

# ADNS-3080

## High-Performance Optical Mouse Sensor



### Data Sheet



Lead (Pb) Free  
RoHS 6 fully  
compliant



#### Description

The ADNS-3080 is a high performance addition to Avago Technologies' popular ADNS family of optical mouse sensors.

The ADNS-3080 is based on a new, faster architecture with improved navigation. The sensor is capable of sensing high speed mouse motion - up to 40 inches per second and acceleration up to 15g - for increased user precision and smoothness.

The ADNS-3080 along with the ADNS-2120 (or ADNS-2120-001) lens, ADNS-2220 (or ADNS-2220-001) assembly clip and HLMP-ED80-XX000 form a complete, compact optical mouse tracking system. There are no moving parts, which means high reliability and less maintenance for the end user. In addition, precision optical alignment is not required, facilitating high volume assembly.

The sensor is programmed via registers through a four-wire serial port. It is packaged in a 20-pin staggered dual inline package (DIP).

#### Theory of Operation

The ADNS-3080 is based on Optical Navigation Technology, which measures changes in position by optically acquiring sequential surface images (frames) and mathematically determining the direction and magnitude of movement.

It contains an Image Acquisition System (IAS), a Digital Signal Processor (DSP), and a four-wire serial port.

The IAS acquires microscopic surface images via the lens and illumination system. These images are processed by the DSP to determine the direction and distance of motion. The DSP calculates the  $\Delta x$  and  $\Delta y$  relative displacement values.

An external microcontroller reads the  $\Delta x$  and  $\Delta y$  information from the sensor serial port. The microcontroller then translates the data into PS2 or USB signals before sending them to the host PC or game console.

#### Features

- High speed motion detection – up to 40 ips and 15g
- New architecture for greatly improved optical navigation technology
- Programmable frame rate over 6400 frames per second
- SmartSpeed self-adjusting frame rate for optimum performance
- Serial port burst mode for fast data transfer
- 400 or 1600 cpi selectable resolution
- Single 3.3 volt power supply
- Four-wire serial port along with Chip Select, Power Down, and Reset pins

#### Applications

- Mice for game consoles and computer games
- Mice for desktop PC's, Workstations, and portable PC's
- Trackballs
- Integrated input devices

## Pinout

| Pin | Name     | Description                              |
|-----|----------|--|
| 1   | NCS      | Chip select (active low input)           |
| 2   | MISO     | Serial data output (Master In/Slave Out) |
| 3   | SCLK     | Serial clock input                       |
| 4   | MOSI     | Serial data input (Master Out/Slave In)  |
| 5   | LED_CTRL | LED control output                       |
| 6   | RESET    | Reset input                              |
| 7   | NPD      | Power down (active low input)            |
| 8   | OSC_OUT  | Oscillator output                        |
| 9   | GUARD    | Oscillator gnd for PCB guard (optional)  |
| 10  | OSC_IN   | Oscillator input                         |
| 11  | NC       | No connect                               |
| 12  | OPTP     | Connect to VDD3                          |
| 13  | REFC     | Reference capacitor                      |
| 14  | REFB     | Reference capacitor                      |
| 15  | VDD3     | Supply voltage                           |
| 16  | GND      | Ground                                   |
| 17  | VDD3     | Supply voltage                           |
| 18  | NC       | No connect                               |
| 19  | GND      | Ground                                   |
| 20  | NC       | No connect                               |

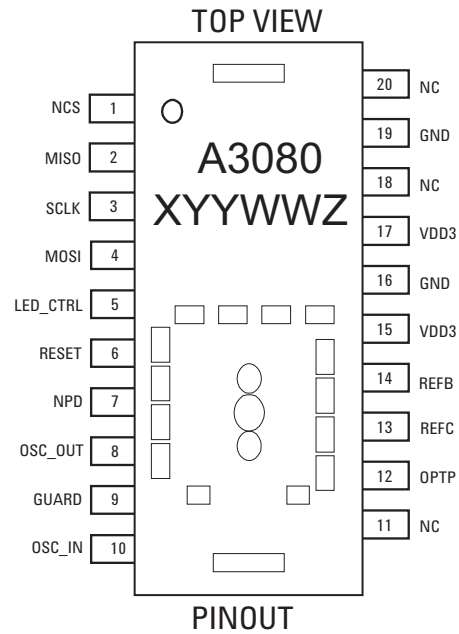


Figure 1. Package outline drawing (top view)

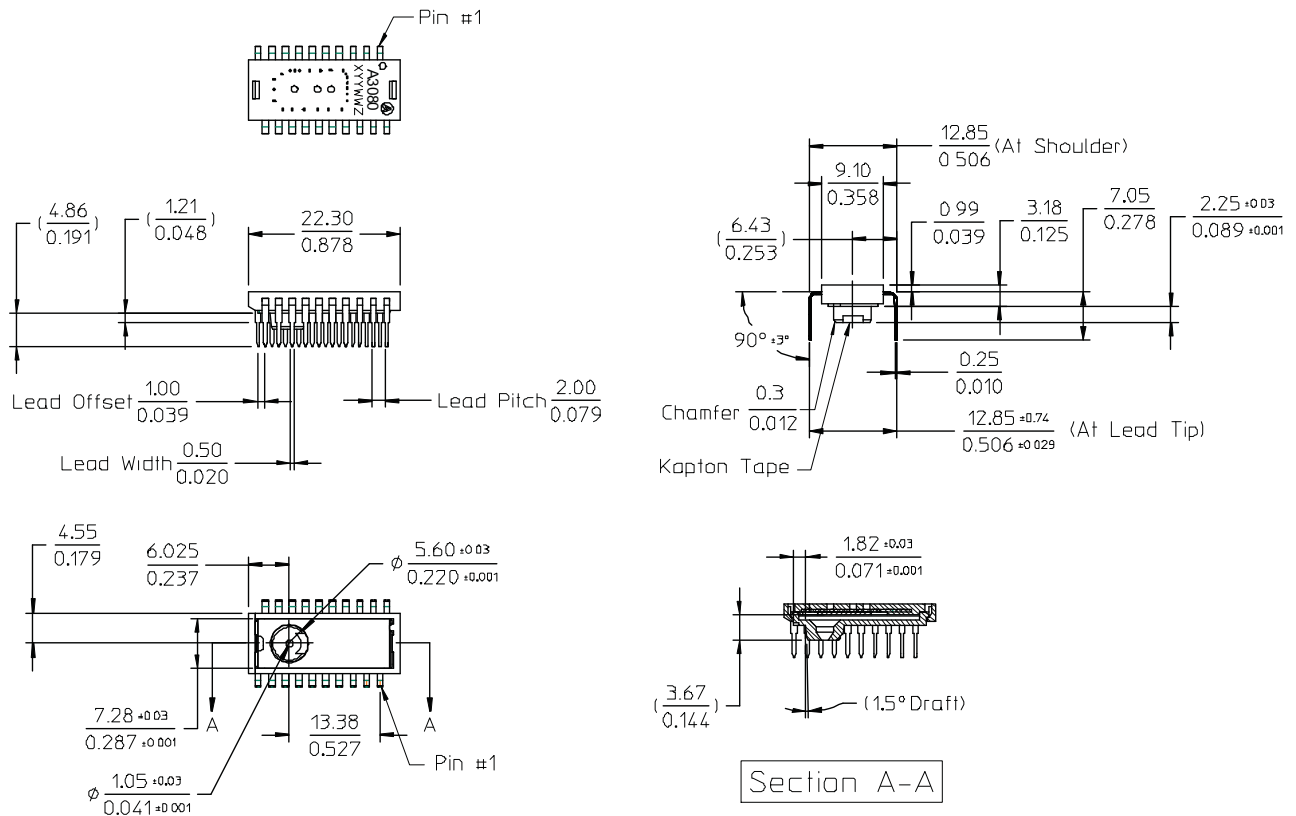


Figure 2. Package outline drawing

**CAUTION:** It is advised that normal static precautions be taken in handling and assembly of this component to prevent damage and/or degradation which may be induced by ESD.

## Overview of Optical Mouse Sensor Assembly

### 2D Assembly Drawing of ADNS-3080

Shown with ADNS-2120, ADNS-2220 and HLMP ED80-XX000.

Avago Technologies provides an IGES file drawing describing the base plate molding features for lens and PCB alignment.

The components interlock as they are mounted onto defined features on the base plate.

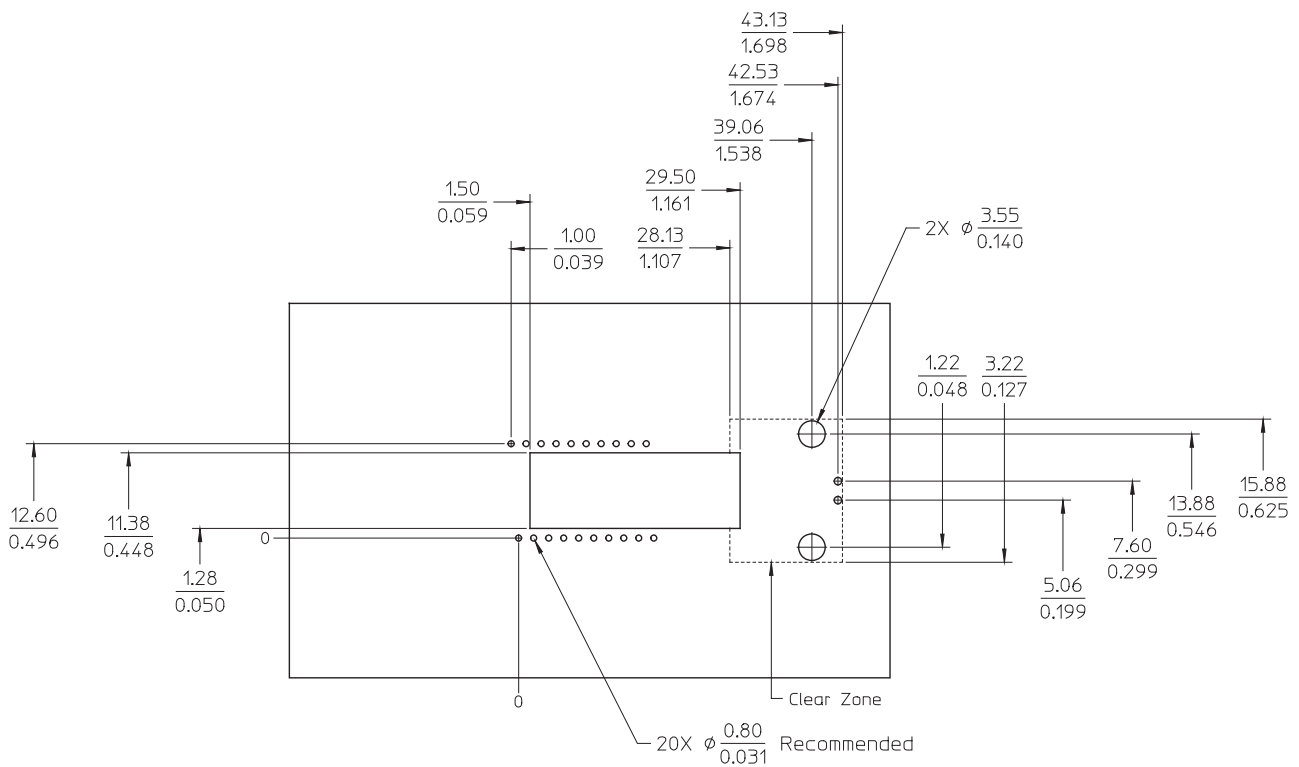
The ADNS-3080 sensor is designed for mounting on a through hole PCB, looking down. There is an aperture stop and features on the package that align to the lens.

The ADNS-2120 lens provides optics for the imaging of the surface as well as illumination of the surface at

the optimum angle. Features on the lens align it to the sensor, base plate, and clip with the LED. The lens also has a large round flange to provide a long creepage path for any ESD events that occur at the opening of the base plate.

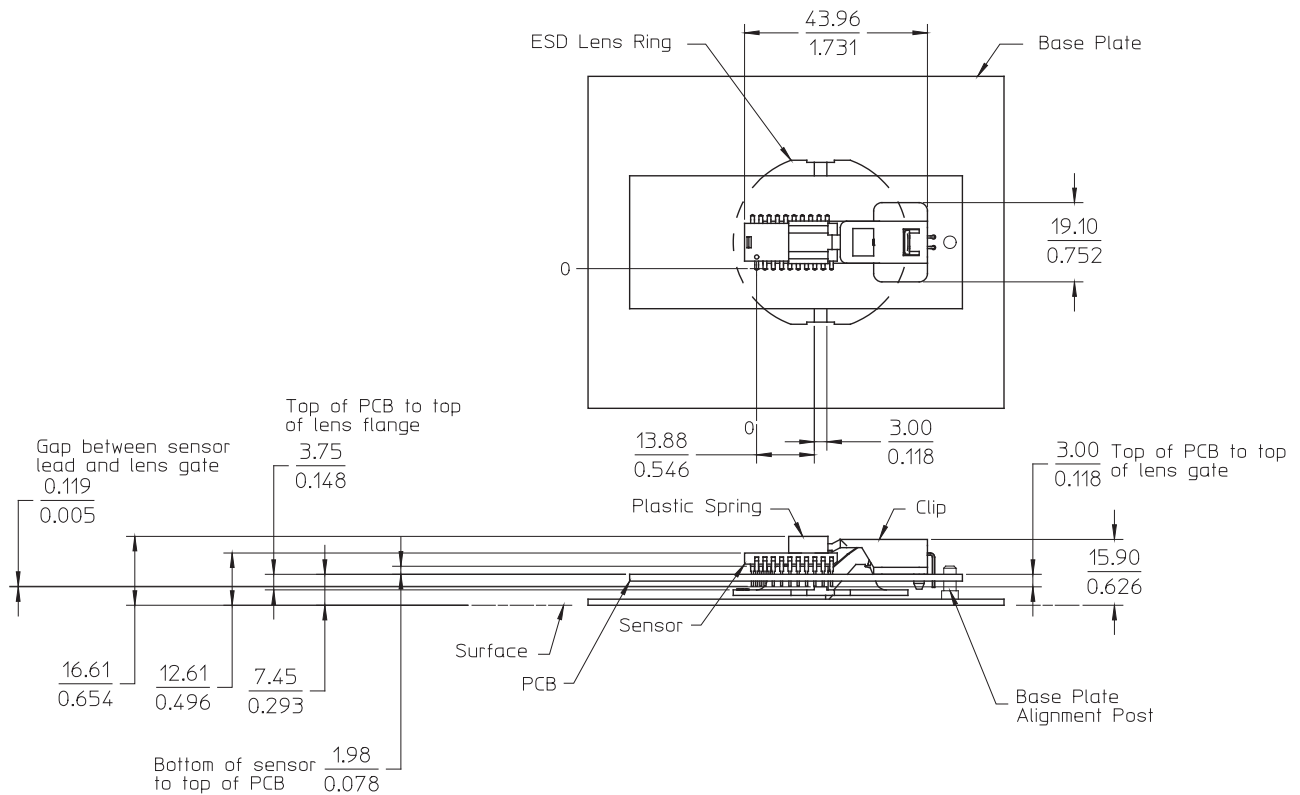
The ADNS-2220-001 clip holds the LED in relation to the lens. The LED must be inserted into the clip and the LED's leads formed prior to loading on the PCB. The clip interlocks the sensor to the lens, and through the lens to the alignment features on the base plate.

