0360</a>	See IPC-7351

### 10.1 Package Top Mark Format

Figure 10-1. Device Top Mark

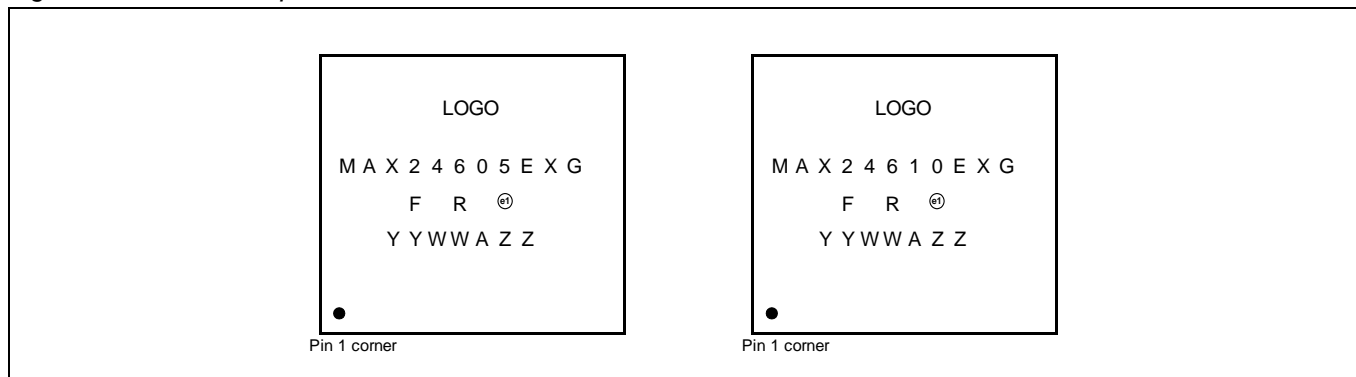


Table 10-1. Package Top Mark Legend

Line	Characters	Description
1	MAX24605EXG or MAX24610EXG	Part Number
2	F	Fab Code
2	R	Product Revision Code
2	e1	Denotes Pb-Free Package
3	YY	Last Two Digits of the Year of Encapsulation
3	WW	Work Week of Assembly
3	A	Assembly Location Code
3	ZZ	Assembly Lot Sequence Code

## 10.2 Thermal Specifications

**Table 10-2. CSBGA Package Thermal Properties**

PARAMETER	SYMBOL	CONDITIONS	VALUE	UNITS
Minimum Ambient Temperature	$T_A$		-40	°C
Maximum Ambient Temperature	$T_A$		85	°C
Minimum Junction Temperature	$T_J$		-40	°C
Maximum Junction Temperature	$T_J$		125	°C
Junction to Ambient Thermal Resistance (Note 1)	$\theta_{JA}$	still air,	24.9	°C/W
		1m/s airflow	22.7	
		2m/s airflow	21.9	
Junction to Board Thermal Resistance	$\theta_{JB}$		14.1	°C/W
Junction to Case Thermal Resistance	$\theta_{JC}$		4.1	°C/W
Junction to Top-Center Thermal Characterization Parameter	$\psi_{JT}$	still air,	0.3	°C/W
		1m/s airflow	0.4	
		2m/s airflow	0.4	

**Note 1:** Theta-JA ( $\theta_{JA}$ ) is the junction to ambient thermal resistance when the package is mounted on a six-layer JEDEC standard test board and dissipating maximum power.

If the maximum ambient temperature seen by the device in the application is greater than 70°C then care must be taken to keep the device's junction temperature below the 125°C max specification. In this case CML outputs should be configured for half-swing mode whenever possible, and air flow may be required, depending on which blocks in the device are enabled in the application. Microsemi offers the MAX24xxx Power and Thermal Calculator spreadsheet to calculate typical and worst-case power consumption and device junction temperature. Contact Microsemi applications support to request this spreadsheet.

## **11. Acronyms and Abbreviations**

APLL	analog phase locked loop
BITS	building integrated timing supply
CML	current mode logic
DFS	digital frequency synthesis
DPLL	digital phase locked loop
EEC	Ethernet equipment clock
GbE	gigabit Ethernet
I/O	input/output
LVDS	low-voltage differential signal
LVPECL	low-voltage positive emitter-coupled logic
MTIE	maximum time interval error
OCXO	oven controlled crystal oscillator
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
SDH	synchronous digital hierarchy
SEC	SDH equipment clock
SETS	synchronous equipment timing source
SONET	synchronous optical network
SSU	synchronization supply unit
STM	synchronous transport module
TDEV	time deviation
TCXO	temperature-compensated crystal oscillator
UI	unit interval
UI <sub>pp</sub> or UI <sub>p-p</sub>	unit interval, peak to peak
XO	crystal oscillator

