AN2960
Application note

LIS331DL: 3-axis - ±2 *g* /±8 *g* digital output ultra compact linear accelerometer

Introduction

This document is intended to provide application information for the LIS331DL low-voltage 3-axis digital output linear MEMS accelerometer housed in an LGA package.

The LIS331DL is an ultra-compact low-power 3-axis linear accelerometer that includes a sensing element and an IC interface capable taking information from the sensing element and providing the measured acceleration data to external applications via an I^2C/SPI serial interface.

The sensing element used to detect acceleration is manufactured using a dedicated process developed by ST to produce inertial sensors and actuators in silicon.

The IC interface is instead manufactured using a CMOS process that allows a high level of integration to design a dedicated circuit which is factory trimmed to better match the sensing element characteristics.

The LIS331DL has a user-selectable full scale of ±2 *g* and ±8 *g* and is capable of measuring accelerations with an output data rate of 100 Hz or 400 Hz. A self-test capability allows the user to check that the system is operating correctly.

The device features two independent, highly programmable interrupt sources that can be configured either to generate an inertial wake-up interrupt signal when a programmable acceleration threshold is exceeded along one of the three axes, to detect a free-fall or to recognize single/double click events.

Two independent pins can be configured to provide interrupt signals to connected devices.

The LIS331DL is available in a plastic SMD package and is designed to operate over a temperature range extending from -40 °C to +85 °C.

The ultra small size and weight of the SMD package make it an ideal choice for handheld portable applications such as cell phones and PDAs, or any other application where reduced package size and weight are required.

Contents

List of tables

List of figures

1 Theory of operation

The LIS331DL is an ultra compact, low-power, digital output 3-axis linear accelerometer packaged in an LGA package. The complete device includes a sensing element and an IC interface capable of taking information from the sensing element and providing a signal to external applications through an I^{2}C/SPI serial interface (*Figure 1*).

A proprietary process is used to create a surface micro-machined accelerometer. The technology allows the creation of suspended silicon structures which are attached to the substrate at a few points, called "anchors", and are free to move in the direction of the sensed acceleration. To be compatible with traditional packaging techniques, a cap is placed on top of the sensing element to avoid the blockage the moving parts during the moulding phase of the plastic encapsulation.

When an acceleration is applied to the sensor, the proof mass displaces from its nominal position, causing an imbalance in the capacitive half-bridge. This imbalance is measured using charge integration in response to a voltage pulse applied to the sense capacitor.

In steady state, the nominal value of the capacitors are a few pico farads, and when an acceleration is applied the maximum variation of the capacitive load is of a few femto farads.

The complete measurement chain is composed of a low-noise capacitive amplifier which converts the capacitive unbalancing of the MEMS sensor into an analog voltage, and by analog-to-digital converters.

The acceleration data may be accessed through an I^{2}C/SPI interface, making the device particularly suitable for direct interfacing with a microcontroller.

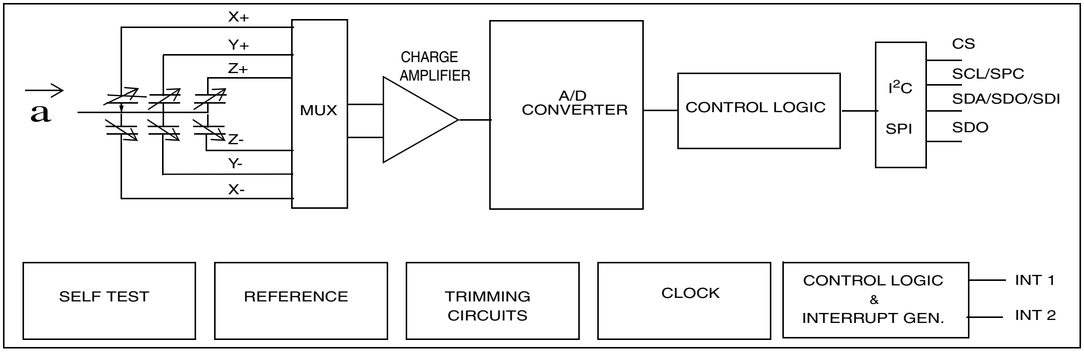
Data synchronization in a digital system employing the device is made simpler through the use of the DataReady signal (RDY), which indicates when a new set of measured acceleration data is available, thus simplifying data synchronization in the digital system employing the device.

The LIS331DL also features two independent, fully-programmable interrupt sources which can be programmed to generate an interrupt signal when a programmable acceleration threshold is exceeded along one of the three axes, or to detect a free-fall event.

The IC interface is factory calibrated for sensitivity (So) and zero-*g* level (Off).

The trimming values are stored in the device by a non-volatile structure. Any time the device is turned on, the trimming parameters are loaded into the registers to be employed during normal operation. This allows the user to use the device without further calibration.

Figure 1. Device block diagram

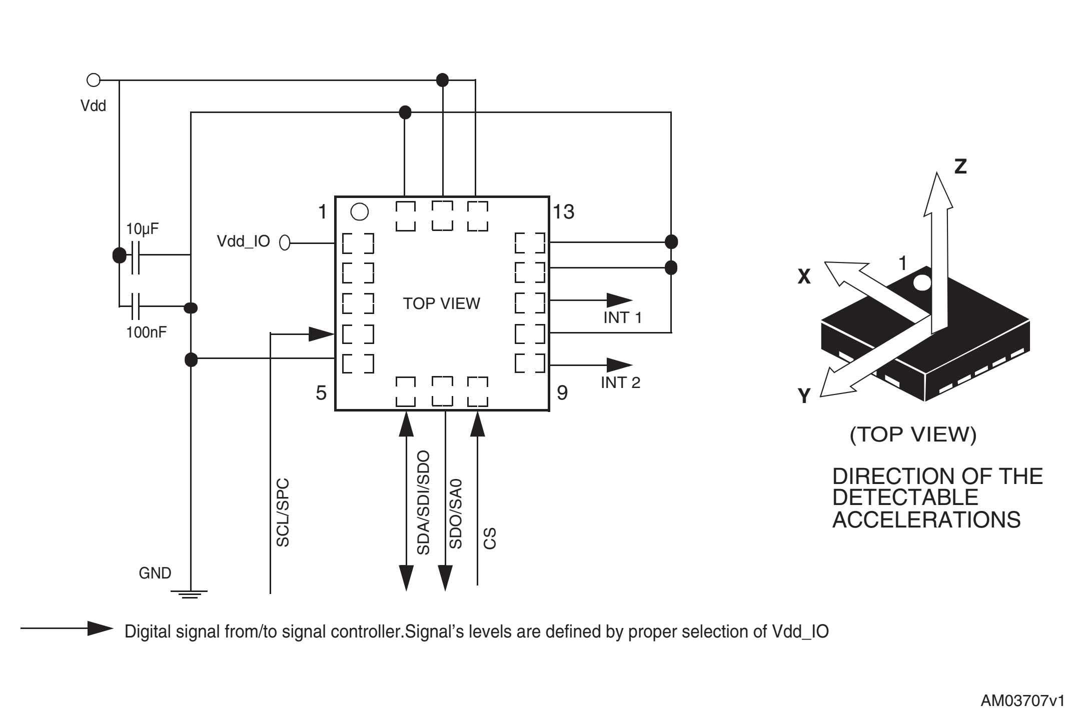


2 Electrical connection and board layout hints

2.1 Electrical connection

The typical electrical connection of the LIS331DL is shown in *Figure 2*.

Figure 2. LIS331DL electrical connection


Vdd
10µF
100nF
Vdd_IO
1
13
5
9
TOP VIEW
INT 1
INT 2
SCL/SPC
SDA/SDI/SDO
SDO/SA0
CS
GND
Z
X
Y
1
(TOP VIEW)
DIRECTION OF THE DETECTABLE ACCELERATIONS
Digital signal from/to signal controller.Signal's levels are defined by proper selection of Vdd_IO
AM03707v1

The LIS331DL is designed to operate with a supply voltage spanning from 2.16 V up to 3.6 V, while the serial interface can function down to 1.8 V.

The device core is supplied through the Vdd line (Vdd typ=2.5 V) while the I/O pads are supplied through the Vdd_IO line. The typical current consumption in normal mode at 2.5 V is 400 µA.

Both voltage supplies must be present at the same time to ensure proper behavior of the IC. It is possible to remove Vdd, maintaining Vdd_IO, without blocking the communication bus.

Adequate power supply decoupling is required to ensure IC performance. Optimum decoupling is achieved by using two capacitors of different types that target different kinds of noise on the power supply leads. To attenuate high frequency transients, spikes, or digital hash on the line, the use of one 100 nF ceramic or polyester capacitor is recommended, which must be placed as close as possible to the device Vdd lead. For filtering lower-frequency noise signals, a larger aluminum capacitor of 10 µF or greater should be placed near the device in parallel with the former capacitor. It is recommended to place these capacitors as near as possible to pin 14 of the device.

The functionality of the device and the measured acceleration data are selectable and accessible through the I^{2}C/SPI interface. When using the I^{2}C, the CS must be tied high while the SDO allows the user to select between two device addresses in cases where two sensors must be connected on the same bus. Whenever a single sensor is present on the same I^2C bus, it is recommended either to connect SDO to Vdd_IO, or to leave it floating.

2.2 Soldering information

The LGA-14 package is lead-free and green package qualified for soldering heat resistance according to JEDEC J-STD-020C. Land pattern and soldering recommendations are available upon request.