The HLMP-ED80-XX000 LED is recommended for illumination. If used with the bin table, sufficient illumination can be guaranteed.



Dimensions in millimeters / inches

Figure 3. Recommended PCB mechanical cutouts and spacing



Dimensions in Millimeters / Inches

**Figure 4. 2D Assembly drawing of ADNS-3080 (top and side view)**

*NOTE: These new Avago Technologies optical mouse sensors, lenses and clips have different physical configurations that require a different PCB mounting method to optimize the navigation performance.*

*Refer Application Notes AN 5035 for further information.*

## PCB Assembly Considerations

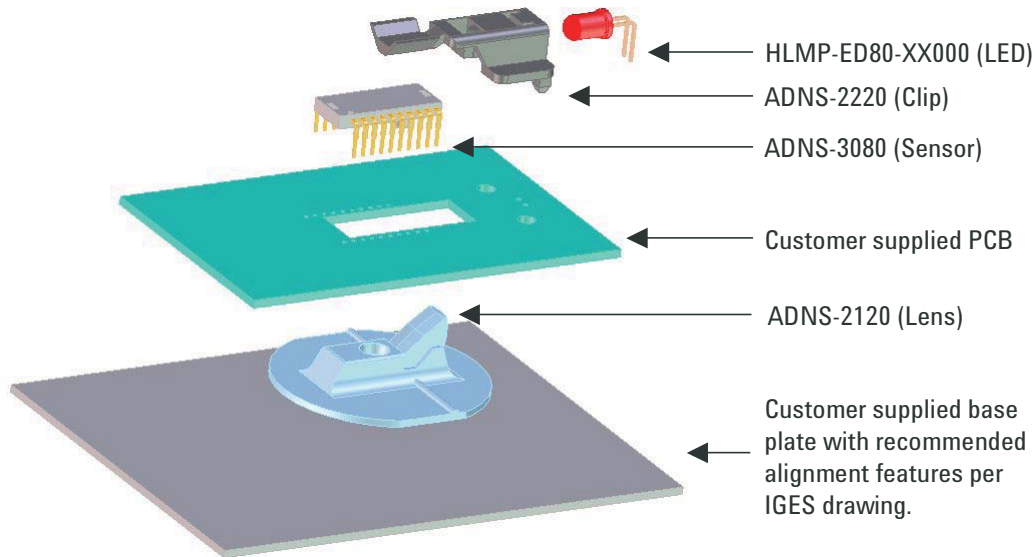


Figure 5. Exploded view drawing

1. Insert the sensor and all other electrical components into PCB.
2. Insert the LED into the assembly clip and bend the leads 90 degrees.
3. Insert the LED/clip assembly into PCB.
4. Wave Solder the entire assembly in a no-wash solder process utilizing solder fixture. The solder fixture is needed to protect the sensor during the solder process. It also sets the correct sensor-to-PCB distance as the lead shoulders do not normally rest on the PCB surface. The fixture should be designed to expose the sensor leads to solder while shielding the optical aperture from direct solder contact.
5. Place the lens onto the base plate.
6. Remove the protective kapton tape from optical aperture of the sensor. Care must be taken to keep contaminants from entering the aperture. During mouse assembly process, it is recommended that the PCB is held vertically when kapton tapes are being removed.
7. Insert PCB assembly over the lens onto the base plate aligning post to retain PCB assembly. The sensor aperture ring should self-align to the lens.
8. The optical position reference for the PCB is set by the base plate and lens. Note that the PCB motion due to button presses must be minimized to maintain optical alignment.
9. Install mouse top case. There MUST be a feature in the top case to press down

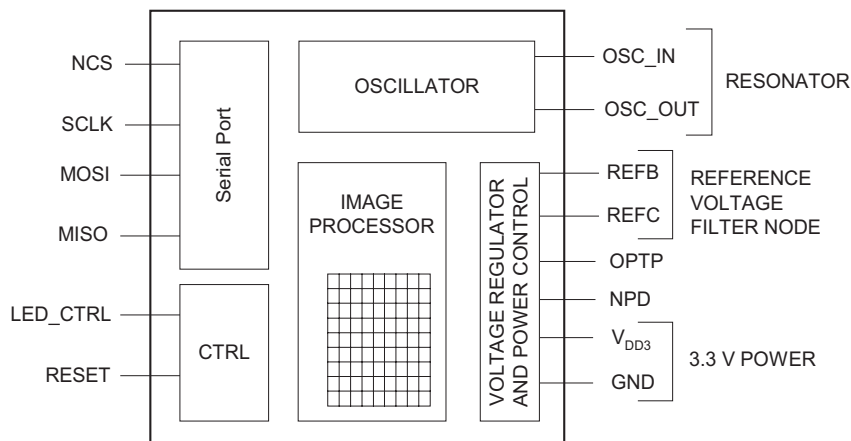


Figure 6. Block diagram of ADNS-3080 optical mouse sensor

### Design considerations for improving ESD Performance

The flange on the lens has been designed to increase the creepage and clearance distance for electrostatic discharge. The table below shows typical values assuming base plate construction per the Avago Technologies supplied IGES file and ADNS-2120 lens flange.

| Typical Distance | Millimeters |
|------------------|-------------|
| Creepage         | 16.0        |
| Clearance        | 2.1         |

For improved ESD performance, the lens flange can be sealed (i.e. glued) to the base plate. Note that the lens material is polycarbonate and therefore, cyanoacrylate based adhesives or other adhesives that may damage the lens should NOT be used.

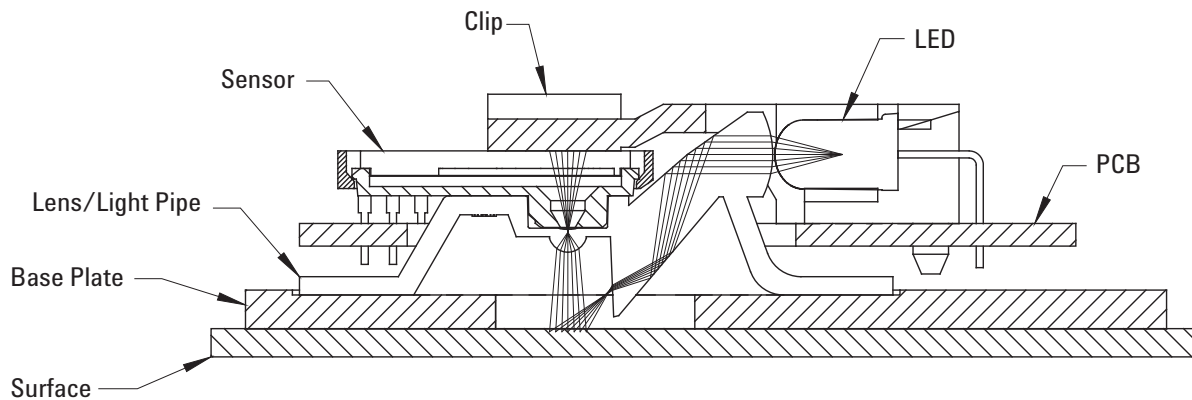
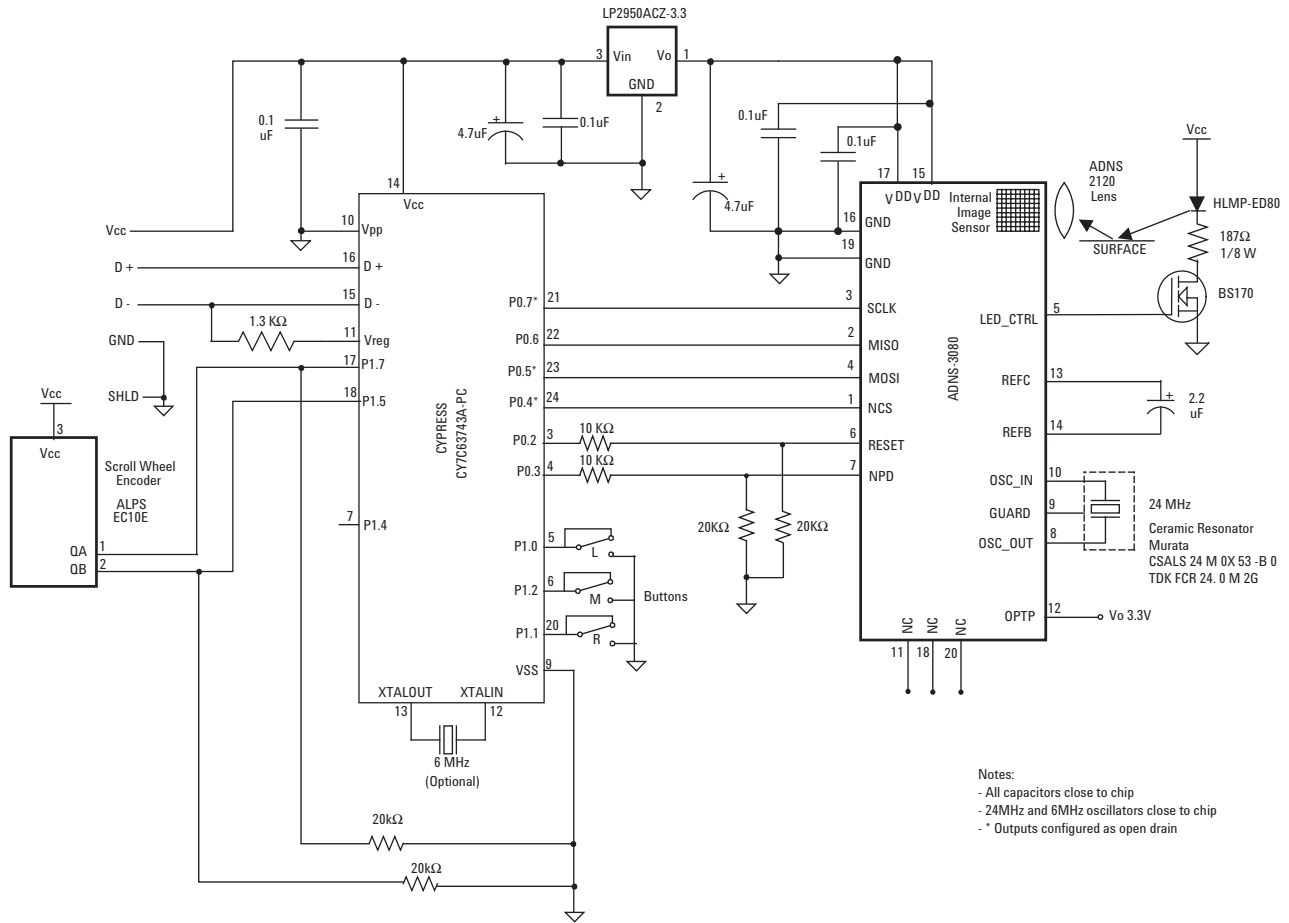


Figure 7. Cross section of PCB assembly



Notes:  
 - All capacitors close to chip  
 - 24MHz and 6MHz oscillators close to chip  
 - \* Outputs configured as open drain

Figure 8. Schematic Diagram for USB, PS/2 mouse application with ADNS-3080

Notes

- Caps for pins 15 and 17 MUST have trace lengths LESS than 5 mm to nearest ground pin.
- Pins 15 and 17 caps MUST use pin 16 GND.
- Pin 9, if used, should not be connected to PCB GND to reduce potential RF emissions.
- The 0.1 uF caps must be ceramic.
- Caps should have less than 5 nH of self inductance.
- Caps should have less than 0.2 Ω ESR.
- NC pins should not be connected to any traces.
- Surface mount parts are recommended.
- Care must be taken when interfacing a 5V microcontroller to the ADNS-3080. Serial port inputs on the sensor should be connected to open-drain outputs from the microcontroller or use an active drive level shifter. NPDP and RESET should be connected to 5V microcontroller outputs through a resistor divider or other level shifting technique.
- VDD3 and GND should have low impedance connections to the power supply.
- Capacitors connected to pin 15 and 17 should be connected to pin 16 and then to pin 19.



## Enabling the SROM

For best tracking performance, SROM is required to be loaded into ADNS-3080. This architecture enables immediate adoption of new features and improved performance algorithms. The external program is supplied by Avago Technologies as a file which may be burned into a programmable device. A micro-controller with sufficient memory may be used. On power-up and reset, the ADNS-3080 program is downloaded into volatile memory using the burst-mode procedure described in the Synchronous Serial Port section. The program size is 1986 x 8 bits.

## Regulatory Requirements

- Passes FCC B and worldwide analogous emission limits when assembled into a mouse with shielded cable and following Avago Technologies' recommendations.
- Passes IEC-1000-4-3 radiated susceptibility level when assembled into a mouse with shielded cable and following Avago Technologies' recommendations.
- Passes EN61000-4-4/IEC801-4 EFT tests when assembled into a mouse with shielded cable and following Avago Technologies' recommendations.
- UL flammability level UL94 V-0.
- Provides sufficient ESD creepage/clearance distance to avoid discharge up to 15kV when assembled into a mouse according to usage instructions above.

