Data Sheet | April 1999 | File Number 4704

## SMBus/I$^2$C ACPI Dual Device Bay Controller

The CDP1020 is an ACPI compliant Device Bay Controller (DBC) that can control two device bays. The controller interfaces to the host system through the industry standard I$^2$C or System Management Bus (SMBus) and is fully compliant with Device Bay Specification 0.90. The CDP1020 is designed to be compatible with the integrated SMBus host controller of the PiiX4/PiiX6 in Intel Architecture platforms.

The CDP1020 is designed to be placed on the host motherboard, on a riser, or adjacent to the Device Bay connectors. The required clock source is generated from an internal oscillator on the CLK pin, with an external RC to set the frequency. This lowers the system cost and allows the CDP1020 to remain active during S3-S5 system states where all clock generators have been stopped.

One of the key features of this device is the on-chip level shifters that provide slew rate controlled, direct gate drive for external N-Channel MOSFETs (Intersil HUF76113DK8 recommended) to switch the device bay $V_{ID}$ supplies. Switching an N-Channel device as opposed to a P-Channel reduces both device cost and device count, resulting in an overall lower system cost.

Configuration data for the CDP1020, including subsystem vendor ID, subsystem revision, bay size and device bay capabilities are designed to be written into the CDP1020 by the system BIOS at power up. The registers for this data are write-once-only and thus become read-only after the initial BIOS write.

The address selection pins (AD1 and AD0) allow the CDP1020 to occupy any one of four I$^2$C/SMBus addresses. This enables up to four CDP1020 devices to coexist in a system.

The CDP1020 implements high current outputs for direct drive (with a limiting resistor) of the optional bay status LEDs. These indicators are two color (green/amber) common anode or anti-parallel LEDs that indicate the device bay status per the Device Bay Specification 0.90.

## Ordering Information

| PART NUMBER | TEMP. RANGE ($^o$C) | PACKAGE | PKG. NO. |
|---|---|---|---|
| CDP1020 | 0 to 85 | 28 Ld SOIC | M28.3 |

For More Information Contact:

**Mike Coletta (714) 433-0600**

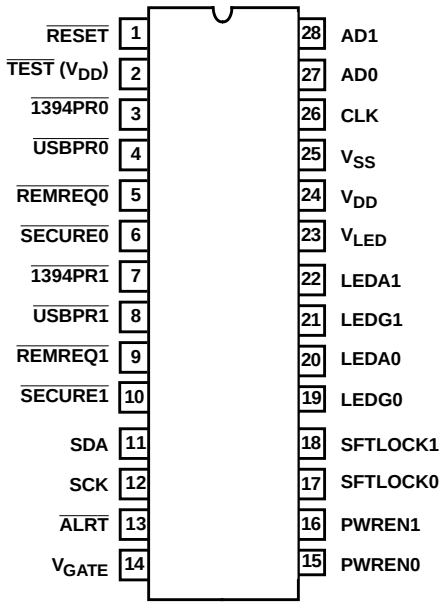
**Terry Pierce (407) 729-5835**

## Features

- Fully Compliant with Device Bay Specification 0.90 and ACPI Specification 1.0
- Industry Standard SMBus/I$^2$C Interface
- Controls for Two Device Bays
- Onboard Level Shifting for Direct Drive of N-Channel MOSFET $V_{ID}$ Switches
- Integrated Pull-up Resistors on 1394PRx, USBPRx, SECUREx, and REMREQx Inputs
- RC Type Oscillator - Low Cost and Low Power Consumption
- Operational Voltage from 3.3 to 5.5V
- “5V Tolerant” Inputs at all Operating Voltages
- Write-Once BIOS/External Configuration
- Removal Request Input for Each Bay
- Security Lock Input for Each Bay
- High Current Device Bay LED Indicator Drivers With Separate High-Side Power Input
- Configurable Level/Pulse Bay Solenoid Drivers
- Programmable Insertion Time Out Delay
- HCMOS Technology; 28 Lead Plastic SOIC

## Pinout

**CDP1020 (SOIC)**
TOP VIEW



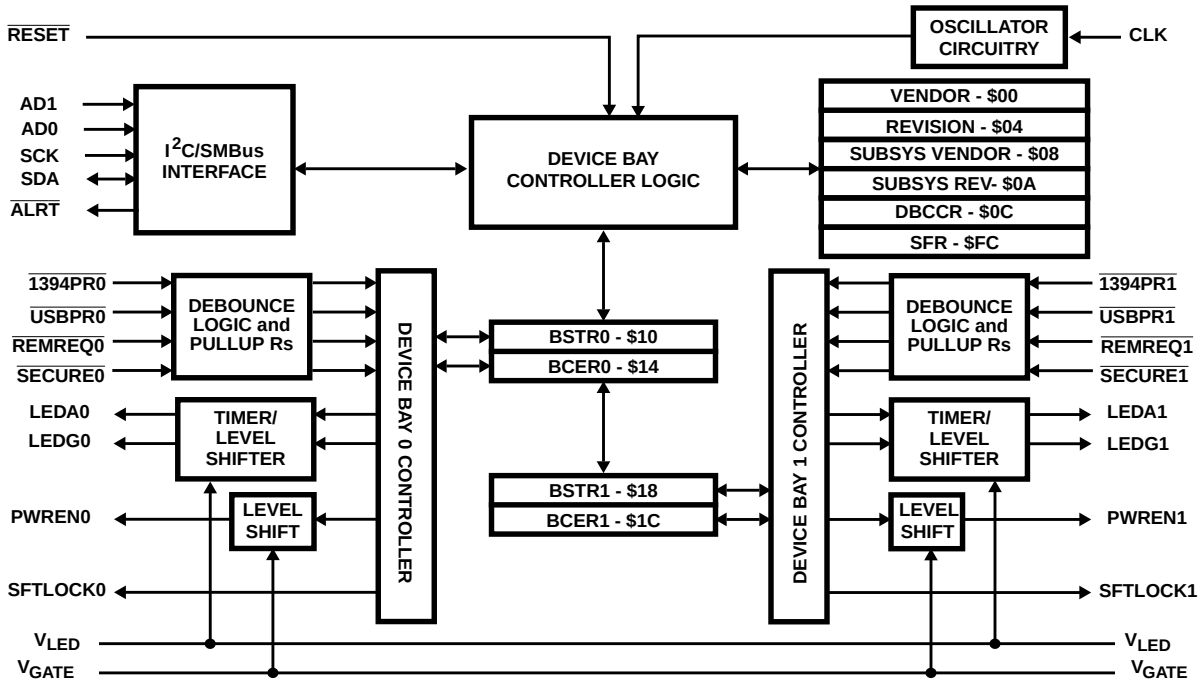



CAUTION: These devices are sensitive to electrostatic discharge; follow proper IC Handling Procedures.
http://www.intersil.com or 407-727-9207 | Copyright © Intersil Corporation 1999

## Pin Descriptions

| PIN NUMBER | PIN NAME | IN/OUT | PIN DESCRIPTION |
|---|---|---|---|
| 1 | $\overline{\text{RESET}}$ | IN | Device Bay Controller Master Reset Schmitt Input |
| 2 | $\overline{\text{TEST}}$ | - | Test pin used by manufacturer only. Must be externally connected to $V_{DD}$ |
| 3 | $\overline{\text{1394PR0}}$ | IN | Bay 0 1394 Presence Input with Active Pull-up |
| 4 | $\overline{\text{USBPR0}}$ | IN | Bay 0 USB Presence Input with Active Pull-up |
| 5 | $\overline{\text{REMREQ0}}$ | IN | Bay 0 Remove Request Input with Active Pull-up |
| 6 | $\overline{\text{SECURE0}}$ | IN | Bay 0 Security Input with Active Pull-up |
| 7 | $\overline{\text{1394PR1}}$ | IN | Bay 1 1394 Presence Input with Active Pull-up |
| 8 | $\overline{\text{USBPR1}}$ | IN | Bay 1 USB Presence Input with Active Pull-up |
| 9 | $\overline{\text{REMREQ1}}$ | IN | Bay 1 Remove Request Input with Active Pull-up |
| 10 | $\overline{\text{SECURE1}}$ | IN | Bay 1 Security Input with Active Pull-up |
| 11 | SDA | IN/OUT | SMBus/$I^2C$ Data Schmitt Input/Open-Drain Output |
| 12 | SCK | IN/OUT | SMBus/$I^2C$ Clock Schmitt Input/Open-Drain Output |
| 13 | $\overline{\text{ALRT}}$ | OUT | SMBus $\overline{\text{Alert}}$ Open-Drain Output |
| 14 | $V_{GATE}$ | - | Power Supply Input for PWREN0/PWREN1 Drivers |
| 15 | PWREN0 | OUT | Bay 0 Power Enable 12V NMOS Gate Drive Output |
| 16 | PWREN1 | OUT | Bay 1 Power Enable 12V NMOS Gate Drive Output |
| 17 | SFTLOCK0 | OUT | Bay 0 Software Controlled Lock Mechanism Driver |
| 18 | SFTLOCK1 | OUT | Bay 1 Software Controlled Lock Mechanism Driver |
| 19 | LEDG0 | OUT | Bay 0 Status Indicator (Green LED) Driver |
| 20 | LEDA0 | OUT | Bay 0 Status Indicator (Amber LED) Driver |
| 21 | LEDG1 | OUT | Bay 1 Status Indicator (Green LED) Driver |
| 22 | LEDA1 | OUT | Bay 1 Status Indicator (Amber LED) Driver |
| 23 | $V_{LED}$ | - | Power Supply Input for LED Driver (LEDAx, LEDGx) |
| 24 | $V_{DD}$ | - | Power Supply Input (Power) |
| 25 | $V_{SS}$ | - | Power Supply Return (Ground or GND) |
| 26 | CLK | IN | External Clock Schmitt Input (for RC oscillator) |
| 27 | AD0 | IN | SMBus/$I^2C$ Address Configuration Bit 0 |
| 28 | AD1 | IN | SMBus/$I^2C$ Address Configuration Bit 1 |

## Block Diagram



## Absolute Maximum Ratings

Supply Voltage, $V_{DD}$ . . . . . -0.5V to +6V
Supply Voltage, $V_{GATE}$ . . . . . $V_{DD}$ to 13V
Supply Voltage, $V_{LED}$ . . . . . -0.5V to +6V
Input Voltage, $V_{IN}$ . . . . . $V_{SS}$ - 0.3V to $V_{DD}$ + 0.3V
Test Mode, $V_{IN}$ . . . . . $V_{SS}$ - 0.3V to 2 x $V_{DD}$ + 0.3V
Current Drain Per Pin Excluding $V_{DD}$ and $V_{SS}$, I . . . . . 40mA

## Operating Conditions

Voltage Range . . . . . +3.0V to +5.0V
Temperature Range . . . . . $0^oC$ to $85^oC$
Input High Voltage . . . . . (0.8 x $V_{DD}$) to $V_{DD}$

## Thermal Information

| Thermal Resistance (Typical, Note 1) | $\theta_{JA}$ ($^oC/W$) |
|---|---|
| 28 Ld SOIC | 60 |

Maximum Junction Temperature . . . . . $150^oC$
Maximum Storage Temperature Range ($T_{STG}$) . . . . $-65^oC$ to $150^oC$
Maximum Lead Temperature (Soldering 10s) . . . . . $300^oC$
(SOIC - Lead Tips Only)

*CAUTION: Stresses above those listed in "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress only rating and operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied.*

NOTE:

1. $\theta_{JA}$ is measured with the component mounted on an evaluation PC board in free air.

## DC Electrical Specifications, 5.0V

$V_{DD}$ = 5.0V ±10%, $T_A$ = $0^oC$ to $85^oC$

| PARAMETER | SYMBOL | CONDITIONS | MIN | TYP | MAX | UNITS |
|---|---|---|---|---|---|---|
| Output Voltage | | -10µA < $I_{LOAD}$ < 10µA | | | | |
| All Outputs | $V_{OL}$ | | - | - | 0.1 | V |
| SFTLOCK0, SFTLOCK1 | $V_{OH}$ | | $V_{DD}$ - 0.1 | - | - | V |
| Output High Voltage | | | | | | |
| LEDA0, LEDG0, LEDA1, LEDG1 | $V_{OH}$ | $I_{LOAD}$ = -16.0mA | $V_{LED}$ - 1.0 | - | - | V |
| SFTLOCK0, SFTLOCK1 | $V_{OH}$ | $I_{LOAD}$ = -0.7mA | $V_{DD}$ - 0.3 | - | - | V |
| Output Low Voltage | | | | | | |
| SCK, SDA, $\overline{ALRT}$, SFTLOCK0, SFTLOCK1, LEDG0, LEDA0, LEDG1, LEDA1 | $V_{OL}$ | $I_{LOAD}$ = 1.6mA | - | - | 0.4 | V |
| PWREN0, PWREN1 | $V_{OL}$ | $I_{LOAD}$ = 60µA | - | - | 0.4 | V |
| Gate Output High Voltage | | | | | | |
| PWREN0, PWREN1 | $V_{GOH}$ | $I_{LOAD}$ < 10µA | $V_{GATE}$ -0.5V | $V_{GATE}$ -0.3V | $V_{GATE}$ | V |
| Gate Output Current Source | | | | | | |
| PWREN0, PWREN1 | $I_{GATE}$ | | -15 | -35 | -50 | µA |
| Input High Voltage | | | | | | |
| $\overline{1394PR0}$, $\overline{USBPR0}$, $\overline{REMREQ0}$, $\overline{SECURE0}$, $\overline{1394PR1}$, $\overline{USBPR1}$, $\overline{REMREQ1}$, $\overline{SECURE1}$ | $V_{IH}$ | | 0.7•$V_{DD}$ | - | $V_{DD}$ | V |
| SCK, SDA | $V_{IH}$ | | 0.7•$V_{DD}$ | - | $V_{DD}$ | V |
| $\overline{RESET}$, CLK | $V_{IH}$ | | 0.7•$V_{DD}$ | - | $V_{DD}$ | V |
| Input Low Voltage | | | | | | |
| $\overline{1394PR0}$, $\overline{USBPR0}$, $\overline{REMREQ0}$, $\overline{SECURE0}$, $\overline{1394PR1}$, $\overline{USBPR1}$, $\overline{REMREQ1}$, $\overline{SECURE1}$, $\overline{RESET}$, CLK | $V_{IL}$ | | $V_{SS}$ | - | 0.2•$V_{DD}$ | V |
| SCK, SDA | $V_{IL}$ | | $V_{SS}$ | - | 0.2•$V_{DD}$ | V |
| Input/Output Leakage Current: | | | | | | |
| $\overline{RESET}$, CLK, AD0, AD1, SCK, SDA, $\overline{ALRT}$ | $I_{IOL}$ | | - | - | ±10 | µA |
| Input Pullup Current | | | | | | |
| $\overline{1394PR0}$, $\overline{USBPR0}$, $\overline{REMREQ0}$, $\overline{SECURE0}$, $\overline{1394PR1}$, $\overline{USBPR1}$, $\overline{REMREQ1}$, $\overline{SECURE1}$ | $I_{IN}$ | | 50 | 200 | 400 | µA |
| Input Hysteresis Voltage | | | | | | |
| SCK, SDA | $V_{HYS}$ | | 0.02 | 0.10 | 0.40 | V |
| CLK | $V_{HYS}$ | | 0.6 | 1.0 | 1.3 | V |
| $\overline{RESET}$ | $V_{HYS}$ | | 0.8 | 1.1 | 1.4 | V |

