## AU5327: 8-Outputs, Quad PLL Frequency Translation, Jitter Attenuator

## General Description

AU5327 is a programmable Quad PLL based jitter attenuating clock synthesizer parts with flexible input to output frequency translation options. It supports up to 4 input clocks and provides 8 clock outputs. The clock outputs can be derived from the 4 PLLs in a highly flexible manner.

It is fully programmable with the I2C/SPI interface or an on-chip two time programmable non-volatile memory for factory preprogrammed devices.
Using advanced design technology, it provides excellent integrated jitter performance as well as low frequency offset noise performance while working reliably for ambient temperatures from $-40^{\circ} \mathrm{C}$ to $85^{\circ} \mathrm{C}$. The chip has best in class transient performance features in terms of clock switching transients and repeatable input to output delays.

Nomenclature:
AU5327BC1:VDDIN $=3.3 \mathrm{~V}, \mathrm{VDD}=1.8 \mathrm{~V} / 2.5 \mathrm{~V} / 3.3 \mathrm{~V}, \mathrm{VDDIO}=\mathrm{VDD}$ AU5327BC2:VDDIN $=3.3 \mathrm{~V}, \mathrm{VDD}=1.8 \mathrm{~V} / 2.5 \mathrm{~V} / 3.3 \mathrm{~V}, \mathrm{VDDIO}=\mathrm{VDDIN}$

Applications:

- Carrier Ethernet,
- OTN Equipment,
- Microwave Backhaul,
- Gigabit Ethernet
- Wireless Infrastructure,
- Network Line Cards
- Small Cells
- Data Center/Storage,
- SONET/SDH
- Test / Instrumentation,
- Broadcast Video


## Features

- Flexible quad PLL frequency translation from a common input: 4 fractional output domains from single input
- Fully Integrated Fractional N PLLs with integrated VCO and programmable loop filter (1 mHz to 4 kHz )
- Wide frequency support
- Differential Output from 8 KHz to 2.1 GHz
- Single Ended Output from 8 KHz to 250 MHz
- Support for 1 Hz frequency on one output
- Differential Input from 8 KHz to 2.1 GHz
- Single Ended Input from 8 KHz to 250 MHz
- Multiple Crystals / XO / TCXO / OCXO support
- LVPECL, CML, HCSL, LVDS and LVCMOS Outputs
- 150 fs typical rms integrated jitter performance
- Synchronized, holdover or free run operation modes
- Meets G. 8262 EEC Option 1,2(Sync E)
- Hitless input clock switching: Auto or manual - Sub 50 ps phase build out mode transients
- Phase Propagation with programmable slopes
- Frequency ramp for plesiochronous clocks with programmable slopes
- Robust and fast cycle slip and frequency step detection for input frequency steps (Clean frequency tracking for large frequency steps)
- Excellent Close-in Phase noise performance with no external discrete VCXOs or passive external filters
- Digitally Controlled Oscillator mode: to 0.005 ppb
- Programmable Output Delay Control
- Programmable Frequency Ramp Slopes for Switching Pleisochronous Clocks
- Indicators: Lock Loss, Clock Loss, Frequency Drift
- Repeatable Input to Output delays for each power up of chip
- Zero Delay Buffer mode also possible on any one PLL
- Output wake up sync with an independent clock also possible


Figure 1 Functional Overview

## Table of Contents

General Description .....  .1
Features .....  .1
Table of Contents .....  3
List of Tables .....  5
List of Figures .....  6
1 Pin Description .....  8
2 Electrical Characteristics ..... 13
3 Functional Description ..... 24
4 Master and Slaves: Architecture Description and Programming Procedures ..... 28
5 Input Slave Description ..... 32
6 Clock Monitor Slave Description ..... 33
6.1 Fault Monitoring System ..... 33
6.1.1 Clock Loss Monitors ..... 34
6.1.2 Frequency Drift Monitors ..... 34
6.1.3 Lock Loss Monitors ..... 35
6.1.4 XO Clock Loss Monitors ..... 35
7 Output Slave Description ..... 36
8 PLL Slave Description ..... 37
9 PLL Input Selection: Manual and Hitless Switching ..... 38
10 Zero Delay Mode ..... 39
11 PLL Bandwidth Control ..... 40
12 PLL Crystal Clock Reference ..... 41
13 PLL Lock Loss Defect Monitoring ..... 42
14 PLL DCO Mode operation ..... 43
15 Programmable Interface Top Level View ..... 44
16 Serial Programming Interface Description ..... 46
16.1 I2C protocol ..... 46
16.2 SPI Protocol ..... 48
16.3 SPI Timing Details ..... 49
16.3.1 SPI Single byte write ..... 50
16.3.2 SPI Single byte read ..... 50
17 Package Information ..... 52
18 Output Termination Information ..... 53
19 Input Termination Information ..... 58
20 Crystal Pathway Connectivity Options ..... 61
21 Monitoring through the register map read back: Status and Notify ..... 64
21.1 Tabular Listing ..... 64
21.2 Examples for Live Status Read Back ..... 68
21.3 Examples of Sticky Bit Clearing ..... 69
22 Device Initialization for a non-programmed device ..... 71
23 Monitoring the Status for Master and Slave Pages ..... 81
24 Programming the Primary E-Fuse ..... 83
24.1 Configuration Bits to Force Power-up of Digital Slave Subsystems ..... 83
24.2 E-Fuse Lock Configuration Bits ..... 84
24.3 E-Fuse Write Configuration Bits ..... 84
24.4 Configuration Bit to Remove Manual Wake Up for Primary E-Fuse ..... 84
24.5 Pseudo Code: Programming the Primary E-Fuse ..... 85
24.5.1 GENERIC_SYS ..... 85
24.5.2 INPUT_SYS ..... 85
24.5.3 CLKMON_SYS ..... 86
24.5.4 OUTPUT_SYS ..... 86
24.5.5 PLLA_SYS ..... 86
24.5.6 PLLB_SYS ..... 86
24.5.7 PLLC_SYS ..... 87
24.5.8 PLLD_SYS. ..... 87
24.5.9 Removing Manual Wake-up and Locking GENERIC_SYS E-Fuse ..... 87
25 Programming the Secondary E-Fuse ..... 89
25.1 Configuration Bit To Escape To PROGRAM_CMD State in GENERIC_SYS ..... 89
25.2 Configuration Bit to Change the E-Fuse pointer ..... 89
25.3 Configuration Bit to Enable Manual Wake Up for Secondary E-Fuse ..... 89
25.4 Psuedo Code: Programming the Secondary E-Fuse ..... 90
25.4.1 GENERIC_SYS ..... 90
25.4.2 INPUT_SYS ..... 90
25.4.3 CLKMON_SYS ..... 91
25.4.4 OUTPUT_SYS ..... 91
25.4.5 PLLA_SYS ..... 91
25.4.6 PLLB_SYS ..... 92
25.4.7 PLLC_SYS ..... 92
25.4.8 PLLD_SYS ..... 92
25.4.9 Programming GENERIC_SYS E-Fuse to Remove Manual Wake Up in Secondary E-Fuse. ..... 93
26 Register Map Details ..... 94
27 Ordering Information ..... 124
28 Revision History ..... 125

## List of Tables

Table 1 Detailed Pin Description .....  8
Table 2 Absolute Maximum Ratings ..... 13
Table 3 Operating Temperature, Thermal Characteristics and Environmental Compliance ..... 13
Table 4 DC Electrical Characteristics ..... 14
Table 5 Input Clock Characteristics ..... 15
Table 6 Serial and Clock Input ..... 16
Table 7 Output Serial and Status Pin ..... 16
Table 8 Output Clock Characteristics ..... 16
Table 9 Fault Monitoring Indicators ..... 18
Table 10 Crystal Requirements ..... 18
Table 11 Output RMS Jitter in Frequency Translation Modes ..... 19
Table 12 Close In Offset Phase Noise ..... 20
Table 13 Power Supply Rejection ..... 21
Table 14 Adjacent Output Cross Talk ..... 21
Table 15 Output Clock Specifications ..... 22
Table 16 PIF Description ..... 27
Table 17 PIF Overview (Top Level Summary of the Programmable Interface) ..... 44
Table 18 I2C Bus Timing Specifications ..... 48
Table 19 SPI Timing ..... 49
Table 20 Tabular Listing of Alarm Registers ..... 64
Table 21 Configuration Bits to Force Power-up of Digital Slave Subsystems ..... 83
Table 22 E-Fuse Lock Configuration Bits for all pages ..... 84
Table 23 E-Fuse Write Configuration Bits for All Pages ..... 84
Table 24 Configuration Bit to Remove Manual Wakeup for Primary E-Fuse ..... 85
Table 25 Configuration Bit To Escape to PROGRAM_CMD State in GENERIC_SYS ..... 89
Table 26 Configuration Bit to change the E-Fuse pointer ..... 89
Table 27 Configuration Bit to Enable Manual Wakeup for Secondary E-Fuse ..... 89
Table 28 Generic Master System Related Registers ..... 94
Table 29 PAGE 1: Clock Monitor System Related registers ..... 99
Table 30 PAGE 2 Input Systems Related Registers ..... 107
Table 31 PAGE 3 : Output System and Output Dividers Related Registers ..... 110
Table 32 Page A: PLL A Related Registers (Similar for Pages B, C, D) ..... 119
Table 33 Ordering Information for AU5327 ..... 124
Table 34 Revision History ..... 125

## List of Figures

Figure 1 Functional Overview .....  2
Figure 2 Top View .....  8
Figure 3 Representative Phase Noise Measurement ..... 20
Figure 4 Representative Close In Phase Noise Measurement ..... 21
Figure 5 Overall Architecture ..... 24
Figure 6 Overall Hierarchy of Clocks ..... 25
Figure 7 Input Clock Distribution ..... 25
Figure 8 PLL Dividers ..... 26
Figure 9 Output Clock Distribution ..... 26
Figure 10 Master Memory Structure ..... 28
Figure 11 Slave Memory Structure ..... 29
Figure 12 Master Wake-up Finite State Machine ..... 30
Figure 13 Slave Wake up Finite State Machine ..... 31
Figure 14 PLL Architecture ..... 37
Figure 15 Zero Delay Mode Set Up: PLLA in ZDB mode with OUTOT routed in to IN3 for the ZDB feedback... ..... 39
Figure 16 I2C Timing Waveforms ..... 48
Figure 17 SPI Timing Diagram ..... 49
Figure 18 SPI Write - AU5327 ..... 51
Figure 19 SPI Read - AU5327 ..... 51
Figure 20 64-QFN Package Dimensions ..... 52
Figure 21 LVPECL DC Termination to VDDO - 2V ..... 53
Figure 22 LVPECL Alternate DC Termination: Thevenin Equivalent ..... 53
Figure 23 DC Coupled LVDS Termination ..... 54
Figure 24 DC Coupled CML ..... 54
Figure 25 AC Coupled Receiver side resistive Termination options ..... 55
Figure 26 Alternate AC Coupled LVPECL with DC coupled resistors on Chip side ..... 56
Figure 27 DC Coupled LVCMOS ..... 56
Figure 28 HCSL AC Coupled Termination. Source Terminated $50 \Omega$ ..... 57
Figure 29 HCSL DC Coupled Termination. Source Terminated $50 \Omega$ ..... 57
Figure 30 AC Coupled Differential LVDS Input / Other AC Coupled Driver without DC Terminations ..... 58
Figure 31 AC Coupled Differential LVPECL or CML ..... 58
Figure 32 DC Coupled Single Ended Driver ..... 59
Figure 33 AC Coupled Single Ended Driver with 50 Ohm Termination on receiver (chip) side. ..... 59
Figure 34 AC Coupled Single Ended LVCMOS input without 50 Ohm Termination ..... 60
Figure 35 Crystal Connection ..... 61
Figure 36 CMOS XO Connection ..... 62
Figure 37 Differential XO Connection ..... 63
Figure 38 STEP 1: Initialize the Main Page- Page 0 ..... 72
Figure 39 STEP 2: Initialize the Input System Page- Page 2 ..... 73
Figure 40 STEP 3: Initialise the Clock Monitor System Page- Page 1 ..... 74
Figure 41 STEP 4: Initialise the Output System Page- Page 3 ..... 75
Figure 42 STEP 5: Initialise the PLL A System Page- Page A ..... 76
Figure 43 STEP 6: Initialise the PLL B System Page- Page B ..... 77
Figure 44 STEP 7: Initialise the PLL C System Page- Page C ..... 78
Figure 45 STEP 8: Initialise the PLL D System Page- Page D ..... 79

Figure 46 Active State Confirmation

## 1 Pin Description



Figure 2 Top View

Table 1 Detailed Pin Description

| Pin Name | I/O Type | Pin No. | Function | Comments |
| :--- | :--- | :---: | :--- | :--- |
| INOP | Input | 63 | $\begin{array}{l}\text { True input for INO differential } \\ \text { pair. Input for LVCMOS INO } \\ \text { input. Need series external } \\ \text { capacitor for differential input. }\end{array}$ | IN0 / IN1 / IN2 / IN3 inputs can be |
| used for output clock |  |  |  |  |
| synchronization. An active clock |  |  |  |  |
| and three spare clocks are chosen |  |  |  |  |
| for each PLL. There is complete |  |  |  |  |
| flexibility in the choice of the Active |  |  |  |  |
| and 3 Spares for each of the 4 |  |  |  |  |
| PLLs. |  |  |  |  |$]$| INON |
| :--- |
| Input |


| Pin Name | I/O Type | Pin No. | Function | Comments |
| :---: | :---: | :---: | :---: | :---: |
| IN2P | Input | 14 | True input for IN2 differential pair. Input for LVCMOS IN2 input. Need series external capacitor for differential input. | and 3 Spares for each of the 4 PLLs. |
| IN2N | Input | 15 | Complement input for IN2 differential pair. Ground with capacitor for LVCMOS IN2 input. Need series external capacitor for differential input. |  |
| IN3P | Input | 61 | True input for IN3 differential pair. Input for LVCMOS IN3 input. Need series external capacitor for differential input. |  |
| IN3N | Input | 62 | Complement input for IN3 differential pair. Ground with capacitor for LVCMOS IN3 input. Need series external capacitor for differential input. |  |
| GND | Power | EPAD | Electrical and Package Ground | Exposed Ground on the bottom EPAD |
| OUT1P | Output | 31 | Output 1 True Output or Output 1 LVCMOS. | LVPECL, LVDS, HCSL, CML and LVCMOS support. |
| OUT1N | Output | 30 | Output 1 Complement Output or Output 1 LVCMOS. |  |
| OUT2P | Output | 35 | Output 2 True Output or Output 2 LVCMOS. |  |
| OUT2N | Output | 34 | Output 2 Complement Output or Output 2 LVCMOS. |  |
| OUT3P | Output | 38 | Output 3 True Output or Output 3 LVCMOS. |  |
| OUT3N | Output | 37 | Output 3 Complement Output or Output 3 LVCMOS. |  |
| OUT5P | Output | 45 | Output 5 True Output or Output 5 LVCMOS. |  |
| OUT5N | Output | 44 | Output 5 Complement Output or Output 5 LVCMOS. |  |
| OUT6P | Output | 51 | Output 6 True Output or Output 6 LVCMOS. |  |
| OUT6N | Output | 50 | Output 6 Complement Output or Output 6 LVCMOS. |  |
| OUT7P | Output | 54 | Output 7 True Output or Output 7 LVCMOS. |  |
| OUT7N | Output | 53 | Output 7 Complement Output or Output 7 LVCMOS. |  |
| OUTOBP | Output | 24 | Output OB True Output or Output OB LVCMOS. |  |


| Pin Name | I/O Type | Pin No. | Function | Comments |
| :---: | :---: | :---: | :---: | :---: |
| OUTOBN | Output | 23 | Output 0B Complement Output or Output 0B LVCMOS. |  |
| OUTOTP | Output | 59 | Output 0T True Output or Output OT LVCMOS. |  |
| OUTOTN | Output | 58 | Output OT Complement Output or Output OT LVCMOS. |  |
| VDDIN | Power | 13 | Power Supply Voltage pin. | Decoupling capacitor close to supply pin required. |
| VDD | Power | 46,60 | Power Supply Voltage pin | Multiple Supply Pins, Decoupling capacitor close to each supply pin required. |
| VDDS | Power | 40 | Supply for determining VIH/VIL for FDEC for AU53x7. <br> Supply for determining LLAb and LLBb VOH/VOL for AU53x6. | Multiple Supply Pins, Decoupling capacitor close to each supply pin required. |
| LLAb | Output | 3 |  |  |
| LLBb | Output | 4 | Loss of Lock Indicator. Can be |  |
| LLCb | Output | 5 |  |  |
| RSTB | Input | 6 | Active low reset internally pulled up to VDDIO; Pull Up Resistor to VDDIO of fixed value ( $25 \mathrm{k} \Omega$ ). Can be left floating or pulled up to VDDIO if not used | Active low signal performs a complete reset of the part |
| OEOb | Input | 11 | Used to disable (when 1) all the output clocks. Can be left floating or pulled down to GND if not used |  |
| INTRb | Output | 12 | Active low indicator of programmable sticky notifies. Can be left floating if not used |  |
| SCLK | Input | 16 | I2C Serial Interface Clock or SPI Clock Input |  |
| SDO | Output | 17 | Serial Data Output (SPI Interface). In I2C mode this is the A1 address pin (see I2C section) |  |
| SDAIO | Input / Output | 18 | I2C Serial Interface Data (SDA) / SPI Input data (SDI) |  |
| CSB | Input | 19 | Chip Select Bar for the SPI Interface. In I2C mode this is the A0 address pin (see I2C section) |  |
| FDEC | Input/Output | 42 | DCO decrement. Can be left floating or pulled down to GND if not used |  |
| I2C1_SPI0 | Input | 39 | Choose between $\operatorname{SPI}(0)$ and I2C(1) interface being used, Pull |  |


| Pin Name | I/O Type | Pin No. | Function | Comments |
| :---: | :---: | :---: | :---: | :---: |
|  |  |  | Up Resistor to VDDIO of fixed value ( $25 \mathrm{k} \Omega$ ) |  |
| LLDb | Output | 47 | Loss of Lock Indicator. Can be left floating if not used. |  |
| LOS_XOb | Output | 25 | XO Loss of Lock Indicator. Can be left floating if not used. |  |
| PLL_SELO | Input | 26 | PLL selection for Pin based |  |
| PLL_SEL1 | Input | 27 | DCO. |  |
| FINC | Input/Output | 48 | DCO Increment. Can be left floating or pulled down to GND if not used |  |
| FLEXIO14 | Output | 55 | Flexible Outputs can be used for |  |
| FLEXIO15 | Output | 56 | (Refer AN53005A). Can be left floating or pulled down to GND if not used |  |
| \{X1, X1G\} | Input/Output | 8,7 | Crystal X1 Pin and accompanying ground pin | G, X2G\} land on a floatin |
| \{X2, X2G\} | Input/Output | 9,10 | Crystal X2 Pin and accompanying ground pin | Pe PC |
| VDDO1 | Power | 29 | Output Power Supply for Bank 1 outputs |  |
| VDDO2 | Power | 33 | Output Power Supply for Bank 2 outputs |  |
| VDDO3 | Power | 36 | Output Power Supply for Bank 3 outputs |  |
| VDDO5 | Power | 43 | Output Power Supply for Bank 5 outputs |  |
| VDDO6 | Power | 49 | Output Power Supply for Bank 6 outputs | Decoupling capacitor close to each supply pin required |
| VDDO7 | Power | 52 | Output Power Supply for Bank 7 outputs |  |
| VDDO0T | Power | 57 | Output Power Supply for Bank 8 outputs |  |
| VDDO0B | Power | 22 | Output Power Supply for Bank 9 outputs |  |
| NC | No Connect | $\begin{gathered} 20,21 \\ 28,32 \\ 41 \end{gathered}$ | No connect. This pin is not connected to the die. |  |

Notes:

1. VDDIO is the voltage used for all the status GPIOs and the serial interface. The default voltage for VDDIO can be chosen as either VDDIN or VDD through the programmable GUI(Enabled only on selected GUI variants).
2. Configure VDDIO as VDD and VDDIN using $0 \times 23[7]$ on Page0 as per below table:

| Variant | Default VDDIO Power Up | VDDIO=VDD | VDDIO=VDDIN |
| :---: | :---: | :---: | :---: |
| AU5327BC1 | VDD | $0 \times 23[7]=0$, Page0 (Default) | Not Supported |
| AU5327BC2 | VDDIN | Not Supported | $0 \times 23[7]=1$, Page0 (Default) |

3. All digital input/output GPIOs (FLEXIOs) have an on-chip $25 \mathrm{k} \Omega$ pull down resistor to ePAD ground (unless mentioned otherwise)and can be left unconnected if not used.
4. The I2C1_SPIO pad has a an on-chip $25 \mathrm{k} \Omega$ pull up resistor to indicate default mode of communication as I2C unless this pin is pulled down on the board to indicate the SPI mode.
5. In I2C mode, the serial data and clock have an on-chip $25 \mathrm{k} \Omega$ pull up resistor to VDDIO.
6. The RSTB pin has an on-chip $25 \mathrm{k} \Omega$ pull up resistor to VDDIO. Writing $0 \times \mathrm{FFE}[0]$ to 1 with delay additon of 10 ms has the same effect as the pulling RSTB pin to GND for chip reset.
7. SDO and CSB pins are used to set the I2C default address as $0 \times 6 \mathrm{D}$ when floating since SDO and CSB has 25k pull down and pull up to GND and VDDIN respectively. Otherwise the I2C address can be changed as $11011\{$ SDO \},\{CSB $\}$ by forcing the SDO and CSB externally to VDDIN or GND accordingly.
a. The chip can be reset from the register map by writing address $0 \times F E$ as $0 \times 01$ using the current I2C address.
b. To disable reset from register map by writing $0 \times F E$ register as $0 \times 00$, Address needs to be 0b11010\{SDO\}\{CSB\}, 5 MSB address bits are 11010, LSB 2 bits are the state of SDO and CSB pins. If these pins are floating, use $0 x 69$ as the address. At all other times default slave address chosen for the part can be used.

## 2 Electrical Characteristics

Table 2 Absolute Maximum Ratings

| Description | Conditions | Symbol | Min | Typ | Max | Units |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Core supply voltage, Analog Input |  | Vdin | -0.5 |  | +3.63 | V |
| Core supply voltage, PLL |  | $V_{D D}$ | -0.5 |  | +3.63 |  |
| Output bank supply voltage |  | V ${ }_{\text {do }}$ | -0.5 |  | +3.63 | V |
| Input voltage, All Inputs | Relative to GND | VIN | -0.5 |  | +3.63 | V |
| XO Inputs | Relative to GND | Vxo | -0.5 |  | +1.4 | V |
| I2C Bus input voltage | SCLK, SDAT pins | VINI2C | -0.5 |  | +3.63 | V |
| SPI Bus input voltage |  | VINSPI | -0.5 |  | +3.63 | V |
| Storage temperature | Non-functional, Non-Condensing | Ts | -55 |  | +150 | ${ }^{\circ} \mathrm{C}$ |
| Programming Temperature |  | Tprog | +25 |  | +85 | ${ }^{\circ} \mathrm{C}$ |
| Maximum Junction Temperature in Operation |  | TJct |  |  | +125 | ${ }^{\circ} \mathrm{C}$ |
| Programming Voltage (for Programming the OTP Fuse Memory). |  | $V_{\text {PRog }}$ | 2.375 | 2.5 | 2.625 | V |
| ESD (human body model) | JESD22A-114 | ESDнвм |  |  | 2000 | V |
| Latchup | JEDEC JESD78D | LU |  |  | 100 | mA |
| Moisture Sensitivity Level | 64-QFN | MSL | 3 |  |  |  |

1. Exceeding maximum ratings may shorten the useful life of the device.
2. Stresses beyond those listed under Absolute Maximum Ratings may cause permanent damage to the device. These are stress ratings only and functional operation of the device at these or at any other conditions beyond those indicated under the DC Electrical Characteristics is not implied. Exposure to Absolute-Maximum-Rated conditions for extended periods may affect device reliability or cause permanent device damage.
3. Refer to AN53012 for the additonal information on the absolute minimum and maximum voltage on XO Inputs before and after the chip power up.

Table 3 Operating Temperature, Thermal Characteristics and Environmental Compliance

| Description | Conditions | Symbol | Min | Typ | Max | Units |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Ambient temperature |  | TA | -40 | - | +85 | ${ }^{\circ} \mathrm{C}$ |
| Junction temperature |  | TJ |  |  | +125 | ${ }^{\circ} \mathrm{C}$ |
| Thermal Resistance Junction to Ambient | Still Air | $\theta_{\text {JA }}$ | 25.5 |  |  | ${ }^{\circ} \mathrm{C} / \mathrm{W}$ |
|  | Air Flow 1m/s |  | 20.8 |  |  | ${ }^{\circ} \mathrm{C} / \mathrm{W}$ |
|  | Air Flow 2m/s |  | 19.6 |  |  | ${ }^{\circ} \mathrm{C} / \mathrm{W}$ |
| Thermal Resistance Junction to Case |  | $\theta_{\text {Jc }}$ | 8.70 |  |  | ${ }^{\circ} \mathrm{C} / \mathrm{W}$ |
| Thermal Resistance Junction to Board |  | $\theta_{\text {Јв }}$ | 7.07 |  |  | ${ }^{\circ} \mathrm{C} / \mathrm{W}$ |
| Thermal Resistance Junction to Top Center |  | $\Psi_{J T}$ | 0.2 |  |  | ${ }^{\circ} \mathrm{C} / \mathrm{W}$ |
| Thermal Resistance Junction to Top Center |  | $\Psi J T$ | 0.2 |  |  | ${ }^{\circ} \mathrm{C} / \mathrm{W}$ |

Table 4 DC Electrical Characteristics

| Description | Conditions | Symbol | Min | Typ | Max | Units |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Supply voltage, Analog Input Pathways and XTAL Pathways | 3.3 V range: $\pm 10 \%$ | Vdoin | 2.97 | 3.3 | 3.63 | V |
| Supply voltage, PLL | 1.8 V range: $\pm 5 \%$ | VDD | 1.71 | 1.80 | 1.89 | V |
|  | 2.5 V range: $\pm 5 \%$ |  | 2.375 | 2.50 | 2.625 | V |
|  | 3.3 V range: $\pm 10 \%$ |  | 2.97 | 3.3 | 3.63 | V |
| Supply Voltage, Output Drivers | 1.8 V range: $\pm 5 \%$ | VDDo | 1.71 | 1.80 | 1.89 | V |
|  | 2.5 V range: $\pm 5 \%$ |  | 2.375 | 2.50 | 2.625 | V |
|  | 3.3 V range: $\pm 10 \%$ |  | 2.97 | 3.3 | 3.63 | V |
| Programmable Status <br> Supply for selected IOs | 1.8 V range: $\pm 5 \%$ | VDDS | 1.71 | 1.80 | 1.89 | V |
|  | 2.5 V range: $\pm 5 \%$ |  | 2.375 | 2.50 | 2.625 | V |
|  | 3.3 V range: $\pm 10 \%$ |  | 2.97 | 3.3 | 3.63 | V |
| Total Power Dissipation (2.5V LVDS Outputs @ 156.25M) | 4 PLLs, 8 Outputs | Pd |  | 930 | 1116 | mW |
|  | 1 PLL, 2 Outputs |  |  | 250 | 300 | mW |
| Supply Current, VDDIN | All Four Inputs assumed to be enabled | $\operatorname{ldDIN}^{[1]}$ |  | 18 | 21.6 | mA |
| Supply Current, VDD | All Four PLLs and All 8 Outputs enabled <br> (Maximum current mode for AU5327) | IDD |  | 320 | 384 | mA |
| Power supply current, VDDO | LVPECL, output pair terminated $50 \Omega$ to $V_{T T}$ (VDDO - 2 V ). | $\mathrm{IDDo}^{[2,3,4,5,6]}$ |  | 40 | 48 | mA |
|  | LVPECL2, output pair terminated $50 \Omega$ to $V_{T T}$ (VDDO -2 V ) or 0 V without common mode current. |  |  | 28 | 36 | mA |
| Power supply current, VDDO | CML, output pair terminated $50 \Omega$ to VDDO | IdDo <br> [2,3,4,5,6,7] |  | 20 | 24 | mA |
| Power supply current, VDDO | HCSL, output pair with HCSL termination | Iddo <br> $[2,3,4,5,6,7]$ |  | 27 | 36 | mA |
| Power supply current, VDDO | LVDS, output pair terminated with an AC or DC Coupled diff $100 \Omega$ | IdDo <br> [2,3,4,5,6,7] |  | 16 | 19.2 | mA |
| Power supply current, VDDO | LVDS Boost, output pair terminated with an AC or DC Coupled diff $100 \Omega$ | IdDo <br> [2,3,4,5,6,7] |  | 20 | 24 | mA |
| Power supply current, VDDO | LVCMOS, 250 MHz , 2.5 V output, $5-\mathrm{pF}$ load | $\begin{aligned} & \text { lDDO } \\ & {[2,3,4,5,6,7]} \end{aligned}$ |  | 15 | 18 | mA |

## Notes:

1. $\mathrm{VDDIN}=3.3 \mathrm{~V}$ and $\mathrm{VDD}=1.8 \mathrm{~V} / 2.5 \mathrm{~V} / 3.3 \mathrm{~V}$ is the recommended supply combination. Additional current consumption of 3 mA for a third overtone crystal instead of a fundamental mode crystal.
2. LVPECL and LVDS Boost standards are supported for VDDO $=\{2.5 \mathrm{~V}, 3.3 \mathrm{~V}\}$. LVPECL2, HCSL, CML and LVDS standards are supported for VDDO $=\{1.8 \mathrm{~V}, 2.5 \mathrm{~V}, 3.3 \mathrm{~V}\}$.
3. LVPECL mode provides 6 mA of common mode current on each output. LVPECL2 mode does not provide this common mode current.
4. A $50 \Omega$ Termination resistor with a DC bias of VDDO -2 V for LVPECL standards is supported for VDDO $=\{2.5 \mathrm{~V}, 3.3 \mathrm{~V}\}$.
5. IDDOx Output driver supply current specified for one output driver in the table. This includes current in each of the output module that includes output dividers, drivers and clock distributions.
6. The LVDS Boost Mode and the LVDS Mode can be used for AC Coupled output terminations. LVDS Boost provides an LVPECL like swing with an AC Coupled $100 \Omega$ differential termination.
7. Refer to the Output Termination Information in the data sheet for the description of the various terminations that are supported.
8. For efuse programming in AU5327, VDD alongwith VDDIN can be set to 2.5 V and has no reliability concerns. Refer to Programming the Primary E-Fuse section for VDD/VDDIN voltage information for efuse programming.

Table 5 Input Clock Characteristics

| Parameter | Conditions | Symbol | Min | Typ | Max | Unit |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Standard Input Buffer with Differential or Single-Ended - AC-coupled (IN0/IN0, IN1/IN1, IN2/IN2, IN3/IN3) |  |  |  |  |  |  |
| Input Frequency Range | Differential | $\mathrm{fin}^{\text {m }}$ | 0.008 | - | 2100 | MHz |
|  | All Single-ended signals (including LVCMOS) |  | 0.008 | - | 250 | MHz |
| Voltage Swing (Differential Amplitude Peak or Single Ended Peak to Peak for the differential signal) ${ }^{[1]}$ | AC-coupled $\mathrm{f}_{\mathrm{IN}}<400 \mathrm{MHz}$ | $\mathrm{V}_{\text {IN }}$ | 100 | - |  | mV |
|  | $\begin{aligned} & 400 \mathrm{MHz}<\text { AC-coupled } \\ & \mathrm{f}_{\mathrm{IN}}<750 \mathrm{MHz} \end{aligned}$ |  | 225 | - |  | mV |
|  | 750 MHz < AC-coupled fIN $<2100 \mathrm{MHz}$ |  | 350 | - |  | mV |
| Single Ended AC Coupled Inputs (Single Ended Peak to Peak Input) ${ }^{[1 /[1 / 4]}$ | AC-Coupled $\mathrm{f}_{\mathrm{IN}}<250 \mathrm{MHz}$ |  | 500 | - | 3600 | mV |
| Slew Rate ${ }^{[2,3]}$ |  | SR | 400 | - | - | V/us |
| Duty Cycle |  | DC | 40 | - | 60 | \% |
| Input Capacitance |  | $\mathrm{C}_{\text {IN }}$ | - | 0.3 | - | pF |
| Input Resistance | AC Coupled SE | $\mathrm{R}_{\text {IN }}$ | - | 15 | - | k $\Omega$ |
|  | Differential |  | - | 10 | - | k $\Omega$ |
| Pulsed CMOS Input Buffer - DC-coupled (INO, IN1, IN2, IN3) ${ }^{[3]}$ |  |  |  |  |  |  |
| Input Frequency |  | $\mathrm{f}_{\text {in_pulsed_cmos }}$ | 0.008 | - | 250 | MHz |
| Input Voltage |  | $\mathrm{V}_{\text {IL }}$ | -0.2 | - | 0.4 | V |
|  |  | $\mathrm{V}_{\mathrm{IH}}$ | 0.8 | - | - | V |
| Slew Rate ${ }^{[2,3]}$ |  | SR | 400 | - | - | V/us |
| Duty Cycle |  | DC | 40 | - | 60 | \% |
| Minimum Pulse Width | Pulse Input | PW | 1.6 | - | - | ns |
| Input Resistance |  | $\mathrm{R}_{\mathrm{IN}}$ | - | 30 | - | k $\Omega$ |
| Reference Clock (Applied to X1), Can be external XO, TCXO or OCXO |  |  |  |  |  |  |
| Reference Clock Frequency | Range for best jitter | Fin_ref | 48 | - | 160 | MHz |
|  | Overall supported range |  | 37.5 | - | 160 | MHz |
| Input Voltage Swing | Single Ended peak to peak | $\mathrm{V}_{\text {IN_SE }}$ | 365 | - | 2000 | mVpp_se |
|  | Differential peak to peak | VIN_DIFF | 365 | - | 2500 | mVpp_diff |
| Slew rate |  | SR | 400 | - | - | V/us |
| Duty Cycle |  | DC | 40 | - | 60 | \% |

Notes:

1. AC Coupled input assumed with series capacitance for differential inputs or single ended AC Coupled inputs. Swing requirement at device pins.
2. Resistor termination for differential input followed by series capacitors for each of true and complement differential input connecting to the device pins.
3. LVCMOS single ended is direct coupled on the true input. Connect complement input to ground with a 100 nF capacitor.
4. Single Ended AC coupled Input Swing requirement (Single Ended Peak to Peak Input) [1][4] is for optimal noise performance.