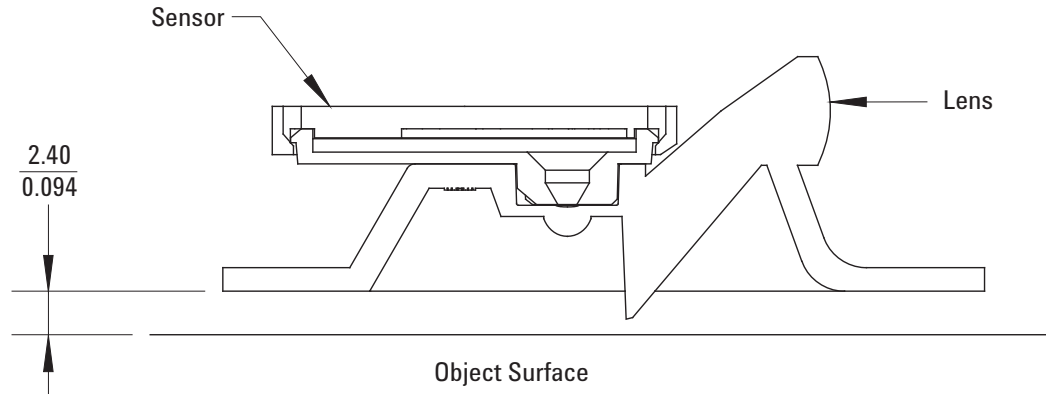


Figure 9. Distance from lens reference plane to surface

## Absolute Maximum Ratings

| Parameter             | Symbol    | Minimum | Typical | Maximum       | Units | Notes  |
|-----------------------|-----------|---------|---------|---------------|-------|--|
| Storage Temperature   | $T_S$     | -40     |         | 85            | °C    |  |
| Operating Temperature | $T_A$     | -15     |         | 55            | °C    |  |
| Lead Solder Temp      |           |         |         | 260           | °C    | For 10 seconds,<br>1.6mm below seating plane.          |
| Supply Voltage        | $V_{DD3}$ | -0.5    |         | 3.7           | V     |  |
| ESD                   |           |         |         | 2             | kV    | All pins, human body model<br>MIL 883 Method 3015      |
| Input Voltage         | $V_{IN}$  | -0.5    |         | $V_{DD3}+0.5$ | V     | NPD, NCS, MOSI, SCLK, RESET,<br>OSC_IN, OSC_OUT, REFC. |
| Output current        | $I_{out}$ |         |         | 20            | mA    | LED_CTRL, MISO   |

## Recommended Operating Conditions

| Parameter   | Symbol      | Minimum                | Typical | Maximum                          | Units             | Notes   |
|---|-------------|------------------------|---------|----------------------------------|-------------------|---|
| Operating Temperature                               | $T_A$       | 0                      |         | 40                               | °C                |   |
| Power supply voltage                                | $V_{DD3B}$  | 3.10                   | 3.30    | 3.60                             | Volts             |   |
| Power supply rise time                              | $V_{RT}$    | 1                      |         |                                  | us                | 0 to 3.0V   |
| Supply noise<br>(Sinusoidal)                        | $V_{NB}$    |                        |         | 30<br>80                         | mV p-p            | 10kHz- 300KHZ<br>300KHz-50MHz   |
| Oscillator capable<br>Frequency                     | $f_{CLK}$   | 23                     | 24      | 25                               | MHz               | Set by ceramic resonator  |
| Serial Port Clock<br>Frequency                      | $f_{SCLK}$  |                        |         | 2<br>500                         | MHz<br>kHz        | Active drive, 50% duty cycle<br>Open drain drive with pull-ups on,<br>50 pF load  |
| Resonator Impedance                                 | $X_{RES}$   |                        |         | 55                               | W                 |   |
| Distance from lens<br>reference plane to<br>surface | Z           | 2.3                    | 2.4     | 2.5                              | mm                | Results in $\pm 0.2$ mm DOF,<br>See drawing below   |
| Speed   | S           | 0                      |         | 40                               | in/sec            | @ 6469fps   |
| Acceleration  | A           |                        |         | 15                               | g                 | @ 6469fps   |
| Light level onto IC                                 | $IRR_{INC}$ | 20<br>24<br>100<br>120 |         | 6,000<br>7,200<br>6,000<br>7,200 | mW/m <sup>2</sup> | I = 639 nm, FR=1500 fps<br>I = 875 nm, FR=1500 fps<br>I = 639 nm, FR=6469 fps<br>I = 875 nm, FR=6469 fps                                    |
| Frame Rate  | FR          | 2000                   |         | 6469                             | Frames/s          | See Frame_Period register section   |
| LED Drive Current                                   | $I_{LED}$   | 10                     |         |                                  | mA                | HLMP-ED80-XX000, bin N and<br>brighter.<br>Maximum frame rate may not be<br>maintained on dark surfaces at the<br>minimum LED drive current |

## AC Electrical Specifications

Electrical Characteristics over recommended operating conditions. Typical values at 25 °C, V<sub>DD3</sub>=3.3V, fclk=24MHz.

| Parameter  | Symbol                               | Min. | Typical | Max. | Units | Notes   |
|--|--------------------------------------|------|---------|------|-------|---|
| VDD to RESET                                       | t <sub>OP</sub>                      |      |         | 250  | μs    | From VDD = 3.0V to RESET sampled  |
| Data delay after RESET                             | t <sub>PU-RESET</sub>                |      |         | 35   | ms    | From RESET falling edge to valid motion data at 2000 fps and shutter bound 8290.  |
| Input delay after reset                            | T <sub>IN-RST</sub>                  |      |         | 500  | μs    | From RESET falling edge to inputs active (NPD, MOSI, NCS, SCLK)   |
| Power Down   | t <sub>PD</sub>                      |      |         | 2.1  | ms    | From NPD falling edge to initiate the power down cycle at 500fps (t <sub>pd</sub> = 1 frame period + 100ms )  |
| Wake from NPD                                      | t <sub>PUPD</sub>                    |      |         | 75   | ms    | From NPD rising edge to valid motion data at 2000 fps and shutter bound 8290. Max assumes surface change while NPD is low.                                    |
| Data delay after NPD                               | t <sub>COMPUTE</sub>                 |      |         | 3.1  | ms    | From NPD rising edge to all registers contain data from new images at 2000fps (see Figure 10) .   |
| RESET pulse width                                  | t <sub>PW-RESET</sub>                | 10   |         |      | μs    |   |
| MISO rise time                                     | t <sub>r-MISO</sub>                  |      | 40      | 200  | ns    | C <sub>L</sub> = 50pF   |
| MISO fall time                                     | t <sub>f-MISO</sub>                  |      | 40      | 200  | ns    | C <sub>L</sub> = 50pF   |
| MISO delay after SCLK                              | t <sub>DLY-MISO</sub>                |      |         | 120  | ns    | From SCLK falling edge to MISO data valid, no load conditions   |
| MISO hold time                                     | t <sub>hold-MISO</sub>               | 250  |         |      | ns    | Data held until next falling SCLK edge  |
| MOSI hold time                                     | t <sub>hold-MOSI</sub>               | 200  |         |      | ns    | Amount of time data is valid after SCLK rising edge   |
| MOSI setup time                                    | t <sub>setup-MOSI</sub>              | 120  |         |      | ns    | From data valid to SCLK rising edge   |
| SPI time between write commands                    | t <sub>SWW</sub>                     | 50   |         |      | μs    | From rising SCLK for last bit of the first data byte, to rising SCLK for last bit of the second data byte.  |
| SPI time between write and read commands           | t <sub>SWR</sub>                     | 50   |         |      | μs    | From rising SCLK for last bit of the first data byte, to rising SCLK for last bit of the second address byte.   |
| SPI time between read and subsequent commands      | t <sub>SRW</sub><br>t <sub>SRR</sub> | 250  |         |      | ns    | From rising SCLK for last bit of the first data byte, to falling SCLK for first bit of the second address byte.   |
| SPI read address-data delay                        | t <sub>SRAD</sub>                    | 50   |         |      | μs    | From rising SCLK for last bit of the address byte, to falling SCLK for first bit of data being read. All registers except Motion & Motion_Burst               |
| SPI motion read address-data delay                 | t <sub>SRAD-MOT</sub>                | 75   |         |      | μs    | From rising SCLK for last bit of the address byte, to falling SCLK for first bit of data being read. Applies to 0x02 Motion, and 0x50 Motion_Burst, registers |
| NCS to SCLK active                                 | t <sub>NCS-SCLK</sub>                | 120  |         |      | ns    | From NCS falling edge to first SCLK rising edge   |
| SCLK to NCS inactive                               | t <sub>SCLK-NCS</sub>                | 120  |         |      | ns    | From last SCLK falling edge to NCS rising edge, for valid MISO data transfer  |
| NCS to MISO high-Z                                 | t <sub>NCS-MISO</sub>                |      |         | 250  | ns    | From NCS rising edge to MISO high-Z state   |
| SROM download and frame capture byte-to-byte delay | t <sub>LOAD</sub>                    | 10   |         |      | μs    | (see Figure 23 and 24)  |
| NCS to burst mode exit                             | t <sub>BEXIT</sub>                   | 4    |         |      | μs    | Time NCS must be held high to exit burst mode   |
| Transient Supply Current                           | I <sub>DDT</sub>                     |      |         | 85   | mA    | Max supply current during a V <sub>DD3</sub> ramp from 0 to 3.6V  |

## DC Electrical Specifications

Electrical Characteristics over recommended operating conditions. Typical values at 25 °C,  $V_{DD3}=3.3V$ ,  $f_{clk}=24MHz$ .

| Parameter                        | Symbol        | Minimum         | Typical | Maximum  | Units   | Notes  |
|----------------------------------|---------------|-----------------|---------|----------|---------|--|
| DC Supply Current                | $I_{DD\_AVG}$ |                 |         | 52       | mA      | DC average at 6469 fps.<br>No DC load on LED_CTRL, MISO. |
| Power Down Supply Current        | $I_{DDPD}$    |                 | 5       | 90       | $\mu A$ | NPD=GND; SCLK, MOSI, NCS=GND or $V_{DD3}$ ;<br>RESET=GND |
| Input Low Voltage                | $V_{IL}$      |                 |         | 0.8      | V       | SCLK, MOSI, NPD, NCS, RESE                               |
| Input High Voltage               | $V_{IH}$      | $0.7 * V_{DD3}$ |         |          | V       | SCLK, MOSI, NPD, NCS, RESET                              |
| Input hysteresis                 | $V_{I\_HYS}$  |                 | 200     |          | mV      | SCLK, MOSI, NPD, NCS, RESET                              |
| Input current, pull-up disabled  | $I_{IH\_DPU}$ |                 | 0       | $\pm 10$ | $\mu A$ | $V_{in}=0.8*V_{DD3}$ , SCLK, MOSI, NCS                   |
| Input current, CMOS inputs       | $I_{IH}$      |                 | 0       | $\pm 10$ | $\mu A$ | NPD, RESET, $V_{in}=0.8*V_{DD3}$                         |
| Output current, pulled-up inputs | $I_{OH\_PU}$  | 150             | 300     | 600      | $\mu A$ | $V_{in}=0.2V$ , SCLK, MOSI, NCS                          |
| Output Low Voltage LED_CTRL      | $V_{OL\_LED}$ |                 |         | 0.5      | V       | $I_{out}=2mA$ , LED_CTRL                                 |
| Output High voltage, LED_CTRL    | $V_{OH\_LED}$ | $0.8*V_{DD3}$   |         |          | V       | $I_{out}=-2mA$ , LED_CTRL                                |
| Output Low Voltage, MISO         | $V_{OL}$      |                 |         | 0.5      | V       | $I_{out}=2mA$ , MISO                                     |
| Output High Voltage, MISO        | $V_{OH}$      | $0.8*V_{DD3}$   |         |          | V       | $I_{out}=-2mA$ , MISO                                    |
| Input Capacitance                | $C_{IN}$      |                 | 14-22   |          | pF      | OSC_IN, OSC_OUT  |

### Detail of NPD rising edge timing

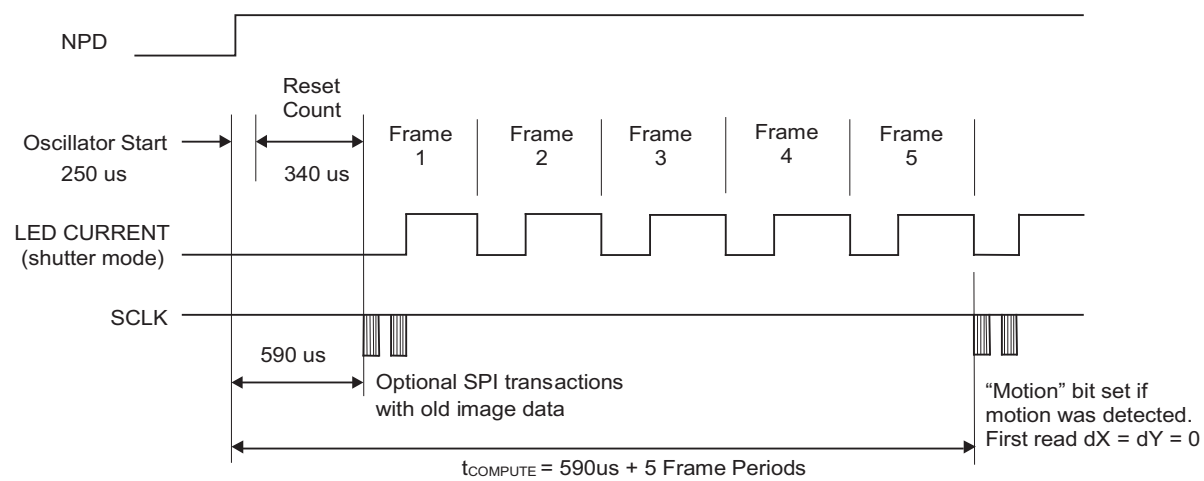


Figure 10. NPD Rising Edge Timing Detail

## Typical Performance Characteristics

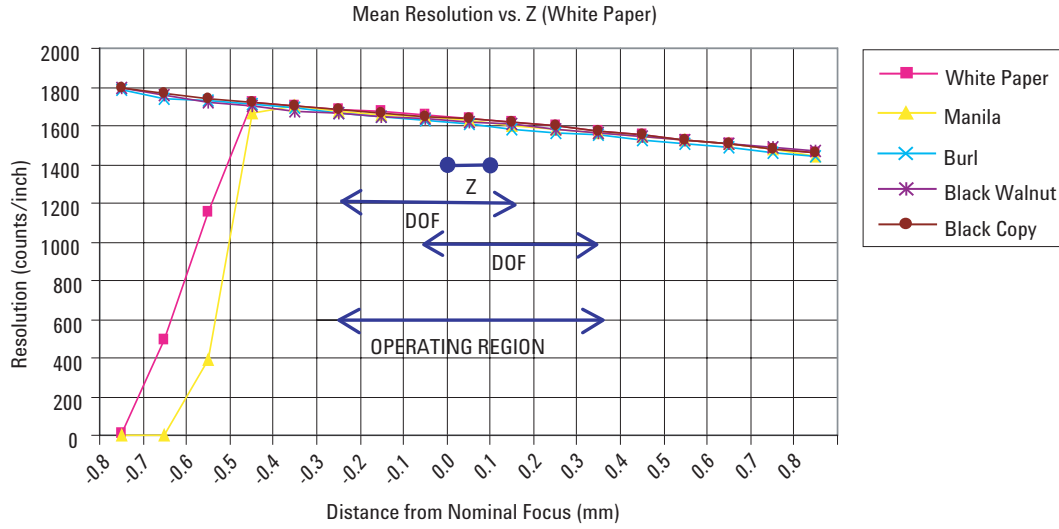


Figure 11. Mean Resolution vs. Z (White Paper)

