Overview

The 8A34002 Synchronization Management Unit (SMU) provides tools to manage timing references, clock sources, and timing paths for IEEE 1588 and Synchronous Ethernet (SyncE) based clocks. The PLL channels can act independently as frequency synthesizers, jitter attenuators, Digitally Controlled Oscillators (DCO), or Digital Phase Lock Loops (DPLL).

Optional clock recovery filter/servo software is available under license from Renesas for use with the 8A34002. The filter/servo software is designed to suppress the affects of Packet Delay Variation (PDV) on packet based timing signals – it can be used with protocol stacks for IEEE 1588 or other packet-based timing protocols.

Typical Applications

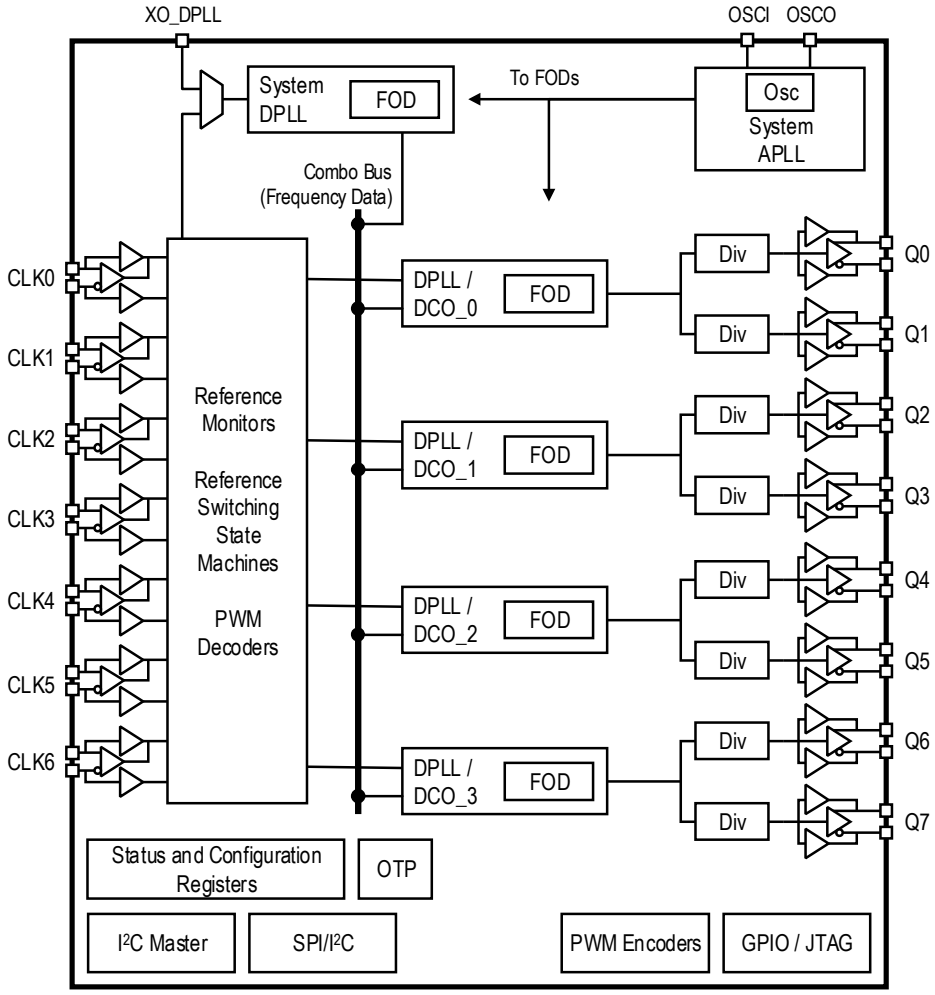
- Core and access IP switches / routers
- Synchronous Ethernet equipment
- Telecom Boundary Clocks (T-BCs) and Telecom Time Slave Clocks (T-TSCs) according to ITU-T G.8273.2
- 10Gb, 40Gb, and 100Gb Ethernet interfaces
- Central Office Timing Source and Distribution
- Wireless infrastructure for 4.5G and 5G network equipment

Features

- Four independent timing channels
  - Each can act as a frequency synthesizer, jitter attenuator, Digitally Controlled Oscillator (DCO), or Digital Phase Lock Loop (DPLL)
  - DPLLs generate telecom compliant clocks
    - Compliant with ITU-T G.8262 for Synchronous Ethernet
    - Compliant with legacy SONET/SDH and PDH requirements
  - DPLL Digital Loop Filters (DLFs) are programmable with cut off frequencies from 12µHz to 22kHz
  - DPLL/DCO channels share frequency information using the Combo Bus to simplify compliance with ITU-T G.8273.2
  - Switching between DPLL and DCO modes is hitless and dynamic
    - Automatic reference switching between DCO and DPLL modes to simplify support for an external phase/time input interface in a T-BC
  - Generates output frequencies that are independent of input frequencies via a Fractional Output Divider (FOD)
  - Each FOD supports output phase tuning with 1ps resolution
- 8 Differential / 16 LVCMOS outputs
  - Frequencies from 0.5Hz to 1GHz (250MHz for LVCMOS)
  - Jitter below 150fs RMS (10kHz to 20MHz)
  - LVCMOS, LVDS, LVPECL, HCSL, CML, SSTL, and HSTL output modes supported
  - Differential output swing is selectable: 400mV / 650mV / 800mV / 910mV
  - Independent output voltages of 3.3V, 2.5V, or 1.8V
  - LVCMOS additionally supports 1.5V or 1.2V
  - The clock phase of each output is individually programmable in 1ns to 2ns steps with a total range of ±180°
- 7 differential / 14 single-ended clock inputs
  - Support frequencies from 0.5Hz to 1GHz
  - Any input can be mapped to any or all of the timing channels
  - Redundant inputs frequency independent of each other
  - Any input can be designated as external frame/sync pulse of PPES (pulse per even second), 1 PPS (Pulse per Second), 5PPS, 10 PPS, 50Hz, 100Hz, 1 kHz, 2 kHz, 4kHz, and 8kHz associated with a selectable reference clock input
  - Per-input programmable phase offset of up to ±1.638µs in 1ps steps
- Reference monitors qualify/disqualify references depending on LOS, activity, frequency monitoring, and/or LOS input pins
  - Loss of Signal (LOS) input pins (via GPIOs) can be assigned to any input clock reference
- Automatic reference selection state machines select the active reference for each DPLL based on the reference monitors, priority tables, revertive / non-revertive, and other programmable settings
- System APLL operates from fundamental-mode crystal: 25MHz to 54MHz or from a crystal oscillator
- System DPLL accepts an XO, TCXO, or OCXO operating at virtually any frequency from 1MHz to 150MHz
- DPLLs can be configured as DCOs to synthesize Precision Time Protocol (PTP) / IEEE 1588 clocks
  - DCOs generate PTP based clocks with frequency resolution less than 1.11×10^{-16}
- DPLL Phase detectors can be used as Time-to-Digital Converters (TDC) with precision below 1ps
- Supports 1MHz I^2C or 50MHz SPI serial processor ports
- The device can configure itself automatically after reset via:
  - Internal customer definable One-Time Programmable memory with up to 16 different configurations
  - Standard external I^2C EPROM via separate I^2C Master Port
- 1149.1 JTAG Boundary Scan
- 10 × 10 mm 72-VFQFPN package

Block Diagram[1]

Figure 1. Block Diagram



Description

The 8A34002 is a Synchronization Management Unit (SMU) for packet based and physical layer based equipment synchronization. The 8A34002 is a highly integrated device that provides tools to manage timing references, clock sources, and timing paths for IEEE 1588 and Synchronous Ethernet (SyncE) based clocks. The PLL channels can act independently as frequency synthesizers, jitter attenuators, Digitally Controlled Oscillators (DCO), or Digital Phase Lock Loops (DPLL).

The 8A34002 supports multiple independent timing paths that can each be configured as a DPLL or as a DCO. Input-to-input, input-to-output, and output-to-output phase skew can all be precisely managed. The device outputs low-jitter clocks that can directly synchronize interfaces such as 100GBASE-R, 40GBASE-R, 10GBASE-R, 10GBASE-W, and lower-rate Ethernet interfaces; as well as SONET/SDH and PDH interfaces and IEEE 1588 Time Stamp Units (TSUs).

The internal System APLL must be supplied with a low phase noise reference clock with frequency between 25MHz and 54MHz. The output of the System APLL is used for clock synthesis by all of the Fractional Output Dividers (FODs) in the device. The System APLL reference can come from an external crystal oscillator connected to the OSCI pin or from an internal oscillator that uses a crystal connected between the OSCI and OSCO pins.

[1] This product is covered by one or more of the following patents: US 9,369,270, US 10,355,699, US 10,075,284, US 9,628,255, and US 9,479,182.

The System DPLL generates an internal system clock that is used by the reference monitors and other digital circuitry in the device. If the reference provided to the System APLL meets the stability and accuracy requirements of the intended application then the System DPLL can free run and a System DPLL reference is not required. Alternatively, the System DPLL can be locked to an external reference that meets the stability and accuracy requirements of the intended application. The System DPLL can accept a reference from the XO_DPLL pin or via the reference selection mux.

The frequency accuracy/stability of the internal system clock determines the frequency accuracy/stability of the DPLLs in Free-Run mode and in Holdover mode; and it affects the wander generation of the DPLLs in Locked and DCO modes. When provided with a suitably stable and accurate system clock, the DPLLs meet the frequency accuracy, pull-in, hold-in, pull-out, noise generation, noise tolerance, transient response, and holdover performance requirements of ITU-T G.8262 synchronous Ethernet Equipment Clock (EEC) options 1 and 2.

The 8A34002 accepts up to 7 differential reference inputs and up to 14 single-ended reference inputs that can operate at common GNSS, Ethernet, SONET/SDH, PDH frequencies, and any input frequency from 0.5Hz to 1GHz (250MHz in single-ended mode). The references are continually monitored for loss of signal and for frequency offset per user-programmed thresholds. All of the references are available to all the DPLLs. The active reference for each DPLL is determined by forced or automatic selection based on user-programmed priorities, locking allowances, reference monitors, revertive and non-revertive settings, and LOS inputs.

The 8A34002 can accept a clock reference and an associated frame pulse or sync signal as a pair. DPLLs can lock to the clock reference and align the sync and clock outputs with the paired sync/frame input. The device allows any of the reference inputs to be configured as sync inputs that can be associated with any of the other reference inputs. The input sync signals can have a frequency of 1 PPS (Pulse per Second), PPES (pulse per even second), 5PPS, 10 PPS, 50Hz, 100Hz, 1 kHz, 2 kHz, 4kHz, and 8 kHz. This feature enables any DPLL to phase align its frame sync and clock outputs with a sync input without the need to use a low bandwidth setting to lock directly to the sync input.

The DPLLs support four primary operating modes: Free-Run, Locked, Holdover, and DCO. In Free-Run mode the DPLLs synthesize clocks based on the system clock alone. In Locked mode the DPLLs filter reference clock jitter with the selected bandwidth. Also in Locked mode, the long-term output frequency accuracy is the same as the long term frequency accuracy of the selected input reference. In Holdover mode, the DPLL uses frequency data acquired while in Locked mode to generate accurate frequencies when input references are not available. In DCO mode, the DPLL control loop is opened and the DCO can be controlled by a PTP clock recovery servo running on an external processor to synthesize PTP clocks.

The DPLLs can be configured with a range of selectable filtering bandwidths. Bandwidths lower than 20mHz can be used to lock the DPLL directly to a 1 PPS reference. Bandwidths in the range of 0.05Hz to 0.1Hz can be used for G.8273.2. Bandwidths in the range of 0.1Hz to 10Hz can be used for G.8262/G.813, Telcordia GR-253-CORE S3, or SMC applications. Bandwidths above 10Hz can be used in jitter attenuation and rate conversion applications.

In Telecom Boundary Clock (T-BC) and Telecom Time Slave Clock (T-TSC) applications per ITU-T G.8275.2, two DPLLs can be used; one DPLL is configured as a DCO to synthesize PTP clocks and the other DPLL is configured as an EEC/SEC to generate physical layer clocks. Combo mode provides physical layer frequency support from the EEC/SEC to the PTP clock.

For applications per ITU-T G.8263, any DPLL can be configured as a DCO to synthesize packet-based clocks.

In Synchronous Equipment Timing Source (SETS) applications per ITU-T G.8264, any of the DPLLs can be configured as an EEC/SEC to output clocks for the T0 reference point and can be used to output clocks for the T4 reference point.

The 8A34002 generates up to 8 differential output clocks at any frequency from 0.5Hz to 1GHz. The differential outputs can support LVPECL, LVDS, HCSL, and CML. The device generates up to 16 single-ended clocks with frequencies from 0.5Hz to 250MHz. LVCMOS output supports 3.3V, 2.5V, 1.8V, 1.5V, or 1.2V. Each output stage can be independently configured.

Clocks generated by the 8A34002 have jitter below 150fs RMS (10kHz to 20MHz), and therefore are suitable for serial 100GBASE-R, 40GBASE-R, and lower rate interfaces.

All control and status registers are accessed through the I^2C / SPI slave microprocessor interface. The SPI interface mode supports high clock rates (up to 50MHz). For configuring the DPLLs, the I^2C master interface can automatically load a configuration from an external EEPROM after reset. The 8A34002 also has an internal customer definable One-Time Programmable memory with up to 16 different configurations.

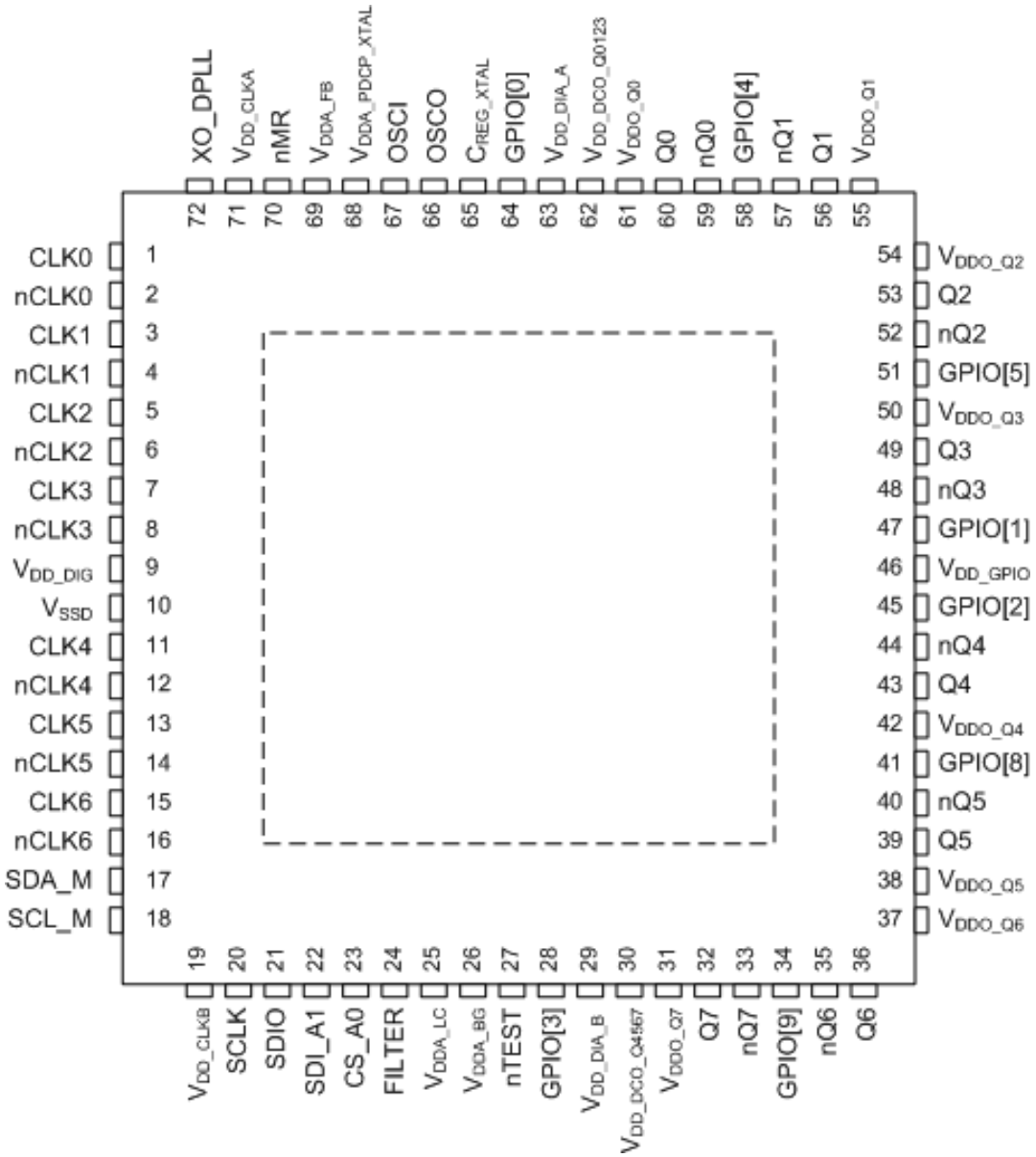
Contents

List of Figures

List of Tables

Pin Assignment

Figure 2. Pin Assignments for 10 × 10 mm 72-QFN Package[1]



[1] Note that indexed signals (e.g. GPIO[5]) are not necessarily numbered sequentially. Some indices may be skipped. This is to maintain software compatibility with other members of the family of devices.

Pin Description and Pin Characteristic Tables

Table 1. Pin Descriptions[a]

Number	Name	Type		Description
1	CLK0	Input	Pull-down	Positive input for differential input Clock 0 or single-ended input for Clock 0. For more information, see Input Stage.
2	nCLK0	Input	Pull-up	Negative input for differential input Clock 0 or single-ended input for Clock 8. For more information, see Input Stage.
3	CLK1	Input	Pull-down	Positive input for differential input Clock 1 or single-ended input for Clock 1. For more information, see Input Stage.
4	nCLK1	Input	Pull-up	Negative input for differential input Clock 1 or single-ended input for Clock 9. For more information, see Input Stage.
5	CLK2	Input	Pull-down	Positive input for differential input Clock 2 or single-ended input for Clock 2. For more information, see Input Stage.
6	nCLK2	Input	Pull-up	Negative input for differential input Clock 2 or single-ended input for Clock 10. For more information, see Input Stage.
7	CLK3	Input	Pull-down	Positive input for differential input Clock 3 or single-ended input for Clock 3. For more information, see Input Stage.
8	nCLK3	Input	Pull-up	Negative input for differential input Clock 3 or single-ended input for Clock 11. For more information, see Input Stage.
9	V_{DD_DIG}	Power		Power Supply for digital logic. 1.2V or 1.8V supported.
10	V_{SSD}	Power		Ground reference rail for digital logic (V_{DD_DIG})
11	CLK4	Input	Pull-down	Positive input for differential input Clock 4 or single-ended input for Clock 4. For more information, see Input Stage.
12	nCLK4	Input	Pull-up	Negative input for differential input Clock 4 or single-ended input for Clock 12. For more information, see Input Stage.
13	CLK5	Input	Pull-down	Positive input for differential input Clock 5 or single-ended input for Clock 5. For more information, see Input Stage.
14	nCLK5	Input	Pull-up	Negative input for differential input Clock 5 or single-ended input for Clock 13. For more information, see Input Stage.
15	CLK6	Input	Pull-down	Positive input for differential input Clock 6 or single-ended input for Clock 6. For more information, see Input Stage.
16	nCLK6	Input	Pull-up	Negative input for differential input Clock 6 or single-ended input for Clock 14. For more information, see Input Stage.
17	SDA_M	I/O	Pull-up	I^2C Bi-directional Data for I^2C Master Operation. For more information, see I2C Master. It can be connected to SDIO if desired and the connected port is configured for I^2C operation. External pull-up recommended.
18	SCL_M	I/O	Pull-up	I^2C Clock Output for I^2C Master Operation. For more information, see I2C Master. It can be connected to SCLK if desired and the connected port is configured for I^2C operation. External pull-up recommended.
19	V_{DD_CLKB}	Power		Power supply for input clock buffers and dividers for CLK4/nCLK4. CLK5/nCLK5, and CLK6/nCLK6. Supports 1.8V, 2.5V, or 3.3V as appropriate for the input clock swing. For more information, see Input Stage.
20	SCLK	Input	Pull-up	Serial port clock input. Used in both SPI and I^2C modes as the clock. An external pull-up is recommended in I^2C mode.

Table 1. Pin Descriptions[a] (Cont.)

Number	Name	Type		Description
21	SDIO	I/O	Pull-up	Serial port bi-directional data pin. Used as a bi-directional data pin in I^2C and 3-wire SPI modes. Used as Serial Data Output pin in 4-wire SPI mode. An external pull-up is recommended in I^2C mode.
22	SDI_A1	Input	Pull-up	Serial port input. Used as Serial Data In in 4-wire SPI mode and optionally as an Address Bit 1 select input in I^2C mode. Unused in 3-wire SPI mode.
23	CS_A0	Input	Pull-up	Serial port input. Used as a Chip Select input in SPI mode and optionally as an Address Bit 0 select input in I^2C mode.
24	FILTER	Analog		Reference capacitor for System Analog PLL Loop Filter. Requires a 2.2nF capacitor to ground.
25	V_{DDA_LC}	Power		Analog power supply voltage for System Analog PLL's LC Resonator, 3.3V or 2.5V supported.[c]
26	V_{DDA_BG}	Power		Analog power supply voltage for System Analog PLL's bandgap regulator, 3.3V or 2.5V supported.[c]
27	nTEST	Input	Pull-up	Test Mode enable pin. Must be high for normal operation
28	GPIO[3]	I/O	Pull-up[b]	General Purpose Input / Output 3. For more information, see General Purpose Input/Outputs (GPIOs).
29	$V_{DD\_DIA\_B}$	Power		Power Supply for FOD control logic for FOD blocks supporting output clocks Q4/nQ4, Q5/nQ5, Q6/nQ6 and Q7/nQ7. 1.8V supply required.
30	$V_{DD\_DCO\_Q4567}$	Power		Power supply for FOD blocks supporting output clocks Q4/nQ4, Q5/nQ5, Q6/nQ6 and Q7/nQ7. 1.8V supply required.
31	V_{DDO_Q7}	Power		Power supply for Q7/nQ7 output buffers. For voltages supported, see FOD Multiplexing and Output Stages.
32	Q7	Output		Q7 clock positive output. For more information, see FOD Multiplexing and Output Stages.
33	nQ7	Output		Q7 clock negative output. For more information, see FOD Multiplexing and Output Stages.
34	GPIO[9]	I/O	Pull-up[b]	General Purpose Input / Output 9. For more information, see General Purpose Input/Outputs (GPIOs).
35	nQ6	Output		Q6 clock negative output. For more information, see FOD Multiplexing and Output Stages.
36	Q6	Output		Q6 clock positive output. For more information, see FOD Multiplexing and Output Stages.
37	V_{DDO_Q6}	Power		Power supply for Q6/nQ6 output buffers. For voltages supported, see FOD Multiplexing and Output Stages.
38	V_{DDO_Q5}	Power		Power supply for Q5/nQ5 output buffers. For voltages supported, see FOD Multiplexing and Output Stages.
39	Q5	Output		Q5 clock positive output. For more information, see FOD Multiplexing and Output Stages.
40	nQ5	Output		Q5 clock negative output. For more information, see FOD Multiplexing and Output Stages.
41	GPIO[8]	I/O	Pull-up[b]	General Purpose Input / Output 8. For more information, see General Purpose Input/Outputs (GPIOs).

Table 1. Pin Descriptions[a] (Cont.)

Number	Name	Type		Description
42	V_{DDO_Q4}	Power		Power supply for Q4/nQ4 output buffers. For voltages supported, see FOD Multiplexing and Output Stages.
43	Q4	Output		Q4 clock positive output. For more information, see FOD Multiplexing and Output Stages.
44	nQ4	Output		Q4 clock negative output. For more information, see FOD Multiplexing and Output Stages.
45	GPIO[2]	I/O	Pull-up[b]	General Purpose Input / Output 2. For more information, see General Purpose Input/Outputs (GPIOs).
46	V_{DD_GPIO}	Power		Power Supply for all the digital pins, including GPIO pins and serial ports pins. 3.3V, 2.5V, 1.8V, or 1.5V supported.
47	GPIO[1]	I/O	Pull-up[b]	General Purpose Input / Output 1. For more information, see General Purpose Input/Outputs (GPIOs).
48	nQ3	Output		Q3 clock negative output. For more information, see FOD Multiplexing and Output Stages.
49	Q3	Output		Q3 clock positive output. For more information, see FOD Multiplexing and Output Stages.
50	V_{DDO_Q3}	Power		Power supply for Q3/nQ3 output buffers. For voltages supported, see FOD Multiplexing and Output Stages.
51	GPIO[5]	I/O	Pull-up[b]	General Purpose Input / Output 5. For more information, see General Purpose Input/Outputs (GPIOs).
52	nQ2	Output		Q2 clock negative output. For more information, see FOD Multiplexing and Output Stages.
53	Q2	Output		Q2 clock positive output. For more information, see FOD Multiplexing and Output Stages.
54	V_{DDO_Q2}	Power		Power supply for Q2/nQ2 output buffers. For voltages supported, see FOD Multiplexing and Output Stages.
55	V_{DDO_Q1}	Power		Power supply for Q1/nQ1 output buffers. For voltages supported, see FOD Multiplexing and Output Stages.
56	Q1	Output		Q1 clock positive output. For more information, see FOD Multiplexing and Output Stages.
57	nQ1	Output		Q1 clock negative output. For more information, see FOD Multiplexing and Output Stages.
58	GPIO[4]	I/O	Pull-up[b]	General Purpose Input / Output 4. For more information, see General Purpose Input/Outputs (GPIOs).
59	nQ0	Output		Q0 clock negative output. For more information, see FOD Multiplexing and Output Stages.
60	Q0	Output		Q0 clock positive output. For more information, see FOD Multiplexing and Output Stages.
61	V_{DDO_Q0}	Power		Power supply for Q0/nQ0 output buffers. For voltages supported, see FOD Multiplexing and Output Stages.
62	$V_{DD\_DCO\_Q0123}$	Power		Power supply for FOD blocks supporting output clocks Q0/nQ0, Q1/nQ1, Q2/nQ2, and Q3/nQ3. 1.8V supply required.

Table 1. Pin Descriptions[a] (Cont.)

Number	Name	Type		Description
63	$V_{DD\_DIA\_A}$	Power		Power Supply for FOD control logic for FOD blocks supporting output clocks Q0/nQ0, Q1/nQ1, Q2/nQ2, and Q3/nQ3. 1.8V supply required.
64	GPIO[0]	I/O	Pull-up[b]	General Purpose Input / Output 0. For more information, see General Purpose Input/Outputs (GPIOs).
65	C_{REG_XTAL}	Power		Filter capacitor for voltage regulator for oscillator circuit associated with OSCI / OSCO pins. Requires a 10µF filter capacitor to ground.
66	OSCO	Output		Crystal Output. This pin should be connected to a crystal. If an oscillator is connected to the OSCI pin, this pin should be left unconnected.
67	OSCI	Input		Crystal Input. Accepts a reference from a clock oscillator or a fundamental mode parallel-resonant crystal. For more information, see Table 35 and Table 36.
68	$V_{DDA\_PDCP\_XTAL}$	Power		Analog power supply voltage for System Analog PLL's phase detector and charge pump, as well as the crystal oscillator circuit. 2.5V or 3.3V operation supported.[c]
69	V_{DDA_FB}	Power		Analog power supply voltage for System Analog PLL's feedback divider, 1.8V required.
70	nMR	Input	Pull-up	Master Reset input. For more information, see Device Initial Configuration.
71	V_{DD_CLKA}	Power		Power supply for input clock buffers and dividers for CLK0/nCLK0. CLK1/nCLK1, CLK2/nCLK2, CLK3/nCLK3. Supports 1.8V, 2.5V, or 3.3V as appropriate for the input clock swing. For more information, see Input Stage.
72	XO_DPLL	Input		Single-ended crystal oscillator input for System Digital PLL. For more information, see Crystal Oscillator Input (XO_DPLL).
ePAD	V_{SS}	Power		Device ePAD must be connected to Ground.

[a] *Pullup* and *Pulldown* refer to internal input resistors. See Table 2, *Pin Characteristics,* for typical values.

[b] GPIO pins may be configured via EEPROM and/or OTP with a pull-up or pull-down. Pull-up is the default configuration.

[c] $V_{DDA\_PDCP\_XTAL}$, V_{DDA_LC} and V_{DDA_BG} may be driven with either 2.5V or 3.3V, however all must use the same voltage level. Register programming is required to configure the device for either 2.5V or 3.3V operation. For more information, see the *8A3xxxx Family Programming Guide*.

Table 2. Pin Characteristics

Symbol	Parameter		Test Conditions	Minimum	Typical	Maximum	Units
C_{IN}	Input Capacitance	OSCI, OSCO			9		pF
		XO_DPLL			1		
		All Other pins			2		
R_{PULLUP}	Input Pullup Resistor	nCLK[6:0]			50		kΩ
$R_{PULLDOWN}$	Input Pulldown Resistor	CLK[6:0]			50		kΩ
C_{PD}	Power Dissipation Capacitance (per output pair)	LVCMOS	V_{DDO_Qx}[a] = 3.465V		9		pF
		LVCMOS	V_{DDO_Qx} = 2.625V		8.8		
		LVCMOS	V_{DDO_Qx} = 1.89V		8.8		
		LVCMOS	V_{DDO_Qx} = 1.575V		9.2		
		LVCMOS	V_{DDO_Qx} = 1.26V		8.7		
		Differential	V_{DDO_Qx} = 3.465V		1.4		
			V_{DDO_Qx} = 2.625V		3.5		
			V_{DDO_Qx} = 1.89V		5		
R_{OUT}[b]	Output Impedance	GPIO[9:8,5:0]	V_{DD_GPIO} = 3.3V		27		Ω
			V_{DD_GPIO} = 2.5V		30		
			V_{DD_GPIO} = 1.8V		38		
			V_{DD_GPIO} = 1.5V		53		
		SCL_M, SDA_M, SDIO	V_{DD_GPIO} = 3.3V		62		Ω
			V_{DD_GPIO} = 2.5V		65		
			V_{DD_GPIO} = 1.8V		75		
			V_{DD_GPIO} = 1.5V		91		

[a] V_{DDO_Qx} denotes: V_{DDO_Q0}, V_{DDO_Q1}, V_{DDO_Q2}, V_{DDO_Q3}, V_{DDO_Q4}, V_{DDO_Q5}, V_{DDO_Q6} or V_{DDO_Q7}

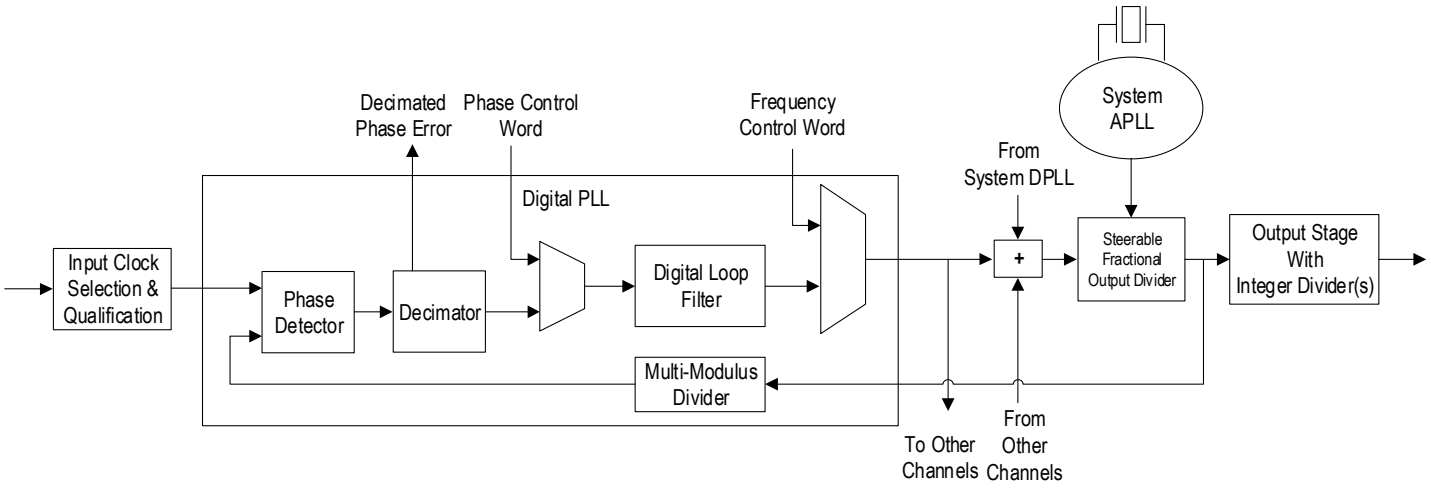
[b] Output impedance values for the Qx / nQx outputs are provided in Table 34.

Overview of the 8A3xxxx Family

The 8A3xxxx family of devices have multiple channels that can operate independently from each other, or in combination with each other (Combo Mode). All devices share a common channel architecture (see Figure 3) with different functional blocks within the channel being available for use in different members of the family. In addition, there are other peripheral blocks that may only be available in specific family members. The number of channels and of certain peripheral blocks (such as extra serial ports) will also vary from one device in the family to another. Across all members of the family, numbering of the functional and peripheral blocks and their associated register locations are kept consistent to enhance software compatibility and portability between members of the family.

The remaining sub-sections of this Functional Description section will describe functions within the clocking channel and peripheral functions that are available in the 8A34002.

Figure 3. Single PLL Channel



Functional Description

This section describes the operational modes and associated functional blocks of the 8A34002. In addition, there are several other areas of the document that describe specific functions or details that would overly burden this document. Table 3 shows related documents.

Table 3. Related Documentation

Document Title	Document Description
8A34002 Datasheet (This document)	Contains a functional overview of the device and hardware-design related details including pinouts, AC and DC specifications, and applications information related to power filtering and terminations.
8A34002-<dash code> Datasheet Addendum	Indicates pre-programmed power-up / reset configurations of this specific “dash code” part number.
8A3xxxx Family Programming Guide (v4.8)	Contains detailed register descriptions and address maps for all members of the family of devices. Please ensure to use the version indicated here for this product. The functionality described in this datasheet assumes that the device is running the update revision referred to here or a later one. For individual updates to determine differences between update revisions, see Release Note documents. Note that the device may not ship from the factory with the indicated update revision included in the device. If this is the case, the indicated revision may need to be loaded from an external EEPROM or over the serial port at each device reset.

Basic Functional Blocks

Crystal Input (OSCI / OSCO)

The 8A34002 requires a 25MHz–54MHz crystal input on the OSCI/OSCO pins at all times. This input is used to drive the System APLL, which in turn is the source for all internal clocks. For more information, see Table 35 and Crystal Recommendation.

Alternatively, the crystal input can be overdriven by a crystal oscillator. For more information, see Overdriving the XTAL Interface.

Frequency Representation

The format for representing a frequency in the registers of the 8A34002 is:

$$f = \frac{M}{N} \text{ where M is a 48-bit integer and N is a16-bit integer}$$

The $\frac{M}{N}$ notation allows non-integer frequencies to be precisely represented as fractions.

For example, the Optical Transport Network OTN OTU2e rate:

$$f = 156,250,000 \times \frac{66}{64} \times \frac{255}{237} \text{Hz}$$

Then:

$$\frac{M}{N} = 156,250,000 \times \frac{66}{64} \times \frac{255}{237} \text{Hz}$$

$$\frac{M}{N} = (2^4 \times 5^{10}) \times \frac{2^1 \times 3^1 \times 11^1}{2^6} \times \frac{3^1 \times 5^1 \times 17^1}{3^1 \times 79^1} \text{Hz}$$ Express terms as the products of prime factors

$$\frac{M}{N} = \frac{3^1 \times 5^{11} \times 11^1 \times 17^1}{2^1 \times 79^1} \text{Hz}$$ Simplify

$$\frac{M}{N} = \frac{27,392,578,125}{158} \text{Hz}$$ Lowest terms

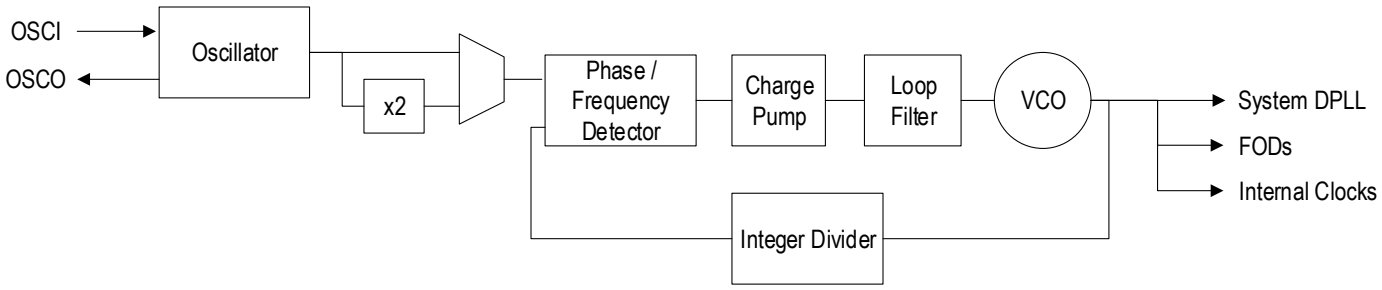
System APLL

The System APLL is managed with bit fields in the SYS_APLL module. See Module: SYS_APLL in the *8A3xxxx Family Programming Guide*.

The System APLL is shown in Figure 4. This consists of a simple analog PLL circuit that takes a reference crystal input and multiplies it up to a frequency in the 13.4-13.9GHz range. That high-speed signal is then used to drive the Fractional Output Divider (FOD) circuits as described in FOD Multiplexing and Output Stages. This combination of the System APLL and the FOD logic results in excellent phase noise performance and a substantial amount of flexibility in frequency and phase for the 8A34002. See the SYS_APLL.SYS_APLL_CTRL bit field.

One user programming option involves selecting whether the crystal reference frequency is to be used directly or run through an internal frequency doubler circuit first. An additional user programming option is to select the feedback divider value from the set of integers between 108 and 556. Between these two settings, the user should select a System APLL operating frequency that is within the above-stated tuning range. See the SYS_APLL.SYS_APLL_CTRL bit field.