## DC Electrical Specifications, 5.0V $V_{DD}$ = 5.0V ±10%, $T_A$ = 0°C to 85°C (Continued)

| PARAMETER | SYMBOL | CONDITIONS | MIN | TYP | MAX | UNITS |
|---|---|---|---|---|---|---|
| Capacitance | $C_{OUT}$ | | - | - | 12 | pF |
| | $C_{IN}$ | | - | - | 8 | pF |
| Supply Current (RUN) | $I_{DD}$ | $f_{CLK}$ = 4.0MHz External | - | 1.3 | 5.0 | mA |

## DC Electrical Specifications, 3.3V $V_{DD}$ = 3.3V ±10%, $T_A$ = 0°C to 85°C

| PARAMETER | SYMBOL | CONDITIONS | MIN | TYP | MAX | UNITS |
|---|---|---|---|---|---|---|
| Output Voltage | | -10µA < $I_{LOAD}$ < 10µA | | | | |
| All Outputs | $V_{OL}$ | | - | - | 0.1 | V |
| SFTLOCK0, SFTLOCK1 | $V_{OH}$ | | $V_{DD}$ - 0.1 | - | - | V |
| Output High Voltage | | | | | | |
| LEDA0, LEDG0, LEDA1, LEDG1 | $V_{OH}$ | $I_{LOAD}$ = -6.0mA | $V_{LED}$ - 1.0 | - | - | V |
| SFTLOCK0, SFTLOCK1 | $V_{OH}$ | $I_{LOAD}$ = -0.4mA | $V_{DD}$ - 0.3 | - | - | V |
| Output Low Voltage | | | | | | |
| SCK, SDA, $\overline{ALRT}$, SFTLOCK0, SFTLOCK1, LEDG0, LEDA0, LEDG1, LEDA1 | $V_{OL}$ | $I_{LOAD}$ = 1.6mA | - | - | 0.4 | V |
| PWREN0, PWREN1 | $V_{OL}$ | $I_{LOAD}$ = 50µA | - | - | 0.4 | V |
| Gate Output High Voltage | | | | | | |
| PWREN0, PWREN1 | $V_{GOH}$ | $I_{LOAD}$ < 10µA | $V_{GATE}$ -0.5V | $V_{GATE}$ -0.3V | $V_{GATE}$ | V |
| Gate Output Current Source | | | | | | |
| PWREN0, PWREN1 | $I_{GATE}$ | | -10 | -20 | -30 | µA |
| Input High Voltage | | | | | | |
| $\overline{1394PR0}$, $\overline{USBPR0}$, $\overline{REMREQ0}$, $\overline{SECURE0}$, $\overline{1394PR1}$, $\overline{USBPR1}$, $\overline{REMREQ1}$, $\overline{SECURE1}$ | $V_{IH}$ | | 0.7•$V_{DD}$ | - | $V_{DD}$ | V |
| SCK, SDA | $V_{IH}$ | | 0.7•$V_{DD}$ | - | $V_{DD}$ | V |
| $\overline{RESET}$, CLK | $V_{IH}$ | | 0.7•$V_{DD}$ | - | $V_{DD}$ | V |
| Input Low Voltage | | | | | | |
| $\overline{1394PR0}$, $\overline{USBPR0}$, $\overline{REMREQ0}$, $\overline{SECURE0}$, $\overline{1394PR1}$, $\overline{USBPR1}$, $\overline{REMREQ1}$, $\overline{SECURE1}$, $\overline{RESET}$, CLK | $V_{IL}$ | | $V_{SS}$ | - | 0.2•$V_{DD}$ | V |
| SCK, SDA | $V_{IL}$ | | $V_{SS}$ | - | 0.2•$V_{DD}$ | V |
| Input/Output Leakage Current: | | | | | | |
| $\overline{RESET}$, CLK, AD0, AD1, SCK, SDA, $\overline{ALRT}$ | $I_{IOL}$ | | - | - | ±10 | µA |
| Input Pullup Current | | | | | | |
| $\overline{1394PR0}$, $\overline{USBPR0}$, $\overline{REMREQ0}$, $\overline{SECURE0}$, $\overline{1394PR1}$, $\overline{USBPR1}$, $\overline{REMREQ1}$, $\overline{SECURE1}$ | $I_{IN}$ | | 20 | 80 | 160 | µA |
| Input Hysteresis Voltage | | | | | | |
| SCK, SDA | $V_{HYS}$ | | 0.05 | 0.15 | 0.45 | V |
| CLK | $V_{HYS}$ | | 0.4 | 0.8 | 1.1 | V |
| $\overline{RESET}$ | $V_{HYS}$ | | 0.3 | 0.5 | 0.8 | V |
| Capacitance | $C_{OUT}$ | | - | - | 12 | pF |
| | $C_{IN}$ | | - | - | 8 | pF |
| Supply Current (RUN) | $I_{DD}$ | | - | 0.9 | 4.5 | mA |

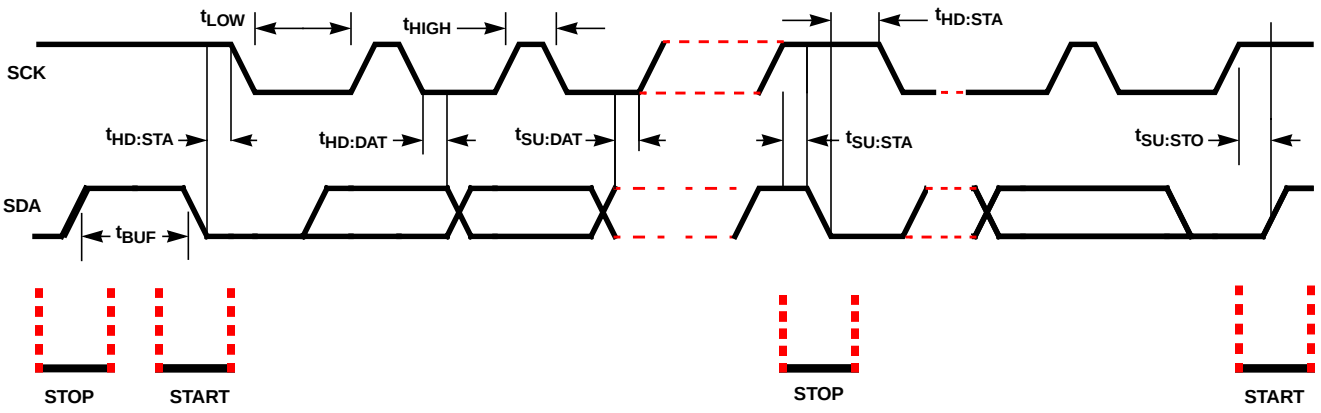



FIGURE 1. CONTROL TIMING

## Control Timing

$V_{DD}$ = 3.3V ±10%, $T_A$ = 0°C to 85°C

| PARAMETER | SYMBOL | MIN | MAX | UNITS |
|---|---|---|---|---|
| Frequency Of Operation (4.0MHz nominal) (CLK Pin) | $f_{CLK}$ | 2.0 | 4.5 | MHz |
| Suspend Recovery Start-up Time | $t_{RSUS}$ | 0.9 | 1 | ms |
| $\overline{\text{RESET}}$ Pulse Width ($\overline{\text{RESET}}$ Pin) | $t_{RL}$ | 6 | - | $t_{OSC}$ |
| Input Debounce Time ($\overline{1394PRx}$, $\overline{USBPRx}$, $\overline{REMREQx}$, $\overline{SECUREx}$ Pins) | $t_{DB}$ | 50 | - | ms |
| **SMBus** SCK and SDA Pins | | | | |
| SCK Frequency | $f_{SMB}$ | 10 | 100 | kHz |
| SMBus Free Time | $t_{BUF}$ | 4.7 | - | µs |
| Hold Time After (Repeated) Start Condition | $t_{HD:STA}$ | 4.0 | - | µs |
| Repeated Start Condition Setup Time | $t_{SU:STA}$ | 4.7 | - | µs |
| Stop Condition Setup Time | $t_{SU:STO}$ | 4.0 | - | µs |
| Data Hold Time | $t_{HD:DAT}$ | 300 | - | ns |
| Data Setup Time | $t_{SU:DAT}$ | 250 | - | ns |
| SCK Time-out Period | $t_{TIMEOUT}$ | 25 | 35 | ms |
| SCK Low Period | $t_{LOW}$ | 4.7 | - | µs |
| SCK High Period | $t_{HIGH}$ | 4.0 | 50 | µs |
| Slave SCK Extend Period (cumulative) | $t_{LOW:SEXT}$ | - | 25 | ms |
| Master SCK Extend Period (cumulative) | $t_{LOW:MEXT}$ | - | 10 | ms |
| SCK/SMBDAT Fall Time | $t_F$ | - | 300 | ns |
| SCK/SMBDAT Rise Time | $t_R$ | - | 1000 | ns |

## Notational Conventions

The following conventions are used throughout this document:

- Hexadecimal numbers are denoted with a “$” symbol preceding the number.
- Binary numbers are represented with a “%” symbol proceeding the number, or a “b” following.
- Because of the large mix of active-low and active-high signals used in connection with the CDP1020, the terms “asserted” and “de-asserted” will be used exclusively. An active low signal is asserted when it is at a logic 0 and de-asserted when it is at a logic 1 state. Conversely, an active high signal is at a logic 1 state when asserted and at a logic 0 state when de-asserted. The terms reset, clear, and “low” can also mean logic 0; set or “high” can also mean logic 1.
- Active low signals are represented with an overline; active high signals have no overline. For example, $\overline{\text{REMREQ0}}$ is active low, PWREN0 is active high.
- There are many pins, signals, registers, and software bits common to both Bay 0 and Bay 1; these names may include the Bay number suffix (0 or 1), an “x” to represent either, or no suffix at all. For example, PWREN, PWREN0, or PWRENx may each be used to describe output pin(s).

## Functional Pin Descriptions

This section provides a description of each of the 28 pins of the CDP1020 as shown in Figure 2.

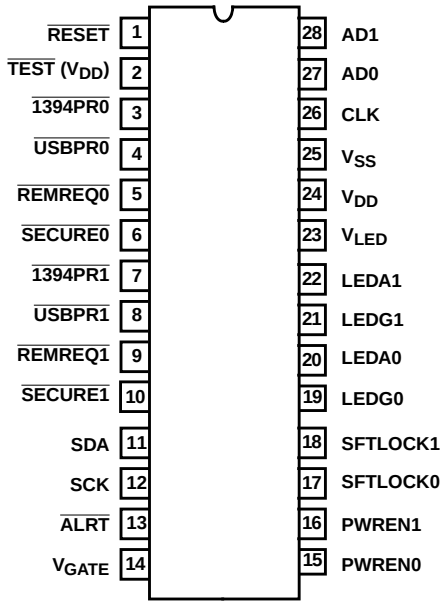



**FIGURE 2. PINOUT DIAGRAM FOR THE CDP1020**

NOTE: The following pins are “5V Tolerant” Inputs at all Operating Voltages: CLK, $\overline{\text{RESET}}$, SCK, SDA, $\overline{\text{ALRT}}$. This means that the input voltages can range up to the maximum allowed (5V typical), regardless of the operating voltage of the IC.

### $V_{DD}$, $V_{GATE}$, $V_{LED}$, and $V_{SS}$ (Power Supply)

Power is supplied to the CDP1020 using these pins. $V_{DD}$ is connected to the positive logic supply (typically either 3.3V or 5V), $V_{GATE}$ is connected to the positive supply for the PWREN0 and PWREN1 gate drivers (typically 12V), $V_{LED}$ is connected to the positive power supply for the LED drivers (typically 5V), and $V_{SS}$ is connected to the negative supply (Ground).

NOTE: $V_{GATE}$ and $V_{LED}$ power supplies should never be connected to the CDP1020 without the presence of the $V_{DD}$ and $V_{SS}$ supplies. Applying power to these inputs without the presence of the main power supply could result in a condition where all level-shifted outputs (PWRENx, LEDGx and LEDAx) track their power supply input voltage, thus enabling any of their output circuitry.

### $\overline{\text{RESET}}$ (Reset Input)

The $\overline{\text{RESET}}$ input is a low level active input, which resets the CDP1020. Resetting the device forces $\overline{\text{ALRT}}$ high and forces the device to reset the state of each bay (see ***Effects Of Reset*** for more details). The $\overline{\text{RESET}}$ pin contains an internal Schmitt Trigger to improve noise immunity.