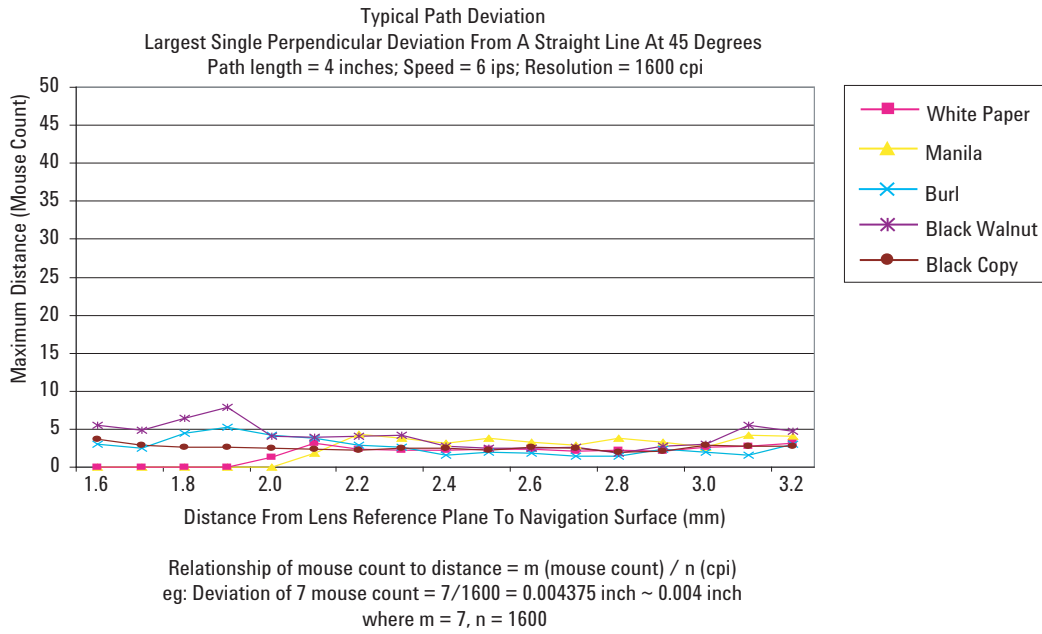
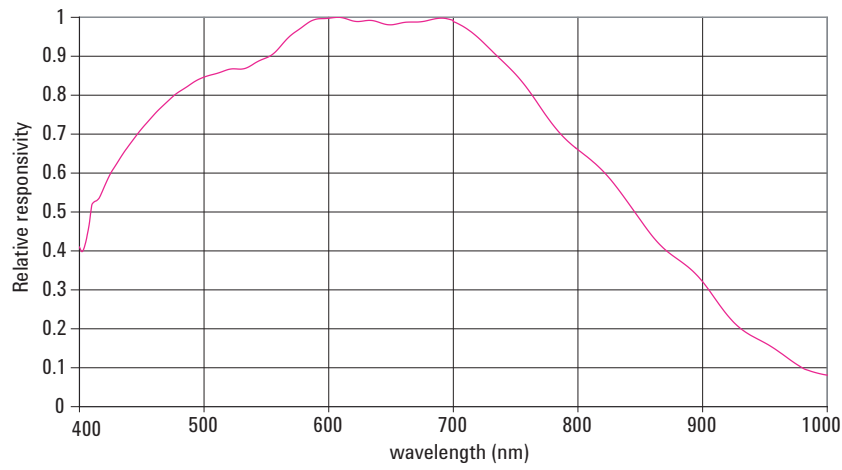
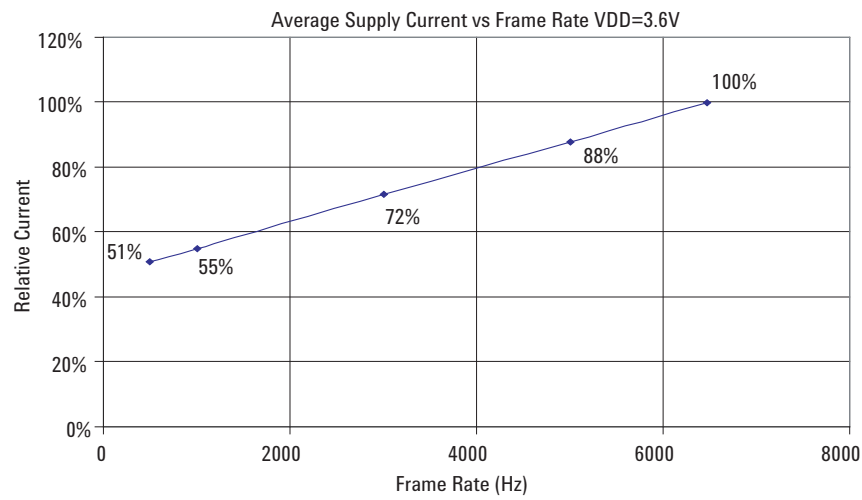


Figure 12. Average error vs. Distance (mm)



**Figure 13. Relative responsivity**



**Figure 14. I<sub>dd</sub> vs. Frame Rate**

## Synchronous Serial Port

The synchronous serial port is used to set and read parameters in the ADNS-3080, and to read out the motion information. The serial port is also used to load SROM data into the ADNS-3080.

The port is a four-wire, serial port. The host micro-controller always initiates communication; the ADNS-3080 never initiates data transfers. The serial port cannot be activated while the chip is in power down mode (NPD low) or reset (RESET high). SCLK, MOSI, and NCS may be driven directly by a 3.3V output from a micro-controller, or they may be placed in an open drain configuration by enabling on-chip pull-up current sources. The open drain drive allows the use of a 5V micro-controller without any level shifting components. The port pins may be shared with other SPI slave devices. When the NCS pin is high, the inputs are ignored and the output is tri-stated.

The lines which comprise the SPI port are:

SCLK: Clock input. It is always generated by the master (the micro-controller).

MOSI: Input data (Master Out/Slave In).

MISO: Output data (Master In/Slave Out).

NCS: Chip select input (active low).

NCS needs to be low to activate the serial port; otherwise, MISO will be high-Z, and MOSI & SCLK will be ignored. NCS can also be used to reset the serial port in case of an error.

## Chip Select Operation

The serial port is activated after NCS goes low. If NCS is raised during a transaction, the entire transaction is aborted and the serial port will be reset. This is true for all transactions including SROM download. After a transaction is aborted, the normal address-to-data or transaction-to-transaction delay is still required before beginning the next transaction. To improve communication reliability, all serial transactions should be framed by NCS. In other words, the port should not remain enabled during periods of non-use because ESD and EFT/B events could be interpreted as serial communication and put the chip into an unknown state. In addition, NCS must be raised after each burst-mode transaction is complete to terminate burst-mode. The port is not available for further use until burst-mode is terminated.

## Write Operation

Write operation, defined as data going from the micro-controller to the ADNS-3080, is always initiated by the micro-controller and consists of two bytes. The first byte contains the address (seven bits) and has a "1" as its MSB to indicate data direction. The second byte contains the data. The ADNS-3080 reads MOSI on rising edges of SCLK.

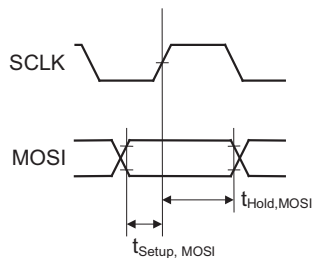


Figure 15. MOSI setup and hold time

## Read Operation

A read operation, defined as data going from the ADNS-3080 to the micro-controller, is always initiated by the micro-controller and consists of two bytes. The first byte contains the address, is sent by the micro-controller over MOSI, and has a "0" as its MSB to indicate data direction. The second byte contains the data and is driven by the ADNS-3080 over MISO. The sensor outputs MISO bits on falling edges of SCLK and samples MOSI bits on every rising edge of SCLK.

### NOTE:

The 250 ns minimum high state of SCLK is also the minimum MISO data hold time of the ADNS-3080. Since the falling edge of SCLK is actually the start of the next read or write command, the ADNS-3080 will hold the state of data on MISO until the falling edge of SCLK.

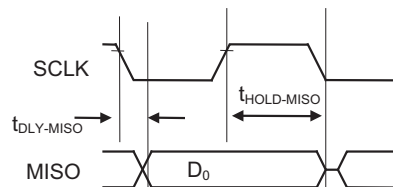


Figure 18. MISO delay and hold time

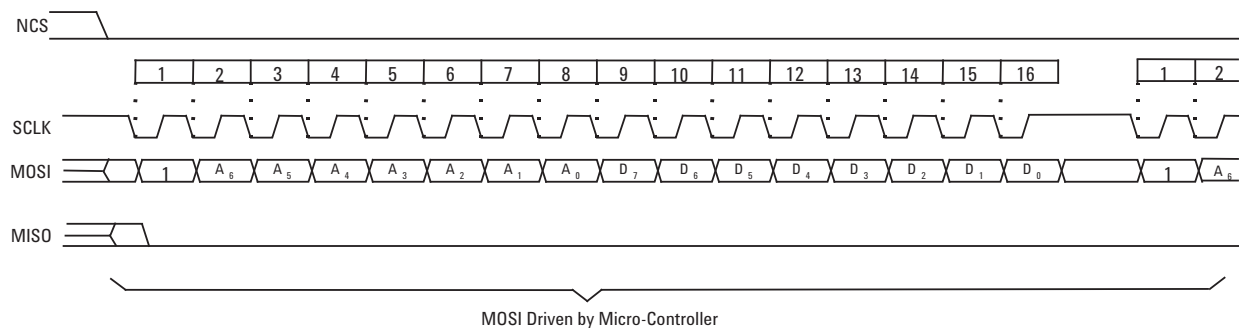


Figure 16. Write Operation

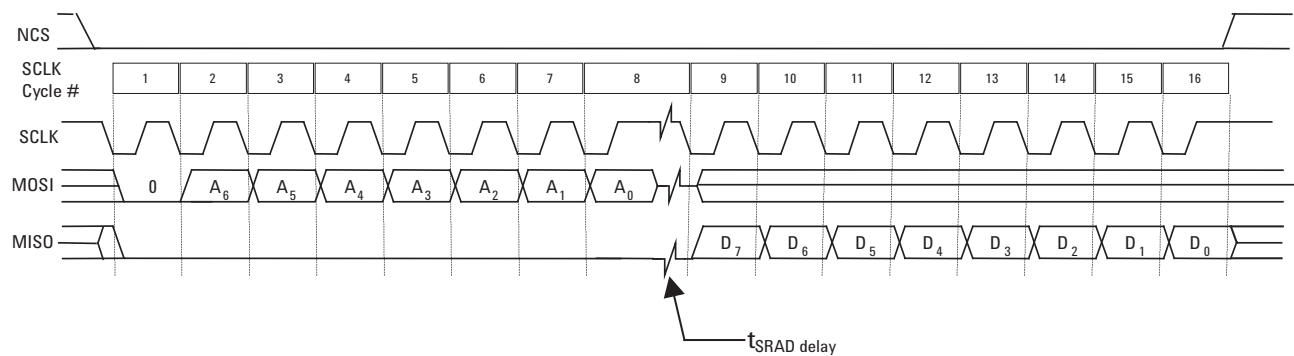
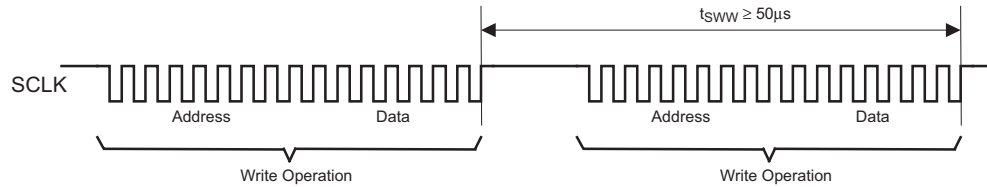


Figure 17. Read operation



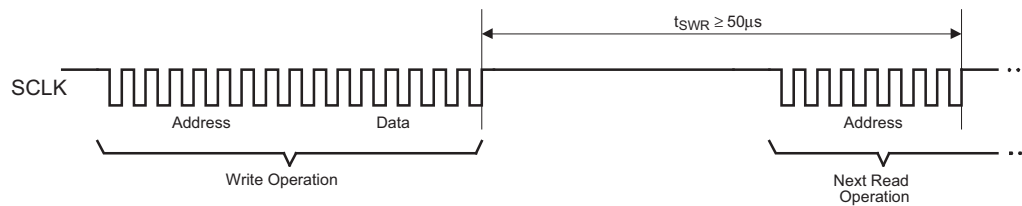
### Required timing between Read and Write Commands ( $t_{sxx}$ )

There are minimum timing requirements between read and write commands on the serial port.



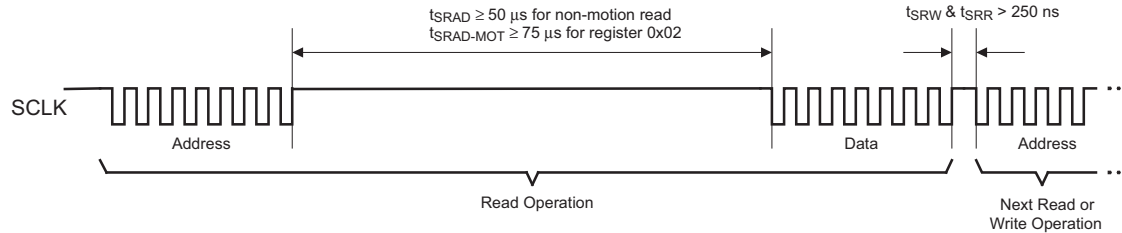
**Figure 19. Timing between two write commands**

If the rising edge of the SCLK for the last data bit of the second write command occurs before the 50 microsecond required delay, then the first write command may not complete correctly.



**Figure 20. Timing between write and read commands**

If the rising edge of SCLK for the last address bit of the read command occurs before the 50 microsecond required delay, the write command may not complete correctly.



**Figure 21. Timing between read and either write or subsequent read commands**

The falling edge of SCLK for the first address bit of either the read or write command must be at least 250 ns after the last SCLK rising edge of the last data bit of the previous read operation. In addition, during a read operation SCLK should be delayed after the last address bit to ensure that the ADNS-3080 has time to prepare the requested data.

## Burst Mode Operation

Burst mode is a special serial port operation mode which may be used to reduce the serial transaction time for three predefined operations: motion read and SROM download and frame capture. The speed improvement is achieved by continuous data clocking to or from multiple registers without the need to specify the register address, and by not requiring the normal delay period between data bytes.

## Motion Read

This mode is activated by reading the Motion\_Burst register. The ADNS-3080 will respond with the contents of the Motion, Delta\_X, Delta\_Y, SQUAL, Shutter\_Upper, Shutter\_Lower and Maximum\_Pixel registers in that order. After sending the register address, the micro-controller must wait  $t_{SRAD-MOT}$  and then begin reading data. All 56 data bits can be read with no delay between bytes by driving SCLK at the normal rate. The data are latched into the output buffer after the last address bit is received. After the burst transmission is complete, the micro-controller must raise the NCS line for at least  $t_{BEXIT}$  to terminate burst mode. The serial port is not available for use until it is reset with NCS, even for a second burst transmission.