Figure 4. System Analog PLL Channel



During a device reset, the System APLL is configured by loading the appropriate control register fields from the internal One-Time Programmable memory or an external serial EEPROM, whichever is enabled and has valid contents. After the reset sequence has completed, the System APLL can be re-configured manually over the serial port at any time.

The System APLL is considered locked when the Loop Filter control voltage is within specified limits for the configuration selected. The 8A34002 automatically calculates these limits based on other parameters specified in the device configuration. Specific user input to set locking limits is not required. A System APLL Loss-Of-Lock alarm is generated internally. This can be read from internal status registers and/or used to drive a GPIO status signal as described in GPIO Modes. See the STATUS.SYS_APLL_STATUS bit field.

Input Stage

The input stage is managed with bit fields in the INPUT_n modules where n ranges from 0 to 15. See Module: INPUT_0 in the *8A3xxxx Family Programming Guide*.

The 8A34002 contains multiple input stages. An input stage can be configured as one differential or dual single-ended inputs. Some of the input stages can also be configured to support one differential plus one single-ended clock. For information on how to connect various input types to the 8A34002, see Table 4 and Applications Information. See the INPUT_0.IN_MODE bit field.

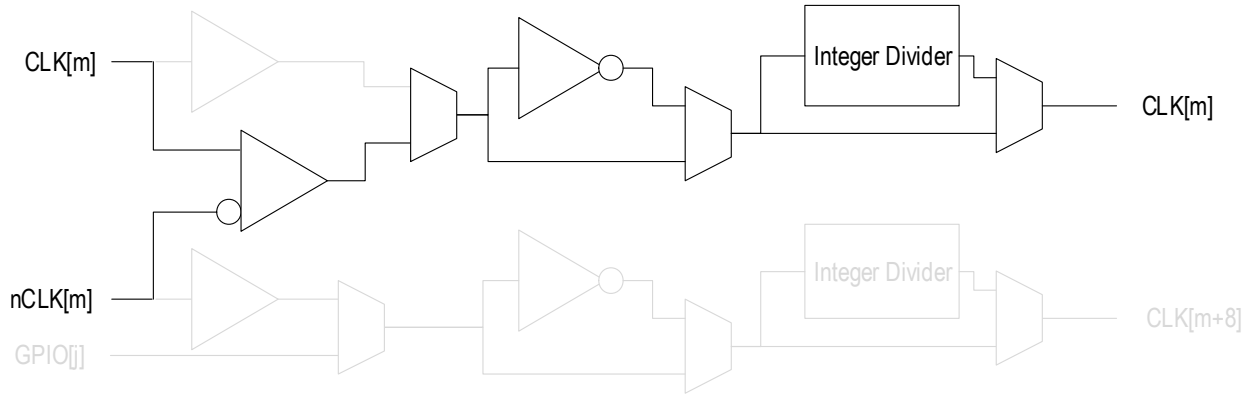
Table 4. Input Stage Setting

Input Protocol	Driver V_{DD} Level	Settings to Use		
		V_{DD_CLKx}[a] Voltage		
		3.3V	2.5V	1.8V
PECL	3.3V	Differential + NMOS		
PECL	2.5V	Differential + PMOS		
LVDS	N/A	Differential + NMOS		
HCSL	N/A	Differential + PMOS		
CML	3.3V	Differential + NMOS		
CML	2.5V	Differential + NMOS		
CML	1.8V	Differential + NMOS		
CMOS	3.3V	Single-ended		
CMOS	2.5V			
CMOS	1.8V			

[a] V_{DD_CLKx} refers to either V_{DD_CLKA} or V_{DD_CLKB} as appropriate for the input being programmed.

When programmed as differential only, as shown in Figure 5, the internal signal will be referred to by the index number of the input pins (e.g., CLK0 is used to refer to the differential input pair CLK0/nCLK0). It is also necessary to select the appropriate mode, PMOS or NMOS so the input buffer will work best with the incoming signal's voltage swing. See the INPUT_0.IN_MODE bit field.

Figure 5. Input Stage Configured as Differential Only



The 8A34002 supports input frequencies up to 1GHz for differential inputs. If the input reference clock frequency is higher than 150MHz, then it must be divided down to the internal frequency (less than or equal to 150MHz) used by the DPLL. An integer divider with a range between 2 to 65536 is provided to divide the signal down to less than or equal to 150MHz. For input reference clock frequencies less than 150MHz, the internal divider can be bypassed. See the INPUT_0.IN_MODE bit field.

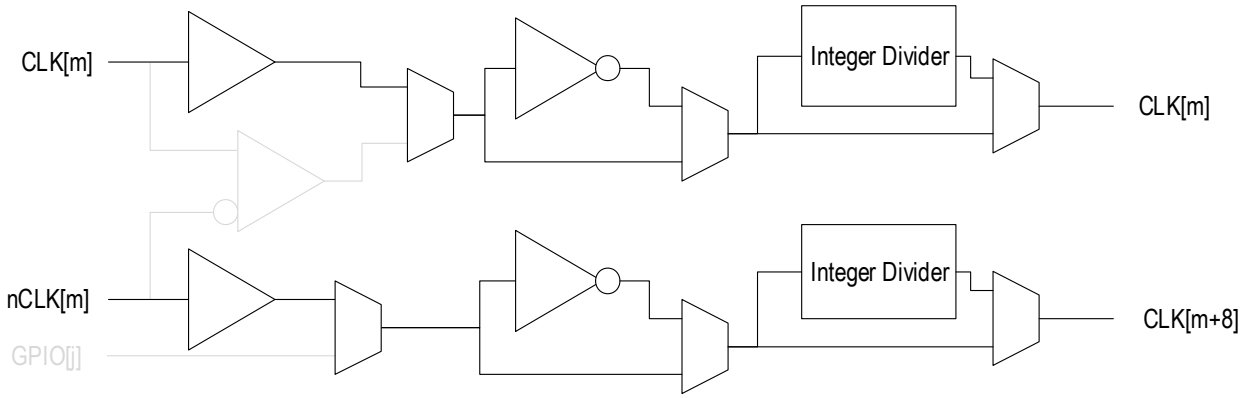
The 8A34002 has the option to lock to the rising or falling edge of the input clock signal by selecting the inverted input path to the divider.

When programmed as dual single-ended as shown in Figure 6, two independent inputs are provided to the 8A34002. The input clock originating from the positive input will be referred to by using the same index number (e.g., CLK0 is used to refer to the signal originating from CLK0). The signal originating from the negative input will be referred to by using the index + 8 (e.g., CLK8 is used to refer to the signal originating from nCLK0). Note that this numbering scheme remains the same on all 8A3xxxx family members, regardless of the number of actual input pins. This is to simplify software portability between family members. PMOS versus NMOS mode does not have any effect for single-ended inputs. See the INPUT_0.IN_MODE bit field.

The 8A34002 supports input frequencies up to 250MHz for single-ended inputs. If the input reference clock frequency is higher than 150MHz, then it needs to be divided down to the internal frequency (less or equal to 150MHz) used by the DPLL with the dividers shown in each path. For input reference clock frequencies less than 150MHz, the internal divider can be bypassed. See the INPUT_0.IN_MODE bit field.

The 8A34002 has the option to lock to the rising or falling edge of the input clock signal for either path independently. See the INPUT_0.IN_MODE bit field.

Figure 6. Input Stage Configured as Dual Single-Ended



When programmed as differential plus one single-ended as shown in Figure 7, two independent inputs are provided to the 8A34002. This mode can only be used with the GPIOs and input stages shown in the following table. See the INPUT_0.IN_MODE bit field.

Table 5. Input Stages Using GPIOs as Reference Clock Inputs

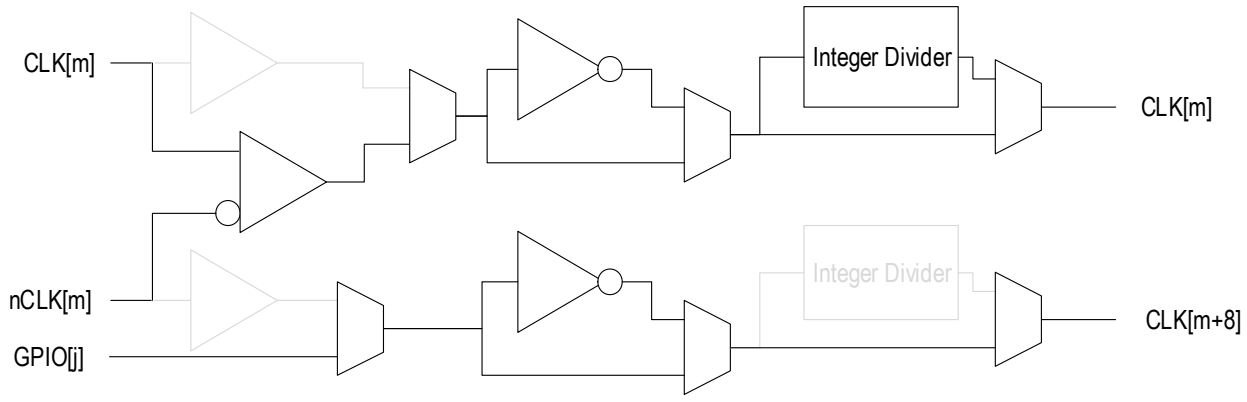
Differential Clock Input	Mapped to Internal Clock	GPIO Input	Mapped to Internal Clock
CLK0 / nCLK0	CLK[0]	GPIO[0]	CLK[8]
CLK1 / nCLK1	CLK[1]	-	CLK[9]
CLK2 / nCLK2	CLK[2]	-	CLK[10]
CLK3 / nCLK3	CLK[3]	-	CLK[11]
CLK4 / nCLK4	CLK[4]	-	CLK[12]
CLK5 / nCLK5	CLK[5]	-	CLK[13]
CLK6 / nCLK6	CLK[6]	-	CLK[14]
-	-	GPIO[3]	CLK[15]

Input stages not shown in this table can be used in the differential only mode or dual single-ended mode only. The differential input clock originating from the positive input will be referred to the same way as in differential only mode. The signal originating from the GPIO input will be referred to by using the index shown in the table. Note that this numbering scheme remains the same on all 8A3xxxx family members, regardless of the number of actual input pins. This is to simplify software portability between family members. PMOS versus NMOS mode does not have any effect for GPIO inputs. See the INPUT_0.IN_MODE bit field.

The 8A34002 supports input frequencies up to 150MHz for GPIO inputs, so no division is necessary. If the input reference clock frequency from the differential input path is higher than 150MHz, then it needs to be divided down to the internal frequency (less or equal to 150MHz) used by the DPLL with the divider shown in its path. For input reference clock frequencies less than 150MHz, the internal divider may be bypassed. See the INPUT_0.IN_DIV bit field.

The 8A34002 has the option to lock to the rising or falling edge of the input clock signal for either path individually. See the INPUT_0.IN_MODE bit field.

Figure 7. Input Stage Configured as Differential Plus Single Single-Ended



In addition to the above, there are a number of other configuration bits that can be used for the input stage.

- Unused inputs can be disabled. This allows a small amount of power saving and eliminates a source of on-die noise.
- Any input can be used either as a sync or frame pulse associated with an input clock (for more information, see Frame Pulse Operation and Sync Pulse Operation).
- The frequency of each input needs to be known by the 8A34002 and so must be programmed in the registers for each active input stage. See the INPUT_0.IN_FREQ bit field.

Reference Monitoring

The reference monitors are managed with bit fields in the REF_MON_n and GPIO_n modules where n ranges from 0 to 15 and the STATUS and ALERT_CFG modules. See Module: REF_MON_0, Module: GPIO_0, Module: STATUS and Module: ALERT_CFG in the *8A3xxxx Family Programming Guide*.

The quality of all input clocks is always monitored for:

- LOS (loss of signal)
- Activity
- Frequency

All input clocks are monitored all the time, including the active reference to ensure that it is still a valid reference. If any monitor detects a failure of the input clock, it will generate an internal alarm. An input clock with an alarm condition is not used for synchronization unless configured to allow it to be considered qualified in spite of the alarm.

For information on how these internal alarms can be signaled and monitored by outside resources, see Alarm Output Operation.

Loss of Signal (LOS) Monitoring

Each input clock is monitored for loss of signal (LOS). The LOS reference monitor supports normal clock operation and gapped clock operation. In normal operation, the user can specify whether the alarm condition should be tight to the expected clock period or loose. Tight monitoring will give minimum response time for loss of the input clock, but may result in false alarms due to normal clock jitter or wander. The loose threshold will take longer to detect an alarm condition but is unlikely to give false alarms. For clocks greater than 500kHz, both loose and tight specifications check for the clock edge being outside ±20nsec of the expected position to declare an alarm. For clocks less than or equal to 500kHz, loose threshold is set at ±25% of the nominal edge position and tight is set to ±1%. See the REF_MON_0.IN_MON_LOS_TOLERANCE, REF_MON_0.IN_MON_LOS_CFG bit fields.

In gapped clock operation, LOS is declared if the clock reference misses consecutive clock cycles. It is cleared once an active clock edge is detected. The number of consecutive clocks that are missed to declare LOS is programmable according to Table 6. A setting of 01 is equivalent to a normal clock monitor. See the REF_MON_0.IN_MON_LOS_CFG bit field.

Table 6. Gapped Clock LOS Settings

LOS_GAP[2:1]	Number of Consecutive Clocks Missed to Declare LOS
00	Gapped Clock Monitoring Disabled (default)
01	1
10	2
11	5

There is a status register for LOS. LOS failure alarm will be set as described above. What actions are taken in the event of an alarm can be configured via registers. The LOS failure can cause a specific alarm on a GPIO and/or be used as one input to an Alert (aggregated alarm) output via GPIO if so configured. See the REF_MON_0.IN_MON_CFG, STATUS.IN0_MON_STATUS, ALERT_CFG.IN1_0_MON_ALERT_MASK, GPIO_0.GPIO_LOS_INDICATOR, and GPIO_0.GPIO_CTRL bit fields.

Activity

All input reference clocks higher than 1kHz can be monitored for activity. Activity monitoring can quickly determine if a clock is within the frequency limits shown in Table 7. The method used by this monitor is not as precise as the Frequency Offset Monitor, but results are available much more quickly. See the REF_MON_0.IN_MON_ACT_CFG bit field.

Table 7. Activity Limit

ACT_LIM[2:0]	Range
000	±1000ppm
001	±260ppm
010	±130ppm
011	±83ppm
100	±65ppm
101	±52ppm
110	±18ppm
111	±12ppm

An activity failure alarm will be set if the input frequency has drifted outside the range set by the programmable range for longer than the period programmed for the activity disqualification timer. What actions are taken in the event of an alarm can be configured via registers. The Activity alarm can be used as one input to an Alert (aggregated alarm) output via GPIO if so configured. See the STATUS.IN0_MON_STATUS, REF_MON_0.IN_MON_CFG, ALERT_CFG.IN1_0_MON_ALERT_MASK and GPIO_0.GPIO_CTRL bit fields.

Timer

There is a timer associated with the activity qualification and disqualification of each input reference.

After an activity or LOS alarm is detected, then the timer starts. If the Activity or LOS alarm remains active for the full duration of the timer, then the reference disqualification alarm will be set to high. Register bits can be used to configure whether or not either the alarm is allowed to affect the disqualification decision or not. The disqualification timer can be selected according to Table 8. See the STATUS.IN0_MON_STATUS, REF_MON_0.IN_MON_ACT_CFG, and ALERT_CFG.IN1_0_MON_ALERT_MASK bit fields.

Table 8. Disqualification Timer

DSQUAL_TIMER[4:3]	Description
00	2.5s (default)
01	1.25ms
10	25ms
11	50ms

After a reference is disqualified, once it returns (all alarms now clear), then a qualification timer is started. If the alarms remain cleared for the full duration selected, then the input is qualified for use again. Qualification timer settings are shown in Table 9. See the REF_MON_0.IN_MON_ACT_CFG bit field.

Table 9. Qualification Timer

QUAL_TIMER[6:5]	Description
00	4 times the Disqualification timer
01	2 times the Disqualification timer
10	8 times the Disqualification timer
11	16 times the Disqualification timer

Frequency Offset Monitoring

Each input reference is monitored for frequency offset failures. The device measures the input frequency and an alarm is raised if the input frequency exceeds the rejection range limit set as per Table 10. To avoid having the alarm toggling in case an input clock frequency is on the edge of the frequency range, a separate, narrower acceptance range must be met before the alarm will clear. The acceptance ranges are also listed in Table 10. See the REF_MON_0.IN_MON_CFG, REF_MON_0.IN_MON_FREQ_CFG, STATUS.IN0_MON_STATUS, ALERT_CFG.IN1_0_MON_ALERT_MASK and GPIO_0.GPIO_CTRL bit fields.

Table 10. Frequency Offset Limits

FREQ_OFFS_LIM[2:0]	Acceptance Range	Rejection Range	Description
000	±9.2 ppm	±12 ppm	Stratum 3, Stratum 3E, G.8262 option 2
001	±13.8 ppm	±18 ppm	
010	±24.6 ppm	±32 ppm	
011	±36.6 ppm	±47.5 ppm	
100	±40 ppm	±52 ppm	SONET Minimum clock. G.813 option 2
101	±52 ppm	±67.5 ppm	
110	±64 ppm	±83 ppm	
111	±100 ppm	±130 ppm	

Advanced Input Clock Qualification

In addition to the Input Clock Selection and Qualification functions mentioned earlier, the following modes are also available.

Input Clock Qualification

For each DPLL the following conditions must be met for the input clock to be valid; otherwise, it is invalid:

- No reference monitor alarms are asserted for that input clock (unless register settings allow the alarms not to affect the decision)
- GPIO used to disqualify that input reference clock is not asserted

Clock Reference Disqualifier through GPIO

GPIO pins can be used to disqualify any input reference clock. If a GPIO is programmed to disqualify a particular input clock, then if that pin is asserted, the corresponding input reference clock will not be available for the DPLL to lock to. For example, a GPIO can be configured as an input to the 8A34002 and connected to a Loss of Signal (LOS) output coming from a PHY device that is providing a recovered clock to one of the DPLLs. If the LOS from the PHY is active, then the DPLL will disqualify that input clock and it will not be available to be locked to. If the disqualified input was the active input for the DPLL, then a switchover process will be triggered if any other valid inputs are available. See the GPIO_0.GPIO_CTRL, GPIO_0.GPIO_REF_INPUT_DSQ_0, GPIO_0.GPIO_REF_INPUT_DSQ_1, GPIO_0.GPIO_REF_INPUT_DSQ_2, and GPIO_0.GPIO_REF_INPUT_DSQ_3 bit fields.

Frame Pulse Operation

Frame pulses are managed with bit fields in the INPUT_n, DPLL_m, OUTPUT_p, and DPLL_CTRL_m modules where: n ranges from 0 to 15; m ranges from 0 to 7; and p ranges from 0 to 11. See Module: INPUT_0, Module: OUTPUT_0, and Module: DPLL_0 in the *8A3xxxx Family Programming Guide*.

In frame pulse operation, two clock signals are working together to signal alignment to a remote receiver. A higher frequency clock is providing a phase aligned reference. A second clock signal (frame signal) is running at a lower, but integer-related rate to the higher frequency clock. The active edge of the frame pulse indicates that the next rising edge of the associated higher frequency clock is to be used as an alignment edge. The 8A34002 supports either rising or falling edges on a frame pulse. The frame signal is usually implemented as a pulse rather than a square wave clock. See the INPUT_0.IN_SYNC, DPLL_0.DPLL_CTRL_2, and DPLL_CTRL_0.DPLL_FRAME_PULSE_SYNC bit fields.

Any input clock and any output clock can be used as frame signal input and output respectively. This is accomplished by configuring the appropriate bits in control registers. A PPES (pulse per even second), 1PPS, 5PPS, 10PPS, 50Hz, 100Hz, 1kHz, 2kHz, 4 kHz, or 8kHz frame input signal can be used with an associated input clock to align a frame output signal and align associated output and frame clock. The frame pulse does not require any specific duty cycle but should have a pulse width of at least 10nsec.

The maximum frequency for the associated input clock is 150MHz, and it can be associated with any supported frame pulse frequency for the frame signal input as long as the integer frequency relationship is maintained. The frame output frequencies are independent of the frame input frequencies; however, the output associated clock and output frame signal must have an integer relationship in order to be aligned.

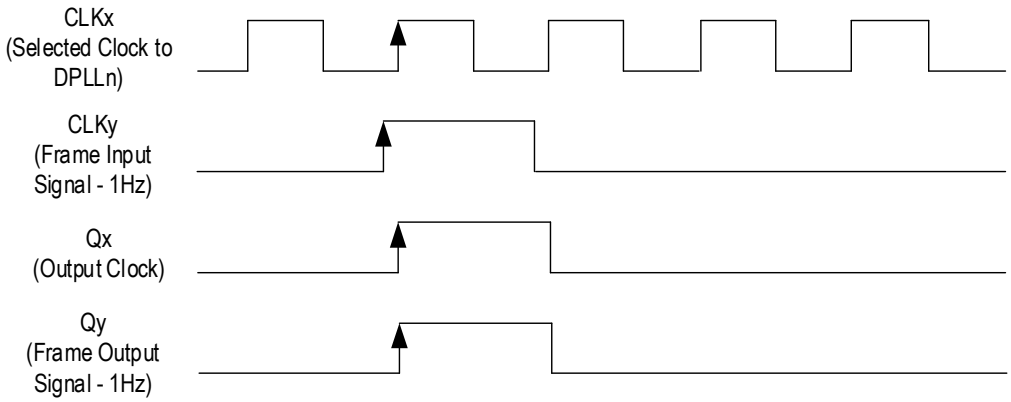
The frame pulse and clock output coming out of the same DPLL are aligned with the first rising edge of the input clock which follows the input frame pulse used by the same DPLL. The 8A34002 allows several different pulse widths to be selected (see Table 15). See the OUTPUT_0.OUT_CTRL_1, OUTPUT_0.OUT_DUTY_CYCLE_HIGH, OUTPUT_0.OUT_DIV, and DPLL_CTRL_0.DPLL_MASTER_DIV bit fields.

When the frame input signal is enabled to synchronize the frame output signal, the output will be adjusted to align itself with the DPLLs selected input clock (associated with the input frame signal) within the input-output alignment limits indicated in AC Electrical Characteristics.

By default, the rising edge of the frame input signal identifies the rising edge of the DPLL's selected input clock. The falling edge of the frame input signal can be used to identify the rising edge of the DPLL's selected input clock by setting the frame pulse configuration register.

An example of the frame pulse operation is provided in Figure 8.

Figure 8. Frame Pulse Operation



In Figure 8, CLKx is the associated input clock and CLKy is the frame pulse, and they are both input to DPLLn. Qx is the output clock that is locked to CLKx, and Qy is the output frame pulse output of DPLLn.

Sync pulses are managed with bit fields in the INPUT_n, DPLL_m, OUTPUT_p, and DPLL_CTRL_m modules where: n ranges from 0 to 15; m ranges from 0 to 7; and p ranges from 0 to 11. See Module: INPUT_0, Module: OUTPUT_0, and Module: DPLL_0 in the *8A3xxxx Family Programming Guide*.

Sync Pulse Operation

A sync pulse scenario occurs similarly to a frame pulse scenario, except that it is the rising edge of the sync signal that is used as the alignment edge rather than an edge of the associated clock.

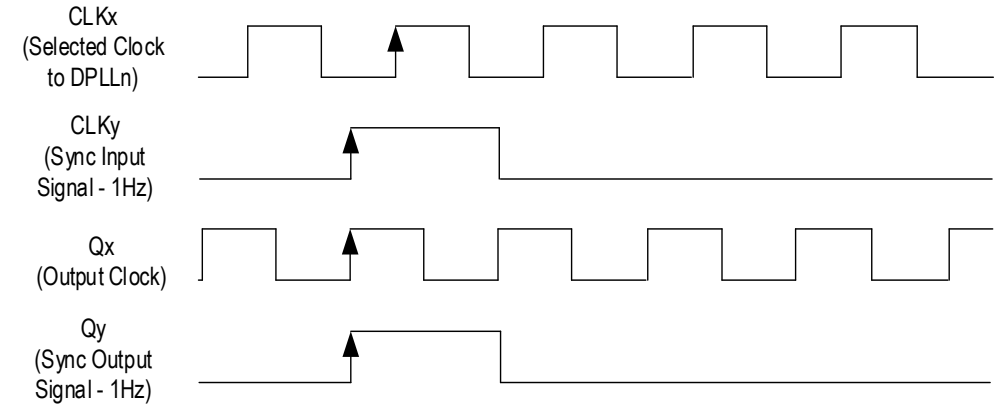
Any input clock and any output clock can be used as sync signal input and output respectively, which is done by configuring the appropriate bits in control registers. A PPES (pulse per even second), 1PPS, 5PPS, 10PPS, 50Hz, 100Hz, 1kHz, 2kHz, 4kHz, or 8kHz sync input signal can be used with an associated input clock to align a sync output signal and output clocks. The sync pulse does not require any specific duty cycle but should have a pulse width of at least 10nsec. See the INPUT_0.IN_SYNC and DPLL_0.DPLL_CTRL_2 bit fields.

The maximum frequency for the associated input clock is 1GHz, and it can be associated with any supported frequency for the sync signal input as long as it is an integer multiple of the sync signal frequency. The sync output frequencies should be an integer relationship of the sync input frequencies.

By default, the sync pulse and clocks output coming out of the same DPLL are aligned with the first rising edge of the sync pulse used by the same DPLL. The falling edge of the sync input signal can be used by setting the frame pulse configuration register.

An example of the sync pulse operation is provided in Figure 9. See the OUTPUT_0.OUT_CTRL_1, OUTPUT_0.OUT_DUTY_CYCLE_HIGH, OUTPUT_0.OUT_DIV, and DPLL_CTRL_0.DPLL_MASTER_DIV bit fields.

Figure 9. Sync Pulse Operation



In Figure 9, CLKx is the associated input clock and CLKy is the sync pulse, and they are both input to DPLLn. Qx is the output clock that is locked to CLKx, and Qy is the output sync pulse output of DPLLn.

Crystal Oscillator Input (XO_DPLL)

The crystal oscillator input is managed with bit fields in the SYS_DPLL_XO module. See Module: SYS_DPLL_XO in the *8A3xxxx Family Programming Guide*.

There is one additional reference clock input that is available: XO_DPLL. This is a single-ended (LVCMOS) input that is intended to be used to provide a stable frequency reference, such as an XO, TCXO or OCXO to the System DPLL. This input is not required in all cases. The crystal oscillator should be chosen accordingly to meet different applications and standard requirements (see application note, *AN-807 Recommended Crystal Oscillators for NetSynchro WAN PLL*). See the SYS_DPLL_XO.XO_FREQ bit field.

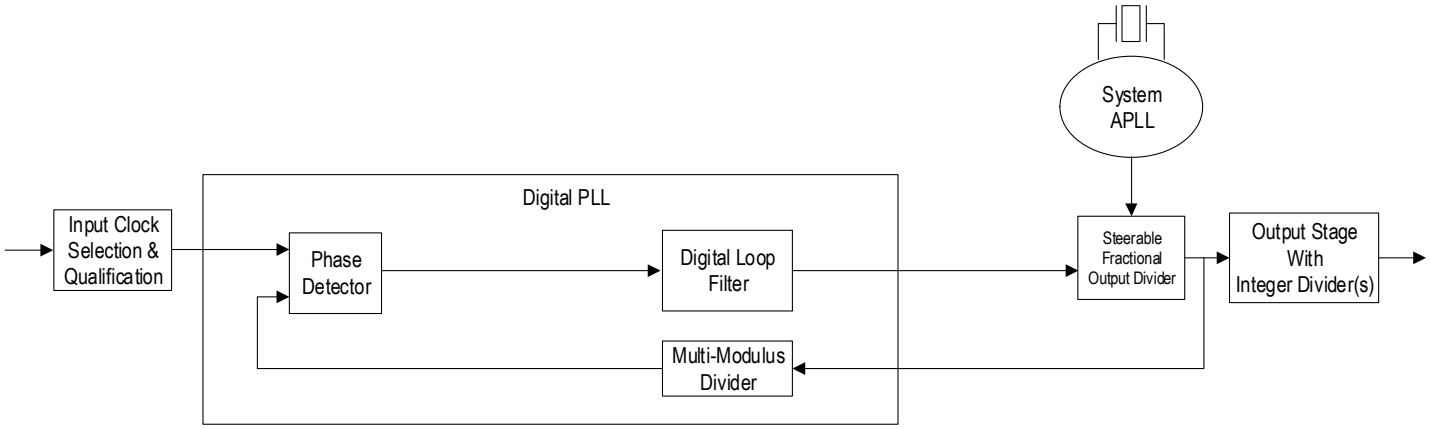
Please note that there is no reference monitoring function on the XO_DPLL input. Failures of this input cannot be detected directly. Since the XO_DPLL input is usually used to drive the SysDPLL which in turn provides a reference clock to the reference monitors for all other input clocks, a simultaneous failure of all monitored input clocks can be inferred to be a failure of the XO_DPLL input in that case.

Digital Phase Locked Loop (DPLL)

DPLLs 0 to 7 are managed with bit fields in the DPLL_n and DPLL_CTRL_n modules where: n ranges from 0 to 7. The System DPLL is managed with bit fields in the SYS_DPLL and SYS_DPLL_CTRL modules. See Module: DPLL_0, Module: DPLL_CTRL_0, Module: SYS_DPLL, and Module: SYS_DPLL_CTRL in the *8A3xxxx Family Programming Guide*.

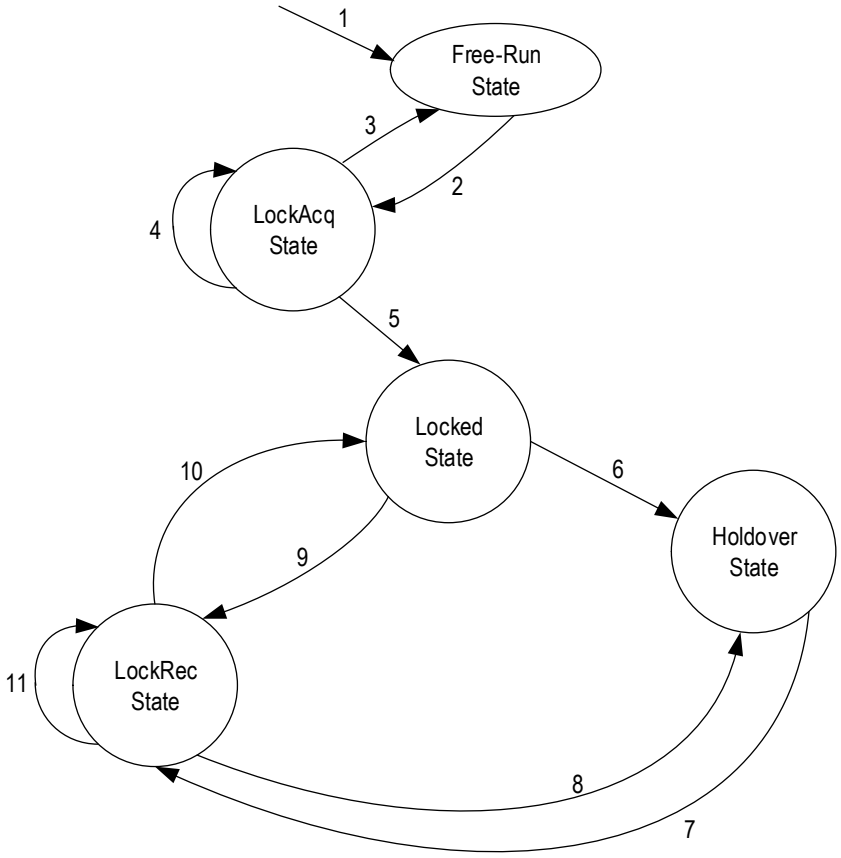
DPLLs 0 to 7 are exactly the same; the System DPLL shares the same functional block diagram as the other DPLLs but it is not connected directly to an output stage. One channel of the DPLL is shown in Figure 10.

Figure 10. DPLL Channel



The DPLL operating mode operation can be set to automatic, forced locked, forced free-run, and forced holdover. The operating mode can be controlled by setting the appropriated bits in the DPLL mode register. When the DPLL is set to automatic then an internal state machine will control the states automatically. The automatic state machine is displayed in Figure 11. See the DPLL_0.DPLL_MODE and DPLL_0.DPLL_CTRL_0 bit fields.

Figure 11. DPLL Automatic State Machine



In Figure 11, the changes of state are based on the following:

1. Reset, the device enters Free-Run State.
2. Once an input clock is qualified and it is selected: enter the LockAcq State.
3. If the DPLL selected input clock is disqualified AND no qualified input clock is available: go back to Free-Run State.
4. DPLL switches to another qualified clock: remain in LockAcq State.
5. The DPLL locks to the selected input clock: enter Locked State.
6. The DPLL selected input clock is disqualified AND No qualified input clock is available: enter Holdover State.
7. A qualified input clock is now available: enter LockRec State.
8. If the DPLL selected input clock is disqualified AND no qualified input clock is available: go back to Holdover State.
9. The DPLL switches to another qualified clock: enter LockRec State.
10. The DPLL locks to the selected input clock: go to Locked State.
11. The DPLL switches to another qualified clock: remain in LockRec State

In items 4, 9, and 11, the DPLL switches to another qualified clock due to the selected input clock being disqualified, or the device is set to revertive mode and a qualified input clock with a higher priority becomes valid, or the device is set to Forced selection to another input clock.

Free-Run Mode

In Free-Run mode, the DPLL synthesizes clocks based on the system clock (crystal oscillator) and has no influence from a current or a previous input clock. See the DPLL_0.DPLL_MODE and SYS_DPLL.SYS_DPLL_MODE bit fields.

Combo mode can be used with Free-Run mode. In this case, the input clock of the combo master affects the combo slave's free-Run frequency (for more information, see Combo Mode).

Locked Mode

In Locked mode, the DPLL is synchronized to an input clock. The frequency and phase of the output clock track the DPLL selected input clock. The bandwidth (BW) and damping factor are programmable and are used by the DPLL when locked to an input reference. The following table includes some common bandwidth settings and their associated applications. See the bit fields in the DPLL_CTRL_0 and DPLL_SYS_DPLL_CTRL modules.

Table 11. DPLL Bandwidth

DPLL Bandwidth	Description
1mHz	GR-1244 Stratum 2/3E, BW ≤ 1mHz G.812 Type II/III, BW ≤ 1mHz
3mHz	G.812 Type I (SSU-A), BW ≤ 3mHz
20mHz	Renesas recommended for locking to 1Hz/1PPS
65mHz	G.8273.2, 0.05 < BW ≤ 0.1Hz
100mHz	GR-253 Stratum 3/SMC, BW ≤ 0.1Hz G.812 Type IV, G.813 SEC2 G.8262 EEC2, BW ≤ 0.1Hz G.8273.2, 0.05 < BW ≤ 0.1Hz
1.1Hz	G.813 SEC1, G.8262 EEC1, 1 ≤ BW ≤ 10Hz G.8262.1 eEEC, 1 < BW ≤ 3Hz
3Hz	GR-1244 Stratum 3, BW < 3Hz G.8262.1 eEEC, 1 < BW ≤ 3Hz
10Hz	G.813 SEC1, G.8262 EEC1, 1 < BW ≤ 10Hz
25Hz	Jitter attenuators and Clock generators (General)
100Hz	G.8251 (OTN 1G)
300Hz	G.8251 (OTN)
1kHz	Jitter attenuators and Clock generators (100G)
10kHz	Jitter attenuators and Clock generators (10G)
12kHz	Jitter attenuators and Clock generators (1G/SONET/SDH)

Holdover Mode

If all the input clocks for a particular DPLL become invalid, then the DPLL will enter the holdover state. See the DPLL_0.DPLL_MODE, DPLL_0.DPLL_HO_ADVCD_HISTORY, DPLL_0.DPLL_HO_ADVCD_BW, and DPLL_0.DPLL_HO_CFG bit fields.

In holdover mode, the DPLL uses stored frequency data acquired in Locked mode to control its output clocks. There are several programmable modes for the frequency offset acquisition method; it can use the frequency offset just before it entered holdover state (simple holdover), or a previously stored post-filtered frequency offset (advanced holdover).

For the advanced holdover mode, the holdover value can be post filtered and is stored in two registers at a programmable rate while the DPLL is in locked state. When the DPLL enters the advanced holdover mode, the oldest register value is restored into the integrator inside the DPLL. The rate at which the holdover registers are updated is programmable between 0s and 63s in steps of 1s.

Note: To establish an accurate holdover value for the advanced holdover mode, a stable estimate of the average input reference frequency is necessary before entering holdover. Therefore, the DPLL must have been in the locked state for a period that is based on the holdover settings (e.g., the lower the bandwidth setting for the holdover filter, the longer it takes to acquire the accurate holdover value).

The DPLL can also be forced into the holdover mode. If the forced holdover mode is used, then the DPLL will stay in holdover even if there are valid references available for the DPLL to lock to.

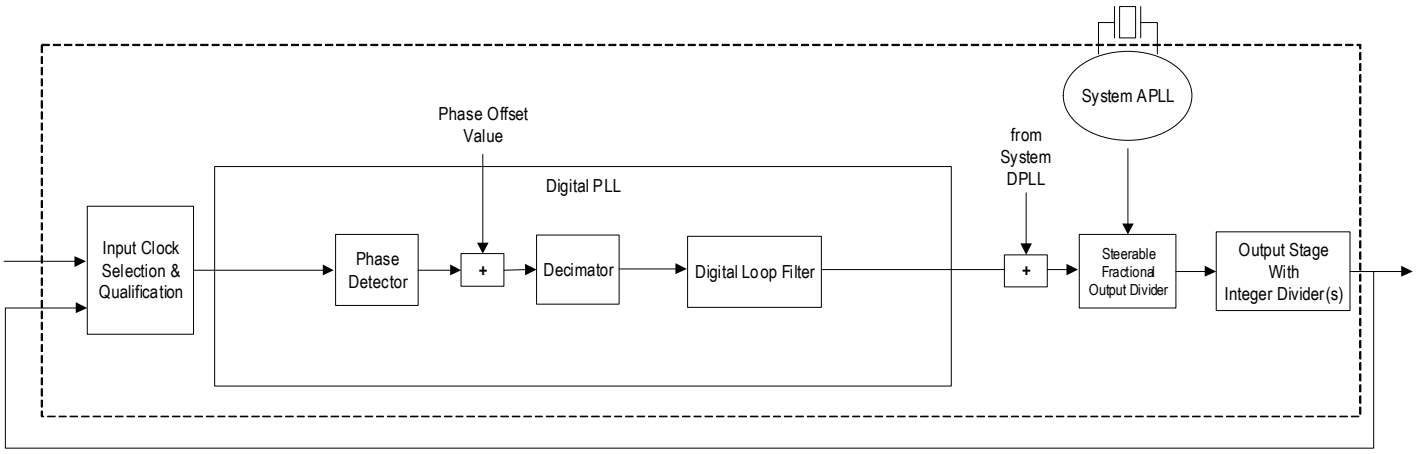
Manual Holdover Mode

In Manual Holdover mode, the DPLL state machine is forced into the Holdover state but the frequency offset is set by the DPLL manual holdover value register bits under user control. See the DPLL_0.DPLL_MODE, DPLL_0.DPLL_HO_CFG, and DPLL_CTRL_0.DPLL_MANUAL_HOLDOVER_VALUE bit fields.