### $\overline{\text{1394PR0}}$, $\overline{\text{1394PR1}}$, $\overline{\text{USBPR0}}$, $\overline{\text{USBPR1}}$

These four pins are the device presence inputs to the device bay controller for both bay 0 ($\overline{\text{1394PR0}}$ and $\overline{\text{USBPR0}}$) and bay 1 ($\overline{\text{1394PR1}}$ and $\overline{\text{USBPR1}}$). If the (peripheral) device uses the 1394 or USB (or both), that pin(s) on its connector is tied to GND (active low); when a device is inserted or removed, these pins are monitored to reflect whether a device is present (or not), and which of the two busses it uses. All of these pins are CMOS inputs with internal active pull-ups to $V_{DD}$.

### $\overline{\text{REMREQ0}}$, $\overline{\text{REMREQ1}}$

The $\overline{\text{REMREQ0}}$ and $\overline{\text{REMREQ1}}$ inputs are driven from the “REMOVE REQUEST” buttons for bay 0 and bay 1, respectively. These pins are CMOS inputs, with internal active pull-ups to $V_{DD}$.

### $\overline{\text{SECURE0}}$, $\overline{\text{SECURE1}}$

$\overline{\text{SECURE0}}$ and $\overline{\text{SECURE1}}$ are inputs which are integral to the bay security feature; each input should be connected such that it will be asserted when an optional hardware lock is engaged for the related bay. The state of these inputs are observable by the operating system through the SL_STS bits in the bay status registers. Both of these pins are CMOS inputs with internal active pull-ups to $V_{DD}$.

### SDA ($I^2C$/SMBus Data Input/Output)

The SDA pin is the serial data input to the SMBus interface logic of the CDP1020; it contains an internal Schmitt Trigger to improve noise immunity. When in slave-transmit mode, this pin is an open drain output. Input thresholds for the SDA input are fully compliant with SMBus Specification 1.0 (see **Electrical Specifications**). Refer to the ***$I^2C$/SMBus Interface*** text for more details.

### SCK ($I^2C$/SMBus Clock Input)

The SCK pin is the serial clock input to the SMBus interface logic of the CDP1020; it contains an internal Schmitt Trigger to improve noise immunity. Since the CDP1020 never acts as a SMBus master, this pin is a dedicated clock input (except for clock-stretching protocol, where it uses an open-drain low-side output). Input thresholds for the SCK input are fully compliant with SMBus Specification 1.0 (see **Electrical Specifications**). Refer to the ***$I^2C$/SMBus Interface*** text for more details.

### PWREN0, PWREN1

PWREN0 and PWREN1 are the outputs from the gate drive level-shifter circuitry located on the CDP1020. These pins will output $V_{GATE}$ (typically 12V) to drive the gates of the $V_{ID}$ control MOSFETs per Device Bay Specification 0.90.

### SFTLOCK0, SFTLOCK1

SFTLOCK0 and SFTLOCK1 are CMOS outputs designed to control the software locking mechanism installed in each bay. These outputs are designed to drive solenoid driver circuitry (such as an NFET), not a solenoid directly. The output can be programmed via the Special Function Register (SFR) to be a level or a pulse of a user defined duration. Refer to the ***Hardware*** text for more details.

### LEDG0, LEDA0, LEDG1, LEDA1

The LEDGx and LEDAx are the output LED drive pins for the device bay status LED located in each bay. The status indicators should be two color, green/amber, common anode or anti-parallel LEDs. These pins drive the LEDs directly through an external current limiting resistor; no additional buffering is necessary. The high side of these output drivers is powered directly from the $V_{LED}$ (typically 5V) power input and is not tied to the $V_{DD}$ rail of the device.

### CLK (Clock Input)

The Clock input (CLK) provides the time base reference for operation of the device bay controller logic, including: debounce timing, state sequencing, LED timing, etc. SMBus transfers between the CDP1020 and the SMBus Host controller are not based on the clock input. The clock input of the CDP1020 has the circuitry necessary for oscillating an external resistor-capacitor circuit, as shown in Figure 3. External clock sources (like those from a can oscillator) should **not** be used with the CDP1020.

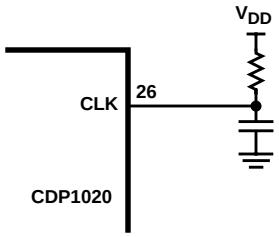



**FIGURE 3. RC OSCILLATOR CONFIGURATION**

The CDP1020 is designed to operate optimally with an input frequency of 4MHz. All of the internal timing, including debounce, insertion delay, and solenoid pulse durations are based on a 4MHz input. While the CDP1020 will operate over a wide range of frequency, a nominal input frequency of 4MHz is strongly recommended. Refer to the ***Hardware Interface*** text at the end of this document for more information on the RC oscillator, including recommended RC values.

### $\overline{ALRT}$ ($I^2C$/SMBus $\overline{Alert}$ Output)

The $\overline{ALRT}$ pin is the used by the CDP1020 to signal the SMBus Host Controller that an interrupt event within the controller has occurred and the device wishes to be read by the operating system. This pin is an open drain output. Refer to the ***$I^2C$/SMBus Interface*** text for more details.

### AD0, AD1

AD0 and AD1 are sampled by the CDP1020 immediately after reset and are used as the 2 least significant bits of the $I^2C$/SMBus slave address of the CDP1020. These pins are CMOS inputs, and should be tied to the same power plane as the device ($V_{DD}$ for a logic high; $V_{SS}$ for a logic low). Through the use of these pins, up to four devices may be placed on the same bus (addresses $90, $92, $94, $96).

### $\overline{TEST}$

$\overline{TEST}$ is a CMOS input used only by Intersil for testing, and is not recommended for the user; it **must** be externally connected to $V_{DD}$.

## $I^2C$/SMBus Interface

The CDP1020 contains a standard implementation of an $I^2C$/SMBus serial interface. The CDP1020 is always an $I^2C$/SMBus slave device. Its serial interface supports single or burst mode reads and writes using standard $I^2C$/SMBus mechanisms.

### Reading from and Writing to the CDP1020

The $I^2C$/SMBus slave address of the CDP1020 is %10010XXY, where the “XX” bits are defined by the AD1:AD0 input pins, and the “Y” is the R/W (Read/Write) bit. Every access of the CDP1020 begins when the $I^2C$/SMBus master generates a start condition (or repeated start condition) followed by transmitting an address/control byte with the address equal to the CDP1020’s slave address and the R/$\overline{W}$ bit set appropriately. Note that the R/$\overline{W}$ bit can be considered the 8th (Least Significant Bit) in the address, even though that may not be the standard $I^2C$/SMBus notation.

A write operation is defined as the condition when the $I^2C$/SMBus master transmits the slave address with the R/$\overline{W}$ bit clear. A read operation is defined as the transmission of the CDP1020 slave address with the R/$\overline{W}$ bit set. Thus, when looked upon as an 8-bit address, write operations will have even addresses (for example, $90), while read operations will be odd ($91).

For write operations, the command byte following the address/control byte is used to set the internal *address pointer* of the CDP1020. An $I^2C$/SMBus *send byte* command will therefore behave as a “set the address pointer” command. The address pointer is initialized to $00 following a reset. The byte written to the CDP1020 immediately following an address/control byte with the R/$\overline{W}$ bit clear will always be used to set the CDP1020 internal address pointer.

Subsequent bytes written to the CDP1020 in a transmission following the address/command and address pointer bytes are written directly into the register space of the CDP1020. The address pointer of the CDP1020 is auto-incrementing; once a byte is written into the register space, the address pointer increments to the next location. In this way multiple byte writes to adjacent address locations within the CDP1020 may be performed within a single $I^2C$/SMBus transmission. Figure 5 shows examples for a single byte and a multiple byte write to the CDP1020.

Read operations are performed when the $I^2C$/SMBus master transmits a start condition and the CDP1020 slave address with the R/$\overline{W}$ bit set. Following the address/command byte

the CDP1020 will transmit data to the master beginning at the CDP1020 register location pointer to by the internal address pointer. In accordance with I$^2$C/SMBus protocols, the CDP1020 will continue to transmit data to the master until it receives a negative acknowledge from the master.

NOTE: Due to the nature of the CDP1020 I$^2$C/SMBus interface module, a master device cannot simply end a read operation by transmitting a stop condition. The master is required to not-acknowledge the last byte of a read operation. Only when the CDP1020 detects this negative acknowledge will it end transmission and wait for another start condition.

As with write operations, the internal address pointer of the CDP1020 will automatically increment after each byte is transmitted during a read operation. In this way multiple bytes can be read from the CDP1020 register space within a single I$^2$C/SMBus transmission.

Figure 4 demonstrates single and multiple byte read sequences, both of which begin with a send byte command followed by a restart and read command. This technique forces the address pointer to the desired address prior to the read(s) and, while not strictly necessary, is strongly recommended.

Note that writing to un-implemented registers will be ignored (the data will not be stored anywhere). Reading unimplemented registers will produce undefined results.

### *I$^2$C/SMBus Alert Function*

The CDP1020 is a slave only I$^2$C/SMBus device. As such, it has no capability to start a transmission on the serial bus to notify the master of an *interrupt event* within the CDP1020 control logic (Interrupt events are described in more detail in the ***Device Bay Control Logic*** text). To notify the master of such an event, the CDP1020 implements the SMBus alert function as detailed in the SMBus specification.

When an interrupt event (interrupt events are described in the ***Device Bay Control Logic*** text) within the CDP1020 occurs, it will assert its $\overline{\text{ALRT}}$ signal. The $\overline{\text{ALRT}}$ pin is an active low, open drain output that must have an external pull-up resistor. The assertion of this signal is an indication to the master that an interrupt condition within the CDP1020 has occurred and needs service.

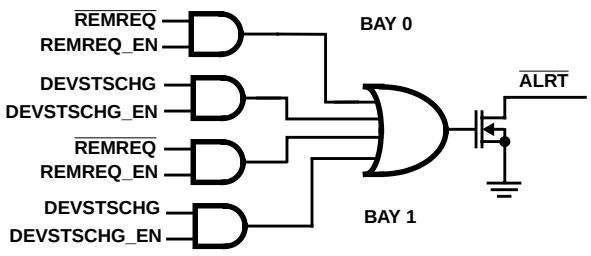



**FIGURE 4. $\overline{\text{ALRT}}$ OUTPUT LOGIC**

The master device may respond in two different ways to the assertion of the $\overline{\text{ALRT}}$ pin - one way using the SMBus Alert feature of SMBus control modules (such as the PiiX4/PiiX6) and the other way using a general purpose I/O to monitor the $\overline{\text{ALRT}}$ signal of the CDP1020 and I$^2$C messaging to service the alert.

In the SMBus method, the $\overline{\text{ALRT}}$ pin of the CDP1020 should be tied to the general purpose SMBus alert signal going into the SMBus controller. This signal may have many other devices connected to it in addition to the CDP1020. When the SMBus alert line is pulled low by any of the SMBus devices, the SMBus controller will send out an Alert Response Address (ARA), %00011001 ($19, with the R/$\overline{\text{W}}$ bit set). Any slave device that is currently asserting its $\overline{\text{ALRT}}$ signal will respond to the ARA by sending the master its slave address. If there are multiple devices asserting the $\overline{\text{ALRT}}$ signal, they will use standard SMBus arbitration techniques to determine ownership of the bus. Once a slave successfully transmits a response to the ARA, it will de-assert its $\overline{\text{ALRT}}$ signal. On the occurrence of any interrupt event, the CDP1020 will assert its $\overline{\text{ALRT}}$ pin. Once this pin is asserted, the CDP1020 will now respond to an SMBus ARA.

NOTE: The present CDP1020 does not support the SMBus ARA response, but a future mask option fully implements it. The same is true for a General Call ($00 address to broadcast to all devices on bus). Thus, if the SMBus alert signal is used, the SMBus controller should poll the CDP1020 (similar to the description in the next paragraph) to determine if it created the interrupt (it can separately issue the ARA to see if any other device responds to it; the CDP1020 will not).

For I$^2$C systems or SMBus systems that do not implement the SMBus alert feature, the $\overline{\text{ALRT}}$ pin of the CDP1020 should be connected to a separate interrupt pin or general purpose input of the I$^2$C/SMBus master and monitored. When the master detects that the $\overline{\text{ALRT}}$ signal of the CDP1020 has been asserted, it should perform a read operation on the CDP1020 using the standard read protocol described in the previous section. If multiple devices share the same input to the master, the software will have to check each device to determine which one caused the Alert.

Note that if the $\overline{\text{ALRT}}$ pin is only monitored in software (instead of using a more immediate interrupt), then the latency needs to be considered, such that there is not a noticeable or objectionable delay in the response time (for example, if the user pushes the $\overline{\text{REMREQ}}$ button repeatedly, or inserts and removes a device repeatedly because of no apparent response).

As shown in Figure 4, the CDP1020 will only de-assert its $\overline{\text{ALRT}}$ signal when the cause of the interrupt event is cleared (by clearing either the status or enable bit). This is true whether the SMBus ARA or the general purpose I/O method is used.