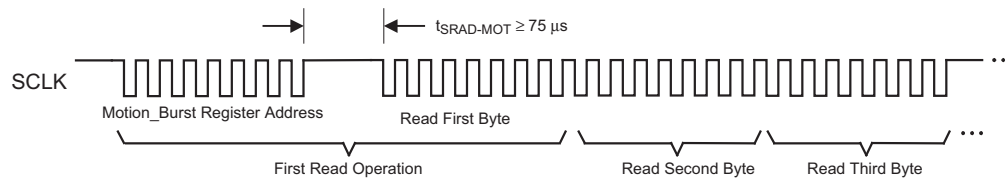


Figure 22. Motion burst timing

## SROM Download

This function is used to load the Avago Technologies-supplied firmware file contents into the ADNS-3080. The firmware file is an ASCII text file with each 2-character byte (hexadecimal representation) on a single line.

This mode is activated by the following steps:

1. Perform hardware reset by toggling the RESET pin
2. Write 0x44 to register 0x20
3. Write 0x07 to register 0x23
4. Write 0x88 to register 0x24
5. Wait at least 1 frame period
6. Write 0x18 to register 0x14 (SROM\_Enable register)
7. Begin burst mode write of data file to register 0x60 (SROM\_Load register)

After the first data byte is complete, the SROM or micro-controller must write subsequent bytes by presenting the data on the MOSI line and driving SCLK at the normal rate. A delay of at least  $t_{LOAD}$  must exist between data

bytes as shown. After the download is complete, the micro-controller must raise the NCS line for at least  $t_{BEXIT}$  to terminate burst mode. The serial port is not available for use until it is reset with NCS, even for a second burst transmission.

Avago Technologies recommends reading the SROM\_ID register to verify that the download was successful. In addition, a self-test may be executed, which performs a CRC on the SROM contents and reports the results in a register. The test is initiated by writing a particular value to the SROM\_Enable register; the result is placed in the Data\_Out register. See those register descriptions for more details.

Avago Technologies provides the data file for download; the file size is 1986 data bytes. The chip will ignore any additional bytes written to the SROM\_Load register after the SROM file.

SROM file is now available for download at Avago Technologies' website.

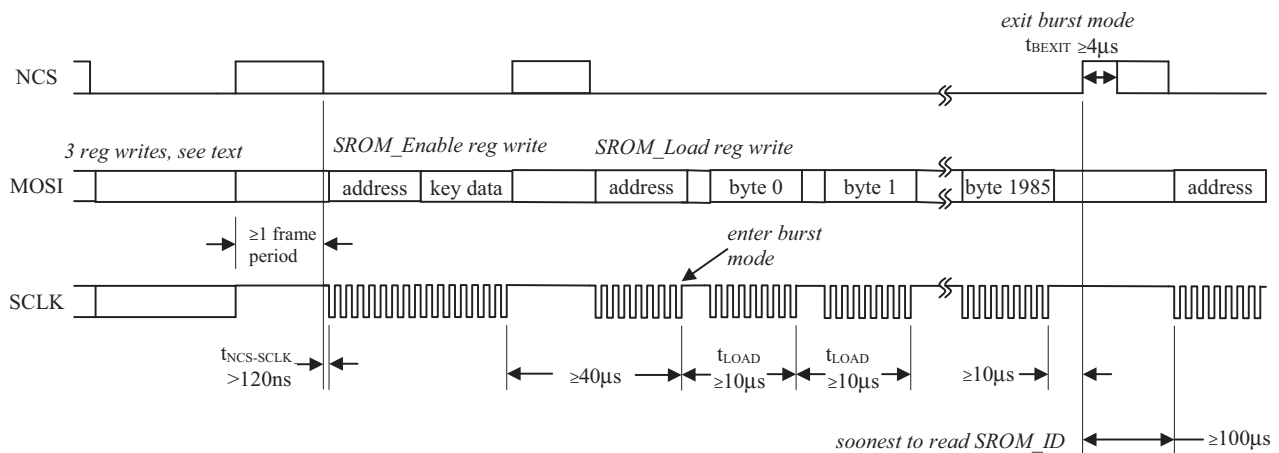


Figure 23. SROM download burst mode

## Frame Capture

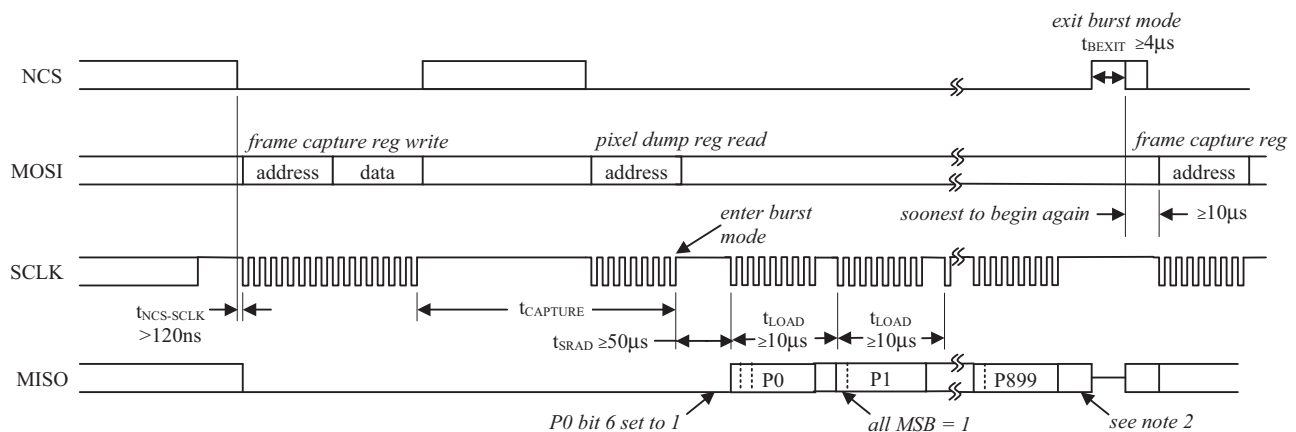
This is a fast way to download a full array of pixel values from a single frame. This mode disables navigation and overwrites any downloaded firmware. A hardware reset is required to restore navigation, and the firmware must be reloaded afterwards if required.

To trigger the capture, write to the Frame\_Capture register. The next available complete 1 2/3 frames (1536 values) will be stored to memory. The data are retrieved by reading the Pixel\_Burst register once using the normal read method, after which the remaining bytes are clocked out by driving SCLK at the normal rate. The byte time must be at least  $t_{LOAD}$ . If the Pixel\_Burst register is read before the data is ready, it will return all zeros.

To read a single frame, read a total of 900 bytes. The next 636 bytes will be approximately 2/3 of the next frame. The first pixel of the first frame (1st read) has bit 6 set to 1 as a start-of-frame marker. The first pixel of the second partial frame (901st read) will also have bit 6 set to 1. All other bytes have bit 6 set to zero. The MSB of all bytes is set to 1. If the Pixel\_Burst register is read past the end of the data (1537 reads and on), the data returned will be zeros.

After the download is complete, the micro-controller must raise the NCS line for at least  $t_{BEXIT}$  to terminate burst mode. The read may be aborted at any time by raising NCS.

Alternatively, the frame data can also be read one byte at a time from the Frame\_Capture register. See the register description for more information.



Notes:

1. MSB = 1 for all bytes. Bit 6 = 0 for all bytes except pixel 0 of both frames which has bit 6 = 1 for use as a frame marker.
2. Reading beyond pixel 899 will return the first pixel of the second partial frame.
3.  $t_{CAPTURE} = 10\mu s + 3$  frame periods.
4. This figure illustrates reading a single complete frame of 900 pixels. An additional 636 pixels from the next frame are available.

**Figure 24. Frame capture burst mode timing**

The pixel output order as related to the surface is shown below.

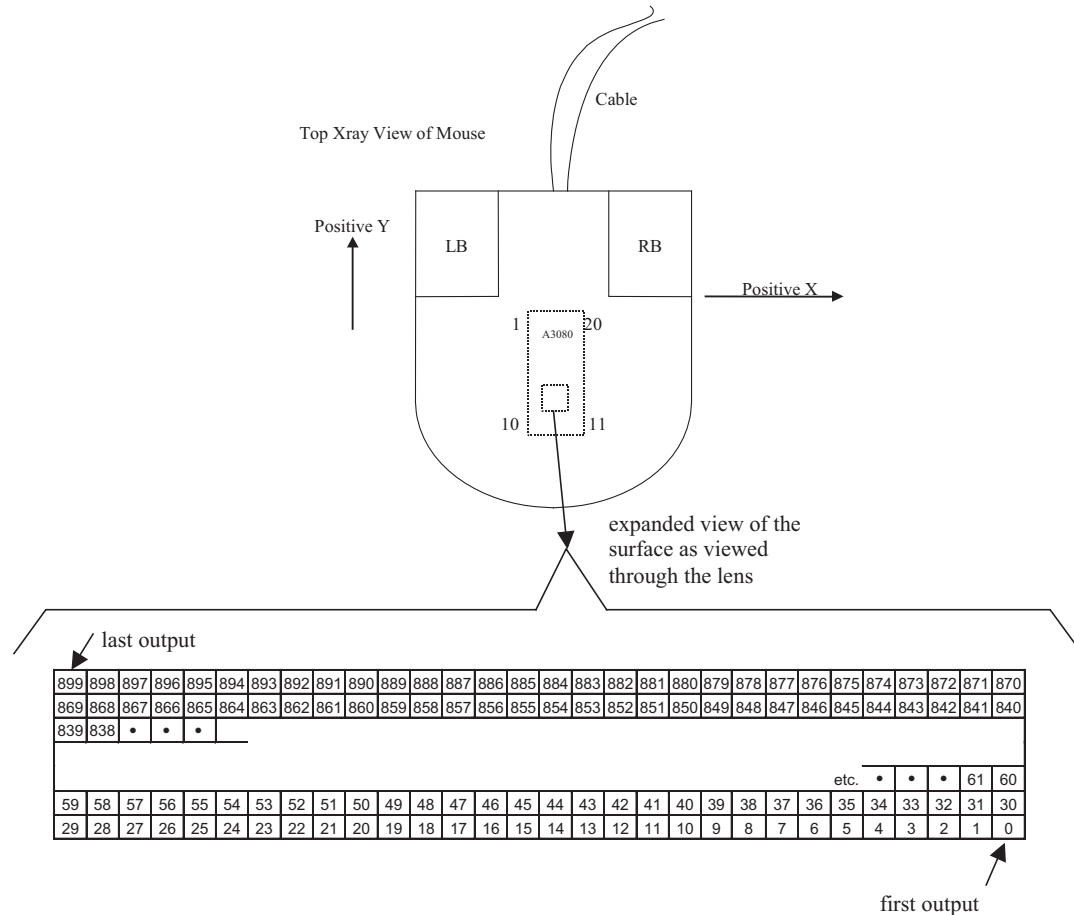


Figure 25. Pixel address map (surface referenced)

### Error detection and recovery

1. The ADNS-3080 and the micro-controller might get out of synchronization due to ESD events, power supply droops or micro-controller firmware flaws. In such a case, the micro-controller should pulse NCS high for at least 1 ms. The ADNS-3080 will reset the serial port (but not the control registers) and will be prepared for the beginning of a new transmission after the normal transaction delay.
2. Invalid addresses: Writing to an invalid address will have no effect. Reading from an invalid address will return all zeros.
3. Termination of a transmission by the micro-controller may sometimes be required (for example, due to a USB suspend interrupt during a read operation). To accomplish this the micro-controller should raise NCS. The ADNS-3080 will not write to any register and will reset the serial port (but not the control registers) and be prepared for the beginning of future transmissions after NCS goes low. The normal delays between reads or writes ( $t_{SWW}$ ,  $t_{SWr}$ ,  $t_{SRAD}$ ,  $t_{SRAD-mot}$ ) are still required after aborted transmissions.
4. The micro-controller can verify success of write operations by issuing a read command to the same address and comparing written data to read data.
5. The micro-controller can verify the synchronization of the serial port by periodically reading the product ID and inverse product ID registers.
6. The microcontroller can read the SROM\_ID register to verify that the sensor is running downloaded SROM code. ESD or similar noise events may cause the sensor to revert to native ROM execution. If this should happen, pulse RESET and reload the SROM instructions.

## Notes on Power-up and the serial port

### Reset Circuit

The ADNS-3080 does not perform an internal power up self-reset. The reset pin must be raised and lowered to reset the chip. This should be done every time power is applied. During power-up there will be a period of time after the power supply is high but before any clocks are available. The table below shows the state of the various pins during power-up and reset when the RESET pin is driven high by a micro-controller.

### Power Down Circuit

The following table lists the pin states during power down.

The chip is put into the power down (PD) mode by lowering the NPD input. When in PD mode, the oscillator is stopped but all register contents are retained. To achieve the lowest current state, all inputs must be held externally within 200mV of a rail, either ground or  $V_{DD3}$ . The chip outputs are driven low or hi-Z during PD to prevent current consumption by an external load.

### LED Drive Mode

The LED has 2 modes of operation: DC and Shutter. In DC mode it is on at all times the chip is powered except when in the power down mode via the NPD pin. In shutter mode the LED is on only during the portion of the frame that light is required. The LED\_MODE bit in the Configuration\_bits register sets the LED mode.