External Feedback

The 8A34002 supports the use of an external feedback path, where one of the channel's output clocks is externally connected to one of the reference clock inputs as shown in Figure 12. External feedback automatically maintains tight alignment of the output phase with the reference input phase. This alignment is done by dynamically compensating for changes in PCB trace delay and external buffer propagation delay caused by changes in temperature and voltage. Use of the external feedback path is referred to as a zero delay phase locked loop (ZDPLL), where output frequencies are different to the reference clock input, or a zero delay buffer (ZDB), where all output frequencies are the same as the reference clock input.

Figure 12. External Feedback



For both ZDPLL and ZDB, the frequency of the external feedback clock must be the same as the reference clock input. For this reason, all of the reference clock inputs must be the same frequency when using automatic reference switching. Otherwise, the external feedback clock must be reconfigured to match the new input reference clock frequency prior to manually switching to the new reference. See the DPLL_0.DPLL_CTRL_2 and INPUT_0.IN_MODE bit fields.

DPLL Input Clock Qualification and Selection

Any Digital PLL (DPLL) can use any of the inputs as its reference. Several options exist to control how the DPLLs select which input to use at any moment in time. Whether a particular input is qualified for use at any time is based on the reference monitors. DPLLs can be set in any of the modes shown in Table 12. There is an independent reference selection process for each DPLL. See the DPLL_0.DPLL_REF_MODE bit field.

Table 12. DPLL Reference Mode

MODE[3:0]	Description
0000	Automatic input clock selection
0001	Manual input clock selection
0010	GPIO
0011	Slave
0100	GPIO_Slave
0101–1111	Reserved

Automatic Input Clock Selection

If automatic input clock selection is used then the input clock selection is determined by the input clock being valid, the priority of each input clock, and the input clock configuration.

Each input can be enabled or disabled by setting register bits. If the input is enabled and reference monitors declare that input valid, then that input is qualified to be used by the DPLL. Within all the qualified inputs, the one with the highest priority is selected by the DPLL. The input clock priority is set by setting the appropriate register bits. If a user wanted to designate several inputs as having the same priority, then an additional table allows several outputs to be placed in a group of equal priority. See the DPLL_0.DPLL_REF_MODE and DPLL_0.DPLL_REF_PRIORITY_0 bit fields.

Manual Input Clock Selection via Register or GPIO

If manual input clock selection is chosen then the DPLL will lock to the input clock indicated by register bits or by selected GPIO pins. The results of input reference monitoring do not affect the clock selection in manual selection mode. If the DPLL is locked to an input clock that becomes invalid, then the DPLL will go into holdover mode even in the case where there are other input clocks that are valid. See the DPLL_0.DPLL_REF_MODE, DPLL_CTRL_0.DPLL_MANU_REF_CFG, GPIO_0.GPIO_MAN_CLK_SEL_0, GPIO_0.GPIO_MAN_CLK_SEL_1, and GPIO_0.GPIO_MAN_CLK_SEL_2 bit fields.

Slave or GPIO Slave Selection

This mode of clock selection is used when the 8A34002 is acting as an inactive, redundant clock source to another timing device. The other device is the master and this device is the slave. When Slave mode is selected via registers, a specific input (from the master timing device) is also indicated. That input and only that input is used in this mode. GPIO Slave mode involves the same configuration settings as if the part were a master, but a GPIO input is used to tell this device that it is now the slave and to switch to and monitor the designated input only. See the DPLL_0.DPLL_REF_MODE, DPLL_0.DPLL_SLAVE_REF_CFG, GPIO_0.GPIO_CTRL, and GPIO_0.GPIO_SLAVE bit fields.

DPLL Switchover Management

Reference switching for DPLLs 0 to 7 is managed with bit fields in the DPLL_n and DPLL_CTRL_n modules where: n ranges from 0 to 7. Reference switching for the System DPLL is managed with bit fields in the SYS_DPLL module. See Module: DPLL_0, Module: DPLL_CTRL_0 and Module: SYS_DPLL in the *8A3xxxx Family Programming Guide*.

Revertive and Non-Revertive Switching

All DPLLs support revertive and non-revertive switching, with the default being non-revertive. During the reference selection process, a DPLL selects the valid reference with the highest priority then the DPLL locks to that input clock. In the case of non-revertive switching, the DPLL only switches to another, higher priority reference if the current reference becomes invalid. Non-revertive switching minimizes the amount of reference switches and therefore is the recommended mode. See the DPLL_0.DPLL_CTRL_0 bit field.

If revertive switching is enabled and a higher priority clock becomes valid, then the DPLL will switch to that higher priority input clock unless that clock is designated as part of the same group (i.e., should be considered of equal priority). See the DPLL_0.DPLL_REF_PRIORITY_0 bit field.

Hitless Reference Switching

All 8A34002 DPLLs support Hitless Reference Switching (HS). HS is intended to minimize the phase changes on DPLL output clocks when switching between input references that are not phase aligned, and when exiting the holdover state to lock to an input reference. HS is enabled or disabled through register settings.

If enabled for a DPLL, HS is triggered if either of the following conditions occurs:

- DPLL is locked to an input reference and switches to a different input reference
- DPLL exits the Holdover state and locks to an input reference

When a DPLL executes a hitless reference switch, it enters a temporary Holdover state (without asserting a holdover alarm), it then measures the initial phase offset between the selected input reference and the DPLL feedback clock. The DPLL uses the measured initial phase offset as the zero point for its phase detector so that it does not align its output with the selected input reference, thereby minimizing the resulting phase transient. The DPLL will track its selected input reference and will maintain the initial phase offset.

Similarly, when a DPLL exits the Holdover state and locks to an input reference it first measures the initial phase offset between the selected input reference and the DPLL feedback clock. The DPLL uses the initial phase offset as the zero point for its phase detector so that it does not align its output with the selected input reference, thereby minimizing the resulting phase transient. The DPLL will track its selected input reference and will maintain the initial phase offset.

There are other cases where hitless reference switching can be used in synchronization applications with physical and/or packet clocks. For information on such applications, please contact Renesas.

Two types of hitless reference switching are supported:

- HS Type 1 - Compliant with ITU-T reference switching requirements. The output phase change due to reference switching is influenced by the frequency of the newly selected reference, see Table 37.
- HS Type 2 - Compliant with ITU-T reference switching requirements. The output phase change due to reference switching is not influenced by the frequency of the newly selected reference, see Table 37.

When a DPLL executes a hitless reference switch using HS Type 2, the outputs of any Satellite Channels associated with that DPLL will exhibit small (several ps) random phase changes relative to the outputs of the DPLL. These phase changes will accumulate with each hitless reference switching event.

HS is designed to compensate for phase differences between the DPLL feedback clock and the selected input reference. When a hitless reference switch is executed, if the fractional frequency offset of the selected input reference is different from that of the DPLL, then the output clocks will experience a transient as the DPLL pulls-in to the input reference.

Phase Slope Limiting

Phase Slope Limiting (PSL) can be enabled and independently programmed for each of the DPLLs. PSL is particularly useful in the initial locking to an input or during switchover between clock inputs. If PSL is enabled then the rate of change of phase of the output clock is limited by the DPLL. The PSL settings for the device are very flexible, allowing any slope from 1ns/s to 65.536µs/s with a granularity of 1ns/s, including the values needed to meet Telecom standards as displayed in Table 13. See the DPLL_CTRL_0.DPLL_PSL bit field.

Table 13. Some Key DPLL Phase-Slope Limits Supported

DPLL PSL	Description
Unlimited	Limited by DPLL loop bandwidth setting
61 µs/s	Telcordia GR-1244 ST3
7.5 µs/s	G.8262 EEC option 1, G.813 SEC option 1
885 ns/s	Telcordia GR-1244 ST2, ST3E, and ST3 (objective)

DPLL Frequency Offset Limit Setting

Each DPLL has an independent setting to limit its maximum frequency range. This setting is used in conjunction with the advanced reference monitoring to provide pull-in / hold-in limit enforcement as required in many telecom standards. It will also limit the frequency deviation during locking, during holdover, and while performing switchovers. This limit must be set wide enough to cover the expected frequency range of the input when locking. See the DPLL_0.DPLL_MAX_FREQ_OFFSET bit field.

DPLL Fast Lock Operation

DPLL fast lock operation is managed with bit fields in the DPLL_n modules where: n ranges from 0 to 7. See Module: DPLL_0 in the *8A3xxxx Family Programming Guide*.

Each DPLL supports a Fast Lock function. There are four options the user can choose from to perform the fast lock:

- Frequency Snap
- Phase Snap
- Open-loop phase pull-in (mutually exclusive with Phase Snap)
- Wide Acquisition Bandwidth

Any of the options can be independently enabled or disabled, and selected to be applied when the DPLL is in either the LOCKACQ state or the LOCKREC state. Although the options are mutually exclusive, the order of precedence is as listed (with frequency snap being the highest). See the DPLL_0.DPLL_FASTLOCK_CFG_0 bit field.

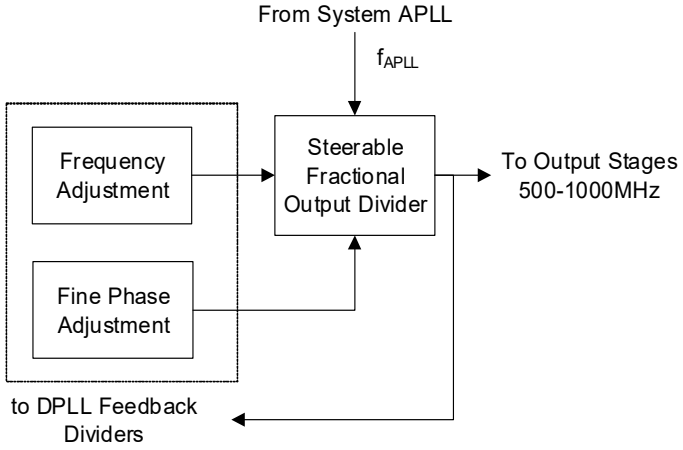
The frequency and phase snap options are recommended for locking to mid-kHz-range input clocks or lower. For frequency snap, the 8A34002 will measure the input clock from the current DPLL operating frequency, determine an approximate frequency offset, and digitally write that directly to the steerable FOD block, causing the output frequency to snap directly to the correct output frequency. The frequency snap can be optionally limited using a Frequency Slope Limit (FSL). For the phase snap and the open loop phase pull-in options, the measurement is used to determine the phase offset. With phase snap, the phase is snapped to the correct value; with open loop pull-in, the DPLL's PFD and LPF are temporarily isolated to allow for an unfiltered phase pull-in to the correct value. The combination of these methods will achieve lock very quickly, but there may be severe disruptions on the output clock while locking occurs; mainly due to the frequency/phase snaps. See the DPLL_0.DPLL_MAX_FREQ_OFFSET and DPLL_0.DPLL_FASTLOCK_FSL bit fields.

The wide acquisition bandwidth option uses the DPLL in a normal operating mode, but with temporary relaxation of items like DPLL loop bandwidth, phase slope limits (PSL), or damping factor until lock is achieved. At that point, the normal DPLL limits are resumed. The user can control what limits are to be applied. In addition, for LOCKACQ state only, the DPLL's bandwidth may be temporarily opened to its maximum for a short duration of time (in ms), with the temporary phase slope limit still being applied. This pre-acquisition option is applied before the wide acquisition bandwidth option. These methods are recommended for higher frequency signals since it results in fewer perturbations on the output clock. It also allows the user to trade-off the level of changes on the clock during the locking process versus the speed of locking. See the DPLL_0.DPLL_FASTLOCK_CFG_1, DPLL_0.DPLL_FASTLOCK_PSL, and DPLL_0.DPLL_FASTLOCK_BW bit fields.

Steerable Fractional Output Divider (FOD)

The 8A34002 has multiple Steerable Fractional Divider blocks as shown in Figure 13. Each block receives a high-frequency, low-jitter clock from the System APLL. It then divides that by a fixed-point (non-integer) divide ratio to produce a low-jitter output clock that is passed to the output stage(s) for further division and/or adjustment and also to the DPLL feedback dividers. The FOD output will be in the frequency range of 500MHz to 1GHz, and is independent of the output frequencies from any other FOD and from the System APLL.

Figure 13. Steerable Fractional Output Divider Block



The output frequency is determined by dividing the System APLL frequency (f_{APLL}) by the Fractional Divider. Since f_{APLL} is between 13.4GHz and 13.9GHz and the FOD output (f_{FOD}) is between 500MHz and 1GHz, there is a limited range of valid FOD divide ratios (from 13.4 to 27.8). The Fractional Divider involves two unsigned integer values, representing the integer (INT) and fraction (FRAC) portion of the divide ratio. The fraction portion is an integer representing the 43-bit numerator of a fraction, where the denominator of that fraction is fixed at 2^{43}. Renesas' Timing Commander Software can be used to determine if a particular output frequency can be represented accurately, and if not, the magnitude of the inaccuracy. If additional information is required, please contact Renesas directly.

The equation for the FOD output frequency is as follows.

$$f_{FOD} = \frac{f_{APLL}}{\left(INT + \frac{FRAC}{2^{43}}\right)}$$

Note: Fractions that approach 0, 1, or 1/2 can result in increased phase noise on the output signal due to integer-boundary spurs. It is recommended that System APLL frequency and FOD divider settings be coordinated to avoid such fractions.

Fine adjustments in the phase of the FOD output may also be made. A phase adjustment is performed by increasing or decreasing the frequency of operation of the FOD for a period of time. This results in the clock edges of the FOD output clock being advanced (increased FOD output frequency will move edges to the left as seen on an oscilloscope relative to some fixed reference point) or delayed (decreased FOD output frequency moves edges to the right) by some amount. The user writes a signed integer value to the fine adjust register of the FOD over the serial port. This value represents the number of picoseconds the clock edges are to be advanced (negative value) or delayed (positive value). The user can also specify a rate of phase change as Fast, Medium, or Slow. A Fast setting will apply a larger frequency change for a shorter period of time, whereas a Slow setting will apply a smaller frequency change for a longer period. Medium will choose an intermediate frequency and duration. This setting is used to accommodate devices on the output clocks that may not be able to track a fast phase change. Any number of phase changes may be applied, so the range of phase change is effectively infinite.

Note that this method of fine phase adjustment should only be used when the FOD is operating in an open-loop manner. If the FOD is being used as part of a closed-loop control, where the output phase is observed and used to track a reference input, the feedback loop may act to remove the phase adjustment. If the FOD is part of a closed-loop operation, then it is recommended that phase or frequency adjustment be performed via the Digital Phase Locked Loop (DPLL) logic.

FOD Multiplexing and Output Stages

FOD Multiplexing and the Output Stages are managed with bit fields in the OUT_DIV_MUX and OUTPUT_DIV_MUX modules. See Module: OUT_DIV_MUX and Module: OUTPUT_0 in the *8A3xxxx Family Programming Guide*.

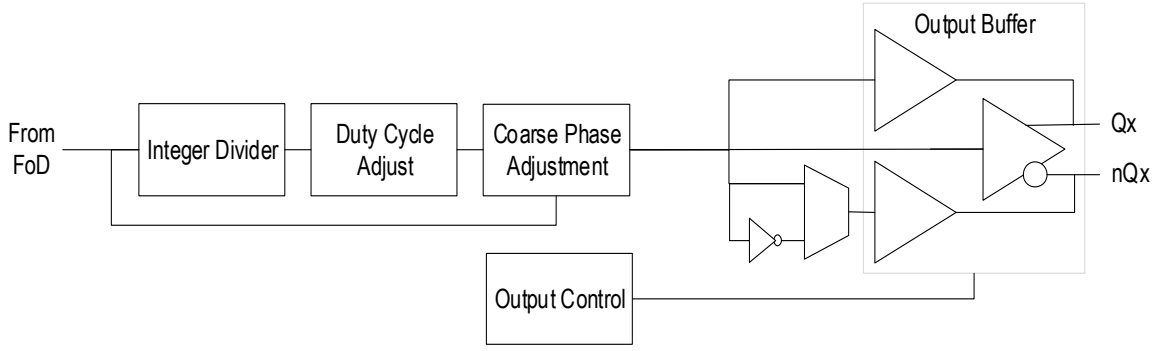
The 8A34002 has multiple output stages that are associated with the FODs and output pins as shown in the following table. See the OUT_DIV_MUX.OUT_DIV8_MUX and OUT_DIV_MUX.OUT_DIV11_MUX bit fields.

Table 14. FOD to Output Stage to Output Pin Mappings

Output Stage	Single / Dual	Output Pins	FODs that can Drive this Stage
0	Dual	Q0 / nQ0, Q1 / nQ1	FOD_0
1	Dual	Q2 / nQ2, Q3 / nQ3	FOD_1
2	Dual	Q4 / nQ4, Q5 / nQ5	FOD_2
3	Dual	Q6 / nQ6, Q7 / nQ7	FOD_3

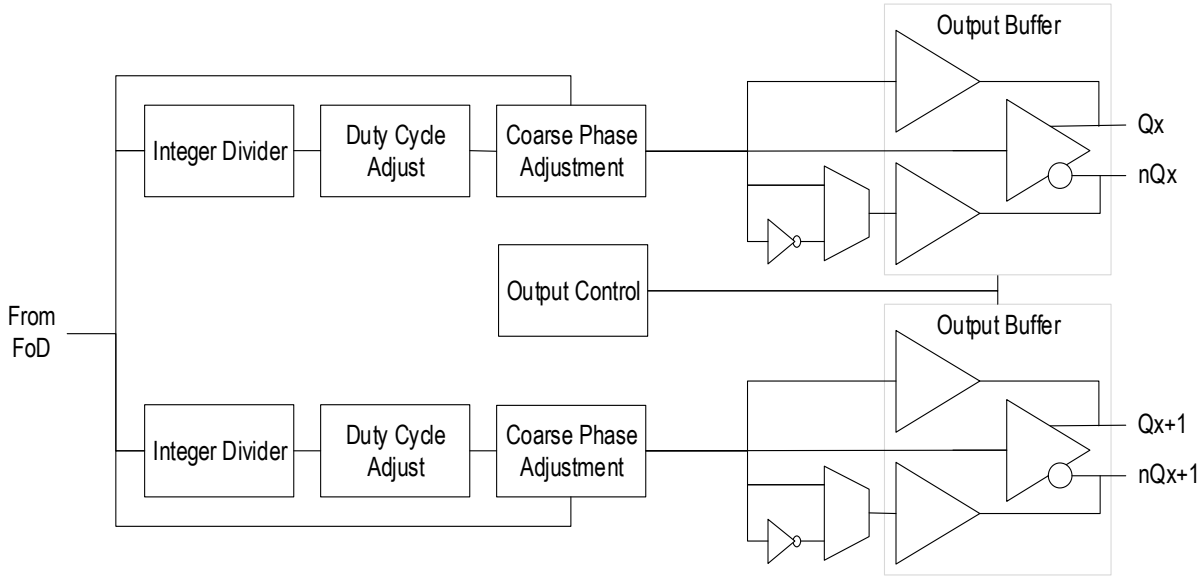
The single output stages are shown in Figure 14 and the dual output stages are shown in Figure 15.

Figure 14. Single Output Stage



Other than having two copies of each functional block fed from the same FOD, both single and dual output stages behave the same. Similarly, the two paths within the dual output stages behave the same as each other. Descriptions of each functional block are provided in the following sub-sections.

Figure 15. Dual Output Stage



Integer Output Divider

The integer output divider takes a clock signal from the FOD stage ranging from 500MHz to 1GHz and divides it by a 32-bit integer value. This results in output frequencies that range from 1GHz down to less than 0.5Hz, depending on the frequency coming from the FOD. For information on which FOD can be used with which output stage / output pins, see the previous table. See the OUTPUT_0.OUT_DIV bit field.

Output Duty Cycle Adjustment

The 8A34002 also has a number of options for generating pulses with different duty cycles. While these are intended primarily for frame pulses or sync pulses, duty cycle adjustment options remain accessible in all modes of operation.

As described in the previous section, each output is a divided down clock from the FOD. By default, this resulting clock will be a 50/50 duty cycle clock. If a pulse, such as a frame or sync pulse, is to be derived from the resulting clock, then the high pulse width can be programmed by a 32-bit integer value, representing the number of FOD clock cycles in the high period. This value must be less than the integer output divider value. Several examples are shown in Table 15. See the OUTPUT_0.OUT_DUTY_CYCLE_HIGH and OUTPUT_0.OUT_CTRL_1 bit fields.

Table 15. Output Duty Cycle Examples

FOD_n Frequency	Integer Output Divider Register Value OUT_DIV[31:0]	Output Frequency	Output Duty Cycle High Register Value[a] OUT_DUTY_CYCLE_HIGH[31:0]	Resulting Pulse width
500MHz	2	250MHz	0	2ns (50% / 50%)[b]
	500,000,000	1Hz (1PPS)	500	1µs[c]
655.36MHz	80	8.192MHz	0	61.035ns (50% / 50%)
	81920	8kHz	80	122ns (1UI[d])

[a] Pulses are always created by this logic as high-going pulses. If a low-going pulse is desired, the nQx output pin can be used with the inverter option selected.

[b] For precision of duty cycle achieved, see Table 37.

[c] This represents the high period of a pulse.

[d] The UI method of specifying a pulse width is often used for generation of a frame pulse. A frame pulse is always associated with another regular clock, so UI = Unit Interval of the clock output associated with the frame pulse. In this example, the associated clock is the 8.192MHz clock.

Output Coarse Phase Adjustment

The 8A34002 supports two methods for adjustment of the phase of an output clock. Fine phase adjustment can be performed in the Digital Phase Locked Loop (DPLL) block, so it can only be adjusted per-channel. In addition, coarse phase adjustment can be performed in the Output Stage and so can be performed on a per-output basis. Coarse adjust will move an output edge in units of the period of the FOD clock (T_{FOD}). Subject to the following rules, an infinite adjustment range is possible and the clock edge can be either advanced or delayed. Note that if an output phase adjustment is needed for a signal that does not meet these rules, fine phase adjustment should be used. See the OUTPUT_0.OUT_CTRL_1 and OUTPUT_0.OUT_PHASE_ADJ bit fields.

Rules for application of coarse phase adjust include the following:

- Coarse phase adjust lengthens or shortens the high and/or low pulses of the output clock in units of T_{FOD}.
- The coarse phase adjust will not shorten the output clock period to anything less than $2 \times T_{FOD}$ high + $2 \times T_{FOD}$ low.
  - This means coarse adjust cannot be used if the integer divider ratio is 1, 2, 3, or 4.
- Coarse phase adjust can lengthen or shorten (subject to the above rule) the output clock period by up to $2^{32} \times T_{FOD}$ high + $2^{32} \times T_{FOD}$low.
  - Such a large change in a single clock period may have serious effects on devices receiving the output clock, so the user is cautioned to consider that before applying a large adjust at one time. Multiple smaller adjustments can be performed by the user over a period of time to avoid this.
- Logic within the 8A34002 will take the positive (lengthen the period) or negative (shorten the period) adjustment value provided by the user and apply it in a single clock period to the limits listed in the preceding rules.
  - For clock signals that are using 50% / 50% duty cycle, adjustments will be applied approximately equally to the high and low portions of the clock.
  - For clock signals using other duty cycle selections, adjustments will only be applied to the low portion of the clock.
- The user can apply as many of these updates as desired, so the range of adjustment is unlimited.

Output Buffer

The output buffer structure will generate either one differential or two single-ended output signals as programmed by the user. A single output stage will have one output buffer structure and a dual stage one will have two output buffers. Each output buffer has a separate V_{DDO_Qx} pin that will affect its output voltage swing as indicated below and in Table 31. See the OUTPUT_0.OUT_CTRL_0 and OUTPUT_0.OUT_CTRL_1 bit fields.

Output Buffer in Differential Mode

When used as a differential output buffer, the user can control the output voltage swing (V_{OVS}) and common mode voltage (V_{CMR}) of the buffer. Which V_{OVS} and V_{SWING} settings may be used with a particular V_{DDO_Qx} voltage are described in Table 16. Note that V_{DDO_Qx} options of 1.5V or 1.2V cannot be used in differential mode. The nominal voltage swing options are 410mV, 600mV, 750mV, and 900mV. The nominal voltage crossing points options are 0.9V, 1.1V, 1.3V, 1.5V, 1.7V, 1.9V, 2.1V, and 2.3V. For actual values under different conditions, see Table 31.

Table 16. Configurable Output Mode Options

V_{DDO_Qx} PAD_VDDO[4:2]	SWING Setting PAD_VSWING[4:3]	V_{OVS} Options Supported[a]	V_{CMR} Options Supported PAD_VOS[7:5]
3.3V	00	410mV	0.9V, 1.1V, 1.3V, 1.5V, 1.7V, 1.9V, 2.1V, 2.3V
	01	600mV	0.9V, 1.1V, 1.3V, 1.5V, 1.7V, 1.9V, 2.1V, 2.3V
	10	750mV	0.9V, 1.1V, 1.3V, 1.5V, 1.7V, 1.9V, 2.1V
	11	900mV	0.9V, 1.1V, 1.3V, 1.5V, 1.7V, 1.9V
2.5V	00	410mV	0.9V, 1.1V, 1.3V, 1.5V, 1.7V
	01	600mV	0.9V, 1.1V, 1.3V, 1.5V
	10	750mV	0.9V, 1.1V, 1.3V, 1.5V
	11	900mV	0.9V, 1.1V, 1.3V
1.8V	00	410mV	0.9V, 1.1V, 1.3V
	01	600mV	0.9V, 1.1V
	10	750mV	0.9V
	11	900mV	0.9V

[a] Voltage swing values are approximate values. For actual swing values, see Table 31, Table 32, and Table 33.

The user can use this programmability to drive LVDS, 2.5V LVPECL, and 3.3V LVPECL receivers without AC-coupling. Most other desired receivers can be addressed with this programmable output, but many will require AC-coupling or additional terminations. For termination recommendations for some common receiver types, see the appropriate section of the Applications Information or contact Renesas using the contact information on the last page of this document.

Output Buffer in Single-Ended Mode

When used as a single-ended output buffer, two copies of the same output clock are created with LVCMOS output levels. Each clock will have the same frequency, phase, voltage, and current characteristics. The only exception to this is that the user can program the clock from the nQx output pad to be inverted in phase relative to the one coming from the Qx output pin. The non-inverted setting may result in greater noise on these outputs and increased coupling to other output clocks in the device, so it should be used with caution. See the OUTPUT_0.OUT_CTRL_0 bit field.

In this mode, the output buffer supports 1.2V, 1.5V, 1.8V, 2.5V, or 3.3V V_{DDO_Qx} voltages. For each output voltage, there are four impedance options that can be selected from. For actual voltage and impedance values under different conditions, see Table 34. See the OUTPUT_0.OUT_CTRL_1 bit field.

General Purpose Input/Outputs (GPIOs)

Unless otherwise specified, any referenced bits or registers throughout this section reside in the GPIO_X section of the register map, where *X* ranges from 0 to 15. Base address for GPIO_0 is C8C2h (offset 000h to 010h). GPIO_USER_CONTROL base address C160h (offset 000h to 001h). For more information about other GPIOs, see the *8A3xxxx Family Programming Guide*.

The GPIO signals are intended to provide a user with a flexible method to manage the control and status of the part via pins without providing dedicated pins for each possible function that may be wasted in a lot of applications. The GPIOs are fully configurable so that any GPIO can perform any function on any target logic block.

GPIO Modes

Each GPIO pin can be individually configured to operate in one of the following modes. Note that these modes are effective only once the 8A34002 has completed its reset sequence. During the reset sequence one or more of these pins can have different functions as outlined in Use of GPIO Pins at Reset:

- General Purpose Input – In this mode of operation, the GPIO pin will act as an input whose logic level will be monitored and reflected in an internal register that may be read over the serial port. This is the default mode if no other option is programmed in OTP or EEPROM.
- General Purpose Output – In this mode of operation, the GPIO pin will act as an output that is driven to the logic level specified in an internal register. That register can be written over the serial port.
- Alarm output – In this mode of operation, the GPIO pin will act as a single-purpose alarm or Alert (aggregated alarm) output. For information on when an alarm output will be asserted or released and alarm sources, see Alarm Output Operation. Note that each GPIO can be independently configured. If multiple GPIOs are configured the same way, they will all have the same output values.
  - Loss-of-Signal status – In this mode of operation, the GPIO pin will act as an active-high Loss-of-Signal output. There is an option to invert this output polarity via register programming. When the GPIO output is asserted, that indicates the selected input reference monitor is indicating an alarm condition. The related reference monitor and the associated GPIO pin are configured via registers. Configuration of the reference monitor will determine what constitutes an alarm. For more information on reference monitor configuration, see Reference Monitoring. Note that the GPIO output reflects the actual state of the alarm signal from the selected reference monitor. This is not a latched or “sticky” signal. This is different than the other alarm sources below.
  - Loss-of-Lock status – In this mode of operation, the GPIO pin will act as a Loss-of-Lock output. The related PLL channel and associated GPIO pin are configured via registers. For more information on alarm conditions, see Digital Phase Locked Loop (DPLL) and System APLL.

  The GPIO can be programmed to show the active Loss-of-Lock status, in which case a high state on the pin will indicate that the associated DPLL or APLL is not currently locked. Alternatively, the GPIO can be programmed to flag any changes in the lock status in a “sticky” bit mode. In this mode of operation, a high state will indicate that the lock status of the associated DPLL or APLL has changed. Either the PLL has entered or left the locked state. The GPIO can be programmed to invert this polarity so that a low state indicates a status change. In either case, since this is a “sticky” status, it must be cleared by register access to the “stick” clear register to remove the alarm signal.
  - Holdover status – In this mode of operation, the GPIO pin will act as a Holdover status. The related PLL channel and associated GPIO pin are configured via registers. For more information on alarm conditions, see Digital Phase Locked Loop (DPLL).

  The GPIO can be programmed to show the active Holdover status, in which case a high state on the pin will indicate that the associated DPLL is currently in holdover state. Alternatively, the GPIO can be programmed to flag any changes in the holdover status in a “sticky” bit mode. In this mode of operation, a high state will indicate that the holdover status of the associated DPLL has changed. Either the PLL has entered or left the holdover state. The GPIO can be programmed to invert this polarity so that a low state indicates a status change. In either case, since this is a “sticky” status, it must be cleared by register access to the “stick” clear register to remove the alarm signal.
  - Alert (aggregated alarm) status – In this mode of operation, the GPIO will act as the logical OR of all alarm indicators that are enabled to drive this output. Only “sticky” bits are available to drive the GPIO in this mode. This output will be asserted if any of the “sticky” bits are asserted and enabled to cause the Alert (aggregated alarm). To clear this output, all contributing “sticky” bits must be individually cleared. This output will be active-high to indicate one or more alarms are asserted. There is an option to invert this output polarity via register programming.

- Output Disable control – In this mode of operation, the GPIO pin will act as a control input. When the GPIO input is high, the selected output clock(s) will be disabled, then placed in high-impedance state. When the GPIO pin is low, the selected output clock(s) will be enabled and drive their outputs as configured. For information on output frequency and output levels, see System APLL and FOD Multiplexing and Output Stages. Selection of which output(s) are controlled by which GPIO(s) is configured via registers over the serial port or by OTP or EEPROM at reset. Each GPIO can be configured to control any or all outputs (or none). So all combinations can be set up from a single GPIO controlling all outputs, to all outputs responding to individual GPIO signals and any grouping in between.
- Single-ended Input Clock – In this mode of operation a single-ended input clock can be applied to certain GPIOs that map to specific input stages (see Input Stage for details, including which GPIOs map to which input references). This can be used if extra single-ended inputs are needed due to all input reference clock pins being taken-up by differential input references. This mode cannot be used if an input stage already has two single-ended input references from the CLKx/nCLKx input pins.
- Manual Clock Selection control – In this mode of operation, the GPIO pin acts as an input that will manually select between one of two inputs for a specific DPLL channel. The specific input references and the PLL channel must be preconfigured via registers. Assertion of the GPIO will select the higher priority input and de-assertion will select the lower priority input. For information on how to configure the input references for a PLL channel, see DPLL Input Clock Qualification and Selection.
- DCO Increment – In this mode of operation, the GPIO pin will act as an increment command input pin for a specific channel configured as a DCO. The rising edge of the GPIO pin will cause an increment function on the indicated DCO. The amount of the increment and the related DCO to increment must be previously configured via registers. For more information, see Increment / Decrement Registers and Pins.
- DCO Decrement – In this mode of operation, the GPIO pin will act as a decrement command input pin for a specific channel configured as a DCO. The rising edge of the GPIO pin will cause a decrement function on the indicated DCO. The amount of the decrement and the related DCO to decrement must be previously configured via registers. For more information, see Increment / Decrement Registers and Pins.
- Clock Disqualification Input – In this mode of operation, the GPIO pin will act as an active-high disqualification input for a preconfigured input and DPLL. This is intended to be connected to the LOS output of a PHY or other device. For more information, see DPLL Input Clock Qualification and Selection.

GPIO Pin Configuration

GPIO pins are all powered off a separate voltage supply that supports 1.5V, 1.8V, 2.5V, or 3.3V operation. An internal register bit must be set to indicate which voltage level is being used on the GPIO pins. This setting is a global one for all GPIOs.

In addition, each GPIO can be enabled or disabled under register control. If enabled and configured in an operating mode that makes it an output, the user can choose if the GPIO output will function as an open-drain output or a CMOS output. The open-drain output drives low but is pulled high by a pull-up resistor. There is a very weak pull-up internal to the 8A34002, but an external pull-up is strongly recommended. In CMOS mode, the output voltage will be driven actively both high and low as needed. Register control can also enable a pull-up (default) or pull-down.

Alarm Output Operation

There are many internal status and alarm conditions within the 8A34002 that can be monitored over the serial port by polling registers. Several of these can be directed to GPIO pins as indicated in General Purpose Input/Outputs (GPIOs). In addition, one of the GPIOs can be designated as an Alert (aggregated alarm) output signal called an Alert output.

The 8A34002 provides both a “live” and a “sticky” status for each potential alarm condition. A “live” bit shows the status of that alarm signal at the moment it is read over the serial port. A “sticky” bit will assert when an alarm condition changes state and will remain asserted until the user clears it by writing to the appropriate clear bit over the serial port. When a GPIO is configured to show the status of a specific alarm, it will show the “live” or sticky status of that alarm, depending on the specific alarm, where a high output on the GPIO indicates the alarm is present. For more information, see GPIO Modes. The GPIO can be programmed to invert the alarm if desired.

The Alert (aggregated alarm) output logic only uses the “sticky” status bit for alarms. This ensures when a software routine reads the 8A34002 there will be an indication of what caused the alarm in the first place. Note that there can be multiple sticky bits asserted. Table 17 shows the alarm conditions possible within the 8A34002. Note that the reference monitor, the DPLL, and the System DPLL blocks can generate the indicated alarms.

Table 17. Alarm Indications

Logic Bloc	Specific Alarm	Conditions for Live Alarm[a] to Assert	Conditions for Live Alarm[a] to Negate[b]
Reference Monitoring	Frequency Offset Limit Exceeded	See Frequency Offset Monitoring	See Frequency Offset Monitoring
	Loss-of-Signal	See Loss of Signal (LOS) Monitoring	See Loss of Signal (LOS) Monitoring
	Activity Alarm	See Activity Monitor	See Activity Monitor
Digital Phase Locked Loop (DPLL)[c][d]	Holdover	DPLL has entered / is in the Holdover state	DPLL no longer in Holdover state
	Locked	DPLL has entered / is in the Locked state and System APLL is in the Locked state	DPLL and/or System APLL no longer in the Locked state

[a] “Sticky” alarm bits are set whenever the associated live alarm changes state. So there will be a new “sticky” alarm on both assertion and negation of the appropriate live alarm indication.

[b] Only the “live” status will negate by itself. The “sticky” needs to be explicitly cleared by the user.

[c] For the Digital PLL, “sticky” alarms are raised when the state machine transitions into specific states and “live” status indicates that the Digital PLL is currently in a specific state. The user can read the current state of the Digital PLL state machine from status registers over the serial port.

[d] This includes the System DPLL, as well as all Digital PLLs.

For each alarm type in each logic block that can generate them, there is a “live” status, a “sticky” status, a “sticky” clear control and a series of control bits that indicate what effects the alarm will have. When the “live” status changes state, the “sticky” status will assert. If so configured via registers, that alarm may generate an external signal via GPIO. That signal may be an individual alarm output or an Alert (aggregated alarm). Once external software responds, it is expected to read the sticky status bits to determine the source(s) of the alarm and any other status information it may need to take appropriate action. The “sticky” clear control can be used to clear any or all of the bits that contributed to the alarm output being asserted.

In addition to the above controls and status, each potential alarming logic block has its own controls and status. Each of the reference monitors has a “sticky” status bit, a “sticky” clear bit and various control bits. Each of the DPLLs and the SysDPLL have a “sticky” status bit, a “sticky” clear bit, control bits and a PLL state status field. These functions behave as described in the previous paragraph. Note that both the individual alarm “sticky” status and the logic block “sticky” status must be cleared to fully remove the source of the alarm output. Individual “sticky” alarms should be cleared first so that all individual alarms associated with a logic block won’t cause a re-assertion of the block “sticky” alarm.

Note: Clearing of all sticky bits via the registers may not result in the Alarm output pin negating for up to 200µsec and so that GPIO should not be used as a direct input to a CPU's interrupt input or multiple interrupts may be generated within that CPU for a single alarm event.

There are also several configuration bits that act on the alarm output logic as a whole. There is a global alarm enable control that will enable or disable all alarm sources. This can be used during alarm service routines to prevent new alarms while that handler is executing in external software. The user can also designate a GPIO as an Alert (aggregated alarm) output and determine which individual alarms will be able to drive it. The GPIO can be programmed to invert the Alert (aggregated alarm) if desired.