Table 6 Serial and Clock Input

| Parameter | Condition | Symbol | Min | Typ | Max | Unit |
| :--- | :--- | :--- | :---: | :---: | :---: | :---: |
| Input Voltage |  | $\mathrm{V}_{\text {IL }}$ | - | - | $0.3 \times \mathrm{V}_{\text {DDIO }}{ }^{1}$ | V |
|  |  | $\mathrm{~V}_{\mathrm{IH}}$ | $0.7 \times \mathrm{V}_{\text {DDIO }}{ }^{1}$ | - | - | V |
| Input Capacitance |  | $\mathrm{CIN}_{I N}$ | - | 1 | - | pF |
| Input Resistance |  | $\mathrm{RIN}^{2}$ | - | 25 | - | $\mathrm{k} \Omega$ |
| Minimum Pulse Width | FINC, FDEC | PW | 100 | - | - | ns |
| Update Rate | FINC, FDEC | FUR | - | - | 1 | $\mu \mathrm{~s}$ |

Notes:

1. VDDIO is the voltage used for all the status GPIOs and the serial interface. The default voltage for VDDIO can be chosen as either VDDIN or VDD with a hard coded eFuse based selection.

Table 7 Output Serial and Status Pin

| Parameter | Test Condition | Symbol | Min | Typ | Max | Unit |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| All VDDIO based GPIOs |  |  |  |  |  |  |
| Output Voltage | $\mathrm{I}_{\mathrm{OH}}=-2 \mathrm{~mA}$ | Voh | $\mathrm{V}_{\text {DDIO }} \times 0.75$ | - | - | V |
|  | $\mathrm{l}_{\mathrm{OL}}=2 \mathrm{~mA}$ | VoL | - | - | $\mathrm{V}_{\text {DIIO }} \times 0.25$ | V |
| All VDD based GPIOs |  |  |  |  |  |  |
| Output Voltage | $\mathrm{I}_{\mathrm{OH}}=-2 \mathrm{~mA}$ | Voh | $V_{\text {DDS }} \times 0.75$ | - | - | V |
|  | $\mathrm{I}_{\mathrm{OL}}=2 \mathrm{~mA}$ | VoL | - | - | $V_{\text {DDS }} \times 0.25$ | V |

Notes:

1. VDDIO is the voltage used for all the status GPIOs and the serial interface. The default voltage for VDDIO can be chosen as either VDDIN or VDD with a hard coded eFuse based selection.

Table 8 Output Clock Characteristics

| Description | Conditions | Symbol | Min | Typ | Max | Units |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Differential output frequency | LVPECL, CML, LVDS outputs | $\mathrm{Fout,OIFF}^{[1]}$ | 1 |  | 2100M | Hz |
| Differential output frequency | HCSL outputs | $\mathrm{Fout,DiFFH}^{[1]}$ | 1 |  | 700 M | Hz |
| Single ended output frequency | LVCMOS outputs | $\mathrm{F}_{\text {oun,SE }}{ }^{[1]}$ | 1 |  | 250 M | Hz |
| PLL loop bandwidth | Programmable | $\mathrm{F}_{\mathrm{BW}}$ | 0.001 |  | 4000 | Hz |
| Jitter peaking | Meets SONET Jitter Peaking requirements in closed loop | $J_{\text {PEAK }}$ |  |  | 0.1 | dB |
| Time delay before the Historical average for output Frequency is considered. | Programmable in register map | $\mathrm{H}_{\text {DELAY }}{ }^{[2,3]}$ | 0.035 | 0.5 | 35 | s |
| Length of time for which the Average of the frequency is considered | Programmable in register map | $\mathrm{H}_{\text {AVG }}{ }^{[2,3]}$ | 0.07 | 1 | 70 | s |
| Power Supply to I2C or SPI interface ready | No I2C transaction valid till 10 ms after all power supplies are ramped to $90 \%$ of final value. | $\mathrm{T}_{\text {Start }}$ |  |  | 10 | ms |
| With Speed-Up mode enabled | Speed-up mode is programmable. This is a Typical number. Actual wake up time depends on fast lock and normal BW settings | $\mathrm{T}_{\text {Lock }}{ }^{[4]}$ |  | 300 |  | ms |


| Description | Conditions | Symbol | Min | Typ | Max | Units |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| DCO Mode Frequency <br> Step <br> Resolution | Frequency Increment or Decrement resolution. This is controlled through the register map. | $\mathrm{F}_{\text {RES, DCo }}{ }^{[5]}$ |  | 0.005 |  | ppb |
| Resolution for output delay | Programmable per output clock with this resolution for a total delay of $\pm 7.5 \mathrm{~ns}$ | $\mathrm{T}_{\text {RES }}{ }^{[6]}$ |  | 35 |  | ps |
| Maximum Phase Hit | Default Hitless Switching Mode (no phase propagation) | $\mathrm{T}_{\text {MAX }}{ }^{[7]}$ | -50 |  | 50 | ps |
| Uncertainty in Input to Output Delay | Maximum variation in the static delay from input to output clock between repeated power ups of the chip | $\Delta T_{\text {deLay }}$ | -175 |  | 175 | ps |
| Pull Range |  | $\omega_{\text {P }}$ |  | 500 |  | ppm |
| POR to Serial Interface Ready |  | $\mathrm{T}_{\text {Ror }}$ |  |  | 15 | ms |
| Input to Output Delay in ZDB mode (matched pathways on external feedback, INO input, IN3 feedback) |  | $\mathrm{T}_{\text {zDELAY }}{ }^{\text {[8] }}$ |  | 100 |  | ps |
| Temperature Variation of delay in ZDB mode |  | $\mathrm{T}_{\text {ZDELAY, TMP }}{ }^{[8]}$ |  |  | 1 | ps/C |
| One free run PLL clock on fuse locked parts | Using a special mode for fuse locked parts to generate one free run output from one PLL | $\mathrm{T}_{\text {Start,Special }}$ |  |  | 10 | ms |

Notes:

1. 1 Hz Output Available only on output OUTOB (OUTOBP, OUTOBN). Range supported is 8 kHz to 2100 MHz for all the other outputs.
2. Hitless Switching enables PLL to switch between input clocks when the current clock is lost,
a. Clock Loss can be defined as 2 / 4 / 8 / 16 consecutive missing pulses.
b. Priority list for the input clocks can be set in the register map independently for each PLL.
c. Output is truly hitless (no phase transient and 0 ppb relative error in frequency) for exactly same frequency input clocks that are switched.
d. Hitless switching support is both revertive and non-revertive
i. Revertive / Non-revertive Support: Assume Clock Input 0 is lost and switch is made to Clock Input 1. Then, PLL reverts to Clock Input 0 when it becomes valid again in Revertive mode. It does not switch back to Clock Input 0 even when it becomes valid again in the non-Revertive mode.
3. PLL enters holdover mode when the active input clock and all spare clocks in the clock priority list for hitless switching are lost,
a. Clock Loss can be defined as 2 / 4 / 8 / 16 consecutive missing pulses
b. Programmable Clock Loss settings ensure Gapped Clocks can be supported by choosing higher number of missing pulses as the trigger for clock being invalid
c. Entering hold over mode is supported with the frequency frozen at a historical average determined from the $H_{\text {delay }}$ and $H_{\text {avg }}$ settings.
4. For low PLL Loop Bandwidths, wake up time can be very large unless the speed up feature is used. The speed up feature provides the user options to use a completely independent loop bandwidth for the wake up transitioning to the regular bandwidth after frequency and phase are locked.
a. Fast Lock Bandwidth needs to be less than 100 times smaller than the input clock frequency (divided input at PLL phase detector) for stable and bounded (in time) lock trajectory of the PLL
5. The 0.005 ppb specification is for the smallest frequency step resolution available. Larger frequency step resolutions up to 100 ppm can be used also. The frequency resolution for the DCO mode frequency step is independently programmable for each DCO step.
6. All output clocks from one specific PLL are phase aligned. Relative delay adjustment is then possible on each clock individually as defined by the $T_{\text {RES }}$ parameter.
7. This test is for 2 inputs at 8 M that are switched to get a 622.08 M output.
8. Both input and feedback at 8 MHz with the delays exactly matched and same slew for both for the chip

Table 9 Fault Monitoring Indicators

| Description | Conditions | Symbol | Min | Typ | Max | Units |
| :--- | :--- | :--- | :--- | :--- | :--- | :--- |
|  | Clock Loss Indicators <br> can be set on any of <br> the four inputs. Loss of <br> $2 / 4 / 8 / 16$ |  |  |  |  |  |
| Clock Loss Indicator <br> Thresholds | consecutive pulses can <br> be used to indicate a <br> clock loss. <br> Programmable in the <br> register map. | CLx ${ }^{11,4]}$ |  |  |  |  |

Table 10 Crystal Requirements

| Description | Conditions | Symbol | Min | Typ | Max | Units |
| :--- | :--- | :--- | :---: | :---: | :---: | :---: |
| High Fundamental Frequency Crystal Reference (HFF) |  |  | 160 | MHz |  |  |
| Crystal Frequency | Can be supported with a <br> fundamental crystal of <br> 100-160 MHz range. | XTALIN | 100 |  | 2 | pF |
| C0 cap for crystal |  | XTALco |  |  | 5 |  |
| CL cap for crystal | Small range around CL <br> only | XTALcL |  | 5 | pF |  |
| ESR for crystal | ESR defined at <br> frequency of oscillation | XTALESR |  |  | 40 | $\Omega$ |
| Rm1 for crystal |  | XTALRm1 |  |  | 20 | $\Omega$ |


| Description | Conditions | Symbol | Min | Typ | Max | Units |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Power delivered to crystal | Drive Level to the crystal | XTALpwr |  | 100 |  | $\mu \mathrm{W}$ |
| Third Overtone Crystal Reference (OT3) |  |  |  |  |  |  |
| Crystal Frequency | Can be supported with an OT3 crystal of $100-160 \mathrm{MHz}$ range. | XTALIN | 100 |  | 160 | MHz |
| C0 cap for crystal |  | XTALco |  |  | 2 | pF |
| CL cap for crystal | Small range around CL only | XTALcl |  | 5 |  | pF |
| ESR for crystal | ESR defined at frequency of oscillation | XTALEsR ${ }^{[1]}$ |  |  | 80 | $\Omega$ |
| Rm3 for crystal |  | XTALRm3 |  |  | 40 | $\Omega$ |
| Power delivered to crystal | Drive Level to the crystal | XTALpwr |  | 100 |  | $\mu \mathrm{W}$ |
| Low Frequency Fundamental Crystal (LFF) |  |  |  |  |  |  |
| Crystal Frequency | Can be supported with a fundamental crystal $>37.5 \mathrm{MHz}$ range. For Best Performance use an LFF crystal $>48 \mathrm{MHz}$ | XTALIn | 48 |  | 54 | MHz |
| C0 cap for crystal |  | XTALco |  |  | 2 | pF |
| CL cap for crystal | Small range around CL only | XTALcl |  | 8 |  | pF |
| ESR for crystal | ESR defined at frequency of oscillation | XTALesr ${ }^{[1]}$ |  |  | 60 | $\Omega$ |
| Rm1 for crystal |  | XTALRm1 |  |  | 40 | $\Omega$ |
| Power delivered to crystal | Drive Level to the crystal | XTALpwr |  | 100 |  | $\mu \mathrm{W}$ |
| Notes: <br> 1. ESR relates to the motional resistance Rm with the relationship $E S R=R m(1+C 0 / C L) 2$ |  |  |  |  |  |  |

Table 11 Output RMS Jitter in Frequency Translation Modes

| Description | Conditions | Symbol | Min | Typ | Max | Units |
| :--- | :--- | :--- | :---: | :---: | :---: | :---: |
| RMS Jitter for <br> $12 \mathrm{kHz}-20 \mathrm{MHz}$ <br> Integration Bandwidth | Fout $=622.08 \mathrm{MHz}$ |  |  | 140 |  | fs rms |
| FIN $=38.88 \mathrm{MHz}$, <br> PLL BW $=100 \mathrm{~Hz}$, <br> Single PLL Profile | Fout $=156.25 \mathrm{MHz}$ |  |  |  |  |  |

1. For best noise performance in jitter attenuation mode, use lowest usable loop bandwidth for the PLL.
2. Does not include noise from the input clocks to the PLL


Figure 3 Representative Phase Noise Measurement
Note: FOUT $=622.08 \mathrm{MHz}, \mathrm{FIN}=38.88 \mathrm{MHz}, \mathrm{BW}=100 \mathrm{~Hz}$, FREF $=54 \mathrm{M} \mathrm{XO}$

Table 12 Close In Offset Phase Noise

| Description | Conditions | Symbol | Min | Typ | Max | Units |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Phase Noise Skirt <br> $\mathrm{F}_{\text {OUT }}=122.88 \mathrm{MHz}$, <br> PLL BW $=100 \mathrm{~Hz}$ | Offset Frequency $=100 \mathrm{~Hz}$ | PN ${ }^{[1]}$ |  | -113 |  | $\mathrm{dBc} / \mathrm{Hz}$ |
|  | Offset Frequency $=1 \mathrm{kHz}$ |  |  | -130 |  |  |
|  | Offset Frequency $=10 \mathrm{kHz}$ |  |  | -138 |  |  |

## Notes:

1. This is the noise contribution of the chip only without including the input and reference self contributions


Figure 4 Representative Close In Phase Noise Measurement
Note: FOUT = 122.88M, BW = 100 Hz , FREF and FIN are provided from R\&S SMA100 equipment to ensure a low close in phase noise for the reference and input to illustrate the chip contribution to close in phase noise.

Table 13 Power Supply Rejection

| Description | Conditions | Symbol | Min | Typ | Max | Units |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| $\begin{aligned} & \text { Fout }=156.25 \mathrm{MHz}, \\ & \text { Fspur }=100 \mathrm{kHz}, \\ & \text { BW }=100 \mathrm{~Hz} \\ & \text { PSRR on VDD Supply } \end{aligned}$ | $\mathrm{V} D \mathrm{D}=1.8 \mathrm{~V}$ | PSRRvdd |  | -75 |  | dBc |
| $\begin{aligned} & \text { Fout }=156.25 \mathrm{MHz}, \\ & \text { Fspur }=100 \mathrm{kHz}, \\ & \text { BW }=100 \mathrm{~Hz} \\ & \text { PSRR on VDDIN Supply } \end{aligned}$ | $\mathrm{V} D \mathrm{DIN}=3.3 \mathrm{~V}$ | PSRR vidin |  | -100 |  | dBc |
| $\begin{aligned} & \text { Fout }=156.25 \mathrm{MHz}, \\ & \text { Fspur }=100 \mathrm{kHz}, \\ & \text { BW }=100 \mathrm{~Hz} \\ & \text { PSRR on VDDO Supply } \end{aligned}$ | $\mathrm{VDDO}=3.3 \mathrm{~V}$ | PSRRvddo |  | -80 |  | dBc |

## Notes:

1. The PSRR is measured with a 50 mVpp sinusoid in series with the supply and checking the spurious level relative to the carrier on the output in terms of phase disturbance impact.
2. Output PSRR measured with LVDS standard which (along with the LVDS boost) are the recommended standards for AC Coupled terminations

Table 14 Adjacent Output Cross Talk

| Description | Conditions | Symbol | Min | Typ | Max | Units |
| :--- | :---: | :--- | :--- | :---: | :---: | :---: |
| 156.25 M and 155.52 M <br> on adjacent outputs | AU5327 | $X_{\text {TALK }}$ |  | -75 |  | dBc |

Notes:

- Measured across adjacent outputs- All adjacent outputs are covered and the typical value for the worst case output to output coupling is reported.
- The adjacent output pairs are chosen at 155.52 MHz and 156.25 MHz frequencies.
- This cross talk between outputs is mainly package dependent therefore terminated outputs are used for reporting these numbers ensuring that there is signal current in the bond wires.

Table 15 Output Clock Specifications

| Descriptions | Conditions | Symbol | Min | Typ | Max | Units |
| :--- | :--- | :--- | :--- | :--- | :---: | :---: |
| DC Electrical Specifications - LVCMOS output (Complementary Out of Phase Outputs or One CMOS Output per Output Driver) |  |  |  |  |  |  |
| Output High Voltage | 4 mA load, VDD $=3.3 \mathrm{~V}$ | $\mathrm{~V}_{\text {OH }}$ | VDDO-0.3 |  | - | V |
| Output High Voltage | 4 mA load, <br> VDD $=1.8 \mathrm{~V}$ and 2.5 V | $\mathrm{~V}_{\text {OH }}$ | VDDO-0.4 |  | - | V |
| Output Low Voltage | 4 mA load | $\mathrm{V}_{\text {OL }}$ |  |  | 0.3 | V |


|  |  |  |  |  |  |  |
| :--- | :--- | :--- | :--- | :--- | :--- | :---: |
| DC Electrical Specifications - LVCMOS output (In Phase Outputs) |  |  |  |  |  |  |
| Output High Voltage | 4 mA load, VDD $=3.3 \mathrm{~V}$ | $\mathrm{~V}_{\mathrm{OH}}$ | VDDO-0.35 |  | - | V |
| Output High Voltage | 4 mA load, VDD $=2.5 \mathrm{~V}$ | $\mathrm{~V}_{\mathrm{OH}}$ | VDDO-0.45 |  | - | V |
| Output High Voltage | 4 mA load, VDD $=1.8 \mathrm{~V}$ | $\mathrm{~V}_{\mathrm{OH}}$ | VDDO- 0.5 |  | - | V |

DC Electrical Specifications - LVDS Outputs (VDDO = 1.8 V, 2.5 V or 3.3 V range)

| Output Common-Mode <br> Voltage | VDDO $=2.5 \mathrm{~V}$ or 3.3 V <br> range | V осм | 1.125 | 1.2 | 1.375 | V |
| :--- | :--- | :--- | :---: | :---: | :---: | :---: |
| Output Leakage Current | Output Off, <br> VOUT $=0.75 \mathrm{~V}$ to 1.75 V | loz | -20 | 20 | $\mu \mathrm{~A}$ |  |

DC Electrical Specifications - LVPECL Outputs (VDDO $=2.5 \mathrm{~V}$ or 3.3 V range)

| Output High Voltage | $\begin{aligned} & \text { Rterm }=50 \Omega \text { to } \\ & \text { VTT(VDDO }-2.0 \mathrm{~V} \text { ) } \end{aligned}$ | $\mathrm{V}_{\mathrm{OH}}$ | VDDO-1.165 |  | VDDO-0.800 | V |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Output Low Voltage | Rterm $=50 \Omega$ to <br> VTT(VDDO - 2.0 V ), w/o common mode current | $\mathrm{V}_{\mathrm{OL}}$ | VDDO-2.0 |  | VDDO-1.45 |  |
| AC Electrical Specifications - HCSL Outputs (VDDO = 1.8 V, 2.5 V or 3.3 V range) |  |  |  |  |  |  |
| Output High Voltage Max | Measurement on singleended signal | $\mathrm{V}_{\text {MAX }}$ |  |  | 1150 | mV |
| Output Low Voltage Min | Measurement on singleended signal | $\mathrm{V}_{\text {MIN }}$ | -300 |  |  | mV |
| Differential Voltage | Measurement taken from differential waveform | VP | 300 |  |  | mV |
| Absolute Crossing point voltage | Measurement taken from single ended waveform | $V_{\text {Cross }}$ | 250 |  | 600 | mV |
| Variation of VCROSS over all rising clock edges | Measurement taken from single ended waveform | $\mathrm{V}_{\text {crossdelta }}$ |  |  | 140 | mV |
| DC Electrical Specifications - CML Outputs (VDDO = 1.8 V, 2.5 V or 3.3 V range) |  |  |  |  |  |  |
| Output High Voltage | Rterm $=50 \Omega$ to VDDO | $\mathrm{V}_{\mathrm{OH}}$ | VDDO-0.085 | VDDO-0.01 | VDDO | V |
| Output Low Voltage | Rterm = $50 \Omega$ to VDDO | $\mathrm{V}_{\mathrm{OL}}$ | VDDO-0.6 | VDDO-0.4 | VDDO-0.3 | V |


| AC Electrical Specifications LVCMOS Output Load: $10 \mathrm{pF}<100 \mathrm{MHz}, 7.5 \mathrm{pF}<150 \mathrm{MHz}, 5 \mathrm{pF} \boldsymbol{>} \mathbf{1 5 0 ~ M H z ~ > ~} 200 \mathrm{MHz}$ |  |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: |
| Output Frequency |  | fout | 8k | 250M | Hz |
| Output Duty cycle | Measured at $1 / 2$ VDDO, loaded, <br> fOUT < 100 MHz | $t_{\text {DC }}$ | 45 | 55 | \% |
| Output Duty cycle | Measured at $1 / 2$ VDDO, loaded, <br> fOUT > 100 MHz | toc | 40 | 60 | \% |
| Rise/Fall time | $\text { VDDO }=1.8 \mathrm{~V}, 20-80 \%,$ <br> Highest Drive setting | $\mathrm{t}_{\text {frcmos }}$ |  | 2 | ns |
| Rise/Fall time | $\text { VDDO }=2.5 \mathrm{~V}, 20-80 \%,$ <br> Highest Drive setting | $\mathrm{t}_{\text {frcmos }}$ |  | 1.5 | ns |
| Rise/Fall time | $\text { VDDO }=3.3 \mathrm{~V}, 20-80 \%,$ <br> Highest Drive setting | $\mathrm{t}_{\text {frcmos }}$ |  | 1.2 | ns |
| AC Electrical Specifications (LVPECL, LVDS, CML) |  |  |  |  |  |
| Clock Output Frequency |  | fout | 8k | 2100 M | Hz |
| PECL Output Rise/Fall Time | $20 \%$ to $80 \%$ of AC levels. Measured at 156.25 MHz for PECL outputs. | $\mathrm{t}_{\text {RF }}$ |  | 350 | ps |
| CML Output Rise/Fall Time | $20 \%$ to $80 \%$ of AC levels. Measured at 156.25 MHz for CML outputs | $t_{\text {RF }}$ |  | 350 | ps |


| Descriptions | Conditions | Symbol | Min | Typ | Max | Units |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| LVDS Output Rise/Fall Time | $20 \%$ to $80 \%$ of AC levels. Measured at 156.25 MHz for LVDS outputs. | $t_{\text {RF }}$ |  |  | 350 | ps |
| Output Duty Cycle | Measured at differential $50 \%$ level, 156.25 MHz | todc | 45 | 50 | 55 | \% |
| LVDS Output differential peak | Measured at 156.25 M Output | VP | 300 | 350 | 454 | mV |
| Boosted LVDS Output differential peak | Measured at 156.25 M Output | VP | 500 | 700 | 950 | mV |
| LVPECL Output Differential peak | Measured at 156.25M Output | VP | 450 | 750 | 900 | mV |
| CML Output Differential Peak | Measured at 156.25M Output | VP | 250 |  | 600 | mV |

## Notes:

Convention for Wave Forms

Convention for Waveforms


## 3 Functional Description



Figure 5 Overall Architecture

AU5327 are a family of jitter attenuating frequency translation device offers four independent PLLs. The four clock inputs can map to any of the four PLLs. The output high frequency voltage controlled oscillators (VCOs) associated with each PLL are mapped to 8 outputs in a very flexible fashion. This offers a very flexible frequency translation arrangement with independent control of each PLL in terms of jitter attenuation, bandwidth control and input clock selection with redundancy.

The hierarchy of the clocks, nomenclature of the various frequency dividers as well as the clock translation pathways available on the chip are shown in the Figure 6.


Figure 6 Overall Hierarchy of Clocks

The four input clocks with frequencies fin_extk translate to PLL input clocks fink following division by the respective input dividers with fractional or integer frequency division ratios DIVN1k where the index $\mathrm{k} \in\{0,1,2$, 3\}. Each of the PLLs chooses one of the four divided input clocks fink as its active input clock and sets the priority for up to three spare clocks from the remaining three input clocks if required for hitless switching to a redundant input.


Figure 7 Input Clock Distribution

The crystal oscillator reference input (called fref) is also routed to each PLL. A TCXO or OCXO based input reference clock can also be used directly in place of the crystal oscillator. Each PLLx ( $x \in\{A, B, C, D\}$ ) has a high frequency VCO whose frequency is determined in the free run mode by fref with the relation $\mathrm{fVCOx}=$ DIVN $x^{*}$ fref. In the frequency translation synchronized mode, the VCO frequency is corrected from its free run frequency to
satisfy the relation $\mathrm{fVCOx}=$ DIVN2x*finx where finx is chosen from one of fink input clocks per the desired input clock priority for PLLx. Nominally the fractional dividers DIVNx and DIVN2x are chosen such that the relation DIVNx*fref $=$ DIVN2x*fin, $\mathrm{x}=\mathrm{fVCO}$ is satisfied.


Figure 8 PLL Dividers

Each of the Output Drivers (ODRj, $\mathrm{j} \in\{0,1,2,3,5,6,7,0 \mathrm{~T}, \mathrm{OB}, 1 \mathrm{~B}\}$ ) then chooses an appropriate VCO frequency and divides it using their respective integer divider DIVOj to get the output frequency foutj.


Figure 9 Output Clock Distribution

The choice of the fractional dividers \{DIVN1k, DIVNx, DIVN2x, DIVOj\} as well as the placement of foutj frequencies at various outputs is facilitated by associated software tools.

The digital architecture of the chip is partitioned into a master digital controller and seven slave controllers. The master controller and each of the seven controllers has an associated volatile programmable interface (PIF). The overall PIF structure is a register map that is divided into several pages according to function. Each controller (master and slaves) has an associated unique Page number. Each Page has an independent 8 bit addressable PIF memory. In all the pages, the last address, FF, holds the current page number and is reserved for changing the page. The current page to be communicated with can be set by writing the page number in hexadecimal form $\{0 \times 00,0 \times 01,0 \times 02,0 \times 03,0 \times 0 \mathrm{~A}, 0 \times 0 \mathrm{~B}, 0 \times 0 \mathrm{C}, 0 \times 0 \mathrm{D}\}$ corresponding to pages $\{0,1,2,3, \mathrm{~A}, \mathrm{~B}, \mathrm{C}, \mathrm{D}\}$ in the address FF on any page. Table 16 shows a summary of the PIF contents residing on each page.

Table 16 PIF Description

| Page | Contents | Summary of contents |
| :---: | :---: | :---: |
| 0 | Master | All Generic Information related to the chip Chip Configuration details Control for the master sequencer FSM Crystal Reference Related Information Fuse Pointer for each of the remaining pages |
| 1 | ClkMon Slave | Clock Loss related function Frequency Drift related function |
| 2 | Input Slave | Input 3/2/1/0 related information (Input type, DIVN1 divider configuration) |
| 3 | Output Slave | Flexible Outputs $7 / 6 / 5 / 4 / 3 / 2 / 1 / 0$ <br> (ODR Standards, DIVO, Programmable delay configurations for each) <br> Fixed Outputs 0T / OB <br> (ODR Standards, DIVO, Programmable delay configurations for each) |
| A | PLL A Slave | All PLL related functionality |
| B | PLL B Slave | All PLL related functionality |
| C | PLL C Slave | All PLL related functionality |
| D | PLL D Slave | All PLL related functionality |

## 4 Master and Slaves: Architecture Description and Programming Procedures

The Master controller is the first system to autonomously wake up on the application of power to the chip due to on-chip power on reset circuitry. All generic system information resides in the Master controller memory and it proceeds to wake up the Slaves as required based on this information. The relative wake up sequences of the Master and the various Slaves are described in more detail later in this section after a description of the memory structures. A complete power up of the chip is also emulated with the release of an active low hard reset (RSTB) from pin while selective Master and Slave sub-system resets are enacted from software using the serial interface (I2C / SPI).
The Master memory structure is shown in Figure 10. It contains a one-time programmable nonvolatile memory (NVM) that stores the settings for the chip associated with the master controller. The master controller also contains a volatile PIF bank (NVMCopy) that has an exact copy of the nonvolatile memory (NVM) at every chip power up. This volatile PIF (NVMCopy) is the memory that is addressable using the serial interface (I2C / SPI) on Page 0 and can be overwritten from the I2C / SPI interface. The "Chip Settings" is the memory space that is not addressable from the I2C / SPI control and is the actual control for the chip. The NVM contains a two bit "Lock Pattern" that can be set to " 10 " or " 01 " to 'lock' the chip configuration once the final configuration is determined and wake up of the entire chip is desired in this configuration. Additionally, there is a bit in the NVM that is an active low indicator of a manual wake up. This bit set to "1" along with the 'lock' for the configuration leads to an autonomous wake up of the chip using the 'locked' configuration. Any number of different configurations can alternatively be tried at all times using only the volatile NVMCopy PIF section. This is useful for evaluations as well as allowing real time programming of the chip in various configurations with complete flexibility. The Master Controller finite state machine (FSM) described later in this section controls the device behavior in accordance with the configuration in this memory structure and as per the wake up mode


Figure 10 Master Memory Structure

The memory structure for each slave is shown in Figure 11 and is similar in construction to the master controller memory structure with some minor differences. The NVMCopy volatile PIF for the slave is addressable by the serial interface with the unique Page number associated with the slave. The "Slave Settings" is the memory space that is not addressable from the I2C / SPI control and is the actual control for the slave. Each Slave has a two time programmable NVM by virtue of two copies of the NVM memory. This makes the slave settings two time programmable with the fuse pointer from the master controller determining which of the two NVM banks is used. The presence of two NVM banks is transparent to the slave controller since the current pointer which determines which of the two NVM banks is used is set by the master controller independently.


Figure 11 Slave Memory Structure

The Master Wake-Up Finite State Machine (FSM) is shown in more detail in Figure 12. At every power up of the device (or release from hard reset), the power-on-reset circuitry resets all systems and then autonomously releases only the master controller from reset. The NVM contents are copied to the NVMCopy volatile space on Page 0 which is in turn copied to the "Chip Settings". The master controller now decides if the chip configuration is locked and it is an autonomous wake up of the entire chip or if a manual wake up is desired through the PIF based on the contents in the "Chip Settings".

In case a "Lock" is detected and an autonomous wake up is desired, the Master controller proceeds to enable the Crystal oscillator and associated fref pathways followed by the Slave systems in a pre-determined sequence. This finally leads the chip to the "Active State" with all desired outputs available as a result of all slave systems released from reset by the master controller. This is according to the requested settings that are programmed in the Master and the Slave NVM banks.

For the case where the final chip settings are not frozen hence the "Lock" pattern is not exercised, the master controller FSM reaches the Program Command Wait State (PRG_CMD). The desired chip settings can be written in the NVMCopy on Page 0 using the serial interface and desired slave sub-systems can be enabled. Several PRG_CMD state directives are available that are exercisable only in this state (refer Figure 12). Using these directives, the desired settings written in NVMCopy can now be copied to "Chip Settings" followed by issuing the directive for the FSM to proceed to the "Active" state where each slave can now be manually written with the desired settings and in turn asked to proceed to its "Active" state.

A similar "Lock" pattern is available in the NVM bank of each slave. The currently used NVM bank for a slave (as determined by the current pointer from the master controller) can be locked for the autonomous wake up of each slave. The slave wake-up FSM is shown in Figure 13 and it similarly has a PRG_CMD state with associated directives. On Proceed to Active state directive on the slave, the slave controller wakes up the various blocks in its sub-system with the correct pre-determined sequence.

The NVM bank for the master and each slave can be programmed with a PRG_CMD directive in that state to lock a configuration / setting specific to the respective sub-system.