#### State of Signal Pins After VDD is Valid

| Pin         | Before Reset             | During Reset             | After Reset           |
|-------------|--------------------------|--------------------------|-----------------------|
| SPI pullups | Undefined                | Off                      | On (default)          |
| NCS         | Hi-Z control functional  | Hi-Z control functional  | Functional            |
| MISO        | Driven or hi-Z (per NCS) | Driven or hi-Z (per NCS) | Low or hi-Z (per NCS) |
| SCLK        | Undefined                | Ignored                  | Functional            |
| MOSI        | Undefined                | Ignored                  | Functional            |
| LED_CTRL    | Undefined                | Low                      | High                  |
| RESET       | Functional               | High (externally driven) | Functional            |
| NPD         | Undefined                | Ignored                  | Functional            |

#### State of Signal Pins During Power Down

| Pin         | NPD low                 | After wake from PD   |
|-------------|-------------------------|----------------------|
| SPI pullups | off                     | pre-PD state         |
| NCS         | hi-Z control functional | functional           |
| MISO        | low or hi-Z (per NCS)   | pre-PD state or hi-Z |
| SCLK        | ignored                 | functional           |
| MOSI        | ignored                 | functional           |
| LED_CTRL    | low                     | high                 |
| RESET       | functional              | functional           |
| NPD         | low (driven externally) | functional           |
| REFC        | $V_{DD3}$               | REFC                 |
| OSC_IN      | low                     | OSC_IN               |
| OSC_OUT     | high                    | OSC_OUT              |

## Registers

The ADNS-3080 registers are accessible via the serial port. The registers are used to read motion data and status as well as to set the device configuration.

| Address   | Register                     | Read/Write | SROM Default Value |
|-----------|------------------------------|------------|--------------------|
| 0x00      | Product_ID                   | R          | 0x17               |
| 0x01      | Revision_ID                  | R          | 0xNN               |
| 0x02      | Motion                       | R          | 0x00               |
| 0x03      | Delta_X                      | R          | 0x00               |
| 0x04      | Delta_Y                      | R          | 0x00               |
| 0x05      | SQUAL                        | R          | 0x00               |
| 0x06      | Pixel_Sum                    | R          | 0x00               |
| 0x07      | Maximum_Pixel                | R          | 0x00               |
| 0x08      | Reserved                     |            |                    |
| 0x09      | Reserved                     |            |                    |
| 0x0a      | Configuration_bits           | R/W        | 0x09               |
| 0x0b      | Extended_Config              | R/W        | 0x00               |
| 0x0c      | Data_Out_Lower               | R          | Any                |
| 0x0d      | Data_Out_Upper               | R          | Any                |
| 0x0e      | Shutter_Lower                | R          | 0x85               |
| 0x0f      | Shutter_Upper                | R          | 0x00               |
| 0x10      | Frame_Period_Lower           | R          | Any                |
| 0x11      | Frame_Period_Upper           | R          | Any                |
| 0x12      | Motion_Clear                 | W          | Any                |
| 0x13      | Frame_Capture                | R/W        | 0x00               |
| 0x14      | SROM_Enable                  | W          | 0x00               |
| 0x15      | Reserved                     |            |                    |
| 0x16      | Reserved                     |            |                    |
| 0x17      | Reserved                     |            |                    |
| 0x18      | Reserved                     |            |                    |
| 0x19      | Frame_Period_Max_Bound_Lower | R/W        | 0xE0               |
| 0x1a      | Frame_Period_Max_Bound_Upper | R/W        | 0x2E               |
| 0x1b      | Frame_Period_Min_Bound_Lower | R/W        | 0x7E               |
| 0x1c      | Frame_Period_Min_Bound_Upper | R/W        | 0x0E               |
| 0x1d      | Shutter_Max_Bound_Lower      | R/W        | 0x00               |
| 0x1e      | Shutter_Max_Bound_Upper      | R/W        | 0x20               |
| 0x1f      | SROM_ID                      | R          | 0x00               |
| 0x20-0x3c | Reserved                     |            |                    |
| 0x3d      | Observation                  | R/W        | 0x00               |
| 0x3e      | Reserved                     |            |                    |
| 0x3f      | Inverse Product ID           | R          | 0xF8               |
| 0x40      | Pixel_Burst                  | R          | 0x00               |
| 0x50      | Motion_Burst                 | R          | 0x00               |
| 0x60      | SROM_Load                    | W          | Any                |

|                   |                  |                      |                  |                  |                  |                  |                  |                  |
|-------------------|------------------|----------------------|------------------|------------------|------------------|------------------|------------------|------------------|
| <b>Product_ID</b> |                  | <b>Address: 0x00</b> |                  |                  |                  |                  |                  |                  |
| Access: Read      |                  | Reset Value: 0x17    |                  |                  |                  |                  |                  |                  |
| Bit               | 7                | 6                    | 5                | 4                | 3                | 2                | 1                | 0                |
| Field             | PID <sub>7</sub> | PID <sub>6</sub>     | PID <sub>5</sub> | PID <sub>4</sub> | PID <sub>3</sub> | PID <sub>2</sub> | PID <sub>1</sub> | PID <sub>0</sub> |

Data Type: 8-Bit unsigned integer

USAGE: This register contains a unique identification assigned to the ADNS-3080. The value in this register does not change; it can be used to verify that the serial communications link is functional.

|                    |                  |                      |                  |                  |                  |                  |                  |                  |
|--------------------|------------------|----------------------|------------------|------------------|------------------|------------------|------------------|------------------|
| <b>Revision_ID</b> |                  | <b>Address: 0x01</b> |                  |                  |                  |                  |                  |                  |
| Access: Read       |                  | Reset Value: 0xNN    |                  |                  |                  |                  |                  |                  |
| Bit                | 7                | 6                    | 5                | 4                | 3                | 2                | 1                | 0                |
| Field              | RID <sub>7</sub> | RID <sub>6</sub>     | RID <sub>5</sub> | RID <sub>4</sub> | RID <sub>3</sub> | RID <sub>2</sub> | RID <sub>1</sub> | RID <sub>0</sub> |

Data Type: 8-Bit unsigned integer.

USAGE: This register contains the IC revision. It is subject to change when new IC versions are released.

NOTE: The downloaded SROM firmware revision is a separate value and is available in the SROM\_ID register.

|               |     |                      |          |     |          |          |          |     |
|---------------|-----|----------------------|----------|-----|----------|----------|----------|-----|
| <b>Motion</b> |     | <b>Address: 0x02</b> |          |     |          |          |          |     |
| Access: Read  |     | Reset Value: 0x00    |          |     |          |          |          |     |
| Bit           | 7   | 6                    | 5        | 4   | 3        | 2        | 1        | 0   |
| Field         | MOT | Reserved             | Reserved | OVF | Reserved | Reserved | Reserved | RES |

Data Type: Bit field.

USAGE: Register 0x02 allows the user to determine if motion has occurred since the last time it was read. If so, then the user should read registers 0x03 and 0x04 to get the accumulated motion. It also tells if the motion buffers have overflowed, and the current resolution setting.

| Field Name | Description  |
|------------|--|
| MOT        | Motion since last report or PD<br><b>0 = No motion</b><br>1 = Motion occurred, data ready for reading in Delta_X and Delta_Y registers |
| Reserved   | Reserved   |
| Reserved   | Reserved   |
| OVF        | Motion overflow, Delta_Y and/or Delta_X buffer has overflowed since last report<br><b>0 = no overflow</b><br>1 = Overflow has occurred |
| Reserved   | Reserved   |
| Reserved   | Reserved   |
| Reserved   | Reserved   |
| RES        | Resolution in counts per inch<br><b>0 = 400</b><br>1 = 1600  |



Notes for Motion:

1. Reading this register freezes the Delta\_X and Delta\_Y register values. Read this register before reading the Delta\_X and Delta\_Y registers. If Delta\_X and Delta\_Y are not read before the motion register is read a second time, the data in Delta\_X and Delta\_Y will be lost.
2. Avago Technologies RECOMMENDS that registers 0x02, 0x03 and 0x04 be read sequentially. See Motion burst mode also.
3. Internal buffers can accumulate more than eight bits of motion for X or Y. If either one of the internal buffers overflows, then absolute path data is lost and the OVF bit is set. This bit is cleared once some motion has been read from the Delta\_X and Delta\_Y registers, and if the buffers are not at full scale. Since more data is present in the buffers, the cycle of reading the Motion, Delta\_X and Delta\_Y registers should be repeated until the motion bit (MOT) is cleared. Until MOT is cleared, either the Delta\_X or Delta\_Y registers will read either positive or negative full scale. If the motion register has not been read for long time, at 400 cpi it may take up to 16 read cycles to clear the buffers, at 1600 cpi, up to 64 cycles. Alternatively, writing to the Motion\_Clear register (register 0x12) will clear all stored motion at once.

**Delta\_X**

**Address: 0x03**

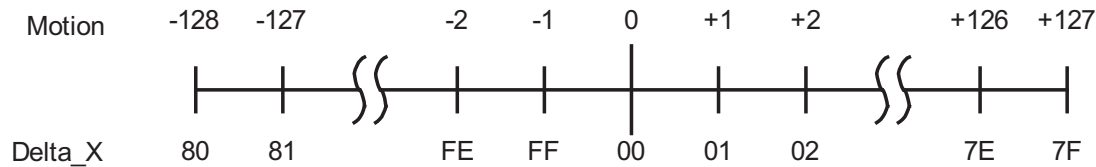
Access: Read

Reset Value: 0x00

|       |                |                |                |                |                |                |                |                |
|-------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| Bit   | 7              | 6              | 5              | 4              | 3              | 2              | 1              | 0              |
| Field | X <sub>7</sub> | X <sub>6</sub> | X <sub>5</sub> | X <sub>4</sub> | X <sub>3</sub> | X <sub>2</sub> | X <sub>1</sub> | X <sub>0</sub> |

Data Type: Eight bit 2's complement number.

USAGE: X movement is counts since last report. Absolute value is determined by resolution. Reading clears the register.



**Delta\_Y**

**Address: 0x04**

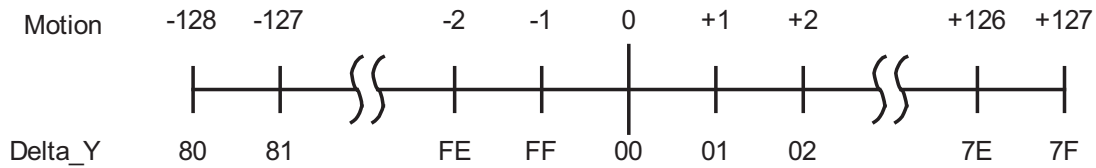
Access: Read

Reset Value: 0x00

|       |                |                |                |                |                |                |                |                |
|-------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| Bit   | 7              | 6              | 5              | 4              | 3              | 2              | 1              | 0              |
| Field | Y <sub>7</sub> | Y <sub>6</sub> | Y <sub>5</sub> | Y <sub>4</sub> | Y <sub>3</sub> | Y <sub>2</sub> | Y <sub>1</sub> | Y <sub>0</sub> |

Data Type: Eight bit 2's complement number.

USAGE: Y movement is counts since last report. Absolute value is determined by resolution. Reading clears the register.



**SQUAL****Address: 0x05**

Access: Read

Reset Value: 0x00

| Bit   | 7               | 6               | 5               | 4               | 3               | 2               | 1               | 0               |
|-------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| Field | SQ <sub>7</sub> | SQ <sub>6</sub> | SQ <sub>5</sub> | SQ <sub>4</sub> | SQ <sub>3</sub> | SQ <sub>2</sub> | SQ <sub>1</sub> | SQ <sub>0</sub> |

Data Type: Upper 8 bits of a 10-bit unsigned integer.

USAGE: SQUAL (Surface Quality) is a measure of  $\frac{1}{4}$  of the number of valid\* features visible by the sensor in the current frame. Use the following formula to find the total number of valid features.

Number of features = SQUAL register value \*4

The maximum SQUAL register value is 169. Since small changes in the current frame can result in changes in SQUAL, variations in SQUAL when looking at a surface are expected. The graph below shows 250 sequentially acquired SQUAL values, while a sensor was moved slowly over white paper. SQUAL is nearly equal to zero, if there is no surface below the sensor. SQUAL is typically maximized when the navigation surface is at the optimum distance from the imaging lens (the nominal Z-height).

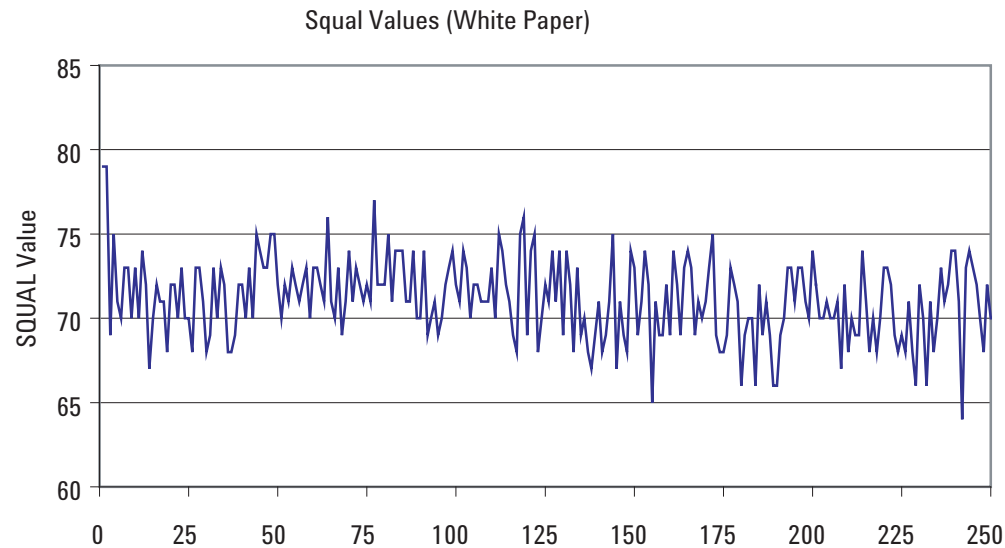


Figure 26. Squal values (white paper)

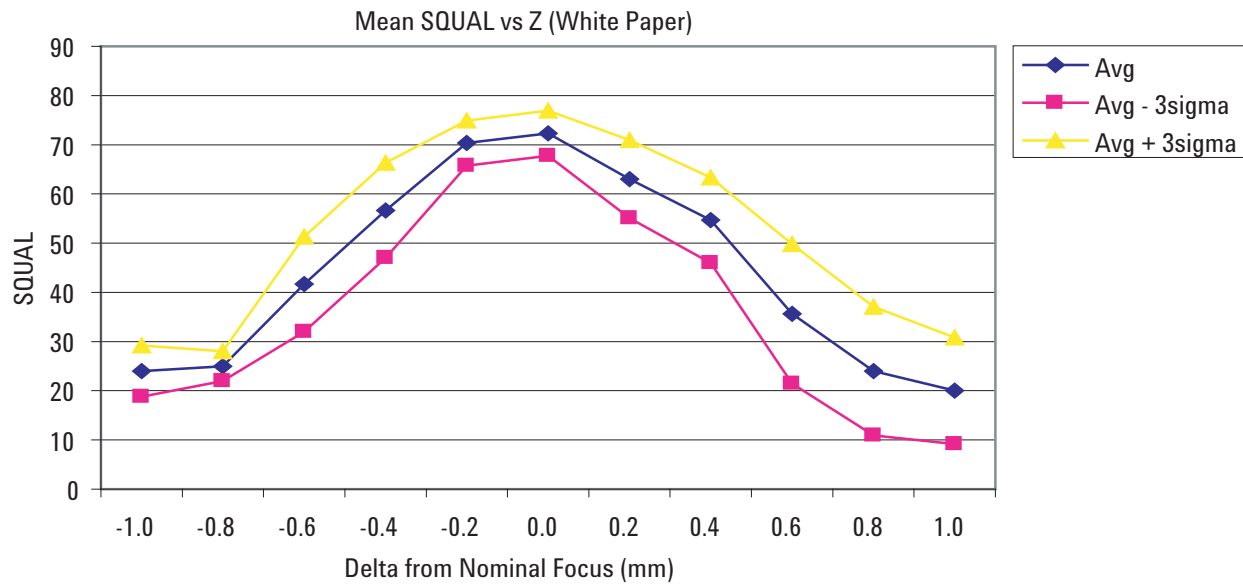


Figure 27. Mean squal vs. Z (white paper)

| <b>Pixel_Sum</b> |                 | <b>Address: 0x06</b> |                 |                 |                 |                 |                 |                 |
|------------------|-----------------|----------------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| Access: Read     |                 | Reset Value: 0x00    |                 |                 |                 |                 |                 |                 |
| Bit              | 7               | 6                    | 5               | 4               | 3               | 2               | 1               | 0               |
| Field            | AP <sub>7</sub> | AP <sub>6</sub>      | AP <sub>5</sub> | AP <sub>4</sub> | AP <sub>3</sub> | AP <sub>2</sub> | AP <sub>1</sub> | AP <sub>0</sub> |

Data Type: High 8 bits of an unsigned 16-bit integer.