Device Initial Configuration

During its reset sequence, the 8A34002 will load its initial configuration, enable internal regulators, establish and enable internal clocks, perform initial calibration of the Analog PLL, and lock it to the reference on the OSCI/OSCO pins. Depending on the initial configuration, it may also bring up Digital PLLs, lock to input references including any OCXO/TCXOs, and generate output clocks.

The following four mechanisms can be used to establish the initial configuration during the reset sequence:

- State of certain GPIO pins (see Table 18) at the rising edge of the nMR signal
- Configuration previously stored in One-Time Programmable memory
- Configuration stored in an external I^2C EEPROM
- Default values for internal registers

Each of these is discussed individually in the following sections and then integrated into the reset sequence.

Use of GPIO Pins at Reset

All of the device GPIO pins are sampled at the rising edge of the nMR (master reset) signal and some of them may be used in setting the initial configuration. Table 18 shows which pins are used to control what aspects of the initial configuration. All of these register settings can be overwritten later via serial port accesses. Note also that several GPIOs can be used as a JTAG port when in Test mode. For information, see JTAG Interface. If these GPIOs are being used as a JTAG interface, it is recommended that they not be used for any of the reset functions outlined below.

Table 18. GPIO Pin Usage at Start-Up

GPIO Number	Function	Internal Pull-up or Pull-down
9	0 = Main serial port uses SPI protocol 1 = Main serial port uses I^2C protocol	Pull-up
8	Must be high during reset active period	Pull-up
4 pins user selectable[a]	Identifies which stored configuration in OTP to use for initial configuration (has no effect with “-000” unprogrammed devices). See details just below this table.	Pull-up
1 pin user selectable[a]	Disables EEPROM accesses during start-up sequence. By default, no GPIO is used for this purpose, so the device will attempt to find an external EEPROM to check for additional start-up information by default. See details just below this table.	Pull-up
1 pin user selectable[a]	Provides pin control for I^2C slave serial port (for serial port selected by GPIO[9] as I^2C) default base address bit A2. Has no effect on serial port selected as SPI. By default no GPIO is used for this purpose, so the default I^2C slave port base address will have a 0 for bit A2. See details just below this table.	Pull-up

[a] Selection of this mode for a GPIO is performed using the Device Information block in the OTP memory, which is programmed by Renesas at the factory for dash codes that are non-zero. “-000” dash code devices are considered unprogrammed and so will have the default behavior indicated above.

Any of the available GPIOs can be used as the following:

- I^2C base address bit A2 – This is for the serial port when selected as I^2C during the start-up sequence using GPIO[9]. If no GPIOs are configured in this mode, bit A2 of the slave serial port base address will be zero. The value of the I^2C base address and the serial port configuration can be overwritten by SCSR configuration data or serial port accesses later in the start-up sequence. If more than one GPIO is programmed with this functionality, only the one with the highest index will be used (e.g., if both GPIO[5] and GPIO[7] are programmed to do this, only GPIO[7] would be used).
- EEPROM Access Disable control – A high input value on a GPIO programmed with this function prevents a device from attempting to read device update information or SCSR configuration data from an external I^2C EEPROM. This will speed up device reset time but prevent access to updated information that may be stored in EEPROM. If no GPIOs are configured in this mode then the device will attempt to locate an external EEPROM at the appropriate point in the start-up sequence. If multiple GPIOs are configured to perform this function then any one of them being active will disable EEPROM accesses, so it is recommended that no more than one GPIO be programmed for this function.
- Default Configuration Select control – If no GPIOs are selected then GPIO[3:0] will be assumed and the value on those pins at the rising edge of the nMR signal will be used to select which of the SCSR configurations in OTP memory is to be used. Note that since a GPIO is pulled-up by default, unless these pins are pulled or driven low during the reset period, this will select SCSR Configuration 15. If one or more GPIOs are selected for this function, then the value on those pins at the rising edge of nMR will be used to select the SCSR configuration to be loaded. The Device Information block of the OTP can be configured to select any of up to four GPIO pins to be used for this purpose if the default GPIOs are not convenient. The GPIOs chosen do not have to be sequential, but whichever ones are chosen, the one with the lowest index number will be the LSB and so on in order of the index until the GPIO with the highest index is the MSB. No GPIO that appears elsewhere in this table should be used for this purpose.
  For example, if GPIO[8], GPIO[6], GPIO[5] and GPIO[2] are used, GPIO[8] is the MSB, GPIO[6] is next most significant, GPIO[5] is next and GPIO[2] is the LSB.

  If less than four GPIO pins are selected, then the selected GPIOs will be used as the least-significant bits of a 4-bit selection value, with the upper bits set to zero. If more than four GPIOs are programmed for this function, then the GPIOs will form a larger bit-length word for selection of internal configuration.

Default Values for Registers

All registers are defined so that the default state (without any configuration data from OTP or EEPROM being loaded) will cause the device to power-up with none of the outputs enabled and all GPIO signals in General-Purpose Input mode. Users can then program any desired configuration data over the serial port once the reset sequence has completed.

One-Time Programmable (OTP) Memory

The 8A34002 contains a 32kbytes One-Time Programmable (OTP) memory block that is customer definable. The term “one-time programmable” refers to individual blocks within the memory structure. Different blocks can be programmed at different times, but each block can only be programmed once. The data structure within the OTP is designed to facilitate multiple updates and multiple configurations being stored, up to the limit of the physical memory space.

After reset of the 8A34002, all internal registers are reset to their default values, then OTP contents are loaded into the device’s internal registers. A Device Information block programmed by Renesas at Final Test will always be loaded. This provides information that is specific to the device, including product ID codes and revision information. In addition there are zero or more device configurations stored in the OTP by Renesas at the factory if a special dash-code part number is requested. Certain GPIO pins are sampled at the rising edge of the external nMR input signal. The state of those pins at that time will be used by the 8A34002 to determine which of up to 16 configurations stored in the OTP to load into the device registers. For information on how to select a configuration, see Use of GPIO Pins at Reset.

Storage of configuration data in OTP does not require having a value stored for every register in the device. Register default values are defined to ensure that most functions will be disabled or otherwise made as neutral as possible. This allows only features that are being used in any particular configuration (and their associated trigger registers as defined in the *8A3xxxx Family Programming Guide*) to need to be stored in OTP for that configuration. The intent of this is to minimize the size a configuration takes in OTP to allow more configurations to be stored there. For this reason, the exact number of configurations storable in OTP cannot be predetermined. There will be a minimum of two configurations and a maximum of 16 configuration capacity in the OTP.

Part numbers with -000 as the dash code number are considered “unprogrammed” parts, but will come with at least a Device Information block pre-programmed with Renesas-proprietary information, including parameters needed to successfully boot the device to the point where it can read its configuration data. One Device Update block may also be programmed if determined to be appropriate by Renesas.

Custom user configurations indicated with non-zero dash code part numbers will in addition have one or more SCSR Configuration sections pre-programmed as indicated in the datasheet addendum for that particular dash code part number.

Note that a programmed configuration, Device Information block, or Device Update block may be invalidated via the OTP programming interface, and if sufficient OTP space remains, a new one added to replace it. Note that this does not erase or remove the original data and the space it consumes. It just marks it to be ignored by the device. This allows for a limited ability to update a device in the field either from a device functional update or configuration data perspective. This is a purely software-driven process handled over the serial port. Please contact Renesas for support if this type of in-field upgrade / change is desired. Note that the ability to perform this type of in-field update is highly dependent on the size of the change versus the remaining space in OTP, so it will not be possible in all cases.

Configuration Data in OTP

Multiple configurations can be programmed into the internal One-Time Programmable memory. By using the GPIO pins at start-up as outlined in Use of GPIO Pins at Reset, one of those configurations can be chosen for use as the initial values in the device registers after reset. Register values can be changed at any time over the serial port, but any such changes are not stored in OTP and will be lost on reset or power-down.

The OTP is organized so that only configuration data that changes from the register default values needs to be stored. This saves OTP space and allows the potential for more configurations to be stored in the OTP.

If the indicated configuration in OTP has a checksum error, it will not be loaded and registers will be left at their default values.

Use of External I^2C EEPROM

The 8A34002 can search for additional configuration or device updates in an external I^2C EEPROM. As described in the Use of GPIO Pins at Reset, a GPIO can be configured to select whether or not this search will be performed during the reset sequence.

The remainder of this description assumes the EEPROM search is enabled.

The 8A34002 will use its I^2C Master Port to attempt to access an external I^2C EEPROM at base address 1010000 (binary) at an I^2C frequency of 1MHz. If there is no response, this will be repeated at base address 1010001 (binary) at 1MHz. This will repeat up to address 1010111 (binary) at 1MHz. If there still are no responses, the search will be repeated at 400kHz and then again at 100kHz. If no response is received after this entire sequence, the device will assume there is no EEPROM available. Any errors in the process will be reported in status registers.

Device Updates in External I^2C EEPROM

As indicated in Reset Sequence, if enabled, the 8A34002 will search for Device Update information in an external I^2C EEPROM. It will first identify all valid EEPROMs attached to the I^2C master port as described above. Each valid EEPROM will be checked for a valid Device Update Block header with valid checksum at address offset 0x0000 within the EEPROM. The first such valid block will be used as described in Reset Sequence.

Configuration Data in External I^2C EEPROM

As a final option for device configuration, the initial configuration can be read from an external I^2C EEPROM. Renesas’ Timing Commander GUI Software can generate the necessary EEPROM load information as an Intel HEX file for this purpose. The 8A34002 will search each EEPROM identified during the above search sequence for a valid configuration data block (valid header and checksum). The first valid block found will be loaded into internal registers after checksum validation. The search will terminate after the first valid block is found and loaded. This means that only a single valid configuration block can be stored via the EEPROM method.

When the device searches for an EEPROM configuration, it will check for a valid block at address offsets 0x0000 and 0xF000 within an EEPROM. If using this configuration method, see the warning in Step 5 – Search for Configuration in External EEPROM.

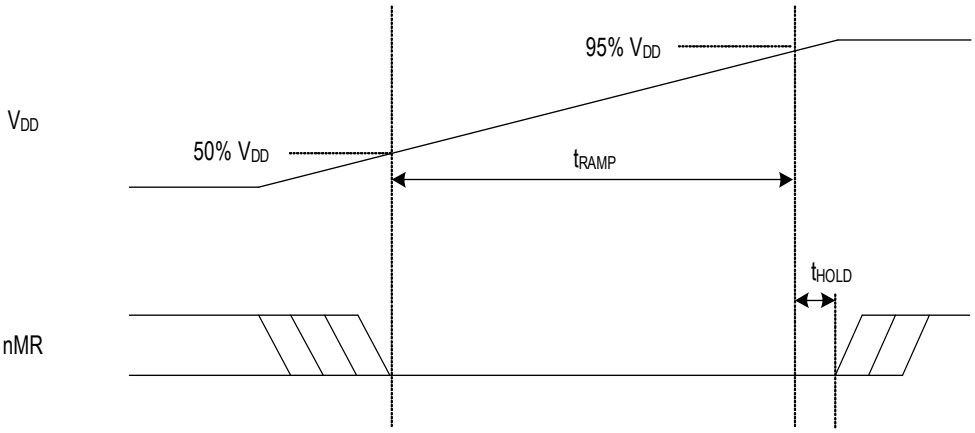
Reset Sequence

Unless otherwise specified, any referenced bits or registers throughout this section reside in the RESET_CTRL section of the register map, base address for which is C000h (offset 000h to 012h). For more information, see the *8A3xxxx Family Programming Guide*.

Figure 16 shows the relationship between the master reset signal (nMR) and the supply voltages for the 8A34002. There are no power sequencing requirements between the power rails, so V_{DD} in the diagram represents any of the supply voltages. To ensure there is no anomalous behavior from the device as it powers up, it is recommended that the nMR signal be asserted (low) before any voltage supply reaches the minimum voltage shown in the figure. nMR should remain asserted until a short hold time (t_{HOLD} ~10nsec) after all supply voltages reach the operating window of 95% of nominal voltage. nMR must be asserted or the device will not function correctly after power-up.

One additional consideration is that once minimum voltage is reached on all voltage supplies, internal regulators and voltage references will need the amount of start-up time specified in Table 37, “Regulators Ready.” If the time t_{RAMP} shown in Figure 16 is less than the voltage regulators’ start-up time indicated in Table 37, then release of nMR should be delayed.

Figure 16. Power-Up Reset Sequencing



In cases where the device is not powering up and just being reset, a low pulse on nMR of 20ns will be sufficient to reset the device.

The following reset sequence will start from the rising (negating) edge of the nMR (master reset) signal when nTEST is de-asserted (high).

Step 0 – Reset Sequence Starting Condition

Once power rails reach nominal values and the nMR signal has been asserted, the 8A34002 will be in the following state:

- All Qx / nQx outputs will be in a high-impedance state.
- All GPIO pins will be set to General-Purpose Inputs, so none will be driving the output.
- The serial port protocols are not set at this point in the reset sequence, so the ports will not respond.
- Device Information block loaded from internal OTP to configure what GPIOs will be used for what start-up functions in Step 1.
- The System APLL will be configured and calibrated based on frequency information in the Device Information block then locked to the reference clock on the OSCI input.

Step 1 – Negation of nMR (Rising Edge)

At the rising edge of the nMR signal, the state on the GPIO pins at that time is latched. After a short hold time (t_{HOLD} ~10nsec), the GPIOs can release their reset levels and assume their normal operation modes. The latched values will be used in later stages of the reset sequence.

Step 2 – Internally Set Default Conditions

An internal image of the device registers will be created in internal RAM with all registers set to their default values. This will not result in any changes to the GPIO or output clock signals from their Step 0 condition.

Based on the serial port protocol selection made via the GPIO pin in Step 1, serial port configuration will be completed as indicated by the GPIO input pin. If SPI mode is selected by the GPIO, the register default values will configure it to use 4-wire SPI mode.

Step 3 – Scan for Device Updates in EEPROM

If enabled to do so, the 8A34002 will check for device functional update information in any available EEPROMs (for information on how EEPROMs are searched for, see Use of External I^2C EEPROM). If such information is found, it will be loaded, the device functionality updated, and then the part will reinitialize to Step 0.

Step 4 – Read Configuration from OTP

Using the GPIO values latched in Step 1, the device will search the internal OTP memory for the indicated configuration number. If no such configuration is found or the configuration has an invalid checksum, the device will skip to Step 5. Any errors in this process will be reported. If loading from OTP was successful, which configuration number was loaded will be reported.

If the requested configuration is found and is valid, the device will load the registers indicated in the configuration data with the stored data values in the internal register image. Any register not included in the configuration data set will remain at its default value in the register image.

Note: Many register modules have explicitly defined trigger registers that when written will cause the other register settings in that module to take effect. Users must ensure that the configuration in OTP will cause a write to all applicable trigger registers, even if that register's contents would be all zero. Multi-byte register fields also require all bytes of the field to be written to ensure triggering. For indications of which trigger registers are associated with which other registers, see the *8A3xxxx Family Programming Guide*.

The contents of several of the registers will be used to guide the remainder of the reset sequence:

- If the System APLL feedback divider value was programmed in this step, perform System APLL calibration in parallel with remaining reset activities.
- Re-configure the serial ports to use I^2C or SPI protocols as indicated. For information, see I2C Slave Operation or SPI Operation.

Step 5 – Search for Configuration in External EEPROM

The 8A34002 will check for configuration information in any available EEPROM (for information on how EEPROMs are searched for, see Use of External I^2C EEPROM).

If a valid configuration data block is found, it will be read, its checksum validated and if that passes, loaded into the internal register image similarly to OTP configuration data described in Step 4. If the data found is not of the correct format or the data block fails a checksum comparison, it will be ignored. The search will continue through the EEPROM and on to the next EEPROM address until the complete range has been searched or a valid configuration block has been found and applied to the internal register image. Then the sequence will proceed to Step 6.

Note: Since OTP and EEPROM configuration data rarely consists of a full register image, reading of configuration data from OTP and then from a configuration block stored in EEPROM may result in internal registers being loaded with conflicting settings drawn partially from each of the configuration data sets being loaded. It is strongly recommended that a configuration block placed in EEPROM only be used when no valid configuration is being pointed to in OTP by GPIO signals (or there is no valid configuration in OTP at all). If multiple configurations are to be used then the user must ensure all registers are set to the desired values by the final configuration block to be loaded.

Step 6 - Load OTP Hotfix and Execute

If the 8A34002 OTP memory contains a hotfix, that information will be loaded into RAM and executed at this point.

Step 7 – Complete Configuration

The 8A34002 will complete the reset and initial configuration process at this point and begin normal operations. Completion steps include the following:

1. If configuration information was loaded in Step 4 or Step 5, recalibrate the System APLL and lock it to the reference clock on the OSCI input.
2. Enable serial port operation as configured.
3. Apply configuration settings from the internal register image to the actual registers and enable output clocks Qx / nQx and GPIOs as configured.
4. Begin operation on input reference monitors and PLL state machine alarms/status.
5. Enable alarm operation as configured.

Note that there are several scenarios in which the reset sequence will reach this point without retrieving any configuration data and with all registers in the default state. This may be intentional for users who wish to configure only via the serial port or the result of a problem in the loading of a configuration. Users can read appropriate status bits to determine what failures, if any, occurred during the reset sequence.

Accessing the Serial Ports After a Reset Sequence

After a reset sequence, there is a wait time (t_{wait}) after nMR de-assertion before the device serial ports can be accessed.

The required wait time depends on the EEPROM and OTP as follows:

- EEPROM load is disabled by OTP programming: t_{wait} = 15ms
- EEPROM load is enabled and no EEPROM is present: t_{wait} = 150ms
- EEPROM load is enabled and EEPROM is present: t_{wait} is given by the equation below:

t_{wait} (s) = { { { EEPROM_Payload_Size (bytes) / 256 } * 2342 + 2250 } / I2C_CLK_Rate (Hz) } * 1.15

EEPROM_Payload_Size is the sum of:

- Size of configuration in bytes
- Size of firmware (if any) in bytes

Clock Gating and Logic Power-Down Control

The 8A34002 can disable the clocks to many logic blocks inside the device. It also can turn off internal power regulators, disabling individual power domains within the part. Because of the potentially complex interactions of the logic blocks within the device, logic within the part will handle the decision-making of what will be powered-off, versus clock-gated, versus fully operational at any time. By default, the device will configure itself with functions in the lowest power-consuming state consistent with powering up the part and reading a user configuration. User configurations, whether stored in internal OTP, external EEPROM, or manually adjusted over the serial port, should make use of register bits to only turn on functions that are needed. Also if a function is no longer needed, register bits should be used to indicate it is no longer required. Internal logic will reduce its power-consumption state in reaction to these indicators to the greatest extent possible.

For more information on how to calculate power consumption for a particular configuration, consult Renesas' Timing Commander software for more precise results for a particular configuration.

Serial Port Functions

The serial ports are managed with bit fields in the SERIAL module. See Module: SERIAL in the *8A3xxxx Family Programming Guide*.

The 8A34002 supports two serial ports. One is a dedicated I^2C Master port used for loading configuration data at reset and the other is a configurable slave I^2C or SPI port that can be used at any time after the reset sequence is complete to monitor and/or configure the device.

The operation of the I^2C Master Port is described in I2C Master. The SCL_M and SDA_M pins are used for this port and can be connected to the same I^2C bus as the slave port if it is configured in I^2C mode.

For information on the operation of the master I^2C and slave I^2C or SPI ports, see the appropriate section below.

A slave serial port can be reconfigured at any time by accessing the appropriate registers within a single burst write. This includes configuration options with each protocol or switching between protocols (I^2C to SPI or vice versa). However, it is recommended that the full operating mode configuration, including page sizes for registers for a serial port be set in the initial configuration data read from OTP or external EEPROM (for information, see Device Initial Configuration).

Note on Signal Naming in the Remainder of the Serial Port Sections

The pin names indicated in the Pin Description table are meant to indicate the function of that signal when used in SPI mode and also the function when in I^2C mode. In the remainder of the Serial Port Functions descriptions, the SPI descriptions will refer to the signals by their function in the selected mode, as shown in Table 19.

Table 19. Serial Port Pin to Function Mapping

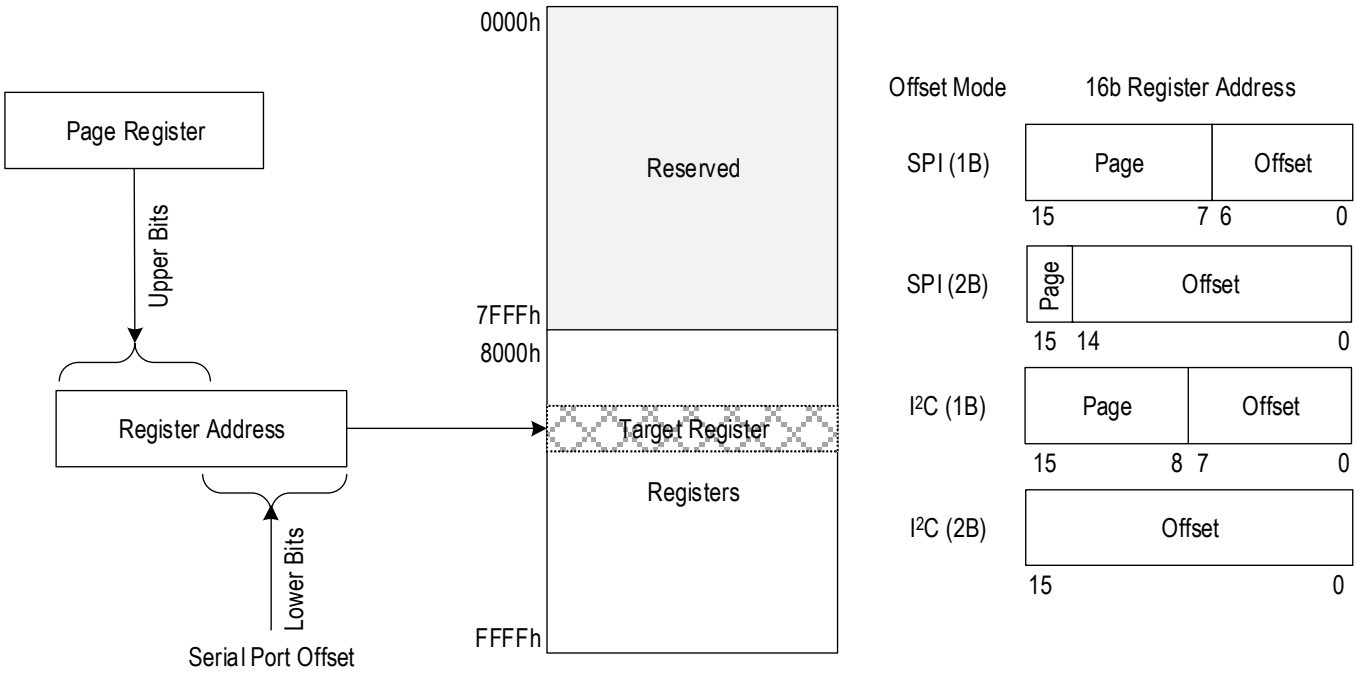
SPI Mode Signal Name	Function	I^2C Mode Signal Name	Function	Package Pin Name
SCLK	SPI Clock Input	SCLK	I^2C Clock Input	SCLK
CS	SPI Chip Select (active low)	A0	I^2C Slave Address Bit 0	CS_A0
SDI	SPI Data Input (unused in 3-wire mode)	A1	I^2C Slave Address Bit 1	SDI_A1
SDIO	SPI Data Out (4-wire mode) SPI Data In/Out (3-wire mode)	SDA	I^2C Data In/Out	SDIO

Addressing Registers within the 8A34002

The address space that is externally accessible within the 8A34002 is 64KB in size, and thus, needs 16 bits of address offset information to be provided during slave serial port accesses. Of that 64KB, only the upper 32KB contains user accessible registers.

The user may choose to operate the serial port providing the full offset address within each burst, or to operate in a paged mode where part of the address offset is provided in each transaction and another part comes from an internal page register in each serial port. Figure 17 shows how page register and offset bytes from each serial transaction interact to address a register within the 8A34002.

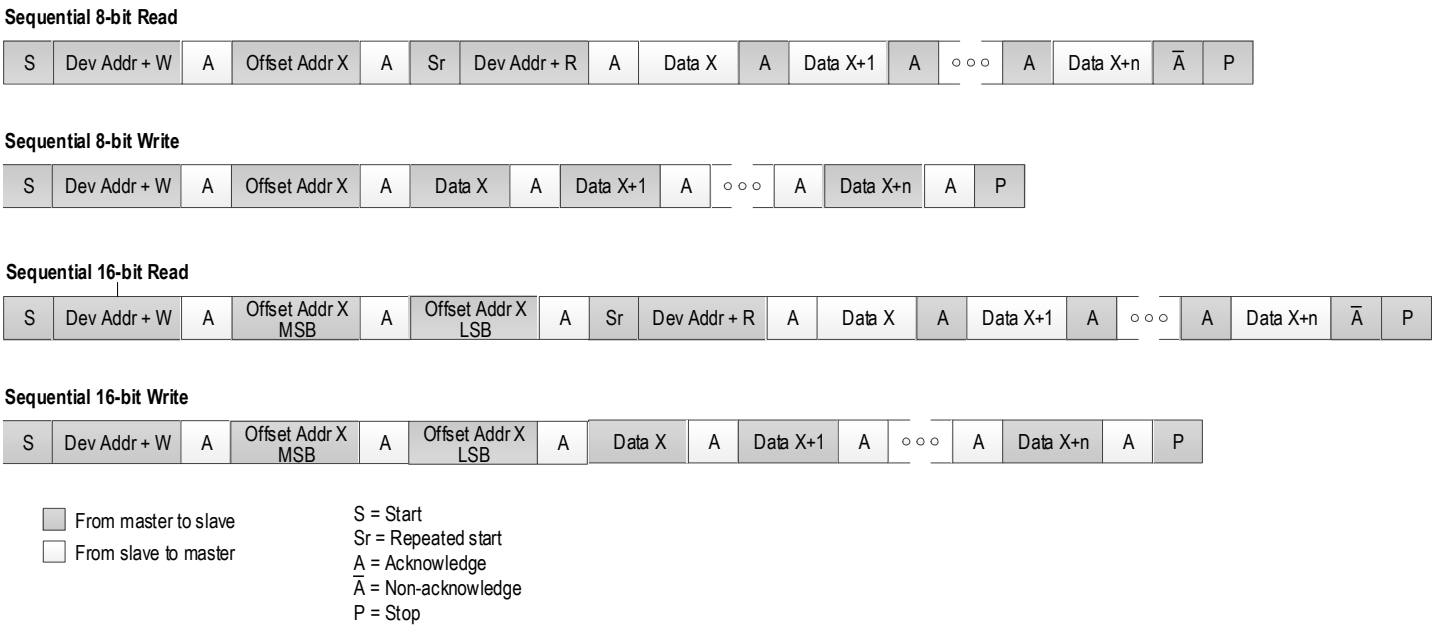
Figure 17. Register Addressing Modes via Serial Port



I²C Slave Operation

The I²C slave protocol of the 8A34002 complies with the I²C specification, version UM10204 Rev.6 – 4 April 2014. Figure 18 shows the sequence of states on the I²C SDA signal for the supported modes of operation.

Figure 18. I²C Slave Sequencing



The Dev Addr shown in the figure represents the base address of the 8A34002. This 7-bit value can be set in an internal register that can have a user-defined value loaded at reset from internal OTP memory or an external EEPROM. The default value if those methods are not used is 1011000b. Note that the levels on the A0 and A1 signals can be used to control Bit 0 and Bit 1, respectively, of this address. There is also an option to designate the reset state of a GPIO pin to set the default value of the A2 bit of the I^2C slave port base address (for information, see Use of GPIO Pins at Reset). In I^2C operation these inputs are expected to remain static. They have different functions when the part is in SPI mode. The resulting base address is the I^2C bus address that this device will respond to. See the SERIAL.SER0_I2C bit field.

When I^2C operation is selected for a slave serial port, the selection of 1-byte (1B) or 2-byte (2B) offset addressing must also be configured. These offsets are used in conjunction with the page register for each serial port to access registers internal to the device. Because the I^2C protocol already includes a read/write bit with the Dev Addr, all bits of the 1B or 2B offset field can be used to address internal registers. See the SERIAL.SER0 bit field.

- In 1B mode, the lower 8 bits of the register offset address come from the Offset Addr byte and the upper 8 bits come from the page register. The page register can be accessed at any time using an offset byte value of 0xFC. This 4-byte register must be written in a single-burst write transaction.
- In 2B mode, the full 16-bit register address can be obtained from the Offset Addr bytes, so the page register only needs to be set up once after reset via a 4-byte burst access at offset 0xFFFC.

Note: I^2C burst mode operation is required to ensure data integrity of multi-byte registers. When accessing a multi-byte register, all data bytes must be written or read in a single I^2C burst access. Bursts can be of greater length if desired, but must not extend beyond the end of the register page (Offset Addr 0xFF in 1B mode, no limit in 2B mode). An internal address pointer is incremented automatically as each data byte is written or read.

Figure 19 and Table 20 show the detailed timing on the interface. 100kHz, 400kHz, and 1MHz operation are supported.

I^2C 1-byte (1B) Addressing Examples

8A34002 I^2C 7-bit I^2C address is 0x5B with LSB = R/W

Example write “0x50” to register 0xCBE4:

```
B6* FC 00 CB 10 20      #Set Page Register, *I2C Address is left-shifted one bit.
B6 E4 50                #Write data 50 to CB E4
```

Example read from register 0xC024:

```
B6* FC 00 C0 10 20      #Set Page Register, *I2C Address is left-shifted one bit.
B6 24*                  #Set I2C pointer to 0xC024, *I2C instruction should use “No Stop”
B7 <read back data>     #Send address with Read bit set.
```

I^2C 2-byte (2B) Addressing

8A34002 I^2C 7-bit I^2C address is 0x5B with LSB = R/W

Example write “50” to register 0xCBE4:

```
B6* FF FD 00 10 20      #Set Page Register, *I2C Address is left-shifted one bit.
B6 CB E4 50             #Write data to CB E4
```

Example read from register 0xC024:

```
B6* FF FD 00 10 20      #Set Page Register (*I2C Address is left-shifted one bit.)
B6 C0 24*               #Set I2C pointer to 0xC024, *I2C instruction should use “No Stop”
B7 <read back data>     #Send address with Read bit set.
```

Figure 19. I²C Slave Timing Diagram

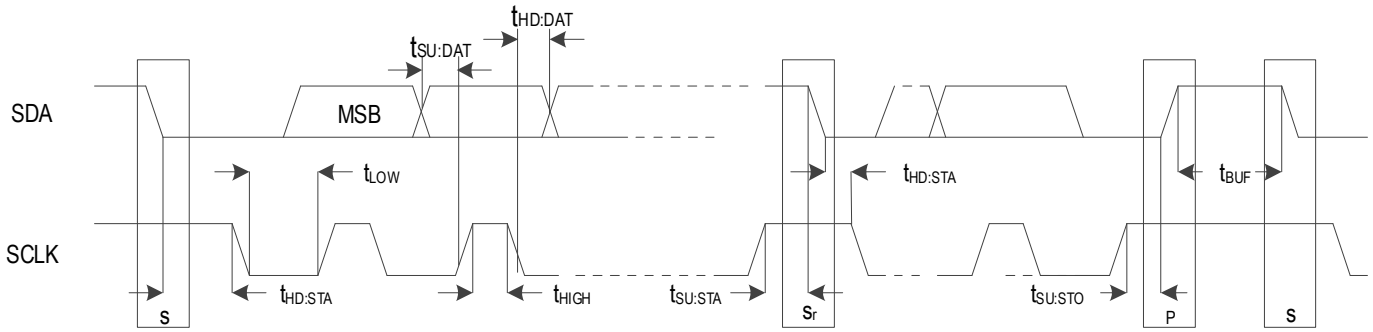


Table 20. I²C Slave Timing

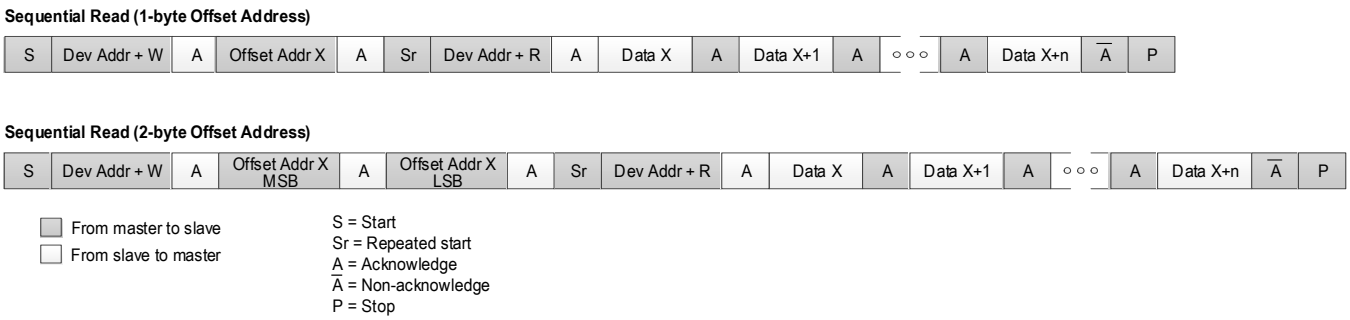
Parameter	Description	Minimum	Typical	Maximum	Units
f_{SCLK}	SCLK Operating Frequency			1	MHz
t_{LOW}	SCLK Pulse Width Low		130		ns
t_{HIGH}	SCLK Pulse Width High		9		ns
$t_{SU:STA}$	Start or Repeat Start Setup Time to SCLK		6		ns
$t_{HD:STA}$	Start or Repeat Start Hold Time from SCLK		18		ns
$t_{SU:DAT}$	Data Setup Time to SCLK rising edge		5		ns
$t_{HD:DAT}$	Data Hold Time from SCLK rising edge		0		ns
$t_{SU:STO}$	Stop Setup Time to SCLK		12		ns
t_{BUF}	Minimum Time from Stop to Next Start		0.5		ns

I²C Master

The 8A34002 can load its register configuration from an external I²C EEPROM during its reset sequence. For information on what accesses occur under what conditions, see Reset Sequence.

As needed during the reset sequence, the 8A34002 will arbitrate for the I²C bus and attempt to access an external I²C EEPROM using the access sequence shown in Figure 20. The I²C master protocol of the 8A34002 complies with the I²C specification, version UM10204 Rev.6 – 4 April 2014. As displayed in the figure, the I²C master port can be configured to support I²C EEPROMs with either 1-byte or 2-byte offset addressing. The I²C master logic will negotiate with any EEPROMs found to use the highest speed of 1MHz, 400kHz, or 100kHz. See the SERIAL.I2CM bit field.

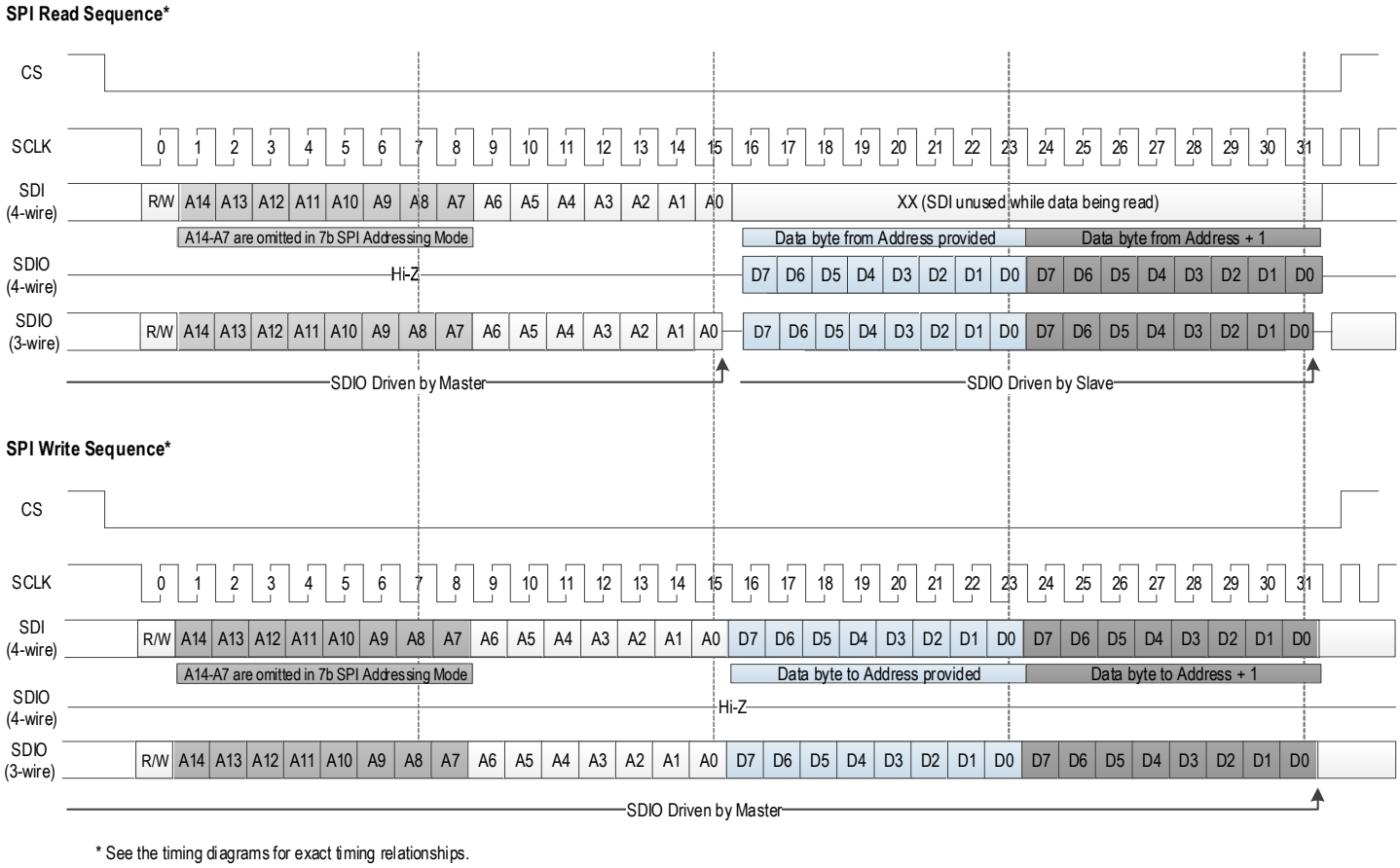
Figure 20. I²C Master Sequencing



SPI Operation

The 8A34002 supports SPI operation as a selectable protocol on the serial port. The port can be configured for either 3-wire or 4-wire operation. In 4-wire mode, there are separate data in (to the 8A34002) and data out signals (SDI and SDIO respectively). In 3-wire mode, the SDIO signal is used as a single, bidirectional data signal. Figure 21 shows the sequencing of address and data on the serial port in both 3-wire and 4-wire SPI mode. 4-wire SPI mode is the default. The R/W bit is high for Read Cycles and low for Write Cycles. See the SERIAL.SER0_SPI bit field.