3 Absolute maximum ratings

Stresses above those listed as “absolute maximum ratings” may cause permanent damage to the device. These are stress ratings only and functional operation of the device under these conditions is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

Table 1. Absolute maximum ratings

Symbol	Ratings	Maximum Value	Unit
Vdd	Supply voltage[1]	-0.3 to 6	V
Vdd_IO	I/O pin supply voltage[1]	-0.3 to 6	V
Vin	Input voltage on any control pin (CS, SCL/SPC, SDA/SDI/SDO, CK)	-0.3 to Vdd_IO +0.3	V
A_{POW}	Acceleration (any axis, powered, Vdd=2.5V)	3000 *g* for 0.5 ms	
		10000 *g* for 0.1 ms	
A_{UNP}	Acceleration (any axis, unpowered)	3000 *g* for 0.5 ms	
		10000 *g* for 0.1 ms	
T_{OP}	Operating temperature range	-40 to +85	°C
T_{STG}	Storage temperature range	-40 to +125	°C
ESD	Electrostatic discharge protection	Class 1: 0 - 2 kV HBM	

[1] Supply voltage on any pin should never exceed 6.0 V

Warning:This is an ESD sensitive device. Improper handling can cause permanent damage to the part.

Warning:This is a mechanical shock-sensitive device, improper handling can cause permanent damage to the part.

4 Digital interfaces

The registers embedded in the LIS331DL may be accessed through I^{2}C and SPI serial interfaces. The latter may be SW configured to operate in either 3-wire or 4-wire interface mode.

The serial interfaces are mapped onto the same pads. To select/exploit the I^{2}C interface, the **CS** line must be tied high (i.e. connected to Vdd_IO).

Table 2. Serial interface pin description

Pin Name	Pin description
CS	SPI chip select (CS) I^{2}C/SPI selector (1: I^{2}C mode; 0: SPI enabled)
SCL/SPC	SPI CK line (SCL) I^{2}C clock line (SPC)
SDI/SDA/SDO	I^{2}C serial data (SDA) SPI data in (SDI) SPI data out (SDO) -for 3-wire SPI mode
SDO	I^{2}C less significant bit of device address SPI data out (SDO) - for 4-wire SPI mode

4.1 I^{2}C bus interface

The LIS331DL I^{2}C is a bus slave. The I^{2}C is employed to write/read the data to/from the registers.

Relevant I^{2}C terminology is provided in *Table 3*:

Table 3. Terminology

Term	Description
Transmitter	The device which sends data to the bus
Receiver	The device which receives data from the bus
Master	The device which initiates a transfer, generates clock signals and terminates a transfer
Slave	The device addressed by the master

There are two signals associated with the I^{2}C bus: the serial clock line (SCL) and the serial data line (SDA). The latter is a bidirectional line used for sending and receiving data to/from the interface. Both lines are connected to Vdd_IO through a pull-up resistor embedded in the LIS331DL. When the bus is free, both the lines are high.

The I^{2}C interface is compliant with fast mode (400 kHz) I^{2}C standards as well as normal mode.

4.1.1 I^{2}C operation

The transaction on the bus is started through a START (ST) condition, which is defined as a HIGH to LOW transition on the data line, while the SCL line is held HIGH. After the START condition has been generated by the master, the bus is considered busy. The next byte of data transmitted contains the address of the slave in the first 7 bits and the 8th bit instructs whether the master is receiving data from the slave or transmitting data to the slave (SAD subsequence). When an address is sent, each device in the system compares the first seven bits after a start condition with its own address. If they match, the device considers itself addressed by the master.

The slave address (SAD) associated with the LIS331DL may be selected from the two predefined values 0011100b or 0011101b depending on the logic level present on the SDO pin. Specifically, if the SDO pin is either connected to Vdd_IO or left unconnected, the slave address is 0011101b. Otherwise, when it is connected to GND, the slave address is 0011100b. When placing two sensors on the same bus is not required, it is recommended to use slave address 0011101b by either connecting the SDO pin to Vdd_IO, or leaving it floating.

Data transfer with acknowledge is mandatory. The transmitter must release the **SDA** line during the acknowledge pulse. The receiver must then pull the data line LOW so that it remains stable low during the HIGH period of the acknowledge clock pulse. A receiver that has been addressed is obligated to generate an acknowledge after each byte of data has been received.

The I^{2}C embedded in the LIS331DL behaves as a slave device and the following protocol must be adhered to. After the start condition (ST), a slave address is sent. Once a slave acknowledge (SAK) has been returned, an 8-bit sub-address will be transmitted: the 7 LSb represents the actual register address, while the MSB enables address auto-increment. If the MSb of the SUB field is 1, the SUB (register address) will be automatically incremented to allow multiple data read/write. Otherwise, if the MSB of the SUB field is ‘0’, the SUB will remain unchanged and multiple read/write on the same address can be performed.

The slave address is completed with a read/write bit. If the bit is ‘1’ (read), a repeated START (SR) condition must be issued after the two sub-address bytes; if the bit is ‘0’ (Write) the master transmits to the slave with unchanged direction.

Transfer when master is writing one byte to slave

Master	ST	SAD + W		SUB		DATA		SP
Slave			SAK		SAK		SAK	

Transfer when master is writing multiple bytes to slave:

Master	ST	SAD + W		SUB		DATA		DATA		SP
Slave			SAK		SAK		SAK		SAK	

Transfer when master is receiving (reading) one byte of data from slave:

Master	ST	SAD + W		SUB		SR	SAD + R			NMAK	SP
Slave			SAK		SAK			SAK	DATA		

Transfer when master is receiving (reading) multiple bytes of data from slave

Master	ST	SAD + W		SUB		SR	SAD + R			MAK
Slave			SAK		SAK			SAK	DATA	

Master		MAK		NMAK	SP
Slave	DATA		DATA		

Data are transmitted in byte format (DATA). Each data transfer contains 8 bits. The number of bytes per transfer is unlimited. Data is transferred with the most significant bit (MSb) first. If a receiver cannot receive another complete byte of data until it has performed some other function, it can hold the clock line **SCL** low to force the transmitter into a wait state. Data transfer continues only when the receiver is ready for another byte and releases the data line. If a slave receiver does not acknowledge the slave address (i.e. it is not able to receive because it is performing some other real-time function) the data line must be left high by the slave. The master can then abort the transfer. A low to high transition on the **SDA** line while the **SCL** line is HIGH is defined as a STOP condition. Each data transfer must be terminated by the generation of a STOP (SP) condition.

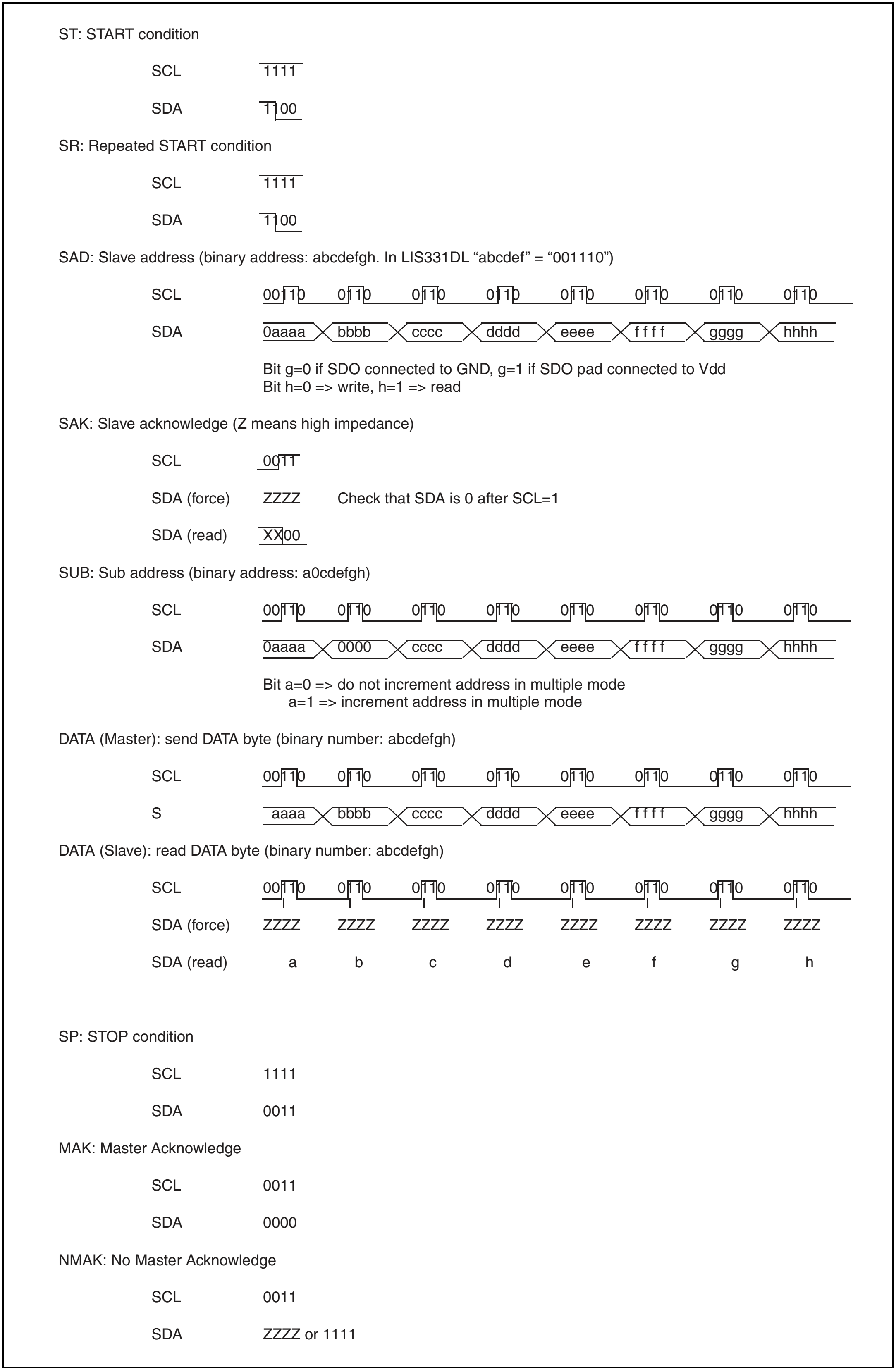
In order to read multiple bytes, it is necessary to assert the most significant bit of the sub-address field. In other words, SUB(7) must be equal to 1 while SUB(6-0) represents the address of the first register to read.