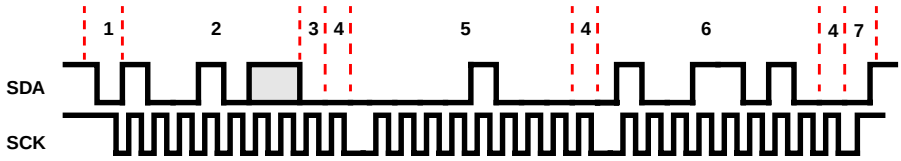



FIGURE 5A. SINGLE BYTE WRITE OPERATION (WRITE $9A TO CDP1020 REGISTER $08)

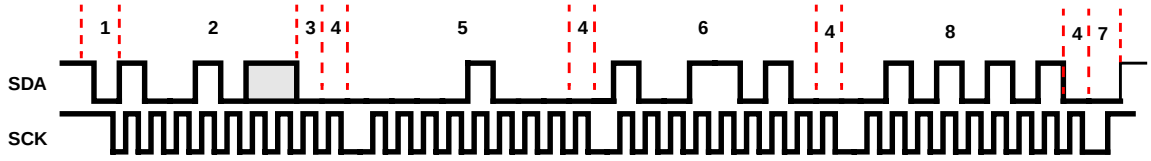



FIGURE 5B. MULTIPLE BYTE WRITE OPERATION (WRITE $9A TO CDP1020 REGISTER $08, $55 TO REGISTER $09)

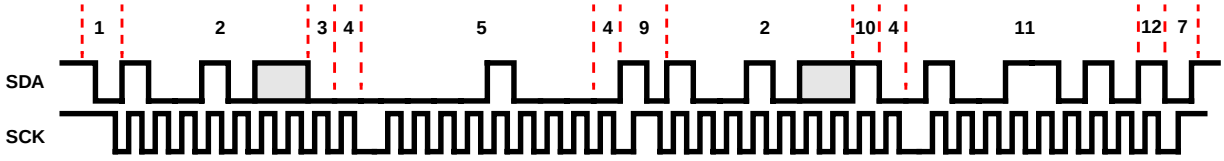



FIGURE 5C. SINGLE BYTE READ OPERATION (READ $9A FROM CDP1020 REGISTER $08)

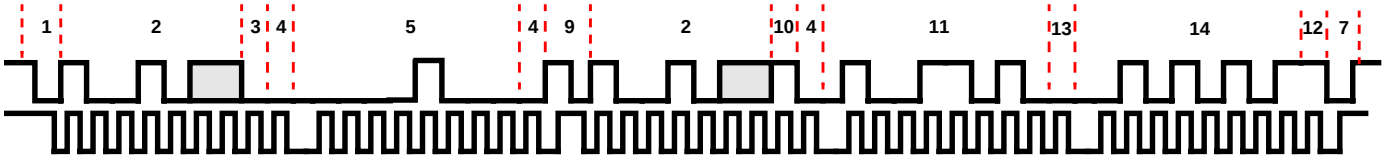



FIGURE 5D. MULTIPLE BYTE READ OPERATION (READ $9A FROM CDP1020 REGISTER $08, $55 FROM REGISTER $09)

FIGURE 5. CDP1020 $I^2C$/SMBUS TRANSMISSION PROTOCOLS

TABLE 1.

| NUMBER | DESCRIPTION | NUMBER | DESCRIPTION |
|---|---|---|---|
| 1 | Start Condition, generated by $I^2C$/SMBus master (defined as negative edge on SDA while SCK is high). | 8 | 2nd data byte written to the CDP1020. This data will be written to the register location set by the command byte, plus one. |
| 2 | CDP1020 $I^2C$/SMBus slave address (7 bits) | 9 | Repeated start condition |
| 3 | R/$\overline{W}$ bit, cleared indicating a write operation | 10 | R/$\overline{W}$ bit, set indicating a read operation |
| 4 | Acknowledge from CDP1020 ($\overline{ack}$; active low on SDA) | 11 | 1st byte of data read from the CDP1020, read from the CDP1020 register location set by the command byte in the write portion of the transmission. |
| 5 | Command byte sent from $I^2C$/SMBus master. This data will be used to set the internal address pointer of the CDP1020. | 12 | Negative master acknowledge. This signals to the CDP1020 that the master is done reading data and the CDP1020 should end transmission. |
| 6 | 1st data byte written to the CDP1020. This data will be written into the register specified by the command byte. | 13 | Master Acknowledge. This is an indication from the master that the read data has been received and the CDP1020 should continue to transmit data. |
| 7 | Stop condition, generated by the $I^2C$/SMBus master (defined as a positive edge on SDA while SCK is high). | 14 | 2nd data byte of read transmission. This data is read from the CDP1020 register location set by the command byte during the write portion of the transmission, plus one. |

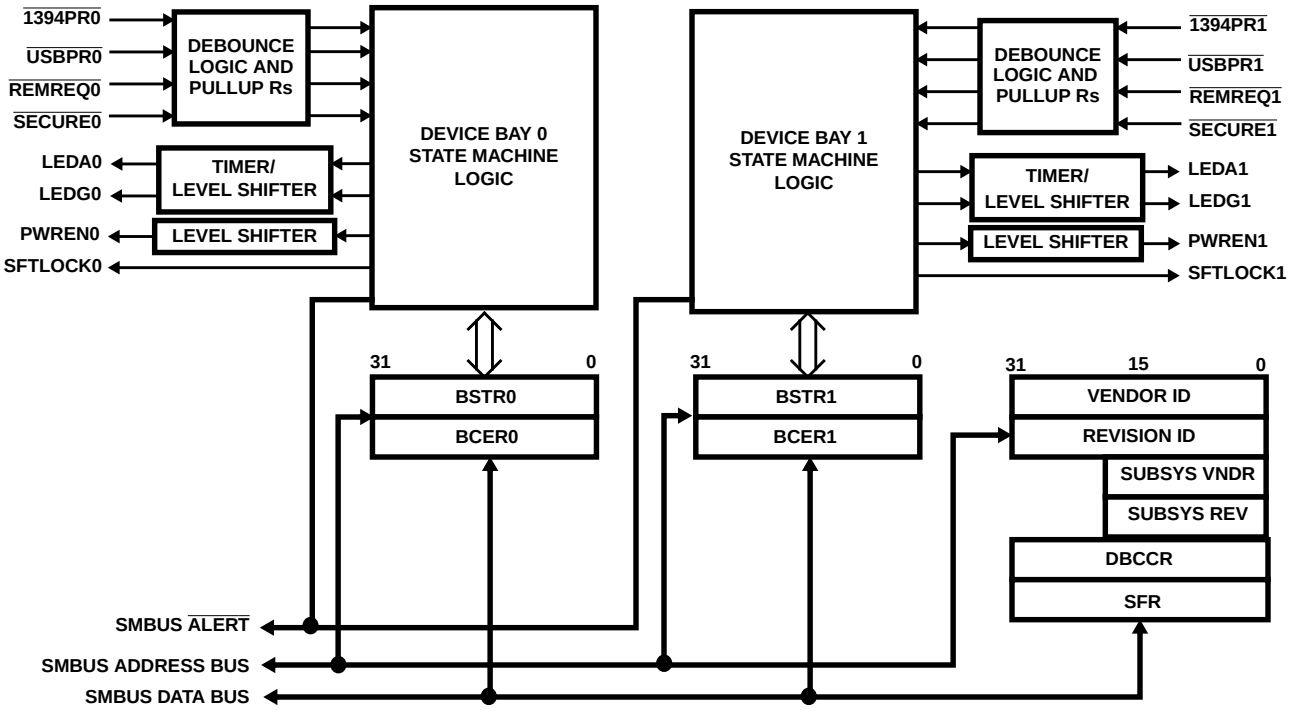



FIGURE 6. CDP1020 CONTROL LOGIC BLOCK DIAGRAM

## Device Bay Control Logic

The Device Bay Control Logic unit of the CDP1020 contains all of the state machine control logic for both device bays, the register set, debounce logic for the presence and remove request inputs, and the timer logic for the Device Bay status LEDs and software lock solenoids. All inputs from the Device Bays are received from the Input/Output Control Block. A block diagram of the Control Logic is shown in Figure 6.

### Programming Model

There are ten registers within the CDP1020, all of which may be read by the operating system at any time through the serial interface. The first nine registers implement the ACPI-based register set compliant with Device Bay Specification 0.90. The tenth register (SFR) provides control of other features not explicitly defined in Device Bay Specification 0.90. A memory map of the CDP1020 register space is shown in Figure 7.

The second byte transferred at the beginning of every SMBus write operation defines the address of a byte to access. Once the address is specified, it is stored in the CDP1020's address pointer. Subsequent reads or writes will access the register selected by the address pointer.

The address pointer will be incremented following each access facilitating burst mode accesses of sequential bytes in the CDP1020. Following access of the byte at $FF the address will increment to $00. Reading of unimplemented registers will produce undefined results.

Although the CDP1020 memory map is organized into ten multi-byte registers (8 4-byte registers and 2 2-byte registers), the BIOS/operating system is not required to read or write entire registers. All reads and writes take place at a byte granularity. For example, to write to the DBCCR, the BIOS needs to only write a single byte to address location $0C. The other three bytes of the DBCCR register ($0D:$0F) do not have to be written to.

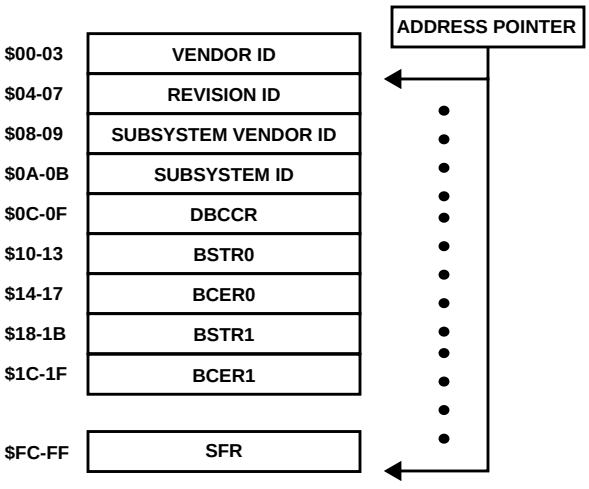



FIGURE 7. CDP1020 REGISTER SET/MEMORY MAP

NOTE: It is strongly recommended that all write-once-only registers and bits are programmed after a reset operation, even if the bits happen to be reset to the desired state, so that a noise transient during the address of a later write operation doesn't inadvertently write new data to any of those locations.

NOTE: *The register set in the CDP1020 is implemented in little-endian format, as specified in Device Bay Specification 0.90. As such, the least significant byte in any register is in the lowest memory address for that register; likewise the most significant byte is in the highest memory address. In the DBCCR, for example, the least significant byte (containing the configuration data) is at address $0C. The most significant byte (containing all 0's) is at address $0F.*

The following subsections describe each of the registers within the CDP1020.

### *Vendor ID Register, $00*

Per Device Bay Specification 0.90, the first register in the CDP1020 register set is the Vendor ID register. The contents of this register identify the manufacturer of the Device Bay Controller. This register is a read-only register that contains $1260, the 16-bit Intersil Corporation PCI SIG identification number. This number is contained in the lower two bytes of the register; the upper two bytes are always read as $0000.

| Bits 31:16 | Bits 15:0 | |
|---|---|---|
| $0000 | $1260 (Intersil PCI-SIG ID) | $00 |

### *Revision ID Register, $04*

The Revision ID register contains the 8-bit device bay controller manufacturer revision ID. This number is used to identify a particular Device Bay controller from the manufacturer specified in the Vendor ID register. The Revision ID is a read-only register that contains the 8-bit revision code for the CDP1020. This number is contained in the lower byte of the register; the upper three bytes are always read as $000000.

| Bits 31:8 | Bits 7:0 | |
|---|---|---|
| $000000 | Revision ID | $04 |

### *Subsystem Vendor ID Register, $08*

The Subsystem Vendor ID Register is used to identify the manufacturer of the device bay system that the CDP1020 is installed in. This register is implemented as a write-once-only register and is designed to be written by the system BIOS immediately after either the power-on-reset, or asserting the $\overline{\text{RESET}}$ pin, which enables a write to this register. Once written, this register becomes read-only and should contain the 16-bit subsystem manufacturer identification number. This register and the Subsystem ID register are the only 16-bit registers in the CDP1020.

| Bits 15:0 | |
|---|---|
| Subsystem Vendor ID | $08 |

### *Subsystem ID Register, $0A*

The Subsystem ID register contains a 16-bit subsystem vendor defined ID number. Typically, this number would define the device bay system revision or model number. Like the Subsystem Vendor ID, this register is implemented as a write-once-only register and is designed to be written by the system BIOS immediately after either the power-on-reset, or asserting the $\overline{\text{RESET}}$ pin, which enables a write to this register. Once written, this register becomes read-only. This register and the Subsystem Vendor ID register are the only 16-bit registers in the CDP1020.

| Bits 15:0 | |
|---|---|
| Subsystem ID | $0A |

### *Device Bay Controller Capabilities Register, $0C*

The Device Bay Controller Capabilities Register (DBCCR) is designed to allow the operating system to easily identify the features of the Device Bay system controlled by the CDP1020. This register contains five write-once-only bits, defined below. These bits, like those in the Subsystem Vendor ID Register, are designed to be written by the system BIOS immediately after power-on. Once written, they become read-only. The upper 27 bits of the DBCCR are always read as 0 This register is set to $00000002 at reset (two bays, no security locks).

| Bits 31:5 | Bit 4 | Bits 3:0 | |
|---|---|---|---|
| $00000 | SECLOCK | BAYCNT[3:0] | $0C |

**BITS 31:5**

Reserved for future use. Always read as 0.

**SECLOCK**

The SECLOCK bit indicates the presence of an optional physical security lock on at least one of the device bays controlled by the CDP1020. If set, then at least one of the device bays has the physical security lock, the state of which is available in the bay status register BSTRx. If clear, no physical security lock exists in the system. SECLOCK is implemented as a write-once-only bit intended to be written by the BIOS immediately after system power-on. Once written to, this bit becomes read-only.

**BAYCNT[3:0]**

The four BAYCNT bits represent the number of bays controlled by the CDP1020 in binary form. These bits, like the SECLOCK bit, are implemented as a write-once-only bits intended to be written by the BIOS immediately after system power-on. Once written to, they become read-only. Since the CDP1020 is a two bay controller, the only valid values that can be written to the BAYCNT[3:0] bits are 0, 1 and 2. If the OS tries to write any other value, the CDP1020 will write a value of 2 (%0010) to this bit field.

### *Bay Control and Enable Register, BCERx*

The CDP1020 incorporates two separate bay control and enable registers, one for each bay. The organization of these two registers is identical. BCER0 is located at address $10; BCER1 is at $18. Both BCER registers are cleared at reset.