Figure 21. SPI Sequencing



* See the timing diagrams for exact timing relationships.

A serial port can be configured for the following settings. These settings can come from register defaults, or from an internal OTP or external EEPROM configuration loaded at reset:

- 1-byte (1B) or 2-byte (2B) offset addressing (see Figure 17). See the SERIAL.SER0 bit field.
  - In 1B operation, the 16-bit register address is formed by using the 7 bits of address supplied in the SPI access and taking the upper 9 bits from the page register. The page register is accessed using an Offset Address of 0x7C with a 4-byte burst access.
  - In 2B operation, the 16-bit register address is formed by using the 15 bits of address supplied in the SPI access and taking the upper 1-bit from the page register. Note that this bit will always be 1 for register accesses, so the page register only needs to be set once in 2B operation. The page register can be accessed using a 3-byte burst access Offset Address of 0x7FFD. It should be accessed in a single burst write transaction to set it.
- Data sampling on falling or rising edge of SCLK
- Output (read) data positioning relative to active SCLK edge
- 4-wire (SCLK, CS, SDIO, SDI) or 3-wire (SCLK, CS, SDIO) operation
- In 3-wire mode, SDIO is a bi-directional data pin.
- Output signal protocol compatibility / drive strength and termination voltage

Note: SPI burst mode operation is required to ensure data integrity of multi-byte registers. When accessing a multi-byte register, all data bytes must be written or read in a single SPI burst access. Bursts can be of greater length if desired, but must not extend beyond the end of the register page. An internal address pointer is incremented automatically as each data byte is written or read.

SPI 1-byte (1B) Addressing Example

Example write to “50” to register 0xCBE4

```
7C 80 CB 10 20          #Set Page register
64* 50                  #*MSB is 0 for write transactions
```

Example read from 0xC024:

```
7C 00 C0 10 20          #Set Page register
A4* 00                  #*MSB is set, so this is a read command
```

SPI 2-byte (2B) Addressing Example

Example write to “50” to register 0xCBE4

```
7F FD 80 10 20          #Set Page register
4B E4* 50               #*MSB is 0 for write transactions
```

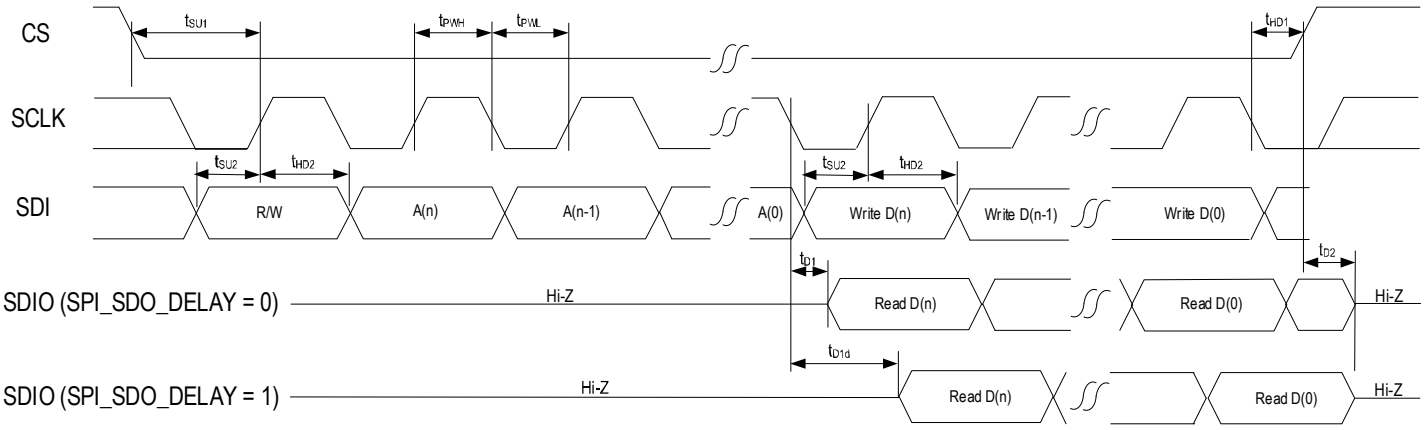
Example read from 0xC024:

```
7F FD 80 10 20          #Set Page register
C0* 24 00               #*MSB is set, so this is a read command
```

SPI timing is shown in Figure 22 and Table 21.

Figure 22. SPI Timing Diagrams

SPI_CLOCK_SELECTION = 0



SPI_CLOCK_SELECTION = 1

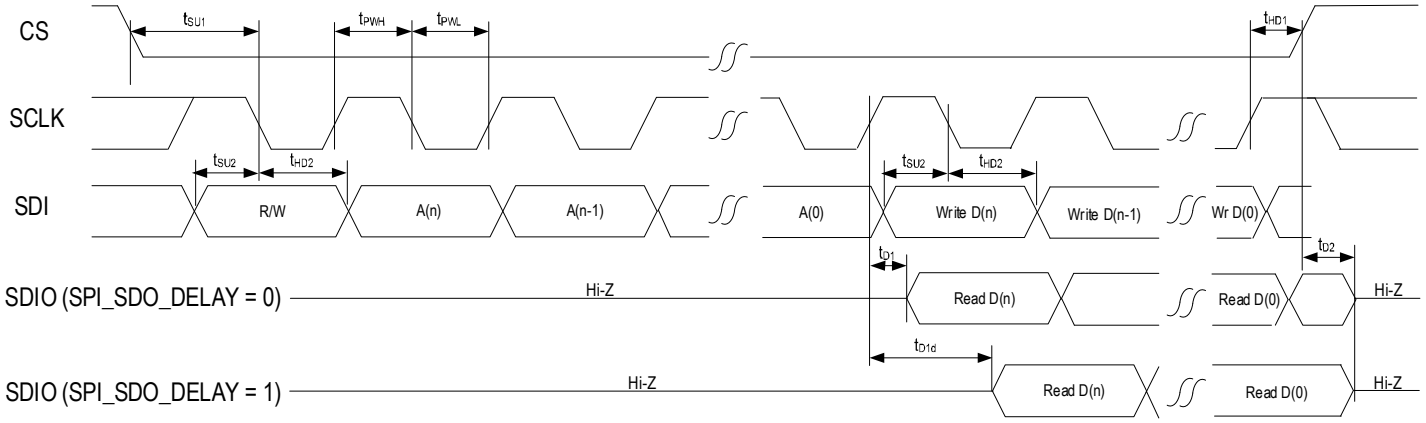


Table 21. SPI Timing

Parameter	Description		Minimum	Typical	Maximum	Units
f_{MAX}	Maximum operating frequency for serial port				50	MHz
t_{PWH}	SCLK Pulse Width High		9			ns
t_{PWL}	SCLK Pulse Width Low		9			ns
t_{SU1}	CS Setup Time to SCLK rising or falling edge		4.2			ns
t_{HD1}	CS Hold Time from SCLK rising or falling edge		0			ns
t_{SU2}	SDIO Setup Time to SCLK rising or falling edge		0			ns
t_{HD2}	SDIO Hold Time from SCLK rising or falling edge		0.6			ns
t_{D1}[a]	Read Data Valid Time from SCLK rising or falling edge with no data delay added	V_{DDS}[b] = 3.3V			7.2	ns
		V_{DDS} = 2.5V			7.3	ns
		V_{DDS} = 1.8V			8.6	ns

Table 21. SPI Timing (Cont.)

Parameter	Description		Minimum	Typical	Maximum	Units
t_{D1d}[a]	Read Data Valid Time from SCLK rising or falling edge including half period of SCLK delay added to data timing[c]	V_{DDS}[b] = 3.3V		6 + half SCLK period		ns
		V_{DDS} = 2.5V		6 + half SCLK period		ns
		V_{DDS} = 1.8V		6 + half SCLK period		ns
t_{D2}	SDIO Read Data Hi-Z Time from CS High[d]			10		ns

[a] Measurement performed approximately 1cm away from device pad. Observing at a greater distance on a heavily loaded trace may show slower edge rates and longer delays. This is highly dependent on PCB loading.

[b] V_{DDS} refers to the power supply for the serial ports pins (see Pin Description and Pin Characteristic Tables).

[c] Adding the extra half period of delay is a register programming option to emulate read data being clocked out on the opposite edge of the SCLK to the write data.

[d] This is the time until the 8A34002 releases the signal. Rise time to any specific voltage is dependent on pull-up resistor strength and PCB trace loading.

JTAG Interface

The 8A34002 provides a JTAG interface that can be used in non-operational situations with the device when nTEST control pin is held low. The JTAG interface is compliant with IEEE-1149.1 (2001) and supports the IDCODE, BYPASS, EXTEST, SAMPLE, PRELOAD, HIGHZ, and CLAMP instructions.

For information on the value the IDCODE instruction will return for the 8A34002, see the “Product Identification” table located at the end of this document.

JTAG port signals share five pins with GPIO functions as outlined in Table 22. Assertion of the nTEST input (active low) will place those pins in JTAG mode and the device in non-operational mode. The nMR signal should be asserted when nTEST is negated to ensure the device resumes operational mode in a clean state.

Table 22. JTAG Signal Mapping

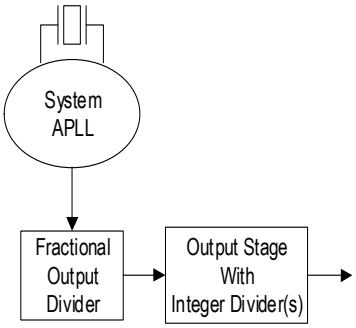
Function with nTEST Active (Low)	Function when nTEST Inactive (High)
TCK	GPIO[0]
TMS	GPIO[1]
TDI	GPIO[2]
TDO	GPIO[3]
TRSTn	GPIO[4]

Basic Operating Modes (Synthesizer / Clock Generator / Jitter Attenuator)

Free-Running Synthesizer Operation

Any DPLL channel can be used in a free-running synthesizer configuration independently of any of the other channels in the part. When configured as a free-running synthesizer, the blocks shown in Figure 23 are used. An external crystal is used as a reference by the System APLL which generates a high-frequency, low phase-noise signal. For information on System APLL configuration, see System APLL.

Figure 23. Free-Running Synthesizer Operation

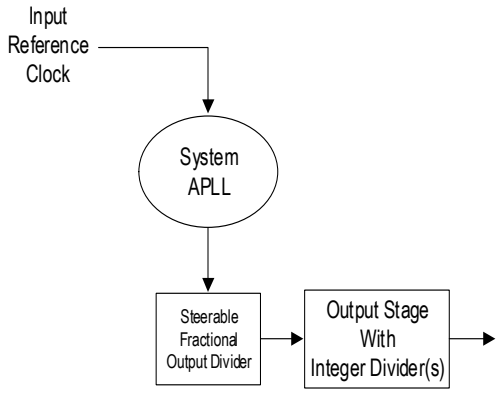


The high-frequency output signal from the System APLL is divided-down by the Fractional Output Divider block to a frequency from 500MHz to 1GHz (for information, see System APLL). The output frequency is unrelated to the System APLL frequency since fractional division is used. Note that in synthesizer mode, the Fractional Output Divider block is not steerable, so it is just performing the divide function. That signal is in turn fed to the output stage, where it is further divided by the integer divider(s) and provided to the output in the selected output format. For more information, see FOD Multiplexing and Output Stages.

Clock Generator Operation

Any DPLL channel can be used in a clock generator configuration. When configured as a clock generator as shown in Figure 24, the external crystal input (OSCI) is over-driven (for more information, see Overdriving the XTAL Interface) by an external clock signal which is used as a reference by the System APLL. The System APLL generates a high-frequency, low phase-noise signal from this reference. Note that because the System APLL is shared by all channels, this mode is not truly independent for all channels. Each channel can generate unrelated output frequencies via the Fractional Divider. Otherwise, this mode of operation functions the same as the free-running synthesizer operation in the previous section.

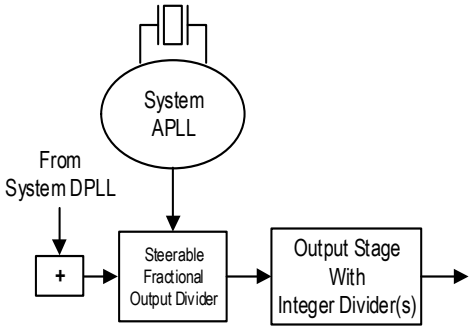
Figure 24. Clock Generator Operation



Synthesizer Disciplined with Oscillator Operation

If an external oscillator, such as an XO, TCXO or OCXO is used, any channel may have its output frequency disciplined by the oscillator for stability and/or close-in phase noise improvement reasons. When configured as shown in Figure 25, the Fractional Output Divider is behaving as indicated in the description, but in this case the System DPLL is locked to the oscillator and providing a steering signal to the Fractional Output Divider. Note that the oscillator may be connected to the dedicated oscillator input pin (XO_DPLL) or to any of the input reference clocks (CLKx / nCLKx). Please refer to the Digital Phase Locked Loop (DPLL) section for details.

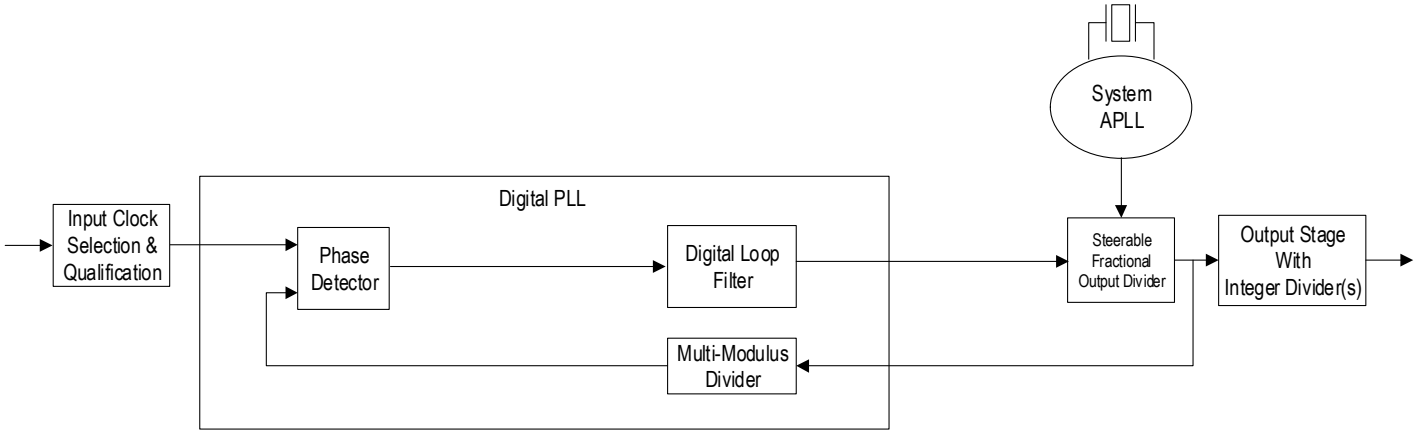
Figure 25. Synthesizer Disciplined with Oscillator Operation



Jitter Attenuator Operation

Any DPLL channel can be operated as a Jitter Attenuator independently from any other channel as shown in Figure 26.

Figure 26. Jitter Attenuator Operation



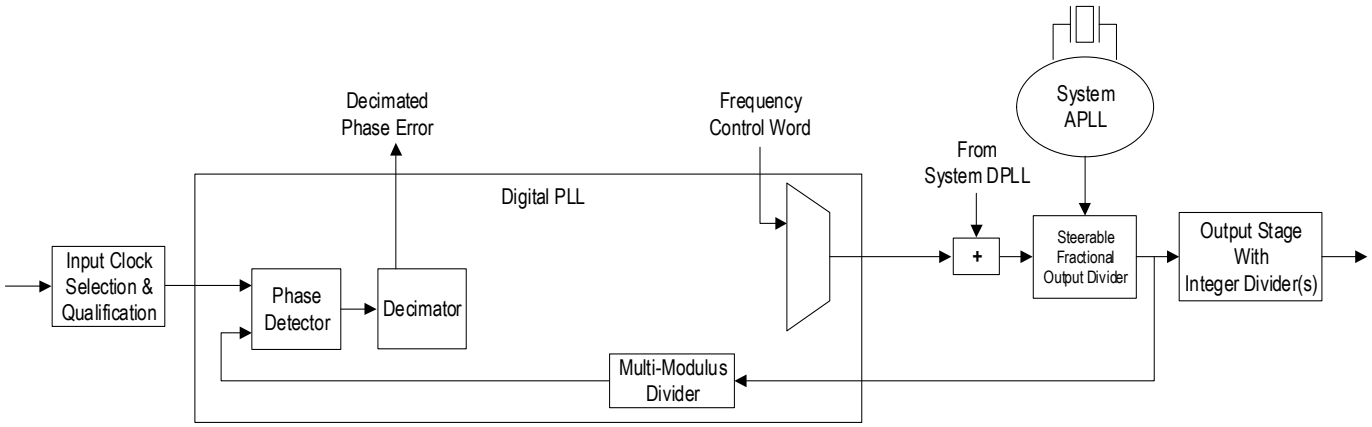
The high-frequency output signal from the System APLL is divided-down by the FOD block to a frequency from 500MHz to 1GHz. The output frequency is unrelated to the System APLL frequency since fractional division is used (for information, see System APLL). That signal is in turn fed to the output stage, where it is further divided by the integer divider(s) and provided to the output in the selected output format. For more information, see FOD Multiplexing and Output Stages.

The DPLL is locked to the input clock, and therefore there is a digital control signal from the Digital PLL (DPLL) block being used to steer the FOD. This signal will adjust the frequency of the FOD to track the input reference signal the DPLL is locked to. So the steerable FOD is acting as a Digitally Controlled Oscillator (DCO) in this mode. The DPLL logic supports several options on how the phase of the output reacts to changes in which input reference is selected, as well as supporting holdover operation if all relevant inputs are lost. For information on how the input reference clock is selected, see DPLL Input Clock Qualification and Selection; for information on DPLL options and operation, see Digital Phase Locked Loop (DPLL).

Jitter Attenuator Operation with External Digital Loop Filter

For some applications, it may be preferable to use an external digital loop filter implemented in software to control a DPLL channel. This function is supported as shown in Figure 27.

Figure 27. Jitter Attenuator Operation with External Digital Loop Filter



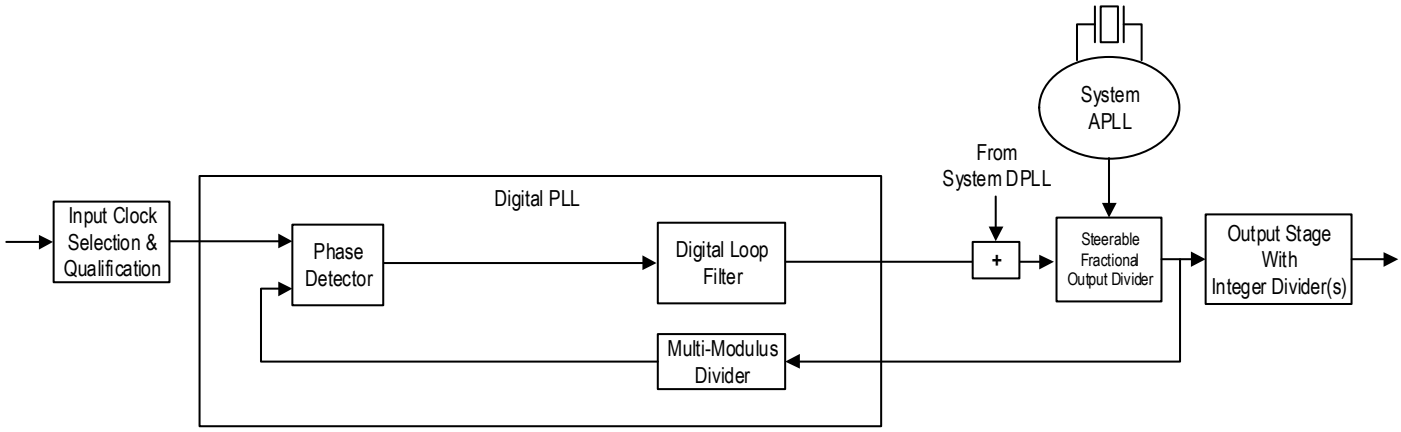
In this case, the phase error measured by the DPLL's phase detector is digitized and decimated to a user selected update rate and provided via registers for use by an external digital filter (for information, see Digital Phase Locked Loop (DPLL)). That data is read from the 8A34002 and provided to the digital filter algorithm. That algorithm then generates a Frequency Control Word (FCW) and writes that back into 8A34002's registers. The FCW will provide direct control of the steerable FOD.

Handling of input reference switchover and holdover operation is under control of the external filter algorithm in this case. Necessary control and status signals to handle these cases in external logic can be provided by proper configuration of the GPIO pins, as described in the General Purpose Input/Outputs (GPIOs).

Jitter Attenuator Disciplined with Oscillator Operation

Similarly to the Synthesizer mode being disciplined by an external oscillator, a jitter attenuator may also be similarly disciplined as shown in Figure 28. This provides stability for the DPLL channel when in holdover. Also, when the jitter attenuator is locked, the oscillator may be used to enhance close-in phase noise performance.

Figure 28. Jitter Attenuator Disciplined with Oscillator Operation



Digitally-Controlled Oscillator Operation via External Control

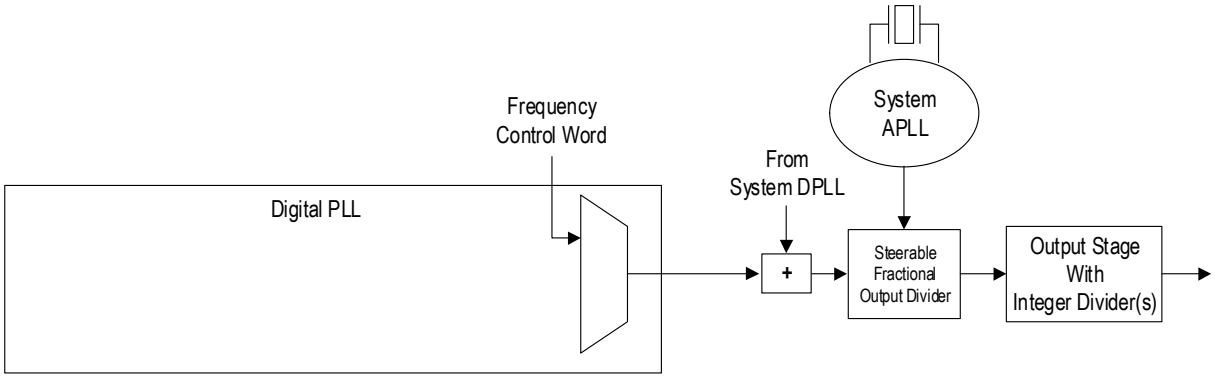
Any DPLL channel can be operated as an externally-controlled DCO independently from any other channel. There are several different control methods that can be used depending on the application needs. Each is described individually in the following sub-sections. Phase and/or frequency updates will be calculated using external methods and written into the 8A34002 over the serial port.

Write-Frequency Mode

For a DPLL channel in this mode a Frequency Control Word (FCW) is used to adjust the frequency output of the DCO (by steering the FOD) and the phase detector and loop filter are essentially bypassed. All the filtering is done by an external device and the frequency offset written into the Write Frequency Configuration register is passed on directly to the output clocks, as displayed in Figure 29. When applied, the FCW will not cause any missing pulses or glitches in the output clock, although a large frequency jump may cause issues with devices receiving this clock. The output will remain at this frequency until a new FCW is written.

If supported by the device, Combo Mode can be used to add additional offsets to the write frequency offset.

Figure 29. External DCO Control via Frequency Control Word



The FCW is a 42-bit 2's-complement value. The FCW has a granularity of 1.11×10^{-10}ppm and a full range of +244.20ppm to -244.08ppm of the nominal DCO frequency. A positive value will increase the output frequency and a negative one will decrease the output frequency. The formula for calculation of the FCW from the fractional frequency offset (FFO) is:

$$FCW = \left(1 - \frac{1}{\left(1 + \frac{FFO}{10^6} \right)} \right) \times 2^{53}$$

Where,
FFO = Fractional Frequency Offset, in ppm
FCW = Frequency Control Word (Positive or Negative Integer)

The value resulting from the above calculation must be converted to a 42-bit 2's complement value and then sign-extended to 48 bits to be written into the register.

Write frequency mode can be used to make phase changes on the output. For fine resolution, phase changes are done by controlling the DCO's frequency. Coarse phase adjustments should be done by snap-alignment method by using the Phase Offset registers (for information, see Output Coarse Phase Adjustment). Using the Phase Offset registers is referred to as the snap-alignment method since the output will snap directly to that new phase rather than moving smoothly to it over time. The snap-alignment method provides fast coarse phase alignments, and therefore, it should be used to bring the phase close to the desired value and then use the DCO in write frequency mode to fine tune it. Since write frequency mode is changing the frequency, the phase will move smoothly over time without any jumps.

Increment / Decrement Registers and Pins

The DCO frequency update can also be done by applying a preset frequency offset value to be added or to be subtracted from a cumulative FCW value. The cumulative FCW value functions as described in the previous section.

One or more GPIO pins can be configured to perform the increment or decrement frequency offset function on a specific DCO. For information on how to configure the GPIOs, see General Purpose Input/Outputs (GPIOs).

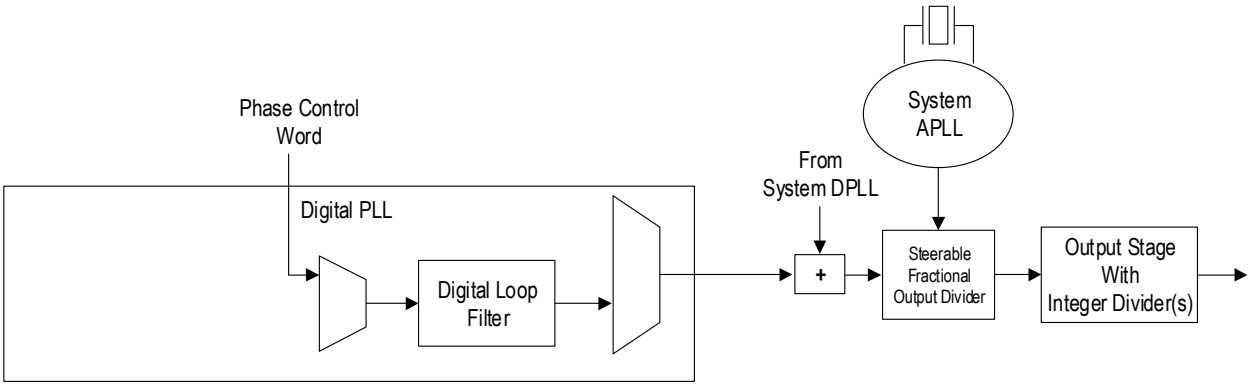
Write-Phase Mode

In this mode for a DPLL channel, the Phase Control Word (PCW) is written by the external control logic over the serial port to directly control the DCO phase with hardware controlled bandwidth and phase slope limiting (see Figure 30). In this mode, the DPLL loop bandwidth and the phase slope limiting are programmable and will affect the output phase as it is adjusted.

The PCW applied to the Digital Loop Filter is equivalent to applying a phase error measured by the on-chip Phase / Frequency Detector to the Digital Loop Filter when the DPLL is operating in closed loop. The update rate needs to be at least 60 times the loop filter bandwidth. As an example, for 0.1Hz per G.8273.2, the update should be greater than 6Hz; but for 17Hz the update should be greater than 1000Hz.The rate of adjustment of phase on the DCO output is controlled by Digital Loop Filter settings. For information on configuring related DPLL parameters such as loop bandwidth and phase slope limiting, see Digital Phase Locked Loop (DPLL). This method allows for a better control of the output clock since all parameters are controlled in hardware. This change will not cause any missing pulses or glitches in the output clock. Also, because the output frequency is changed only at a rate determined by the loop filter, this should not cause any issues, if properly configured, with devices receiving this clock.

Note that the PCW must be reduced over time or the DPLL will continue to adjust the DCO frequency to remove the “phase error”. This can be adjusted by external software.

Figure 30. External DCO Control via Phase Control Word



To assist in the above, there is an optional timer associated with the PCW. This allows a phase control word to be applied for a limited period of time after which it will be automatically reset to zero or placed into holdover by the 8A34002, and therefore, it will avoid the DCO continuing to apply the phase adjustment indefinitely until it reaches its tuning range limits. The timer value is a 16-bit integer that has a granularity of 1 millisecond and a full range of up to 65.535 seconds.

The PCW is a 32-bit 2’s-complement value. The resolution of the PCW is 50ps and the range is ±107.3741824ms. Writing a positive value will result in the output frequency getting faster. This will shorten the clock periods, moving the clock edges to the left as seen on an oscilloscope. A negative value will slow the output frequency.

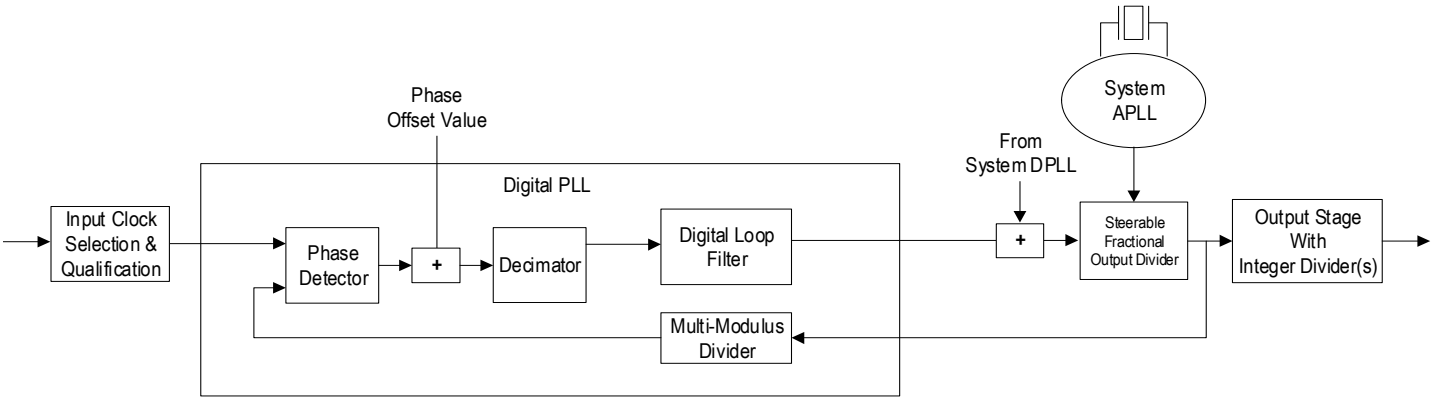
Adjusting Phase while in Closed Loop Operation

There may be usage scenarios that require adding a phase offset from an external software-controlled process to an output clock that is locked to an input clock. That function can also be supported as shown in Figure 31. In this mode, the amount of phase offset needed consists of two components. The first is dependent on which input the DPLL is locked to. So a phase offset register is provided for each input to allow individual offsets to be specified per-input.

The second part of the phase offset configuration is for each DPLL. There is a register for each DPLL that allows for another offset value to be specified that is independent of which input is active. The actual Phase Offset Value applied will be calculated by the 8A34002 using the per-input phase offset value for the currently active input summed with the phase offset value for the DPLL channel. During input reference switching, this value will be automatically recalculated at any switchover and applied as shown. Note that if an input is used on multiple DPLL channels, it may not be possible to maintain unique values per-input-per-DPLL. The calculated offset value is then summed with the measured phase error for that channel (phase difference between input reference and feedback value) to drive the DPLL to the desired phase.

The Phase Offset Value applied to the Digital Loop Filter is equivalent to applying a phase error measured by the on-chip Phase / Frequency Detector to the Digital Loop Filter when the DPLL is operating in closed loop.

Figure 31. Phase Control in Closed Loop Operation

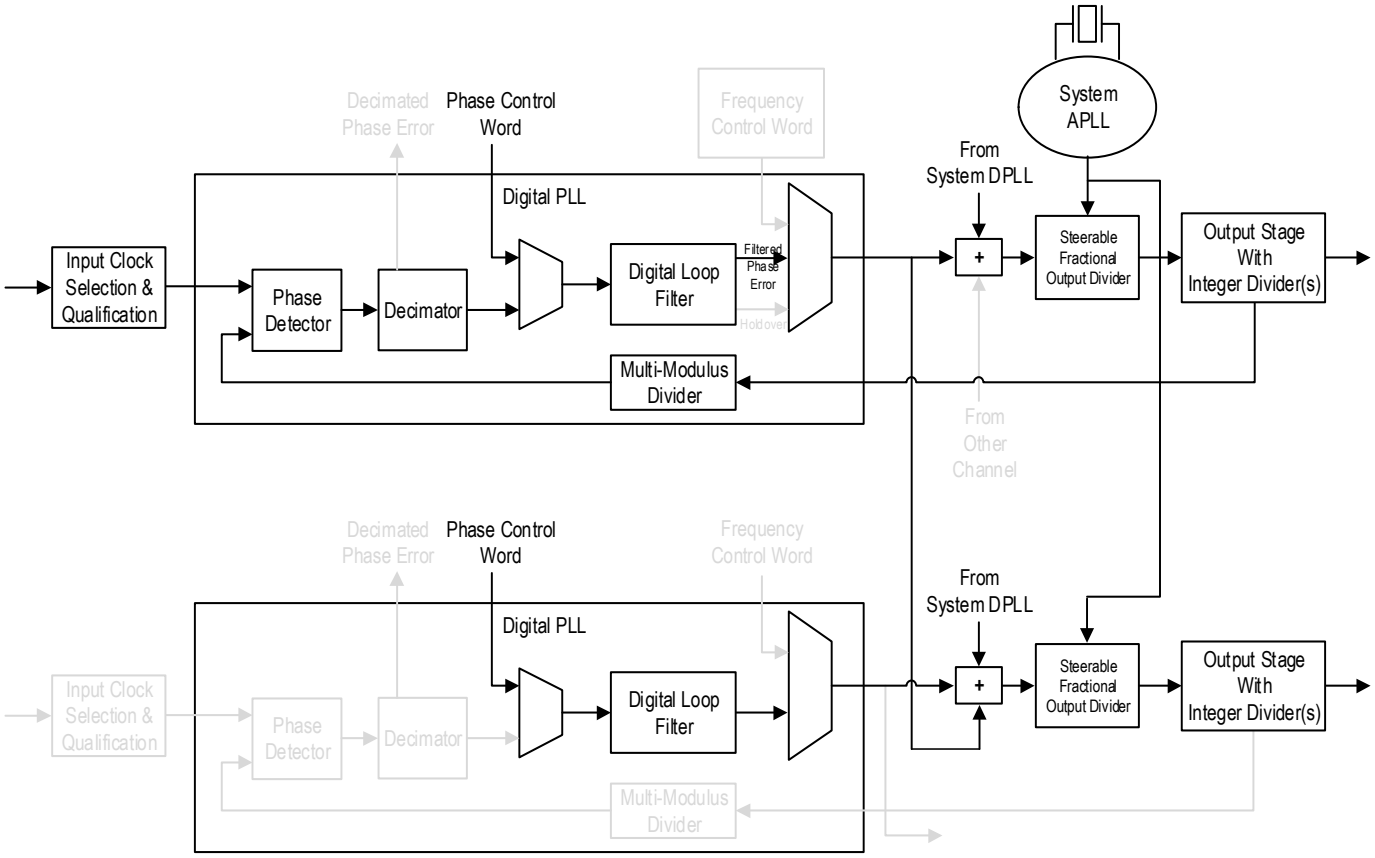


Combo Mode

The combo mode is shown in the following figure. In this mode, up to three channels are used, one of which is usually a DPLL channel. Each receiving DPLL or Satellite channel can be sourced from up to two other DPLLs including the System DPLL. In this mode, one DPLL is locked to an input reference clock, such as a Synchronous Ethernet clock, and can generate output clocks of different frequencies that track the Synchronous Ethernet input reference clock. The second channel is used as a DCO and it will be controlled externally, as an example by an IEEE 1588 clock recovery servo algorithm running in an external processor, or just track the first channel directly. This will not cause any missing pulses or glitches in the output clocks from either channel since all frequency changes are limited by at least one loop filter.

The physical layer clock and its output clock will act as the local oscillator for the DCO, and therefore the DCO can rely on a very stable clock. This is done all inside the device, no need to route the clocks externally. It is also important to be able to control phase transients in the SyncE clock. This can be done by either using an internal filter that will filter the SyncE transients, or by suppressing the SyncE based on SSM clock quality level.

Figure 32. Combo Mode



In this example, the IEEE 1588 timestamps are used to calculate the phase offset between the IEEE 1588 master's 1PPS pulse and the IEEE 1588 slave's 1PPS pulse and then align the two pulses by moving the slave's 1PPS pulse in phase. The slave must be able to move the 1PPS pulse by ±0.5s, and the 8A34002 provides this capability.