USAGE: This register is used to find the average pixel value. It reports the upper byte of a 16-bit counter which sums all 900 pixels in the current frame. It may be described as the full sum divided by 256. To find the average pixel value, use the following formula:

$$\text{Average Pixel} = \text{Register Value} * 256 / 900 = \text{Register Value} / 3.51$$

The maximum register value is 221 (63 \* 900/256 truncated to an integer). The minimum is 0. The pixel sum value can change on every frame.

| <b>Maximum_Pixel</b> |   | <b>Address: 0x07</b> |                 |                 |                 |                 |                 |                 |
|----------------------|---|----------------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| Access: Read         |   | Reset Value: 0x00    |                 |                 |                 |                 |                 |                 |
| Bit                  | 7 | 6                    | 5               | 4               | 3               | 2               | 1               | 0               |
| Field                | 0 | 0                    | MP <sub>5</sub> | MP <sub>4</sub> | MP <sub>3</sub> | MP <sub>2</sub> | MP <sub>1</sub> | MP <sub>0</sub> |

Data Type: Six bit number.

USAGE: Maximum Pixel value in current frame. Minimum value = 0, maximum value = 63. The maximum pixel value can vary with every frame.

---

**Reserved** **Address: 0x08**

---

**Reserved** **Address: 0x09**

**Configuration\_bits****Address: 0x0a**

Access: Read/Write

Reset Value: 0x09

|       |   |          |          |     |          |          |          |          |
|-------|---|----------|----------|-----|----------|----------|----------|----------|
| Bit   | 7 | 6        | 5        | 4   | 3        | 2        | 1        | 0        |
| Field | 0 | LED_MODE | Sys Test | RES | Reserved | Reserved | Reserved | Reserved |

Data Type: Bit field

USAGE: Register 0x0a allows the user to change the configuration of the sensor. Shown below are the bits, their default values, and optional values.

| Field Name | Description  |
|------------|--|
| BIT 7      | Must always be zero  |
| LED_MODE   | LED Shutter Mode<br><b>0 = Shutter mode off (LED always on)</b><br>1 = Shutter mode on (LED only on when illumination is required)   |
| Sys Test   | System Tests<br><b>0 = no tests</b><br>1 = perform all system tests, output 16 bit CRC via Data_Out_Upper and Data_Out_Lower registers.<br><br>NOTE: The test will fail if SROM is loaded. Perform a hardware reset before executing this test. Reload SROM after the test is completed.<br><br>NOTE: Since part of the system test is a RAM test, the RAM and SRAM will be overwritten with the default values when the test is done. If any configuration changes from the default are needed for operation, make the changes AFTER the system test is run. The system test takes 200ms (@24MHz) to complete.<br><br>NOTE: Do not access the Synchronous Serial Port during system test. |
| RES        | Resolution in counts per inch<br><b>0 = 400</b><br>1 = 1600  |
| Reserved   | Reserved   |
| Reserved   | Reserved   |
| Reserved   | Reserved   |
| Reserved   | Reserved   |

**Extended\_Config****Address: 0x0b**

Access: Read/Write

Reset Value: 0x00

|       |      |          |          |          |          |            |      |          |
|-------|------|----------|----------|----------|----------|------------|------|----------|
| Bit   | 7    | 6        | 5        | 4        | 3        | 2          | 1    | 0        |
| Field | Busy | Reserved | Reserved | Reserved | Reserved | Serial_NPU | NAGC | Fixed_FR |

Data Type: Bit field

USAGE: Register 0x0b allows the user to change the configuration of the sensor. Shown below are the bits, their default values, and optional values.

| Field Name | Description   |
|------------|---|
| Busy       | Read-only bit. Indicates if it is safe to write to one or more of the following registers:<br>Frame_Period_Max_Bound_Upper and Lower<br>Frame_Period_Min_Bound_Upper and Lower<br>Shutter_Max_Bound_Upper and Lower<br>After writing to the Frame_Period_Max_Bound_Upper register, at least two frames must pass before writing again to any of the above registers. This bit may be used in lieu of a timer since the actual frame rate may not be known when running in auto mode.<br><b>0 = writing to the registers is allowed</b><br>1 = do not write to the registers yet |
| Reserved   | Reserved  |
| Reserved   | Reserved  |
| Reserved   | Reserved  |
| Reserved   | Reserved  |
| Serial_NPU | Disable serial port pull-up current sources<br><b>0 = no, current sources are on</b><br>1 = yes, current sources are off  |
| NAGC       | Disable AGC. Shutter will be set to the value in the Shutter_Max_Bound registers.<br><b>0 = no, AGC is active</b><br>1 = yes, AGC is disabled   |
| Fixed_FR   | Fixed frame rate (disable automatic frame rate control). When this bit is set, the frame rate will be determined by the value in the Frame_Period_Max_Bound registers.<br><b>0 = automatic frame rate</b><br>1 = fixed frame rate   |

**Data\_Out\_Lower****Address: 0x0c**

Access: Read

Reset Value: Undefined

|       |                 |                 |                 |                 |                 |                 |                 |                 |
|-------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| Bit   | 7               | 6               | 5               | 4               | 3               | 2               | 1               | 0               |
| Field | DO <sub>7</sub> | DO <sub>6</sub> | DO <sub>5</sub> | DO <sub>4</sub> | DO <sub>3</sub> | DO <sub>2</sub> | DO <sub>1</sub> | DO <sub>0</sub> |

**Data\_Out\_Upper****Address: 0x0d**

Access: Read

Reset Value: Undefined

|       |                  |                  |                  |                  |                  |                  |                 |                 |
|-------|------------------|------------------|------------------|------------------|------------------|------------------|-----------------|-----------------|
| Bit   | 7                | 6                | 5                | 4                | 3                | 2                | 1               | 0               |
| Field | DO <sub>15</sub> | DO <sub>14</sub> | DO <sub>13</sub> | DO <sub>12</sub> | DO <sub>11</sub> | DO <sub>10</sub> | DO <sub>9</sub> | DO <sub>8</sub> |

Data Type: Sixteen bit word.

USAGE: Data in these registers come from the system self test or the SROM CRC test. The data can be read out 0x0d, or 0x0d first, then 0x0c.

|                              | <b>Data_Out_Upper</b> | <b>Data_Out_Lower</b> |
|------------------------------|-----------------------|-----------------------|
| <b>System test results:</b>  | 0x1B                  | 0xBF                  |
| <b>SROM CRC Test Result:</b> | 0xBE                  | 0xEF                  |

System Test: This test is initiated via the Configuration\_Bits register. It performs several tests to verify that the hardware is functioning correctly. Perform a hardware reset just prior to running the test. SROM contents and register settings will be lost.

SROM CRC Test: Performs a CRC on the SROM contents. The test is initiated by writing a particular value to the SROM\_Enable register.

**Shutter\_Lower****Address: 0x0e**

Access: Read

Reset Value: 0x85

|       |                |                |                |                |                |                |                |                |
|-------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| Bit   | 7              | 6              | 5              | 4              | 3              | 2              | 1              | 0              |
| Field | S <sub>7</sub> | S <sub>6</sub> | S <sub>5</sub> | S <sub>4</sub> | S <sub>3</sub> | S <sub>2</sub> | S <sub>1</sub> | S <sub>0</sub> |

**Shutter\_Lower****Address: 0x0e**

Access: Read

Reset Value: 0x85

|       |                 |                 |                 |                 |                 |                 |                |                |
|-------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|----------------|----------------|
| Bit   | 7               | 6               | 5               | 4               | 3               | 2               | 1              | 0              |
| Field | S <sub>15</sub> | S <sub>14</sub> | S <sub>13</sub> | S <sub>12</sub> | S <sub>11</sub> | S <sub>10</sub> | S <sub>9</sub> | S <sub>8</sub> |

**Shutter\_Upper****Address: 0x0f**

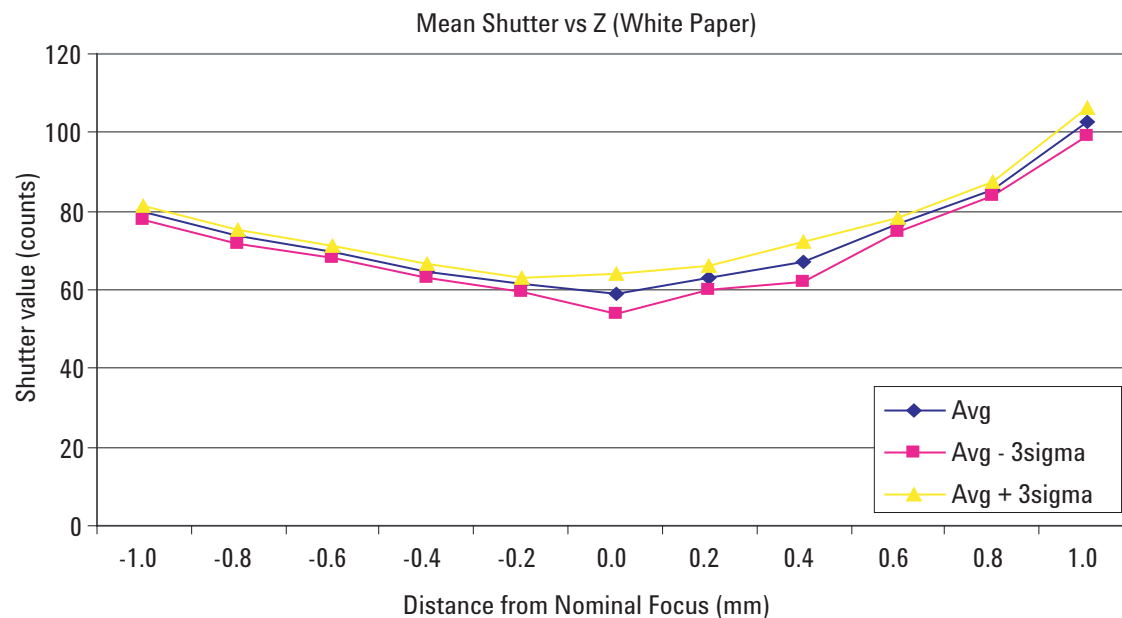
Access: Read

Reset Value: 0x00

Data Type: Sixteen bit unsigned integer.

USAGE: Units are clock cycles. Read Shutter\_Upper first, then Shutter\_Lower. They should be read consecutively. The shutter is adjusted to keep the average and maximum pixel values within normal operating ranges. The shutter value is checked and automatically adjusted to a new value if needed on every frame when operating in default mode. When the shutter adjusts, it changes by  $\pm 1/16$  of the current value. The shutter value can be set manually by setting the AGC mode to Disable using the Extended\_Config register and writing to the Shutter\_Maximum\_Bound registers. Because the automatic frame rate feature is related to shutter value. It may also be appropriate to enable the Fixed Frame Rate mode using the Extended\_Config register.

Shown below is a graph of 250 sequentially acquired shutter values, while the sensor was moved slowly over white paper.

**Figure 28. Mean shutter vs. Z (white paper)**

The maximum value of the shutter is dependent upon the setting in the Shutter\_Max\_Bound\_Upper and Shutter\_Max\_Bound\_Lower registers.

---

**Frame\_Period\_Lower****Address: 0x10**

Access: Read

Reset Value: Undefined

|       |                 |                 |                 |                 |                 |                 |                 |                 |
|-------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| Bit   | 7               | 6               | 5               | 4               | 3               | 2               | 1               | 0               |
| Field | FP <sub>7</sub> | FP <sub>6</sub> | FP <sub>5</sub> | FP <sub>4</sub> | FP <sub>3</sub> | FP <sub>2</sub> | FP <sub>1</sub> | FP <sub>0</sub> |

---

**Frame\_Period\_Upper****Address: 0x11**

Access: Read

Reset Value: Undefined

|       |                  |                  |                  |                  |                  |                  |                 |                 |
|-------|------------------|------------------|------------------|------------------|------------------|------------------|-----------------|-----------------|
| Bit   | 7                | 6                | 5                | 4                | 3                | 2                | 1               | 0               |
| Field | FP <sub>15</sub> | FP <sub>14</sub> | FP <sub>13</sub> | FP <sub>12</sub> | FP <sub>11</sub> | FP <sub>10</sub> | FP <sub>9</sub> | FP <sub>8</sub> |

Data Type: Sixteen bit unsigned integer.

USAGE: Read these registers to determine the current frame period and to calculate the frame rate. Units are clock cycles. The formula is

$$\text{Frame Rate} = \text{Clock Frequency} / \text{Register value}$$

To read from the registers, read Frame\_Period\_Upper first followed by Frame\_Period Lower.

To set the frame rate manually, disable automatic frame rate mode via the Extended\_Config register and write the desired count value to the Frame\_Period\_Maximum\_Bound registers.

The following table lists some Frame\_Period values for popular frame rates with a 24MHz clock.

| Frames/second | Counts  |      | Frame_Period |       |
|---------------|---------|------|--------------|-------|
|               | Decimal | Hex  | Upper        | Lower |
| 6469          | 3,710   | 0E7E | 0E           | 7E    |
| 5000          | 4,800   | 12C0 | 12           | C0    |
| 3000          | 8,000   | 1F40 | 1F           | 40    |
| 2000          | 12,000  | 2EE0 | 2E           | E0    |

---

**Motion\_Clear****Address: 0x12**

Access: Write

Reset Value: Undefined

Data Type: Any.

USAGE: Writing any value to this register will cause the Delta\_X, Delta\_Y, and internal motion registers to be cleared. Use this as a fast way to reset the motion counters to zero without resetting the entire chip.



---

**Frame\_Capture****Address: 0x13**

Access: Read/Write

Reset Value: 0x00

| Bit   | 7               | 6               | 5               | 4               | 3               | 2               | 1               | 0               |
|-------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| Field | FC <sub>7</sub> | FC <sub>6</sub> | FC <sub>5</sub> | FC <sub>4</sub> | FC <sub>3</sub> | FC <sub>2</sub> | FC <sub>1</sub> | FC <sub>0</sub> |

Data Type: Bit field

USAGE: Writing 0x83 to this register will cause the next available complete 1 2/3 frames of pixel values to be stored to SROM RAM. Writing to this register is required before using the Frame Capture burst mode to read the pixel values (see the Synchronous Serial Port section for more details). Writing to this register will stop navigation and cause any firmware loaded in the SROM to be overwritten. A hardware reset is required to restore navigation, and the firmware must be reloaded using the SROM Download burst method.