**BITS 31:8**

Reserved. Always read as 0.

**BIT 7, LOCK_CTL**

The LOCK_CTL is a read/write bit that controls the software controlled solenoid interlock for each bay. This bit has two distinct modes of operation, depending on the value written into the SOL[3:0] bits in the SFR.

If the SOL[3:0] bits in the SFR contain any nonzero value, the LOCK_CTL bit logic is in “pulsed” mode. In pulse mode, the SFTLOCK output will be asserted for a fixed time duration when the LOCK_CTL is changed from a logic 1 to a logic 0. Writing a 0 to this bit while it is already 0 has no effect. Writing a 1 to this bit while it is a 0 will set the bit, but will not affect the SFTLOCK output. The duration of the SFTLOCK pulse is controlled by the SPD and SOL[3:0] bits in the SFR. Refer to the ***Special Function Register*** text for more details.

If the SOL[3:0] bit in the SFR are all 0, the LOCK_CTL circuitry is in “level” mode. In this case, the SFTLOCK output corresponding to the LOCK_CTL bit will simply follow the state of the LOCK_CTL bit. When the LOCK_CTL is set, the SFTLOCK output will be high; likewise, when the LOCK_CTL bit is clear, the SFTLOCK output will be low. Figure 7 shows the relationship between the LOCK_CTL bit and its corresponding SFTLOCK output in both level and pulsed modes.

**BITS 6:4, BAY_STREQ[2:0]**

This three bit field represents the state of the bay as requested by the operating system. It does not necessarily represent the actual state of the bay. The states are represented as such:

| | |
|---|---|
| 000 | No change requested |
| 001 | Request Bay State = Device Inserted |
| 010 | Request Bay State = Device Enabled |
| 011 | Request Bay State = Removal Requested |
| 100 | Request Bay State = Removal Allowed |
| 101 | Reserved |
| 110 | Reserved |
| 111 | Reserved |

If 000 is written, then no change to the current bay state is requested and the current nonzero value of this field is retained. This allows the operating system to modify other bits in this register without affecting the bay state. A bay state change will only occur when these fields are written if a device is inserted ($\overline{1394PRx}$ & $\overline{USBPRx}$ = 0). These bits may be read or written at any time by the operating system. These bits are cleared by any hardware transition to the Bay Empty State (i.e., device removal).

**BIT 3, REMREQ_EN**

This read/write bit allows the operating system to enable/disable internal CDP1020 interrupts and bay state transitions due to a logic “0” input value of the $\overline{REMREQx}$ pin. If this bit is clear, the CDP1020 will not notify the OS and will not transition the bay state to Removal Requested when the $\overline{REMREQx}$ button has been pushed. If this bit is set after the REMREQ_STS bit in the BSTR has been set, an interrupt event will be generated and a bay status change will occur. This bit is cleared by reset.

**BIT 2, DEVSTSCHG_EN**

This is a read/write bit that enables/disables internal CDP1020 interrupt events and bay state transitions due to the setting of the DEVSTSCHG bit in the BSTR. If this bit is clear, the CDP1020 will not notify the OS whenever the bay state has changed. The DEVSTSCHG bit in the BSTRx will still reflect a bay state change.

The DEVSTSCHG_EN bit also allows the CDP1020 to automatically transition the bay state to Device Inserted when an insertion event is the cause of the DEVSTSCHG. Hardware transitions to the Bay Empty state will always occur on a device removal, regardless of the state of the DEVSTSCHG_EN bit. This bit is cleared by reset.

**BIT 1, REMEVTWAK_EN**

This bit enables/disables internal CDP1020 interrupt events due to device removal. This bit gates the device removal event in the DEVSTSCHG logic (see Bay Status Register, below). The intent is to conditionally allow device removal as a wake-up event.

If clear, this bit will prevent the DEVSTSCHG flag in the BSTR from being set when a device is removed and the bay is in the Removal Allowed state. A hardware transition to the Bay Empty state will still occur. This bit does not affect any of the interrupt logic if the bay is not in the Removal Allowed state. This bit is cleared by reset.

**BIT 0, PWR_CTL**

The PWR_CTL is the enable bit for the $V_{ID}$ power rail. When set, the internal logic of the CDP1020 will output the $V_{GATE}$ voltage level on the PWREN pin associated with this register. (Refer to the **Power Enable System** text for more details) When clear, the PWREN pin will be pulled down to $V_{SS}$ by a standard N-Channel output driver. This allows the gate voltage of the $V_{ID}$ control MOSFET to be discharged quickly and the device switched off. No external pull down resistor is necessary.

This bit is cleared by reset. This bit cannot be set if there is no device in the bay ($\overline{1394PRx}$ & $\overline{USBPRx = 1}$) or if the LOCK_CTL bit is clear. If set and a device is suddenly removed (i.e., without OS permission), the CDP1020 will clear this bit and disable the PWRENx output.

Note: A single write to the BCER may set both the LOCK_CTL and PWR_CTL bits at the same time.

### *Bay Status Register, BSTRx*

Like the Bay Control and Enable register, there is one Bay Status register associated with each bay controlled by the CDP1020. The addresses for the BSTR registers are $14 (BSTR0) and $20 (BSTR1). Both BSTR registers are cleared at reset.

**BITS 31:11**

Reserved, always read as 0.

**BITS 10:8, BAY_FF[2:0]**

This three bit field indicates the form factor of the controlled bay:

| | |
|---|---|
| 000 | DB32 |
| 001 | DB20 |
| 010 | DB13 |
| 011- 111 | Reserved |

These bits are write-once only bits after a power-on reset and should be written by the system BIOS or the operating system at start-up. Once written, these bits become read-only. Subsequent internal and external resets do not affect the write status of these bits. The value of these bits is indeterminate at power on and are not affected by any type of reset.

**BIT 7, SL_STS**

The read only SL_STS indicates the state of the external security lock. This bit simply reflects the inverted state of the $\overline{\text{SECURE}}$x pin. If clear, external security lock is disengaged. If set, the lock is engaged.

The state of this bit depends on the external state of the $\overline{\text{SECURE}}$x pin *and* the state of the SECLOCK bit in the DBCCR register (see above). If the SECLOCK bit is clear, this bit will always read as a “0”. If SECLOCK is set, the state of this bit will reflect the inverted state of the $\overline{\text{SECURE}}$x pin.

**BIT 6:4, BAY_ST[2:0]**

This three bit field represents the actual state of the bay. These bits are read only. The bay state is represented as such:

| | |
|---|---|
| 000 | Bay Empty |
| 001 | Device Inserted |
| 010 | Device Enabled |
| 011 | Removal Requested |
| 100 | Device Removal Allowed |
| 101 | Reserved |
| 110 | Reserved |
| 111 | Reserved |

Bay states and how they are controlled is described in the ***State Machine Logic*** text.

**BIT 3, REMREQ_STS**

This bit indicates that the removal request button for this bay has been pressed ($\overline{\text{REMREQ}}$x pin has been driven low). This bit is referred to as a “sticky status bit”; once set, the you must write a “1” to this bit position to clear it. If the REMREQ_EN bit in the associated control register is set, the CDP1020 will generate a $\overline{\text{REMREQ}}$ interrupt event when this bit is set. A $\overline{\text{REMREQ}}$ interrupt event will cause a hardware transition of the bay state to Removal Allowed and assert the $\overline{\text{ALRT}}$ pin of the CDP1020 to notify the OS of the $\overline{\text{REMREQ}}$x button press.

**BIT 2, DEVSTSCHG**

This bit indicates that a hardware event has occurred that has changed the status of the device bay. This could be caused by a device insertion or a device removal. This bit, like the REMREQ_STS bit, is a sticky status bit; once set, the OS must write a “1” to clear it.

This bit will be set on all device removals except when the REMEVTWAK_EN bit in the BCER is clear and the bay is in the Removal Allowed state.

If the DEVSTSCHG bit is set due to an insertion event and the DEVSTSCHG_EN bit in the BCER is set, the CDP1020 will hardware transition the bay state to Device Inserted and assert the $\overline{\text{ALRT}}$ pin to notify the system of the insertion. This bit is cleared by reset.

**BIT 1, 1394PRSN_STS**

This bit reflects the inverted state of the $\overline{\text{1394PR}}$x pin associated with this status register. If the $\overline{\text{1394PR}}$x pin is high, this bit will read cleared. If the $\overline{\text{1394PR}}$x pin is low (1394 device inserted) this pin will read high. The setting of this bit can generate a device status change event. This bit is cleared by reset.

**BIT 0, USBPRSN_STS**

This bit reflects the inverted state of the $\overline{\text{USBPR}}$x pin associated with this status register. If the $\overline{\text{USBPR}}$x pin is high, this bit will read cleared. If the $\overline{\text{USBPR}}$x pin is low (USB device inserted) this pin will read high. The setting of this bit can generate a device status change event. This bit is cleared by reset.

### *Special Function Register, $FC*

The Special Function Register (SFR) allows control of various features not explicitly defined in the Device Bay Specification 0.90. This register contains write-once-only bits, which are designed to be written by the system BIOS immediately after power-on. Once written, they become read-only. The upper 24 bits of the SFR are always read as 0. All bits are cleared on reset.

**BITS 31:8**

Reserved for future use. Always read as 0...

| Bits 31:8 | Bits 7:5 | Bits 4:1 | Bit 1 | |
|---|---|---|---|---|
| $000000 | ITO[2:0] | SOL[3:0] | SPD | $FC |

**BITS 31:8**

Reserved for future use. Always read as 0.

**BITS 7:5, ITO[2:0]**

Bits 7, 6 and 5 define the Insertion Time-Out (ITO) bits field of the SFR.

These bits allow the OS/BIOS to specify the amount of time the CDP1020 will wait, from when it detects the insertion of a device until it notifies the OS. The insertion time-out should be used to allow Device Bay devices time to settle mechanically into the bay before they are enabled.

The 3-bit ITO field defines the time-out in 8 discrete increments of 800ms (nominal at 4MHz). The table below shows typical time-out values.

**INSERTION TIME-OUT VALUES (NOMINAL AT 4MHz)**

| ITO[2:0] | TIME-OUT VALUE |
|---|---|
| 000 | 0s |
| 001 | 0.8s |
| 010 | 1.6s |
| 011 | 2.4s |
| 100 | 3.2s |
| 101 | 4.0s |
| 110 | 4.8s |
| 111 | 5.6s |

Please note that these time-out values do not account for the fact that all presence inputs are debounced for 50ms before any time-out period begins. The time-out defaults to 0 after reset.

**SOL[3:0]**

The SOL[3:0] bits, along with the SPD bit, control the configuration and duration of the software lock solenoid drive pulse. Once written to, these bits become read-only.

If %0000 is written to the SOL[3:0] bits, the solenoid output is put into “level” mode. In level mode, the SFTLOCK output simply follows the state of the corresponding LOCK_CTL bit (see Figure 8).

If a nonzero value is written to the SOL[3:0] bits, then the solenoid control output is put into “pulsed” mode. In this mode, the solenoid output is pulsed anytime the LOCK_CTL bit is written from a 1 to a 0 (refer to Figure 8). The length of the pulse is determined by both the value written into the SOL[3:0] bits and the SPD bit.

With the SPD bit clear, the solenoid control circuitry is set to output short pulses. In terms of a prescaler, the solenoid pulse width is the value of the SOL[3:0] x 50ms. With the SPD bit set, the solenoid control is in long pulse mode. Here, the prescaler is set to SOL[3:0] x 800ms. The table below shows solenoid pulse widths for all values of SOL[3:0] and the SPD bit.

**SOLENOID PULSE WIDTHS (NOMINAL AT 4MHz)**

| SOL[3:0] | SOLENOID PULSE, SPD = 0 | SOLENOID PULSE, SPD = 1 |
|---|---|---|
| **0000** | **LEVEL** | **LEVEL** |
| 0001 | 50ms | 0.8ms |
| 0010 | 100ms | 1.6s |
| 0011 | 150ms | 2.4s |
| 0100 | 200ms | 3.2s |
| 0101 | 250ms | 4.0s |
| 0110 | 300ms | 4.8s |
| 0111 | 350ms | 5.6s |
| 1000 | 400ms | 6.4s |
| 1001 | 450ms | 7.2s |
| 1010 | 500ms | 8.0s |
| 1011 | 550ms | 8.8s |
| 1100 | 600ms | 9.6s |
| 1101 | 650ms | 10.4s |
| 1110 | 700ms | 11.2s |
| 1111 | 750ms | 12.0s |

NOTE: *Writing to SOL[3:0] bits will clear all LOCK_CTL bits and SFTLOCK outputs. Because these bits in the SFR are write-once-only, this situation will only occur on the first write. Subsequent writes to these bits will not cause the LOCK_CTL bits to clear.*

**SPD**

The SPD bit controls the length of the solenoid pulse when in pulse mode: long pulses If set, short pulses If clear. If the SOL[3:0] bits are clear, the SPD bit has no effect.

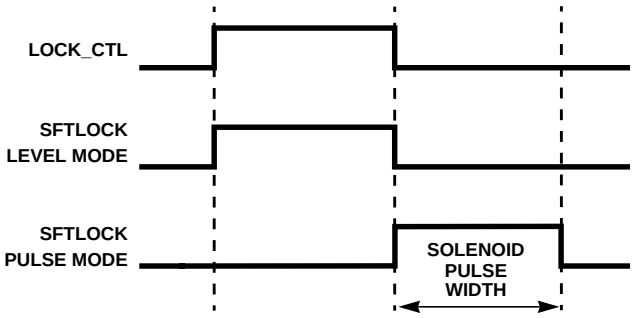



**FIGURE 8. LOCK_CTL BIT/SFTLOCK OUTPUT RELATIONSHIP IN LEVEL AND PULSE MODES**

## Effects of Reset

### Resets

The CDP1020 has two reset modes: an active low external reset pin ($\overline{\text{RESET}}$) and an internal power-on reset function. Both are logically OR'ed together internally.