Satellite Channel

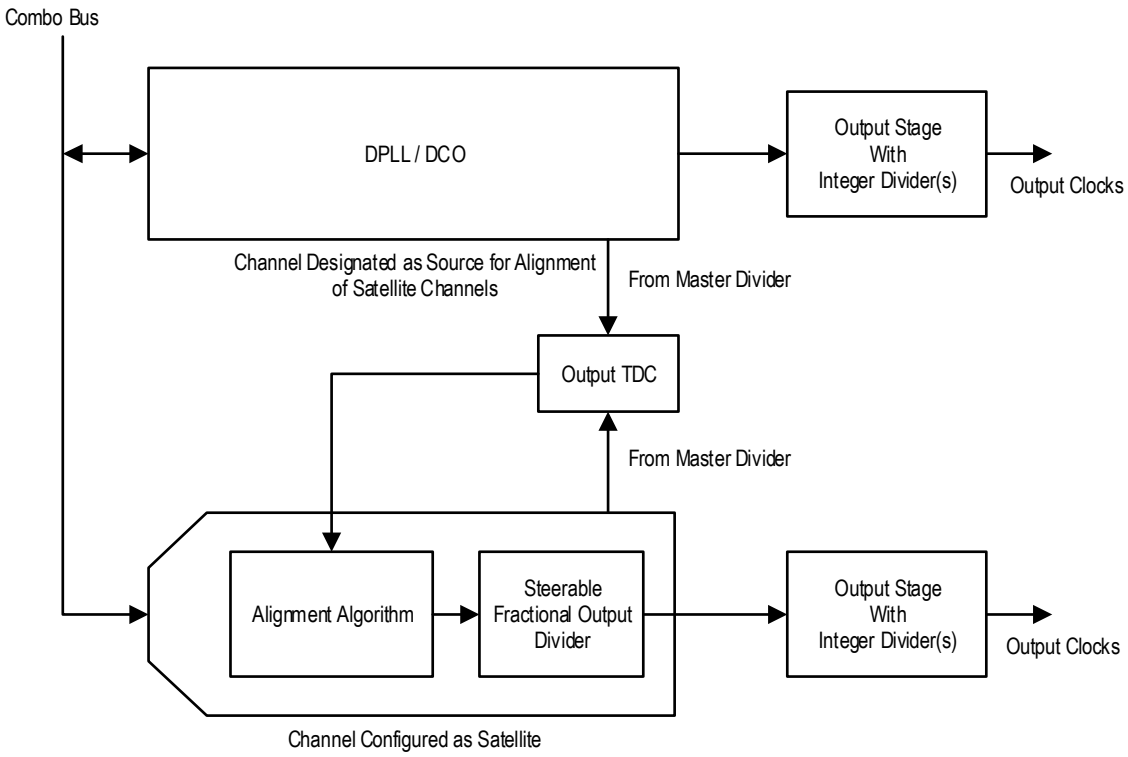
One or more satellite channels can be associated with a source DPLL or DCO channel to increase the number of independently programmable FODs and output stages available to the source channel. Except for the System DPLL, any channel can be designated a source channel or be configured as a satellite channel.

Figure 33 shows a DPLL/DCO channel designated as the source for an associated satellite channel. The satellite tracks the fractional frequency offset of the source by applying the same frequency steering information used by the source channel. The frequency information is provided to the satellite via the Combo Bus. An internal alignment algorithm compares the phases of the source and satellite channels using an output Time to Digital Converter (TDC) and it controls the satellite to ensure tight phase alignment.

For satellite channel applications, the source and satellite master divider output clocks must be of the same frequency to enable phase comparison by the output TDC. In addition, the master divider output frequency must be a common factor of the frequencies of all output clocks that are to be aligned.

For master divider output frequencies lower than 1MHz, GLOBAL_SYNC_EN must be enabled for the source and satellite channels, this will prevent long initial alignment times. The initial phase difference between the source and satellite channel can be up to half the period of the master divider output clock. Initial phase differences larger than 500ns can require significant time to eliminate. After initial alignment is achieved, then GLOBAL_SYNC_EN can be disabled if desired.

Figure 33. Satellite Channel and Source Channel



For more information about configuring satellite channels, please see the Renesas application note titled *Auto-Alignment of Outputs*.

Time-of-Day (ToD) Operation

Unless otherwise specified, any referenced bits or registers throughout this section reside in the TOD_X section of the register map, where *X* ranges from 0 to 3. Base address for TOD_0 is CBCCh (offset 000h). For more information about other TOD registers, see the *8A3xxxx Family Programming Guide.*

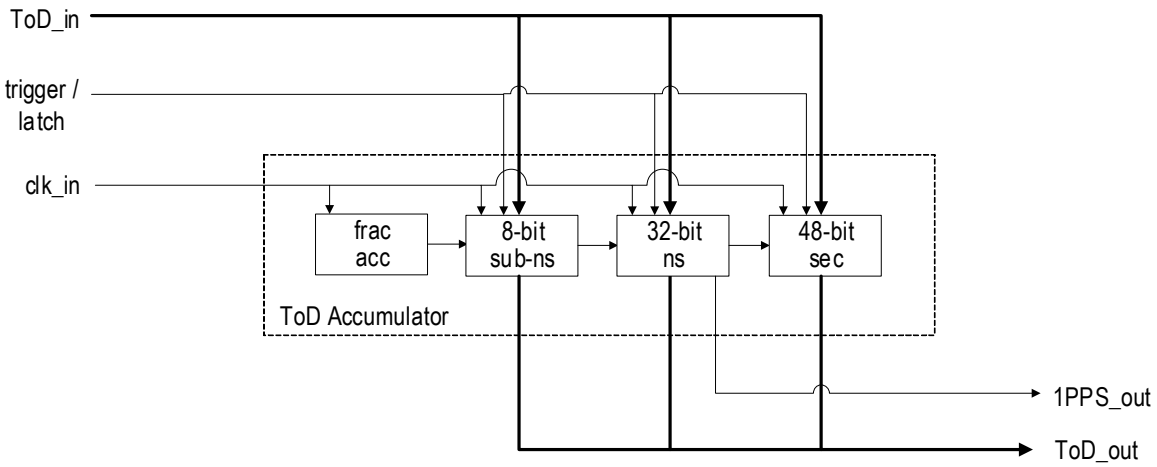
Four of the DPLL paths (DPLL0 to DPLL3) are connected to an independent Time of Day (ToD) counter or accumulator, which can be used to time-stamp external events using various triggers/latches described below, and/or can be used to represent local PTP clock (i.e., centralized for the system). The ToD accumulator tracks the input reference clock or packet stream that the DPLL is locked to, and synchronizes the output clocks and a 1PPS sync pulse signal generated from the same DPLL.

The ToD accumulator is based on the PTP EPOCH and records the Time of Day in PTP format: 48-bit seconds, 32-bit nanoseconds (roll-over at 9999,9999,999, making it an effective 30-bit counter), and an 8-bit fractional sub-nanoseconds. So all ToD registers are 11-byte (11B) in length. The ToD accumulator is clocked by the same FOD clock that goes to the output integer dividers, and supports the following clock rates:

- Any clock rate based on M/N × 500MHz (M and N are 16-bit numbers), from 500MHz up to a maximum of 1GHz[1]
- Standard Ethernet or FEC Ethernet rates, based on M/N × 625MHz (M and N are 8-bit numbers), from 625MHz up to a maximum of 1GHz[1]
- For all other clock rates, the Time of Date accumulator can be used as a cycle counter. In that case, a count of 1 is accumulated and the 32-bit nanosecond bit-field with the roll-over tracked in the 48-bit second bit-field, making it an 80-bit cycle counter. The 8-bit sub-ns bitfield will remain at zero.

In order to avoid truncation errors for the sub-nanoseconds, a flexible modulus accumulator is used, as displayed below.

Figure 34. ToD Accumulator



ToD Triggers and Latches

In order to synchronize the 8A34002 Time of Day with a Time of Day source (such as a IEEE 1588 server), the device has several triggers and latches available. These can be used to select on which trigger the ToD accumulator can be synchronized to a preprogrammed value and to latch the ToD accumulator to sample the Time of Day.

Sources of triggers/latches are:

- Read/write access to the ToD registers
- Active edge of internal 1PPS (as a read trigger only)
- Active edge of an external Sync Pulse (1PPS, PPES, etc)
- Active edge of a GPIO signal (as a read trigger only)

[1] DPLL 0 and 1 only; DPLL 2 and 3 have maximum of 750MHz.

- Internal Global Timer, based on System PLL
- Read/write access to a pre-configured CSR
- Interrupt source, directed to a GPIO

The ToD accumulator value can also be updated and distributed by the 8A34002 using PWM (see Pulse-Width Modulation Encoders, Decoders, and FIFO).

Read/Write to ToD or ToD-based Registers

Unless otherwise specified, any referenced bits or registers throughout this section reside in the TOD_WRITE_X section of the register map, where *X* ranges from 0 to 3. TOD_WRITE_0 is CC00 (Offset 000h to 00Fh). TOD_READ_PRIMARY_0 base address CC40h (offset 000h to 000Eh). For more information about other TOD registers, see the *8A3xxxx Family Programming Guide.*

For general 1588 applications, the ToD accumulator must be synchronized to the overall network Time of Day. This can be done by programming the ToD accumulator with the network Time of Day using a simple write to the 11B ToD register. The update of the ToD accumulator with the 11B ToD register value is triggered on the final write to the MSB of the ToD second register. The ToD accumulator will be updated with the ToD value after a maximum of X clock cycles of the ToD accumulator clock. For better precision on programming the ToD accumulator, the Active Edge Sampling and Loading and Other Asynchronous Events mechanisms should be used.

Similarly, the ToD accumulator can be read at any time by the microprocessor. The latch of the ToD accumulator value is triggered on an initial read to the previous word before the ToD register. The full 11B ToD value can be subsequently read after this access, or a burst access starting at the previous word can be performed. The ToD register is updated with the ToD accumulator value after a maximum of X clock cycles of the ToD accumulator clock. For better precision on latching the ToD accumulator, the Active Edge Sampling and Loading and Other Asynchronous Events mechanisms should be used.

The 8A34002 also supports latching of the ToD accumulator on read/write access to specific programmed register. The desired register to trigger or latch the ToD accumulator is programmed via register. The latched ToD value is contained in the Register ToD FIFO.

Active Edge Sampling and Loading

The 8A34002 provides precise triggering and latching of the ToD accumulator using an active edge of an internal synchronous clock (i.e., 1PPS) or an external Sync Pulse or event (i.e., GPIO).

The following active-edge sources can latch a Time of Day value from the ToD accumulator on a single edge or on continuous active edges.

- External Sync Pulse (1PPS, PPES, etc.)
  - 11B ToD value will be latched in the FIFO on next active edge
- PWM
  - 11B ToD will be latched on internal 1PPS. The 11B ToD value will be sent out on the corresponding PWM_PPS frame (see Pulse-Width Modulation Encoders, Decoders, and FIFO)
  - Alternately, 11B ToD may be latched on a PWM_PPS “reply” request and sent out on a PWM_PPS frame (see Ranging Operation)

The following active edge sources can trigger the programming of the ToD accumulator on a single edge or on continuous active edges.

- External Sync Pulse (1PPS, PPES, etc.)
  - Value in 11B ToD register will be programmed into ToD accumulator on next active edge
- PWM
  - 11B ToD value from the PWM_PPS frame will be programmed into ToD accumulator on the next PWM_PPS Frame reception (see Pulse-Width Modulation Encoders, Decoders, and FIFO)

When using GPIOs, there will be an inaccuracy due to internal delays. For better precision on latching the ToD accumulator, use an unused reference clock/pulse input.

Other Asynchronous Events

The 8A34002 also provides precise triggering and latching of the ToD accumulator of an asynchronous event, such as an interrupt source directed to a GPIO or the global system timer.

When a GPIO is used for interrupt(s), the ToD accumulator can be latched to indicate when the interrupt event occurred. As previously mentioned, due to internal delays to the GPIO block, the precision of the interrupt active edge sampling will have an error.

The global system timer will can be programmed to trigger on the expiration of a countdown timer. The counter is in steps of the system clock (2.5ns ± the accuracy of the oscillator).

GPIO Functions Associated with ToD Operation

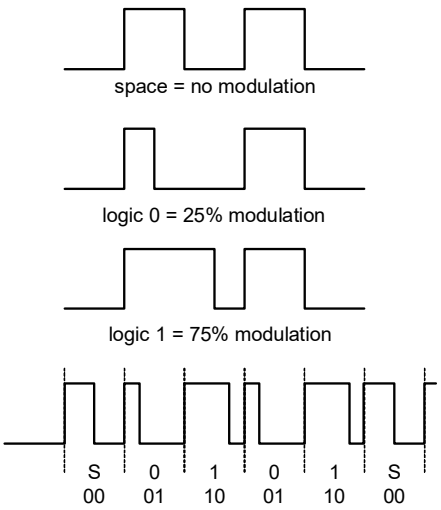
- Time-of-Day (ToD) trigger input – In this mode of operation, the GPIO acts as an input that will latch the current Time-of-Day value into one of internal ToD registers. The register affected by which GPIO must be pre-configured via registers over the serial port. For information on ToD registers, see Time-of-Day (ToD) Operation.
- Time-of-Day Trigger Output – In this mode of operation, the GPIO acts as an output that will assert at a specific Time-of-Day value programmed into one of internal ToD registers. The register affected by the GPIO must be pre-configured via registers over the serial port. For information on ToD registers, see Time-of-Day (ToD) Operation.

Pulse-Width Modulation Encoders, Decoders, and FIFO

Unless otherwise specified, any referenced bits or registers throughout this section reside in the PWM_ENCODER_X and PWM_DECODER_X section of the register map, where *X* ranges from 0 to 7, and 0 to 15, respectively. Base address for PWM_ENCODER_0 is CB00h (offset 000h to 0004h) and PWM_DECODER_0 is CB40h (offset 000h to 005h). For more information about other PWM_ENCODER and PWM_DECODER registers, see the *8A3xxxx Family Programming Guide*.

Each output on the 8A34002 can encode information onto a carrier clock using pulse-width modulation (PWM). The rising edge of the carrier clock is not affected by the pulse-width modulation, allowing for unaware devices to still lock to the carrier clock. The falling edge of the clock is modulated to represent one of three symbols: space (no modulation), logic 0 (25% modulation), and logic 1 (75% modulation). This is displayed below, along with a sample of an encoded stream.

Figure 35. PWM Coded Symbols



Each input on the 8A34002 can decode information from a carrier clock using pulse-width modulation (PWM). The carrier clock and/or decoded information can then be sent to any DPLL or to the System DPLL.

The supported carrier frequencies are in the range of 8kHz[1] to 25MHz. For the encoder, the source of the carrier clock can be from a DPLL or the System DPLL.

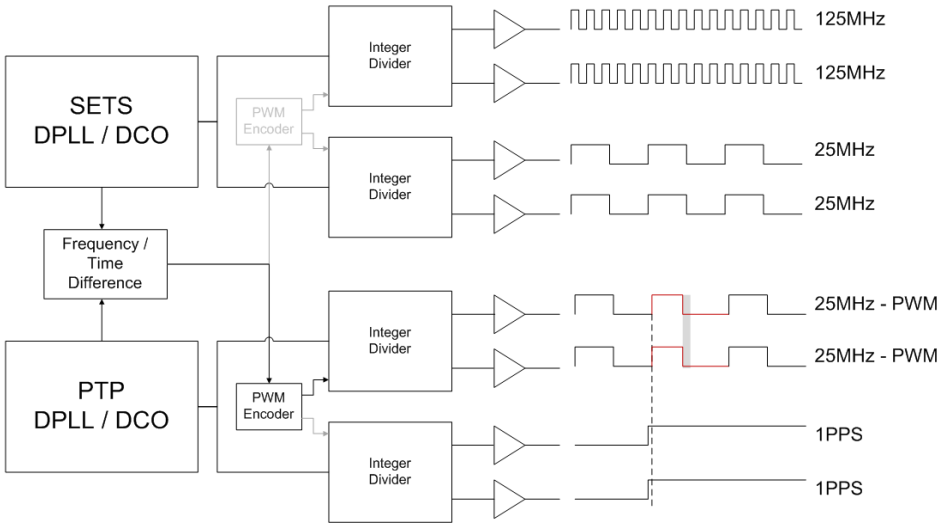
The PWM mechanism can use a “Signature” to mark the position of a 1 second boundary (1PPS) that is synchronous[1] to the carrier. Or, the PWM mechanism can use “Framing”, where data channels are modulated onto the carrier to carry additional information; such as the 1PPS boundary (which can be synchronous or asynchronous to the carrier), Time of Day (ToD), and/or the difference between the frequency and phase of a clock that is asynchronous to the carrier clock. These data channels use fixed size frames to send information on the carrier (see PWM Frames).

PWM Signature

When using Signature mode, the 8-symbol Signature is programmable and includes any combination of ZERO, ONE, or SPACE symbols. The only restriction is that the first symbol transmitted must be either a ZERO or a ONE. In Signature mode, no additional PWM Frames are sent out on the carrier, and the 1PPS source (from the ToD accumulator or the other divided down output clock) is synchronous with the carrier (i.e., sourced from the same DPLL).

The PWM Signature is transmitted immediately following reception of a 1PPS pulse from the ToD accumulator (default) or the other divided down output (DPLL0~3 only). As displayed in Figure 36, a ZERO symbol is sent out as the first symbol of the Signature. For DPLL0~3, the selection of the carrier can be either of the two outputs (with the other being the available divided down source for 1PPS).

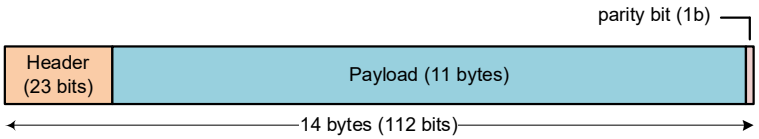
Figure 36. PWM Signature – 1PPS Encoding Example



PWM Frames

The PWM Frame is an alternative to the PWM Signature, and must be used if multiple data channels are required, if ToD information needs to be transferred in addition to 1PPS, and/or if the carrier is asynchronous with the 1PPS source. The PWM data channel mechanism works by encoding/decoding “frames” contained on a carrier clock. These 112-bit (14 Bytes) frames consist of a 23-bit header followed by 11 bytes of payload and a parity bit. The parity is calculated over the entire 14B frame.

Figure 37. PWM Frame



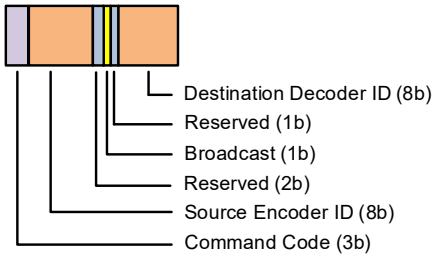
[1] The lower carrier frequency of 8kHz is only recommended when using the PWM “Signature” to mark the position of a synchronous[2] 1 second boundary. For PWM “Frames”, it is recommended to use a higher carrier frequency.

[1] If the second boundary (1PPS) is asynchronous to the carrier, then a PWM_PPS frame must be sent (see 1PPS and ToD Distribution (PWM_PPS)).

The PWM Frame Header is comprised of the following:

- Command Code (3-bits)
- Source Encoder ID (8-bits), defined in register SCSR_PWM_ENCODER_ID
- Broadcast (1-bit)
- Destination Decoder ID (8-bits), defined in register SCSR_PWM_DECODER_ID

Figure 38. PWM Frame Header



The 8A34002 encoders and decoders supports four Command Codes natively in hardware: PWM_PPS (000b), PWM_SYNC (011b), PWM_READ (010b), and PWM_WRITE (1000b).

- The Source Encoder ID represents a unique identifier of the originator of the PWM frame. An 8A34002 can be assigned a unique identifier but the originator can also be internal to the originator device, such as one of the DPLLs or System PLL.
- The Broadcast bit means that all devices on the carrier clock should process the PWM frame. If this bit is “1”, the Destination Decoder ID is ignored. This bit is only set by a “Master” device, such as an 8A34002 on a Timing Card.
- The Destination Decoder ID represents a unique identifier of a remote device. An 8A34002 can be assigned a unique identifier but the destination can also be internal to the destination device, such as one of the DPLLs or System PLL.

1PPS and ToD Distribution (PWM_PPS)

The most practical use of the PWM encoder is the ability to encode the 1PPS phase information onto the 1588 carrier clock (see Figure 36). As previously mentioned, each occurrence of 1PPS can send out an 8-symbol Signature frame. However, if any other PWM frame data channel is required (e.g., PWM_SYNC, PWM_READ/WRITE) or if ToD information needs to be transferred or if the carrier is asynchronous with the 1PPS source, then the PWM_PPS framing must be enabled.

This mechanism uses a standard PWM Frame to represent the 1PPS and, optionally, ToD information. For DPLL/DOCO0~3 encoders, any of the four ToD accumulators can be selected as the source of the PPS, which allows for the use of an asynchronous carrier. For DPLL/DOCO0~3 decoders, any of the four ToD accumulators can be selected to be updated with received ToD information.

Figure 39 shows the PWM_PPS Frame format. A PWM_PPS frame is transmitted immediately following reception of a 1PPS pulse from a ToD accumulator (default), or the other divided down output (DPLL0~3 only). If the ToD accumulator is asynchronous to the carrier, then the PWM_PPS frame will be sent out earlier on the carrier, and the offset will be included in the ToD data field. On reception, the decoder can optionally generate an internal 1PPS that can be selected by any of the DPLLs.

Figure 39. PWM Frame – PWM_PPS

Multi-Clock Distribution (PWM_SYNC)

One of the advance capabilities of the 8A34002 is to allow an asynchronous clock to be encoded onto a carrier clock, which allows for multiple timing channels to be distributed on a single carrier. An example may be to encode the SETS clock onto the Time (1588) clock, along with 1588 1PPS+ToD information. Another example may be to encode the System DPLL (i.e., OCXO) clock onto the SETS clock, along with 1588 1PPS+ToD information, for OCXO redundancy, or to provide a stable clock source to line cards (i.e., reducing BOM costs at the line card).

This mechanism uses a PWM Frame to represent the Synchronization data of a DPLL clock. For the encoder, any of the DPLLs, including the System DPLL, can be selected as a clock source. For the decoder, any of the DPLLs, including the System DPLL, can be selected to be updated with the synchronization data.

The multi-clock distribution is a proprietary mechanism controlled via SCSRs between compatible Renesas devices.

Register Read/Write (PWM_READ/PWM_WRITE)

In addition to sending 1PPS/ToD and synchronization data, a “Master” device can send a read/write request to a remote device’s Control and Status Registers (CSRs). This uses a PWM Frame to represent the read/write data, and requires a two-way data channel with the remote device.

Figure 40 and Figure 41 show the PWM_READ and PWM_WRITE Frame formats, respectively.

Figure 40. PWM Frame – PWM_READ

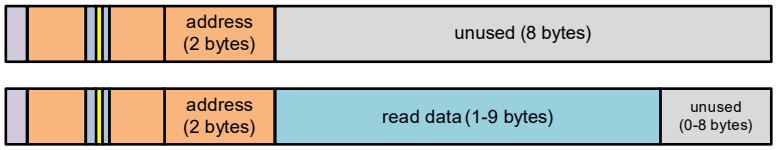
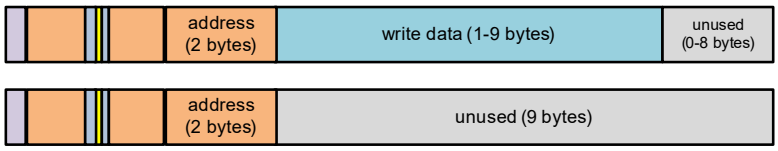


Figure 41. PWM Frame – PWM_WRITE



For PWM_READ frames:

- Bytes [0:1] – 2-byte CSR address
- Bytes [2:10] – Read data
- Remaining bytes – Reserved

For PWM_WRITE frames:

- Bytes [0:1] – 2-byte CSR address
- Bytes [2:10] – Write data
- Remaining bytes – Reserved

External Data Channel (PWM FIFO)

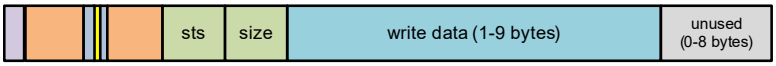
The 8A34002 allows for a data channel to be established between microprocessors. If the microprocessor writes to the data channel transmit FIFO (SCSR_SET_BYTE), then the 11B payload will be encoded and sent to the destination 8A34002. At the remote end, the received PWM data will be decoded and the 11B payload will be sent to the data channel receive FIFO.

This data channel can co-exist with the other PWM channels, such as 1PPS and ToD Distribution, and Multi-Clock Distribution.

Up to 128 bytes can be sent. The user PWM data channel uses a PWM Frame to represent the read/write data, and is unidirectional; thus, there is no automatic acknowledgment from the remote end that the data was received.

Figure 42 shows the PWM_USER Frame format.

Figure 42. User PWM Frame – PWM_USER



For PWM_USER frames, the first two bytes contain status information and payload size, allowing for up to 9 bytes to be transferred per frame.

- Byte [0] – Status, made up of the following
  - First frame flag (1-bit)
  - Remaining bytes (7-bits) – The maximum size of the block is 128 bytes, so the remaining bytes is always less than 128.
- Byte [1] – Data size, indicates the length of the current data (1 to 9)
- Bytes [2:10] – User data
- Remaining bytes – Reserved

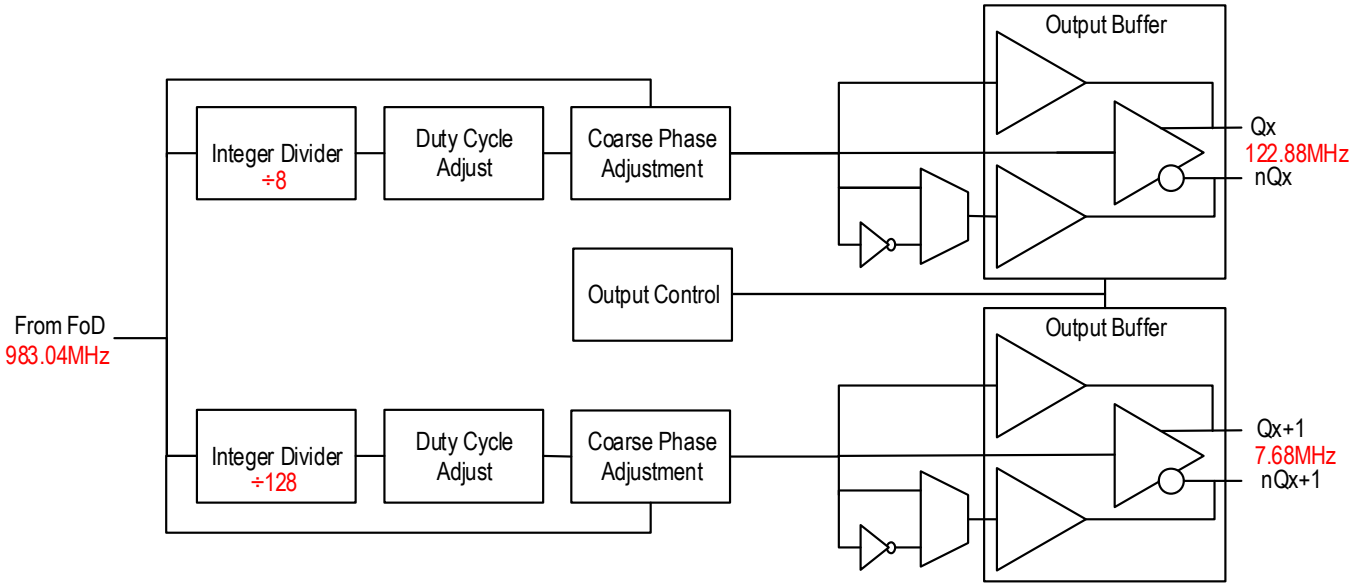
Other Operating Modes

There are additional operating modes and functions supported by the 8A34002 that assist in switchovers between electrical and packet clocks as well as numerous other ways of adjusting for input and output phase relationships. Please contact Renesas for Application Notes or support for needs that are not described here.

JESD204B SYSREF Output Operation

The 8A34002 allows any of the output clock signals to be used as a JESD204B SYSREF output, referenced to any other output clock that is synchronous to it. Figure 43 shows how this would work if using a dual-output stage block. The same principles will apply if two single-stage outputs are used as long as they are connected to the same FOD (via appropriate setup on the FOD muxes; for more information, see FOD Multiplexing and Output Stages), or connected to separate FODs that are running at the same frequency and have been phase synchronized with one another.

Figure 43. SYSREF Usage Example



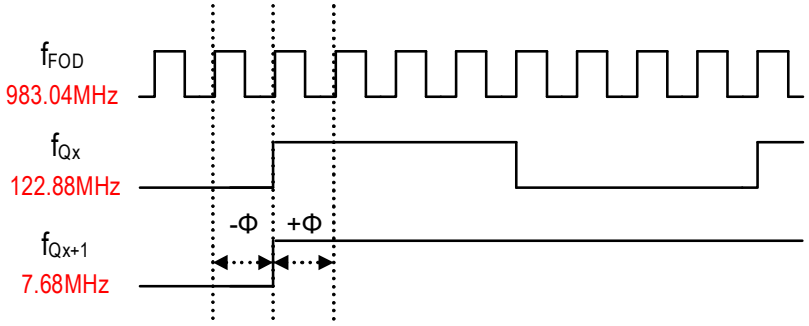
For the purposes of this example, the output clock will be on the Qx / nQx output pair (upper path in the figure) with a frequency of 122.88MHz and the associated SYSREF on the Qx+1/nQx+1 output pair (lower path in the figure) with a frequency of 7.68MHz. The output clock and the SYSREF are synchronous to one another since they are derived from the same frequency source: the FOD running at 983.04MHz. The SYSREF frequency is usually a lower frequency that is an integer divisor of the output clock. As a result, the integer divider in the lower path will be set to a multiple (128 in this example) of the integer divider ratio in the upper path (8 in this example).

The SYSREF output can be enabled or disabled glitchlessly at any time via the serial port, or via a GPIO pin configured as an Output Enable.

Output Phase Alignment between Clock and SYSREF

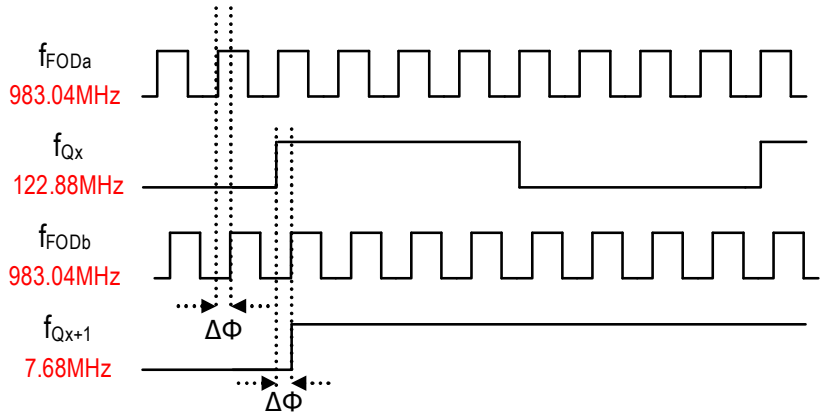
The 8A34002 provides precise control of phase alignment between all outputs. With no adjustment used, all outputs will align in phase within the output-output skew limits specified in the “AC Electrical Characteristics” table. The phase of any output, whether a regular clock or a SYSREF, can be adjusted using coarse adjustment for each output path independently. This coarse adjustment is in steps of the FOD clock period (see Figure 44), and any number of steps can be added or subtracted. For more information, see Output Coarse Phase Adjustment. For this example, the coarse phase step is 1.02nsec.

Figure 44. SYSREF Coarse Phase Adjustment



For more precise adjustment, fine phase adjustment can be performed in the FOD or DPLL block. In this case, the SYSREF and regular clock are being generated from separate PLL channels that are operating in frequency synchronization with each other and at some point in time were synchronized in phase. At a later point in time, a fine phase adjust was applied to one channel to offset their phases from each other in steps of 1psec in either direction. This is displayed in Figure 45 with the regular clock being derived from FODa and the SYSREF being derived from FODb. In the figure, FODa and FODb are both running at the same frequency, but FODb is delayed in phase by an amount ΔΦ. This results in the SYSREF output being delayed by that same amount. For information how to do this if the PLL channel is locked to an input clock, see Write-Phase Mode. If the PLL channel is in open-loop mode, and is operating as a FOD, information can be found in Steerable Fractional Output Divider (FOD).

Figure 45. SYSREF Fine Phase Adjustment



A mixture of coarse and fine adjustments can be used to achieve any desired phase relationship between the regular and SYSREF outputs.

AC and DC Specifications

Abbreviations Used in this Section

Many signals will be concatenated in the specification tables that follow. Table 23 shows a list of abbreviations used and will be referred to in footnotes for the various other tables.

Table 23. Abbreviated Signal Names and the Detailed Signal Names Referenced by Them

Abbreviation	Signals Referenced by this Abbreviation
V_{DD_CLKx}	V_{DD_CLKA}, V_{DD_CLKB}
Input CLK	CLK[6:0], nCLK[6:0]
Output Q	Q[7:0], nQ[7:0]
Status Outputs	GPIO[9,8,5:0], SDIO, SDA_M, SCL_M
GPIO	GPIO[9,8,5:0]
V_{DDx}	$V_{DDA\_PDCP\_XTAL}$, V_{DD_CLKA}, V_{DD_CLKB}, V_{DDA_FB}, V_{DDA_BG}, V_{DDA_LC}, V_{DD_DIG}, V_{DD_GPIO}, $V_{DDA\_DIA\_A}$, $V_{DDA\_DIA\_B}$, $V_{DD\_DCO\_Q0123}$, $V_{DD\_DCO\_Q4567}$, V_{DDO_Q0}, V_{DDO_Q1}, V_{DDO_Q2}, V_{DDO_Q3}, V_{DDO_Q4}, V_{DDO_Q5}, V_{DDO_Q6}, V_{DDO_Q7}
V_{DDO_Qx}	V_{DDO_Q0}, V_{DDO_Q1}, V_{DDO_Q2}, V_{DDO_Q3}, V_{DDO_Q4}, V_{DDO_Q5}, V_{DDO_Q6}, V_{DDO_Q7}
$V_{DD\_DCO\_Qx}$	$V_{DD\_DCO\_Q0123}$, $V_{DD\_DCO\_Q4567}$
V_{DDA_DIAx}	$V_{DDA\_DIA\_A}$, $V_{DDA\_DIA\_B}$

Absolute Maximum Ratings

The absolute maximum ratings are stress ratings only. Stresses greater than those listed below can cause permanent damage to the device. Functional operation of the 8A34002 at absolute maximum ratings is not implied. Exposure to absolute maximum rating conditions may affect device reliability.

Table 24. Absolute Maximum Ratings

Symbol	Parameter	Test Condition	Minimum	Maximum	Units
V_{DDx}[a]	Any voltage supply		-0.5	3.63	V
V_{IN}	Voltage on any input	OSCI[b], OSCO, FILTER, C_{REG_XTAL}	0	2.75	V
		All other inputs[c]	-0.5	3.63	V
I_{IN}	Differential Input Current	Input CLK[a]		±50	mA
I_O	Output Current - Continuous	Output Q[a]		30	mA
		Status Outputs[a]		25	mA
	Output Current - Surge	Output Q		60	mA
		Status Outputs		50	mA
T_{JMAX}	Maximum Junction Temperature			150	°C
T_S	Storage temperature		-65	150	°C
-	ESD - Human Body Model			2000	V
-	ESD - Charged Device Model			1500	V

[a] For information on the signals referenced by this abbreviation, see Table 23.

[b] This limit only applies to the OSCI input when being over-driven by an external signal. No limit is implied when this is connected directly to a crystal.

[c] For minimum and maximum VIH and VIL for each input referenced to its associated V_{DD}, see Table 29.

Recommended Operating Conditions

Table 25. Recommended Operating Conditions[a][b]

Symbol	Parameter	Minimum	Typical	Maximum	Units
T_A	Ambient air temperature	-40		85	°C
T_J	Junction temperature			125	°C

[a] It is the user's responsibility to ensure that device junction temperature remains below the maximum allowed.

[b] All conditions in this table must be met to guarantee device functionality.

Supply Voltage Characteristics

Table 26. Power Supply DC Characteristics[a][b]

Symbol	Parameter	Test Conditions	Minimum	Typical	Maximum	Units
V_{DD_CLKx}[c]	Supply Voltage for Input Clock Buffers and Dividers		1.71	[d]	3.465	V
I_{DD_CLKx}[e]	Supply Current for V_{DD_CLKx}	V_{DD_CLKx} = 3.465V, PMOS mode		3	4	mA
		V_{DD_CLKx} = 3.465V, NMOS mode		6	7	
		V_{DD_CLKx} = 3.465V, CMOS mode		1.5	4	
		V_{DD_CLKx} = 2.625V, PMOS mode		2.6	3	
		V_{DD_CLKx} = 2.625V, NMOS mode		6	7	
		V_{DD_CLKx} = 2.625V, CMOS mode		1	2	
		V_{DD_CLKx} = 1.89V, PMOS mode		2.5	3	
		V_{DD_CLKx} = 1.89V, NMOS mode		5	7	
		V_{DD_CLKx} = 1.89V, CMOS mode		1	2	
$V_{DDA\_PDCP\_XTAL}$	Analog Supply Voltage for oscillator and for SysAPLL Phase detector & Charge Pump		2.375	[f]	3.465	V
$I_{DDA\_PDCP\_XTAL}$	Supply Current for $V_{DDA\_PDCP\_XTAL}$	$V_{DDA\_PDCP\_XTAL}$ = 3.3V		48	55	mA
		$V_{DDA\_PDCP\_XTAL}$ = 2.5V		33	40	mA
V_{DDA_FB}	Analog Supply Voltage for SysAPLL Feedback Divider		1.71	1.8	1.89	V
I_{DDA_FB}	Supply Current for V_{DDA_FB}	V_{DDA_FB} = 1.89V		22	37	mA
V_{DDA_BG}	Analog Supply Voltage for SysAPLL Bandgap reference		2.375	[f]	3.465	V
I_{DDA_BG}	Supply Current for V_{DDA_BG}	V_{DDA_BG} = 3.465V		19	26	mA
		V_{DDA_BG} = 2.625V		16	22	mA
V_{DDA_LC}	Analog Supply Voltage for SysAPLL LC Resonator		2.375	[f]	3.465	V
I_{DDA_LC}	Supply Current for V_{DDA_LC}	V_{DDA_LC} = 3.465V		97	121	mA
		V_{DDA_LC} = 2.625V		65	71	mA
V_{DD_GPIO}	Supply Voltage for GPIO and other status / control pins		1.425	[g]	3.465	V

Table 26. Power Supply DC Characteristics[a][b]

Symbol	Parameter	Test Conditions	Minimum	Typical	Maximum	Units
I_{DD_GPIO}	Supply Current for V_{DD_GPIO}	V_{DD_GPIO} = 3.465V		9	15	mA
		V_{DD_GPIO} = 2.625V		7	12	mA
		V_{DD_GPIO} = 1.89V		4	9	mA
		V_{DD_GPIO} = 1.575V		1	6	mA
V_{DD_DIG}	Digital Supply Voltage		1.14	[h]	1.89	V
I_{DD_DIG}	Supply Current for V_{DD_DIG}	V_{DD_DIG} = 1.89V		190	380	mA
		V_{DD_DIG} = 1.26V		180	295	mA
$V_{DD\_DCO\_Qx}$[c]	Supply Voltage for each FOD block		1.71	1.8	1.89	V
$I_{DD\_DCO\_Qx}$[i]	Supply Current for $V_{DD\_DCO\_Qx}$[j]	$V_{DD\_DCO\_Qx}$ = 1.89V Base current (FOD Off) $I_{DD(DCOBASE)}$		24	44	mA
		Adder for FOD at 500MHz $I_{DD(DCOPERFOD)}$		30		mA
		Adder per 1MHz over 500MHz on FOD $I_{DD(DCOPERMHZ)}$		0.012		mA/MHz
$V_{DDA\_DIA\_A}$	Supply Voltage for: ▪ FOD_0 control logic ▪ FOD_1 control logic		1.71	1.8	1.89	V
$I_{DDA\_DIA\_A}$	Supply Current for $V_{DDA\_DIA\_A}$[k]	$V_{DDA\_DIA\_A}$ = 1.89V $I_{DD(DIABASE)}$		9	17	mA
		Adder per FOD at 500MHz $I_{DD(DIAPERFOD)}$		10		mA
		Adder per FOD per 1MHz over 500MHz $I_{DD(DIAPERMHZ)}$		0.018		mA/MHz
$V_{DDA\_DIA\_B}$	Supply Voltage for: ▪ FOD_2 control logic ▪ FOD_3 control logic		1.71	1.8	1.89	V
$I_{DDA\_DIA\_B}$[l]	Supply Current for $V_{DDA\_DIA\_B}$[k]	$V_{DDA\_DIA\_B}$ = 1.89V $I_{DD(DIABASE)}$		32	44	mA
		Adder per FOD at 500MHz $I_{DD(DIAPERFOD)}$		10		mA
		Adder per FOD per 1MHz over 500MHz $I_{DD(DIAPERMHZ)}$		0.012		mA/MHz
V_{DDO_Qx}[c]	Output Clock Q Supply Voltage[m]		1.14		3.465	V

[a] V_{SS} = 0V, T_A = -40°C to 85°C

[b] Current consumption figures represent a worst-case consumption with all functions associated with the particular voltage supply being all enabled and running at full capacity. This information is provided to allow for design of appropriate power supply circuits that will support all possible register-based configurations for the device. Please refer to the section Power Consumption to determine actual consumption for the exact configuration of the device.