Figure 12 Master Wake-up Finite State Machine


Figure 13 Slave Wake up Finite State Machine

Each FSM (Master and Slaves) allows an escape sequence to go back to PRG_CMD state from its Active State. This can be used to selectively change the settings for that particular sub-system. Such an escape to the PRG_CMD state in the master FSM can be used for example to change current NVM pointers for any of the slaves.

Note that the NVM for the master controller and current NVM for all slaves should be locked after writing desired settings for a completely autonomous wake up of the entire chip. The NVM pointer can then be changed for any slave independently if alternate settings are desired for that slave. In that case, the new NVM is unlocked and can be written with new settings and locked. For evaluations of the chip as well as cases where flexible on-thefly programmable settings are desired, the chip can be used without engaging the NVM banks at all by using the NVMCopy space for the master and each slave in conjunction with the PRG_CMD directives. It is also possible to lock some of the slaves (to not re-write their settings for each wake up) while use programmable settings for other slaves.

This provides complete flexibility in terms of programming and using the chip in all scenarios.

## 5 Input Slave Description

Four independent clock inputs are available on the chip that can be routed to any PLL with complete flexibility. Both single ended and AC coupled differential clock inputs are possible. The input clock receiver settings (to receive a single ended or differential clock) as well as the input clock divider settings are configurable on Page 2 that is assigned to the Input Slave. It is possible to bypass the input clock divider and use the input clock directly as an input to the PLL.

## 6 Clock Monitor Slave Description

Various fault monitoring indicators are available on the chip. The Clock Loss and the Frequency Drift indicators are configurable with the Clock Monitor Slave that is accessible on Page 1. The specifications of these fault monitors are indicated in the specifications section of the data sheet.

Defect monitoring on any of the clock monitors can be accessed using multiple techniques. The current status of the defect is available as an Active High defect that can be read from the PIF. The "status" is a current indicator of the defect that is high only during the defect (for example during the time that a Clock Loss event is on-going). Additionally, a sticky indicator of the defect called "Notify" can be enabled in the PIF. In this case, the concerned "notify" bit is high the first time the respective defect occurs and stays high till cleared.

There are multiple FLEXIOs (Flexible IOs) available in the system that can be programmed to monitor individual "notify" signals or a combination of them (as an OR logic). The choice of which fault defect is monitored as an output on the FLEXIO pin is flexible and can be programmed. Additionally there are selected GPIOs that are hard coded for the information for the clock defects.

### 6.1 Fault Monitoring System

The AU5327 parts provide an elaborate arrangement of fault monitoring indicators. There are 4 categories of clock monitoring that are necessary for the chip namely: Clock Loss Monitor (CL), Frequency Drift Monitor (FD), Lock Loss Monitor (LL) and XO Clock Loss Monitor (CL_XO).

Clock Loss (CL) monitors loss of input clocks defined as a pre-determined number of consecutive edges missing. Frequency Drift (FD) monitors frequency drift of a particular clock against a pre-determined Golden Reference. Lock Loss (LL) monitors the loss of lock in any PLL by monitoring the difference in frequency between the feedback and input clocks.
XO Clock Loss (CL_XO) monitors the loss of the XO reference that is generated from either an external oscillator (XO / TCXO / OCXO) or using the on chip XO amplifier that can work with a crystal blank on the PCB.

Each of these categories monitors the health of a particular clock for a certain failure type as illustrated in the name of the clock monitoring category.

For each clock failure observed by the clock monitor block there are two types of indicators provided to the user using the register map:
Live Failure Bit: There is a bit to indicate the live status of a particular failure. [Status]
Sticky Failure Bit: For each live failure bit there is a corresponding sticky bit that is set the first time that corresponding failure is encountered and stays set even if the failure has gone away. Only when the user clears the bit does it clear. [Notify]

The status of these can be either read from the register map or from the pins as a dynamic alarm monitoring arrangement. Additionally, sticky notify registers are available which have sticky status read back from the register map for the various defects. These can be selectively chosen to create an INTRB de-assertion on the INTRB pin as well.

An important point to note is that all of the fault monitoring indicators mentioned above that work with respect to the input clock work on the divided input clock post the DIVN1,k dividers. This implies that the fault monitoring indicators use the frequency fink that is input to the PLL ( $k \in\{0,1,2,3\}$ ) post the DIVN1,k divider translation rather than the external frequencies fin_extk ( $\mathrm{k} \in\{0,1,2,3\}$ ).

Section 21 describes the read back of the alarms for the various fault monitoring arrangements using the chip register map.

### 6.1.1 Clock Loss Monitors

Each of the 4 inputs (IN0, IN1, IN2, IN3) are monitored for Clock Loss in terms of missing edges to indicate a loss of input signal. The number of edges used to indicate a clock loss (or recovery from a clock loss) is programmable in the AU5327 GUI interface allowing for flexibility in choosing these thresholds. In addition there is a programmable "Wait Time" all of which are to be interpreted as follows:

## Assertion of Clock Loss-

We declare a CL if "Trigger Edge" number of consecutive edges are missing. The "Trigger Edge" parameter is programmable in the chip GUI.

## De-Assertion of Clock Loss-

We declare a $\sim$ CL if the clock is back and has less than "Clear Edge" consecutive edges missing. The "Clear Edge" parameter is programmable in the chip GUI.

Wait Time: After the clock is established to have returned, it is ensured that no CL error as defined by the deassertion threshold occurs for "Val Time" seconds. This valid wait time is programmable using the chip GUI using the "Val Time" parameter which is programmable from the following options: $\{2 \mathrm{~m}, 100 \mathrm{~m}, 200 \mathrm{~m}, 1\} \mathrm{sec}$. The use of the this valid wait time ensures that sporadic edges in the input clock (such as ones caused by noise on floating nodes or intermittent unstable clock edges) does not de-assert clock loss and it is established over a user determined period of time that the input clock is available and stable.

### 6.1.2 Frequency Drift Monitors

Any one of the 4 input clocks or the XO clock can be used as the Golden Clock for calculating the frequency drifts of the other 4 clocks. The Golden Clock can be chosen in the GUI and is used as the " 0 ppm" Reference Clock for all monitoring.

Fine Frequency Drift has a step size of $\pm 2 \mathrm{ppm}$.
Fine Frequency Drift has a range of $\pm 2$ to $\pm 510 \mathrm{ppm}$ and an independent threshold is programmable for "Set" (for setting the FD monitor) and for "Clear" (for clearing the FD monitor).
Fine Frequency Drift has an implicit hysteresis with resolution of $\pm 2 \mathrm{ppm}$ since the same range is available for the FD assertion and de-assertion. Use of hysteresis prevents unwanted oscillation of the FD monitor output at the decision threshold and is recommended for robust operation. The value of the FD threshold hysteresis is implicit in the choice of the set and clear thresholds.
The Fine Frequency Drift monitors provide precise information for input clock frequency drift. However, since the resolution of the measurement determines time for the measurement- an alternate faster measurement mechanism for drift is needed. This is Coarse Frequency Drift which has coarser measurement but is fast. It is available for cases where the drift is very fast in the input frequency and is programmable from options as shown below.

Coarse Frequency Drift has a step size of $\pm 100$ ppm.
Coarse Frequency Drift has a range of $\pm 100$ to $\pm 1600 \mathrm{ppm}$ and an independent threshold is programmable for "Set" (for setting the FD monitor) and for "Clear" (for clearing the FD monitor).
Coarse Frequency Drift has an implicit hysteresis with resolution of $\pm 100 \mathrm{ppm}$ since the same range is available for the FD assertion and de-assertion. Use of hysteresis prevents unwanted oscillation of the FD monitor output
at the decision threshold and is recommended for robust operation. The value of the FD threshold hysteresis is implicit in the choice of the set and clear thresholds.

## Important Note regarding the above monitors with respect to clock switch in the PLL:

Normally the CL monitor is used for ascertaining a clock is lost for the PLL to switch to a secondary reference or proceed to Holdover. However, the Fine and/or Coarse FD monitors can also be used in addition to the CL monitor to cause a PLL switch. This implements an "OR" logic for the FD Monitors to be used in addition to the CL monitors for triggering a PLL input clock switch or entry to Holdover. This is programmable as an option in the GUI.

### 6.1.3 Lock Loss Monitors

Lock loss is programmable for each PLL with lock loss triggered if the frequency of the input reference to the PLL phase detection arrangement and the feedback clock to same PLL are different as per the programmed assertion and de-assertion thresholds.

The Set threshold for asserting the LL monitor is programmable from $\{ \pm 0.2, \pm 0.4, \pm 2, \pm 4, \pm 20, \pm 40, \pm 200$, $\pm 400, \pm 2000, \pm 4000\}$ ppm while the Clear threshold for de-asserting the LL monitor is programmable from $\{ \pm 0.2$, $\pm 0.4, \pm 2, \pm 200\}$ ppm. A pre-determined level of hysteresis is implicit by choosing appropriately the set and clear thresholds for the LL monitor.

Additionally from the point of view of LL de-assertion, there is a delay from the point in time that lower than the specified ppm value is achieved to the point where the actual LL is de-asserted to the user such that LL never asserts during this delay period. The choice of this delay is with a timer that ensures that the delay is in line with the BW of the PLL loop. It is fully programmable from the GUI and is useful to ensure complete settling of the PLL without un-necessary toggling before LL de-assertion.

### 6.1.4 XO Clock Loss Monitors

The XO Clock Loss Monitor asserts the XO Clock Loss Alarm when the external reference input to the X1 pin (XO or TCXO or OCXO) or the internal XO clock generated with the crystal blank is not available.

## 7 Output Slave Description

The Output Slave accessible on Page 3 is used to configure the output divider (DIVO) and output standard for each output individually. The output load and terminations for each differential output standard are shown in the Output Terminations section of the data sheet. The LVDS and LVDS Boosted modes are recommended for AC coupled termination loads with the termination at the far end. Additionally, an internal termination mode for differential outputs is available where the resistive terminations are internally provided and a differential output is available that can be AC coupled to a clock receiver. The differential clock output pins are shared for LVCMOS outputs as well. LVCMOS outputs can be either enabled on both outputs individually or on any one of the two differential outputs \{OUTjP, OUTjN\}. The LVCMOS outputs can be used in-phase or out-of-phase on \{OUTjP, OUTjN\} in case both outputs are chosen. Out of phase LVCMOS toggling on the complementary outputs is recommended for best spur performance.

## 8 PLL Slave Description

All settings with respect to each PLLx slave $(x \in\{A, B, C, D\})$ are accessible on the respective Page $\{A, B, C, D\}$. The PLL architecture is shown in Figure 14. There are three distinct modes of operation of the PLL: free run mode, synchronized mode and holdover mode. The frequency of the high frequency VCO in the PLL is determined by the specific mode of operation. The VCO frequency is then divided down to get the output frequency on the ODR as described with relation to the overall hierarchy of clocks described earlier.

The PLL in the free run mode can be described as a crystal based oscillator where the output frequency is determined by the relation $\mathrm{fVCOx}=$ DIVNx*fref. This is the mode of operation before the loop is locked to the selected input clock or the mode of operation for the case none of the input clocks is available. After locking to the chosen input clock, the PLL enters the synchronized mode of operation where the output is now locked to the input frequency with the relation fVCOx = DIVN2x*finx. The PLL Loop that synchronizes (locks) the output to the input clock has a programmable loop bandwidth between 1 mHz to 4 KHz and is not affected by static or dynamic drifts in the crystal oscillator based fref frequency. In case the input clock is lost, the PLL locks to the highest priority spare clock available. If all specified input clocks are lost, the PLL remembers the correction based on historical average of the input clock as specified to enter the Holdover mode of operation.

In synchronized mode, the PLL is also able to lock to a Gapped Input clock with some edges missing producing a smooth output clock without any gaps with the requested frequency translation from input to output. Frequency translation ratios in this case should be specified with respect to the average input frequency of the gapped clock rather than the faster instantaneous frequency.


Figure 14 PLL Architecture

## 9 PLL Input Selection: Manual and Hitless Switching

Each PLL Slave chooses its clock priority in terms of the four input clocks. This is programmed in to the Clock Monitor slave memory. The PLL Slave then looks at Clock Loss status from the Clock Monitor slave to lock to the highest priority available clock to lock. Three spare clocks with an order of priority can be specified in case the highest priority active clock is not available. Additionally, a forced manual selection of the active clock with no spares is possible. Less than three spares can also be specified making the clock priority arrangement completely flexible in terms of choosing the input clock for operation.

Phase Build Out Mode of hitless switching ensures that phase transients are not propagated to the output (the phase difference between redundant input clocks is absorbed by the PLL) and desired MTIE characteristics are seen in the output clock. This is the default mode of hitless switching for the PLL. The transition of input clock for a PLL from one clock to another is hitless in nature (with maximum phase hit limited to be less than 50 ps ) for the case of the switched input clocks being same in frequency. Hitless switch is also supported for the switched clocks being fractionally related such that the same frequency can be obtained for both clocks at the input of the PLL using the input clock dividers (DIVN1k).

For redundant input clocks to the PLL that are not exactly the same frequency (plesichronous clocks), the frequency ramp feature can be enabled that ramps the output frequency of the PLL at a slope that is programmable to one of the following 4 settings: $\{0.2,2,20,40000\}$ ppm/s. For redundant input clocks to the PLL that are exactly the same frequency, the frequency ramp feature should not be enabled.

An alternate mode of hitless switching is the Phase Propagation mode where the phase difference between redundant input clocks is not absorbed by the PLL but is rather propagated to the output. The phase difference that is propagated to the output can either be allowed to propagate as per the PLL bandwidth or can be limited to a phase propagation slope that is programmable to one of the following 3 settings: $\{10,40,160\} \mathrm{us} / \mathrm{s}$.

Zero Delay Buffer mode is available on any of the 4 PLLs by routing the output clock back to the IN3 input. This ensures minimum delay between the input and output. It can be used for one of the four PLLs at any time.

## 10 Zero Delay Mode

A zero delay mode is available and can be configured for any one of the PLLs in the chip. This provides the option to close the feedback loop of the PLL on the PCB and therefore bypasses the internal feedback dividers cancelling therefore the delays introduced by internal dividers and clock distribution pathways. The IN3 input pins are used as the external feedback and any of the outputs from the PLL which is being set up in zero delay mode should be routed to the IN3 differential inputs. It is recommended to use INO as the input clock when using IN3 as the external feedback clock in the zero delay mode. The terminations used for IN3 would depend on the driver type chosen- the preferred option is to use an LVDS or LVDS boost output ac coupled into a differential 100 ohm termination at the IN3 input side.

The diagram below shows the configuration recommended as an example. In this example the PLLA is set up in zero delay mode with OUTOT routed in to IN3 for the ZDB feedback.


Figure 15 Zero Delay Mode Set Up: PLLA in ZDB mode with OUTOT routed in to IN3 for the ZDB feedback

## 11 PLL Bandwidth Control

Each PLL Slave independently chooses the Bandwidth for jitter attenuation from 1 mHz to 4 kHz . This is the bandwidth that is normally used for steady state operation. However, an independent choice for a fast bandwidth is also available that can be used for speeding up the initial lock. After the PLL lock is achieved and the system is in the synchronized mode, the bandwidth is automatically transitioned to the steady state jitter attenuation bandwidth. This feature avoids the abnormally large wake up times that may be needed for very low PLL bandwidths. For stability considerations of the PLL, the fast lock bandwidth or regular bandwidth for the PLL should be no larger than $1 / 100^{\text {th }}$ of the input frequency at the input of the PLL (post the DIVN1k dividers).

## 12 PLL Crystal Clock Reference

An external crystal can be connected between the $\{\mathrm{X} 1, \mathrm{X} 2\}$ pins on the die to work with the internal crystal oscillator circuitry to produce the fref clock for the system. Alternatively, a TCXO based external clock source can be directly connected on the X1 pin. The requirements for the crystal and the external clock source are presented in the Electrical Specifications section. It is recommended to place the crystal on a floating metal island on the PCB that is provided the ground connection by the chip with the X1G and X2G pins. This metal island should not be connected to the PCB ground to prevent extraneous fref related currents to distribute on the board.

## 13 PLL Lock Loss Defect Monitoring

PLL Lock Loss is another fault monitor whose specifications are available in the Section 2 of this data sheet. Various programmable thresholds are available that can be used to detect lock loss in the PLL. Lock loss is indicated by the programmable drift between the frequency of the input clock for the PLL and the divided VCO clock. Similar to the faults monitored by the Clock Monitor Slave, this defect can be tracked with status, notify and on the FLEXIOs. This defect monitoring is described in detail in the section on "Clock Monitor Slave Description".

## 14 PLL DCO Mode operation

The Digitally Controlled Oscillator (DCO) mode of operation is used for changing the output frequency of a PLL using software control on the serial interface or pin control. A pre-defined change in frequency is programmed in the PIF of the respective PLL. After that an increase (FINC) or decrease (FDEC) command can be given on the PIF of the same PLL to make the change in output frequency effective. Alternatively, appropriate GPIOs are chosen for the trigger of the DCO function. A low to high transition (as an edge detect) is used for the trigger of the DCO increment or decrement. Any relative change in frequency from as fine as 5 ppt to as coarse as 100 ppm is available with the DCO mode. DCO mode is available in both free run and synchronized modes of operation.

## 15 Programmable Interface Top Level View

Table 17 PIF Overview (Top Level Summary of the Programmable Interface)
All Addresses are in Hexadecimal

| Page Number | Function | Comments | Fuse repeated twice |
| :---: | :---: | :---: | :---: |
| Oh | Generic | 2Fh[7:6]: Lock Pattern for the Fuse <br> FFh[7:0]: Current Page Number <br> 22h[7:0]: Current Fuse Pointer <br> 00h-01h: Customer- Chip Information <br> 02h-04h: First set of Defect / Notify / Interrupt <br> 06h-08h: Second set of Defect / Notify / Interrupt <br> 05h: DCO increment/decrement control <br> 0Fh: Program Command Directives and Active Trigger Directives <br> 10h: PLL enable control <br> 11h - 18h: Fuse GPIO (FlexIO) Multiplexed Control <br> 19h: VDD Padring Control and External CLKIN Switch Control <br> 1Ah - 21h: Die ID + Wafer Co-ordinates <br> 22h: Fuse Pointer Generic <br> 24h: Clock Input / Output Enable Control <br> 25h: Clock Output Enable Control <br> 26h: OEb, Clock Output Enable Control Settings <br> 27h - 28h: Masking of sticky bits status for Interrupt generation (INTR_b) <br> 29h: Chip GPIOs (FlexIO) Configuration <br> 2Ah: Fuse Based I2C Addr <br> 2Bh: Calibrations and Misc Settings <br> 2Ch - 2Fh: XO/XTAL/Reference Pathway Settings | NO |
| 1h | Clock Monitor | 2Fh[7:6] : Customer - Lock Pattern for the Fuse <br> FFh[7:0]: Current Page Number <br> 00h-01h: Chip Information <br> 02h - 04h: First set Defect / Notify / Interrupt for Clock Monitor Sub-system. <br> 06h-08h: Second set of Defect / Notify / Interrupt for Clock Monitor Subsystem <br> 0Fh: Program Command Directives and Active Trigger Directives 10h - 29h, 46h - 48h: Clock Loss Monitor Configuration 2Ah - 45h, 4Ch - 4Fh: Frequency Drift Coarse/Fine Configuration <br> 49h - 4Bh: PLLs Input Clock Priority Information | YES |
| 2h | Input | 2Fh[7:6] : Customer - Lock Pattern for the Fuse <br> FFh[7:0]: Current Page Number <br> 00h-01h: Chip Information <br> 02h - 04h: Defect / Notify / Interrupt for Input Sub-system <br> 0Fh: Program Command Directives and Active Trigger Directives <br> 10h - 19h: CLKINO Fuse Configuration (IDR, DIVNO, Clock MUX) <br> 20h - 2Fh: CLKIN1 Fuse Configuration (IDR, DIVN1, Clock MUX) <br> 30h - 3Fh: CLKIN2 Fuse Configuration (IDR, DIVN1, Clock MUX) <br> 40h - 4Fh: CLKIN3 Fuse Configuration (IDR, DIVN1, Clock MUX) | YES |


| Page Number | Function | Comments | Fuse repeated twice |
| :---: | :---: | :---: | :---: |
| 3h | 8 Flexi- <br> Outputs/ 4 <br> Fixed-Output  <br> Blocks  | 2Fh[7:6] : Customer - Lock Pattern for the Fuse <br> FFh[7:0]: Current Page Number <br> 00h-01h: Chip Information <br> 02h - 04h: Defect / Notify / Interrupt for Output Sub-system <br> 0Fh: Program Command Directives and Active Trigger Directives <br> 10h - 17h: Output Block 0 Fuse Configuration (ODR, DIVO, DIVO-Delay) <br> 18h - 1Fh: Output Block 1 Fuse Configuration (ODR, DIVO, DIVO-Delay) <br> 20h - 27h: Output Block 2 Fuse Configuration (ODR, DIVO, DIVO-Delay) <br> 28h - 2Fh: Output Block 3 Fuse Configuration (ODR, DIVO, DIVO-Delay) <br> 30h-37h: Output Block 4 Fuse Configuration (ODR, DIVO, DIVO-Delay) <br> 38h - 3Fh: Output Block 5 Fuse Configuration (ODR, DIVO, DIVO-Delay) <br> 40h - 47h: Output Block 6 Fuse Configuration (ODR, DIVO, DIVO-Delay) <br> 48h - 4Fh: Output Block 7 Fuse Configuration (ODR, DIVO, DIVO-Delay) <br> 50h - 57h: Output Block OT Fuse Configuration (ODR, DIVO, DIVO-Delay) <br> 58h-5Fh: Output Block 1T Fuse Configuration (ODR, DIVO, DIVO-Delay) <br> 60h - 67h: Output Block OB Fuse Configuration (ODR, DIVO, DIVO-Delay) <br> 68h - 6Fh: Output Block 1B Fuse Configuration (ODR, DIVO, DIVO-Delayt | YES |
| Ah | PLL A | 2Fh[7:6] : Customer - Lock Pattern for the Fuse <br> FFh[7:0]: Current Page Number <br> 00h-01h: Customer- Chip Information <br> 02h - 04h: First set of Defect / Notify / Interrupt for PLLA <br> 06h - 08h: Second set of Defect / Notify / Interrupt for PLLA <br> 05h: Customer- PLL Generic Directives <br> OFh: Program Command Directives and Active Trigger Directives <br> 10h - 2Fh: PLL Fuse Configuration (All PLL specific settings for this PLL) <br> 30h-37h: Customer- DCO Functionality | YES |

## 16 Serial Programming Interface Description

The device has two serial programming interface options, I2C and SPI, for reconfiguring the device settings. The protocol option can be selected through the I2C1_SPIO pin. A 1/HIGH on the pin sets the device in I2C mode and a 0/LOW in SPI mode.

### 16.1 I2C protocol

The device uses the SDAIO and SCLK pins for a 2-wire serial interface that operates up to $400 \mathrm{~Kb} / \mathrm{s}$ in Read and Write modes. It complies with the I2C bus standard. The I2C access protocol in device is byte access (random access) only for write and both random and sequential access for read.

The I2C serial interface can operate at either Standard rate ( 100 Kbps ) or Fast rate ( 400 Kbps ). For write operation, the device supports only single write operation. For read, the device supports both Single and Multiple read operation.

The default Slave address is $11011\{S D O\},\{C S B\}$ where SDO and CSB values are controlled by pins on the device in the I2C mode. Default addess is 0x6D. The device also supports variable Slave addresses which can be provided via the efuse. Therein too, the LSbs of A1 and A0 are controlled via the pins on the device. This thus allows four choices of Slave addresses for any system where in the first 5 bits of the slave address can be the same.


## Write Operation - Single Byte

| S | Slave Addr [6:0] | 0 | A | $\operatorname{Reg} \operatorname{Addr}[7: 0]$ | A | Data $[7: 0]$ | A | P |
| :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- |



## 1- Read, 0 - Write, A - Acknowledge (SDA LOW), $\mathbf{N}$ - Not Acknowledge (SDA HIGH), S - Start Condition, P - Stop Condition

- Single Byte Write
- The master initiates the transaction by issuing a start condition, writes 7 bit slave address and then the read/write bit is written as 0 (write)
- The slave acknowledges by driving zero on the bus
- The master then writes the 8 bit register map address
- The slave acknowledges by driving zero on the bus
- The master then writes the 8 bit data to be written to the register map address specified
- The slave acknowledges by driving zero on the bus
- The master ends the transaction by issuing a stop condition
- Single Byte Read
- The master initiates the transaction by issuing a start condition, writes 7 bit slave address and then the read/write bit is written as 0 (write)
- The slave acknowledges by driving zero on the bus
- The master then writes the 8 bit register map address
- The slave acknowledges by driving zero on the bus
- The master ends the transaction by issuing a stop condition
- The master re-initiates the transaction by issuing a start condition, writes 7 bit slave address and then the read/write bit is written as 1 (read)
- The slave then writes the 8 bit data to be written to the register map address specified
- The master does not acknowledge this transaction as the slave may assume a multi-byte read operation and there is a risk of slave holding the bus low
- The master ends the transaction by issuing a stop condition
- Multi Byte Read

The multi-byte read mode is used to read a continuous segment of the register map. The multi-byte read is faster than performing multiple single byte reads as the device address and register map address need not be specified for every byte read from the register map

- The master initiates the transaction by issuing a start condition, writes 7 bit slave address and then the read/write bit is written as 0 (write)
- The slave acknowledges by driving zero on the bus
- The master then writes the 8 bit register map address
- The slave acknowledges by driving zero on the bus
- The master ends the transaction by issuing a stop condition
- The master re-initiates the transaction by issuing a start condition, writes 7 bit slave address and then the read/write bit is written as 1 (read)
- The slave then writes the 8 bit data to be written to the register map address specified
- The master acknowledges by driving zero on the bus
- The slave automatically increments the register map address and writes the data in at that address to the bus and the master acknowledges
- When all bytes of data are read, master ends the operation by not acknowledging the last read
- The master then ends the transaction by issuing a stop condition


Figure 16 I2C Timing Waveforms

Table 18 I2C Bus Timing Specifications

| Description | Symbol | Standard Mode |  | Fast Mode |  | Units |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  |  | Min | Max | Min | Max |  |
| SCLK clock frequency | fscl | - | 100 | - | 400 | kHz |
| Hold time START condition | thd:sta | 4.0 | - | 0.6 | - | $\mu \mathrm{s}$ |
| Low period of the SCK clock | tow | 4.7 | - | 1.3 | - | $\mu \mathrm{s}$ |
| High period of the SCK clock | thigh | 4.0 | - | 0.6 | - | $\mu \mathrm{s}$ |
| Setup time for a repeated START condition | tsu:Sta | 4.7 | - | 0.6 | - | $\mu \mathrm{s}$ |
| Data hold time | thd:dAT | 10 | - | 10 | - | ns |
| Data setup time | tsu:dat | 100 | - | 100 | - | ns |
| Rise time | tR | - | 1000 | - | 300 | ns |
| Fall time | $t_{\text {F }}$ | - | 300 | - | 300 | ns |
| Setup time for STOP condition | tsu:sto | 4.0 | - | 0.6 | - | $\mu \mathrm{s}$ |
| Bus-free time between STOP and START conditions | tbuF | 4.7 | - | 1.3 | - | $\mu \mathrm{s}$ |
| Data valid time | tvd;DAT | - | 3.45 | - | 0.9 | $\mu \mathrm{s}$ |
| Data valid acknowledge time | tvo;Ack | - | 0.9 | - | 0.9 | $\mu \mathrm{s}$ |

### 16.2 SPI Protocol

The SPI is a four-pin interface with Chip Select (CSB), Serial Input (SDAIO), Serial Output (SDO), and Serial Clock (SCLK) pins. The SPI bus on the device can run at speed up to 20 MHz . The SPI is a synchronous serial interface, which uses clock and data pins for serial access. When I2C1_SPIO pin is Low, a Low on the CSB pin activates the SPI access.

1. The SPI can operate up to 20 MHz for regular write/read operations.
2. The SPI receives serial data from the external master and provides $\mathrm{Wr} / \mathrm{rdn}$ (set to $0 \times 01 \mathrm{~h}$ ), address and data to the register map during the write operation.
3. The SPI receives serial data from the external master and provides $\mathrm{Wr} / \mathrm{rdn}$ (set to $0 \times 00 \mathrm{~h}$ ), address to the register map and uses the read data obtained from the register map, serializes the same and transmit to the master.
4. The total packet size for each SPI transaction is 24 bits where the 8 bits are $\mathrm{Wr} / \mathrm{rdn}$ ( $0 \times 01$ for write and $0 \times 00$ for read), the next 8 bits are address and the last 8 bits are data
5. For write operation, the master assembles the $\mathrm{Wr} / \mathrm{rdn}$ byte, address and data for write operation on the falling edge of the SPI clock and the slave in the AU5327 captures the same on the rising edge of the SPI clock. There is no loopback provided here.
6. For read operation, the master assembles the $\mathrm{Wr} / \mathrm{rdn}$ byte, address for read operation on the falling edge of the SPI clock and the slave in the AU5327 captures the same on the rising edge of the SPI clock and there is no loopback. The falling edge after the $16^{\text {th }}$ rising SPI clock (i.e. the last address bit), is used by the slave to assemble the first read data which is captured by the master on the $17^{\text {th }}$ edge of the SPI clock. Subsequent 7 more clocks are used for the 7 remaining data bits.
7. The transmitter always sends data on the falling edge of the SPI clock to be captured in the receiver by the rising edge of the SPI clock. The transmitter can be the master for the whole operation of the write and for the control and address portions of the read. The slave is the transmitter during the data portion of the read cycle.
8. The register can be written to or read from one address at a time. The SPI implemented in AU5327 does not support burst address write or read operations.

### 16.3 SPI Timing Details



Figure 17 SPI Timing Diagram

Table 19 SPI Timing

| Description | Symbol | Min | Typ | Max | Units |
| :--- | :--- | :---: | :---: | :---: | :---: |
| SCLK clock frequency | fscLk | - | - | 20 | MHz |
| Clock pulse width HIGH | tch | 20 |  |  | ns |
| Clock pulse width LOW | tcL | 20 |  |  | ns |
| CSB HIGH time | tcs | 50 |  |  | ns |
| CSB setup time | tcss | 25 |  |  | ns |
| CSB hold time | tcsh | 25 |  |  | ns |
| Data in setup time | tsD | 10 |  |  | ns |
| Data in hold time | thD | 10 |  |  | ns |
| Output valid | tVo |  |  | 10 | ns |
| Output valid | tSo |  |  | 10 | ns |
| Output valid | tHo |  |  | 10 | ns |

### 16.3.1 SPI Single byte write

- The master initiates the transaction by issuing a start condition of pulling csn_i to active low
- The master assembles the serial data on the falling edge of the SPI clock so the SPI slave can capture the same on the rising edge of the SPI clock
- The first 7 bits are don't care with the $8^{\text {th }}$ bit being set to 1 to indicate a write operation
- The next 8 bits (second byte) are used for the register map address
- The next 8 bits (third byte) are used for the register map data
- The $24^{\text {th }}$ rising edge of the SPI clock is used to capture the last data bit. The SPI slave then assembles the address, data, enable and wr_rdn to the PIF slave block. The inverted version of the next falling edge of the SPI clock is used by the SPI slave to capture the address, data, enable and wr_rdn to write to the respective register
- The CSN is then de-activated (by going high) by the master
- For the next write operation, CSN is held high for at least a duration of two SPI clocks following which the entire operation can start again.


### 16.3.2 SPI Single byte read

- The master initiates the transaction by issuing a start condition of pulling csn_i to active low
- The master assembles the serial data on the falling edge of the SPI clock so the SPI slave can capture the same on the rising edge of the SPI clock
- The first 7 bits are don't care with the $8^{\text {th }}$ bit being set to 0 to indicate a read operation
- The next 8 bits (second byte) are used for the register map address
- The next 8 bits (third byte) are used for the register map read data that is supplied by the pif slave block
- The $16^{\text {th }}$ rising edge of the SPI clock is used to capture the last address bit. The SPI slave then assembles the address, enable and wr_rdn to the PIF slave block. The slave block then uses the address when enable is high to provide the read back data via a multiplexer. This operation has to be completed within half a SPI clock since the SPI slave has to assemble the first read back data bit on the falling edge of the SPI clock so the SPI slave can capture the same on the next rising edge. After 7 additional clocks, all the 8 serial read back data bits are sent out from the SPI slave.
- The CSN is then de-activated (by going high) by the master
- For the next read operation, CSN is held high for at least a duration of two SPI clocks following which the entire operation can start again.
- When CSB goes high after an SPI transaction is complete, it is recommended that the SPI clock is shut off and stays in the low stay till the next SPI transaction starts. During this time, the SDO will hold the last transmitted bit. The duration for which the CSB is expected to be held high is equivalent of two SPI clocks. If for some reason, the SPI clocks cannot be held static at zero and need to toggle when CSB is high, then the SDO will output a low after the first rising edge of the SPI clock after the CSB has transitioned to high. The duration for which CSB is expected to be held high is equivalent of two SPI clock.