This register can also be used to read the frame capture data. The same data available by reading the Pixel\_Burst register using burst mode is available by reading this register in the normal fashion. The data pointer is automatically incremented after each read so all 1536 pixel values (1 and 2/3 frames) may be obtained by reading this register 1536 times in a row. Both methods share the same pointer such that reading pixel values from this register will increment the pointer causing subsequent reads from the Pixel\_Burst register (without initiating a new frame dump) to start at the current pointer location. This register will return all zeros if read before the frame capture data is ready. See the Frame Capture description in the Synchronous Serial Port section for more information.

This register will not retain the last value written. Reads will return zero or frame capture data.

---

**SROM\_Enable****Address: 0x14**

Access: Write

Reset Value: 0x00

| Bit   | 7               | 6               | 5               | 4               | 3               | 2               | 1               | 0               |
|-------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| Field | SE <sub>7</sub> | SE <sub>6</sub> | SE <sub>5</sub> | SE <sub>4</sub> | SE <sub>3</sub> | SE <sub>2</sub> | SE <sub>1</sub> | SE <sub>0</sub> |

Data Type: 8-bit number.

USAGE: Write to this register to start either SROM download or SROM CRC test.

Write 0x18 to this register before downloading SROM firmware to the SROM\_Load register. The download will not be successful unless this register contains the correct value.

Write 0xA1 to start the SROM CRC test. Wait 7ms plus one frame period, then read result from the Data\_Out\_Lower and Data\_Out\_Upper registers. Navigation is halted and the SPI port should not be used during this test.

---

**Reserved****Address: 0x15 – 0x18**

**Frame\_Period\_Max\_Bound\_Lower****Address: 0x19**

Access: Read/Write

Reset Value: 0xE0

|       |                  |                  |                  |                  |                  |                  |                  |                  |
|-------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|
| Bit   | 7                | 6                | 5                | 4                | 3                | 2                | 1                | 0                |
| Field | FBm <sub>7</sub> | FBm <sub>6</sub> | FBm <sub>5</sub> | FBm <sub>4</sub> | FBm <sub>3</sub> | FBm <sub>2</sub> | FBm <sub>1</sub> | FBm <sub>0</sub> |

**Frame\_Period\_Max\_Bound\_Upper****Address: 0x1A**

Access: Read/Write

Reset Value: 0x2E

|       |                   |                   |                   |                   |                   |                   |                  |                  |
|-------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|------------------|------------------|
| Bit   | 7                 | 6                 | 5                 | 4                 | 3                 | 2                 | 1                | 0                |
| Field | FBm <sub>15</sub> | FBm <sub>14</sub> | FBm <sub>13</sub> | FBm <sub>12</sub> | FBm <sub>11</sub> | FBm <sub>10</sub> | FBm <sub>9</sub> | FBm <sub>8</sub> |

Data Type: 16-bit unsigned integer.

USAGE: This value sets the maximum frame period (the MINIMUM frame rate) which may be selected by the automatic frame rate control, or sets the actual frame period when operating in manual mode. Units are clock cycles. The formula is

$$\text{Frame Rate} = \text{Clock Frequency} / \text{Register value}$$

To read from the registers, read Upper first followed by Lower. To write to the registers, write Lower first, followed by Upper. To set the frame rate manually, disable automatic frame rate mode via the **Extended\_Config** register and write the desired count value to these registers.

Writing to the Frame\_Period\_Max\_Bound\_Upper and Lower registers also activates any new values in the following registers:

- Frame\_Period\_Max\_Bound\_Upper and Lower
- Frame\_Period\_Min\_Bound\_Upper and Lower
- Shutter\_Max\_Bound\_Upper and Lower

Any data written to these registers will be saved but will not take effect until the write to the Frame\_Period\_Max\_Bound\_Upper and Lower is complete. After writing to this register, two complete frame times are required to implement the new settings. Writing to any of the above registers before the implementation is complete may put the chip into an undefined state requiring a reset. The "Busy" bit in the Extended\_Config register may be used in lieu of a timer to determine when it is safe to write. See the Extended\_Config register for more details.

The following table lists some Frame\_Period values for popular frame rates (clock rate = 24MHz).

In addition, the three bound registers must also follow this rule when set to non-default values:

$$\text{Frame\_Period\_Max\_Bound} \geq \text{Frame\_Period\_Min\_Bound} + \text{Shutter\_Max\_Bound}.$$

| Frames/second | Counts  |      | Frame_Period |       |
|---------------|---------|------|--------------|-------|
|               | Decimal | Hex  | Upper        | Lower |
| 6469          | 3,710   | 0E7E | 0E           | 7E    |
| 5000          | 4,800   | 12C0 | 12           | C0    |
| 3000          | 8,000   | 1F40 | 1F           | 40    |
| 2000          | 12,000  | 2EE0 | 2E           | E0    |

---

**Frame\_Period\_Min\_Bound\_Lower****Address: 0x1B**

Access: Read/Write

Reset Value: 0xAC (before SROM download)  
0x7E (after SROM download)

| Bit   | 7                | 6                | 5                | 4                | 3                | 2                | 1                | 0                |
|-------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|
| Field | FBm <sub>7</sub> | FBm <sub>6</sub> | FBm <sub>5</sub> | FBm <sub>4</sub> | FBm <sub>3</sub> | FBm <sub>2</sub> | FBm <sub>1</sub> | FBm <sub>0</sub> |

---

**Frame\_Period\_Min\_Bound\_Upper****Address: 0x1C**

Access: Read/Write

Reset Value: 0x0D (before SROM download)  
0x0E (after SROM download)

| Bit   | 7                 | 6                 | 5                 | 4                 | 3                 | 2                 | 1                | 0                |
|-------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|------------------|------------------|
| Field | FBm <sub>15</sub> | FBm <sub>14</sub> | FBm <sub>13</sub> | FBm <sub>12</sub> | FBm <sub>11</sub> | FBm <sub>10</sub> | FBm <sub>9</sub> | FBm <sub>8</sub> |

Data Type: 16-bit unsigned integer.

USAGE: This value sets the minimum frame period (the MAXIMUM frame rate) that may be selected by the automatic frame rate control. Units are clock cycles. The formula is

Frame Rate = Clock Rate / Register value

To read from the registers, read Upper first followed by Lower. To write to the registers, write Lower first, followed by Upper, then execute a write to the Frame\_Period\_Max\_Bound\_Upper and Lower registers. The minimum allowed write value is 0x7E0E; the maximum is 0xFFFF.

Reading this register will return the most recent value that was written to it. However, the value will take effect only after a write to the Frame\_Period\_Max\_Bound\_Upper and Lower registers. After writing to Frame\_Period\_Max\_Bound\_Upper, wait at least two frame times before writing to Frame\_Period\_Min\_Bound\_Upper or Lower again. The "Busy" bit in the Extended\_Config register may be used in lieu of a timer to determine when it is safe to write. See the Extended\_Config register for more details.

In addition, the three bound registers must also follow this rule when set to non-default values:

Frame\_Period\_Max\_Bound ≥ Frame\_Period\_Min\_Bound + Shutter\_Max\_Bound.

---

**Shutter\_Max\_Bound\_Lower****Address: 0x1D**

Access: Read/Write

Reset Value: 0x8C (before SROM download)  
0x00 (after SROM download)

|       |                 |                 |                 |                 |                 |                 |                 |                 |
|-------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| Bit   | 7               | 6               | 5               | 4               | 3               | 2               | 1               | 0               |
| Field | SB <sub>7</sub> | SB <sub>6</sub> | SB <sub>5</sub> | SB <sub>4</sub> | SB <sub>3</sub> | SB <sub>2</sub> | SB <sub>1</sub> | SB <sub>0</sub> |

---

**Shutter\_Max\_Bound\_Upper****Address: 0x1E**

Access: Read/Write

Reset Value: 0x20

|       |                  |                  |                  |                  |                  |                  |                 |                 |
|-------|------------------|------------------|------------------|------------------|------------------|------------------|-----------------|-----------------|
| Bit   | 7                | 6                | 5                | 4                | 3                | 2                | 1               | 0               |
| Field | SB <sub>15</sub> | SB <sub>14</sub> | SB <sub>13</sub> | SB <sub>12</sub> | SB <sub>11</sub> | SB <sub>10</sub> | SB <sub>9</sub> | SB <sub>8</sub> |

Data Type: 16-bit unsigned integer.

USAGE: This value sets the maximum allowable shutter value when operating in automatic mode. Units are clock cycles. Since the automatic frame rate function is based on shutter value, the value in these registers can limit the range of the frame rate control. To read from the registers, read Upper first followed by Lower. To write to the registers, write Lower first, followed by Upper, then execute a write to the Frame\_Period\_Max\_Bound\_Upper and Lower registers. To set the shutter manually, disable the AGC via the Extended\_Config register and write the desired value to these registers.

Reading this register will return the most recent value that was written to it. However, the value will take effect only after a write to the Frame\_Period\_Max\_Bound\_Upper and Lower registers. After writing to Frame\_Period\_Max\_Bound\_Upper, wait at least two frame times before writing to Shutter\_Max\_Bound\_Upper or Lower again. The "Busy" bit in the Extended\_Config register may be used in lieu of a timer to determine when it is safe to write. See the Extended\_Config register for more details.

In addition, the three bound registers must also follow this rule when set to non-default values:

$$\text{Frame\_Period\_Max\_Bound} \geq \text{Frame\_Period\_Min\_Bound} + \text{Shutter\_Max\_Bound}.$$

---

**SROM\_ID****Address: 0x1F**

Access: Read

Reset Value: 0x00

|       |                 |                 |                 |                 |                 |                 |                 |                 |
|-------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| Bit   | 7               | 6               | 5               | 4               | 3               | 2               | 1               | 0               |
| Field | SR <sub>7</sub> | SR <sub>6</sub> | SR <sub>5</sub> | SR <sub>4</sub> | SR <sub>3</sub> | SR <sub>2</sub> | SR <sub>1</sub> | SR <sub>0</sub> |

Data Type: 8-bit unsigned integer.

USAGE: Contains the revision of the downloaded Shadow ROM firmware. If the firmware has been successfully downloaded and the chip is operating out of SROM, this register will contain the SROM firmware revision, otherwise it will contain 0x00.

Note: The IC hardware revision is available by reading the Revision\_ID register (register 0x01).

---

**Reserved****Address: 0x20 – 0x3C**

**Observation****Address: 0x3D**

Access: Read/Write

Reset Value: 0x00

|       |                 |          |                 |          |          |          |                 |                 |
|-------|-----------------|----------|-----------------|----------|----------|----------|-----------------|-----------------|
| Bit   | 7               | 6        | 5               | 4        | 3        | 2        | 1               | 0               |
| Field | OB <sub>7</sub> | Reserved | OB <sub>5</sub> | Reserved | Reserved | Reserved | OB <sub>1</sub> | OB <sub>0</sub> |

Data Type: Bit field

USAGE: Each bit is set by some process or action at regular intervals, or when the event occurs. The user must clear the register by writing 0x00, wait an appropriate delay, and read the register. The active processes will have set their corresponding bit(s). This register may be used as part of a recovery scheme to detect a problem caused by EFT/B or ESD.

| Field Name      | Description                       |
|-----------------|-----------------------------------|
| OB <sub>7</sub> | If set, chip is running SROM code |
| Reserved        | Reserved                          |
| OB <sub>5</sub> | NPD pulse was detected            |
| Reserved        | Reserved                          |
| Reserved        | Reserved                          |
| Reserved        | Reserved                          |
| OB <sub>1</sub> | Set once per frame                |
| OB <sub>0</sub> | Set once per frame                |

**Reserved****Address: 0x3E****Inverse\_Product\_ID****Address: 0x3F**

Access: Read

Reset Value: 0xF8

|       |                   |                   |                   |                   |                   |                   |                   |                   |
|-------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|
| Bit   | 7                 | 6                 | 5                 | 4                 | 3                 | 2                 | 1                 | 0                 |
| Field | NPID <sub>7</sub> | NPID <sub>6</sub> | NPID <sub>5</sub> | NPID <sub>4</sub> | NPID <sub>3</sub> | NPID <sub>2</sub> | NPID <sub>1</sub> | NPID <sub>0</sub> |

Data Type: Inverse 8-Bit unsigned integer

USAGE: This value is the inverse of the Product\_ID, located at the inverse address. It can be used to test the SPI port.

**Pixel\_Burst****Address: 0x40**

Access: Read

Reset Value: 0x00

|       |                 |                 |                 |                 |                 |                 |                 |                 |
|-------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| Bit   | 7               | 6               | 5               | 4               | 3               | 2               | 1               | 0               |
| Field | PB <sub>7</sub> | PB <sub>6</sub> | PB <sub>5</sub> | PB <sub>4</sub> | PB <sub>3</sub> | PB <sub>2</sub> | PB <sub>1</sub> | PB <sub>0</sub> |

Data Type: Eight bit unsigned integer

USAGE: The Pixel\_Burst register is used for high-speed access to all the pixel values from one and 2/3 complete frame. See the Synchronous Serial Port section for use details.

---

**Motion\_Burst****Address: 0x50**

Access: Read

Reset Value: 0x00

|       |                 |                 |                 |                 |                 |                 |                 |                 |
|-------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| Bit   | 7               | 6               | 5               | 4               | 3               | 2               | 1               | 0               |
| Field | MB <sub>7</sub> | MB <sub>6</sub> | MB <sub>5</sub> | MB <sub>4</sub> | MB <sub>3</sub> | MB <sub>2</sub> | MB <sub>1</sub> | MB <sub>0</sub> |

Data Type: Various, depending on data

USAGE: The Motion\_Burst register is used for high-speed access to the Motion, Delta\_X, and Delta\_Y, SQUAL, Shutter\_Upper, and Shutter\_Lower and Maximum\_Pixel registers. See the Synchronous Serial Port section for use details.

---

**SROM\_Load****Address: 0x 60**

Access: Write

Rset Value: N/A

Data Type: Eight bit unsigned integer

USAGE: The SROM\_Load register is used for high-speed programming of the ADNS-3080 from an external SROM or microcontroller. See the Synchronous Serial Port section for use details.

### **Read Also**

ADNK-3083 Design Guide

ADNK-3080 Sample Kit

### **Relevant Application Notes**

Application Note AN 5035\*

Application Note AN 5034\*

Application Note AN 5036\*

\* The application notes content are applicable for ADNS-3080 as well.

### **Ordering Information**

Specify part number as follows:

ADNS-3080 = Sensor IC in a 20 pin plastic optical package, 20 per tube.

ADNS-2120 = Round Optical Mouse Lens

ADNS-2120-001 = Trim Optical Mouse Lens

ADNS-2220 = LED Assembly Clip (Clear)

ADNS-2220-001 LED Assembly Clip (Black)

HLMP-ED80-XX000 = LED

For product information and a complete list of distributors, please go to our web site: [www.avagotech.com](http://www.avagotech.com)

Avago, Avago Technologies, and the A logo are trademarks of Avago Technologies in the United States and other countries. Data subject to change. Copyright © 2005-2009 Avago Technologies. All rights reserved. Obsoletes 5989-3422EN AV02-0366EN - December 3, 2009