## 12. Data Sheet Revision History

REVISION DATE	DESCRIPTION
15-Jan-2013	First preliminary data sheet version with new general-purpose clock multiplier / jitter attenuator scope.
17-Jan-2013	Added note to section 5.12 that several bits must be changed from their reset default values for proper operation of the device.
2013-02	<p>On page 1 and in section 3.3, reduced jitter numbers from “0.35 to 0.5ps and as low as 0.24ps” to “0.18 to 0.3ps RMS for an APLL-only integer multiply and 0.25 to 0.4ps RMS otherwise”</p> <p>In section 5.2.1 second paragraph and Table 8-8 changed typical APLL jitter transfer bandwidth from 200kHz to 400kHz.</p> <p>In Table 8-8, changed output jitter max from 0.6 to 0.48 ps RMS. Also added text to Note 5 to specify 204.8MHz DPLL master clock frequency and 77.76MHz DFS frequency.</p> <p>In Table 8-9 and Table 8-10 revised all numbers lower and specified XOs used for rev B jitter measurement.</p> <p>Edited the PLL1LSR.MCFAIL bit description to clarify behavior during continuing MCLK defects.</p> <p>Added 49.152MHz to Note 1 of Table 5-1 and added 98.304MHz to Table 5-2 and its Note 1.</p> <p>In section 5.2.2, section 5.3.3 and the MCR2.MCDIV, MCDNOM1 and MCINOM1 register descriptions changed the DPLL master clock range to 190MHz – 208.333MHz.</p> <p>Edited the DFSCR1.DFSFREQ register field description to say DFS frequency choices are limited when the DPLL’s nominal master clock frequency is different than 204.8MHz.</p> <p>Edited FREQ, HOFREQ, MCFREQ, HRDLIM and SOFTLIM register descriptions to include the factor <math>R = f_{MCLK} / 204.8\text{MHz}</math> in the equations to convert register values to ppm or ppb values.</p> <p>Changed the method for setting the MCAC field from table look-up to calculation to handle both the FMRES=0 and FMRES=1 cases.</p> <p>In the FMEAS register description, added text to specify the offset error if the DPLL’s nominal master clock frequency is not an integer multiple of 500Hz.</p> <p>In the ICCR3.FMONLEN description, clarified the existing options are for ICCR4.FMRES=0 and added the alternative options that are available when ICCR4.FMRES=1.</p>
2013-05	In section 10 replaced the land pattern hyperlink with the recommendation to see IPC-7351.
2013-08	<p>In Table 8-8, renamed spec “Output Jitter, 622.08MHz” to “Output Jitter, DPLL+APLL, 622.08MHz” and added new spec “Output Jitter, APLL-Only, 622.08MHz”.</p> <p>In Table 8-9 heading, corrected typo: 50MHz to 78.125MHz.</p> <p>In Table 8-10 heading, corrected typo: 98.306MHz to 98.304MHz.</p> <p>Changed the constant in the HRDLIM register description from 1.2272 to 1.226433036 and changed the constant in the SOFTLIM register description from 0.3141632 to 0.313966857 to more accurately represent the implementation.</p>
2014-01	In the JTRST_N pin description in Table 4-6 specified that JTRST_N should be held low during device power-up.
2014-08	<p>Changed title to Any-to-Any.</p> <p>Changed DPLLCR2 bit 4 from “set to 1” to unnamed (and therefore should be left set to factory default of 0).</p> <p>Edited section 5.5.1 to add “≥1MHz” to item 3.</p> <p>Edited the ICCR1 register description to say that &lt;1MHz lock frequencies should not be used with input fractional scaling.</p> <p>In Table 8-5 changed differential output voltage symbols (regular and peak-to-peak) to have</p>

REVISION DATE	DESCRIPTION
	abosolute value bars and added definition figure below the table.
2014-10	In <a href="#">Table 8-6</a> corrected typo: changed VCCOx to VDDOx. Added section <a href="#">10.1</a> to document package top mark.
2015-06	In section <a href="#">5.5.1</a> third bullet, specified that input frequency must be $\geq 1\text{MHz}$ and must divide by at least 4. Above <a href="#">Table 8-7</a> in the <i>Interfacing to HCSL Components</i> paragraph, added component values and settings for VDDOx=1.8V.
2016-09	Added content to the <a href="#">OFFSET</a> register description to describe the need to avoid OFFSET $\neq 0$ during DPLL state transition to Free-Run and to provide guidance on how to do that. Added a row for <a href="#">HOFREQ1-HOFREQ4</a> to the table in section <a href="#">6.1.4</a> because it was mistakenly left out.
2016-11	In <a href="#">Table 8-15</a> updated JTAG interface timing from 1MHz to 15.625MHz.





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](mailto:sales.support@microsemi.com)

©2016 Microsemi Corporation. All rights reserved. Microsemi and the Microsemi logo are trademarks of Microsemi Corporation. All other trademarks and service marks are the property of their respective owners.

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](http://www.microsemi.com).

Microsemi makes no warranty, representation, or guarantee regarding the information contained herein or the suitability of its products and services for any particular purpose, nor does Microsemi assume any liability whatsoever arising out of the application or use of any product or circuit. The products sold hereunder and any other products sold by Microsemi have been subject to limited testing and should not be used in conjunction with mission-critical equipment or applications. Any performance specifications are believed to be reliable but are not verified, and Buyer must conduct and complete all performance and other testing of the products, alone and together with, or installed in, any end-products. Buyer shall not rely on any data and performance specifications or parameters provided by Microsemi. It is the Buyer's responsibility to independently determine suitability of any products and to test and verify the same. The information provided by Microsemi hereunder is provided "as is, where is" and with all faults, and the entire risk associated with such information is entirely with the Buyer. Microsemi does not grant, explicitly or implicitly, to any party any patent rights, licenses, or any other IP rights, whether with regard to such information itself or anything described by such information. Information provided in this document is proprietary to Microsemi, and Microsemi reserves the right to make any changes to the information in this document or to any products and services at any time without notice.