### $\overline{\text{RESET}}$ Pin

The $\overline{\text{RESET}}$ input pin is used to provide an orderly start-up procedure and return the CDP1020 to a known state. When using the external reset mode, the $\overline{\text{RESET}}$ pin must stay low for a minimum of six (6) oscillator cycles (typically 1.5µs with a 4MHz clock). The $\overline{\text{RESET}}$ pin contains an internal Schmitt Trigger to improve noise immunity.

### Power-On Reset

The internal power-on reset occurs when a positive transition is detected on $V_{DD}$. The power-on reset is used strictly for power turn-on conditions and should not be used to detect any drops in the power supply voltage. There is no provision for a power-down reset.

The power-on circuitry provides a two oscillator cycle delay from the time that the oscillator becomes active. If the external $\overline{\text{RESET}}$ pin is low at the end of the time out, the CDP1020 remains in the reset condition until $\overline{\text{RESET}}$ goes high. The following list contains the actions of reset on internal circuits, but not necessarily in order of occurrence.

- BCER0 and BCER1 reset to $00000000
- BSTR0 and BSTR1 reset to $00000000
- SFR reset to $00000000
- All PWREN, SFTLOCK, and LED outputs cleared
- DBCCR set to two bays, no security locks - $00000002
- All Write-Once permissions reset
- AD0 and AD1 inputs sampled for $I^2C$/SMBus address
- Internal address pointer reset to $00

## State Machine Logic

The CDP1020 contains two functionally identical state machine logic blocks, one for each bay. Each of these blocks is responsible for monitoring external Device Bay events (i.e., device insertion, device removal, remove requests), controlling the locking mechanism and power enable signal for each bay, updating the state of the bay in response to OS commands and external events as shown in Figure 9.

There are five separate functional states that each bay of the CDP1020 can be in at any one time. Each of the two bay controllers within the CDP1020 functions completely independently of the other. The following sections detail state machine operation in each of the five states, including state transitions, operating system responsibility, and I/O functions.

In the following sections, the bay state controller will be referred to generically; that is as a bay "x" controller. Thus, bay "x" has a control register BCERx, a status register BSTRx, and so on.

The state of the device bay controller is changed through either hardware events (device inserted, device removed) and software events (OS writes into CDP1020 registers can change the bay state under certain conditions).

### Interrupt Events

An interrupt event is defined as one of the following:

- The insertion of a device into a bay with the corresponding DEVSTSCHG_EN bit set
- Removal of a device after the insertion time-out in any state other than Removal Allowed and the DEVSTSCHG_EN bit is set
- Removal of a device in the Removal Allowed state with both the REMWAKEVT and DEVSTSCHG_EN flags set
- User assertion of the $\overline{\text{REMREQ}}$ input when the associated REMREQ_EN bit is set and a device is present in the bay

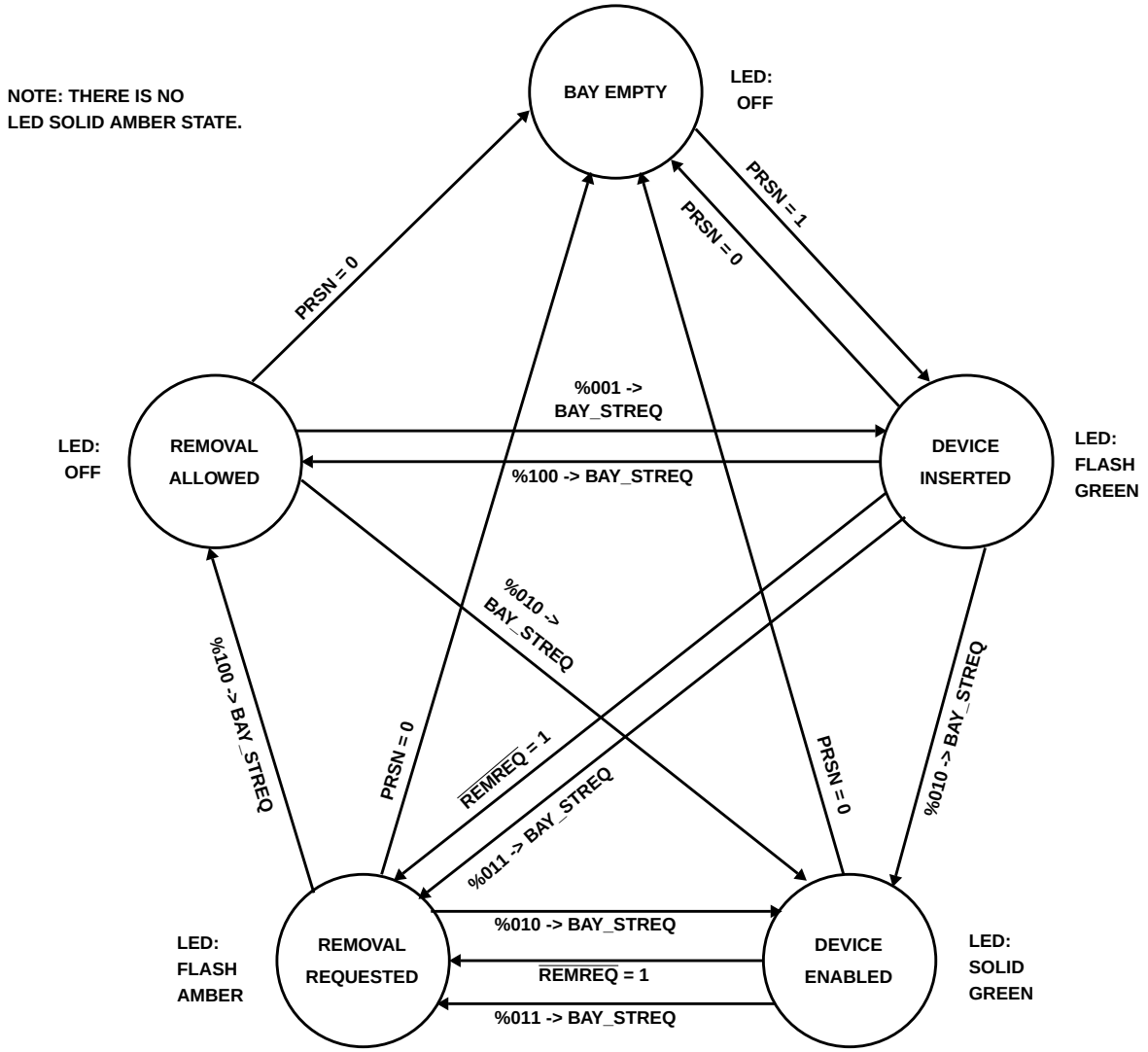



**FIGURE 9. CDP1020 BAY STATE DIAGRAM**

The first events listed above is known as an *insertion interrupt event*. On an insertion interrupt event, the bay state machine is transitioned to the Device Inserted state, the bay status LED is set to flash green, and the $\overline{ALRT}$ output is asserted to notify the OS of the device insertion.

The second and third events are *removal interrupt events*. If the DEVSTSCHG_EN flag is set, the CDP1020 will assert its $\overline{ALRT}$ output to notify the OS of the device removal. Regardless of the state of the DEVSTSCHG_EN bit, the CDP1020 will always transition the bay state to Bay Empty, clear the PWREN flag, turn of the PWREN output, and clear the bay state request field of the associated BCER.

The last event is a *remove request event*. When a user presses the $\overline{REMREQ}$ button on a bay with a device inserted, the CDP1020 will set the REMREQ_STS flag for that bay. If the REMREQ_EN bit is set, the CDP1020 will assert its $\overline{ALRT}$ output to notify the OS.

Figure 4 shows the output logic for the $\overline{ALRT}$ pin. As shown in this diagram, an interrupt event can be generated by setting an enable flag after the status bit has been set by some hardware event. For example, the REMREQ_EN flag is clear and a user presses the $\overline{REMREQ}$ button. The CDP1020 will set the REMREQ_STS flag, but will not generate an interrupt event. Now, the OS sets the REMREQ_EN flag. As soon as this occurs, the CDP1020 will assert its $\overline{ALRT}$ output and generate a $\overline{REMREQ}$ interrupt event.

### *Insertion Time Out*

When a device is inserted into a bay, it will make electrical contact with device bay connector very quickly. However, in most systems there will need to a delay while the user releases the device and it settles into the bay itself.

The insertion time out function of the CDP1020 allows a system designer to specify a delay time that the CDP1020 will wait before notifying the OS that a device has been inserted. This

delay time is set at system power on whenever the CDP1020 is initialized by writing the appropriate value into the ITO[2:0] bit field of the SFR. As shown in the SFR text, the insertion time-out delay can range from 0s to 5.6s in increments of 800ms.

The feature that makes the CDP1020 insertion time out function different from a simple insertion delay is that the CDP1020 will begin to flash the bay status LED green immediately after the device has been detected (if the DEVSTSCHG_EN flag for the bay is set). This is important because it gives the user instant feedback that the device has been recognized.

It is important to note that even though the bay status LED is flashing, the CDP1020 has not responded to the device insertion. During the insertion time out period, the status bits (1394PRSN_STS and USBPRSN_STS) remain clear and the bay state is NOT transitioned in the Device Inserted state. Thus, if the OS were to read the CDP1020 during the insertion time out period, it would not know that a device was in the bay.

Once the insertion time out period is over, the CDP1020 will generate an insertion event (assuming the DEVSTSCHG_EN flag is set) and the state controller will enter the Device Inserted state.

It is important to note that while waiting for the insertion time out, the CDP1020 has not fully registered the device in the bay. Thus, if the device were forcibly removed during this time, a removal event would not be generated. The CDP1020 would simply reset the insertion time out counter and stop flashing the bay status LED.

### Bay Empty

A bay empty state is defined exclusively as both the presence inputs for the bay ($\overline{1394PRx}$ and $\overline{USBPRx}$) de-asserted. The state of these pins is controlled by the insertion and removal of devices in the bay. As shown in Figure 9, the insertion of a device (and the assertion of one or both of the presence pins) causes the CDP1020 to recognize that a device is in the bay. If the DEVSTSCHG_EN flag for the bay is set, an insertion interrupt event will be generated and the state machine to transition bay state from Bay Empty (%000) to Device Inserted (%001).

Typically the Bay Empty state will be returned to from either the Device Inserted state (before the device has been enabled) or the Device Removal Allowed state (after the OS has powered down the device). However, the Bay Empty state can also be entered from Removal Requested and Device Enabled states if the device was forcibly removed from the system. In all cases, if the CDP1020 detects that both presence pins have been deasserted, the Bay Empty state will be entered and the PWREN output for the $V_{ID}$ MOSFET gate driver will be disabled.

The bay empty state is reflected by the bay state machine by setting BAY_ST[2:0] field (bits 6:4 of the bay status register) to %000. The bay empty state can be entered from any of the other four bay states and is entered exclusively through hardware transitions controlled by the CDP1020; OS writing into the BCER cannot change the bay state to Bay Empty.

Entering the Bay Empty state from any state other than Removal Allowed will cause the CDP1020 to generate a removal interrupt event if the DEVSTSCHG_EN bit in the BCER is set. A removal event will be generated if the bay was in the Removal Allowed state and both the REMEVTWAK the DEVSTSCHG_EN bits in the BCER are set. In both cases, the CDP1020 will notify the host system via the $\overline{ALRT}$ pin.

In the Bay Empty state the bay status LED (LEDGx and LEDAx) outputs will be off.

### Device Inserted State

The Device Inserted state, %001, is entered in one of two manners. When a device is inserted, the CDP1020 will transition to the Device Inserted state (after the insertion time out period) if the DEVSTSCHG_EN bit for the bay in question is set. In such a case, the CDP1020 will generate an insertion event, set the DEVSTSCHG bit in the BSTR, flash the bay status LED green and notify the OS through the $\overline{ALRT}$ pin.

Alternately, the OS can transition the CDP1020 to the Device Inserted state by writing a %001 to the BAY_STREQ bit field in the BCER. This can only be done from any other bay state as long as a device is inserted in the bay.

While in the CDP1020 is in the Device Inserted state the OS will typically engage the software controlled lock for the bay, enable $V_{ID}$ to the bay and enumerate the device on its native communication bus. While in this state, the bay status LED will flash green at 1Hz.

### Device Enabled

In the Device Enabled state, the device inserted into the bay has $V_{ID}$ enabled and is fully functional. This state cannot be entered through hardware action; only the OS writing %010 to the BAY_STREQ bits in the BCER can transition the bay state to Device Enabled.

In the Device Enabled state, if the presence pins are de-asserted at any time, the CDP1020 hardware will automatically transition the bay state to Bay Empty, clear the PWR_CTL bit and disable the PWRENx outputs. The PWRENx and PWR_CTL states are not affected by the state transitions to the Device Enabled state.

While in this state the bay status LED will be solid green.

### Device Removal Requested

The Device Removal Requested state, like the Device Inserted state, can be entered either through an OS write to the BCERx or through hardware actions.

Upon the assertion of the bay $\overline{REMREQ}$ input, the CDP1020 will set the REMREQ_STS bit in the BSTR if there is a device in the bay (i.e., 1394PRSN_STS or USBPRPRSN_STS = 1). A removal request interrupt event will be generated if the

REMREQ_EN bit is set. This event will transition the bay into the Removal Requested state, flash the bay status LED amber, and assert the $\overline{ALRT}$ pin to notify the OS.

The Device Removal Requested state is also entered when the OS writes a %011 to the BAY_STREQ field in the BCERx.

In either case, the CDP1020 will enter the Device Removal Requested state only if a device is inserted into the bay ($\overline{1394PRx}$ & $\overline{USBPRx}$ = 0).

While in this state the bay status LED will flash amber at 1Hz.

### *Device Removal Allowed*

The Device Removal Allowed state is the last of the five bay states. This state can only be entered through an OS write to the BCERx register. The CDP1020 hardware cannot transition the bay state into Device Removal Allowed.