[c] Please refer to Table 23 for details on the signals referenced by this abbreviation.

[d] Supports 1.8V±5%, 2.5V±5% or 3.3V±5% operation, not a continuous range.

[e] I_{DD_CLKx} denotes the current consumed by the appropriate V_{DD_CLKx} supply voltage

[f] Supports 2.5V±5% or 3.3V±5% operation, not a continuous range.

[g] Supports 1.5V, 1.8V, 2.5v or 3.3V operation.

[h] Supports 1.2V±5% or 1.8V±5% operation, not a continuous range.

[i] $I_{DD\_DCO\_Qx}$ denotes the current consumed by the appropriate $V_{DD\_DCO\_Qx}$ supply voltage. This is the current consumption for each supply, not the total for all $V_{DD\_DCO\_Qx}$.

[j] The $I_{DD\_DCO\_Qx}$ current consumed by an FOD is highly dependent on the frequency it is running at. So a calculation needs to be performed as shown below, where f_{FOD} is the frequency the FOD is operating at in MHz and the 3 current values are the ones shown in the table for that particular power supply. Note that only the base current is needed if the FOD is disabled.

$$I_{DD(DCO)} = I_{DD(DCOBASE)} + I_{DD(DCOPERFOD)} + (f_{FOD} - 500) \times I_{DD(DCOPERMHZ)}$$

[k] The I_{DDA_DIA} current consumed is dependent on the number of FODs attached to the voltage rail that are supported and the frequency of operation of those FODs. A calculation needs to be performed using the formula below, where f_{FOD} is the operating frequency of each FOD, NumFOD is the number of FODs on that supply that are enabled. Note that only the base current is needed if all FODs are disabled.

$$I_{DD(DIA)} = I_{DD(DIABASE)} + NumFOD \times I_{DD(DIAPERFOD)} + \sum_{operatingDCO} (f_{FOD} - 500) \times I_{DD(DIAPERMHZ)}$$

[l] $V_{DDA\_DIA\_B}$ consumes higher current than $V_{DDA\_DIA\_A}$ because it has some additional circuitry, besides the FODs on it.

[m]Currents for the outputs are shown in Table 27 or Table 28 as appropriate for the mode the individual output is operating in.

Table 27. Output Supply Current (Output Configured as Differential)[a][b][c]

Symbol	Parameter	Test Condition	SWING[d] = 00		SWING = 01		SWING = 10		SWING = 11		Units
			Typ.	Max.	Typ.	Max.	Typ.	Max.	Typ.	Max.	
I_{DDO_Qx}[e]	Qx / nQx Supply Current[f]	V_{DDO_Qx}[g] = 3.465V	15	22	17	24	19	26	20	26	mA
		V_{DDO_Qx} = 2.625V	14	20	16	21	18	22	19	23	mA
		V_{DDO_Qx} = 1.89V	14	19	15	20	16	21	16	21	mA

[a] Output current consumption is not affected by any of the core device power supply voltage levels.

[b] Internal dynamic switching current at maximum f_{OUT} is included.

[c] V_{DDO_Qx} = 3.3V±5% or 2.5V±5% or 1.8V±5%, V_{SS} = 0V, T_A = -40°C to 85°C.

[d] Refers to the output voltage (swing) setting programed into device registers for each output.

[e] I_{DDO_Qx} denotes the current consumed by each V_{DDO_Qx} supply.

[f] Measured with outputs unloaded.

[g] For information on the signals referenced by this abbreviation, see Table 23.

Table 28. Output Supply Current (Output Configured as LVCMOS)[a][b][c]

Symbol	Parameter	Test Condition	TERM[d] = 00		TERM = 01		TERM = 10		TERM = 11		Units
			Typ.	Max.	Typ.	Max.	Typ.	Max.	Typ.	Max.	
I_{DDO_Qx}[e]	Qx, nQx Supply Current[f] Qx and nQx Both Enabled	V_{DDO_Qx}[g] = 3.465V	24	32	25	35	25	37	25	39	mA
		V_{DDO_Qx} = 2.625V	18	25	19	27	19	29	20	30	
		V_{DDO_Qx} = 1.89V	12	20	14	21	15	22	15	23	
		V_{DDO_Qx} = 1.575V	9	17	11	18	11	19	12	20	
		V_{DDO_Qx} = 1.26V	6	13	6	13	6	14	6	14	
	Qx, nQx Supply Current[h] Qx enabled and nQx Tri-stated	V_{DDO_Qx} = 3.465V	14	23	14	24	14	25	14	26	mA
		V_{DDO_Qx} = 2.625V	11	19	11	20	11	20	11	21	
		V_{DDO_Qx} = 1.89V	9	16	10	17	10	17	10	18	
		V_{DDO_Qx} = 1.575V	8	15	8	16	9	16	9	16	
		V_{DDO_Qx} = 1.26V	5	12	5	12	5	12	5	12	

[a] Output current consumption is not affected by any of the core device power supply voltage levels.

[b] Internal dynamic switching current at maximum f_{OUT} is included.

[c] V_{SS} = 0V, T_A = -40°C to 85°C

[d] Refers to the LVCMOS output drive strength (termination) setting programed into device registers for each output.

[e] I_{DDO_Qx} denotes the current consumed by each V_{DDO_Qx} supply.

[f] Measured with outputs unloaded.

[g] For information on the signals referenced by this abbreviation, see Table 23.

[h] Measured with outputs unloaded.

DC Electrical Characteristics

Table 29. LVCMOS/LVTTL DC Characteristics[a][b][c][d][e]

Symbol	Parameter		Test Condition	Minimum	Typical	Maximum	Units
V_{IH}	Input High Voltage	nMR, nTEST, GPIO[9,8,5:0], SCLK, SDIO, SDI_A1, CS_A0, SDA_M, SCL_M	V_{DD_GPIO} = 3.3V±5%	2		V_{DD_GPIO} + 0.3	V
			V_{DD_GPIO} = 2.5V±5%	1.7		V_{DD_GPIO} + 0.3	V
			V_{DD_GPIO} = 1.8V±5%	0.65 x V_{DD_GPIO}		V_{DD_GPIO} + 0.3	V
			V_{DD_GPIO} = 1.5V±5%	0.65 x V_{DD_GPIO}		V_{DD_GPIO} + 0.3	V
V_{IH}	Input High Voltage	XO_DPLL	V_{DD_DIG} = 1.8V±5%	1.17		3.465V	V
			V_{DD_DIG} = 1.2V±5%	1.17		3.465V	V
V_{IH}	Input High Voltage	CLK[3:0], nCLK[3:0][e]	V_{DD_CLKA} = 3.3V±5%	2		V_{DD_CLKA} + 0.3	V
			V_{DD_CLKA} = 2.5V±5%	1.7		V_{DD_CLKA} + 0.3	
			V_{DD_CLKA} = 1.8V±5%	0.65 × V_{DD_CLKA}		V_{DD_CLKA} + 0.3	
		CLK[6:4], nCLK[6:4][e]	V_{DD_CLKB} = 3.3V±5%	2		V_{DD_CLKB} + 0.3	
			V_{DD_CLKB} = 2.5V±5%	1.7		V_{DD_CLKB} + 0.3	
			V_{DD_CLKB} = 1.8V±5%	0.65 × V_{DD_CLKB}		V_{DD_CLKB} + 0.3	
V_{IL}	Input Low Voltage	nMR, nTEST, GPIO[9,8,5:0], SCLK, SDIO, SDI_A1, CS_A0, SDA_M, SCL_M	V_{DD_GPIO} = 3.3V±5%	-0.3		0.8	V
			V_{DD_GPIO} = 2.5V±5%	-0.3		0.7	
			V_{DD_GPIO} = 1.8V±5%	-0.3		0.35 x V_{DD_GPIO}	
			V_{DD_GPIO} = 1.5V±5%	-0.3		0.35 x V_{DD_GPIO}	
V_{IL}	Input Low Voltage	XO_DPLL	V_{DD_DIG} = 1.8V±5%	-0.3		0.35 × V_{DD_DIG}	V
			V_{DD_DIG} = 1.2V±5%	-0.3		0.35 × V_{DD_DIG}	
V_{IL}	Input Low Voltage	CLK[3:0], nCLK[3:0][e]	V_{DD_CLKA} = 3.3V±5%	-0.3		0.8	V
			V_{DD_CLKA} = 2.5V±5%	-0.3		0.7	
			V_{DD_CLKA} = 1.8V±5%	-0.3		0.35 × V_{DD_CLKA}	
		CLK[6:4], nCLK[6:4][e]	V_{DD_CLKB} = 3.3V±5%	-0.3		0.8	
			V_{DD_CLKB} = 2.5V±5%	-0.3		0.7	
			V_{DD_CLKB} = 1.8V±5%	-0.3		0.35 × V_{DD_CLKB}	
I_{IH}	Input High Current	nMR, nTEST, GPIO[9,8,5:0], SDA_M, SCLK, SDIO, SDI_A1, CS_A0	V_{IN} = V_{DD_GPIO} = V_{DD_GPIO} (max)			5	µA
I_{IH}	Input High Current	XO_DPLL	V_{IN} = 3.465V, V_{DD_DIG} = V_{DD_DIG} (max)			150	µA
I_{IH}	Input High Current	CLK[3:0]	V_{IN} = V_{DD_CLKA} = V_{DD_CLKA} (max)			150	µA
		nCLK[3:0]				5	
		CLK[6:4]	V_{IN} = V_{DD_CLKB} = V_{DD_CLKB} (max)			150	
		nCLK[6:4]				5	

Table 29. LVCMOS/LVTTL DC Characteristics[a][b][c][d][e] (Cont.)

Symbol	Parameter		Test Condition	Minimum	Typical	Maximum	Units
I_{IL}	Input Low Current	nMR, nTEST, GPIO[9,8,5:0], SDA_M, SCLK, SDIO, SDI_A1, CS_A0	V_{IN} = 0V, V_{DD_GPIO} = V_{DD_GPIO} (max)	-150			µA
I_{IL}	Input Low Current	XO_DPLL	V_{IN} = 0V, V_{DD_DIG} = V_{DD_DIG} (max)	-5			µA
I_{IL}	Input Low Current	CLK[3:0]	V_{IN} = 0V, V_{DD_CLKA} = V_{DD_CLKA} (max)	-5			µA
		nCLK[3:0]		-150			
		CLK[6:4]	V_{IN} = 0V, V_{DD_CLKB} = V_{DD_CLKB} (max)	-5			
		nCLK[6:4]		-150			
V_{OH}	Output High Voltage	GPIO[9,8,5:0], SDA_M, SCL_M, SCLK, SDIO	V_{DD_GPIO} = 3.3V±5%, I_{OH} = -100µA	V_{DD_GPIO} - 0.2			V
			V_{DD_GPIO} = 3.3V±5%, I_{OH} = -12mA	2.6			
			V_{DD_GPIO} = 2.5V±5%, I_{OH} = -100µA	V_{DD_GPIO} - 0.2			
			V_{DD_GPIO} = 2.5V±5%, I_{OH} = -8mA	1.8			
			V_{DD_GPIO} = 1.8V±5%, I_{OH} = -100µA	V_{DD_GPIO} - 0.2			
			V_{DD_GPIO} = 1.8V±5%, I_{OH} = -2mA	V_{DD_GPIO} - 0.45			
			V_{DD_GPIO} = 1.5V±5%, I_{OH} = -100µA	1.2			
			V_{DD_GPIO} = 1.5V±5%, I_{OH} = -2mA	0.75 x V_{DD_GPIO}			

Table 29. LVCMOS/LVTTL DC Characteristics[a][b][c][d][e] (Cont.)

Symbol	Parameter		Test Condition	Minimum	Typical	Maximum	Units
V_{OL}	Output Low Voltage	GPIO[9,8,5:0], SDA_M, SCL_M, SCLK, SDIO	V_{DD_GPIO} = 3.3V±5%, I_{OL} = 100µA			0.2	V
			V_{DD_GPIO} = 3.3V±5%, I_{OL} = 12mA			0.5	
			V_{DD_GPIO} = 2.5V±5%, I_{OL} = 100µA			0.2	
			V_{DD_GPIO} = 2.5V±5%, I_{OL} = 8mA			0.5	
			V_{DD_GPIO} = 1.8V±5%, I_{OL} = 100µA			0.2	
			V_{DD_GPIO} = 1.8V±5%, I_{OL} = 2mA			0.45	
			V_{DD_GPIO} = 1.5V±5%, I_{OL} = 100µA			0.2	
			V_{DD_GPIO} = 1.5V±5%, I_{OL} = 2mA			0.25 x V_{DD_GPIO}	

[a] V_{IL} should not be less than -0.3V.

[b] 3.3V characteristics in accordance with JESD8C-01,
2.5V characteristics in accordance with JESD8-5A.01,
1.8V characteristics in accordance with JESD8-7A,
1.5V characteristics in accordance with JESD8-11A.01,
1.2V characteristics in accordance with JESD8-12A.01

[c] V_{SS} = 0V, T_A = -40°C to 85°C.

[d] When Output Q are configured as LVCMOS, their output characteristics are specified in Table 34.

[e] Input pair used as two single-ended clocks rather than as a differential clock.

Table 30. Differential Input DC Characteristics[a]

Symbol	Parameter		Test Condition	Minimum	Typical	Maximum	Units
I_{IH}	Input High Current	CLK[3:0]	$V_{IN} = V_{DD\_CLKA} = V_{DD\_CLKA}$ (max)			150	µA
		nCLK[3:0]				5	
		CLK[6:4]	$V_{IN} = V_{DD\_CLKB} = V_{DD\_CLKB}$ (max)			150	
		nCLK[6:4]				5	
I_{IL}	Input Low Current	CLK[3:0]	V_{IN} = 0V, $V_{DD\_CLKA} = V_{DD\_CLKA}$ (max)	-5			µA
		nCLK[3:0]		-150			
		CLK[6:4]	V_{IN} = 0V, $V_{DD\_CLKB} = V_{DD\_CLKB}$ (max)	-5			
		nCLK[6:4]		-150			
V_{PP}	Peak-to-Peak Voltage[b][c]		Any input protocol	0.15		1.3	V
V_{CMR}	Common Mode Input Voltage[b][d]	CLK[3:0], nCLK[3:0]	Input protocol = HCSL, HSTL, SSTL	0.1		V_{DD_CLKA} - 1.2	V
			Input protocol = LVDS, LVPECL, CML	0.7		V_{DD_CLKA} - V_{PP} / 2	
		CLK[6:4], nCLK[6:4]	Input protocol = HCSL, HSTL, SSTL	0.1		V_{DD_CLKB} - 1.2	
			Input protocol = LVDS, LVPECL, CML	0.7		V_{DD_CLKB} - V_{PP} / 2	

[a] V_{SS} = 0V, T_A = -40°C to 85°C.

[b] V_{IL} should not be less than -0.3V.

[c] V_{PP} is the single-ended amplitude of the input signal. The differential specs is 2*V_{PP}.

[d] Common mode voltage is defined as the cross-point.

Table 31. Differential Output DC Characteristics (V_{DDO_Qx} = 3.3V+5%, V_{SS} = 0V, T_A = -40°C to 85°C)[a][b][c][d]

Symbol	Parameter		Test Condition	Minimum	Typical	Maximum	Units
V_{OVS}[e]	Output Voltage Swing	Output Q[a]	SWING = 00[f]	336	402	462	mV
			SWING = 01	478	605	698	
			SWING = 10	658	791	910	
			SWING = 11	739	870	997	
V_{CMR}[g]	Output Common Mode Voltage	Output Q[a]	CENTER = 000[h]	0.86	0.95	1.07	V
			CENTER = 001	0.98	1.14	1.28	
			CENTER = 010	1.13	1.33	1.51	
			CENTER = 011	1.30	1.53	1.73	
			CENTER = 100	1.46	1.73	1.95	
			CENTER = 101	1.63	1.93	2.17	
			CENTER = 110	1.80	2.12	2.39	
			CENTER = 111	1.96	2.30	2.59	

[a] For information on the signals referenced by this abbreviation, see Table 23.

[b] Terminated with 100Ω across Qx and nQx.

[c] If LVDS operation is desired, the user should select SWING = 00 and CENTER = 001 or 010.

[d] If LVPECL operation is desired, the user should select SWING = 10 and CENTER = 101 or 110 for 3.3V LVPECL, and SWING = 10 and CENTER = 001 or 010 for 2.5V LVPECL operation.

[e] V_{OVS} is the single-ended amplitude of the output signal. The differential specs is 2*V_{OVS}.

[f] Refers to the differential voltage swing setting programed into device registers for each output.

[g] Not all V_{CMR} selections can be supported with particular V_{DDO_Qx} and V_{OVS} settings. For information on which combinations are supported, see Table 16.

[h] Refers to the differential voltage crossing point (center voltage) setting programed into device registers for each output.

Table 32. Differential Output DC Characteristics (V_{DDO_Qx} = 2.5V+5%, V_{SS} = 0V, T_A = -40°C to 85°C)[a][b][c][d]

Symbol	Parameter		Test Condition	Minimum	Typical	Maximum	Units
V_{OVS}[e]	Output Voltage Swing	Output Q[a]	SWING = 00[f]	295	393	448	mV
			SWING = 01	457	591	677	
			SWING = 10	587	761	881	
			SWING = 11	733	835	943	
V_{CMR}[g]	Output Common Mode Voltage	Output Q[a]	CENTER = 000[h]	0.85	0.93	1.03	V
			CENTER = 001	0.94	1.10	1.23	
			CENTER = 010	1.09	1.28	1.44	
			CENTER = 011	1.24	1.46	1.65	
			CENTER = 100	1.39	1.65	1.86	
			CENTER = 101	Not Supported			
			CENTER = 110				
			CENTER = 111				

[a] For information on the signals referenced by this abbreviation, see Table 23.

[b] Terminated with 100Ω across Qx and nQx.

[c] If LVDS operation is desired, the user should select SWING = 00 and CENTER = 001 or 010.

[d] If LVPECL operation is desired, the user should select SWING = 10 and CENTER = 001 or 010 for 2.5V LVPECL operation. For V_{DDO} = 2.5V, 3.3V LVPECL levels cannot be generated.

[e] V_{OVS} is the single-ended amplitude of the output signal. The differential specs is 2*V_{OVS}.

[f] Refers to the differential voltage swing setting programed into device registers for each output.

[g] Not all V_{CMR} selections can be supported with particular V_{DDO_Qx} and V_{OVS} settings. For information on which combinations are supported, see Table 16.

[h] Refers to the differential voltage crossing point (center voltage) setting programed into device registers for each output.

Table 33. Differential Output DC Characteristics (V_{DDO_Qx} = 1.8V+5%, V_{SS} = 0V, T_A = -40°C to 85°C)[a][b][c]

Symbol	Parameter		Test Condition	Minimum	Typical	Maximum	Units
V_{OVS}[d]	Output Voltage Swing	Output Q[a]	SWING = 00[e]	299	411	485	mV
			SWING = 01	470	586	700	
			SWING = 10	582	713	852	
			SWING = 11	612	750	899	
V_{CMR}[f]	Output Common Mode Voltage	Output Q[a]	CENTER = 000[g]	0.84	0.91	0.99	V
			CENTER = 001	0.91	1.05	1.18	
			CENTER = 010	1.05	1.21	1.36	
			CENTER = 011	Not Supported			
			CENTER = 100				
			CENTER = 101				
			CENTER = 110				
			CENTER = 111				

[a] For information on the signals referenced by this abbreviation, see Table 23.

[b] Terminated with 100Ω across Qx and nQx.

[c] If LVDS operation is desired, the user should select SWING = 00 and CENTER = 010.

[d] V_{OVS} is the single-ended amplitude of the output signal. The differential specs is 2*V_{OVS}.

[e] Refers to the differential voltage swing setting programed into device registers for each output.

[f] Not all V_{CMR} selections can be supported with particular V_{DDO_Qx} and V_{OVS} settings. For information on which combinations are supported, see Table 16.

[g] Refers to the differential voltage crossing point (center voltage) setting programed into device registers for each output.

Table 34. LVCMOS Clock Output DC Characteristics[a][b][c]

Symbol	Parameter	Test Condition	TERM[d] = 00			TERM = 01			TERM = 10			TERM = 11			Units
			Min.	Typ.	Max.	Min.	Typ.	Max.	Min.	Typ.	Max.	Min.	Typ.	Max.	
V_{OH}	Output High Voltage	V_{DDO_Qx} = 3.3V±5%	0.74 × V_{DDO_Qx}			0.75 × V_{DDO_Qx}			0.75 × V_{DDO_Qx}			0.75 × V_{DDO_Qx}			V
		V_{DDO_Qx} = 2.5V±5%	0.70 × V_{DDO_Qx}			0.75 × V_{DDO_Qx}			0.75 × V_{DDO_Qx}			0.75 × V_{DDO_Qx}			
		V_{DDO_Qx} = 1.8V±5%	0.65 × V_{DDO_Qx}			0.71 × V_{DDO_Qx}			0.75 × V_{DDO_Qx}			0.75 × V_{DDO_Qx}			
		V_{DDO_Qx} = 1.5V±5%	0.61 × V_{DDO_Qx}			0.66 × V_{DDO_Qx}			0.70 × V_{DDO_Qx}			0.72 × V_{DDO_Qx}			
		V_{DDO_Qx} = 1.2V±5%	0.56 × V_{DDO_Qx}			0.59 × V_{DDO_Qx}			0.63 × V_{DDO_Qx}			0.66 × V_{DDO_Qx}			
V_{OL}	Output Low Voltage	V_{DDO_Qx} = 3.3V±5%			0.29 × V_{DDO_Qx}			0.25 × V_{DDO_Qx}			0.25 × V_{DDO_Qx}			0.25 × V_{DDO_Qx}	V
		V_{DDO_Qx} = 2.5V±5%			0.32 × V_{DDO_Qx}			0.27 × V_{DDO_Qx}			0.25 × V_{DDO_Qx}			0.25 × V_{DDO_Qx}	
		V_{DDO_Qx} = 1.8V±5%			0.39 × V_{DDO_Qx}			0.33 × V_{DDO_Qx}			0.30 × V_{DDO_Qx}			0.26 × V_{DDO_Qx}	
		V_{DDO_Qx} = 1.5V±5%			0.44 × V_{DDO_Qx}			0.38 × V_{DDO_Qx}			0.35 × V_{DDO_Qx}			0.31 × V_{DDO_Qx}	
		V_{DDO_Qx} = 1.2V±5%			0.50 × V_{DDO_Qx}			0.46 × V_{DDO_Qx}			0.42 × V_{DDO_Qx}			0.38 × V_{DDO_Qx}	
Z_{OUT}	Output Impedance	V_{DDO_Qx} = 3.3V±5%		35			25			21			18		Ω
		V_{DDO_Qx} = 2.5V±5%		31			23			20			17		
		V_{DDO_Qx} = 1.8V±5%		42			31			25			21		
		V_{DDO_Qx} = 1.5V±5%		71			47			35			29		
		V_{DDO_Qx} = 1.2V±5%		101			86			66			49		

[a] V_{SS} = 0V, T_A = -40°C to 85°C.

[b] V_{DDO_Qx} is used to refer to the appropriate V_{DDO_Qx} power supply voltage for each output. For more information, see Table 23 and the "Pin Description" table.

[c] Measured with outputs terminated with 50Ω to V_{DDO_Qx} / 2.

[d] This refers to the register settings for the LVCMOS output drive strength within the device.

Table 35. Input Frequency Characteristics[a]

Symbol	Parameter		Test Condition	Minimum	Typical	Maximum	Units
f_{IN}	Input Frequency	OSCI, OSCO	Using a Crystal[b]	25		54	MHz
			Over-driving Crystal Input Doubler Logic Enabled[c]	25		62.5	
			Over-driving Crystal Input Doubler Logic Disabled	50		125	
		Input CLK[d][e]	Differential Mode	0.0000005		1000	
			Single-ended Mode			250	
		GPIO	Used as Clock Input			150	
f_{IN}	Input Frequency	XO_DPLL		1		150[f]	MHz
f_{SCLK}	Serial Port Clock SCLK (slave mode)	I^2C Operation		100		1200	kHz
		SPI Operation		0.1		50	MHz

[a] V_{SS} = 0V, T_A = -40°C to 85°C

[b] For crystal characteristics, see Table 36.

[c] Refer to Overdriving the XTAL Interface.

[d] For information on the signals referenced by this abbreviation, see Table 23.

[e] For proper device operation, the input frequency must be divided down to 150MHz or less (DPLL Phase Detector maximum frequency = 150MHz).

[f] If the System DPLL needs to be driven with a higher frequency, one of the CLKx / nCLKx inputs can be routed via register settings to the System DPLL instead of using XO_DPLL.

Table 36. Crystal Characteristics[a]

Parameter	Test Condition	Minimum	Typical	Maximum	Units
Mode of Oscillation		Fundamental			
Frequency		25		54	MHz
Equivalent Series Resistance (ESR)	C_L= 18pF, crystal frequency ≤ 40MHz			50	Ω
	C_L= 18pF, crystal frequency > 40MHz			25	
	C_L= 12pF			50	
Load Capacitance (C_L)		8	12		pF
Crystal Drive Level			250		µW

[a] V_{SS} = 0V, T_A = -40°C to 85°C

AC Electrical Characteristics

Table 37. AC Characteristics[a][b]

Symbol	Parameter		Test Condition		Minimum	Typical	Maximum	Units
f_{VCO}	Analog PLL VCO Operating Frequency		V_{DDA_X}[c] = 3.3V±5%		13.4		13.8	GHz
			V_{DDA_X}[c] = 2.5V±5%		13.5		13.9	
f_{FOD}	Fractional Output Divider Operating Frequency		Measured with output divider set to /1		500		1000	MHz
f_{OUT}	Output Frequency	Differential Output			0.0000005		1000	MHz
		LVCMOS Output			0.0000005		250	
Δf_{OUT}	Output Frequency Accuracy[d]					0		ppb
	Initial Frequency Offset[e]		Switchover or Entering Holdover State			1		ppb
	Output Phase Change due to Hitless Switching[f]		HS Type 1, switching to an input reference with frequency ≥ 1MHz			100	200	ps
			HS Type 1, switching to an input reference with frequency < 1MHz			400	1700	
			HS Type 2			90	200	
t_{SK}	Output-to-Output Skew[g][h]		Any two differential outputs[i]			80	160	ps
			Any two outputs configured as LVCMOS in-phase[j]	V_{DDO_Qx} = 3.3V±5%, 2.5V±5%, 1.8V±5% or 1.5V±5%		100	300	
				V_{DDO_Qx} = 1.2V ±5%		160	360	

Table 37. AC Characteristics[a][b] (Cont.)

Symbol	Parameter	Test Condition		Minimum	Typical	Maximum	Units
$t_{SK(B)}$	Output to Output Skew within a Bank[g][h]	Differential Outputs	1st Bank: Q0, Q1, Q6, Q7		25	65	ps
			2nd Bank: Q2, Q3, Q4, Q5		25	75	
		Outputs Configured as LVCMOS in-phase V_{DDO_Qx} = 3.3V±5%, 2.5V±5% or 1.8V±5%	1st Bank: Q0, Q1, Q6, Q7		30	110	
			2nd Bank: Q2, Q3, Q4, Q5		30	110	
		Outputs Configured as LVCMOS in-phase V_{DDO_Qx} = 1.5V ±5% or 1.2V ±5%	1st Bank: Q0, Q1, Q6, Q7		50	150	
			2nd Bank: Q2, Q3, Q4, Q5		50	225	
		Q to nQ of same output pair, cfg. as LVCMOS, in-phase[j]	V_{DDO_Qx} = 3.3V ±5% or 2.5V ±5%		10	60	
			V_{DDO_Qx} = 1.8V ±5% or 1.5V ±5% or 1.2V ±5%		15	100	
Δt_{SK}	Temperature Variation[k] Output-Output					4	ps/°C
Δt_{ALIGN}	Temperature Variation[k] Input-Output					4	ps/°C
t_{ALIGN}	Input - Output Alignment Variation[l][m]	Delay variation as shown in Figure 46 for any non-inverted CLK/nCLK input pair to any Q/nQ output pair in differential mode when using internal loopback.		-500		500	ps
		Delay variation as shown in Figure 47 for any non-inverted PMOS CLK/nCLK input pair to any Q/nQ output pair in differential mode when used as external loopback.[n][o]		-130		130	
ITDCMA	Input TDC Measurement Accuracy[p][q][r]	REF = CLK2/nCLK2, FB = CLK3/nCLK3, DPLL0. All other input clocks and DPLLs disabled. 10MHz differential signal applied to inputs CLK2/nCLK2 and CLK3/nCLK3. Fine phase measurements enabled.			±20	±90	ps

Table 37. AC Characteristics[a][b] (Cont.)

Symbol	Parameter		Test Condition		Minimum	Typical	Maximum	Units
t_R / t_F	Output Rise and Fall Times 20% to 80%	Differential Output[s][t]	V_{DDO_Qx}[u] = 3.3V±5%, 2.5V±5% or 1.8V±5%	SWING[v] = 00	100		450	ps
				SWING = 01				
				SWING = 10				
				SWING = 11				
		LVCMOS Output[w]	V_{DDO_Qx} = 3.3V±5%	TERM[x] = 00	100	254	380	ps
				TERM = 01	100	262	400	
				TERM = 10	110	275	460	
				TERM = 11	115	268	510	
			V_{DDO_Qx} = 2.5V±5%	TERM = 00	115	285	405	ps
				TERM = 01	120	293	470	
				TERM = 10	120	315	525	
				TERM = 11	140	347	565	
			V_{DDO_Qx} = 1.8V±5%	TERM = 00	205	417	590	ps
				TERM = 01	205	458	715	
				TERM = 10	230	459	800	
				TERM = 11	235	482	880	
			V_{DDO_Qx} = 1.5V±5%[y]	TERM = 00	415	558	730	ps
				TERM = 01	545	747	985	
				TERM = 10	615	890	1145	
				TERM = 11	690	1011	1305	
			V_{DDO_Qx} = 1.2V±5%[y]	TERM = 00	800	986	1250	ps
				TERM = 01	1180	1416	1835	
				TERM = 10	1415	1715	2195	
				TERM = 11	1650	1980	2520	

Table 37. AC Characteristics[a][b] (Cont.)

Symbol	Parameter		Test Condition		Minimum	Typical	Maximum	Units
odc	Output Duty Cycle	Differential Output	PULSE = 50%	f_{OUT} < 500MHz	47	50	53	%
				500MHz ≤ f_{OUT} < 800MHz	45	50	55	%
				f_{OUT} ≥ 800MHz	40	50	60	%
		LVCMOS	PULSE = 50%	V_{DDO_Qx} = 3.3V or 2.5V	47	50	53	%
				V_{DDO_Qx} = 1.8V or 1.5V	45	50	55	
				V_{DDO_Qx} = 1.2V	42	50	58	
		Any Output Type Operating as a Frame or Sync Pulse	PULSE = Sync Pulse, 100ns		100			ns
			PULSE = Sync Pulse, 1µs		1			µs
			PULSE = Sync Pulse, 10µs		10			µs
			PULSE = Sync Pulse, 100µs		200			µs
			PULSE = Sync Pulse, 1ms		2			ms
			PULSE = Sync Pulse, 10ms		10			ms
			PULSE = Sync Pulse, 100ms		100			ms
			PULSE = Frame Pulse, 0.2UI		0.2			UI
			PULSE = Frame Pulse, 1UI		1			UI
			PULSE = Frame Pulse, 2UI		2			UI
tjit(Φ)[z]	Phase Jitter, RMS (Random) Integration Range: 10KHz to 20MHz	Synthesizer Mode V_{CCA_SEL} = **3.3V**	Crystal 49.152MHz	122.88MHz		126	177	fs
			Crystal 39.0625MHz	156.25MHz		112	139	
			Crystal 49.152MHz	156.25MHz		133	207	
			Crystal 49.152MHz	245.76MHz		124	166	
			Crystal 49.152MHz	312.5MHz		119	159	
			Crystal 49.152MHz	322.265625MHz		125	180	
			Crystal 49.152MHz	983.04MHz		96	149	
		Jitter Attenuation Mode, DPLL Bandwidth: 25Hz, Input Frequency: 10MHz VCCA_SEL = **3.3V**	Crystal 50MHz	122.88MHz		170	248	
			Crystal 49.152MHz	156.25MHz		138	226	
			Crystal 50MHz	245.76MHz		145	233	
			Crystal 49.152MHz	312.5MHz		126	172	
			Crystal 49.152MHz	322.265625MHz		129	185	
			Crystal 50MHz	983.04MHz		103	176	

Table 37. AC Characteristics[a][b] (Cont.)

Symbol	Parameter		Test Condition		Minimum	Typical	Maximum	Units
tjit(Φ)[z]	Phase Jitter, RMS (Random) Integration Range: 10KHz to 20MHz	Synthesizer Mode V_{CCA_SEL} = **2.5V**	Crystal 49.152MHz	122.88MHz		162	232	fs
			Crystal 39.0625MHz	156.25MHz		131	204	
			Crystal 49.152MHz	156.25MHz		166	240	
			Crystal 49.152MHz	245.76MHz		161	229	
			Crystal 49.152MHz	312.5MHz		153	193	
			Crystal 49.152MHz	322.265625MHz		142	197	
			Crystal 49.152MHz	983.04MHz		137	221	
		Jitter Attenuation Mode, DPLL Bandwidth: 25Hz, Input Frequency: 10MHz VCCA_SEL = **2.5V**	Crystal 50MHz	122.88MHz		197	287	
			Crystal 49.152MHz	156.25MHz		177	229	
			Crystal 50MHz	245.76MHz		167	243	
			Crystal 49.152MHz	312.5MHz		161	207	
			Crystal 49.152MHz	322.265625MHz		145	201	
			Crystal 50MHz	983.04MHz		149	227	
PSRR	Power Supply Rejection Ratio[aa][ab]		f_{NOISE} = 10kHz V_{DDO_Qx} = 3.3V			-48.3		dBc
			f_{NOISE} = 25kHz V_{DDO_Qx} = 3.3V			-50.5		
			f_{NOISE} = 50kHz V_{DDO_Qx} = 3.3V			-52.6		
			f_{NOISE} = 100kHz V_{DD_CLKA} = V_{DD_CLKB} = 3.3V			-74.3		
			f_{NOISE} = 500kHz $V_{DD\_DIA\_B}$ = 1.8V			-59.5		
			f_{NOISE} = 1MHz $V_{DD\_DIA\_B}$ = 1.8V			-64.5		
$t_{STARTUP}$[ac]	Start-up Time[ad]	Regulators Ready[ae]				3		µs
		Internal OTP Start-up	Synthesizer mode			7	10	ms
t_{LOCK}	DPLL Lock Time		DPLL Bandwidth = 300Hz			1.5		s
t_{LOCK}	DPLL Lock Time		DPLL Bandwidth = 20mHz, Phase and frequency snap enabled			20		s
t_{LOCK}	System DPLL Lock Time[af]		DPLL Bandwidth = 300Hz[ag]			48	55	ms
PW	Pulse Width[ah]	Input CLK	Differential Mode		0.45			ns
			Single-ended Mode		1.6			
		GPIO	Used as Clock Input; Input divider not bypassed		2.66			
			Used as Clock Input; Input divider bypassed		4.6			

[a] V_{SS} = 0V, T_A = -40°C to 85°C.

[b] Electrical parameters are guaranteed over the specified ambient operating temperature range, which is established when the device is mounted in a test socket with maintained transverse airflow greater than 500lfpm. The device will meet specifications after thermal equilibrium has been reached under these conditions.

[c] V_{DDA_X} refers to V_{DDA_PDCP}, V_{DDA_XTAL}, V_{DDA_LC}, and V_{DDA_BG}.

[d] Long-term frequency error with respect to the DPLL input reference. The typical value shown assumes the DPLL has been phase-locked to a stable input reference for at least 306 minutes (based on a 0.1mHz advanced holdover filter setting) before going into an advanced holdover state on disqualification of the input reference.