Note:
W/RN - 1:SPI WRITE 0:SPI READ
Ax - Address
Dx - Data In

Figure 18 SPI Write - AU5327


Figure 19 SPI Read - AU5327
Note:
W/RN - 1:SPI WRITE 0:SPI READ
Ax - Address
Dx - Data Out

## 17 Package Information



|  | SYMBOL | MIN | NOM | Max |
| :---: | :---: | :---: | :---: | :---: |
| TOTAL THICKNESS | A | 0.8 | 0.85 | 0.9 |
| STAND OFF | A1 | 0 | 0.035 | 0.05 |
| MOLD THICKNESS | A2 | --- | 0.65 | --- |
| 1/F THICKNESS | A3 | 0.203 REF |  |  |
| LEAD WIDTH | b | 0.2 | 0.25 | 0.3 |
| BODY SIZE $X$ | D | 9 BSC |  |  |
| PODY SIZE | E | 9 BSC |  |  |
| EEAD PITCH | e | 0.5 BSC |  |  |
| EP SIZE | $J$ | 4.7 | 4.8 | 4.9 |
|  | K | 4.7 | 4.8 | 4.9 |
| LEAD LENGTH | L | 0.35 | 0.4 | 0.45 |
|  | L1 | 0.3 | 0.4 | 0.45 |
| PACKAGE EDGE TOLERANCE | aos | 0.1 |  |  |
| MOLD FLATNESS | bbb | 0.1 |  |  |
| COPLANARITY | ccc | 0.08 |  |  |
| LEAD OFFSET | ddd | 0.1 |  |  |
| EXPOSED PAD OFFSET | eee | 0.1 |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |



Figure 20 64-QFN Package Dimensions
Notes:

1. Coplanarity applies to LEADS, CORNER LEADS and DIE ATTACH PAD
2. Total Thickness does not include SAW BURR

## 18 Output Termination Information



Figure 21 LVPECL DC Termination to VDDO - 2V
$\mathrm{VDDO}=2.5 \mathrm{~V}, 3.3 \mathrm{~V}$


| VDDO | R1 | R2 |
| :--- | :--- | :--- |
| 2.5 V | $250 \Omega$ | $62.5 \Omega$ |
| 3.3 V | $127 \Omega$ | $82.5 \Omega$ |

Figure 22 LVPECL Alternate DC Termination: Thevenin Equivalent
$\mathrm{VDDO}=1.8 \mathrm{~V}, 2.5 \mathrm{~V}, 3.3 \mathrm{~V}$


Figure 23 DC Coupled LVDS Termination


Figure 24 DC Coupled CML
$\mathrm{VDDO}=18 \mathrm{~V}, 2.5 \mathrm{~V}, 3.3 \mathrm{~V}$

$\mathrm{VDDO}=1.8 \mathrm{~V}, 2.5 \mathrm{~V}, 3.3 \mathrm{~V}$


Figure 25 AC Coupled Receiver side resistive Termination options


Figure 26 Alternate AC Coupled LVPECL with DC coupled resistors on Chip side


Figure 27 DC Coupled LVCMOS


Figure 28 HCSL AC Coupled Termination. Source Terminated $50 \Omega$.
Note: Rs is sometimes used for limiting overshoot - Can be $0 \Omega$


Figure 29 HCSL DC Coupled Termination. Source Terminated $50 \Omega$.
Note: Rs is sometimes used for limiting overshoot - Can be $0 \Omega$

## 19 Input Termination Information



Figure 30 AC Coupled Differential LVDS Input / Other AC Coupled Driver without DC Terminations
Note: Uses Differential Buffer Pathway


Figure 31 AC Coupled Differential LVPECL or CML
Note: Resistor and Voltage termination is as per the standard. Uses Differential Buffer Pathway. Please refer to the termination requirements of the driver.


| Drive Supply | R1 (Ohms) | R2 (Ohms) |
| :--- | :--- | :--- |
| 1.8 V | 140 | 665 |
| 2.5 V | 325 | 475 |
| 3.3 V | 445 | 365 |

Figure 32 DC Coupled Single Ended Driver
Note: Uses Single Ended Buffer Pathway in DC Coupled Mode. Recommended for non-standard duty cycle applications. Please refer above table for the recommended resistor values for frequencies $<1 \mathrm{MHz}$.


Figure 33 AC Coupled Single Ended Driver with 50 Ohm Termination on receiver (chip) side.
Note: Uses Single Ended Buffer pathway in AC coupled mode.


Figure 34 AC Coupled Single Ended LVCMOS input without 50 Ohm Termination
Note: Uses Single Ended Buffer pathway in AC Coupled Mode. The LVCMOS driver in this case needs to ensure source termination to match to the transmission line.

## 20 Crystal Pathway Connectivity Options

The CMOS XO/TCXO output and the termination components should be placed as close as possible to the X1/X2 pins

## Crystal Connection



Figure 35 Crystal Connection
CMOS XO OUTPUT
(1.8V or 2.5 V or 3.3 V )


| CMOS XO Driver Supply | R1 (Ohms) | R2 (Ohms) |
| :---: | :---: | :---: |
| 1.8 V | 0 | DNP |
| 2.5 V | 274 | 732 |
| 3.3 V | 453 | 549 |

Figure 36 CMOS XO Connection
Note: R1 and R2 can be reduced by $2 x$ to improve the slew rate. Additionally, the 5 pF can be placed across R1 to improve the slew rate further if PCB routing length is significant from the attenuator network to the X 1 pin.

## Differential XO/Clock



Figure 37 Differential XO Connection

## 21 Monitoring through the register map read back: Status and Notify

AU5327 provides various Status and notify bits that can be accessed from the register map. Below are the details of the procedure to be followed to access the same.

The alarm registers are a set of three types of registers distributed between the various pages as described in the Table 20 and illustrated with some examples in the section 21.2 and 21.3.

- The Status registers are the current dynamic status of a defect. The live status defects are active high with a ' 1 ' indicating the defect is present.
- The Notify registers are the sticky bits for a defect. Sometimes, we may get a very short pulse for the status and it may not be possible for the external user to capture the same. Hence a notify register is provided. The notify register is set to 1 whenever there is a rising edge of the corresponding status register. This is a sticky bit and stays at 1 till the user writes a 1 to that specific bit to clear it.
- Each notify has a masking bit to enable or disable its operation. The notify sticky bit operates only if the corresponding masking bit is set to 1 . If the masking register bit is set to 0 , notify will not be asserted even when status toggles. The default value for the mask register is $0 x f f$ so all the notify signals are enabled. Once the user writes a 1 to clear the notify, the notify bit can again go high on the next rising edge of the status.
These registers operate on the internal 4 MHz RC clock. When there is a defect (i.e. status) of any bit in register it gets asserted and de-asserted in a live mode. User can read the corresponding register location to see the current status at any time.

On the other hand, the default value of the notify and masking register is 0xff - user has to write 0xff to both these registers to clear them at the beginning and use all notifies.It is recommended to clear all notifies after programming a profile.
The INTRB pin is used as a NOR operation of the selected notifies. The choice of the Notify listing that is used for the INTRB pin is selectable in the GUI when creating the profile. The sticky notifies that are selected and used for INTRB need to be cleared for restoring the INTRB to 1 for further sticky defect monitoring using this pin.

### 21.1 Tabular Listing

The Table 20 details the name of the alarm, the page it is located in, the address in the page and the bit number in the address. In order to access a page of the register map, the particular page number has to be written to address 0xff. For instance, to either write to or read from page 1, first the user needs to write to 0xff a value of $0 \times 01$. Following this, any number of write or read operations can be done with page 1.

Table 20 Tabular Listing of Alarm Registers

| S.No | Name of Signal | Description | Page <br> Number | Register <br> Address | Bit <br> Number |
| :--- | :--- | :--- | :---: | :---: | :---: |
| 1 | xo_clkloss_dynamic_status | Xo clock dynamic status | 00 | $0 \times 02$ | 2 |
| 2 | xo_clkloss_dynamic_ntfy | Xo clock dynamic Notify (write <br> $0 \times 04$ with 0x02 to enable) | 00 | $0 \times 03$ | 2 |
| 3 | plla_lol_dyn_status | pll lol dynamic status | 00 | $0 \times 06$ | 0 |
| 4 | pllb_lol_dyn_status | pll lol dynamic status | 00 | $0 \times 06$ | 1 |
| 5 | pllc_lol_dyn_status | pll lol dynamic status | 00 | $0 \times 06$ | 2 |
| 6 | plld_lol_dyn_status | pll lol dynamic status | 00 | $0 \times 06$ | 3 |
| 7 | plla_ho_frz_status | pll ho freeze dynamic status | 00 | $0 \times 06$ | 4 |
| 8 | pllb_ho_frz_status | pll ho freeze dynamic status | 00 | $0 \times 06$ | 5 |
| 9 | pllc_ho_frz_status | pll ho freeze dynamic status | 00 | $0 \times 06$ | 6 |
| 10 | plld_ho_frz_status | pll ho freeze dynamic status | 00 | $0 \times 06$ | 7 |
| 11 | plla_lol_dyn_ntfy | pll lol dynamic Notify (write 0x08 <br> with 0x01 to enable selectively, <br> $0 x f f ~ t o ~ e n a b l e ~ a l l ~ n o t i f i e s ~ i n ~$ | 00 | $0 \times 07$ | 0 |


| S.No | Name of Signal | Description | $\begin{aligned} & \text { Page } \\ & \text { Number } \end{aligned}$ | Register Address | Bit Number |
| :---: | :---: | :---: | :---: | :---: | :---: |
| 12 | pllb_lol_dyn_ntfy | pll lol dynamic Notify (write 0x08 with $0 \times 02$ to enable selectively, 0xff to enable all notifies in 0x07) | 00 | $0 \times 07$ | 1 |
| 13 | pllc_lol_dyn_ntfy | pll lol dynamic Notify (write 0x08 with $0 \times 04$ to enable selectively, 0xff to enable all notifies in $0 \times 07$ ) | 00 | $0 \times 07$ | 2 |
| 14 | plld_lol_dyn_ntfy | pll lol dynamic Notify (write 0x08 with $0 \times 08$ to enable selectively, 0xff to enable all notifies in $0 \times 07$ ) | 00 | $0 \times 07$ | 3 |
| 15 | plla_ho_frz_ntfy | pll ho freeze dynamic Notify (write $0 \times 08$ with $0 \times 10$ to enable selectively, Oxff to enable all notifies in 0x07) | 00 | $0 \times 07$ | 4 |
| 16 | pllb_ho_frz_ntfy | pll ho freeze dynamic Notify (write $0 \times 08$ with $0 \times 20$ to enable selectively, 0xff to enable all notifies in $0 \times 07$ ) | 00 | $0 \times 07$ | 5 |
| 17 | pllc_ho_frz_ntfy | pll ho freeze dynamic Notify (write 0x08 with $0 \times 40$ to enable selectively, 0xff to enable all notifies in $0 \times 07$ ) | 00 | $0 \times 07$ | 6 |
| 18 | plld_ho_frz_ntfy | pll ho freeze dynamic Notify (write $0 \times 08$ with $0 \times 80$ to enable selectively, Oxff to enable all notifies in 0x07) | 00 | $0 \times 07$ | 7 |
| 19 | in0_clock_loss_dyn_status | Clock Loss Dynamic Status | 01 | $0 \times 02$ | 0 |
| 20 | In1_clock_loss_dyn_status | Clock Loss Dynamic Status | 01 | $0 \times 02$ | 1 |
| 21 | In2_clock_loss_dyn_status | Clock Loss Dynamic Status | 01 | $0 \times 02$ | 2 |
| 22 | In3_clock_loss_dyn_status | Clock Loss Dynamic Status | 01 | 0x02 | 3 |
| 23 | in0_clock_loss_fd_dyn_status | Clock Loss + FD Dynamic Status | 01 | $0 \times 02$ | 4 |
| 24 | In1_clock_loss_fd_dyn_status | Clock Loss + FD Dynamic Status | 01 | $0 \times 02$ | 5 |
| 25 | In2_clock_loss_fd_dyn_status | Clock Loss + FD Dynamic Status | 01 | $0 \times 02$ | 6 |
| 26 | In3_clock_loss_fd_dyn_status | Clock Loss + FD Dynamic Status | 01 | $0 \times 02$ | 7 |
| 27 | in0_clock_loss_ntfy | Clock Loss Dynamic Notify (write $0 \times 04$ with $0 \times 01$ to enable selectively, 0xff to enable all notifies in 0x03) | 01 | $0 \times 03$ | 0 |
| 28 | In1_clock_loss_ntfy | Clock Loss Dynamic Notify (write $0 \times 04$ with $0 \times 02$ to enable selectively, Oxff to enable all notifies in 0x03) | 01 | $0 \times 03$ | 1 |
| 29 | In2_clock_loss_ntfy | Clock Loss Dynamic Notify (write 0x04 with 0x04 to enable selectively, Oxff to enable all notifies in 0x03) | 01 | $0 \times 03$ | 2 |
| 30 | In3_clock_loss_ntfy | Clock Loss Dynamic Notify (write 0x04 with 0x08 to enable selectively, 0xff to enable all notifies in $0 \times 03$ ) | 01 | $0 \times 03$ | 3 |


| S.No | Name of Signal | Description | $\begin{gathered} \text { Page } \\ \text { Number } \end{gathered}$ | Register Address | $\begin{gathered} \text { Bit } \\ \text { Number } \end{gathered}$ |
| :---: | :---: | :---: | :---: | :---: | :---: |
| 31 | in0_clock_loss_ntfy | Clock Loss with Fd_dynamic Notify (write 0x04 with 0x10 to enable selectively, 0xff to enable all notifies in $0 \times 03$ ) | 01 | $0 \times 03$ | 4 |
| 32 | In1_clock_loss_ntfy | Clock Loss with Fd_dynamic Notify (write 0x04 with 0x20 to enable selectively, 0xff to enable all notifies in 0x03) | 01 | $0 \times 03$ | 5 |
| 33 | In2_clock_loss_ntfy | Clock Loss with Fd_dynamic Notify (write 0x04 with 0x40 to enable selectively, 0xff to enable all notifies in $0 \times 03$ ) | 01 | $0 \times 03$ | 6 |
| 34 | In3_clock_loss_ntfy | Clock Loss with Fd_dynamic Notify (write 0x04 with 0x80 to enable selectively, 0xff to enable all notifies in 0x03) | 01 | $0 \times 03$ | 7 |
| 35 | in0_fd_fine_dyn_status | Frequency drift dynamic status | 01 | $0 \times 06$ | 0 |
| 36 | In1_fd_fine_dyn_status | Frequency drift dynamic status | 01 | 0x06 | 1 |
| 37 | In2_fd_fine_dyn_status | Frequency drift dynamic status | 01 | 0x06 | 2 |
| 38 | In3_fd_fine_dyn_status | Frequency drift dynamic status | 01 | $0 \times 06$ | 3 |
| 39 | in0_fd_coarse_dyn_status | Frequency drift dynamic status | 01 | $0 \times 06$ | 4 |
| 40 | In1_fd_coarse_dyn_status | Frequency drift dynamic status | 01 | $0 \times 06$ | 5 |
| 41 | In2_fd_coarse_dyn_status | Frequency drift dynamic status | 01 | $0 \times 06$ | 6 |
| 42 | In3_fd_coarse_dyn_status | Frequency drift dynamic status | 01 | 0x06 | 7 |
| 43 | in0_fd_fine_ntfy | Frequency drift dynamic notify (write $0 \times 08$ with $0 \times 01$ to enable selectively, 0xff to enable all notifies in $0 \times 07$ ) | 01 | $0 \times 07$ | 0 |
| 44 | In1_fd_fine_ntfy | Frequency drift dynamic notify (write $0 \times 08$ with $0 \times 02$ to enable selectively, 0xff to enable all notifies in 0x07) | 01 | $0 \times 07$ | 1 |
| 45 | In2_fd_fine_ntfy | Frequency drift dynamic notify (write 0x08 with 0x04 to enable selectively, 0xff to enable all notifies in 0x07) | 01 | $0 \times 07$ | 2 |
| 46 | In3_fd_fine_ntfy | Frequency drift dynamic notify (write 0x08 with 0x08 to enable selectively, 0xff to enable all notifies in 0x07) | 01 | $0 \times 07$ | 3 |
| 47 | in0_fd_coarse_ntfy | Frequency drift dynamic notify (write $0 \times 08$ with $0 \times 10$ to enable selectively, Oxff to enable all notifies in 0x07) | 01 | $0 \times 07$ | 4 |
| 48 | In1_fd_coarse_ntfy | Frequency drift dynamic notify (write 0x08 with 0x20 to enable selectively, 0xff to enable all notifies in 0x07) | 01 | $0 \times 07$ | 5 |
| 49 | In2_fd_coarse_ntfy | Frequency drift dynamic notify (write $0 \times 08$ with $0 \times 40$ to enable selectively, 0xff to enable all notifies in $0 \times 07$ ) | 01 | $0 \times 07$ | 6 |
| 50 | In3_fd_coarse_ntfy | Frequency drift dynamic notify (write 0x08 with 0x80 to enable selectively, Oxff to enable all notifies in 0x07) | 01 | $0 \times 07$ | 7 |


| S.No | Name of Signal | Description | Page Number | Register <br> Address | $\begin{gathered} \text { Bit } \\ \text { Number } \end{gathered}$ |
| :---: | :---: | :---: | :---: | :---: | :---: |
| 51 | fast_lock_dynamic_status_plla | PLL Fast lock dynamic status | 0a | $0 \times 06$ | 1 |
| 52 | fast_lock_dynamic_status_pllb | PLL Fast lock dynamic status | 0b | $0 \times 06$ | 1 |
| 53 | fast_lock_dynamic_status_pllc | PLL Fast lock dynamic status | 0c | $0 \times 06$ | 1 |
| 54 | fast_lock_dynamic_status_plld | PLL Fast lock dynamic status | 0d | $0 \times 06$ | 1 |
| 55 | fast_lock_dynamic_ntfy_plla | PLL Fast lock dynamic Notify (write 0x08 with 0x02 to enable selectively, 0xff to enable all notifies in 0x07) | 0a | $0 \times 07$ | 1 |
| 56 | fast_lock_dynamic_ntfy_pllb | PLL Fast lock dynamic Notify (write $0 \times 08$ with $0 \times 02$ to enable selectively, 0xff to enable all notifies in 0x07) | Ob | $0 \times 07$ | 1 |
| 57 | fast_lock_dynamic_ntfy_pllc | PLL Fast lock dynamic Notify (write $0 \times 08$ with $0 \times 02$ to enable selectively, 0xff to enable all notifies in 0x07) | 0c | $0 \times 07$ | 1 |
| 58 | fast_lock_dynamic_ntfy_plld | PLL Fast lock dynamic Notify (write $0 \times 08$ with $0 \times 02$ to enable selectively, 0xff to enable all notifies in 0x07) | 0d | $0 \times 07$ | 1 |
| 59 | ho_valid_dynamic_status_plla | PLL Ho Valid dynamic status | 0a | $0 \times 06$ | 2 |
| 60 | ho_valid_dynamic_status_pllb | PLL Ho Valid dynamic status | 0b | $0 \times 06$ | 2 |
| 61 | ho_valid_dynamic_status_pllc | PLL Ho Valid dynamic status | 0c | $0 \times 06$ | 2 |
| 62 | ho_valid_dynamic_status_plld | PLL Ho Valid dynamic status | 0d | $0 \times 06$ | 2 |
| 63 | ho_valid_dynamic_ntfy_plla | PLL Ho Valid dynamic Notify (write 0x08 with 0x04 to enable) | 0a | $0 \times 07$ | 2 |
| 64 | ho_valid_dynamic_ntfy_pllb | PLL Ho Valid dynamic Notify (write 0x08 with 0x04 to enable selectively, 0xff to enable all notifies in 0x07) | Ob | $0 \times 07$ | 2 |
| 65 | ho_valid_dynamic_ntfy_pllc | PLL Ho Valid dynamic Notify (write 0x08 with 0x04 to enable selectively, 0xff to enable all notifies in 0x07) | 0c | $0 \times 07$ | 2 |
| 66 | ho_valid_dynamic_ntfy_plld | PLL Ho Valid dynamic Notify (write 0x08 with 0x04 to enable selectively, 0xff to enable all notifies in $0 \times 07$ ) | 0d | $0 \times 07$ | 2 |

### 21.2 Examples for Live Status Read Back

Some examples are presented based on the above Table 20 for reading the live status of the defects.
In the pseudo code presented below:
wr_cmd(address, data): refers to a "Write Command" where the corresponding data is written in to the specified register address
$\mathrm{x}=$ rd_cmd(address): refers to a "Read Command" where the corresponding data is read from the specified register address and stored in the variable ' $x$ '
$y \gg x$ : denotes a bit wise right shift on the number $y$ by $x$ bit locations
$\mathrm{y} \ll \mathrm{x}$ :denotes a bit wise left shift on the number y by x bit locations
\& is the logical AND operation (bit wise)

Dynamic registers to read the various alarm registers in the RealTime page.

1. Input Clocks CL and FD Related Real Time live status read back:
wr_cmd(0xff, 0x01) Program the CLKMON_SYS page number

## Clock Loss dynamic status

```
clock_loss_dyn_status = rd_cmd(0x02) & 0xff
```

(clock_loss_dyn_status >> 0) \& 0x01
(clock_loss_dyn_status >> 1) \& $0 \times 01$
(clock_loss_dyn_status >> 2) \& $0 \times 01$
(clock_loss_dyn_status >> 3) \& $0 \times 01$

## Frequency Drift dynamic status

```
fd_fine_dyn_status = rd_cmd(0x06) & 0x0f
fine =(fd_fine_dyn_status >> 0) & 0x01
fine =(fd_fine_dyn_status >> 1) & 0x01
fine =(fd_fine_dyn_status >> 2) & 0x01
fine =(fd_fine_dyn_status >> 3) & 0x01
fd_coarse_dyn_status = rd_cmd(0x06) >> 4
coarse =(fd_coarse_dyn_status >> 0) & 0x01
coarse = (fd_coarse_dyn_status >> 1) & 0x01
coarse =(fd_coarse_dyn_status >> 2) & 0x01
coarse =(fd_coarse_dyn_status >> 3) & 0x01
```

INO Status for CL, Read bit position [0]
IN1 Status for CL, Read bit position [1]
IN2 Status for CL, Read bit position [2]
IN3 Status for CL, Read bit position [3]

## 2. PLL Related Real Time live status read back:

wr_cmd(0xff, 0x00) Program the GENERIC_SYS page number, Page 0
pll_Iol_ho_freeze_dyn_status = rd_cmd $(0 x 06) \& 0 x f f$

## PLL Lock Loss dynamic status

(pll_lol_ho_freeze_dyn_status >> 0) \& 0x01
(pll_lol_ho_freeze_dyn_status >> 1) \& $0 \times 01$
(pll_lol_ho_freeze_dyn_status >> 2) \& 0x01
(pll_lol_ho_freeze_dyn_status >> 3) \& $0 \times 01$

PLLA Status for LL, Read bit position [0]
PLLB Status for LL, Read bit position [1]
PLLC Status for LL, Read bit position [2]
PLLD Status for LL, Read bit position [3]

## Holdover Status

```
(pll_lol_ho_freeze_dyn_status >> (0 + 4)) & 0x01 PLLA Status for HO, Read bit position [4]
(pll_lol_ho_freeze_dyn_status >> (1 + 4)) & 0x01 PLLA Status for HO, Read bit position [5]
(pll_lol_ho_freeze_dyn_status >> (2 + 4)) & 0x01 PLLA Status for HO, Read bit position [6]
(pll_lol_ho_freeze_dyn_status >> (3+4)) & 0x01 PLLA Status for HO, Read bit position [7]
```


## 3. XO clock loss Related Real Time live status read back:

## CLOS_X1X2, XO Clock Loss

wr_cmd(0xff, 0x00) Program the GENERIC_SYS page number, Page 0
clos_x1x2 $=$ rd_cmd(0x02) \& $0 \times 04 / / \mathrm{XO}$ CL Status, Read bit position [2]

### 21.3 Examples of Sticky Bit Clearing

As described earlier, the sticky notify bits are cleared by writing a ' 1 ' to the corresponding notify bit itself. The notify bit by itself is enabled by writing a ' 1 ' to the corresponding mask bit.
In the pseudo code presented below,
rmw_cmd(addr,bit_loc,no_of_bits,data): denotes the read/modify/write operation where no_of_bits number of bits at bit_loc location (denoted as 7:0) is replaced with the data at address location addr.

```
def clr_intb_XO_CL():
    This function is used to clear the sticky notify for XO Clock Loss
    Write the page number
    wr_cmd(0xff, 0)
    Information to clr ** Page 0: reg03[2]=1 **
    addr = 0x3
    bit_loc = 2
    no_of_bits = 1
    data = 1
    rmw_cmd(addr,bit_loc,no_of_bits,data)
def clr_intb_LOL_HO_Freeze():
    This function is used to clear the sticky notify for loss of lock and holdover notify for all PLLs
    Write the page number
    wr_cmd(0xff, 0)
    Information to clr ** Page 0: reg07[7:0] = 0xff **
    addr = 0x7
    bit_loc = 7
    no_of_bits = 8
    data = 0xff
    rmw_cmd(addr,bit_loc,no_of_bits,data)
def clr_intb_CL():
    This function is used to clear the sticky notify for clear clock loss notify
    Write the page number
    wr_cmd(0xff, 1)
    Information to clr ** Page1: reg03[3:0]=0x0f **
    addr = 0x3
    bit_loc = 3
    no_of_bits = 4
```

```
    data = 0x0f
    rmw_cmd(addr,bit_loc,no_of_bits,data)
def clr_intb_drift():
    This function is used to clear the sticky notify for clear drift notify
    Write the page number
    wr_cmd(0xff, 1)
    Information to clr ** Page1: reg07[7:0]=0xff **
    addr = 0x7
    bit_loc = 7
    no_of_bits = 8
    data = 0xff
    rmw_cmd(addr,bit_loc,no_of_bits,data)
def clr_intrb():
    '"
    This is the main clear function
    which calls the 4 clear functions
    ""
    clr_intb_XO_CL()
    clr_intb_LOL_HO_Freeze()
    clr_intb_CL()
    clr_intb_drift()
```


## 22 Device Initialization for a non-programmed device

This section describes a device initialization flow chart for an unlocked device. An unlocked device is a device on which the NVM is not programmed and where an autonomous wake up does not happen. It is assumed that the user is using a device that is not programmed for the description in this section.
The AU5327 device register initialization flowchart for Master Control Page and the slave pages is as below. As explained in the chip functional description in the data sheet, the following is the sequence of the wake up of the various sub systems in the chip.

- First, the master control (Page 0) is initialized and programmed. The Slaves are powered up at this stage.
- Next, the Input System (Page 2) is initialized and programmed for cases where at least one input is enabled.
- Next, the Clock Monitor System (Page 1) is initialized and programmed for cases where at least one input is enabled.
- Next, the Output System (Page 3) is initialized and programmed.
- Finally, the PLLs that are expected to be used for the particular profile (PLLs A, B, C, D correspond to Pages A, B, C, D) are initialized and programmed.

The register 0xFF is written with the Page Number the user would like to access. The chip changes the current page once the user has written the register 0xFF. The page numbers corresponding to each slave and master are described in the respective table in the data sheet. At any point in time the register 0xFF can be read to find out the current page.

Please note that the flow chart in the following pages is to be used together with the sequence of register writes that are obtained from the GUI with the "Save NVM" button after loading a profile. This file obtained from the GUI describes the set of registers to be written in the exact order and with appropriate delays. This file obtained from the GUI is the master sequence to be followed for programming a part. The figures on the following pages describe the flow of the same register write sequence file using flow charts.


Figure 38 STEP 1: Initialize the Main Page- Page 0.
We can move to Page 0 by writing $0 \times 00$ to the address $0 x F F$. Read the current page at any time by reading the contents of the register 0xFF


Figure 39 STEP 2: Initialize the Input System Page- Page 2.
We can move to Page 2 by writing $0 x 02$ to the address $0 x F F$. Read the current page at any time by reading the contents of the register 0xFF. This page is not initialized for a purely free run profile where no inputs are engaged.


Figure 40 STEP 3: Initialise the Clock Monitor System Page- Page 1.
We can move to Page 1 by writing $0 \times 01$ to the address $0 x F F$. Read the current page at any time by reading the contents of the register 0xFF. This page is not initialized for a purely free run profile where no inputs are engaged.


Figure 41 STEP 4: Initialise the Output System Page- Page 3.
We can move to Page 3 by writing $0 x 03$ to the address $0 x F F$. Read the current page at any time by reading the contents of the register 0xFF.


Figure 42 STEP 5: Initialise the PLL A System Page- Page A.
We can move to Page $A$ by writing $0 \times 0 \mathrm{~A}$ to the address $0 x F F$. Read the current page at any time by reading the contents of the register 0xFF. Initialize PLL A only if it is used in the profile being used.


Figure 43 STEP 6: Initialise the PLL B System Page- Page B.
We can move to Page B by writing $0 \times 0 \mathrm{~B}$ to the address $0 \times \mathrm{FFF}$. Read the current page at any time by reading the contents of the register 0xFF. Initialize PLL B only if it is used in the profile being used.


Figure 44 STEP 7: Initialise the PLL C System Page- Page C.
We can move to Page C by writing $0 \times 0 \mathrm{C}$ to the address $0 \times F F$. Read the current page at any time by reading the contents of the register 0xFF. Initialize PLL C only if it is used in the profile being used.


Figure 45 STEP 8: Initialise the PLL D System Page- Page D.

We can move to Page D by writing 0x0D to the address 0xFF. Read the current page at any time by reading the contents of the register 0xFF. Initialize PLL D only if it is used in the profile being used.


Figure 46 Active State Confirmation
Confirm from each page that was initialized that their respective state machines are in Active State. Check Lock Status of PLLs that are enabled.

## 23 Monitoring the Status for Master and Slave Pages

Once the AU5327 is powered up, the NVM register contents are read, it is preferable to provide sufficient time delay for the read operation.

As explained in the data sheet above, we need to read that the Master Control Page 0 is in the Program Command Wait State (PRG_CMD State) before starting to write to the part. Please use the register write sequence from the GUI generated from the "Save NVM" button to write in to the device.

Once the entire sequence to program the profile has been written, each Page (Pages $0,1,2,3$, PLL Pages A, B, C, D based on which PLLs are enabled) can be checked to ensure they are in the ACTIVE state.

The sequence that needs to be followed to check the status of the Master and Slave Pages is as below

## MAIN PAGE (0)

1) Write $0 x F F$ register to $0 \times 00$ Go to Page 0
2) Write 0XD1 register on Page 0 to $0 \times 40$.
3) Read the register 0xD0.

If read data $=9$ (Main Sys is in PRG_CMD state) read data $=0$ (Main Sys is in IDLE state) read data $=36$ (Main Sys is in Active state)

## CLK MON PAGE (1)

1) Write $0 \times F F$ register to $0 \times 01$. Go to Page 1
2) Write 0XD1 register on Page 1 to $0 \times 40$.
3) Read 0xD0[3:0]

If read data $=5$ (CLK MON is in PRG_CMD state) read data $=0$ (CLK MON is in IDLE state) read data $=12$ (CLK MON is in Active state)

## INPUT SYS PAGE (2)

1) Write $0 x F F$ register to $0 \times 02$.

Go to Page 2
2) Write 0XD1 register on Page 2 to $0 \times 40$.
4) Read $0 x D 0[4: 0]$
3) If read data $=7$ (INPUT_SYS is in PRG_CMD state) read data $=0$ (INPUT_SYS is in IDLE state) read data $=23$ (INPUT_SYS is in Active state)

## OUT_SYS PAGE (3)

1) Write 0xFF register to $0 x 03$. Go to Page 3
2) Write 0XD1 register on Page 3 to $0 \times 60$.
3) Read $0 x \mathrm{DO} 0[4: 0]$

If read data $=12$ (OUT_SYS is in PRG_CMD state) read data $=0$ (OUT_SYS is in IDLE state) read data $=20$ (OUT_SYS is in Active state)

## PLL PAGE (A, B, C, D)

1) Write $0 x F F$ register to $0 \times 0 \mathrm{~A}(\mathrm{PLLA})$.
(0x0A - PLLA, 0x0B - PLLB, 0x0C-PLLC, OXOD-PLLD)
2) Write $0 \times D 1$ register on Page $A$ to $0 \times 20$.
3) Read 0xD0.

If read data $=8$ (PLL PAGE is in PRG_CMD state)
read data $=0$ (PLL PAGE is in IDLE state)
read data $=48$ (PLL PAGE is in Active state)

## Monitoring the Loss of Lock Status for PLL

The sequence that needs to be followed to monitor the dynamic LOL status of the PLL's is as below:

1) Write $0 \times F F$ register to $0 \times 0 A(P L L A)$ ( $0 \times 0 \mathrm{~A}-\mathrm{PLLA}, 0 \times 0 B-\mathrm{PLLB}, 0 \times 0 C-\mathrm{PLLC}, 0 \times 0 D-P L L D)$
2) Write $0 \times 04$ register on PLL Page to $0 \times 01$.