In the communication format presented, MAK signifies master acknowledge and NMAK is no master acknowledge.

4.1.2 I^2C subsequences

In order to better define subsequences and to clarify line **SCL** and **SDA** behavior, a description containing the discrete values of **SCL** and **SDA** follow. These simple subsequences are used to implement the complex commands described in the following paragraph.

Figure 3. I²C subsequences


ST: START condition
SCL 1111
SDA 1100
SR: Repeated START condition
SCL 1111
SDA 1100
SAD: Slave address (binary address: abcdefgh. In LIS331DL “abcdef” = “001110”)
SCL 00110 0110 0110 0110 0110 0110 0110 0110
SDA 0aaaa bbbb cccc dddd eeee ffff gggg hhhh
Bit g=0 if SDO connected to GND, g=1 if SDO pad connected to Vdd
Bit h=0 => write, h=1 => read
SAK: Slave acknowledge (Z means high impedance)
SCL 0011
SDA (force) ZZZZ Check that SDA is 0 after SCL=1
SDA (read) XX00
SUB: Sub address (binary address: a0cdefgh)
SCL 00110 0110 0110 0110 0110 0110 0110 0110
SDA 0aaaa 0000 cccc dddd eeee ffff gggg hhhh
Bit a=0 => do not increment address in multiple mode
a=1 => increment address in multiple mode
DATA (Master): send DATA byte (binary number: abcdefgh)
SCL 00110 0110 0110 0110 0110 0110 0110 0110
S aaaa bbbb cccc dddd eeee ffff gggg hhhh
DATA (Slave): read DATA byte (binary number: abcdefgh)
SCL 00110 0110 0110 0110 0110 0110 0110 0110
SDA (force) ZZZZ ZZZZ ZZZZ ZZZZ ZZZZ ZZZZ ZZZZ ZZZZ
SDA (read) a b c d e f g h
SP: STOP condition
SCL 1111
SDA 0011
MAK: Master Acknowledge
SCL 0011
SDA 0000
NMAK: No Master Acknowledge
SCL 0011
SDA ZZZZ or 1111

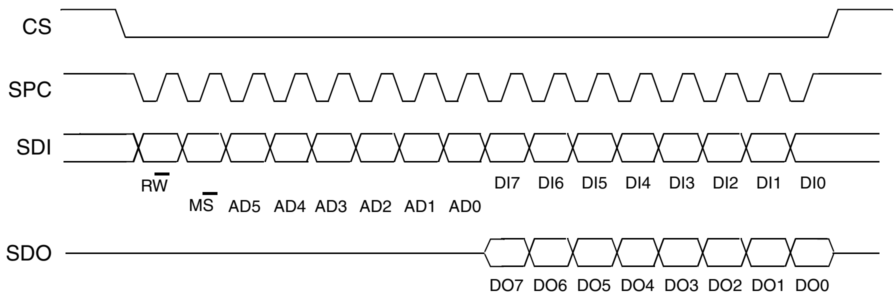
4.2 SPI bus interface

The LIS331DL SPI is a bus slave. The SPI allows writing and reading to the registers of the device.

The serial interface interacts with the outside world through 4 wires: **CS**, **SPC**, **SPDI** and **SPDO**.

4.2.1 Read & write protocol

Figure 4. Read & write protocol



CS is the chip select and is controlled by the **SPI** master. It goes low at the start of the transmission and returns high at the end. SPC is the serial port clock and is controlled by the SPI master. It is stopped high when **CS** is high (no transmission). **SDI** and **SDO** are, respectively, the serial port data input and output. These lines are driven at the falling edge of **SPC** and should be captured at the rising edge of **SPC**.

Both the read register and write register commands are completed in 16 clock pulses or in multiples of 8 in cases of multiple byte read/write. Bit duration is the time between two falling edges of **SPC**. The first bit (bit 0) starts at the first falling edge of **SPC** after the falling edge of CS while the last bit (bit 15, bit 23, etc.) starts at the last falling edge of **SPC** just before the rising edge of CS.

bit 0: R$\overline{\text{W}}$ bit. When 0, the data DI(7:0) is written into the device. When 1, the data DO(7:0) from the device is read. In the latter case, the chip drives **SDO** at the start of bit 8.

bit 1: M$\overline{\text{S}}$ bit. When 0, the address remains unchanged in multiple read/write commands. When 1, the address is auto-incremented in multiple read/write commands.

bit 2-7: address AD(5:0). This is the address field of the indexed register.

bit 8-15: data DI(7:0) (write mode). This is the data that is written to the device (MSb first).

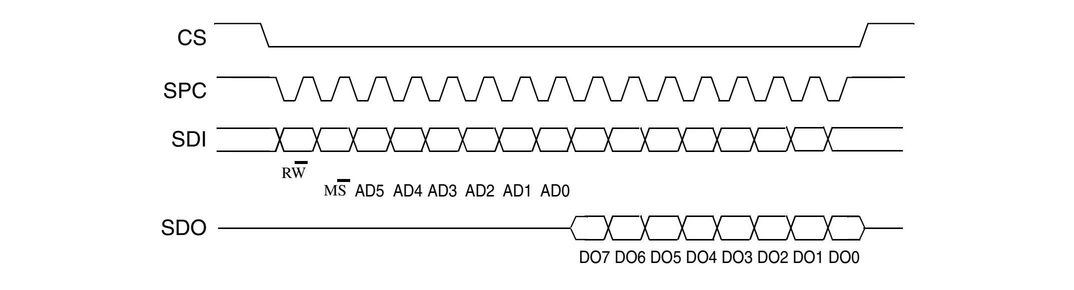
bit 8-15: data DO(7:0) (read mode). This is the data that is read from the device (MSb first).

In multiple read/write commands, further blocks of 8 clock periods are added. When the MS bit is 0, the address used to read/write data remains the same for every block. When the MS bit is 1, the address used to read/write data is incremented at every block.

The function and the behavior of **SDI** and **SDO** remain unchanged.

4.2.2 SPI read

Figure 5. SPI read protocol



The SPI Read command is performed with 16 clock pulses. A Multiple byte read command is performed by adding blocks of 8 clocks pulses to the previous one.

bit 0: READ bit. The value is 1.

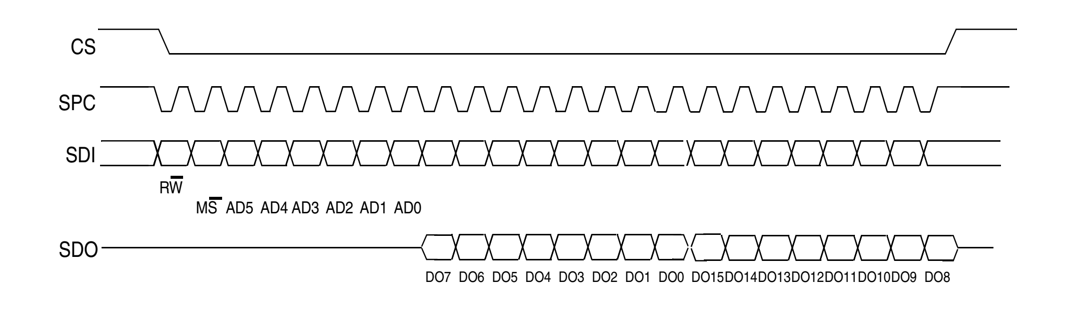
bit 1: $M\overline{S}$ bit. When 0, do not increment the address. When 1, increment the address in multiple readings.

bit 2-7: address AD(5:0). This is the address field of the indexed register.

bit 8-15: data DO(7:0) (read mode). This is the data that is read from the device (MSb first).

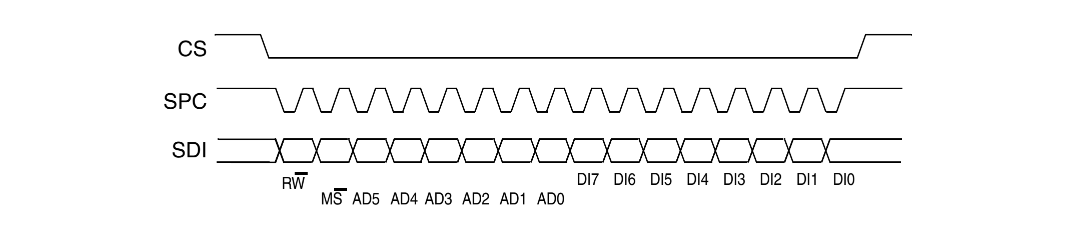
bit 16-... : data DO(...-8). Further data in multiple byte reading.

Figure 6. Multiple byte SPI read protocol (example using 2 bytes)


CS
SPC
SDI
RW
MS AD5 AD4 AD3 AD2 AD1 AD0
SDO
DO7 DO6 DO5 DO4 DO3 DO2 DO1 DO0 DO15DO14DO13DO12DO11DO10DO9 DO8

4.2.3 SPI write

Figure 7. SPI write protocol



The SPI Write command is performed with 16 clocks pulses. A Multiple byte write command is performed by adding blocks of 8 clocks pulses to the previous one.

bit 0: Write bit. The value is 0.

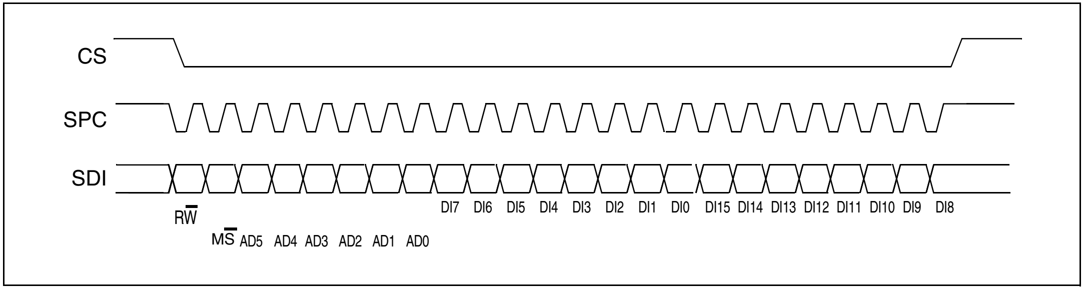
bit 1: MS bit. When 0, do not increment the address. When 1, increment the address in multiple writings.

bit 2 -7: address AD(5:0). This is the address field of the indexed register.

bit 8-15: data DI(7:0) (write mode). This is the data that is written to the device (MSb first).

bit 16-... : data DI(...-8). Further data in multiple byte writing.