[e] This parameter will vary with the stability of the system clock; the typical value shown assumes an ideal system clock.

[f] This parameter was characterized using low phase noise references for the input clocks and system clock.

[g] Defined as the time between the rising edges of two outputs of the same frequency, configuration, loading, and supply voltage

[h] This parameter is defined in accordance with JEDEC Standard 65.

[i] Measured at the differential cross points.

[j] Measured at V_{DDO_Qx} / 2.

[k] This parameter is measured across the full operating temperature range and the difference between the slowest and fastest numbers is the variation.

[l] Measured from the differential cross point of the input to the differential cross point of the associated output after device is locked and input is stable. Measured using integer-related input and output frequencies.

[m] Measured with the channel in DPLL mode.

[n] Characterized using input and output signals with swing = 0.9V, common mode voltage = 0.9V with respect to GND, and matching edge rates.

[o] Characterized using input and output signals with swing = 0.410V, common mode voltage = 1.3V with respect to GND (LVDS signals), and matching edge rates.

[p] Characterized over offset between REF and FB signals in the range of -20ns to 20ns.

[q] Characterized using BGA-144 package devices.

[r] The typical specification applies for all combinations of DPLLs and REF and FB differential pairs.

[s] Rise and fall times on differential outputs are independent of the power supply voltage on the output.

[t] Measured with outputs terminated with 50Ω to GND.

[u] For information on the signals referenced by this abbreviation, see Table 23.

[v] Refers to the differential voltage swing setting programed into device registers for each output.

[w] Measured with outputs terminated with 50Ω to V_{DDO_Qx} / 2.

[x] Refers to the LVCMOS output drive strength (termination) setting programed into device registers for each output.

[y] This parameter has been characterized with F_{OUT} = 50MHz.

[z] All outputs configured for LVDS, enabled and operating at the same frequency.

[aa] 100mV peak-peak sine-wave noise signal injected on indicated power supply pin(s).

[ab] Noise spur amplitude measured relative to 156.25MHz carrier.

[ac] Start-up time will depend on the actual configuration used. For more information on estimating start-up time, please contact Renesas technical support.

[ad] Measured from the rising edge of nMR after all power supplies have reached > 80% of nominal voltage to the first stable clock edge on the output. A stable clock is defined as one generated from a locked analog or digital PLL (as appropriate for the configuration listed) with no further perturbations in frequency expected.

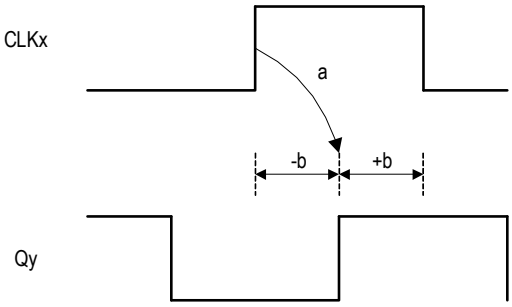
[ae] At power-up, the nMR signal must be asserted for at least this period of time.

[af] Measured from the falling edge to the rising edge of a GPIO reporting the System DPLL lock status after internal OTP start-up.

[ag] System DPLL reference frequency = 10MHz, fractional frequency offset between the System DPLL and System APLL references = ±104.6ppm.

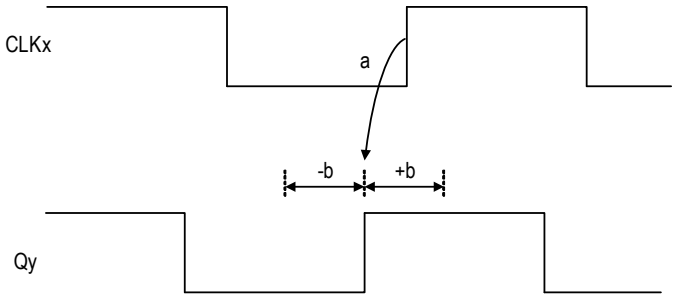
[ah] For proper device operation, the input frequency must be divided down to 150MHz or less (DPLL Phase Detector maximum frequency = 150MHz).

Figure 46. Input-Output Delay with Internal Feedback



a = Fixed delay
b = Delay variation

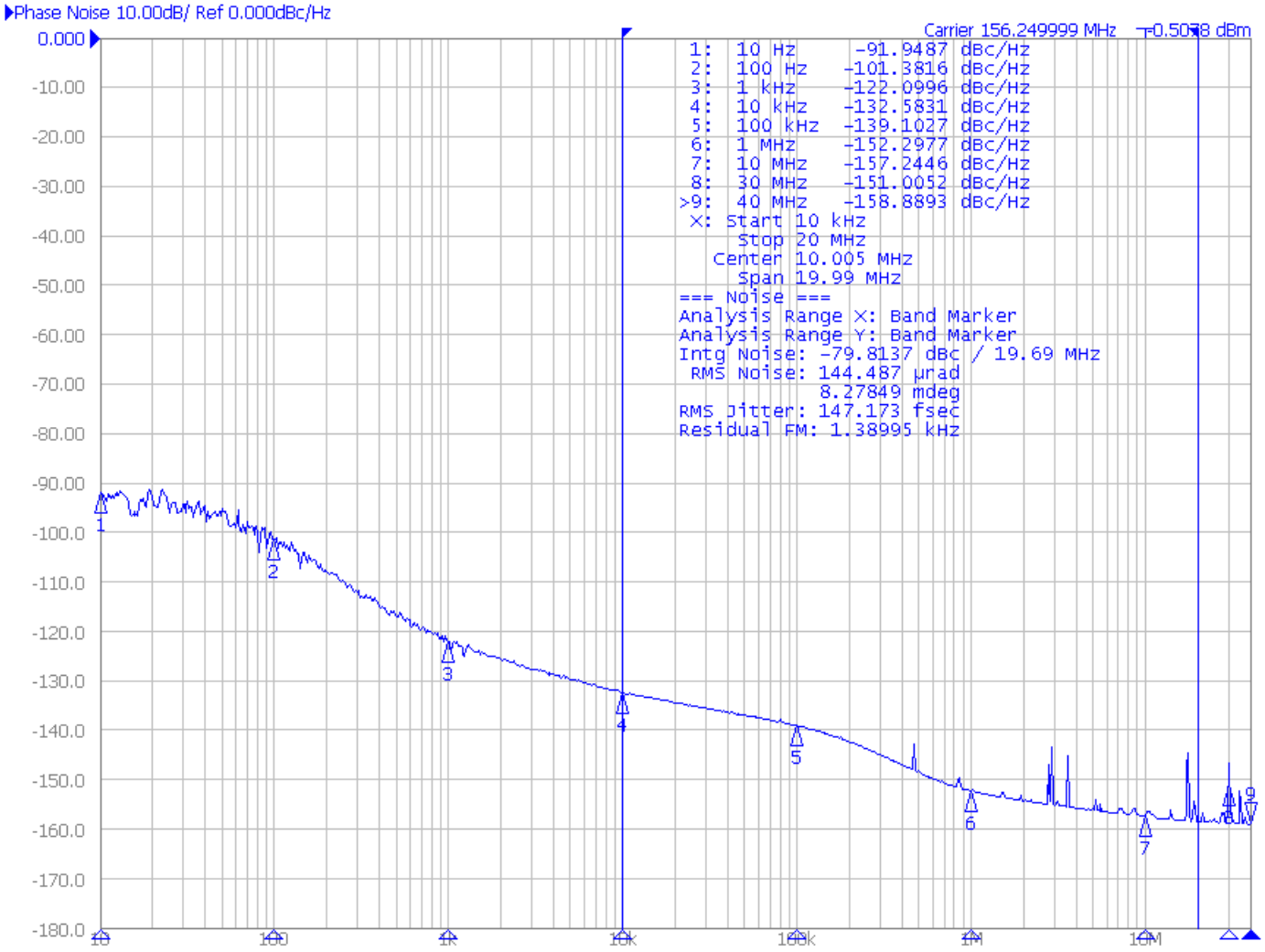
Figure 47. Input-Output Delay with External Feedback



a = Trace delay from Qy to the CLK input used for external feedback
b = Delay variation

Clock Phase Noise Characteristics

Figure 48. 156.25MHz Output Phase Noise



Applications Information

Recommendations for Unused Input and Output Pins

Inputs

CLKx / nCLKx Input

For applications that do not require the use of the reference clock input, both CLK and nCLK should be left floating. If the CLK/nCLK input is connected but not used by the device, it is recommended that CLK and nCLK not be driven with active signals.

LVCMOS Control Pins

LVCMOS control pins have internal pull-ups; additional resistance is not required but can be added for additional protection. A 1kΩ resistor can be used.

Outputs

LVCMOS Outputs

Any LVCMOS output can be left floating if unused. There should be no trace attached. The mode of the output buffer should be set to tri-stated to avoid any noise being generated.

Differential Outputs

All unused differential outputs can be left floating. Renesas recommends that there is no trace attached. Both sides of the differential output pair should either be left floating or terminated.

Power Connections

The power connections of the 8A34002 can be grouped as shown if all members of the groups are using the same voltage level:

- V_{DD_DIG},V_{DD_GPIO}, V_{DD_CLKA}, V_{DD_CLKB}
- $V_{DDA\_PDCP\_XTAL}$
- V_{DDA_FB}
- V_{DDA_BG}, V_{DDA_LC} (not ideal to combine, so if board space allows keep separated),
- $V_{DD\_DIA\_A}$, $V_{DD\_DIA\_B}$ (combining these is a possible source of coupling between frequency domains; should remain separate unless all outputs are in the same frequency domain)
- $V_{DD\_DCO\_Qn}$ (should keep separated except where FOD blocks are all running at the same frequency)
  - If all outputs Qn/nQn and functions associated with any particular $V_{DD\_DCO\_Qn}$ pin are not used, the power pin can be left floating
- V_{DDO_Qn} (can share supplies if output frequencies are the same, otherwise keep separated to avoid spur coupling)
  - If all outputs Qn/nQn associated with any particular V_{DDO_Qn} pin are not used, the power pin can be left floating

Clock Input Interface

The 8A34002 accepts both single-ended and differential inputs. For information on input terminations, see *Quick Guide - Output Terminations (AN-953)* located on the 8A34002 product page.

If you have additional questions on input types not covered in the application discussion, or if you require information about register programming sequences for changing the differential inputs to accept LVCMOS inputs levels, see *Termination - AC Coupling Clock Receivers (AN-844)* or contact Renesas technical support.

Overdriving the XTAL Interface

The OSCI input can be overdriven by an LVCMOS driver or by one side of a differential driver through an AC coupling capacitor. The OSCI input is internally biased at 1V. The OSCO pin can be left floating. The amplitude of the input signal should be between 500mV and 1.8V and the slew rate should not be less than 0.2V/ns. For 1.8V LVCMOS, inputs can be DC-coupled into the device as shown in Figure 49. For 3.3V LVCMOS inputs, the amplitude must be reduced from full swing to at least half the swing in order to prevent signal interference with the power rail and to reduce internal noise. For limits on the frequency that can be used, see Table 35.

Figure 49. 1.8V LVCMOS Driver to XTAL Input Interface

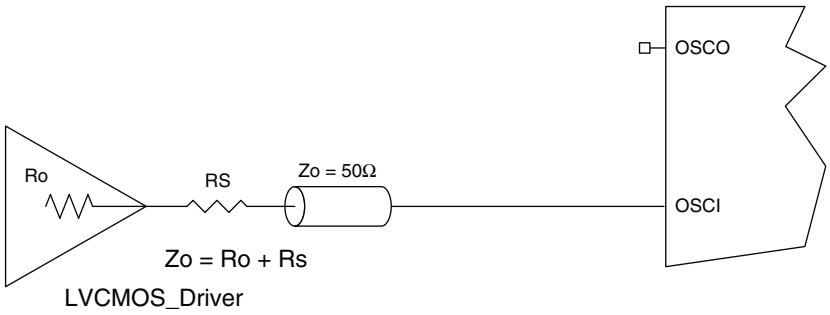


Figure 50 shows an example of the interface diagram for a high-speed 3.3V LVCMOS driver. This configuration requires that the sum of the output impedance of the driver (Ro) and the series resistance (Rs) equals the transmission line impedance. In addition, matched termination at the crystal input will attenuate the signal in half. This can be done in one of two ways. First, R1 and R2 in parallel should equal the transmission line impedance. For most 50Ω applications, R1 and R2 can be 100Ω. This can also be accomplished by removing R1 and changing R2 to 50Ω. The values of the resistors can be increased to reduce the loading for a slower and weaker LVCMOS driver.

Figure 50. LVCMOS Driver to XTAL Input Interface

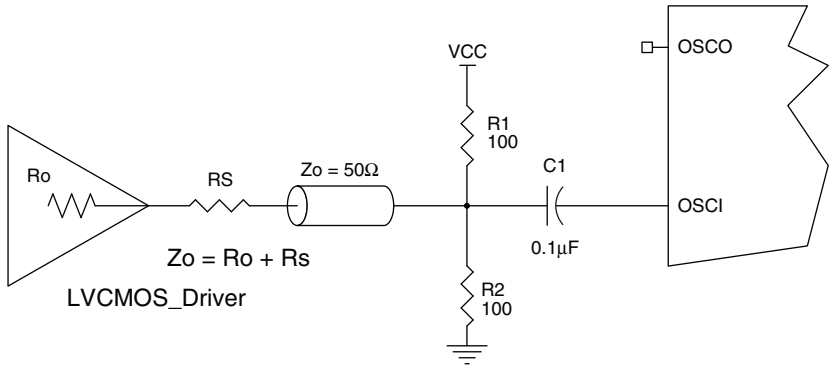
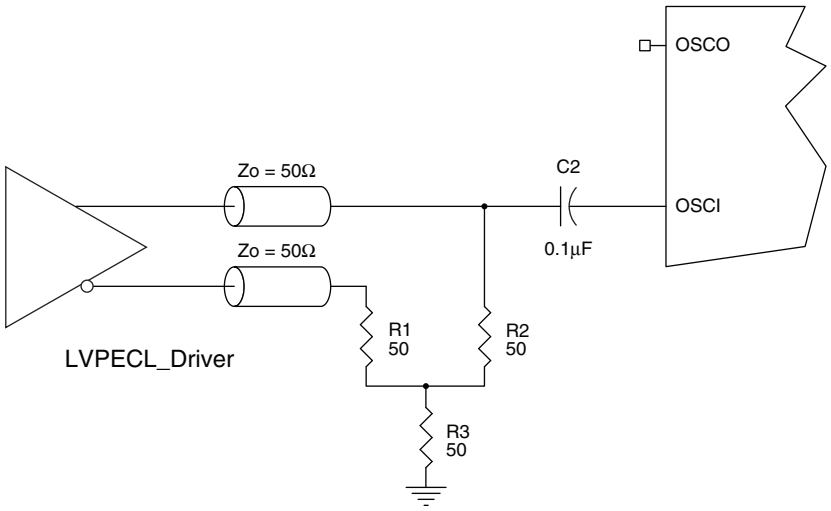


Figure 51 shows an example of the interface diagram for an LVPECL driver. This is a standard LVPECL termination with one side of the driver feeding the XTAL_IN input. It is recommended that all components in the schematics be placed in the layout. Though some components may not be used, they can be utilized for debugging purposes. The datasheet specifications are characterized and guaranteed by using a quartz crystal as the input.

Figure 51. LVPECL Driver to XTAL Input Interface



Wiring the Differential Input to Accept Single-Ended Levels

For information, see the *Differential Input to Accept Single-ended Levels Application Note (AN-836)*.

Differential Output Termination

For all types of differential protocols, the same termination schemes are recommended (see Figure 52 and Figure 53). These schemes are the same as normally used for an LVDS output type.

The recommended value for the termination impedance (Z_T) is between 90Ω and 132Ω. The actual value should be selected to match the differential impedance (Z_{Diff}) of your transmission line. A typical point-to-point LVDS design uses a 100Ω parallel resistor at the receiver and a 100Ω differential transmission-line environment. To avoid any transmission-line reflection issues, the components should be surface-mounted and must be placed as close to the receiver as possible.

Figure 52. Standard LVDS Termination

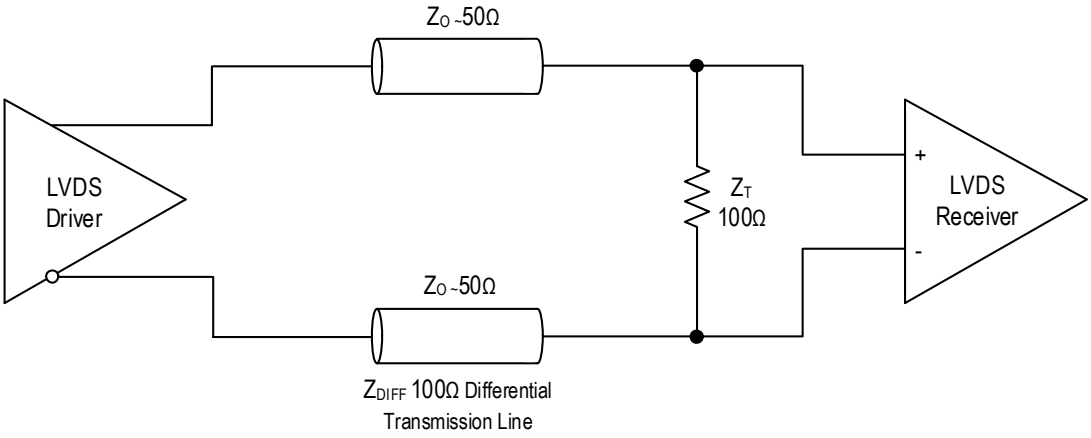
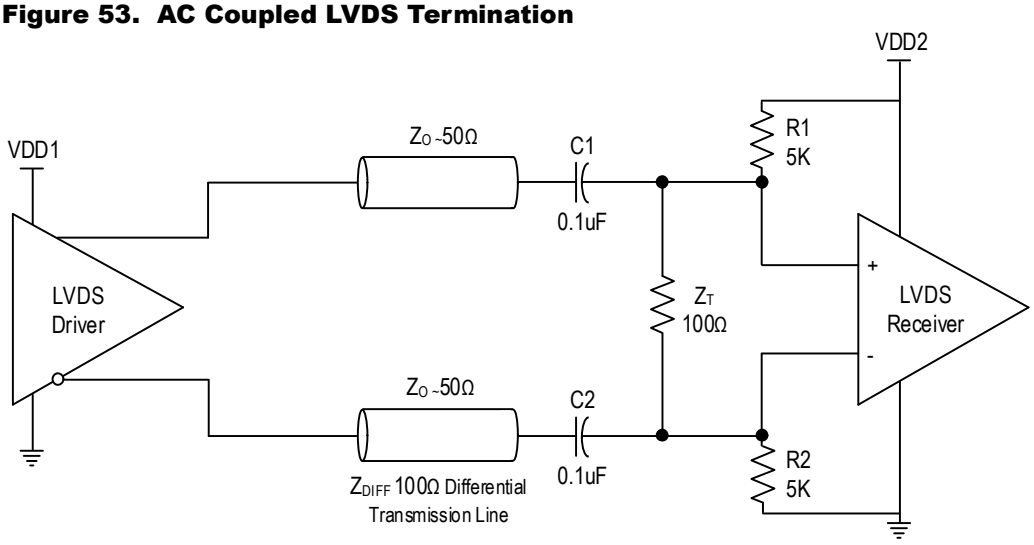


Figure 53. AC Coupled LVDS Termination



For alternate termination schemes, see “LVDS Termination” in *Quick Guide - Output Terminations (AN-953)* located on the device product page, or contact Renesas for support.

Crystal Recommendation

For the latest vendor / frequency recommendations, please contact Renesas.

External I²C Serial EEPROM Recommendation

An external I²C EEPROM can be used to store configuration data and/or to contain device update data. An EEPROM with 8Kbit capacity is sufficient to store a full configuration. However, the recommendation is to use an EEPROM with a 1Mbit capacity in order to support future device updates. Renesas has validated and recommends the use of the Microchip 24FC1025 or OnSemi CAT24M01 1Mbit EEPROM.

Schematic and Layout Information

The 8A34002 requires external load capacitors to ensure the crystal will resonate at the proper frequency. For recommended values for external tuning capacitors, see Table 38.

Table 38. Recommended Tuning Capacitors for Crystal Input

Crystal Nominal C_L Value (pF)	Recommended Tuning Capacitor Value (pF)[a]	
	OSCI Capacitor (pF)	OSCO Capacitor (pF)
8	2.7	2.7
10	13	3.3
12	27	3.3
18[b]	27	3.3

[a] Recommendations are based on 4pF stray capacitance on each leg of the crystal. Adjust according to the PCB capacitance.

[b] This will tune the crystal to a CL of 12pF, which is fine when channels are running in jitter attenuator mode or referenced to an XO. It will present a positive ppm offset for channels running exclusively in Synthesizer mode and referenced only to the crystal.

Power Considerations

For power and current consumption calculations, refer to Renesas’ Timing Commander tool.

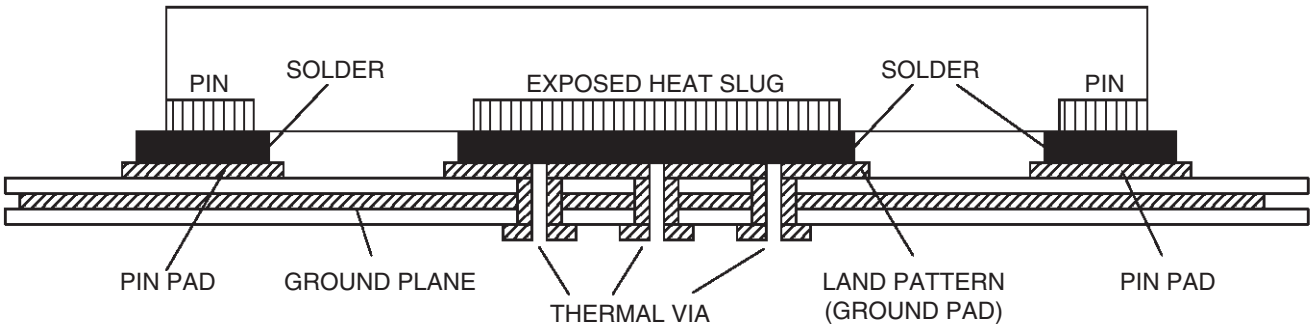
VFQFPN EPAD Thermal Release Path

In order to maximize both the removal of heat from the package and the electrical performance, a land pattern must be incorporated on the Printed Circuit Board (PCB) within the footprint of the package corresponding to the exposed metal pad or exposed heat slug on the package, as shown in Figure 54. The solderable area on the PCB, as defined by the solder mask, should be at least the same size/shape as the exposed pad/slug area on the package to maximize the thermal/electrical performance. Sufficient clearance should be designed on the PCB between the outer edges of the land pattern and the inner edges of pad pattern for the leads to avoid any shorts.

While the land pattern on the PCB provides a means of heat transfer and electrical grounding from the package to the board through a solder joint, thermal vias are necessary to effectively conduct from the surface of the PCB to the ground plane(s). The land pattern must be connected to ground through these vias. The vias act as “heat pipes”. The number of vias (i.e. “heat pipes”) are application specific and dependent upon the package power dissipation as well as electrical conductivity requirements. Thus, thermal and electrical analysis and/or testing are recommended to determine the minimum number needed.

Maximum thermal and electrical performance is achieved when an array of vias is incorporated in the land pattern. It is recommended to use as many vias connected to ground as possible. It is also recommended that the via diameter should be 12 to 13mils (0.30 to 0.33mm) with 1oz copper via barrel plating. This is desirable to avoid any solder wicking inside the via during the soldering process which may result in voids in solder between the exposed pad/slug and the thermal land. Precautions should be taken to eliminate any solder voids between the exposed heat slug and the land pattern. Note: These recommendations are to be used as a guideline only. For further information, please refer to the Application Note on the Surface Mount Assembly of Amkor’s Thermally/ Electrically Enhance Lead frame Base Package, Amkor Technology.

Figure 54. P.C. Assembly for Exposed Pad Thermal Release Path – Side View (Drawing Not to Scale)



Thermal Characteristics

Table 39. Thermal Characteristics for 72-pin QFN Package

Symbol	Parameter		Value	Units
θ_{JA}	Theta J_A. Junction to Ambient Air Thermal Coefficient[a][b]	0 m/s air flow	13.71	°C/W
		1 m/s air flow	10.67	°C/W
		2 m/s air flow	9.46	°C/W
θ_{JB}	Theta J_B. Junction to Board Thermal Coefficient[a]		0.702	°C/W
θ_{JC}	Theta J_C. Junction to Device Case Thermal Coefficient[a]		12.87	°C/W
-	Moisture Sensitivity Rating (Per J-STD-020)		3	

[a] Multi-Layer PCB with 2 ground and 2 voltage planes.

[b] Assumes ePAD is connected to a ground plane using a grid of 9x9 thermal vias.

Package Outline Drawings

The package outline drawings are located at the end of this document and are accessible from the Renesas website (see package links in Ordering Information). The package information is the most current data available and is subject to change without revision of this document.

Marking Diagram

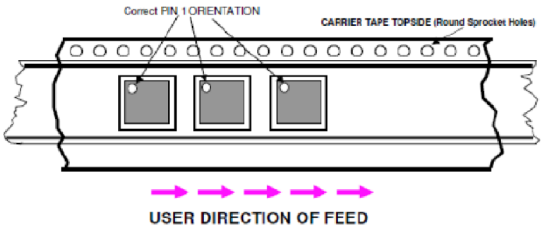
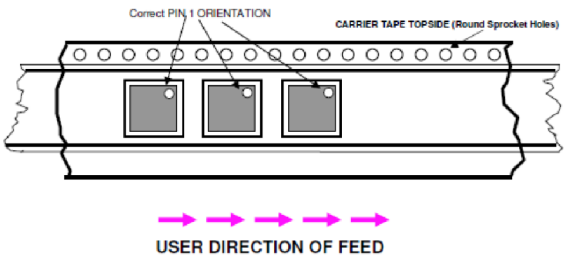
- Line 1 and 2 are the part number.
  - “000” denotes the dash code.
  - “NLG” denotes the package code.
- Line 3:
  - “YYWW” is the last digits of the year and week that the part was assembled
  - “$” denotes mark code.
- Line 4 denotes sequential lot number.

Ordering Information

Part Number	Package	MSL Rating	Carrier Type	Temperature Range
8A34002E-dddNLG[a]	10 × 10 × 0.9 mm, 72-VFQFPN	3	Tray	-40° to +85°C
8A34002E-dddNLG8	10 × 10 × 0.9 mm, 72-VFQFPN	3	Tape and Reel, Pin 1 Orientation: EIA-481-C	-40° to +85°C
8A34002E-dddNLG#	10 × 10 × 0.9 mm, 72-VFQFPN	3	Tape and Reel, Pin 1 Orientation: EIA-481-D	-40° to +85°C

[a] Replace “ddd” with the desired pre-programmed configuration code provided by Renesas in response to a custom configuration request or use “000” for unprogrammed parts.

Table 40. Pin 1 Orientation in Tape and Reel Packaging

Part Number Suffix	Pin 1 Orientation	Illustration
NLG8	Quadrant 1 (EIA-481-C)	
NLG#	Quadrant 2 (EIA-481-D)	

Product Identification

Table 41: Product Identification

Part Number	JTAG ID	Product_ID
8A34002	0x064F	0x4002

Glossary

Term	Definition
1PPS	One Pulse Per Second.
eCLK	Embedded clock.
eCSR	Embedded CSR access.
eDATA	Embedded DATA channel.
ePP2S	Embedded PP2S.
ePPS	Embedded PPS. This describes a means to embed 1PPS on a clock using PWM.
PPES	Pulse per even second.
ESEC	Even Second pulse. PP2S and ESEC are used interchangeably or sometimes combined as PP2S/ESEC.
eSYNC	Embedded SYNC pulse.
PP2S	Pulse Per 2 Second. This represents a 0.5Hz pulse.
PPS	Pulse Per Second.
REF-SYNC	Combination of high-speed clock (i.e., > 1MHz) and low-speed frame/sync pulse (i.e., < 8kHz).
SCSR	Standard Control / Status Register
ZDB	Zero Delay Buffer
ZDPLL	Zero Delay Phase Locked Loop

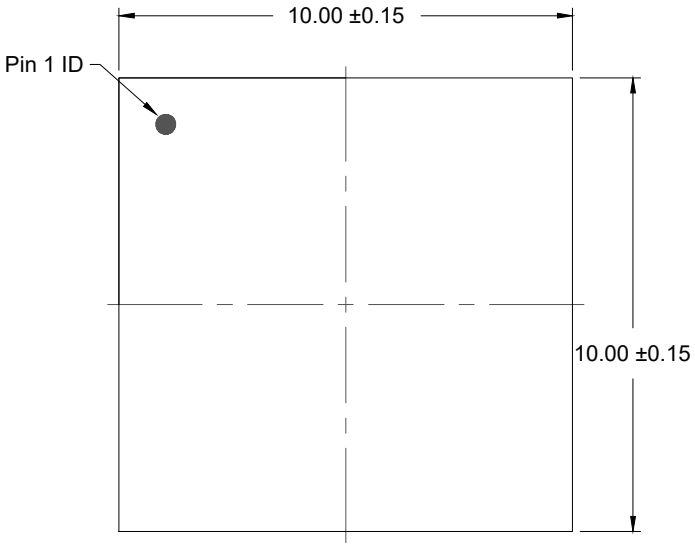
Revision History

Revision Date	Description of Change
August 13, 2023	▪ Updated JESD204B SYSREF Output Operation
December 20, 2021	▪ Replaced ePPS with PPES throughout the document ▪ Added patent numbers in footnote [1] ▪ Updated pin descriptions for pins 17 and 18 in Table 1 ▪ Updated descriptions of HS Type 1 and HS Type 2 in Hitless Reference Switching ▪ Updated Step 1 – Negation of nMR (Rising Edge) ▪ Added Accessing the Serial Ports After a Reset Sequence ▪ Updated Table 21 ▪ Added footnote to Table 24 ▪ Added SCL_M to VIH and VIL to Table 29 ▪ Updated V_{CMR} in Table 30 ▪ Updated footnote [c] in Table 30 ▪ Added footnote to Table 34 ▪ Updated footnotes [ag] and [ah] in Table 37 ▪ Added footnote [aa] to Table 37 ▪ Added footnote to tjit(Φ) in Table 37 ▪ Updated Output Phase Change due to Hitless Switching in Table 37 ▪ Updated $t_{STARTUP}$ in Table 37 ▪ Edited footnote [e] in Table 37 ▪ Added t_{LOCK} to Table 37 ▪ Updated Product Identification ▪ Completed other minor changes
September 8, 2020	▪ Clarified Figure 4 ▪ Expanded Frequency Representation ▪ Added information for minimum pulse width to Frame Pulse Operation and Sync Pulse Operation ▪ Corrected one acceptance range in Table 10 ▪ Clarified Hitless Reference Switching and added HS Type 1 and HS Type 2 ▪ Clarified Step 7 – Complete Configuration of the reset sequence ▪ Added description of satellite channels Satellite Channel ▪ Added crystal drive level to Table 36 ▪ Added Input Frequency minimum for GPIO in Table 35 ▪ Added Load Capacitance minimum in Table 36 ▪ Updated Phase Jitter, RMS in Table 37 ▪ Added Input-Output Alignment Variation for external feedback to Table 37 ▪ Added Input TDC Measurement Accuracy to Table 37 ▪ Added Pulse Width to Table 37 ▪ Added AC coupled LVDS termination to Differential Output Termination ▪ Added internal bias voltage to Overdriving the XTAL Interface ▪ Removed references to customer-programmable OTP ▪ Replaced System Analog PLL (APLL) and Low-Noise Analog PLL with System APLL throughout ▪ Replaced Steerable Fractional Divider with Steerable Fractional Output Divider throughout ▪ Added references to bit fields in the *8A3xxxx Family Programming Guide* to many sections and table headings

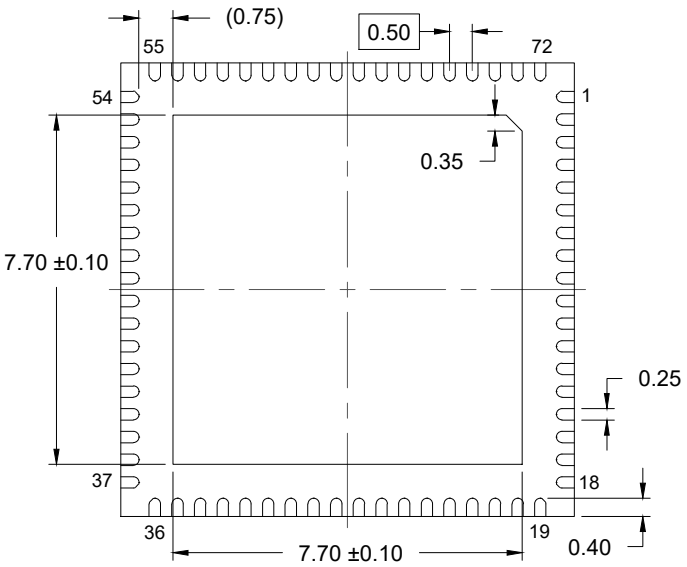
Revision Date	Description of Change
June 17, 2019	▪ Adjusted descriptions and expanded DPLL loop bandwidth limits ▪ Adjusted OSCI / OSCO input capacitance values in Table 2 ▪ Updated Table 7 to remove activity limits that were only available in obsolete Device Update revisions ▪ Added a description of External Feedback ▪ Fixed Increment / Decrement Registers and Pins to remove reference to a single 16-bit register that is not implemented ▪ Fixed inconsistency between DC Specifications table for CMOS mode output clocks and output buffer descriptive text with respect to 1.2V and 1.5V CMOS operation ▪ Removed temperature sensor due to inconsistent operation ▪ Added Power Supply Noise Rejection rows to AC characteristics in Table 37 ▪ Table 37Updated Tuning capacitor recommendations in Table 38 to limit crystal drive strength ▪ Added application information for 1.8V LVCMOS to over-drive the crystal input (see Overdriving the XTAL Interface) ▪ Updated marking diagram and ordering information to Revision E ▪ Removed separate marking / ordering code for parts that interact with IEEE-1588 software.
February 7, 2019	▪ Revision C device (which has Device Update v4.7 embedded) has the following functional differences: • Reset sequence sped-up, altering the way external EEPROMs are searched. Changes made to Reset Sequence and Use of External I^2C EEPROM sections • Changed Activity Monitor limits (this is with Device Update v4.7 regardless of hardware revision) in Table 7 ▪ Corrected the clock and GPIO mapping ▪ Clarified the following areas of the datasheet: Steerable Fractional Output Divider (FOD), JTAG Interface, and External I2C Serial EEPROM Recommendation ▪ Changed Marking Diagram and Package Outline Drawings to show C revision ▪ Added a “P” based part number identifier to Table 41
November 9, 2018	Added a missing revision letter to the Marking Diagram and Ordering Information
October 29, 2018	Initial release.

Package Outline Drawing

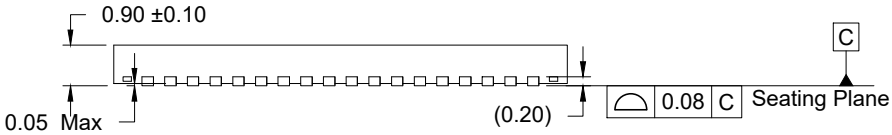
Package Code:NLG72P4
72-VFQFPN 10.0 x 10.0 x 0.90 mm Body, 0.50 mm Pitch
PSC-4208-04, Revision: 05, Date Created: Jul 6, 2023



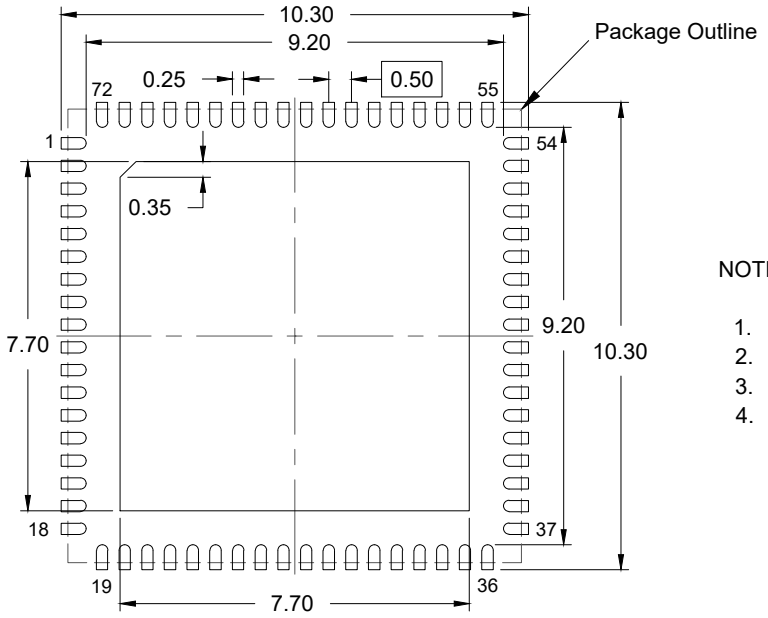
TOP VIEW



BOTTOM VIEW



SIDE VIEW



RECOMMENDED LAND PATTERN
(PCB Top View, NSMD Design)

NOTES:

1. JEDEC compatible.
2. All dimensions are in mm and angles are in degrees.
3. Use ±0.05 mm for the non-toleranced dimensions.
4. Numbers in ( ) are for references only.

IMPORTANT NOTICE AND DISCLAIMER

RENESAS ELECTRONICS CORPORATION AND ITS SUBSIDIARIES (“RENESAS”) PROVIDES TECHNICAL SPECIFICATIONS AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT OF THIRD-PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for developers who are designing with Renesas products. You are solely responsible for (1) selecting the appropriate products for your application, (2) designing, validating, and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. Renesas grants you permission to use these resources only to develop an application that uses Renesas products. Other reproduction or use of these resources is strictly prohibited. No license is granted to any other Renesas intellectual property or to any third-party intellectual property. Renesas disclaims responsibility for, and you will fully indemnify Renesas and its representatives against, any claims, damages, costs, losses, or liabilities arising from your use of these resources. Renesas' products are provided only subject to Renesas' Terms and Conditions of Sale or other applicable terms agreed to in writing. No use of any Renesas resources expands or otherwise alters any applicable warranties or warranty disclaimers for these products.

(Disclaimer Rev.1.01 Jan 2024)

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

Contact Information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit www.renesas.com/contact-us/.

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.