Remove the mask for lock loss notify status
3) Read 0x02[0]

LOL dynamic status_PLLA_B_C_D
If read data $=1$ (Loss of Lock is asserted: PLL is not locked)
read data $=0(\mathrm{PLL}$ is locked $)$

## Monitoring the Hold Over Status for PLLs

The sequence that needs to be followed to monitor the Hold Over status of the PLL's is as below:

1) Write $0 x F F$ register to $0 \times 0 \mathrm{~A}(\mathrm{PLLA})$. ( $0 \times 0 \mathrm{~A}-\mathrm{PLLA}, 0 \times 0 \mathrm{~B}$ - PLLB, $0 \times 0 \mathrm{C}-\mathrm{PLLC}, 0 \times 0 D-\mathrm{PLLD}$ )
2) Read $0 \times 17[7]$

If read data $=1$
To check if the PLL Outer Loop is Enabled/Disabled read data $=0$ It indicates a Free Run Profile --- The PLL's will always be in Hold Over It is not a Free Run Profile ------ PLL may or may not be in Hold Over
3) Write $0 x B 3$ register to $0 x 0 \mathrm{D}$
4) Read 0xb9[3]

Read the 3rd bit in B9 register for Hold Over Status Information
If read data $=1$
PLL is in Holdover State
PLL is not in Holdover State

## Note:

0XD1 - Status Readback Address Register for Master and Slave Pages.
0XD0 - Status Readback Data Register for Master and Slave Pages. Need to be read atleast 3 times for reliable readback.

## 24 Programming the Primary E-Fuse

This section describes primary E-Fuse program configuration for all the pages [GENERIC_SYS, INPUT_SYS, CLKMON_SYS, OUTPUT_SYS, PLLA_SYS, PLLB_SYS, PLLC_SYS, PLLD_SYS].
$\mathrm{VDD}=2.5 \mathrm{~V}$ and $\mathrm{VDDIN}=2.5 \mathrm{~V}$ should be used for programming the E-Fuse

### 24.1 Configuration Bits to Force Power-up of Digital Slave Subsystems

The respective subsystems needs to be powered up before programming the E-Fuse
This section describes GENERIC_SYS page configuration required in register 0xe0 and 0xe1 to enable respective slave subsystems [INPUT_SYS, CLKMON_SYS, OUTPUT_SYS, PLLA_SYS, PLLB_SYS, PLLC_SYS, PLLD_SYS ] as described in Table 21.

Table 21 Configuration Bits to Force Power-up of Digital Slave Subsystems

| S.NO | PAGE NUMBER | REGISTER ADDRESS | BIT <br> NUMBER | VALUE \& IT'S DESCRIPTION |
| :---: | :---: | :---: | :---: | :---: |
| 1. | Page 0 | 0xE0 | 0 | 1 'h1 (write to bit number 0 in register address $0 x E 0$ with value 1'h1 to enable force overwrite INPUT_SYS) |
|  |  |  | 1 | 1'h1 (write to bit number 1 in register address 0xE0 with value 1'h1 to enable force overwrite CLKMON_SYS) |
|  |  |  | 2 | 1 'h1 (write to bit number 2 in register address $0 \times E 0$ with value 1'h1 to enable force overwrite OUTPUT_SYS) |
|  |  |  | 3 | 1 'h1 (write to bit number 3 in register address 0xE0 with value 1'h1 to enable force overwrite PLLA_SYS |
|  |  |  | 4 | 1 'h1 (write to bit number 4 in register address $0 \times E 0$ with value 1'h1 to enable force overwrite PLLB_SYS) |
|  |  |  | 5 | 1'h1 (write to bit number 5 in register address 0xE0 with value 1'h1 to enable force overwrite PLLC_SYS) |
|  |  |  | 6 | 1'h1 (write to bit number 6 in register address 0xE0 with value 1'h1 to enable force overwrite PLLD_SYS) |
| 2. | Page 0 | 0xE1 | 0 | 1 'h1 (write to bit number 0 in register address $0 x E 1$ with value 1'h1 INPUT_SYS to be Enabled) |
|  |  |  | 1 | 1'h1 (write to bit number 1 in register address $0 \times E 1$ with value 1'h1 CLKMON_SYS to be Enabled) |
|  |  |  | 2 | 1 'h1 (write to bit number 2 in register address $0 x E 1$ with value 1'h1 OUTPUT_SYS to be Enabled) |
|  |  |  | 3 | 1'h1 (write to bit number 3 in register address 0xE1 with value 1'h1 PLLA_SYS to be Enabled) |
|  |  |  | 4 | 1'h1 (write to bit number 4 in register address 0xE1 with value 1'h1 PLLB_SYS to be Enabled) |
|  |  |  | 5 | 1'h1 (write to bit number 5 in register address 0xE1 with value 1'h1 PLLC_SYS to be Enabled) |
|  |  |  | 6 | 1'h1 (write to bit number 5 in register address $0 \times E 1$ with value1'h1PLLD_SYS to be Enabled) |

### 24.2 E-Fuse Lock Configuration Bits

This section describes the location of the two bits to lock the respective slave subsystem E-Fuse by writing into the register 0x2F in all pages as described in Table 22.

Table 22 E-Fuse Lock Configuration Bits for all pages

| S.No | Page Number | Register Address | Bit <br> Number | Value and Its Description |
| :---: | :---: | :---: | :---: | :---: |
| 1. | Page 0 | 0x2f | 7:6 | 2'h1: E-Fuse of GENERIC_SYS is locked by writing into bit number [7:6] in register address $0 \times 2 \mathrm{f}$ with value 2'h1 |
| 2. | Page 1 | 0x2f | 7:6 | 2'h1 : E-Fuse of CLKMON_SYS is locked by writing into bit number $[7: 6]$ in register address $0 \times 2 \mathrm{f}$ with value 2'h1 |
| 3. | Page 2 | 0x2f | 7:6 | 2'h1 : E-Fuse of INPUT_SYS is locked by writing into bit number [7:6] in register address $0 \times 2 \mathrm{f}$ with value 2 'h1 |
| 4. | Page 3 | 0x2f | 7:6 | 2'h1: E-Fuse of OUTPUT_SYS is locked by writing into bit number $[7: 6]$ in register address $0 \times 2 \mathrm{f}$ with value 2'h1 |
| 5. | Page A | 0x2f | 7:6 | 2'h1: E-Fuse of PLLA_SYS is locked by writing into bit number $[7: 6]$ in register address $0 \times 2 f$ with value 2 'h1 |
| 6. | Page B | 0x2f | 7:6 | 2'h1: E-Fuse of PLLB_SYS is locked by writing into bit number [7:6] in register address 0x2f with value 2'h1 |
| 7. | Page C | 0x2f | 7:6 | 2'h1 : E-Fuse of PLLC_SYS is locked by writing into bit number [7:6] in register address 0x2f with value 2'h1 |
| 8. | Page D | 0x2f | 7:6 | 2'h1 : E-Fuse of PLLD_SYS is locked by writing into bit number $[7: 6$ ] in register address $0 \times 2 f$ with value 2 'h1 |

### 24.3 E-Fuse Write Configuration Bits

This section will describe how to program the E-Fuse by writing into the register $0 \times 0 \mathrm{~F}$ in all the pages as mentioned in Table 23.

Table 23 E-Fuse Write Configuration Bits for All Pages

| S.No | Page Number | Register Address | $\begin{gathered} \text { Bit } \\ \text { Number } \end{gathered}$ | Value and Its Description |
| :---: | :---: | :---: | :---: | :---: |
| 1. | Page 0 | 0x0f | 7:3 | 5'h0C: writing to bit number [7:3] in register address 0x0f with value 5'h0C to do E-Fuse write in GENERIC_SYS |
| 2. | Page 1 | 0x0f | 7:3 | 5'h0C: writing to bit number [7:3] in register address 0x0f with value 5'h0C to do E-Fuse write in CLKMON SYS |
| 3. | Page 2 | 0x0f | 7:3 | 5'h0C: writing to bit number [7:3] in register address 0x0f with value 5'h0C to do E-Fuse write in INPUT SYS |
| 4. | Page 3 | 0x0f | 7:3 | 5'h0C: writing to bit number [7:3] in register address 0x0f with value 5'h0C to do E-Fuse write in OUTPUT SYS |
| 5. | Page A | 0x0f | 7:3 | 5'h0C: writing to bit number [7:3] in register address 0x0f with value 5'h0C to do E-Fuse write in PLLA SYS |
| 6. | Page B | 0x0f | 7:3 | 5'h0C: writing to bit number [7:3] in register address 0x0f with value 5'h0C to do E-Fuse write in PLLB SYS |
| 7. | Page C | 0x0f | 7:3 | 5'h0C: writing to bit number [7:3] in register address 0x0f with value $5^{\prime} \mathrm{hOC}$ to do E-Fuse write in PLLC SYS |
| 8. | Page D | 0x0f | 7:3 | 5'h0C: writing to bit number [7:3] in register address 0x0f with value $5^{\prime}$ h0C to do E-Fuse write in PLLD SYS |

### 24.4 Configuration Bit to Remove Manual Wake Up for Primary E-Fuse

Manual wake-up of slave subsystems is enabled while programming the E-Fuse. As a final step the manual wake-up mode needs to be disabled in the GENERIC_SYS page so that the chip will wake-up autonomously This section will describe how to remove manual wakeup by writing into a register 0x2f in GENERIC_SYS as mentioned in Table 24

Table 24 Configuration Bit to Remove Manual Wakeup for Primary E-Fuse

| S.No | Page <br> Number | Register <br> Address | Bit <br> Number | Value \& It's Description |
| :--- | :---: | :---: | :---: | :--- |
| 1. | Page 0 | $0 \times 2 \mathrm{f}$ | 4 | 1'h1: (writing to bit number 4 in register address 0x2f with <br> value 'h1 to remove manual wake up) |

### 24.5 Pseudo Code: Programming the Primary E-Fuse

VDD and VDDIN supply should be set to 2.5 V while programming the E-Fuse

### 24.5.1 GENERIC_SYS

STEP 1: Write the GENERIC_SYS page number configuration

```
i2c.i2cw(device_address,0xff,0x00)
```

STEP 2: Write the GENERIC_SYS NVM Registers configuration

STEP 3: Refer E-Fuse Write Configuration Bits section described earlier
Write to bit number [7:3] in register address 0x0f with value 5 'hC to program E-Fuse registers of GENERIC_SYS

```
i2c.i2cw(device address,0x0f,0x00)
i2c.i2cw(device_address,0x0f,0xc0)
i2c.i2cw(device_address,0x0f,0x00)
```

STEP 4: Refer Configuration Bits to Force Power-up of Digital Slave Subsystems described earlier
 PLLD_SYS] by programming $0 \times \mathrm{EO}$ \& $0 \times \mathrm{E} 1$ both with value 7'h7F

```
i2c.i2cw(device_address,0xe0,0x7f)
```

i2c.i2cw (device_address, 0xe1, 0x7f)

### 24.5.2 INPUT_SYS

STEP 5: Write the INPUT_SYS page number configuration

```
i2c.i2cw(device_address,0xff,0x02)
```

STEP 6: Refer E-Fuse Lock Configuration Bits
Write the INPUT_SYS NVM Registers configuration and write 0x2F register for bit number [7:6] with 2'h1 to lock E-Fuse of INPUT_SYS

STEP 7: Refer E-Fuse Write Configuration Bits
Write to bit number [7:3] in register address 0xOf with value 5 'hC to program E-Fuse Registers of INPUT_SYS

```
i2c.i2cw(device_address,0x0f,0x00)
i2c.i2cw(device-address, 0x0f,0xc0)
i2c.i2cw(device_address,0x0f,0x00)
```


### 24.5.3 CLKMON_SYS

STEP 8: Write the CLKMON_SYS page number configuration

```
i2c.i2cw(device_address,0xff,0x01)
```

STEP 9: Refer E-Fuse Lock Configuration Bits
Write the CLKMON_SYS NVM Registers configuration and write 0x2F register for bit number [7:6] with 2'h1 to lock E-Fuse of CLKMON_SYS
STEP 10: Refer E-Fuse Write Configuration Bits
Write to bit number [7:3] in register address 0x0F with value 5'hC to program E-Fuse Registers of CLKMON_SYS
i2c.i2cw (device_address, 0x0f, 0x00)
i2c.i2cw (device_address, 0x0f, 0xc0)
i2c.i2cw (device_address, 0x0f, 0x00)

### 24.5.4 OUTPUT_SYS

STEP 11: Write the OUTPUT_SYS page number configuration
i2c.i2cw(device_address, 0xff, 0x03)
STEP 12: Refer E-Fuse Lock Configuration Bits
Write the OUTPUT_SYS NVM Registers configuration and write 0x2F register for bit number [7:6] with 2'h1 to lock E-Fuse of OUTPUT_SYS

## STEP 13: Refer E-Fuse Write Configuration Bits

Write to bit number [7:3] in register address 0x0F with value 5'hC to program E-Fuse Registers of OUTPUT_SYS
i2c.i2cw (device_address, 0x0f, 0x00)
i2c.i2cw (device_address, 0x0f, 0xc0)
i2c.i2cw (device_address, 0x0f, 0x00)

### 24.5.5 PLLA_SYS

STEP 14: Write the PLLA_SYS page number configuration
i2c.i2cw(device_address,0xff,0x0a)
STEP 15: Refer E-Fuse Lock Configuration Bits
Write the PLLA_SYS NVM Registers configuration and write 0x2F register for bit number [7:6] with 2'h1 to lock E-Fuse of PLLA_SYS

## STEP 16: Refer E-Fuse Write Configuration Bits

Write to bit number [7:3] in register address 0x0F with value 5'hC to program E-Fuse Registers of PLLA_SYS

```
i2c.i2cw(device_address, 0x0f,0x00)
```

i2c.i2cw (device_address, 0x0f,0xc0)
i2c.i2cw (device_address, 0x0f, 0x00)

### 24.5.6 PLLB_SYS

STEP 17: Write the PLLB_SYS page number configuration

```
i2c.i2cw(device_address,0xff,0x0b)
```

STEP 18: Refer E-Fuse Lock Configuration Bits
Write the PLLB_SYS NVM Registers configuration and write 0x2F register for bit number [7:6] with 2'h1 to lock E-Fuse of PLLB_SYS

## STEP 19: Refer E-Fuse Write Configuration Bits

Write to bit number [7:3] in register address 0x0F with value 5'hC to program E-Fuse Registers of PLLB_SYS
i2c.i2cw (device_address, 0x0f, 0x00)
i2c.i2cw (device_address, 0x0f, 0xc0)
i2c.i2cw (device_address, 0x0f, 0x00)

### 24.5.7 PLLC_SYS

STEP 20: Write the PLLC_SYS page number configuration
i2c.i2cw(device_address, 0xff, 0x0c)

STEP 21: Refer E-Fuse Lock Configuration Bits
Write the PLLC_SYS NVM Registers configuration and write 0x2F register for bit number [7:6] with 2'h1 to lock E-Fuse of PLLC_SYS

## STEP 22: Refer E-Fuse Write Configuration Bits

Write to bit number [7:3] in register address 0x0F with value 5'hC to program E-Fuse Registers of PLLC_SYS i2c.i2cw (device_address, 0x0f, 0x00)
i2c.i2cw (device_address, 0x0f, 0xc0)
i2c.i2cw (device_address, 0x0f, 0x00)

### 24.5.8 PLLD_SYS

STEP 23: Write the PLLD_SYS page number configuration
i2c.i2cw(device_address, 0xff, 0x0d)
STEP 24: Refer E-Fuse Lock Configuration Bits
Write the PLLD_SYS NVM Registers configuration and write 0x2F register for bit number [7:6] with 2 'h1 to lock E-Fuse of PLLD_SYS

STEP 25: Refer E-Fuse Write Configuration Bits
Write to bit number [7:3] in register address 0x0F with value 5'hC to program E-Fuse Registers of PLLD_SYS
i2c.i2cw (device_address, 0x0f, 0x00)
i2c.i2cw (device_address, 0x0f, 0xc0)
i2c.i2cw (device_address, 0x0f, 0x00)

### 24.5.9 Removing Manual Wake-up and Locking GENERIC_SYS E-Fuse

STEP 26: Write the GENERIC_SYS page number configuration

```
i2c.i2cw(device_address, 0xff,0x00)
```

STEP 27: Refer Table 24 Configuration Bit to Remove Manual Wakeup for Primary E-Fuse
Write to bit number 4 in register address 0x2F with value 1'h1 in GENERIC_SYS to Configuration Bit to Remove Manual Wake Up

```
i2c.i2crmw(device_address,0x2f,0x4,0x1,0x01)
```

I2Crmw function register 0x2f bit number 4 writing with the value 1 ' h 1
Note: i2crmw(dev_address, register_address,bit_postion,total_bits,value)

## STEP 28: Refer E-Fuse Lock Configuration Bits

Write to bit number [7:6] in register address 0x2F with value 2'h1 in in GENERIC_SYS to lock the E-Fuse
i2c.i2crmw (device_address, $0 \times 2 \mathrm{f}, 0 \times 7,0 \times 2,0 \times 01$ ) //I2Crmw function register $0 \times 2 \mathrm{f}$ bit number $7: 6$ writing with the value 2'h1

STEP 29: Refer E-Fuse Write Configuration Bits
Write to bit number [7:3] in register address 0x0F with value 5'hC to program E-Fuse Registers of GENERIC_SYS
i2c.i2cw (device_address, 0x0f, 0x00)
i2c.i2cw (device_address, 0x0f, 0xc0)
i2c.i2cw(device_address, 0x0f,0x00)

STEP 30: Reset the chip or recycle Power for the chip to wake-up autonomously based on NVM configuration programmed in the primary E-Fuse

## 25 Programming the Secondary E-Fuse

This section describes Secondary E-Fuse program configuration for all the pages [GENERIC_SYS, INPUT_SYS, CLKMON_SYS, OUTPUT_SYS, PLLA_SYS, PLLB_SYS, PLLC_SYS,PLLD_SYS].

### 25.1 Configuration Bit To Escape To PROGRAM_CMD State in GENERIC_SYS

With the primary E-Fuse programmed and locked the chip will autonomously wake-up and reach ACTIVE_STATE For the E-Fuse writes the chip needs to be in PROGRAM_CMD state
This section will describe how to do escape from ACTIVE_STATE to PROGRAM_CMD state in GENERIC_SYS by writing into the register 0x0F in GENERIC_SYS as mentioned in Table 25

Table 25 Configuration Bit To Escape to PROGRAM_CMD State in GENERIC_SYS

| S.No | Page <br> Number | Register <br> Address | Bit <br> Number | Value \& It's Description |
| :--- | :---: | :---: | :---: | :--- |
| 1. | Page 0 | 0x0f | 1 | 1'h1: writing to bit number 1 in register address 0xOf with <br> value 1'h1 to do Escape to PROGRAM_CMD state in <br> GENERIC_SYS |

### 25.2 Configuration Bit to Change the E-Fuse pointer

This section will describe how to point slaves to secondary E-Fuse by writing into a register $0 \times 22$ in GENERIC_SYS as mentioned in Table 26

Table 26 Configuration Bit to change the E-Fuse pointer

| S.No | Page <br> Number | Register <br> Address | Bit <br> Number | Value \& It's Description |
| :--- | :---: | :---: | :---: | :--- |
| 1. | Page 0 | $0 \times 22$ | $7: 0$ | 8'hFF: (writing to bit number [7:0] in register address 0x22 <br> with value 8'hFF to point slaves [INPUT_SYS, <br> CLKMON_SYS, OUTPUT_SYS, PLLA_SYS, <br> PLLB_SYS, PLLC_SYS, PLLD_SYS] to secondary E-Fuse) |

### 25.3 Configuration Bit to Enable Manual Wake Up for Secondary E-Fuse

This section will describe how to enable manual wakeup for secondary E-Fuse by writing into a register $0 \times 10$ in GENERIC_SYS as mentioned in Table 27

Table 27 Configuration Bit to Enable Manual Wakeup for Secondary E-Fuse

| S.No | Page <br> Number | Register <br> Address | Bit <br> Number | Value \& It's Description |
| :--- | :---: | :---: | :---: | :--- |
| 1. | Page 0 | $0 \times 10$ | 7 | 1'h1: (writing to bit number 7 in register address $0 \times 10$ with <br> value 1'h1 to select manual wake up) |
| 2. | Page 0 | $0 \times 10$ | 6 | 1'h0: (writing to bit number 6 in register address $0 \times 10$ with <br> value 1'h0 to select manual wake up) |

### 25.4 Psuedo Code: Programming the Secondary E-Fuse

### 25.4.1 GENERIC_SYS

STEP 1: Write the GENERIC_SYS page number configuration

```
i2c.i2cw(device address,0xff,0x00)
```

STEP 2: Refer Configuration Bit To Escape To PROGRAM_CMD State in GENERIC_SYS
Write to bit number 1 in register address $0 \times 0 F$ with value 1 'h1 to escape to PROGRAM_CMD state in GENERIC_SYS

```
i2c.i2cw(device_address,0x0f,0x00)
i2c.i2cw(device_address,0x0f,0x02)
i2c.i2cw(device_address,0x0f,0x00)
```

STEP 3: Refer Configuration Bit to Change the E-Fuse pointer

Write to bit number [7:0] in register address $0 \times 22$ with value 8 'hFF to point slaves
[INPUT_SYS,CLKMON_SYS,OUTPUT_SYS,PLLA_SYS,PLLB_SYS,PLLC_SYS,PLLD_SYS] to secondary EFuse in GENERIC_SYS

```
i2c.i2cw(device_address,0x22,0xff)
```


## STEP 4: Refer Configuration Bit to Enable Manual Wake Up for Secondary E-Fuse

Write to bit number 7 in register address $0 \times 10$ with value 1 'h1 to select for manual wakeup in GENERIC_SYS i2c.i2crmw (device_address, $0 \times 10,0 \times 7,0 \times 1,0 \times 01$ ) I2Crmw function register $0 \times 10$ bit number 7 writing with the value 1 ' h 1

STEP 5: Refer E-Fuse Write Configuration Bits
Write to bit number [7:3] in register address $0 \times 0 \mathrm{~F}$ with value 5 'hC to program E-Fuse Registers of GENERIC_SYS

```
i2c.i2cw(device address, 0x0f, 0x00)
i2c.i2cw(device_address,0x0f,0xc0)
i2c.i2cw(device_address,0x0f,0x00)
```

STEP 6: Reset the chip or recycle power for the E-Fuse pointers to get updated

## STEP 7: Refer Configuration Bits to Force Power-up of Digital Slave Subsystems

Force Enable all slaves [INPUT_SYS, CLKMON_SYS, OUTPUT_SYS, PLLA_SYS, PLLB_SYS, PLLC_SYS, PLLD_SYS] by programming 0xE0 \& 0xE1 both with value 7'h7F

```
i2c.i2cw(device address,0xe0,0x7f)
```

i2c.i2cw (device_address, 0xe1,0x7f)

### 25.4.2 INPUT_SYS

STEP 8: Write the INPUT_SYS page number configuration
i2c.i2cw(device_address,0xff,0x02)

STEP 9: Refer E-Fuse Lock Configuration Bits
Write the INPUT_SYS NVM Registers configuration and write 0x2F register for bit number [7:6] with 2 'h1 to lock E-Fuse of INPUT_SYS

## STEP 10: Refer E-Fuse Write Configuration Bits

Write to bit number [7:3] in register address 0x0F with value 5'hC to program E-Fuse Registers of INPUT_SYS
i2c.i2cw (device_address, 0x0f, 0x00)
i2c.i2cw (device_address, 0x0f, 0xc0)
i2c.i2cw (device_address, 0x0f, 0x00)

### 25.4.3 CLKMON_SYS

STEP 11: Write the CLKMON_SYS page number configuration

```
i2c.i2cw(device_address,0xff,0x01)
```


## STEP 12: Refer E-Fuse Lock Configuration Bits

Write the CLKMON_SYS NVM Registers configuration and write $0 \times 2 F$ register for bit number [7:6] with 2 'h1 to lock E-Fuse of CLKMON_SYS

## STEP 13: Refer E-Fuse Write Configuration Bits

Write to bit number [7:3] in register address $0 \times 0 \mathrm{~F}$ with value 5 ' hC to program E-Fuse Registers of CLKMON_SYS

```
i2c.i2cw(device_address,0x0f,0x00)
i2c.i2cw(device_address, 0x0f,0xc0)
i2c.i2cw(device_address, 0x0f,0x00)
```


### 25.4.4 OUTPUT_SYS

STEP 14: Write the OUTPUT_SYS page number configuration

```
i2c.i2cw(device_address,0xff,0x03)
```

STEP 15: Refer E-Fuse Lock Configuration Bits
Write the OUTPUT_SYS NVM Registers configuration and write 0x2F register for bit number [7:6] with 2'h1 to lock E-Fuse of OUTPUT_SYS

STEP 16: Refer E-Fuse Write Configuration Bits
Write to bit number [7:3] in register address 0x0F with value 5 'hC to program E-Fuse Registers of OUTPUT_SYS i2c.i2cw (device_address, $0 \times 0$ f, $0 \times 00$ )
i2c.i2cw (device_address, $0 \times 0$ f, $0 \times \mathrm{xc} 0$ )
i2c.i2cw (device_address, 0x0f, 0x00)

### 25.4.5 PLLA_SYS

STEP 17: Write the PLLA_SYS page number configuration

```
i2c.i2cw(device_address,0xff,0x0a)
```

STEP 18: Refer E-Fuse Lock Configuration Bits
Write the PLLA_SYS NVM Registers configuration and write 0x2F register for bit number [7:6] with 2'h1 to lock E-Fuse of PLLA_SYS

## STEP 19: Refer E-Fuse Write Configuration Bits

Write to bit number [7:3] in register address $0 \times 0 \mathrm{~F}$ with value 5 'hC to program E-Fuse Registers of PLLA_SYS

```
i2c.i2cw(device_address,0x0f,0x00)
i2c.i2cw(device_address,0x0f,0xc0)
i2c.i2cw(device_address,0x0f,0x00)
```


### 25.4.6 PLLB_SYS

STEP 20: Write the PLLB_SYS page number configuration

```
i2c.i2cw(device_address,0xff,0x0b)
```

STEP 21: Refer E-Fuse Lock Configuration Bits
Write the PLLB_SYS NVM Registers configuration and write 0x2F register for bit number [7:6] with 2 'h1 to lock E-Fuse of PLLB_SYS

## STEP 22: Refer E-Fuse Write Configuration Bits

Write to bit number [7:3] in register address 0x0F with value 5'hC to program E-Fuse Registers of PLLB_SYS
i2c.i2cw (device address, 0x0f, 0x00)
i2c.i2cw (device_address, 0x0f, 0xc0)
i2c.i2cw (device_address, 0x0f, 0x00)

### 25.4.7 PLLC_SYS

STEP 23: Write the PLLC_SYS page number configuration

```
i2c.i2cw(device_address,0xff,0x0c)
```

STEP 24: Refer E-Fuse Lock Configuration Bits
Write the PLLC_SYS NVM Registers configuration and write 0x2F register for bit number [7:6] with 2'h1 to lock E-Fuse of PLLC_SYS

## STEP 25: Refer E-Fuse Write Configuration Bits

Write to bit number [7:3] in register address 0x0F with value 5'hC to program E-Fuse Registers of PLLC_SYS
i2c.i2cw (device_address, 0x0f, 0x00)
i2c.i2cw (device_address, 0x0f, 0xc0)
i2c.i2cw (device_address, 0x0f, 0x00)

### 25.4.8 PLLD_SYS

STEP 26: Write the PLLD_SYS page number configuration

```
i2c.i2cw(device_address,0xff,0x0d)
```

STEP 27: Refer E-Fuse Lock Configuration Bits
Write the PLLD_SYS NVM Registers configuration and write 0x2F register for bit number [7:6] with 2 'h1 to lock E-Fuse of PLLD_SYS

STEP 28: Refer E-Fuse Write Configuration Bits
Write to bit number [7:3] in register address 0x0F with value 5'hC to program E-Fuse Registers of PLLD_SYS
i2c.i2cw (device_address, 0x0f, 0x00)
i2c.i2cw (device_address, 0x0f, 0xc0)
i2c.i2cw (device_address, 0x0f, 0x00)
25.4.9 Programming GENERIC_SYS E-Fuse to Remove Manual Wake Up in Secondary E-Fuse STEP 29: Write the GENERIC_SYS page number configuration

```
i2c.i2cw(device_address,0xff,0x00)
```

STEP 30: Refer Configuration Bit To Escape To PROGRAM_CMD State in GENERIC_SYS
Write to bit number 1 in register address 0x0F with value 1'h1 to escape to PROGRAM_CMD state in GENERIC_SYS
i2c.i2cw (device_address, 0x0f, 0x00)
i2c.i2cw (device_address, 0x0f, 0x02)
i2c.i2cw (device_address, 0x0f, 0x00)
STEP 31: Write to bit number 6 in register address $0 \times 10$ with value 1 'h1 in GENERIC_SYS to Configuration Bit to Remove Manual Wake Up in secondary E-Fuse
i2c.i2crmw (device_address, $0 \times 10,6,0 \times 1,0 \times 01$ ) I2Crmw function register $0 \times 10$ bit number 6 writing with the value 1'h1
note: i2crmw(dev_address,register_address,bit_postion,total_bits,value)

## STEP 32: Refer E-Fuse Write Configuration Bits

Write to bit number [7:3] in register address $0 \times 0 \mathrm{~F}$ with value 5 'hC to program E-Fuse Registers of GENERIC_SYS

```
i2c.i2cw(device_address,0x0f,0x00)
i2c.i2cw(device_address,0x0f,0xc0)
i2c.i2cw(device_address,0x0f,0x00)
```

STEP 33: Reset Chip or recycle power for the chip to wake-up autonomously from the second E-Fuse

## 26 Register Map Details

Table 28 Generic Master System Related Registers
Registers from 10h to 4Fh are equivalent NVMCopy Registers for this Page.