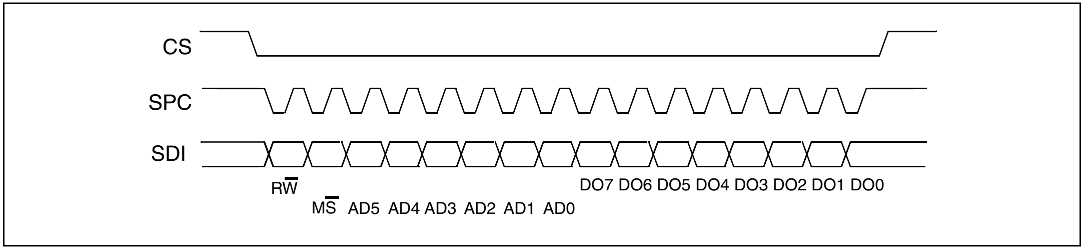
Figure 8. Multiple bytes SPI write protocol (example using 2 bytes)


CS
SPC
SDI
RW
MS AD5 AD4 AD3 AD2 AD1 AD0
DI7 DI6 DI5 DI4 DI3 DI2 DI1 DI0 DI15 DI14 DI13 DI12 DI11 DI10 DI9 DI8

4.2.4 SPI Read in 3-wires mode

3-wires mode is entered by setting the bit SIM (SPI serial interface mode selection) to 1 in CTRL_REG2.

Figure 9. SPI read protocol in 3-wires mode



The SPI Read command is performed with 16 clock pulses:

bit 0: READ bit. The value is 1.

bit 1: MS bit. When 0, do not increment the address. When 1, increment the address in multiple readings.

bit 2-7: address AD(5:0). This is the address field of the indexed register.

bit 8-15: data DO(7:0) (read mode). This is the data that is read from the device (MSb first).

The Multiple read command is also available in 3-wires mode.

5 Register description

5.1 Register address map

The table below provides a listing of the registers embedded in the device.

Table 4. Registers address map

Name	Type	Register address		Default	Comment
		Hex	Binary		
Reserved (do not modify)		00-0E			Reserved
Who_Am_I	r	0F	00 1111	00111011	Dummy register
Reserved (do not modify)		10-1F			Reserved
CTRL_REG1	rw	20	10 0000	00000111	
CTRL_REG2	rw	21	10 0001	00000000	
CTRL_REG3	rw	22	10 0010	00000000	
HP_filter_reset	r	23	10 0011	dummy	Dummy register
Reserved (do not modify)		24-26			Reserved
Status_Reg	r	27	10 0111	00000000	
--	r	28	10 1000		Not used
Out_X	r	29	10 1001	output	
--	r	2A	10 1010		Not used
Out_Y	r	2B	10 1011	output	
--	r	2C	10 1100		Not used
Out_Z	r	2D	10 1101	output	
Reserved (do not modify)		2E-2F			Reserved
FF_WU_CFG_1	rw	30	11 0000	00000000	
FF_WU_SRC_1(ack1)	r	31	11 0001	00000000	
FF_WU_THS_1	rw	32	11 0010	00000000	
FF_WU_DURATION_1	rw	33	11 0011	00000000	
FF_WU_CFG_2	rw	34	11 0100	00000000	
FF_WU_SRC_2 (ack2)	r	35	11 0101	00000000	
FF_WU_THS_2	rw	36	11 0110	00000000	
FF_WU_DURATION_2	rw	37	11 0111	00000000	
Reserved (do not modify)		38-3F			Reserved

5.1.1 Reserved registers

Registers marked as reserved must not be changed. Random changes of the content of these registers may cause permanent damage to the device.

5.1.2 Registers loaded at boot

The LIS331DL is factory-trimmed. The content of the registers loaded at boot must not be changed. Their content is automatically restored when the device is powered up.

6 About control registers

6.1 CTRL_REG1 (20h)

Control register #1.

DR	PD	FS	ST P	ST M	Zen	Yen	Xen

DR	Data rate selection. Default value: 0 (0: 100 Hz output data rate; 1: 400 Hz output data rate)
PD	Power-down control. Default value: 0 (0: power-down mode; 1: active mode)
FS	Full scale selection. Default value: 0 (0: +/- 2 *g*; 1: +/- 8 *g*)
STP, STM	Self test enable. Default value: 00 (00: normal mode; 10: self test P; 01 self test M; 11: forbidden)
Zen	Z-axis enable. Default value: 1 (0: Z-axis disabled; 1: Z-axis enabled)
Yen	Y-axis enable. Default value: 1 (0: Y-axis disabled; 1: Y-axis enabled)
Xen	X-axis enable. Default value: 1 (0: X-axis disabled; 1: X-axis enabled)

The **DR** bit allows the selection of the data rate at which acceleration samples are produced. The default value is 0, which corresponds to a data rate of 100 Hz. By changing the content of DR to 1, the selected data rate is set equal to 400 Hz.

The **PD** bit allows the device to exit power-down mode. The device is in power-down mode when PD= “0” (default value after boot). The device is in normal mode when PD is set to 1.

DR (Data rate)	PD (Power-down)	Status
0	0	Power-down (default)
0	1	Active (100 Hz output)
1	0	Power-down
1	1	Active (400 Hz output)

The **STP** and **STM** bits are used to activate the self-test function. When one of the bits is set to 1, a change to the device outputs occurs (refer to datasheet for specification), allowing checking of the functionality of the whole measurement chain. STP and STM move the output in opposite directions.

The **Zen** bit enables the generation of a DataReady signal for the Z-axis measurement channel when set to 1. The default value is 1.

The **Yen** bit enables the generation of a DataReady signal for the Y-axis measurement channel when set to 1. The default value is 1.

The **Xen** bit enables the generation of a DataReady signal for the X-axis measurement channel when set to 1. The default value is 1.

6.2 CTRL_REG2 (21h)

Control register #2.

SIM	BOOT	--	FDS	HP FF_WU2	HP FF_WU1	HP coeff2	HP coeff1

SIM	SPI serial interface mode selection. Default value: 0 (0: 4-wire interface; 1: 3-wire interface)
BOOT	Reboot memory content. Default value: 0 (0: normal mode; 1: reboot memory content)
FDS	Filtered data selection. Default value: 0 (0: internal filter bypassed; 1: data from internal filter sent to output register)
HP FF_WU2	High-pass filter enabled for free-fall/wakeup # 2. Default value: 0 (0: filter bypassed; 1: filter enabled)
HP FF_WU1	High-pass filter enabled for free-fall/wake-up #1. Default value: 0 (0: filter bypassed; 1: filter enabled)
HP coeff2 HP coeff1	High-pass filter cut-off frequency configuration. Default value: 00

The **SIM** bit selects the SPI serial interface mode. When SIM is ‘0’ (default value), the 4-wire interface mode is selected. The data coming from the device are sent to the SDO pad. In 3-wire interface mode, the output data are sent to the SDA/SDI pad.

The **BOOT** bit is used to refresh the content of internal registers stored in the Flash memory block. At device power-up, the content of the Flash memory block is transferred to the internal registers related to the trimming functions, to permit good device behavior. If for any reason the content of the trimming registers is changed, this bit can be used to restore the correct values. When the BOOT bit is set to ‘1’, the content of the internal Flash is copied to the corresponding internal registers and used to calibrate the device. These values are factory-trimmed, are different for every accelerometer and normally should not be changed. At the end of the boot process, the BOOT bit is set again to ‘0’.

The **FDS** bit enables (FDS=1) or bypasses (FDS=0) the high-pass filter in the signal chain of the sensor.

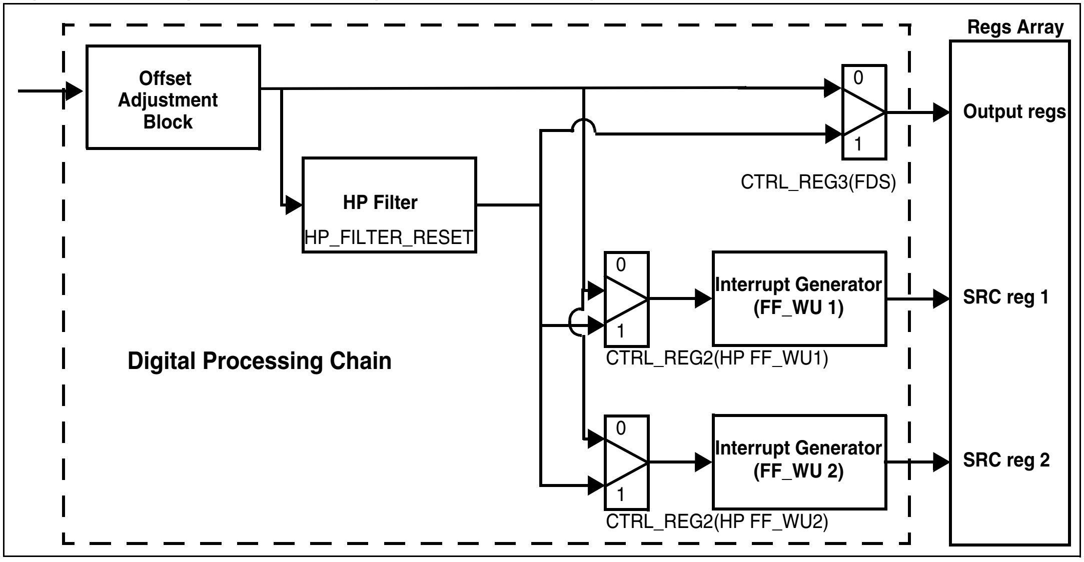
The **HP FF_WU[2:1]** bits enable (HP FF_WU=1) or bypass (HP FF_WU=0) the high-pass filter in interrupt generation blocks.