The CDP1020 will transition the bay state controller to the Device Removal Allowed state when the OS writes a %100 to the BAY_STREQ field of the BCERx register. The CDP1020 will enter this state if and only if there is a device detected in the bay ($\overline{1394PRx}$ & $\overline{USBPRx}$ = 0).

The Device Removal Allowed state is defined as the last state that a device is in before being removed by a user. All bay controls, including PWREN outputs and solenoid control through the LOCK_CTL bit, are available in this state. Typically, though, a device in the Removal Allowed state will be disabled with its $V_{ID}$ supply off. From this state the OS may transition the CDP1020 state controller into any other state (except for Bay Empty, of course) by simply writing to the BAY_STREQ bits in the BCER. Also, removal request events will be generated if the REMREQ_EN bit is set and a user asserts the $\overline{REMREQ}$ input.

Finally, when in the Removal Allowed state, the CDP1020 will NOT generate a removal event if the device is removed from the bay and the REMEVTWAK bit is clear. This allows the OS to not be disturbed by the removal of a device that has already been powered down. If the REMEVTWAK flag is set, the CDP1020 will generate a removal event when a device is removed (assuming the DEVSTSCHG_EN bit is set). In any case, the CDP1020 will always transition the bay state to Bay Empty upon the removal of a device, regardless of the state of any of the control or status bits.

In the Device Removal Allowed state, the bay status LED (LEDGx and LEDAx) outputs will be off.

## *State Transitions*

The following tables illustrate the hardware (CDP1020 controlled) and software (OS controlled) state transitions for the CDP1020 bay state controllers.

**TABLE 2. HARDWARE EVENT BAY STATE TRANSITION TABLE**

| CURRENT STATE | NEXT STATE | HARDWARE EVENT | CONDITIONS | NOTES |
|---|---|---|---|---|
| ---- | Bay Empty (no device present) | Power-On Reset | | Establish initial state. |
| ---- | Bay Empty (device present) | Power-On Reset | | A device occupying a bay at power-on time will not necessarily cause DEVSTSCHG to be set. Furthermore, since DEVSTSCHG_EN defaults to ‘0’ no bay state transition will result. Either the system BIOS or the OS, upon loading, will automatically enumerate all occupied bays and MUST clear DEVSTSCHG if it was set. |
| Device Inserted | Bay Empty | Device removed from the bay. | Present bit(s) has transitioned from 1 to 0 independent of the state of DEVSTSCHG_EN. | Unexpected user behavior. The device was removed prior to being properly enabled by the OS. In this case the software controlled interlock mechanism may have been overridden. |
| Device Enabled | Removal Requested | User pressed the hardware removal request button, if present | REMREQ_STS and REMREQ_EN are both set. | User requested device removal through the hardware removal request button. |
| Removal Requested | Bay Empty | Device removed from the bay. | Present bit(s) has transitioned from 1 to 0 independent of the state of DEVSTSCHG_EN. | Unexpected user behavior. The device was removed prior to completion of the proper removal request sequence. In this case the software controlled interlock mechanism may have been overridden. |
| Device Removal Allowed | Bay Empty | Device was removed from the bay | Present bit(s) has transitioned from 1 to 0 independent of the state of DEVSTSCHG_EN. | Completion of normal device removal sequence. |

**TABLE 3. SOFTWARE ACTION BAY STATE TRANSITION TABLE**

| CURRENT STATE | NEXT STATE | SOFTWARE ACTION | NOTES |
|---|---|---|---|
| Bay Empty (device is present) | Device Inserted | 001b -> BAY_STREQ | This transition request could be the result of a device currently in the bay with DEVSTSCHG_EN cleared (as such the CDP1020 hardware could not transition to the bay state). |
| Bay Empty (device is present) | Device Enabled | 010b -> BAY_STREQ | This transition request is a result of a device that has been properly enumerated and enabled on its native bus(es). In all likelihood the DEVSTSCHG_EN bit was cleared so CDP1020 hardware could not first transition the bay state to Device Inserted. |
| Bay Empty (device is present) | Removal Requested | 011b -> BAY_STREQ | Unexpected OS behavior. |
| Bay Empty (device is present) | Device Removal Allowed | 100b -> BAY_STREQ | Unexpected OS behavior. |
| Device Inserted | Device Enabled | 010b -> BAY_STREQ | Device properly enumerated and enabled on its native bus(es). |
| Device Inserted | Removal Requested | 011b -> BAY_STREQ | Unexpected user behavior. User requested device removal through the UI before the device was enabled. |
| Device Inserted | Device Removal Allowed | 100b -> BAY_STREQ | OS could not enable the device. This could be a result of the OS's failure to properly enumerate the device, lack of sufficient operational power, etc. |
| Device Enabled | Device Inserted | 001b -> BAY_STREQ | Unexpected OS behavior. |
| Device Enabled | Removal Requested | 011b -> BAY_STREQ | User request device removal through the UI. |
| Device Enabled | Device Removal Allowed | 100b -> BAY_STREQ | Unexpected OS behavior. |
| Removal Requested | Device Inserted | 001b -> BAY_STREQ | Unexpected OS behavior. |
| Removal Requested | Device Enabled | 010b -> BAY_STREQ | OS decided that device removal was not allowed. This could be the result of an active application disallowing the removal. Alternatively, the user could have cancelled the removal request through the UI. |
| Removal Requested | Device Removal Allowed | 100b -> BAY_STREQ | OS has completed the device removal sequence. |
| Device Removal Allowed | Device Inserted | 001b -> BAY_STREQ | User has requested to "re-use" the device through the UI. This state transition eliminates the need for the user to remove the device and then immediately re-insert it. This could be especially useful in the presence of an engaged physical security lock. This bay state transition might be used in order to provide user feedback (via the bay status indicator) while OS is performing steps necessary to re-enable the device. |
| Device Removal Allowed | Device Enabled | 010b -> BAY_STREQ | User has requested to "re-use" the device through the UI. This state transition eliminates the need for the user to remove the device and immediately re-insert it. This could be especially useful in the presence of an engaged physical security lock. This state transition indicates that the device has been properly re-enumerated and re-enabled on its native bus(es). |
| Device Removal Allowed | Removal Requested | 011b -> BAY_STREQ | Unexpected OS behavior. |

## Hardware Interface

The hardware interface of the CDP1020 is designed to be fully compliant with the Device Bay specification 0.90 and includes many optional and value-added features that reduce system component count, system complexity and overall system cost.

### CLK Input

As stated at the beginning of this document, the Clock input (CLK) provides the basic time base reference for operation of all CDP1020 control logic (SMBus transfers between the CDP1020 and the SMBus Host controller are not based on the clock input). The clock input of the CDP1020 has the circuitry necessary for oscillating an external resistor-capacitor circuit, as shown in Figure 10. External clock sources (like those from a can oscillator) should not be used with the CDP1020.

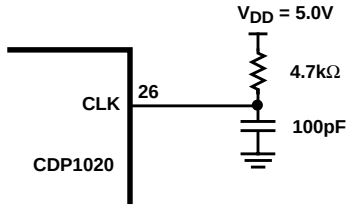



**FIGURE 10. RC OSCILLATOR CONFIGURATION**

The input frequency of the CDP1020 will vary for different values of the oscillator resistor, the oscillator capacitor and $V_{DD}$. The CDP1020 is designed to operate optimally with an input frequency of 4MHz. All of the internal timing of the CDP1020, including debounce, insertion delay, and solenoid pulse durations are based on a 4MHz input. While the CDP1020 will operate at any frequency in the 3.0MHz to 5.0MHz range, an input frequency of 4MHz is strongly recommended. Also, it is recommended that the value of the oscillator capacitor be fixed at 100pF for optimal operation of the oscillator.

Figure 11 shows a plot of oscillator frequency vs. resistance at 5.0V and 3.3V when the oscillator capacitance is held fixed at 100pF. Recommended values for the oscillator when $V_{DD}$ = 5.0V are 4.7kΩ and 100pF; 3.3kΩ and 100pF when $V_{DD}$ = 3.3V.

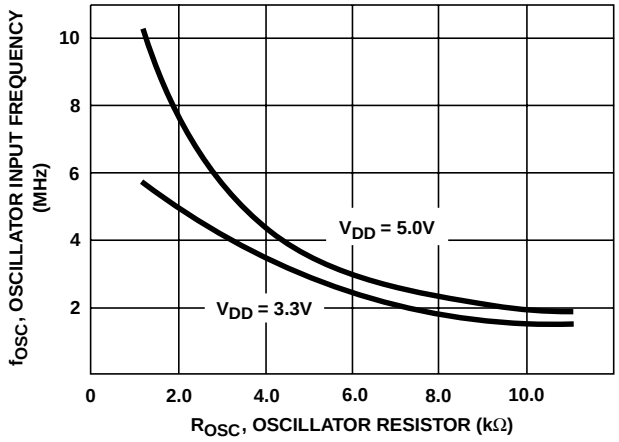



**FIGURE 11. OSCILLATOR FREQUENCY vs $R_{OSC}$ ($C_{OSC}$ = 100pF)**

### Presence Pins

The CDP1020 has two device presence detect inputs ($\overline{1394PRx}$ and $\overline{USBPRx}$) for each bay, for a total of four in all. These inputs are mandatory in the Device Bay specification and are the primary means of notifying the system that a device has been inserted into a bay. Figure 18 shows a typical connection of the CDP1020 presence pins.

The 1394 and USB presence lines are pulled up to the $V_{DD}$ of the CDP1020 through active pull-up resistors located on the CDP1020. These on chip resistors eliminate the need for external pull-ups. Whenever any device is inserted into a bay, that device must pull at least one (and possibly both) of the presence pins to ground. Doing so indicates to the CDP1020 that a new device has been inserted and what communications protocol it uses. Also, all presence pin inputs to the CDP1020 are debounced internally. The debounce time is set internally to approximately 1 second to allow for transitions in the presence pin state while the device is being inserted.

Typical current vs. voltage curves for the presence pin pull-up resistor are shown in Figure 12. Pin assignments for $\overline{1394PR0}$, $\overline{USBPR0}$, $\overline{1394PR1}$ and $\overline{USBPR1}$ are shown on the cover page and in Figure 2.

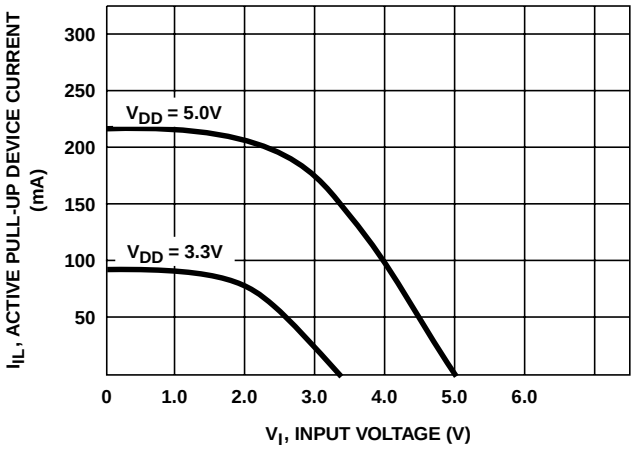



**FIGURE 12. TYPICAL ACTIVE PULL-UP CURRENT AT 25°C**

### Remove Request ($\overline{REMREQ}$) Inputs

The $\overline{REMREQ}$ inputs connect to an external, panel mounted push button that allows the user to request removal of a device. The implementation of the $\overline{REMREQ}$ feature is an optional but encouraged feature of Device Bay specification 0.90. Each bay is assigned its own unique $\overline{REMREQ}$ input, as such the CDP1020 implements $\overline{REMREQ0}$ and $\overline{REMREQ1}$. Typical connection to a push button is shown in Figure 18.

The $\overline{REMREQ}$ inputs, like the presence pins, incorporate integrated internal active pull-up devices. This allows the system designer to implement the push button as a simple momentary-on switch that pulls the input to ground when pressed. When pressed, the CDP1020 will debounce the switch press internally. No external pull-up resistors or lowpass debounce filters are necessary.

### $\overline{\text{SECURE}}$ Inputs

The $\overline{\text{SECURE0}}$ and $\overline{\text{SECURE1}}$ inputs are used by the CDP1020 to monitor the state of an external device security lock. The purpose of this lock is to prevent unauthorized removal or theft of a device in a device bay system. Like the $\overline{\text{REMREQ}}$ inputs, this is an optional feature of the device bay system. Typical connection of the SECURE inputs is shown in Figure 18.

Like the presence and $\overline{\text{REMREQ}}$ inputs, the $\overline{\text{SECURE}}$ inputs implement internal active pull-up devices. When the security lock is disengaged (i.e., the device may be removed from the bay), the lock switch should be in an open state, thus rendering the $\overline{\text{SECURE}}$ input at a logic high level. When the lock is engaged, the lock switch should then close and pull the $\overline{\text{SECURE}}$ input to ground. If the security lock function of the device bay system is not implemented, these pins should be tied to $V_{DD}$ of the CDP1020.

### Address Select Pins AD1 and AD0

The address select pins are used by the CDP1020 as the lower two bits of its 7-bit $I^2C$/SMBus slave address. By making these two pins configurable externally, up to four CDP1020 devices can share the common bus. The upper five bits of the $I^2C$/SMBus address are internally hard-wired to %10010xx. Note that the R/$\overline{W}$ bit can be considered the 8th bit (LSB) in the address.

The AD0 and AD1 pins should be connected to the $V_{DD}$ or the $V_{SS}$ power supplies of the CDP1020, as appropriate. They should not be left floating; doing so will cause the CDP1020 to randomly configure its slave address on power-up. These pins are sampled immediately after reset.

### Serial Interface Connections

The serial interface connections of the CDP1020 include the SDA, SCK and $\overline{\text{ALRT}}$ pins. All three of these pins are open-drain when in the output mode and are +5V tolerant. The functionality of these pins is described in the ***$I^2C$/SMBus Interface*** text. Typical drive characterization is shown in Figure 13.