| Reg Name | Register Number | $\begin{gathered} \text { Bit } \\ \text { Range } \end{gathered}$ | Access Type | Default Value | Description |
| :---: | :---: | :---: | :---: | :---: | :---: |
| VERSION_ID | 01h | 7:0 | R | $\begin{gathered} \mathrm{C} 2 \mathrm{~h} \\ (\mathrm{AU} 532 \mathrm{x}) \end{gathered}$ | Overall Aura Platform Revision |
| STATUS_1_GENERIC | 02h | 7 | R | 00h | SPARE |
|  |  | 6 | R | 00h | SPARE |
|  |  | 5 | R | 00h | SPARE |
|  |  | 4 | R | 00h | SPARE |
|  |  | 3 | R | 00h | SPARE |
|  |  | 2 | R | 00h | Dynamic status for xoclk_loss |
|  |  | 1 | R | 00h | SPARE |
|  |  | 0 | R | 00h | Dynamic status for rccal_done |
| NOTIFY_1_GENERIC | 03h | 7 | R/W | 1h | SPARE |
|  |  | 6 | R/W | 1h | SPARE |
|  |  | 5 | R/W | 1h | SPARE |
|  |  | 4 | R/W | 1h | SPARE |
|  |  | 3 | R/W | 1h | SPARE |
|  |  | 2 | R/W | 1n | Sticky/Notify status for_xoclk_loss, |
|  |  | 1 | R/W | 1n | SPARE |
|  |  | 0 | R/W | 1n | SPARE |
| MASKb_1_GENERIC | 04h | 7 | R/W | 1h | SPARE |
|  |  | 6 | R/W | 1n | SPARE |
|  |  | 5 | R/W | 1h | SPARE |
|  |  | 4 | R/W | 1h | SPARE |
|  |  | 3 | R/W | 1h | SPARE |
|  |  | 2 | R/W | 1h | Mask bit for NOTIFY_1_GENERIC (03h) If programmed as '0': Mask sticky/Notify bit generation for 03h[2]. |
|  |  | 1 | R/W | 1h | SPARE |
|  |  | 0 | R/W | 1h | SPARE |
| Directives_GENERIC | 05h | 3:2 | R/W | 00h | 0 $\vdots$ DCO_PLLA <br> 1 $\vdots$ DCO_PLLB <br> 2 $\vdots$ DCO_PLLC <br> 3 $\vdots$ DCO_PLLD |
|  |  | 1 | R/W | 00h | dco increment commonly used for all plls if $0 x e 7[7]$ is set as ' 1 ' |
|  |  | 0 | R/W | 00h | dco decrement commonly used for all plls if $0 x e 7[7]$ is set as ' 1 ' |
| STATUS_2_GENERIC | 06h | 7 | R | 00h | Dynamic status for plld_ho_freeze |
|  |  | 6 | R | 00h | Dynamic status for pllc_ho_freeze |
|  |  | 5 | R | 00h | Dynamic status for pllb_ho_freeze |
|  |  | 4 | R | 00h | Dynamic status for plla_ho_freeze |
|  |  | 3 | R | 00h | Dynamic status for plld_loss_of_lock |
|  |  | 2 | R | 00h | Dynamic status for pllc_loss_of_lock |
|  |  | 1 | R | 00h | Dynamic status for pllb_loss_of_lock |
|  |  | 0 | R | 00h | Dynamic status for plla_loss_of_lock |
| NOTIFY_2_GENERIC | 07h | 7 | R/W | 1h | Sticky/Notify status for plld_ho_freeze |
|  |  | 6 | R/W | 1h | Sticky/Notify status for pllc_ho_freeze |


| Reg Name | Register Number | $\begin{gathered} \text { Bit } \\ \text { Range } \end{gathered}$ | Access Type | Default Value | Description |
| :---: | :---: | :---: | :---: | :---: | :---: |
|  |  | 5 | R/W | 1h | Sticky/Notify status for pllb_ho_freeze |
|  |  | 4 | R/W | 1h | Sticky/Notify status for plla_ho_freeze |
|  |  | 3 | R/W | 1h | Sticky/Notify status for plld_loss_of_lock |
|  |  | 2 | R/W | 1h | Sticky/Notify status for pllc_loss_of_lock |
|  |  | 1 | R/W | 1h | Sticky/Notify status for pllb_loss_of_lock |
|  |  | 0 | R/W | 1h | Sticky/Notify status for plla_loss_of_lock |
| MASKb_2_GENERIC | 08h | 7 | R/W | 1h | Mask bit for NOTIFY_2_GENERIC (07h) If programmed as '0': Mask sticky/Notify bit generation for $07 \mathrm{~h}[7]$. |
|  |  | 6 | R/W | 1h | Mask bit for NOTIFY_2_GENERIC (07h) If programmed as ' 0 ': Mask sticky/Notify bit generation for 07h[6]. |
|  |  | 5 | R/W | 1h | Mask bit for NOTIFY_2_GENERIC (07h) If programmed as '0': Mask sticky/Notify bit generation for 07h[5]. |
|  |  | 4 | R/W | 1h | Mask bit for NOTIFY_2_GENERIC (07h) If programmed as ' 0 ': Mask sticky/Notify bit generation for 07h[4]. |
|  |  | 3 | R/W | 1h | Mask bit for NOTIFY_2_GENERIC (07h) If programmed as ' 0 ': Mask sticky/Notify bit generation for 07h[3]. |
|  |  | 2 | R/W | 1h | Mask bit for NOTIFY_2_GENERIC (07h) If programmed as ' 0 ': Mask sticky/Notify bit generation for 07h[2]. |
|  |  | 1 | R/W | 1h | Mask bit for NOTIFY_2_GENERIC (07h) If programmed as '0': Mask sticky/Notify bit generation for 07h[1]. |
|  |  | 0 | R/W | 1h | Mask bit for NOTIFY_2_GENERIC (07h) If programmed as '0': Mask sticky/Notify bit generation for 07h[0]. |
| PRG_Directives_GENERIC | OFh | 7:3 | R/W | Oh | PRG CMD Directives: <br> 5b1_1000: PROGRAM_EFUSE <br> 5b0_1100: READ_EFUSE <br> 5b0_0110: Copy NVM Copy to Settings <br> 5b1_1011: Proceed to Active |
|  |  | 2 | R/W | Oh | SPARE |
|  |  | 1 | R/W | Oh | Escape to the PRG_CMD state from ACTIVE state |
| NVMPLLEN_GENERIC | 10h | 7 | R/W | Oh | Selects between maual_wake_upb and manual_wake_up2b |
|  |  | 6 | R/W | Oh | select '0' to enable manual wake up sequence |
|  |  | 5:4 | R/W | Oh | $\begin{aligned} & \text { VDD_DEF : VDD }\{1.8(00), 2.5(01), \\ & 3.3(10)\} \end{aligned}$ |
|  |  | 3:0 | R/W | Oh | Bits $\{3,2,1,0\}$ correspond to enable of PLL \{D,C,B,A\}: Active Low. |
| NVMFLEXIO8_GENERIC | 18h | 7 | R/W | Oh | Select the direction of FLEXIO14. ' 0 ' Input, '1' - Output |


| Reg Name | Register Number | $\begin{gathered} \text { Bit } \\ \text { Range } \end{gathered}$ | Access Type | Default Value | Description |
| :---: | :---: | :---: | :---: | :---: | :---: |
| NVMFLEXIO8_GENERIC | 18h | 6:4 | R/W | Oh | Mux select to bring out internal signals on FLEXIO14, if programmed as output <br> 3'b000 : 0 <br> 3'b001: Ored_all_notify (clock loss for clkin0/1/2/3/ + Freq drift coarse for clkin $0 / 1 / 2 / 3+$ Freq drift coarse for clkin 0/1/2/3 + Loss of lock for PLL A/B/C/D+HO freeze for PLL A/B/C/D) <br> 3'b010 : Ored_all_pll_ntfy (Loss of lock for PLL A/B/C/D+HO freeze for PLL A/B/C/D) <br> 3'b011 : Ored_all_clkmon_ntfy ((clock loss for clkin0/1/2/3/ + Freq drift coarse for clkin $0 / 1 / 2 / 3$ + Freq drift coarse for clkin 0/1/2/3) <br> 3'b100 : <br> Dynamic_Clock_Monitoring_Status <br> 3'b101: Internal debug <br> 3'b110 : Internal debug <br> 3'b111: Internal debug |
|  |  | 3 | R/W | Oh | Select the direction of FLEXIO15. ' 0 ' Input, '1' - Output |
|  |  | 2:0 | R/W | Oh | Mux select to bring out internal signals on FLEXIO15, if programmed as output <br> 3'b000: 0 <br> 3'b001: Ored_all_notify (clock loss for clkin0/1/2/3/ + Freq drift coarse for clkin $0 / 1 / 2 / 3+$ Freq drift coarse for clkin 0/1/2/3 + Loss of lock for PLL A/B/C/D+HO freeze for PLL A/B/C/D) <br> 3'b010 : Ored_all_pll_ntty (Loss of lock for PLL A/B/C/D+HO freeze for PLL A/B/C/D) <br> 3'b011 : Ored_all_clkmon_ntfy ((clock loss for clkin0/1/2/3/ + Freq drift coarse for clkin $0 / 1 / 2 / 3$ + Freq drift coarse for clkin 0/1/2/3) <br> 3'b100 : <br> Dynamic_Clock_Monitoring_Status <br> 3'b101 : Internal debug <br> 3'b110 : Internal debug <br> 3'b111: Internal debug |
| NVMSPARE1_GENERIC | 19h | 7 | R/W | Oh | if set ' 1 ', Enables vdd padring functionality ( $0 \times 23[7]$ ) for padring selection. Keeping at 0 enables pad rail switch, select 1 to ensure no switch and keep rail to left |
|  |  | 6 | R/W | Oh | To map the external clock in select and allow the each pll independently to force into holdover mode. Each pll ctrl - refer 0x2B[3:0] |
| FUSE_PTR_GENERIC | 22h | 7:0 | R/W | Oh | One Hot Decode for Fuse Pointer: Bits $\{7,6,5,4,3,2,1,0\}$ correspond to fuse pointers for Pages 1, 2, 3, 4, A, B, C, D respectively <br> Page 4 is reserved and not used. |
| XO_4 GENERIC | 23h | 7 | R/W | Oh | ' 1 ' => Set VDDIO as VDDIN \& '0' => Set VDDIO as VDD for AU532x parts |
| BTOUT_IN_EN_GENERIC | 24h | 7:4 | R/W | Oh | One Hot Input Enable for the Clock Inputs 3:0 which are the 4 Clock inputs defined on Page 2: Active Low |
|  |  | 3:2 | R/W | Oh | One Hot Output Enable for Bottom Outputs 1:0 which are the 2 Bottom FixedOutputs defined on Page 3: Active Low |
|  |  | 1:0 | R/W | Oh | One Hot Output Enable for Top Outputs 1:0 which are the 2 Top Fixed-Outputs defined on Page 3: Active Low |


| Reg Name | Register Number | $\begin{gathered} \text { Bit } \\ \text { Range } \end{gathered}$ | Access Type | Default Value | Description |
| :---: | :---: | :---: | :---: | :---: | :---: |
| FLEXOUTPUT_EN_GENERIC | 25h | 7:0 | R/W | Oh | One Hot Output Enable for Outputs 7:0 which are the 8 Flex-Outputs defined on Page 3: Active Low |
| OEB_CTRL | 26h | 7:6 | R/W | Oh | Selects which PLL to run in free mode at wake up, this feature is enables by 0x23[5]: <br> 0 : PLLA <br> 1: PLLB <br> 2 : PLLC <br> 3 : PLLD |
|  |  | 5:4 | R/W | Oh | Reserved |
|  |  | 3:2 | R/W | Oh | To program delay between enabling PLLA and other PLLs in fuse locked modes. <br> 2'b00-4ms <br> 2'b01-40 ms <br> 2'b10-400 ms <br> 2'b11-4s |
|  |  | 1 | R/W | Oh | if set as '1' then set PLLX_OEB as '1' if loss of lock status is asserted for respective PLL |
|  |  | 0 | R/W | Oh | if set as '1' then set PLLX_OEB as '1' if there is XO clock loss |
| INTR_MASK_1_CONFIG | 27h | 7 | R/W | Oh | If set '1', Interrupt will not be generated for DRIFT in freq for clk_in3 |
|  |  | 6 | R/W | Oh | If set ' 1 ', Interrupt will not be generated for DRIFT in freq for clk_in2 |
|  |  | 5 | R/W | Oh | If set ' 1 ', Interrupt will not be generated for DRIFT in freq for clk_in1 |
|  |  | 4 | R/W | Oh | If set '1', Interrupt will not be generated for DRIFT in freq for clk_in0 |
|  |  | 3 | R/W | Oh | If set '1', Interrupt will not be generated for CLOCK LOSS for clk_in3 |
|  |  | 2 | R/W | Oh | If set '1', Interrupt will not be generated for CLOCK LOSS for clk_in2 |
|  |  | 1 | R/W | Oh | If set '1', Interrupt will not be generated for CLOCK LOSS for clk_in1 |
|  |  | 0 | R/W | Oh | If set '1', Interrupt will not be generated for CLOCK LOSS for clk_in0 |
| INTR_MASK_2_CONFIG | 28h | 7 | R/W | Oh | If set '1', Interrupt will not be generated for LOSS OF LOCK for PLLA |
|  |  | 6 | R/W | Oh | If set '1', Interrupt will not be generated for LOSS OF LOCK for PLLB |
|  |  | 5 | R/W | Oh | If set '1', Interrupt will not be generated for LOSS OF LOCK for PLLC |
|  |  | 4 | R/W | Oh | If set '1', Interrupt will not be generated for LOSS OF LOCK for PLLD |
|  |  | 3 | R/W | Oh | If set ' 1 ', Interrupt will not be generated for HOLDOVER FREEZE for PLLA |
|  |  | 2 | R/W | Oh | If set ' 1 ', Interrupt will not be generated for HOLDOVER FREEZE for PLLB |
|  |  | 1 | R/W | Oh | If set ' 1 ', Interrupt will not be generated for HOLDOVER FREEZE for PLLC |
|  |  | 0 | R/W | Oh | If set '1', Interrupt will not be generated for HOLDOVER FREEZE for PLLD |
| CHIP_FLEXIO_CONFIG | 29h | 7 | R/W | Oh | Reserved |
|  |  | 6 | R/W | Oh |  |
|  |  | 5 | R/W | Oh |  |
|  |  | 4 | R/W | Oh |  |
|  |  | 3 | R/W | Oh |  |


| Reg Name | Register Number | $\begin{gathered} \text { Bit } \\ \text { Range } \end{gathered}$ | Access Type | Default Value | Description |
| :---: | :---: | :---: | :---: | :---: | :---: |
| CHIP_FLEXIO_CONFIG | 29h | 2:0 | R/W | Oh | 010: AU5327 |
| I2C_GENERIC | 2Ah | 7 | R/W | Oh | Enable the new I2C Address (if changed from the default 0x69) |
|  |  | 6:0 | R/W | Oh | I2C Address (when changed from the default 0x6D) |
| XO2_GENERIC | 2Dh | 7:6 | R/W | Oh | OT3 GM programmability: Frequency selectivity configuration settings to support different frequency range of 3rd OT crystals |
|  |  | 5:0 | R/W | Oh | Crystal frequency trim settings |
| XO3_GENERIC | 2Eh | 7 | R/W | Oh | Crystal Pathway Settings: Use defaults from GUl based on crystal type and pathway used. |
|  |  | 6 | R/W | Oh |  |
|  |  | 5:3 | R/W | Oh |  |
|  |  | 2:0 | R/W | Oh |  |
| OE_PATTERN_GENERIC | 2Fh | 7:6 | R/W | Oh | PATTERN: :'01'/10'\} =] Efuse Locked, $\left\{{ }^{\prime} 001 / 11\right.$ ' $\}=$ =] Efuse NOT Locked |
|  |  | 5 | R/W | Oh | High Speed Enable for the I2C pads, HS_EN=0 ensures true I2C function (no pull up) for I2C. Don't care for SPI. |
|  |  | 4 | R/W | Oh | select '0' to enable manual wake up sequence |
|  |  | 3:0 | R/W | Oh | Crystal Pathway Settings: Use defaults from GUI based on crystal type and pathway used. |
| RESET_REGISTER | FEh | 7:0 | R/W | Oh | Register FE is a READ / WRITE register used to reset the chip. Writing $0 \times 01$ to this register will apply the reset to the digital and this is propagated asynchronously to the digital blocks. On writing a 0 to bit 0 of register FE , the reset is de-asserted and internal to the digital, we use the various clocks to perform the de-assertion reset synchronization for the appropriate clock domains. |
| PAGE_NUMBER | FFh | 7:0 | R/W | Oh | On all pages register FF is a READ / WRITE register used to change the page number |

Table 29 PAGE 1: Clock Monitor System Related registers
Registers from 11h to 4Fh are equivalent NVMCopy Registers for this Page.

| Reg Name | Register Number | Bit Range | Access Type | Default Value | Description |
| :---: | :---: | :---: | :---: | :---: | :---: |
| STATUS_CLKMON | 02h | 7 | R | 00h | Dynamic status for clk_in3_loss_OR_with_FD |
|  |  | 6 | R | 00h | Dynamic status for clk_in2_loss_OR_with_FD |
|  |  | 5 | R | 00h | Dynamic status for clk_in1_loss_OR_with_FD |
|  |  | 4 | R | 00h | Dynamic status for clk_in0_loss_OR_with_FD |
|  |  | 3 | R | 00h | Dynamic status for clk_in3_loss |
|  |  | 2 | R | 00h | Dynamic status for clk_in2_loss |
|  |  | 1 | R | 00h | Dynamic status for clk_in1_loss |
|  |  | 0 | R | 00h | Dynamic status for clk_in0_loss |
| NOTIFY_CLKMON | 03h | 7 | R/W | 1h | Sticky/Notify status for clk_in3_loss_OR_with_FD |
|  |  | 6 | R/W | 1h | Sticky/Notify status for clk_in2_loss_OR_with_FD |
|  |  | 5 | R/W | 1h | Sticky/Notify status for clk_in1_loss_OR_with_FD |
|  |  | 4 | R/W | 1h | Sticky/Notify status for clk_in0_loss_OR_with_FD |
|  |  | 3 | R/W | 1h | Sticky/Notify status for clk_in3_loss |
|  |  | 2 | R/W | 1h | Sticky/Notify status for clk_in2_loss |
|  |  | 1 | R/W | 1h | Sticky/Notify status for clk_in1_loss |
|  |  | 0 | R/W | 1h | Sticky/Notify status for clk_in0_loss |
| MASKb_CLKMON | 04h | 7 | R/W | 1h | Mask bit for NOTIFY_CLKMON (03h) <br> If programmed as '0': Mask sticky/Notify bit generation for 03h[7] |
|  |  | 6 | R/W | 1h | Mask bit for NOTIFY_CLKMON (03h) <br> If programmed as '0': Mask sticky/Notify bit generation for 03h[6] |
|  |  | 5 | R/W | 1h | Mask bit for NOTIFY_CLKMON (03h) <br> If programmed as '0': Mask sticky/Notify bit generation for 03h[5] |
|  |  | 4 | R/W | 1h | Mask bit for NOTIFY_CLKMON (03h) <br> If programmed as '0': Mask sticky/Notify bit generation for 03h[4] |
|  |  | 3 | R/W | 1h | Mask bit for NOTIFY_CLKMON (03h) <br> If programmed as '0': Mask sticky/Notify bit generation for 03h[3] |
|  |  | 2 | R/W | 1h | Mask bit for NOTIFY_CLKMON (03h) <br> If programmed as '0': Mask sticky/Notify bit generation for 03h[2] |


| Reg Name | Register Number | Bit Range | Access Type | Default Value | Description |
| :---: | :---: | :---: | :---: | :---: | :---: |
| MASKb_CLKMON | 04h | 1 | R/W | 1h | Mask bit for NOTIFY_CLKMON (03h) <br> If programmed as '0': Mask sticky/Notify bit generation for 03h[1] |
|  |  | 0 | R/W | 1h | Mask bit for NOTIFY_CLKMON (03h) <br> If programmed as '0': Mask sticky/Notify bit generation for 03h[0] |
| STATUS_FDCOARSE_CLKMON | 06h | 7 | R | 00h | Dynamic status for clk3_freq_coarse_drifted |
|  |  | 6 | R | 00h | Dynamic status for clk2_freq_coarse_drifted |
|  |  | 5 | R | 00h | Dynamic status for clk1_freq_coarse_drifted |
|  |  | 4 | R | 00h | Dynamic status for clk0_freq_coarse_drifted |
|  |  | 3 | R | 00h | Dynamic status for clk3_freq_fine_drifted |
|  |  | 2 | R | 00h | Dynamic status for clk2_freq_fine_drifted |
|  |  | 1 | R | 00h | Dynamic status for clk1_freq_fine_drifted |
|  |  | 0 | R | 00h | Dynamic status for clko_freq_fine_drifted |
| NOTIFY_FDCOARSE_CLKMON | 07h | 7 | R/W | 1h | Sticky/Notify status for clk3_freq_coarse_drifted |
|  |  | 6 | R/W | 1h | Sticky/Notify status for clk2_freq_coarse_drifted |
|  |  | 5 | R/W | 1h | Sticky/Notify status for clk1_freq_coarse_drifted |
|  |  | 4 | R/W | 1h | Sticky/Notify status for clkO_freq_coarse_drifted |
|  |  | 3 | R/W | 1h | Sticky/Notify status for clk3_freq_fine_drifted |
|  |  | 2 | R/W | 1h | Sticky/Notify status for clk2_freq_fine_drifted |
|  |  | 1 | R/W | 1h | Sticky/Notify status for clk1_freq_fine_drifted |
|  |  | 0 | R/W | 1h | Sticky/Notify status for clk0_freq_fine_drifted |
| MASKb_FDCOARSE_CLKMON | 08h | 7 | R/W | 1h | Mask bit <br> for NOTIFY_FDCOARSE_CLKMON (07h) <br> If programmed as '0’: Mask sticky/Notify bit generation for 07h[7] |
|  |  | 6 | R/W | 1h | Mask bit <br> for NOTIFY_FDCOARSE_CLKMON <br> (07h) <br> If programmed as '0’: Mask sticky/Notify bit generation for 07h[6] |


| Reg Name | Register Number | Bit Range | Access Type | Default Value | Description |
| :---: | :---: | :---: | :---: | :---: | :---: |
| MASKb_FDCOARSE_CLKMON | 08h | 5 | R/W | 1h | Mask bit for NOTIFY_FDCOARSE_CLKMON (07h) <br> If programmed as '0': Mask sticky/Notify bit generation for 07h[5] |
|  |  | 4 | R/W | 1h | Mask bit for NOTIFY_FDCOARSE_CLKMON (07h) <br> If programmed as '0': Mask sticky/Notify bit generation for 07h[4] |
|  |  | 3 | R/W | 1h | Mask bit for NOTIFY_FDCOARSE_CLKMON (07h) <br> If programmed as ' 0 ': Mask sticky/Notify bit generation for 07h[3] |
|  |  | 2 | R/W | 1h | Mask bit for NOTIFY_FDCOARSE_CLKMON (07h) <br> If programmed as '0': Mask sticky/Notify bit generation for 07h[2] |
|  |  | 1 | R/W | 1h | Mask bit for NOTIFY_FDCOARSE_CLKMON (07h) <br> If programmed as '0': Mask sticky/Notify bit generation for $07 \mathrm{~h}[1]$ |
|  |  | 0 | R/W | 1h | Mask bit for NOTIFY_FDCOARSE_CLKMON (07h) <br> If programmed as '0': Mask sticky/Notify bit generation for $07 \mathrm{~h}[0]$ |
| FD32_STATUS_COARSE_CLKMON | 0Dh | 7:0 | R/W | Oh | Reserved |
| FD10_STATUS_COARSE_CLKMON | 0Eh | 7:0 | R/W | Oh | Reserved |
| PRG_Directives_CLKMON | OFh | 7:3 | R/W | Oh | PRG_CMD Directives: <br> 5b1_1000: PROGRAM_EFUSE <br> 5b0_1100: READ_EFUSE <br> 5b0_0110: Copy NVM Copy to Settings <br> 5b1_1011: Proceed to Active |
|  |  | 2 | R/W | Oh | Spare |
|  |  | 1 | R/W | Oh | Escape to the PRG_CMD state from ACTIVE state |
| CL_REG2_CLKMON | 11h | 7:6 | R/W | Oh | IN3_VAL_TIME: timer setting for deassertion of clock loss for FD3 Values $\{0,1,2,3\}$ correspond to $\{2 \mathrm{~ms}, 100 \mathrm{~ms}$, $200 \mathrm{~ms}, 1 \mathrm{sec}\}$ |
|  |  | 5:4 | R/W | Oh | IN2_VAL_TIME : timer setting for deassertion of clock loss for FD2 Values $\{0,1,2,3\}$ correspond to $\{2 \mathrm{~ms}, 100 \mathrm{~ms}$, 200ms, 1sec $\}$ |
|  |  | 3:2 | R/W | Oh | IN1_VAL_TIME : timer setting for deassertion of clock loss for FD1 Values $\{0,1,2,3\}$ correspond to $\{2 \mathrm{~ms}, 100 \mathrm{~ms}$, 200ms, 1sec\} |


| Reg Name | Register Number | Bit Range | Access Type | Default Value | Description |
| :---: | :---: | :---: | :---: | :---: | :---: |
| CL_REG2_CLKMON | 11h | 1:0 | R/W | Oh | INO_VAL_TIME : timer setting for deassertion of clock loss for FDO Values $\{0,1,2,3\}$ correspond to $\{2 \mathrm{~ms}, 100 \mathrm{~ms}$, 200ms, 1sec\} |
| CL3_SET_THR2_CLKMON | 12h | 7:3 | R/W | Oh | SPARE |
|  |  | 2:0 | R/W | Oh | IN3 19-bit Threshold Value <br> GUI calculates the correct CL register threshold trigger value for IN3 |
| CL3_SET_THR1_CLKMON | 13h | 7:0 | R/W | Oh |  |
| CL3_SET_THRO_CLKMON | 14h | 7:0 | R/W | Oh |  |
| CL2_SET_THR2_CLKMON | 15h | 7:3 | R/W | Oh | SPARE |
|  |  | 2:0 | R/W | Oh | IN2 19-bit Threshold Value <br> GUI calculates the correct CL register threshold trigger value for IN2 |
| CL2_SET_THR1_CLKMON | 16h | 7:0 | R/W | Oh |  |
| CL2_SET_THR0_CLKMON | 17h | 7:0 | R/W | Oh |  |
| CL1_SET_THR2_CLKMON | 18h | 7:3 | R/W | Oh | SPARE |
|  |  | 2:0 | R/W | Oh | IN1 19-bit Threshold Value <br> GUI calculates the correct CL register threshold trigger value for IN1 |
| CL1_SET_THR1_CLKMON | 19h | 7:0 | R/W | Oh |  |
| CL1_SET_THR0_CLKMON | 1Ah | 7:0 | R/W | Oh |  |
| CLO_SET_THR2_CLKMON | 1Bh | 7:3 | R/W | Oh | SPARE |
|  |  | 2:0 | R/W | Oh | INO 19-bit Threshold Value <br> GUI calculates the correct CL register threshold trigger value for INO |
| CLO_SET_THR1_CLKMON | 1 Ch | 7:0 | R/W | Oh |  |
| CLO_SET_THRO_CLKMON | 1Dh | 7:0 | R/W | Oh |  |
| CL3_CLR_THR2_CLKMON | 1Eh | 7:3 | R/W | Oh | SPARE |
|  |  | 2:0 | R/W | Oh | IN3 19-bit Threshold Value <br> GUI calculates the correct CL register threshold clear value for IN3 |
| CL3_CLR_THR1_CLKMON | 1Fh | 7:0 | R/W | Oh |  |
| CL3_CLR_THR0_CLKMON | 20h | 7:0 | R/W | Oh |  |
| CL2_CLR_THR2_CLKMON | 21h | 7:3 | R/W | Oh | SPARE |
|  |  | 2:0 | R/W | Oh | IN2 19-bit Threshold Value <br> GUI calculates the correct CL register threshold clear value for IN2 |
| CL2_CLR_THR1_CLKMON | 22h | 7:0 | R/W | Oh |  |
| CL2_CLR_THR0_CLKMON | 23h | 7:0 | R/W | Oh |  |
| CL1_CLR_THR2_CLKMON | 24h | 7:3 | R/W | Oh | SPARE |
|  |  | 2:0 | R/W | Oh | IN1 19-bit Threshold Value <br> GUI calculates the correct CL register threshold clear value for IN1 |
| CL1_CLR_THR1_CLKMON | 25h | 7:0 | R/W | Oh |  |
| CL1_CLR_THR0_CLKMON | 26h | 7:0 | R/W | Oh |  |
| CLO_CLR_THR2_CLKMON | 27h | 7:3 | R/W | Oh | SPARE |
|  |  | 2:0 | R/W | Oh | INO 19-bit Threshold Value <br> GUI calculates the correct CL register threshold clear value for INO |
| CL0_CLR_THR1_CLKMON | 28h | 7:0 | R/W | Oh |  |
| CLO_CLR_THRO_CLKMON | 29h | 7:0 | R/W | Oh |  |


| Reg Name | Register Number | Bit Range | Access Type | Default Value | Description |
| :---: | :---: | :---: | :---: | :---: | :---: |
| FD_REG1_CLKMON | 2 Ah | 7:4 | R/W | Oh | FDCOARSE Enable Bit <br> Value of position $\{3,2,1,0\}$ correspond to Inputs $\{3,2,1,0\}$ |
|  |  | 3:0 | R/W | Oh | FDFINE Enable Bit Value of position $\{3,2,1,0\}$ correspond to Inputs $\{3,2,1,0\}$ |
| FD_REG2_CLKMON | 2 Bh | 7 | R/W | Oh | SPARE |
|  |  | 6:3 | R/W | Oh | XO divider configuration for FD monitors. GUI configures these dividers |
|  |  | 2:0 | R/W | Oh | Golden reference clock selection for frequency drift monitorsValue of 4 corresponds to the XO Reference Value of $\{3,2,1,0\}$ correspond to Inputs $\{3,2,1,0\}$ |
| FD_REG3_CLKMON | 2 Ch | 7:4 | R/W | Oh | FD3 divider configuration for FD monitors. GUI configures these dividers |
|  |  | 3:0 | R/W | Oh | FD2 divider configuration for FD monitors. GUI configures these dividers |
| FD_REG4_CLKMON | 2Dh | 7:4 | R/W | Oh | FD1 divider configuration for FD monitors. GUI configures these dividers |
|  |  | 3:0 | R/W | Oh | FDO divider configuration for FD monitors. GUI configures these dividers |
| FXOBYFIN3_LOG2_BAND | 2Eh | 7:4 | R/W | Oh | SPARE |
|  |  | 3:0 | R/W | Oh | IN3 FD monitor configuration bits calculated by GUI |
| FXOBYFIN2_LOG2_BAND | 2Fh | 7:6 | R/W | Oh | PATTERN :\{'01'/'10'\} => Efuse Locked, \{'00'/'11'\} => Efuse NOT Locked |
|  |  | 5:4 | R/W | Oh | SPARE |
|  |  | 3:0 | R/W | Oh | IN2 FD monitor configuration bits calculated by GUI |
| FXOBYFIN1_FIN0_LOG2_BAND | 30h | 7:4 | R/W | Oh | IN1 FD monitor configuration bits calculated by GUI |
|  |  | 3:0 | R/W | Oh | INO FD monitor configuration bits calculated by GUI |
| PLLA_ACT_SPARE_SEL | 31h | 7:6 | R/W | Oh | PLLA ACTIVE Clock Selection 0 : CLKINO, 1 : CLKIN1, 2 : CLKIN2, 3 : CLKIN3 |
|  |  | 5:4 | R/W | Oh | PLLA SPAREO Clock Selection <br> 0 : CLKINO, 1 : CLKIN1, 2 : CLKIN2, 3 : CLKIN3 |
|  |  | 3:2 | R/W | Oh | PLLA SPARE1 Clock Selection <br> 0 : CLKIN0, 1 : CLKIN1, 2 : CLKIN2, 3 : CLKIN3 |
|  |  | 1:0 | R/W | Oh | PLLA SPARE2 Clock Selection <br> 0 : CLKINO, 1 : CLKIN1, 2 : CLKIN2, 3 : CLKIN3 |
| FD3_CLR_THRFINE_CLKMON | 32h | 7:0 | R/W | Oh | IN3 FD clear threshold is calculated as 2X of FD3_CLR_THRFINE_CLKMON |


| Reg Name | Register Number | Bit Range | Access Type | Default Value | Description |
| :---: | :---: | :---: | :---: | :---: | :---: |
| FD2_CLR_THRFINE_CLKMON | 33h | 7:0 | R/W | Oh | IN2 FD clear threshold is calculated as 2 X of FD 2 CLR THRFINE CLKMON |
| FD1_CLR_THRFINE_CLKMON | 34h | 7:0 | R/W | Oh | IN1 FD clear threshold is calculated as 2X of FD1_CLR_THRFINE_CLKMON |
| FDO_CLR_THRFINE_CLKMON | 35h | 7:0 | R/W | Oh | INO FD clear threshold is calculated as 2 X of FDO _CLR_THRFINE_CLKMON |
| FD32_CLR_THRCOARSE_CLKMON | 36h | 7:4 | R/W | Oh | IN3 FD clear threshold is calculated as $100^{*}$ ( <br> FD3_CLR_THRCOARSE_CLKMON +1) |
|  |  | 3:0 | R/W | Oh | IN2 FD clear threshold is calculated as $100^{*}$ ( <br> FD2_CLR_THRCOARSE_CLKMON +1) |
| FD10_CLR_THRCOARSE_CLKMON | 37h | 7:4 | R/W | Oh | IN1 FD clear threshold is calculated as $100^{*}$ ( <br> FD1_CLR_THRCOARSE_CLKMON +1) |
|  |  | 3:0 | R/W | Oh | INO FD clear threshold is calculated as $100^{*}$ ( <br> FDO_CLR_THRCOARSE_CLKMON +1) |
| FD32_SET_THRCOARSE_CLKMON | 38h | 7:4 | R/W | Oh | IN3 FD set threshold is calculated as $100^{*}$ (SET <br> FD3_SET_THRCOARSE_CLKMON+1) |
|  |  | 3:0 | R/W | Oh | IN2 FD set threshold is calculated as 100*( <br> FD2_SET_THRCOARSE_CLKMON +1) |
| FD10_SET_THRCOARSE_CLKMON | 39h | 7:4 | R/W | Oh | IN1 FD set threshold is calculated as $100^{*}$ (SET TH <br> FD1_SET_THRCOARSE_CLKMON +1) |
|  |  | 3:0 | R/W | Oh | INO FD set threshold is calculated as $100^{*}$ ( <br> FDO_SET_THRCOARSE_CLKMON $+1)$ |
| FD3_FGBYFM2_CLKMON | 3Ah | 7:0 | R/W | Oh | IN3 FD monitor configuration bits calculated by GUI |
| FD3_FGBYFM1_CLKMON | 3Bh | 7:0 | R/W | Oh | IN3 FD monitor configuration bits calculated by GUI |
| FD3_FGBYFM0_CLKMON | 3Ch | 7:0 | R/W | Oh | IN3 FD monitor configuration bits calculated by GUI |
| FD2_FGBYFM2_CLKMON | 3Dh | 7:0 | R/W | Oh | IN2 FD monitor configuration bits calculated by GUI |
| FD2_FGBYFM1_CLKMON | 3Eh | 7:0 | R/W | Oh | IN2 FD monitor configuration bits calculated by GUI |
| FD2_FGBYFM0_CLKMON | 3Fh | 7:0 | R/W | Oh | IN2 FD monitor configuration bits calculated by GUI |