The **HP coeff[2:1]** bits are used to configure the high-pass filter cut-off frequency ft, accordingly to the following table:

HPcoeff2,1	ft (Hz) (DR=100 Hz)	ft (Hz) (DR=400 Hz)
00	2	8
01	1	4
10	0.5	2
11	0.25	1

Figure 10 shows the block diagram of the digital processing chain and the related control signals.

Figure 10. Digital processing chain block diagram



6.3 CTRL_REG3 (22h)

Control register #3.

IHL	PP_OD	I^2CFG2	I^2CFG1	I^2CFG0	I1CFG2	I1CFG1	I1CFG0

IHL	Interrupt active high, low. Default value 0. (0: active high; 1: active low)
PP_OD	Push-pull/open drain selection on interrupt pad. Default value 0. (0: push-pull; 1: open drain)
I^2CFG2, I^2CFG0	Interrupt 2 configuration bits. Default value 000. (see table below)
I1CFG2, I1CFG0	Interrupt 1 configuration bits. Default value 000. (see table below)

The **IHL** bit selects the polarity of the interrupt signal. When IHL is ‘0’ (default value) any interrupt event is signalled with a logical 1.

The **PP_OD** bit defines whether the interrupt pad should operate in Push-pull or in Open Drain mode. The latter is specifically intended for wired-OR connection of multiple interrupt signals on the same interrupt line. The default value is ‘0’ which corresponds to Push-pull mode.

The **I^{2}CFG[2:0] and I1CFG[2:0]** bits select which signal should be sent out from the INT2 and INT1 interrupt pads as described in the following table:

I^{2}CFG2	**I^{2}CFG1**	**I^{2}CFG0**	**INT2 pin**
I1CFG2	**I1CFG1**	**I1CFG0**	**INT1 pin**
0	0	0	GND
0	0	1	FF_WU 1
0	1	0	FF_WU 2
0	1	1	FF_WU1 or FF_WU2
1	0	0	DataReady
1	1	1	--

Two completely independent interrupt blocks are available in the LIS331DL: FF_WU1 and FF_WU2. They can be configured using the registers described in *Section 8*.

HP filtered data can be selected for further processing by the Interrupt Generator block by setting the desired values for the HP FF_WU1 and HP FF_WU2 bits in CTRL_REG2. The default value is '0' and corresponds to the use of non-filtered data. The output of the Interrupt Generator block is used to load the FF_WU_SRC_1 and FF_WU_SRC_2 registers.

7 Data and status registers

7.1 WHO_AM_I (0Fh)

Device identification register.

0	0	1	1	1	0	1	1

This register contains a device identifier which, for the LIS331DL, is set to 3Bh.

7.2 STATUS_REG (27h)

Data output status register.

ZYXOR	ZOR	YOR	XOR	ZYXDA	ZDA	YDA	XDA

ZYXOR	X, Y and Z axes data overrun. Default value: 0 (0: no overrun has occurred; 1: new data has overwritten the previous data before it was read)
ZOR	Z-axis data overrun. Default value: 0 (0: no overrun has occurred; 1: new data for the Z-axis has overwritten the previous data)
YOR	Y-axis data overrun. Default value: 0 (0: no overrun has occurred; 1: new data for the Y-axis has overwritten the previous data)
XOR	X-axis data overrun. Default value: 0 (0: no overrun has occurred; 1: new data for the X-axis has overwritten the previous data)
ZYXDA	X, Y and Z-axis new data available. Default value: 0 (0: a new set of data is not yet available; 1: a new set of data is available)
ZDA	Z-axis new data available. Default value: 0 (0: new data for the Z-axis is not yet available; 1: new data for the Z-axis is available)
YDA	Y-axis new data available. Default value: 0 (0: new data for the Y-axis is not yet available; 1: new data for the Y-axis is available)
XDA	X-axis new data available. Default value: 0 (0: new data for the X-axis is not yet available; 1: new data for the X-axis is available)

ZYXOR is set to one whenever new acceleration data is produced before completing the retrieval of the previous set. When this occurs, the content of at least one acceleration data register (i.e. OUTX, OUTY, OUTZ) has been overwritten. ZYXOR is cleared when the acceleration data (OUTX, OUTY, OUTZ) of all the active channels are read.

ZOR is set to 1 whenever a new acceleration sample related to the Z-axis is generated before the retrieval of the previous sample. When this occurs, the previous sample is overwritten. ZOR is cleared anytime the OUTZ register is read.

YOR is set to 1 whenever a new acceleration sample related to the Y-axis is generated before the retrieval of the previous sample. When this occurs, the previous sample is overwritten. YOR is cleared anytime the OUTY_H register is read.

XOR is set to 1 whenever a new acceleration sample related to the X-axis is generated before the retrieval of the previous sample. When this occurs, the previous sample is overwritten. XOR is cleared anytime the OUTX_H register is read.

The ZYXDA bit signals that a new sample for all the enabled channels is available. ZYXDA is cleared when the acceleration data (OUTX, OUTY, OUTZ) of all the enabled channels are read.

ZDA is set to 1 whenever a new acceleration sample related to the Z-axis is available. ZDA is cleared anytime the OUTZ register is read. In order to trigger, the ZDA bit requires the Z-axis to be enabled (bit Zen=1 in CTRL_REG1).

YDA is set to 1 whenever a new acceleration sample related to the Y-axis is available. YDA is cleared anytime the OUTY register is read. In order to trigger, the YDA bit requires the Y-axis to be enabled (bit Yen=1 in CTRL_REG1).

XDA is set to 1 whenever a new acceleration sample related to the X-axis is available. XDA is cleared anytime the OUTX register is read. In order to trigger, the XDA bit requires the X-axis to be enabled (bit Xen=1 in CTRL_REG1).

7.3 OUTX (29h)

X-axis output register.

XD7	XD6	XD5	XD4	XD3	XD2	XD1	XD0

7.4 OUTY (2Bh)

Y-axis output register.

YD7	YD6	YD5	YD4	YD3	YD2	YD1	YD0

7.5 OUTZ (2Dh)

Z-axis output register.

ZD7	ZD6	ZD5	ZD4	ZD3	ZD2	ZD1	ZD0

.

8 Free-fall and wake-up registers

The following sections describe the registers that are involved in the generation of the interrupt signals associated with inertial wake-up and free-fall events.

8.1 HP_FILTER_RESET (23h)

Dummy register. A reading at this address forces the high-pass filter to recover instantaneously the dc level of the acceleration signal provided to its inputs. After the above reading, the output of the high-pass filter is zero.

X	X	X	X	X	X	X	X

8.2 FF_WU_CFG_1 (30h)

Free-fall and wake-up configuration register for interrupt source 1.

AOI	LIR	ZHIE	ZLIE	YHIE	YLIE	XHIE	XLIE

AOI	And/Or combination of interrupt events. Default value: 0 (0: OR combination of interrupt events; 1: AND combination of interrupt events)
LIR	Latch interrupt request to FF_WU_SRC reg with the FF_WU_SRC reg cleared by reading FF_WU_SRC_1 reg. Default value: 0 (0: interrupt request not latched; 1: interrupt request latched)
ZHIE	Enable interrupt generation on Z high event. Default value: 0 (0: disable interrupt request; 1: enable interrupt request on measured accel. value higher than preset threshold)
ZLIE	Enable interrupt generation on Z low event. Default value: 0 (0: disable interrupt request; 1: enable interrupt request on measured accel. value lower than preset threshold)
YHIE	Enable interrupt generation on Y high event. Default value: 0 (0: disable interrupt request; 1: enable interrupt request on measured accel. value higher than preset threshold)
YLIE	Enable interrupt generation on Y low event. Default value: 0 (0: disable interrupt request; 1: enable interrupt request on measured accel. value lower than preset threshold)
XHIE	Enable interrupt generation on X high event. Default value: 0 (0: disable interrupt request; 1: enable interrupt request on measured accel. value higher than preset threshold)
XLIE	Enable interrupt generation on X low event. Default value: 0 (0: disable interrupt request; 1: enable interrupt request on measured accel. value lower than preset threshold)

The **AOI** bit allows selection of wake-up (OR combination of interrupt events) or free-fall (AND combination of interrupt events) detection

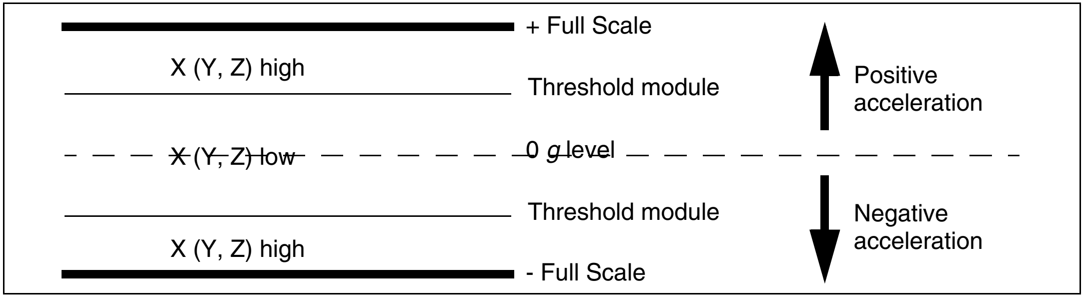
LIR defines whether the configured interrupt event should be latched by the device once it has occurred. The interrupt request is cleared by reading the related source register (FF_WU_SRC_1).

XHIE (YHIE, ZHIE) set an interrupt event to occur when the measured acceleration data on the X (Y, Z) channel is higher than the threshold set in the FF_WU_THS_1 register.