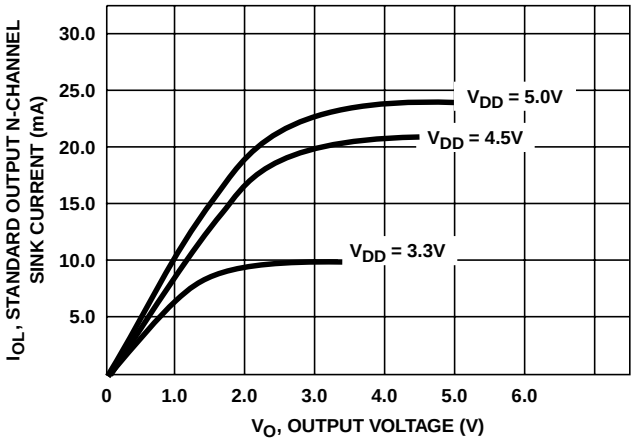



**FIGURE 13. SDA, SCK AND $\overline{\text{ALRT}}$ OUTPUT N-CHANNEL SINK CURRENT AT 25°C**

### $\overline{\text{ALRT}}$ - Master Alert Signal

The $\overline{\text{ALRT}}$ pin of the CDP1020 is an open drain output used to signal the $I^2C$/SMBus master that a device state change has occurred and that this state should be processed by the operating system. Unlike the presence, $\overline{\text{REMREQ}}$ and security inputs, there is no internal pull-up on the $\overline{\text{ALRT}}$ pin. As such, there needs to be one present in the external system. Typical connection of the $\overline{\text{ALRT}}$ pin is shown in Figure 18.

The functionality of the $\overline{\text{ALRT}}$ pin is described in the ***State Machine Logic*** text.

### $\overline{\text{RESET}}$ (Reset Input)

The $\overline{\text{RESET}}$ pin can be connected in the following ways:

- If not used, must be tied to $V_{DD}$.
- Connected to a system reset (such as from the PiiX4 South Bridge). Be sure it is an active low polarity.
- Connected to an external RC (Resistor to $V_{DD}$, Capacitor to Ground), to act as an additional power-on reset. The time constant should be chosen to be longer than the turn-on ramp of the $V_{DD}$ power supply used.

### Power Supply

The power supply input for the CDP1020 are the $V_{DD}$ (positive) and $V_{SS}$ (negative) pins. The $V_{SS}$ pin should be connected to the system ground. The $V_{DD}$ pin should be connected to a positive power supply from +3.3V to +5.0V. There are few differences in performance between voltages; in general, the higher voltage will allow more output drive current (LED pins, but the current should be limited by an external resistor anyway), and a higher gate voltage for the SFTLOCK pins (driving Logic-Level FETs). However, the higher voltage may also consume a higher supply current (the amount of available supply current may help decide which one to use). Keep in mind that the Oscillator RC needs to be chosen to match the $V_{DD}$ (in order to get the optimal 4MHz frequency).

### Power Enable System

The Power Enable System enables the CDP1020 to directly drive the gate of the $V_{ID}$ control MOSFETs. Logically, the PWRENx output matches the state of the PWR_CTLx bit in the corresponding BCER register. However, instead of being a logic level CMOS output, the PWREN outputs are “level-shifted” such that their output current is supplied directly from the $V_{GATE}$ power input pin.

When the PWR_CTL bit is a logic 0, the PWREN output will sink current to $V_{SS}$ through its N-Channel output driver. However, when the PWR_CTL bit is a logic 1, the PWREN output will source current, limited to a maximum of 25µA, from the $V_{GATE}$ power input. The $V_{GATE}$ input is typically tied to the +12V power supply of the system. Figure 14 shows a block diagram of this system.

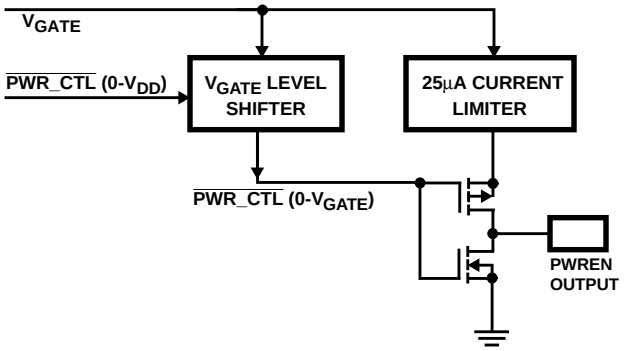



**FIGURE 14. POWER ENABLE OUTPUT CIRCUITRY**

The result is a +12V signal that can be used to directly drive the gate of a N-Channel MOSFET used to control the $V_{ID}$ to the bay. For best operation, a Intersil HUF76113DK8 Dual N-Channel Logic Level Power MOSFET is recommended for $V_{ID}$ switching.

The CDP1020 controls the turn on time of the $V_{ID}$ MOSFET by limiting the amount of current that can be drawn from the PWREN outputs. The constant current source capabilities of the PWREN pins is typically 25μA. For designs that require slower turn-on times, external capacitors can be added in parallel with the gate of the $V_{ID}$ MOSFET. Figure 15 below shows the rise time of the PWREN outputs into the gate of a Intersil HUF76113DK8 MOSFET.

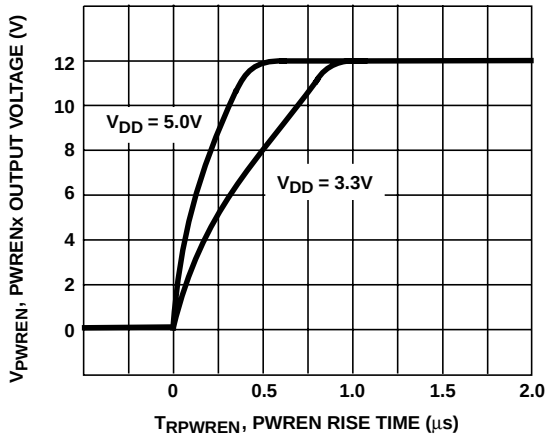



**FIGURE 15. PWREN OUTPUT INTO GATE OF HUF76113DK8 $V_{GATE}$ = 12.0V (TYPICAL, 25°C)**

### *LED Drive Pins*

As discussed in the ***State Machine Logic*** and ***User Interface*** texts, the CDP1020 has the capability to directly drive two dual color bay status LED indicators. Each bay controlled by the CDP1020 has a green (LEDGx) and amber (LEDAx) drive pin. These pins are capable of sourcing or sinking 6mA at 3.3V and 15mA at 5.0V. Since their drive capability is symmetric, the LED drive pins can control three terminal common anode LEDs, three terminal common cathode LEDs, or two terminal two color LEDs. Note that external limiting resistors are required. Typical circuit connections are shown in Figures 16A and 16B.

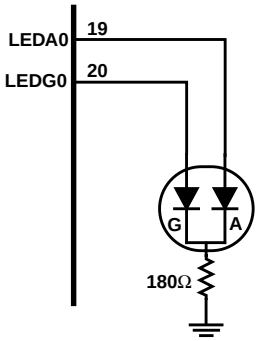



**FIGURE 16A. LED DRIVER CONNECTIONS TO A 3-TERMINAL AMBER/GREEN LED**

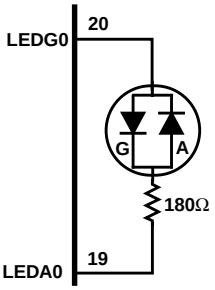



**FIGURE 16B. LED DRIVER CONNECTIONS TO 2-TERMINAL AMBER/GREEN LED**

As mentioned at the beginning of this document, the high side of the LED drive pins is switched directly to the $V_{LED}$ power input. Thus, all current used by the bay status LEDs is drawn from the $V_{LED}$ power input rather that $V_{DD}$. This method is similar to that used on the PWREN outputs, except that there is no current limiting and the $V_{LED}$ input cannot rise above 5.0V. A block diagram of one of the LED drive pins (all four drive pins are identical) is shown in Figure 17. The purpose of $V_{LED}$ is to allow system architects to drive the bay status LEDs from a separate supply than that which is powering the CDP1020. For those systems that do not wish to drive the LED outputs from a separate supply, simply tie $V_{DD}$ and $V_{LED}$ together. $V_{LED}$ may be greater than, less than or equal to the $V_{DD}$ input of the CDP1020.

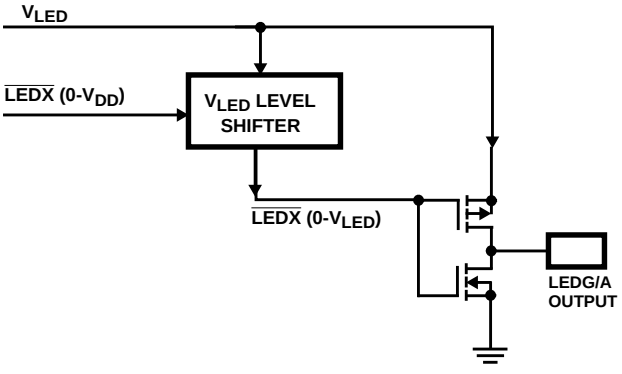



**FIGURE 17. LED DRIVER OUTPUT CIRCUITRY**

### Solenoid Drivers

For control of the device bay software controlled solenoids, the SFTLOCK pins have drive capability to gate an external, solenoid control MOSFET. Typical circuit configuration is shown in Figure 18. Note that since the output voltage is not charge-pumped, only the $V_{DD}$ voltage (3.3V to 5V) is available to drive a FET input gate; therefore, a logic level FET such as the Intersil HUF76113DK8 is recommended.

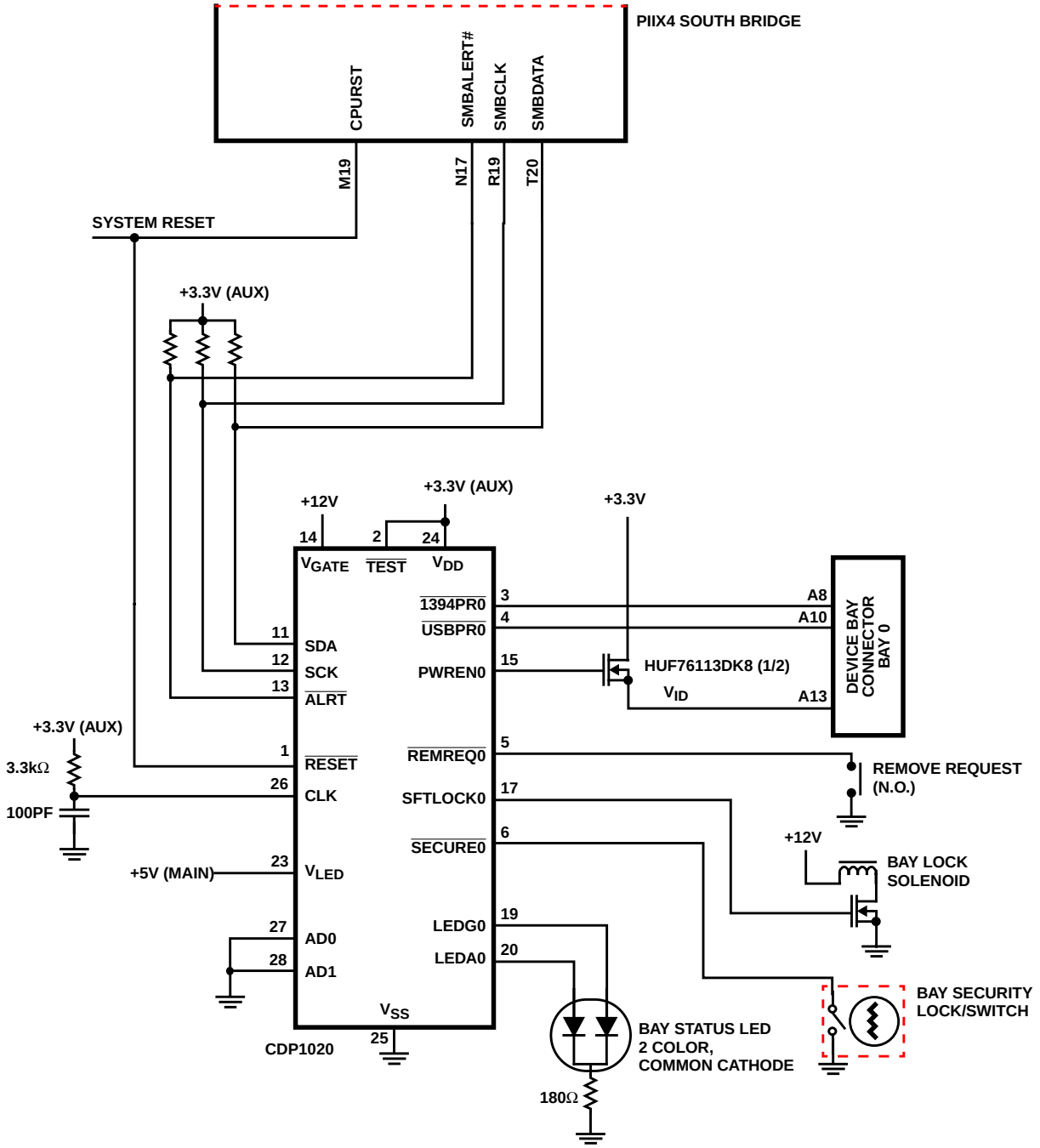



**FIGURE 18. TYPICAL CDP1020 SYSTEM HARDWARE CONNECTIONS - PiiX4 BASED INTEL ARCHITECTURE PLATFORM (ONLY BAY 0 SHOWN)**

All Intersil semiconductor products are manufactured, assembled and tested under **ISO9000** quality systems certification.

*Intersil semiconductor products are sold by description only. Intersil Corporation reserves the right to make changes in circuit design and/or specifications at any time without notice. Accordingly, the reader is cautioned to verify that data sheets are current before placing orders. Information furnished by Intersil is believed to be accurate and reliable. However, no responsibility is assumed by Intersil or its subsidiaries for its use; nor for any infringements of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of Intersil or its subsidiaries.*

For information regarding Intersil Corporation and its products, see web site **http://www.intersil.com**