| Reg Name | Register Number | Bit Range | Access Type | Default Value | Description |
| :---: | :---: | :---: | :---: | :---: | :---: |
| FD1_FGBYFM2_CLKMON | 40h | 7:0 | R/W | Oh | IN1 FD monitor configuration bits calculated by GUI |
| FD1_FGBYFM1_CLKMON | 41h | 7:0 | R/W | Oh | IN1 FD monitor configuration bits calculated by GUI |
| FD1_FGBYFM0_CLKMON | 42h | 7:0 | R/W | Oh | IN1 FD monitor configuration bits calculated by GUI |
| FD0_FGBYFM2_CLKMON | 43h | 7:0 | R/W | Oh | INO FD monitor configuration bits calculated by GUI |
| FD0_FGBYFM1_CLKMON | 44h | 7:0 | R/W | Oh | INO FD monitor configuration bits calculated by GUI |
| FDO_FGBYFMO_CLKMON | 45h | 7:0 | R/W | Oh | INO FD monitor configuration bits calculated by GUI |
| CLKX_HITLESS_SW_SOURCE | 46h | 7:6 | R/W | Oh | $\begin{aligned} & 0: \text { CL3 } \\ & 1: C L 3+\text { FD3 COARSE } \\ & \text { 2:CL3 + FD3 FINE } \\ & 3: C L 3+\text { FD3 COARSE + FD3 FINE } \end{aligned}$ |
|  |  | 5:4 | R/W | Oh | $\begin{aligned} & 0: C L 2 \\ & 1: C L 2+\text { FD2 COARSE } \\ & 2: \mathrm{CL2}+\text { FD2 FINE } \\ & 3: \mathrm{CL} 2+\text { FD2 COARSE + FD2 FINE } \end{aligned}$ |
|  |  | 3:2 | R/W | Oh | $\begin{aligned} & 0: C L 1 \\ & 1: C L 1+\text { FD1 COARSE } \\ & 2: C L 1+\text { FD1 FINE } \\ & 3: C L 1+\text { FD1 COARSE + FD1 FINE } \end{aligned}$ |
|  |  | 1:0 | R/W | Oh | ```0:CLO 1: CLO + FDO COARSE 2: CLO + FDO FINE 3:CLO + FDO COARSE + FDO FINE``` |
| PLLB_ACT_SPARE_SEL | 49h | 7:6 | R/W | Oh | PLLB ACTIVE Clock Selection 0 : CLKINO, 1 : CLKIN1, 2 : CLKIN2, 3 CLKIN3 |
|  |  | 5:4 | R/W | Oh | PLLB SPAREO Clock Selection <br> 0 : CLKINO, 1 : CLKIN1, 2 : CLKIN2, 3 : CLKIN3 |
|  |  | 3:2 | R/W | Oh | PLLB SPARE1 Clock Selection 0 : CLKINO, 1 : CLKIN1, 2 : CLKIN2, 3 : CLKIN3 |
|  |  | 1:0 | R/W | Oh | PLLB SPARE2 Clock Selection <br> 0 : CLKINO, 1 : CLKIN1, 2 : CLKIN2, 3 : CLKIN3 |
| PLLC_ACT_SPARE_SEL | 4Ah | 7:6 | R/W | Oh | PLLC ACTIVE Clock Selection 0 : CLKINO, 1 : CLKIN1, 2 : CLKIN2, 3 CLKIN3 |
|  |  | 5:4 | R/W | Oh | PLLC SPAREO Clock Selection <br> 0 : CLKINO, 1 : CLKIN1, 2 : CLKIN2, 3 : CLKIN3 |
|  |  | 3:2 | R/W | Oh | PLLC SPARE1 Clock Selection 0 : CLKINO, 1 : CLKIN1, 2 : CLKIN2, 3 : CLKIN3 |


| Reg Name | Register Number | Bit Range | Access Type | Default Value | Description |
| :---: | :---: | :---: | :---: | :---: | :---: |
| PLLC_ACT_SPARE_SEL | 4Ah | 1:0 | R/W | Oh | PLLC SPARE2 Clock Selection <br> 0 : CLKINO, 1 : CLKIN1, 2 : CLKIN2, 3 : CLKIN3 |
| PLLD_ACT_SPARE_SEL | 4Bh | 7:6 | R/W | Oh | PLLD ACTIVE Clock Selection 0 : CLKINO, 1 : CLKIN1, 2 : CLKIN2, 3 : CLKIN3 |
|  |  | 5:4 | R/W | Oh | PLLD SPAREO Clock Selection 0 : CLKINO, 1 : CLKIN1, 2 : CLKIN2, 3 : CLKIN3 |
|  |  | 3:2 | R/W | Oh | PLLD SPARE1 Clock Selection 0 : CLKINO, 1 : CLKIN1, 2 : CLKIN2, 3 : CLKIN3 |
|  |  | 1:0 | R/W | Oh | PLLD SPARE2 Clock Selection <br> 0 : CLKINO, 1 : CLKIN1, 2 : CLKIN2, 3 : CLKIN3 |
| FD3_SET_THRFINE_CLKMON | 4Ch | 7:0 | R/W | Oh | IN3 FD set threshold is calculated as 2X of FD3_SET_THRFINE_CLKMON |
| FD2_SET_THRFINE_CLKMON | 4Dh | 7:0 | R/W | Oh | IN2 FD set threshold is calculated as 2 X of FD2_SET_THRFINE_CLKMON |
| FD1_SET_THRFINE_CLKMON | 4Eh | 7:0 | R/W | Oh | IN1 FD set threshold is calculated as 2X of FD1_SET_THRFINE_CLKMON |
| FDO_SET_THRFINE_CLKMON | 4Fh | 7:0 | R/W | Oh | INO FD set threshold is calculated as 2 X FDO_SET_THRFINE_CLKMON _THR |
| PAGE_NUMBER | FFh | 7:0 | R/W | Oh | On all pages register FF is a READ / WRITE register used to change the page number |

Table 30 PAGE 2 Input Systems Related Registers
Registers from 10h to 4Fh are equivalent NVMCopy Registers for this Page.

| Reg Name | Register <br> Number | Bit <br> Range | Access <br> Type | Default <br> Value | Description |
| :--- | :---: | :---: | :---: | :---: | :--- |
| PRG_Directives_INPAGE |  |  |  | PRG_CMD Directives: <br> 5b1_1000: PROGRAM_EFUSE <br> 5b0_1100: READ_EFUSE <br> 5b0_0110: Copy NVM Copy to Settings <br> $5 b 111011: P r o c e e d ~ t o ~ A c t i v e ~$ |  |


| Reg Name | Register Number | Bit Range | Access Type | Default Value | Description |
| :---: | :---: | :---: | :---: | :---: | :---: |
| CLKIN1_DIVN1_FRACD3_INPAGE | 28h | 7:0 | R/W | Oh | IN1 DIVN1 Divider Fractional Value Denominator |
| CLKIN1_DIVN1_FRACD4_INPAGE | 29h | 7:0 | R/W | Oh | IN1 DIVN1 Divider Fractional Value Denominator |
| CLKIN1_CFG1_INPAGE | 2Ah | 7:6 | R/W | Oh | IN1 DIVN1 Divider Integer Value |
|  |  | 5 | R/W | Oh | IN1 Direct Bypass for DIVN1 |
|  |  | 4 | R/W | Oh | IN1 Single Ended Enable, Default is Differential |
| CLKIN1_CFG2_INPAGE | 2Bh | 7:0 | R/W | Oh | IN1 FD Ramp monitor configuration bits calculated by GUI |
| CLKIN1_CFG3_INPAGE | 2Ch | 7:0 | R/W | Oh |  |
| CLKIN1_CFG4_INPAGE | 2Dh | 7:0 | R/W | Oh |  |
|  |  | 3:0 | R/W | Oh | IN1 divider configuration for FD Ramp monitors. GUI configures these dividers |
| CLKIN1_RAMPFD_MEAS_COUNT | 2Fh | 7:6 | R/W | Oh | PATTERN :\{'01'/'10'\} =] Efuse Locked , $\left\{{ }^{\prime} 00^{\prime} / 11\right.$ ' $\left.\}=\right]$ Efuse NOT Locked |
|  |  | 4:0 | R/W | Oh | Exponent that determines the ideal measurement count for Ramp: Calculated by the GUI. |
| CLKIN2_DIVN1_INT1_INPAGE | 30h | 7:0 | R/W | Oh | IN2 DIVN1 Divider Integer Value |
| CLKIN2_DIVN1_INT2_INPAGE | 31h | 7 | R/W | Oh | IN2 DIVN1 Integer Mode of Division |
|  |  | 5:0 | R/W | Oh | IN2 DIVN1 Divider Integer Value |
| CLKIN2_DIVN1_FRACN1_INPAGE | 32h | 7:0 | R/W | Oh | IN2 DIVN1 Divider Fractional Value Numerator |
| CLKIN2_DIVN1_FRACN2_INPAGE | 33h | 7:0 | R/W | Oh | IN2 DIVN1 Divider Fractional Value Numerator |
| CLKIN2_DIVN1_FRACN3_INPAGE | 34h | 7:0 | R/W | Oh | IN2 DIVN1 Divider Fractional Value Numerator |
| CLKIN2_DIVN1_FRACN4_INPAGE | 35h | 7:0 | R/W | Oh | IN2 DIVN1 Divider Fractional Value Numerator |
| CLKIN2_DIVN1_FRACD1_INPAGE | 36h | 7:0 | R/W | Oh | IN2 DIVN1 Divider Fractional Value Denominator |
| CLKIN2_DIVN1_FRACD2_INPAGE | 37h | 7:0 | R/W | Oh | IN2 DIVN1 Divider Fractional Value Denominator |
| CLKIN2_DIVN1_FRACD3_INPAGE | 38h | 7:0 | R/W | Oh | IN2 DIVN1 Divider Fractional Value Denominator |
| CLKIN2_DIVN1_FRACD4_INPAGE | 39h | 7:0 | R/W | Oh | IN2 DIVN1 Divider Fractional Value Denominator |
| CLKIN2_CFG1_INPAGE | 3Ah | 7:6 | R/W | Oh | IN2 DIVN1 Divider Integer Value |
|  |  | 5 | R/W | Oh | IN2 Direct Bypass for DIVN1 |
|  |  | 4 | R/W | Oh | IN2 Single Ended Enable, Default is Differential |
| CLKIN2_CFG2_INPAGE | 3Bh | 7:0 | R/W | Oh | IN2 FD Ramp monitor configuration bits calculated by GUI |
| CLKIN2_CFG3_INPAGE | 3Ch | 7:0 | R/W | Oh |  |
| CLKIN2_CFG4_INPAGE | 3Dh | 7:0 | R/W | Oh |  |
| CLKIN2_OUTSEL_INPAGE | 3Eh | 7:4 | R/W | Oh | Internal FD Ramp related settings |
|  |  | 3:0 | R/W | Oh | IN2 divider configuration for FD Ramp monitors. GUI configures these dividers |
|  |  | 4:0 | R/W | Oh | Exponent that determines the ideal measurement count for Ramp: Calculated by the GUI. |
| CLKIN3_DIVN1_INT1_INPAGE | 40h | 7:0 | R/W | Oh | IN3 DIVN1 Divider Integer Value |
| CLKIN3_DIVN1_INT2_INPAGE | 41h | 7 | R/W | Oh | IN3 DIVN1 Integer Mode of Division |
|  |  | 5:0 | R/W | Oh | IN3 DIVN1 Divider Integer Value |
| CLKIN3_DIVN1_FRACN1_INPAGE | 42h | 7:0 | R/W | Oh | IN3 DIVN1 Divider Fractional Value Numerator |
| CLKIN3_DIVN1_FRACN2_INPAGE | 43h | 7:0 | R/W | Oh | IN3 DIVN1 Divider Fractional Value Numerator |
| CLKIN3_DIVN1_FRACN3_INPAGE | 44h | 7:0 | R/W | Oh | IN3 DIVN1 Divider Fractional Value Numerator |
| CLKIN3_DIVN1_FRACN4_INPAGE | 45h | 7:0 | R/W | Oh | IN3 DIVN1 Divider Fractional Value Numerator |
| CLKIN3_DIVN1_FRACD1_INPAGE | 46h | 7:0 | R/W | Oh | IN3 DIVN1 Divider Fractional Value Denominator |


| Reg Name | Register Number | $\begin{gathered} \text { Bit } \\ \text { Range } \end{gathered}$ | Access Type | Default Value | Description |
| :---: | :---: | :---: | :---: | :---: | :---: |
| CLKIN3_DIVN1_FRACD2_INPAGE | 47h | 7:0 | R/W | Oh | IN3 DIVN1 Divider Fractional Value Denominator |
| CLKIN3_DIVN1_FRACD3_INPAGE | 48h | 7:0 | R/W | Oh | IN3 DIVN1 Divider Fractional Value Denominator |
| CLKIN3_DIVN1_FRACD4_INPAGE | 49h | 7:0 | R/W | Oh | IN3 DIVN1 Divider Fractional Value Denominator |
| CLKIN3_CFG1_INPAGE | 4Ah | 7:6 | R/W | Oh | IN3 DIVN1 Divider Integer Value |
|  |  | 5 | R/W | Oh | IN3 Direct Bypass for DIVN1 |
|  |  | 4 | R/W | Oh | IN3 CLKIN3 Single Ended Enable, Default is Differential |
| CLKIN3_CFG2_INPAGE | 4Bh | 7:0 | R/W | Oh | IN3 FD Ramp monitor configuration bits calculated by GUI |
| CLKIN3_CFG3_INPAGE | 4Ch | 7:0 | R/W | Oh |  |
| CLKIN3_CFG4_INPAGE | 4Dh | 7:0 | R/W | Oh |  |
| CLKIN3_OUTSEL_INPAGE | 4Eh | 7:6 | R/W | Oh | SPARE |
|  |  | 5:4 | R/W | Oh | SPARE |
|  |  | 3:0 | R/W | Oh | IN3 divider configuration for FD Ramp monitors. GUI configures these dividers |
| CLKIN3_RAMPFD_MEAS_COUNT | 4Fh | 4:0 | R/W | Oh | Exponent that determines the ideal measurement count for Ramp: Calculated by the GUI. |
| PAGE_NUMBER | FFh | 7:0 | R/W | Oh | On all pages register FF is a READ / WRITE register used to change the page number |

Table 31 PAGE 3 : Output System and Output Dividers Related Registers.
Registers from 18h to 67h are equivalent NVMCopy Registers for this Page.

| Reg Name | Register Number | Bit Range | Access Type | Default Value | Description |
| :---: | :---: | :---: | :---: | :---: | :---: |
| PRG_Directives_OUTPAGE | OFh | 7:3 | R/W | Oh | PRG_CMD Directives: <br> 5b1_1000: PROGRAM_EFUSE <br> 5b0_1100: READ_EFUSE <br> 5b0_0110: Copy NVM Copy to Settings <br> 5b1_1011: Proceed to Active |
|  |  | 2 | R/W | Oh | Spare |
|  |  | 1 | R/W | Oh | Escape to the PRG_CMD state from ACTIVE state |
|  |  | 2 | R/W | Oh |  |
|  |  | 1 | R/W | Oh |  |
|  |  | 0 | R/W | Oh |  |
|  |  | 4:3 | R/W | Oh | VDD_DEF : VDD \{1.8(00) 2.5(01) 3.3(10). <br> VDD Definition for this particular Output |
| DIVO0_MISC2_OUTPAGE | 15h | 2:0 | R/W | Oh | DRV_TYPE : Output Driver <br> Standard: <br> Ob010: DC Coupled CML <br> 0b011: DC Coupled HCSL <br> 0b000: LVDS (Can be AC Coupled or DC <br> Coupled) <br> Ob100: Boosted LVDS (Use for LVPECL <br> like swings with AC Coupled loads) <br> 0b001: DC Coupled LVPECL (with <br> Common mode current) <br> 0b101: DC Coupled LVPECL2 (without <br> Common mode current) |
| DIVO1_DIV2_OUTPAGE | 18h | 7:4 | R/W | Oh | Spare |
|  |  | 3:0 | R/W | Oh | DIVO Divider for this output: Bits [19:16] |
| DIVO1_DIV1_OUTPAGE | 19h | 7:0 | R/W | Oh | DIVO Divider for this output: Bits [15:8] |
| DIVO1_DIV0_OUTPAGE | 1Ah | 7:0 | R/W | Oh | DIVO Divider for this output: Bits [7:0] |
| DIVO1_PROG1_OUTPAGE | 1Bh | 7:6 | R/W | Oh | CMOS Driver Phase Selection |
|  |  | 2:0 | R/W | Oh | Single Ended Driver Programming of strength: Use 0b111= 0d7 |
| DIVO1_PROG0_OUTPAGE | 1Ch | 7:0 | R/W | Oh | Programmable Output Delay <br> PRG_DELAY[5:0] is the coarse delay on the clock output. It determines relative delay on this clock programmable from 0 to 63 VCO clock delays based on this number <br> PRG_DELAY[7:6] is the fine delay on the clock output. It determines relative delay on this clock programmable in $3,2,1,0$ times 30 ps based on this 2 bit code |


| Reg Name | Register Number | Bit Range | Access Type | Default Value | Description |
| :---: | :---: | :---: | :---: | :---: | :---: |
|  |  | 4:3 | R/W | Oh | VDD_DEF : VDD $\{1.8(00)$ 2.5(01) 3.3(10). <br> VDD Definition for this particular Output |
| DIVO1_MISC2_OUTPAGE | 1Dh | 2:0 | R/W | Oh | DRV_TYPE : Output Driver Standard: <br> 0b010: DC Coupled <br> CML <br> 0b011: DC Coupled <br> HCSL <br> 0b000: LVDS (Can be AC Coupled or DC <br> Coupled) <br> 0b100: Boosted LVDS (Use for LVPECL <br> like swings with AC Coupled loads) <br> 0b001: DC Coupled LVPECL (with <br> Common mode current) <br> 0b101: DC Coupled LVPECL2 (without <br> Common mode current) |
| DIVO2_DIV2_OUTPAGE | 20h | 7:4 | R/W | Oh | Spare |
|  |  | 3:0 | R/W | Oh | DIVO Divider for this output: Bits [19:16] |
| DIVO2_DIV1_OUTPAGE | 21h | 7:0 | R/W | Oh | DIVO Divider for this output: Bits [15:8] |
| DIVO2_DIV0_OUTPAGE | 22h | 7:0 | R/W | Oh | DIVO Divider for this output: Bits [7:0] |
| DIVO2_PROG1_OUTPAGE | 23h | 7:6 | R/W | Oh | CMOS Driver Phase Selection phase_sel<1> phase_sel<0> ODR_P ODR_N $\begin{array}{llll} 0 & 0 & \text { CLKP } & \text { CLKN } \\ 0 & 1 & \text { CLKP } & \text { CLKP } \\ 1 & 0 & \text { CLKN } & \text { CLKN } \\ 1 & 1 & \text { CLKN } & \text { CLKP } \end{array}$ |
|  |  | 2:0 | R/W | Oh | Single Ended Driver Programming of strength: Use 0b111=0d7 |
| DIVO2_PROGO_OUTPAGE | 24h | 7:0 | R/W | Oh | Programmable Output Delay <br> PRG_DELAY[5:0] is the coarse delay on the clock output. It determines relative delay on this clock programmable from 0 to 63 VCO clock delays based on this number <br> PRG_DELAY[7:6] is the fine delay on the clock output. It determines relative delay on this clock programmable in $3,2,1,0$ times 30 ps based on this 2 bit code |
|  |  | 4:3 | R/W | Oh | VDD_DEF : VDD $\{1.8(00) 2.5(01) 3.3(10)$. <br> VDD Definition for this particular Output |
| DIVO2_MISC2_OUTPAGE | 25h | 2:0 | R/W | Oh | DRV_TYPE : Output Driver Standard: <br> 0b010: DC Coupled <br> CML <br> 0b011: DC Coupled <br> HCSL <br> Ob000: LVDS (Can be AC Coupled or DC Coupled) <br> Ob100: Boosted LVDS (Use for LVPECL <br> like swings with AC Coupled loads) <br> 0b001: DC Coupled LVPECL (with <br> Common mode current) <br> 0b101: DC Coupled LVPECL2 (without <br> Common mode current) |
| DIVO3_DIV2_OUTPAGE | 28h | 7:4 | R/W | Oh | spare |
|  |  | 3:0 | R/W | Oh | DIVO Divider for this output: Bits [19:16] |


| Reg Name | Register Number | Bit Range | Access Type | Default Value | Description |
| :---: | :---: | :---: | :---: | :---: | :---: |
| DIVO3_DIV1_OUTPAGE | 29h | 7:0 | R/W | Oh | DIVO Divider for this output: Bits [15:8] |
| DIVO3_DIV0_OUTPAGE | 2 Ah | 7:0 | R/W | Oh | DIVO Divider for this output: Bits [7:0] |
| DIVO3_PROG1_OUTPAGE | 2 Bh | 7:6 | R/W | Oh | CMOS Driver Phase Selection phase_sel<1> phase_sel<0> ODR_P ODR_N |
|  |  | 2:0 | R/W | Oh | Single Ended Driver Programming of strength: Use 0b111=0d7 |
| DIVO3_PROG0_OUTPAGE | 2Ch | 7:0 | R/W | Oh | Programmable Output Delay <br> PRG_DELAY[5:0] is the coarse delay on the clock output. It determines relative delay on this clock programmable from 0 to 63 VCO clock delays based on this number <br> PRG_DELAY[7:6] is the fine delay on the clock output. It determines relative delay on this clock programmable in $3,2,1,0$ times 30 ps based on this 2 bit code |
|  |  | 4:3 | R/W | Oh | VDD_DEF : VDD \{1.8(00) 2.5(01) 3.3(10). <br> VDD Definition for this particular Output |
| DIVO3_MISC2_OUTPAGE | 2Dh | 2:0 | R/W | Oh | DRV_TYPE : Output Driver Standard: <br> 0b010: DC Coupled CML <br> 0b011: DC Coupled HCSL <br> 0b000: LVDS (Can be AC Coupled or DC Coupled) <br> 0b100: Boosted LVDS (Use for LVPECL like swings with AC Coupled loads) 0b001: DC Coupled LVPECL (with Common mode current) 0b101: DC Coupled LVPECL2 (without Common mode current) |
| DIVO3_MISC0_OUTPAGE | 2Fh | 7:6 | R/W | Oh | PATTERN: :'01'/10'\} =] Efuse Written, $\left\{{ }^{\prime} 00 / ' 11\right.$ ' $\left.\}=\right]$ Efuse NOT Written |
|  |  | 4 | R/W | Oh | PRG_DELAY[8:6] is the fine delay on the clock output. It determines relative delay on this clock programmable in 4,3, 2, 1, 0 times 30 ps based on this 3 bit code |
|  |  | 3 | R/W | Oh | $\{0,0,0,0\}$ External Termination, Differential Output <br> $\{0,0,1,0\}$ Internal Pull Up, Differential Output <br> $\{0,0,0,1\}$ Internal Pull Dn, Differential Output <br> $\{0,1,0,0\}$ CMOS On OutP, Nothing on OutN <br> $\{1,0,0,0\}$ Nothing on OutP, CMOS on OutN <br> $\{1,1,0,0\}$ CMOS on OutP, CMOS on OutN |
|  |  | 2 | R/W | Oh |  |
|  |  | 1 | R/W | Oh |  |
|  |  | 0 | R/W | Oh |  |
|  |  | 2 | R/W | Oh |  |
|  |  | 1 | R/W | Oh |  |
|  |  | 0 | R/W | Oh |  |
| DIVO4_MISC2_OUTPAGE | 35h | 4:3 | R/W | Oh | VDD_DEF : VDD $\{1.8(00)$ 2.5(01) 3.3(10). <br> VDD Definition for this particular Output |


| Reg Name | Register Number | Bit Range | Access Type | Default Value | Description |
| :---: | :---: | :---: | :---: | :---: | :---: |
| DIVO4_MISC2_OUTPAGE | 35h | 2:0 | R/W | Oh | DRV_TYPE : Output Driver Standard: <br> 0b010: DC Coupled <br> CML <br> 0b011: DC Coupled HCSL <br> 0b000: LVDS (Can be AC Coupled or DC <br> Coupled) <br> 0b100: Boosted LVDS (Use for LVPECL <br> like swings with AC Coupled loads) <br> Ob001: DC Coupled LVPECL (with <br> Common mode current) <br> Ob101: DC Coupled LVPECL2 (without <br> Common mode current) |
| DIVO5_DIV2_OUTPAGE | 38h | 7:4 | R/W | Oh | spare |
|  |  | 3:0 | R/W | Oh | DIVO Divider for this output: Bits [19:16] |
| DIVO5_DIV1_OUTPAGE | 39h | 7:0 | R/W | Oh | DIVO Divider for this output: Bits [15:8] |
| DIVO5_DIV0_OUTPAGE | 3Ah | 7:0 | R/W | Oh | DIVO Divider for this output: Bits [7:0] |
| DIVO5_PROG1_OUTPAGE | 3Bh | 7:6 | R/W | Oh | CMOS Driver Phase Selection $\begin{array}{lllll}\text { phase_sel<1> phase_sel<0> ODR_P } \\ \text { ODR_N } \\ 0 & 0 & \text { CLKP } & \text { CLKN } \\ 0 & 1 & \text { CLKP } & \text { CLKP } \\ 1 & 0 & \text { CLKN } & \text { CLKN } \\ 1 & 1 & \text { CLKN } & \text { CLKP }\end{array}$ |
|  |  | 2:0 | R/W | Oh | Single Ended Driver Programming of strength: Use 0b111= 0d7 |
| DIVO5_PROG0_OUTPAGE | 3Ch | 7:0 | R/W | Oh | Programmable Output Delay <br> PRG_DELAY[5:0] is the coarse delay on the clock output. It determines relative delay on this clock programmable from 0 to 63 VCO clock delays based on this number <br> PRG_DELAY[7:6] is the fine delay on the clock output. It determines relative delay on this clock programmable in $3,2,1,0$ times 30 ps based on this 2 bit code |
|  |  | 4:3 | R/W | Oh | VDD_DEF : VDD \{1.8(00) 2.5(01) 3.3(10). <br> VDD Definition for this particular Output |
| DIVO5_MISC2_OUTPAGE | 3Dh | 2:0 | R/W | Oh | DRV_TYPE : Output Driver Standard: <br> 0b010: DC Coupled <br> CML <br> 0b011: DC Coupled HCSL <br> 0b000: LVDS (Can be AC Coupled or DC Coupled) <br> 0b100: Boosted LVDS (Use for LVPECL <br> like swings with AC Coupled loads) <br> 0b001: DC Coupled LVPECL (with <br> Common mode current) <br> 0b101: DC Coupled LVPECL2 (without <br> Common mode current) |
| DIVO6_DIV2_OUTPAGE | 40h | 7:4 | R/W | Oh | spare |
|  |  | 3:0 | R/W | Oh | DIVO Divider for this output: Bits [19:16] |
| DIVO6_DIV1_OUTPAGE | 41h | 7:0 | R/W | Oh | DIVO Divider for this output: Bits [15:8] |
| DIVO6_DIV0_OUTPAGE | 42h | 7:0 | R/W | Oh | DIVO Divider for this output: Bits [7:0] |


| Reg Name | Register Number | Bit <br> Range | Access Type | Default Value | Description |
| :---: | :---: | :---: | :---: | :---: | :---: |
| DIVO6_PROG1_OUTPAGE | 43h | 7:6 | R/W | Oh | CMOS Driver Phase Selection |
|  |  | 2:0 | R/W | Oh | Single Ended Driver Programming of strength: Use 0b111= 0d7 |
| DIVO6_PROG0_OUTPAGE | 44h | 7:0 | R/W | Oh | Programmable Output Delay <br> PRG_DELAY[5:0] is the coarse delay on the clock output. It determines relative delay on this clock programmable from 0 to 63 VCO clock delays based on this number <br> PRG_DELAY[7:6] is the fine delay on the clock output. It determines relative delay on this clock programmable in $3,2,1,0$ times 30 ps based on this 2 bit code |
|  |  | 4:3 | R/W | Oh | VDD_DEF : VDD \{1.8(00) 2.5(01) 3.3(10). <br> VDD Definition for this particular Output |
| DIVO6_MISC2_OUTPAGE | 45h | 2:0 | R/W | Oh | DRV_TYPE : Output Driver Standard: <br> 0b010: DC Coupled CML <br> 0b011: DC Coupled HCSL <br> 0b000: LVDS (Can be AC Coupled or DC Coupled) <br> 0b100: Boosted LVDS (Use for LVPECL like swings with AC Coupled loads) <br> 0b001: DC Coupled LVPECL (with <br> Common mode current) <br> 0b101: DC Coupled LVPECL2 (without <br> Common mode current) |
| DIVO6_MISC0_OUTPAGE | 47h | 7:6 | R/W | Oh | DIVO Divider for OB output: Bits [33:32] |
|  |  | 4 | R/W | Oh | PRG_DELAY[8:6] is the fine delay on the clock output. It determines relative delay on this clock programmable in $4,3,2,1,0$ times 30 ps based on this 3 bit code |
|  |  | 3 | R/W | Oh | \{0, 0, 0, 0\} External Termination, |
|  |  | 2 | R/W | Oh |  |
|  |  | 1 | R/W | Oh | Output |
|  |  | 0 | R/W | Oh | Output <br> $\{0,1,0,0\}$ CMOS On OutP, Nothing on OutN <br> $\{1,0,0,0\}$ Nothing on OutP, CMOS on OutN <br> $\{1,1,0,0\}$ CMOS on OutP, CMOS on OutN |
| DIVO7_DIV2_OUTPAGE | 48h | 7:4 | R/W | Oh | spare |
|  |  | 3:0 | R/W | Oh | DIVO Divider for this output: Bits [19:16] |
| DIVO7_DIV1_OUTPAGE | 49h | 7:0 | R/W | Oh | DIVO Divider for this output: Bits [15:8] |
| DIVO7_DIV0_OUTPAGE | 4Ah | 7:0 | R/W | Oh | DIVO Divider for this output: Bits [7:0] |


| Reg Name | Register Number | Bit Range | Access Type | Default Value | Description |
| :---: | :---: | :---: | :---: | :---: | :---: |
| DIVO7_PROG1_OUTPAGE | 4Bh | 7:6 | R/W | Oh | CMOS Driver Phase Selection |
|  |  | 2:0 | R/W | Oh | Single Ended Driver Programming of strength: Use 0b111=0d7 |
| DIVO7_PROG0_OUTPAGE | 4Ch | 7:0 | R/W | Oh | Programmable Output Delay <br> PRG_DELAY[5:0] is the coarse delay on the clock output. It determines relative delay on this clock programmable from 0 to 63 VCO clock delays based on this number <br> PRG_DELAY[7:6] is the fine delay on the clock output. It determines relative delay on this clock programmable in $3,2,1,0$ times 30 ps based on this 2 bit code |
|  |  | 4:3 | R/W | Oh | VDD_DEF : VDD \{1.8(00) 2.5(01) 3.3(10). <br> VDD Definition for this particular Output |
| DIVO7_MISC2_OUTPAGE | 4Dh | 2:0 | R/W | Oh | DRV_TYPE : Output Driver Standard: <br> 0b010: DC Coupled CML <br> 0b011: DC Coupled HCSL <br> 0b000: LVDS (Can be AC Coupled or DC <br> Coupled) <br> Ob100: Boosted LVDS (Use for LVPECL <br> like swings with AC Coupled loads) <br> 0b001: DC Coupled LVPECL (with <br> Common mode current) <br> 0b101: DC Coupled LVPECL2 (without <br> Common mode current) |
| DIVO7_MISC0_OUTPAGE | 4Fh | 7:6 | R/W | Oh | DIVO Divider for OB output: Bits [31:30] |
|  |  | 4 | R/W | Oh | PRG_DELAY[8:6] is the fine delay on the clock output. It determines relative delay on this clock programmable in 4,3,2,1,0 times 30 ps based on this 3 bit code |
|  |  | 3 | R/W | Oh | $\{0,0,0,0\}$ External Termination, Differential Output <br> $\{0,0,1,0\}$ Internal Pull Up, Differential Output <br> $\{0,0,0,1\}$ Internal Pull Dn, Differential Output <br> $\{0,1,0,0\}$ CMOS On OutP, Nothing on OutN <br> $\{1,0,0,0\}$ Nothing on OutP, CMOS on OutN <br> $\{1,1,0,0\}$ CMOS on OutP, CMOS on OutN |
|  |  | 2 | R/W | Oh |  |
|  |  | 1 | R/W | Oh |  |
|  |  | 0 | R/W | Oh |  |
| DIVO0T_DIV2_OUTPAGE | 50h | 7:4 | R/W | Oh | spare |
|  |  | 3:0 | R/W | Oh | DIVO Divider for this output: Bits [19:16] |
| DIVO0T_DIV1_OUTPAGE | 51h | 7:0 | R/W | Oh | DIVO Divider for this output: Bits [15:8] |
| DIVOOT_DIVO_OUTPAGE | 52h | 7:0 | R/W | Oh | DIVO Divider for this output: Bits [7:0] |