XLIE (YLIE, ZLIE) set an interrupt event to occur when the measured acceleration data on the X (Y, Z) channel is lower than the threshold set in the FF_WU_THS_1 register.

The threshold module used by the system to detect free-fall or inertial wake-up events is defined by FF_WU_THS_1. The threshold value is expressed over 7 bits as an unsigned number and X, (Y, Z) high is true when the unsigned acceleration value of the X (Y, Z) channel is higher than or equal to FF_WU_THS_1. Similarly, X, (Y, Z) low is true when the unsigned acceleration value of the X (Y, Z) channel is lower than FF_WU_THS_1. Refer to *Figure 11* for more details.

Figure 11. FF_WU_CFG_1,2 high and low value



8.3 FF_WU_SRC_1 (31h)

Free-fall and wake-up source register for interrupt 1. Read only register.

x	IA	ZH	ZL	YH	YL	XH	XL

IA	Interrupt active. Default value: 0 (0: no interrupt has been generated; 1: one or more interrupt event has been generated)
ZH	Z high. Default value: 0 (0: no interrupt; 1: ZH event has occurred)
ZL	Z low. Default value: 0 (0: no interrupt; 1: ZL event has occurred)
YH	Y high. Default value: 0 (0: no interrupt; 1: YH event has occurred)
YL	Y low. Default value: 0 (0: no interrupt; 1: YL event has occurred)
XH	X high. Default value: 0 (0: no interrupt; 1: XH event has occurred)
XL	X low. Default value: 0 (0: no interrupt; 1: XL event has occurred)

This register keeps track of the acceleration event triggering (or that has triggered, if the LIR bit in FF_WU_SRC_1 reg is set to 1) the interrupt signal. In particular, IA is equal to 1 when the combination of acceleration events specified in FF_WU_CFG_1 register is true. This bit is used for the generation of the interrupt signal associated with free-fall/wake-up events.

X, (Y, Z) high is true when the module of the acceleration value of the X (Y, Z) channel is higher than the preset threshold which is defined as the concatenation of FF_WU_THS_1.

Similarly, X, (Y, Z) low is true when the module of the acceleration value of the X (Y, Z) channel is lower than FF_WU_THS_1.

Reading at this address clears the FF_WU_SRC_1 register and the FF_WU1 interrupt and allows the refreshment of data in the FF_WU_SRC_1 register itself, if the latched option was chosen.

8.4 FF_WU_THS_1 (32h)

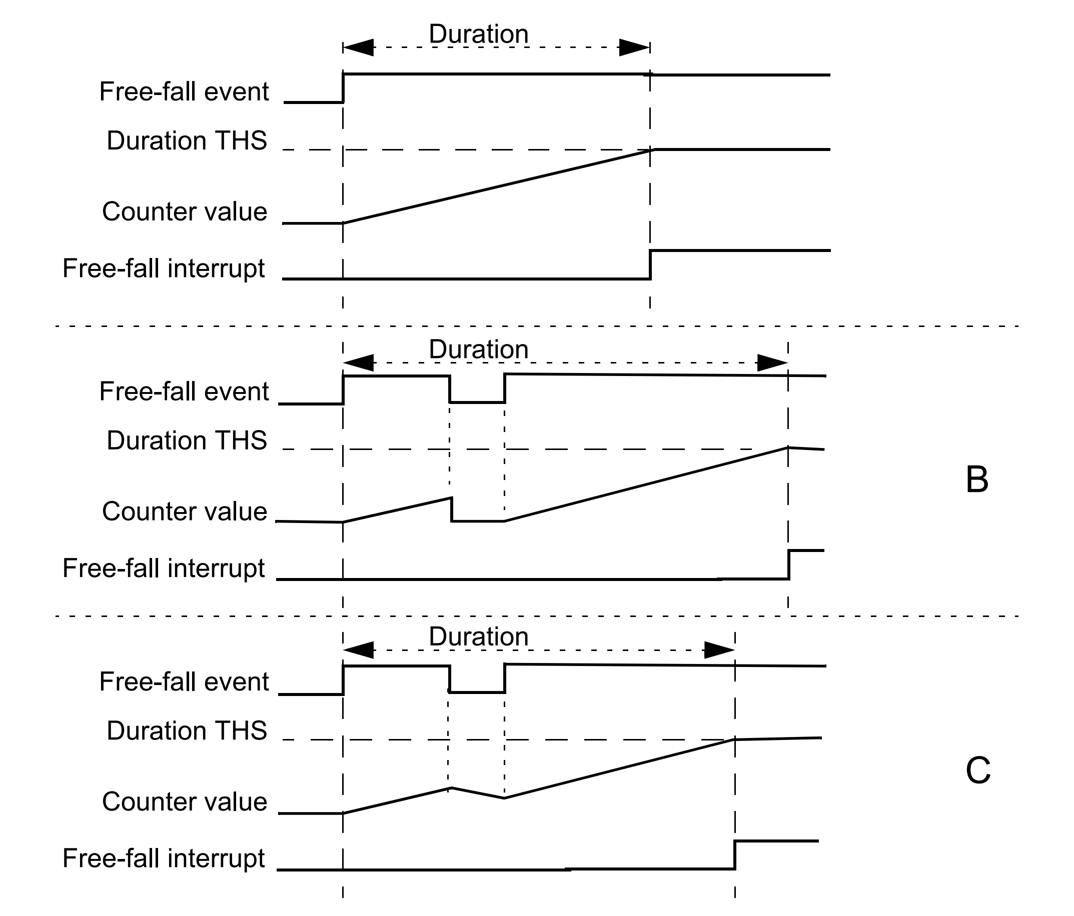
Free-fall, wake-up threshold for interrupt 1.

DCRM	THS6	THS5	THS4	THS3	THS2	THS1	THS0

DCRM	Duration counter reset mode selection. Default value: 0 (0: counter reset; 1: counter decremented)
THS6, THS0	Free-fall / wake-up threshold: default value: 000 0000

The **DCRM** bit allows selection of the way in which the duration counter is reset. When DCRM is ‘0’, the duration counter is reset immediately whenever the internal inertial event programmed by the user is not active (*Figure 12* part B) while it is decremented when DCRM is set to 1 (*Figure 12* part C). The latter configuration allows the filtering out of spurious spikes which might impair the recognition and validation of inertial events.

Figure 12. DCRM bit function



8.5 FF_WU_DURATION_1 (33h)

Set the minimum duration of the free-fall, wake-up event that must be recognized by the LIS331DL.

D7	D6	D5	D4	D3	D2	D1	D0

D7, D0	Duration value. Default value: 0000 0000

D7, D0 define the minimum duration of the programmed inertial event such as free-fall and wake-up that must be recognized by the device. Duration increments and maximum value depend on the ODR chosen. For an output data rate of 400 Hz, the register allows setting a duration spanning from 0 to 637.5 ms in increments of 2.5 ms. Conversely, when the output data rate is set to 100 Hz, it is possible to define an event duration spanning from 0 to 2.55 seconds in increments of 10 ms. The counter used to implement the duration function is blocked when the LIR bit in the configuration register is set to one and the interrupt event has occurred.

8.6 FF_WU_CFG_2 (34h)

Free-fall and wake-up configuration register for interrupt source 2.

AOI	LIR	ZHIE	ZLIE	YHIE	YLIE	XHIE	XLIE

AOI	And/Or combination of interrupt events. Default value: 0 (0: OR combination of interrupt events; 1: AND combination of interrupt events)
LIR	Latch interrupt request to FF_WU_SRC reg with the FF_WU_SRC reg cleared by reading FF_WU_SRC_1 reg. Default value: 0 (0: interrupt request not latched; 1: interrupt request latched)
ZHIE	Enable interrupt generation on Z high event. Default value: 0 (0: disable interrupt request; 1: enable interrupt request on measured accel. value higher than preset threshold)
ZLIE	Enable interrupt generation on Z low event. Default value: 0 (0: disable interrupt request; 1: enable interrupt request on measured accel. value lower than preset threshold)
YHIE	Enable interrupt generation on Y high event. Default value: 0 (0: disable interrupt request; 1: enable interrupt request on measured accel. value higher than preset threshold)
YLIE	Enable interrupt generation on Y low event. Default value: 0 (0: disable interrupt request; 1: enable interrupt request on measured accel. value lower than preset threshold)
XHIE	Enable interrupt generation on X high event. Default value: 0 (0: disable interrupt request; 1: enable interrupt request on measured accel. value higher than preset threshold)
XLIE	Enable interrupt generation on X low event. Default value: 0 (0: disable interrupt request; 1: enable interrupt request on measured accel. value lower than preset threshold)

Two structures can be used to generate interrupts. They have exactly the same configuration capability and are completely independent.

The interrupt result of each structure can be sent with indifference to the INT1 or INT2 pins using the CTRL_REG3 register.

8.7 FF_WU_SRC_2 (35h)

Free-fall and wake-up source register for interrupt 2. Read-only register. Refer to FF_WU_SRC_1 for details.

x	IA	ZH	ZL	YH	YL	XH	XL

IA	Interrupt active. Default value: 0 (0: no interrupt has been generated; 1: one or more interrupt event has been generated)
ZH	Z high. Default value: 0 (0: no interrupt; 1: ZH event has occurred)
ZL	Z low. Default value: 0 (0: no interrupt; 1: ZL event has occurred)
YH	Y high. Default value: 0 (0: no interrupt; 1: YH event has occurred)
YL	Y low. Default value: 0 (0: no interrupt; 1: YL event has occurred)
XH	X high. Default value: 0 (0: no interrupt; 1: XH event has occurred)
XL	X low. Default value: 0 (0: no interrupt; 1: XL event has occurred)

Reading at this address clears the FF_WU_SRC_2 register and the FF_WU2 interrupt and allows the refreshment of data in the FF_WU_SRC_2 register if the latched option was chosen.

8.8 FF_WU_THS_2 (36h)

Free-fall, wake-up threshold for interrupt source 2.

The

DCRM	THS6	THS5	THS4	THS3	THS2	THS1	THS0

DCRM	Duration counter reset mode selection. Default value: 0 (0: counter reset; 1: counter decremented)
THS6, THS0	Free-fall / wake-up threshold: default value: 000 0000

The **DCRM** bit and THS6 to THS0 behave exactly like the bit in FF_WU_THS_1 but they are applied to interrupt source 2.

8.9 FF_WU_DURATION_2 (37h)

Set the minimum duration of the free-fall, wake-up event that must be recognized by the LIS331DL.

D7	D6	D5	D4	D3	D2	D1	D0

D7, D0	Duration value. Default value: 0000 0000

Refer to FF_WU_DURATION_1 for details.

9 Application information

9.1 Start-up sequence

Once the device is powered up, it automatically downloads the calibration coefficients from the embedded Flash to the internal registers. When the boot procedure is complete (after approximately 3 ms) the device automatically enters power-down mode.

To turn-on the device and gather acceleration data, it is necessary to write 47h in the CTRL_REG1. With this command the three acceleration channels (i.e. X, Y and Z-axis) are enabled and the output data rate is set to 100 Hz.

9.2 Reading acceleration data

9.2.1 Using the status register

The device is provided with a STATUS_REG which should be polled to check when a new set of data is available. The reading procedure should be the following:

```
1    read STATUS_REG
2    if STATUS_REG(3)=0 then goto 1
3    if STATUS_REG(7)=1 then some data have been overwritten
4    read OUTX
5    read OUTY
6    read OUTZ
7    data processing
8    goto 1
```

The check performed at step 3 permits understanding of whether the reading rate is adequate compared to the data production rate. In case one or more acceleration samples have been overwritten by new data because of a too slow reading rate, the bit STATUS_REG(7) is set to 1.

The overrun bits are automatically cleared when all the data present inside the device have been read and new data have not been produced in the meantime.

9.2.2 Using the DataReady signal

The device may be configured to generate one HW signal (DataReady) on either pin 9 or 11, to flag that a new set of measurement data is available for reading. This signal corresponds to the ZYXDA bit present in the STATUS_REG. The polarity of the signal is defined through the IHL bit present in CTRL_REG3 and is deasserted when the acceleration data of all the enabled channels have been read. To enable the DataReady signal on pin 11 (corresponding to INT1), it is necessary to set the I1CFG[2:0] bit present in CTRL_REG3 to 100. Conversely, to enable the DataReady signal on pin 9 (corresponding to INT2) it is necessary to set the I^{2}CFG[2:0] bit present in the CTRL_REG3 to 100.

9.3 Understanding acceleration data

The measured acceleration data are sent to the OUTX, OUTY and OUTZ registers. The acceleration values are expressed as a 2’s complement number. When the full-scale is set to 2 *g*, each LSB corresponds to 18 m*g*.

The table below provides a few basic examples of the data that is read in the data registers when the device is subject to a given acceleration. The values listed in the table are given under the hypothesis of perfect device calibration (i.e. no offset, no gain error, etc.) and rounded to the closest integer.

Table 5. Output data register content vs. acceleration

Acceleration values	**FS bit = 0**	**FS bit = 1**
	Output register content	
0 *g*	00h	00h
350 m*g*	14h	05h
1 *g*	38h	0Eh
2 *g*	6Fh	1Ch
-350 m*g*	ECh	FBh
-1 *g*	C8h	F2h
-2 *g*	91h	E4h

9.4 Interrupt generation description

The LIS331DL provides two fully-programmable interrupt sources which may be configured to trigger different inertial events. Among them, it is worth mentioning, is the recognition of free-fall and wake-up events. When an interrupt condition is verified, the interrupt signal is asserted and by reading either the FF_WU_SRC_1 or FF_WU_SRC_2 registers, it is possible to understand which condition has occurred.

The block diagram of the interrupt block is shown below.

Figure 13. Interrupt generation block diagram

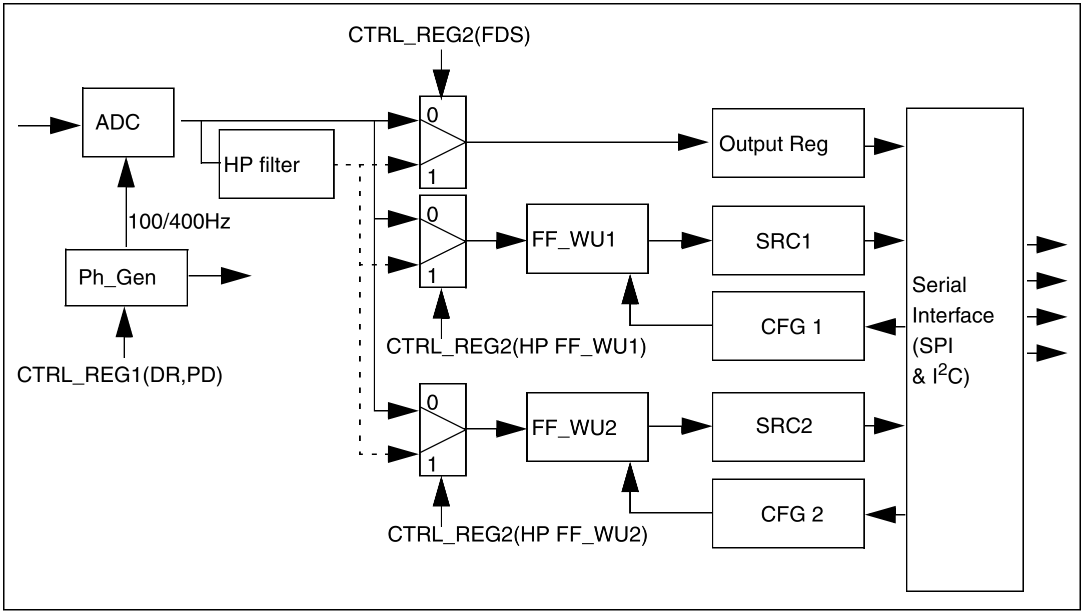
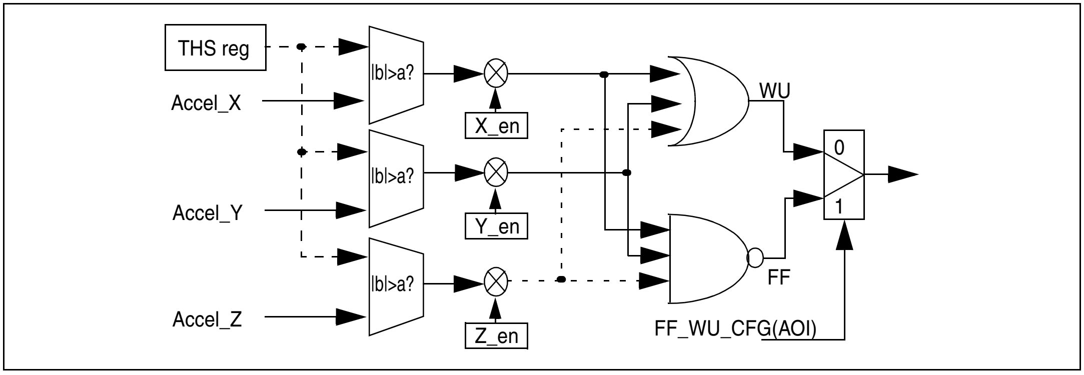


Figure 14. Free-fall, wake-up interrupt generator



FF or WU interrupt generation is selected for each interrupt generation channel through the AOI bit in the corresponding FF_WU_CFG register. If the AOI bit is ‘0’, signals coming from the comparators are put in logical OR. Depending on the values written in the FF_WU_CFG_1 and FF_WU_CFG_2 registers, each time the value of at least one of the enabled axes exceeds the threshold written in the module in the corresponding FF_WU_THS_1 and FF_WU_THS_2 registers, a FF, WU interrupt is generated. Otherwise, if the AOI bit is ‘1’, signals coming from the comparators go into a “NAND” port. In this case, an interrupt signal is generated whenever the acceleration signal of all the enabled axes is below the threshold written in the FF_WU_THS_1 or FF_WU_THS_2 registers.

The FF_WU_CFG(LIR) bit permits the determination of whether or not the interrupt request should be latched. If the LIR bit is ‘0’ (default value), the interrupt signal goes high when the interrupt condition is satisfied and comes back low immediately if the interrupt condition is no longer verified. Otherwise, if the LIR bit is ‘1’, whenever an interrupt condition is applied the interrupt signal remains high even if the condition returns to a non-interrupt status, until a reading to the FF_WU_SRC register is performed.

The remaining bits of the FF_WU_CFG register allow the decision as to which axis the interrupt should be performed on and in which direction the threshold must be exceeded to generate the interrupt request.