| Reg Name | Register Number | Bit <br> Range | Access Type | Default Value | Description |
| :---: | :---: | :---: | :---: | :---: | :---: |
| DIVO0T_PROG1_OUTPAGE | 53h | 7:6 | R/W | Oh | CMOS Driver Phase Selection |
|  |  | 2:0 | R/W | Oh | Single Ended Driver Programming of strength: Use 0b111= 0d7 |
| DIVOOT_PROG0_OUTPAGE | 54h | 7:0 | R/W | Oh | Programmable Output Delay <br> PRG_DELAY[5:0] is the coarse delay on the clock output. It determines relative delay on this clock programmable from 0 to 63 VCO clock delays based on this number <br> PRG_DELAY[7:6] is the fine delay on the clock output. It determines relative delay on this clock programmable in $3,2,1,0$ times 30 ps based on this 2 bit code |
|  |  | 4:3 | R/W | Oh | VDD_DEF : VDD \{1.8(00), 2.5(01), 3.3(10)\}. VDD Definition for this particular output. |
| DIVOOT_MISC2_OUTPAGE | 55h | 2:0 | R/W | Oh | DRV_TYPE : Output Driver Standard: <br> 0b010: DC Coupled <br> CML <br> 0b011: DC Coupled HCSL <br> 0b000: LVDS (Can be AC Coupled or DC Coupled) <br> 0b100: Boosted LVDS (Use for LVPECL like swings with AC Coupled loads) <br> 0b001: DC Coupled LVPECL (with Common mode current) <br> 0b101: DC Coupled LVPECL2 (without Common mode current) |
| DIVOOT_MISC0_OUTPAGE | 57h | 7:6 | R/W | Oh | DIVO Divider for OB output: Bits [29:28] |
|  |  | 4 | R/W | Oh | PRG_DELAY[8:6] is the fine delay on the clock output. It determines relative delay on this clock programmable in $4,3,2,1,0$ times 30 ps based on this 3 bit code |
|  |  | 3 | R/W | Oh | $\{0,0,0,0\}$ External Termination, Differential Output <br> $\{0,0,1,0\}$ Internal Pull Up, Differential Output <br> $\{0,0,0,1\}$ Internal Pull Dn, Differential Output <br> $\{0,1,0,0\}$ CMOS On OutP, Nothing on OutN <br> $\{1,0,0,0\}$ Nothing on OutP, CMOS on OutN <br> $\{1,1,0,0\}$ CMOS on OutP, CMOS on OutN |
|  |  | 2 | R/W | Oh |  |
|  |  | 1 | R/W | Oh |  |
|  |  | 0 | R/W | Oh |  |
| DIVO1T_MISC0_OUTPAGE | 5Fh | 7:6 | R/W | Oh | DIVO Divider for 0B output: Bits [27:26] |
| DIVO0B_DIV2_OUTPAGE | 60h | 7:0 | R/W | Oh | DIVO Divider for 0B output: Bits [23:16] |
| DIVO0B_DIV1_OUTPAGE | 61h | 7:0 | R/W | Oh | DIVO Divider for 0B output: Bits [15:8] |


| Reg Name | Register Number | Bit Range | Access Type | Default Value | Description |
| :---: | :---: | :---: | :---: | :---: | :---: |
| DIVOOB_DIVO_OUTPAGE | 62h | 7:0 | R/W | Oh | DIVO Divider for OB output: Bits [7:0] |
| DIVO0B_PROG1_OUTPAGE | 63h | 7:6 | R/W | Oh | CMOS Driver Phase Selection    <br> phase_sel<1> phase_sel<0> ODRP ODR_N  <br> 0 0 CLKP CLKN <br> 0 1 CLKP CLKP <br> 1 0 CLKN CLKN <br> 1 1 CLKN CLKP |
|  |  | 2:0 | R/W | Oh | Single Ended Driver Programming of strength: Use 0b111=0d7 |
| DIVOOB_PROGO_OUTPAGE | 64h | 7:0 | R/W | Oh | Programmable Output Delay <br> PRG_DELAY[5:0] is the coarse delay on the clock output. It determines relative delay on this clock programmable from 0 to 63 VCO clock delays based on this number <br> PRG_DELAY[7:6] is the fine delay on the clock output. It determines relative delay on this clock programmable in $3,2,1,0$ times 30 ps based on this 2 bit code |
|  |  | 4:3 | R/W | Oh | VDD_DEF : VDD \{1.8(00) 2.5(01) 3.3(10). VDD Definition for this particular Output |
| DIVOOB_MISC2_OUTPAGE | 65h | 2:0 | R/W | Oh | DRV_TYPE : Output Driver Standard: <br> 0b010: DC Coupled <br> CML <br> 0b011: DC Coupled <br> HCSL <br> 0b000: LVDS (Can be AC Coupled or DC Coupled) <br> 0b100: Boosted LVDS (Use for LVPECL <br> like swings with AC Coupled loads) <br> 0b001: DC Coupled LVPECL (with <br> Common mode current) <br> 0b101: DC Coupled LVPECL2 (without <br> Common mode current) |
| DIVOOB_MISCO_OUTPAGE | 67h | 7:6 | R/W | Oh | DIVO Divider for OB output: Bits [27:26] |
|  |  | 4 | R/W | Oh | PRG_DELAY[8:6] is the fine delay on the clock output. It determines relative delay on this clock programmable in $4,3,2,1,0$ times 30 ps based on this 3 bit code |
|  |  | 3 | R/W | Oh | $\{0,0,0,0\}$ External Termination, Differential Output <br> $\{0,0,1,0\}$ Internal Pull Up, Differential Output <br> $\{0,0,0,1\}$ Internal Pull Dn, Differential Output <br> $\{0,1,0,0\}$ CMOS On OutP, Nothing on OutN <br> $\{1,0,0,0\}$ Nothing on OutP, CMOS on OutN <br> $\{1,1,0,0\}$ CMOS on OutP, CMOS on OutN |
|  |  | 2 | R/W | Oh |  |
|  |  | 1 | R/W | Oh |  |
|  |  | 0 | R/W | Oh |  |
|  |  | 2 | R/W | Oh |  |
|  |  | 1 | R/W | Oh |  |
|  |  | 0 | R/W | Oh |  |


| Reg Name | Register <br> Number | Bit <br> Range | Access <br> Type | Default <br> Value | Description |
| :--- | :---: | :---: | :---: | :---: | :--- |
| RESET_REGISTER | FEh | $7: 0$ | R/W | Register FF is a READ / WRITE register <br> used to reset the chip. Writing 0x01 to this <br> register will apply the reset to the digital <br> and this is propagated asynchronously to <br> the digital blocks. On writing a 0 to bit 0 of <br> register FE, the reset is de-asserted and <br> internal to the digital, we use the various <br> clocks to perform the de-assertion reset <br> synchronization for the appropriate clock <br> domains. |  |
| PAGE_NUMBER | FFh | $7: 0$ | R/W | Oh | On all pages register FF is a READ / <br> WRITE register used to change the page <br> number |

Table 32 Page A: PLL A Related Registers (Similar for Pages B, C, D)
The Registers from 10h to 2Fh are equivalent NVMCopy Registers for this Page.

| Reg Name | Register Number | $\begin{gathered} \text { Bit } \\ \text { Range } \end{gathered}$ | Access Type | Default Value | Description |
| :---: | :---: | :---: | :---: | :---: | :---: |
| STATUS_PLLA | 02h | 7 | R | Oh | SPARE |
|  |  | 6 | R | Oh | SPARE |
|  |  | 5 | R | Oh | SPARE |
|  |  | 4 | R | Oh | SPARE |
|  |  | 3 | R | Oh | SPARE |
|  |  | 2 | R | Oh | SPARE |
|  |  | 1 | R | Oh | SPARE |
|  |  | 0 | R | Oh | Dynamic status of Loss of lock for PLLA |
| NOTIFY_PLLA | 03h | 7 | R/W | 1h | SPARE |
|  |  | 6 | R/W | 1h | SPARE |
|  |  | 5 | R/W | 1h | SPARE |
|  |  | 4 | R/W | 1h | SPARE |
|  |  | 3 | R/W | 1h | SPARE |
|  |  | 2 | R/W | 1h | SPARE |
|  |  | 1 | R/W | 1h | SPARE |
|  |  | 0 | R/W | 1h | Sticky/Notify status Loss of lock for PLLA |
| MASKb_PLLA | 04h | 7 | R/W | 1h | SPARE |
|  |  | 6 | R/W | 1h | SPARE |
|  |  | 5 | R/W | 1h | SPARE |
|  |  | 4 | R/W | 1h | SPARE |
|  |  | 3 | R/W | 1h | SPARE |
|  |  | 2 | R/W | 1h | SPARE |
|  |  | 1 | R/W | 1h | SPARE |
|  |  | 0 | R/W | 1h | Mask bit for NOTIFY_PLLA (03h) If programmed as '0': Mask sticky/Notify bit generation for 03h[0] |
| Directives_GENERIC_PLLA | 05h | 7 | R/W | Oh | Reserved |
|  |  | 6 | R/W | Oh | Spare |
|  |  | 5 | R/W | 1h | DLPF co-efficient selection provided by the GUI |
|  |  | 4 | R/W | Oh | Force external clock in switch |
|  |  | 3 | R/W | Oh | Force the PLL in holdover mode |
|  |  | 2 | R/W | Oh | Large change for resetting entire DIVO system: Edge triggered |
|  |  | 0 | R/W | Oh | Spare |
| STATUS_1_PLLA | 06h | 7 | R | Oh | SPARE |
|  |  | 6 | R | Oh | SPARE |
|  |  | 5 | R | Oh | SPARE |
|  |  | 4 | R | Oh | SPARE |
|  |  | 3 | R | Oh | Dynamic status for cycle slip detection |
|  |  | 2 | R | Oh | Dynamic status for indicating holdover window is valid |
|  |  | 1 | R | Oh | Dynamic status for fast lock mode |
|  |  | 0 | R | Oh | Dynamic status for holdover |
| NOTIFY_1_PLLA | 07h | 7 | R/W | 1h | SPARE |
|  |  | 6 | R/W | 1h | SPARE |
|  |  | 5 | R/W | 1h | SPARE |


| Reg Name | Register Number | Bit Range | Access Type | Default Value | Description |
| :---: | :---: | :---: | :---: | :---: | :---: |
| NOTIFY_1_PLLA | 07h | 4 | R/W | 1h | SPARE |
|  |  | 3 | R/W | 1h | Sticky/Notify status for cycle slip detection |
|  |  | 2 | R/W | 1h | Sticky bit indicating holdover window is valid |
|  |  | 1 | R/W | 1h | Sticky/Notify status for fast lock mode |
|  |  | 0 | R/W | 1h | Sticky/Notify status for holdover |
| MASKb_1_PLLA | 08h | 7 | R/W | 1h | SPARE |
|  |  | 6 | R/W | 1h | SPARE |
|  |  | 5 | R/W | 1h | SPARE |
|  |  | 4 | R/W | 1h | SPARE |
|  |  | 3 | R/W | 1h | Mask bit for NOTIFY_1_PLLA (07h)If programmed as '0': Mask sticky/Notify bit generation for 07h[3] |
|  |  | 2 | R/W | 1h | Mask bit for NOTIFY_1_PLLA (07h)If programmed as '0': Mask sticky/Notify bit generation for 07h[2] |
|  |  | 1 | R/W | 1h | Mask bit for NOTIFY_1_PLLA FASTLOCK (07h)If programmed as ' 0 ': Mask sticky/Notify bit generation for 07h[1] |
|  |  | 0 | R/W | 1h | Mask bit for NOTIFY_1_PLLA (07h)If programmed as '0': Mask sticky/Notify bit generation for 07h[0] |
| PRG_Directives_PLLA | OFh | 7:3 | R/W | Oh | PRG_CMD Directives: <br> 5b1_1000: PROGRAM_EFUSE <br> 5b0_1100: READ_EFUSE <br> 5b0_0110: Copy NVM Copy to Settings <br> 5b1_1011: Proceed to Active |
|  |  | 2 | R/W | Oh | Spare |
|  |  | 1 | R/W | Oh | Escape to the PRG_CMD state from ACTIVE state |
| PPATH_PLLA | 10h | 7:5 | R/W | Oh |  |
|  |  | 4:0 | R/W | Oh |  |
| IPATH1_PLLA | 11h | 7:3 | R/W | Oh |  |
|  |  | 2:0 | R/W | Oh |  |
| IPATH2_PLLA | 12h | 7:5 | R/W | Oh |  |
|  |  | 4:0 | R/W | Oh |  |
| FASTLOCK_PPATH_PLLA | 13h | 7:5 | R/W | Oh | PF Settings from the |
|  |  | 4:0 | R/W | Oh |  |
| FASTLOCK_IPATH1_PLLA | 14h | 7:3 | R/W | Oh |  |
|  |  | 2:0 | R/W | Oh |  |
| FASTLOCK_IPATH2_PLLA | 15h | 7:5 | R/W | Oh |  |
|  |  | 4:0 | R/W | Oh |  |
| CYCLESLIP_MISC_CTRL_PLLA | 16h | 7 | R/W | Oh | CP gain configuration settings provided by GUI |
|  |  | 6 | R/W | Oh |  |
|  |  | 5 | R/W | Oh | PLL loop filer configuration provided by GUI |
|  |  | 4 | R/W | Oh | PLL loop filer configuration provided by GUI |
|  |  | 3 | R/W | Oh | ZDB related setting provided by GUI |
|  |  | 2:1 |  | Oh | Cycle Slip detector threshold settings computed by GUI |
|  |  | 0 | R/W | Oh | Enable for cycle slip detector for DLPF |
| DIVNINT_PLLA | 17h | 7 | R/W | Oh | Master Disable for PLLA in sync mode and all associated functions |


| Reg Name | Register Number |  | Access Type | Default Value | Description |
| :---: | :---: | :---: | :---: | :---: | :---: |
| DIVNINT_PLLA | 17h | 6:0 | R/W | Oh | DIVN Integer part computed by GUI |
| MISCXO_PLLA | 18h | 7 | R/W | Oh | Enable ADC Dither |
|  |  | 6 | R/W | Oh | Offset Enable in the PLL Charge Pump |
|  |  | 5 | R/W | Oh | Select ACTIVE clock manually from manual input select pins when in manual active select mode |
|  |  | 4 | R/W | Oh | DIVN Integer part computed by GUI |
|  |  | 3:1 | R/W | Oh | Charge Pump Binning for VCO KV |
|  |  | 0 | R/W | Oh | 0: Fractional Mode; 1: Integer Mode |
| DLPF_PHI_CORRECTION_PLLA | 19h | 7:6 | R/W | Oh | Phase averager configuration settings |
|  |  | 5:2 | R/W | Oh | Phase measurement filter BW setting (3 is spare) |
|  |  | 1 | R/W | Oh | Phase Correction Setting: Use default from GUI |
| DIVNFRAC2_PLLA | 1Ah | 7:0 | R/W | Oh | DIVN Fractional part computed by GUI |
| DIVNFRAC3_PLLA | 1Bh | 7:0 | R/W | Oh | DIVN Fractional part computed by GUI |
| DIVNFRAC4_PLLA | 1Ch | 7:0 | R/W | Oh | DIVN Fractional part computed by GUI |
| DIVN2_INT1_PLLA | 1Dh | 7:0 | R/W | Oh | DIVN2 Integer part computed by GUI |
| DIVN2_INT2_PLLA | 1Eh | 7:0 | R/W | Oh | DIVN2 Integer part computed by GUI |
| DIVN2_INT3_PLLA | 1Fh | 7 | R/W | Oh | Enable Revertive switching for input clock switching |
|  |  | 6 | R/W | Oh | Force manual selection of ACTIVE clock |
|  |  | 5 | R/W | Oh | Force Integer mode for the DIVN2 DSM |
|  |  | 4:0 | R/W | Oh | DIVN2 Integer part computed by GUI |
| DIVN2_FRN1_PLLA | 20h | 7:0 | R/W | Oh | DIVN2 Fractional part computed by GUI |
| DIVN2_FRN2_PLLA | 21h | 7:0 | R/W | Oh | DIVN2 Fractional part computed by GUI |
| DIVN2_FRN3_PLLA | 22h | 7:0 | R/W | Oh | DIVN2 Fractional part computed by GUI |
| DIVN2_FRN4_PLLA | 23h | 7:0 | R/W | Oh | DIVN2 Fractional part computed by GUI |
| DIVN2_FRD1_PLLA | 24h | 7:0 | R/W | Oh | DIVN2 Fractional part computed by GUI |
| DIVN2_FRD2_PLLA | 25h | 7:0 | R/W | Oh | DIVN2 Fractional part computed by GUI |
| DIVN2_FRD3_PLLA | 26h | 7:0 | R/W | Oh | DIVN2 Fractional part computed by GUI |
| DIVN2_FRD4_PLLA | 27h | 7:0 | R/W | Oh | DIVN2 Fractional part computed by GUI |
| OUTPUT_EN_PLLA | 28h | 7:6 | R/W | Oh | $\begin{aligned} & \text { PSEQ_SYNC_DELAY } \\ & 0: 500 \mathrm{us} \\ & 1: 8.17 \mathrm{~ms} \\ & 2: 131.07 \mathrm{~ms} \\ & 3: 2.09 \mathrm{sec} \end{aligned}$ |
|  |  | 5:0 | R/W | Oh | One Hot Output Enable for Outputs 5:0 |
| LL_REG1_PLLA | 29h | 7:6 | R/W | Oh | Programmable Phase Propagation Mode slope for the PLL <br> 00: Use the Bandwidth <br> 01: 10 usec/sec <br> 10: 40 usec/sec <br> 11: 160 usec/sec |
|  |  | 5 | R/W | Oh | Wait for Input Clock in power up in the PLL wake-up sequence |
|  |  | 4 | R/W | Oh | Use Fast Lock BW for exit from holdover to latch on fast to the new clock |
|  |  | 3:1 | R/W | Oh | Loss of Lock Delay = (2^(26(2*LLDELAYTIMER))/4M). Wait for this delay time before announcing LL deassertion |


| Reg Name | Register Number |  | Access Type | Default Value | Description |
| :---: | :---: | :---: | :---: | :---: | :---: |
| LL_REG1_PLLA | 29h | 0 | R/W | Oh | ```LL Clear Threshold \{LL_CLR_VALUE_PLLA[1], 0x2A[0]\} 2'b00: 0.2 PPM 2'b01: 0.4 PPM 2'b10:2 PPM 2'b11: 200 PPM``` |
| LL_REG2_PLLA | 2Ah | 7:5 | R/W | Oh | $\begin{aligned} & \text { LL_SET_VALUE_PLLA[7:4] } \\ & 0: 0.2 \text { PPM } \\ & 1: 0.4 \text { PPM } \\ & 2: 2 \mathrm{PPM} \\ & 3: 4 \text { PPM } \\ & 4: 20 \mathrm{PPM} \\ & 5: 40 \mathrm{PPM} \\ & 6: 200 \mathrm{PPM} \\ & 7: 400 \mathrm{PPM} \\ & 8: 2000 \mathrm{PPM} \\ & 9: 4000 \mathrm{PPM} \\ & 10: 0.2 \mathrm{PPM} \\ & 11: 0.2 \mathrm{PPM} \\ & 12: 0.2 \mathrm{PPM} \\ & 13: 0.2 \mathrm{PPM} \\ & 14: 0.2 \mathrm{PPM} \\ & 15: 0.2 \mathrm{PPM} \end{aligned}$ |
|  |  | 4 | R/W | Oh |  |
|  |  | 3:1 | R/W | Oh | ```PLL_PSEQ_DELAY 0: 500us 1: 2 ms 2: 8.19 ms 3: 32.76 ms 4: 131.07 ms 5: 524.28 ms 6: 2.09sec 7: 8.38sec``` |
|  |  | 0 | R/W | Oh | LL Clear Threshold $\{0 \times 29[0]$, LL_CLR_VALUE_PLLA[0]\} 2'b00: 0.2 PPM 2'b01: 0.4 PPM <br> 2'b10: 2 PPM <br> 2'b11: 200 PPM |
| HOLDOVER1_PLLA | 2Bh | 7:5 | R/W | Oh | Holdover Tdelay settings for the PLL computed by the GUI |
|  |  | 4 | R/W | Oh | Enable Zero Delay Buffer mode with feedback clock routed from the PCB on in3 |
|  |  | 3:1 | R/W | Oh | Holdover Average settings for the PLL computed by the GUI |
|  |  | 0 | R/W | Oh | Enable output clock sync on an independent input coming from IN3 |
| HOLDOVER2_PLLA | 2 C | 7 | R/W | Oh | Cycle Slip Detector related default |
|  |  | 6 | R/W | Oh | 1: Enable Phase Propagation during input clock switch <br> 0 : Enable Phase Build Out Mode where the phase difference between input clocks is absorbed by the PLL |
|  |  | 5 | R/W | 1h | Dither configuration for DIVN2 DSM |
|  |  | 4 | R/W | Oh | Enable revert to spare input clock during clock switching |
|  |  | 3:1 | R/W | Oh | DLPF related constant from the GUI |
|  |  | 0 | R/W | Oh | Internal voltage programming: Use default from GUI Profile |
| DECIMATION_RATIO_PLLA | 2Dh | 7 | R/W | Oh | Enable fast lock mode based on loss of lock status |


| Reg Name | Register Number | $\begin{gathered} \text { Bit } \\ \text { Range } \end{gathered}$ | Access Type | Default Value | Description |
| :---: | :---: | :---: | :---: | :---: | :---: |
| DECIMATION_RATIO_PLLA | 2Dh | 6:3 | R/W | Oh | Internal Rate Change factors in DLPF computed by GUI |
|  |  | 2:0 | R/W | Oh | Internal Rate Change factors in DLPF computed by GUI |
| ONEBYR2_PLLA | 2Eh | 7:3 | R/W | Oh | Internal Rate Change factors in DLPF computed by GUI |
|  |  | 2:0 | R/W | Oh | Internal Rate Change factors in DLPF computed by GUI |
| LOCKPATTERN_PLLA | 2Fh | 7:6 | R/W | Oh | PATTERN :\{'01'/10'\} =] Efuse Locked, \{'00'/'11'\} =] Efuse NOT Locked |
|  |  | 5:4 | R/W | Oh | Program the frequency ramp slope from the following- <br> 00: $0.2 \mathrm{ppm} / \mathrm{s}, 01: 2 \mathrm{ppm} / \mathrm{s}, 10: 20 \mathrm{ppm} / \mathrm{s}$, <br> 11: $200 \mathrm{ppm} / \mathrm{s}$ |
|  |  | 3 | R/W | Oh | Enable the frequency ramp feature |
|  |  | 2:0 | R/W | Oh | Internal Frequency Divider : Computed by the GUI |
| DCO_FRAC1_PLLA | 31h | 7:0 | R/W | Oh | DCO fractional control code |
| DCO_FRAC2_PLLA | 32h | 7:0 | R/W | Oh | DCO fractional control code |
| DCO_FRAC3_PLLA | 33h | 7:0 | R/W | Oh | DCO fractional control code |
| DCO_FRAC4_PLLA | 34h | 7:0 | R/W | Oh | DCO fractional control code |
| DCO_FUNCTION_PLLA | 35h | 7:4 | R/W | Oh | Reserved |
|  |  | 3 | R/W | Oh | DCO Increment in Frequency from registers |
|  |  | 2 | R/W | Oh | DCO Decrement in Frequency from registers |
|  |  | 1 | R/W | Oh | Enable DCO free run mode |
|  |  | 0 | R/W | Oh | DCO Mask for this PLL |
| DCO_BUMP2_PLLA | 36h | 7:0 | R/W | Oh | DCO integer control code |
| DCO_BUMP3_PLLA | 37h | 7:6 | R/W | Oh | DCO integer control code |
|  |  | 5 | R/W | Oh | Enable DCO sync mode |
|  |  | 4 | R/W | Oh | SPARE |
|  |  | 3:0 | R/W | Oh | SPARE |
| PAGE_NUMBER | FFh | 7:0 | R/W | Oh | On all pages register FF is a READ / WRITE register used to change the page number |

## 27 Ordering Information

Table 33 Ordering Information for AU5327

| Ordering Part Number (OPN) | Marking | No of Input/ Output Clocks | Output Clock Frequency Range (MHz) | Supported <br> Frequency Synthesis modes | VDDIN, VDD, VDDIO | Package | Temp. <br> Range |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| AU5327BC1-QMR ${ }^{1,2}$ | AU5327BC ${ }^{4}$ | 4/8 | $8 \mathrm{KHz-2.1} \mathrm{GHz}$ | Integer and Fractional | 3.3,1.8/2.5/3.3,3.3 | $\begin{aligned} & \text { 64-QFN } \\ & 9 \times 9 \mathrm{~mm} \end{aligned}$ | -40 to $85^{\circ} \mathrm{C}$ |
| AU5327BC1-QMT ${ }^{1,2}$ | AU5327BC ${ }^{4}$ | 4/8 | $8 \mathrm{KHz-2.1} \mathrm{GHz}$ | Integer and Fractional | 3.3,1.8/2.5/3.3,3.3 | $\begin{aligned} & \text { 64-QFN } \\ & 9 \times 9 \mathrm{~mm} \end{aligned}$ | -40 to $85^{\circ} \mathrm{C}$ |
| AU5327BC2-QMR ${ }^{1,2}$ | AU5327BC ${ }^{4}$ | 4/8 | $8 \mathrm{KHz-2.1} \mathrm{GHz}$ | Integer and Fractional | 3.3,1.8/2.5/3.3,3.3 | $\begin{aligned} & \text { 64-QFN } \\ & 9 \times 9 \mathrm{~mm} \end{aligned}$ | -40 to $85^{\circ} \mathrm{C}$ |
| AU5327BC2-QMT ${ }^{1,2}$ | AU5327BC ${ }^{4}$ | 4/8 | $8 \mathrm{KHz-2.1} \mathrm{GHz}$ | Integer and Fractional | 3.3,1.8/2.5/3.3,3.3 | $\begin{aligned} & \text { 64-QFN } \\ & 9 \times 9 \mathrm{~mm} \end{aligned}$ | -40 to $85^{\circ} \mathrm{C}$ |
| AU53x7C1-EVB |  | - | - | - | - | Evaluation Board | - |

Notes:

1. Add an R at the end of the OPN to denote tape and reel ordering option. Add a $T$ at the end of the OPN to denote a tray option.
2. Custom and factory preprogrammed devices are available. Ordering part numbers are assigned by Aurasemir. Please contact local sales to request the unique part number. Custom part number format is "AU53xxACxQM" where " $x$ " is a unique numerical sequence representing the preprogrammed configuration.
3. BC1 and BC2 version has superior PSRR and spur performance compared to AC1 and is recommended for new designs.
4. Please refer to $5^{\text {th }}$ row of the Die Marking to differentiate between AU5327BC1 and AU5327BC2. AU5327BC1 starts with C 1 in the $5^{\text {th }}$ row and AU5327BC2 starts with C 2 in the $5^{\text {th }}$ row.


## 28 Revision History

Table 34 Revision History

| Revision | Date | Description | Author |
| :---: | :---: | :---: | :---: |
| 0.1 | 01 Sept 2018 | AU53x7 and AU53x6 Data Sheet First Advance Draft | Aurasemi |
| 0.2 | 17 Sept 2018 | Added Ordering information. Corrected a typo in Table 10 for range of crystal frequency. | Aurasemi |
| 0.3 | 29 Sept 2018 | Added Register Map Details | Aurasemi |
| 0.4 | 24 Oct 2018 | Added Zero Delay Mode Added I2C address details | Aurasemi |
| 0.5 | 27 Nov 2018 | Updated XO Termination diagrams <br> Updated Clock Monitoring and Alarm Register Information Updated HCSL termination diagrams <br> Added ZDB delay specification <br> Added PN plots for various scenarios and specification for typical PN | Aurasemi |
| 0.6 | 04 Jan 2019 | Added information on single ended AC Coupled input pathway <br> Added information on the SPI Mode SDO <br> Added maximum current consumption numbers in addition to the typical numbers <br> Updated the ordering code information <br> Additional details of XO registers <br> Cleaned I2C addressing typo in section 17 | Aurasemi |
| 0.7 | 20 May 2019 | Updated LVPECL2 AC Coupled diagram with more detail on far end termination information <br> $\mathrm{VOH} / \mathrm{VOL}$ spec for LVCMOS output for 100 uA is removed since this is already checked with the 4 mA specification test in ATE <br> VOL specification maximum spec for LVPECL is increased by 50 mV and CML by 20 mV based on ATE data <br> Description of register 0xFE is added <br> Added pictorial description of Vpp and Vp conventions for swing <br> Default Ordering information code changed to QMR to indicate Tape and Reel <br> Added clarification regarding I2C1_SPIO pin pull down and multiplexed I2C addressing in the pin list section. | Aurasemi |
| 0.8 | 18 July 2019 | Changed typical jitter performance to 156.25 M use case instead of the 622.08 M case. <br> Functional overview diagram is improved for resolution Relaxed LVPECL static VOL by 50 mV to provide ATE guard band <br> Added Specifications for the LVCMOS In Phase Outputs also Added Power Consumption Upper Limit specifications Increased LVPECL2 and HCSL IDDO specification by 2.4 mA to add ATE guard band <br> Added HCSL VP AC specification instead of VOH_DIFF and VOL_DIFF to make it consistent with ATE <br> Updated the SPI timing diagram to make it more descriptive Updated the INTRB pin related information in the Notify and Clear tabs including the recommendation for the clearing of notifies at wake up | Aurasemi |


| Revision | Date | Description | Author |
| :---: | :---: | :---: | :---: |
| 1.0 | $10^{\text {th }}$ March 2020 | Updated Note 4 after Table 5 regarding Single Ended AC Coupled Input Swing Requirement <br> Unused Pin Information updated in Table1 for FLEXIO's and Input/Output Pins. <br> Device Soft Reset additional Information added in Note 5 after Table 1 <br> XTALcl spec made as Typical in the Table 10 LFF Section Table 32 updated with additional Register Information on the Output Page <br> Note 6 after Table 1 changed to define the SDO and CSB default states. <br> Updated Note 1 after Table 1 to define the VDDIO selection as VDD/VDDIN <br> Added Note 8 after Table 4 for efuse programming voltage information for AU532x parts <br> Updated Table 32 register 29h description <br> RevC1 is updated to RevC2 in the register information Miscellaneous Datasheet Format changes | Aurasemi |
| 1.1 | $1^{\text {st }}$ April 2020 | Additional Information added to Note 1 after Table 1 regarding the VDDIO selection for AU531x and AU532x Parts <br> Register Information added for register 23h in the Register Map Details Table 29 | Aurasemi |
| 1.2 | $18^{\text {th }}$ Sept 2020 | Table 15 updated with Max Swing Spec for Boosted LVDS and LVPECL Max Swing <br> Table 15 Updated the Lower Limit LVDS Swing from 247 mV to 300 mV <br> Table 19 updated with correct SPI "Output Valid" Timing Specs <br> General Description updated with BC1 and BC2 <br> Nomenclature <br> Table 34 updated with BC1 and BC2 parts Ordering Information with additional Note (4) <br> Additional Note (3) added after Table 2 for the XO Input Absolute Min and Max Voltage <br> Updated Table 1 with RSTB pin default connection instruction" Changed the default pull up from VDD to VDDIO" | Aurasemi |
| 1.3 | $30^{\text {th }}$ Dec 2020 | Logo Changed <br> AU532x parts VDD support changed to $1.8 \mathrm{~V} / 2.5 \mathrm{~V} / 3.3 \mathrm{~V}$ from 1.8V. | Aurasemi |
| 1.4 | $26^{\text {th }}$ May 2021 | Added 'Meets G. 8262 EEC Option 1,2(Sync E)' to Features | Aurasemi |
| 1.5 | $5^{\text {th }}$ July 2021 | 1.Table 33 and Table 34 changed to update the Ordering Part Number and Marking information. <br> 2. Table 18 I2C Bus Timing updated with standard and fast mode specifications. | Aurasemi |


| Revision | Date | Description | Author |
| :---: | :---: | :---: | :---: |
| 1.6 | $25^{\text {th }}$ April 2022 | Au5317 Information is removed from the datasheet as the product has reached EOL. <br> 44 -QFN variants Au53x6 information is removed from the datasheet. <br> VDDIO Information for $B C 1$ and $B C 2$ variants updated. <br> SPI Timing Read Figure 19 updated. <br> MSL Information included in Table 2. <br> Additional recommendation Note for improving XO slew rate added below Figure 36. <br> thd:DAT min spec changed to 10 ns in Table 18. <br> Table 33 Ordering Information table updated with AU5327 only information. | Aurasemi |

## IMPORTANT NOTICE AND DISCLAIMER

AURASEMI 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 skilled in the art designing with Aurasemi 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. Aurasemi grants you permission to use these resources only for development of an application that uses Aurasemi products. Other reproduction or use of these resources is strictly prohibited. No license is granted to any other Aurasemi intellectual property or to any third-party intellectual property. Aurasemi disclaims responsibility for, and you will fully indemnify Aurasemi and its representatives against, any claims, damages, costs, losses, or liabilities arising out of your use of these resources. Aurasemi products are provided only subject to Aurasemi Terms and Conditions of Sale or other applicable terms agreed to in writing. No use of any Aurasemi resources expands or otherwise alters any applicable warranties or warranty disclaimers for these products.

## Contact Information

For more information visit www.aurasemi.com
For sales related information please send an email to sales@aurasemi.com

## Trademarks

Aurasemi and Aurasemi Logo are trademarks of Ningbo Aura Semiconductor Private Limited. All referenced brands, product names, service names and trademarks are the property of their respective owners.