9.5 Inertial wake-up

9.5.1 HP filter bypassed

This paragraph provides a basic algorithm which shows the practical use of the inertial wake-up feature. In particular, with the code below, the device is configured to recognize when the absolute acceleration along either X or Y axis exceeds a preset threshold (180 m*g* is used in the example). The event which triggers the interrupt is latched inside the device using FF_WU interrupt source 1 and its occurrence is signalled through the use of the INT2 pin.

```
 1   write C7h into CTRL_REG1                  // Turn-on the sensor and set ODR=400Hz
 2   write 00h into CTRL_REG2                  // Default value: high-pass filter bypassed
 3   write 08h into CTRL_REG3                  // FF_WU1 interrupt sent to INT2 pin
 4   write 0Ah into FF_WU_THS_1 reg            // Set wake-up threshold = 180 mg
 5   write 00h into FF_WU_DURATION_1 reg       // No filtering/confirmation on the event
 6   write 4Ah into FF_WU_CFG_1                // Configure desired wake-up event
 7   poll INT2 pin; if INT2=0 then goto 7      // Poll INT2 pin waiting for the wake-up event
                                               // Return the event that has triggered the
 8   read FF_WU_SRC_1 reg                      interrupt
                                               // Clear interrupt request
     (Wake-up event has occurred; insert your code
 9   here)                                     // Event handling
10   goto 7
```

9.5.2 Using the HP filter

The code provided below gives a basic routine which shows the practical use of the inertial wake-up feature performed on high-pass filtered data. In particular, the device is configured to recognize when the high-frequency component of the acceleration applied along either the X, Y or Z axis exceeds a preset threshold (180 m*g* is used in the example). The event which triggers the interrupt is latched inside the device using FF_WU interrupt source 1 and its occurrence is signalled through the usage of the INT1 pin.

```
 1   write C7h into CTRL_REG1                      // Turn-on the sensor and set ODR=400 Hz
 2   write 04h into CTRL_REG2                      // High-pass filter enabled on FF_WU1, fcut-
                                                   off= 8Hz
 3   write 01h into CTRL_REG3                      // FF_WU1 interrupt sent to INT1 pin
 4   write 0Ah into FF_WU_THS_1 reg                // Set wake-up threshold = 180 mg
 5   write 00h into FF_WU_DURATION_1 reg           // No filtering/confirmation on the event
                                                   // Dummy read to force the HP filter to
 6   read HP_FILTER_RESET register                 // actual acceleration value
                                                   // (i.e. set reference acceleration/tilt value)
 7   write 6Ah into FF_WU_CFG_1                    // Configure desired wake-up event
 8   poll INT1 pin; if INT1=0 then goto 7          // Poll INT1 pin waiting for the wake-up event
 9   (Wake-up event has occurred; insert your code // Event handling
     here)
                                                   // Return the event that has triggered the
10   read FF_WU_SRC_1 reg                          interrupt
                                                   // Clear interrupt request
11   (Insert your code here)                       // Event handling
12   goto 8
```

At step 6, a dummy read at the HP_FILTER_RESET register is performed to set the current/reference acceleration/tilt state against which the device performed the threshold comparison.

This read may be performed any time required to set the orientation/tilt of the device as a reference state without waiting for the filter to settle.

9.6 Free-fall detection

9.6.1 Roll function not used

This paragraph provides the basics for the use of the free-fall detection feature. The SW routine that configures the device to detect free-fall events and to signal them is as follows:

```
1   write 47h into CTRL_REG1                    // Turn-on the sensor and set ODR=100 Hz
2   write 00h into CTRL_REG2                    // Default value: high-pass filter bypassed
3   write 08h into CTRL_REG3                    // FF_WU1 interrupt sent to INT2 pin
4   write 14h into FF_WU_THS_1 reg              // Set the free-fall threshold
5   write 09h into FF_WU_DURATION_1 reg         // Set minimum event duration to 90 ms
6   write D5h into FF_WU_CFG_1                  // Configure free-fall recognition and latch
                                                // interrupt request
7   poll INT2 pad; if INT2=0 then goto 7        // Poll INT2 pin waiting for the free-fall event
8   (Free-fall event has occurred; insert your code
    here)                                       // Event handling
9   read FF_WU_SRC_1 register                   // Clear interrupt request
10  goto 7
```

The code sample exploits a threshold set at 350 m*g* for free-fall recognition and the event is notified by the hardware signal INT2 pin. At step 5, the FF_WU_DURATION_1 register is configured to ignore events that are shorter than 9/ODR=9/100~90 ms (ODR=output data rate) in order to avoid false detections.

Once the free-fall event has occurred, a dummy read at FF_WU_SRC_1 reg clears the request and the device is ready to recognize other events.

9.6.2 Roll function applied

The roll function can be added to free-fall condition recognition using both interrupt sources. Interrupt signals can be sent to two different pins (INT1, INT2) or can be sent to the same pin so that the first condition verified controls the pin (logic “OR” of interrupt signals).

9.7 Output data rate selection and reading timing

The output data rate is user-selectable through the DR bit stored in the CTRL_REG1 (20h) register. At power-on-reset, DR is reset to 0, thus providing a default output data rate set to 100 Hz.

The selectable output data rates are given in *Table 6* below:

Table 6. Output data rate

DR	Output data rate
0	100 Hz
1	400 Hz

The output data rate precision is related to internal oscillator or to external clock precision, and an error of +/- 10% should be taken into account.

The output data rate precision is related to internal oscillator or to external clock precision, and an error of +/- 10% should be taken into account.

A minimum reading period 150 µs shorter than the output data rate period is defined to avoid the loss of any data produced. During this time period the reading of the data must be performed and the DataReady signal can be used as a trigger to begin the reading sequence. At the end of the complete sequence, the DataReady signal goes down and the following rising edge advises that new data are available. If this minimum reading frequency is not observed, it is possible to lose some data and the DataReady signal loses its function as a trigger signal. The status register can be used to infer whether an overrun condition has occurred.

Figure 15. Reading timing

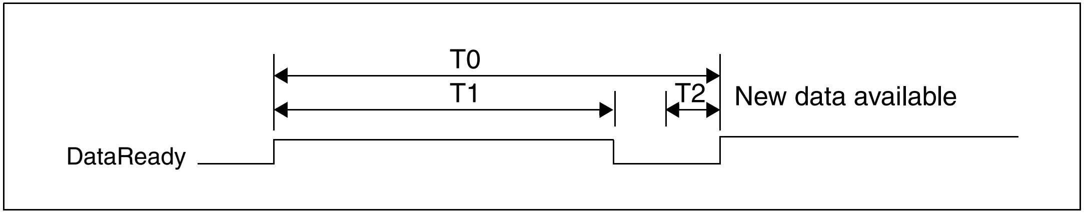


Table 7. Timing value to avoid data loss

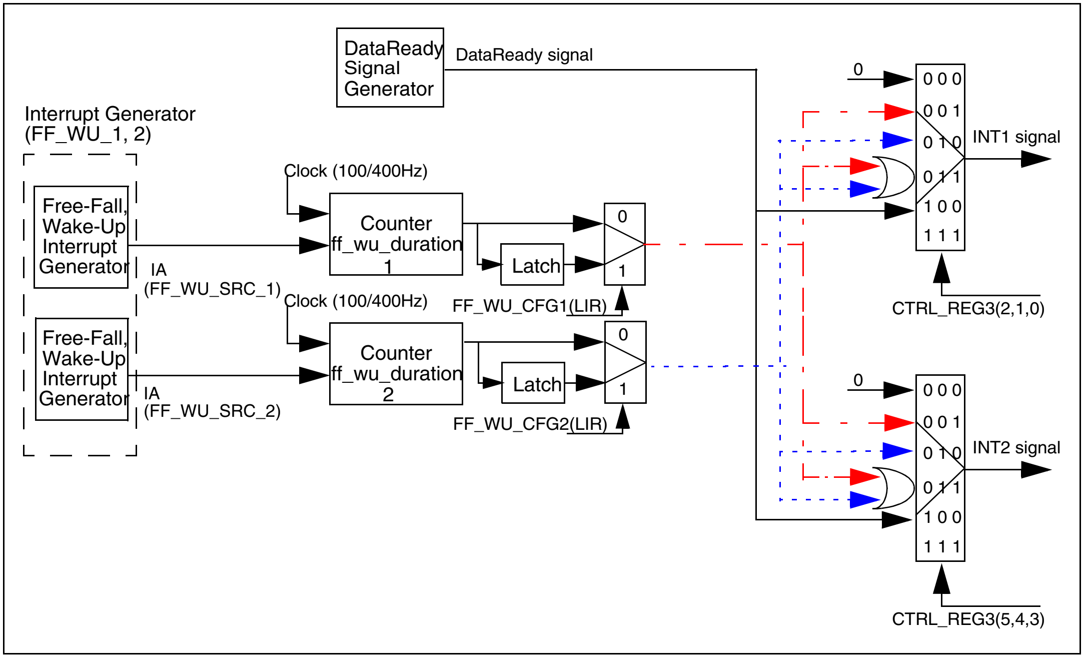
Time	Description	Min.	Typ.	Max.
T0	Data rate	2.5 ms	-	10 ms
T1	Reading period		-	T0-T2
T2	New data generation	150 µs	-	

9.8 DataReady vs. interrupt signal

The device includes two pins which can be activated to generate either the DataReady or the interrupt signal. The functionality of the pins is selected by setting the interrupt configuration bits located in CTRL_REG3(2-0) for interrupt pin 1 (INT1) and in CTRL_REG3(5-3) for interrupt pin 2 (INT2), in accordance with the block diagram given in *Figure 16*.

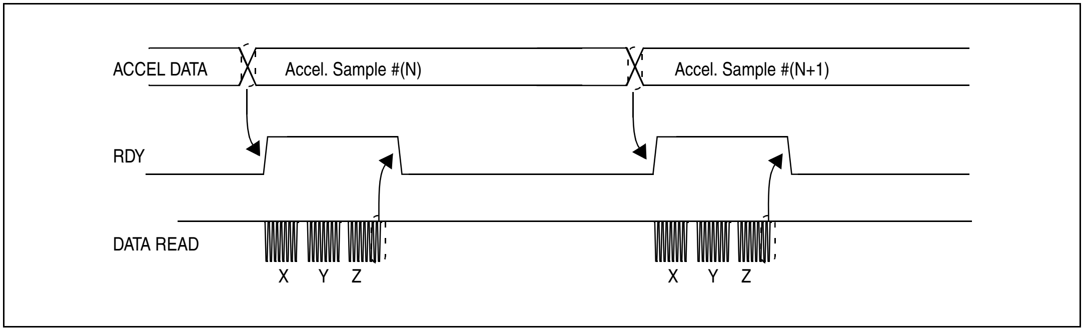
The DataReady signal stored in STATUS_REG(3), which indicates when a new set of acceleration data is ready, is made available by setting the cfg bit to 100. The interrupt source 1 signal is sent to the pin when the cfg bit is set to 001 and the interrupt source 2 when the cfg bit is set to 010. A logic "OR" combination of interrupt source 1 and 2 can be sent to an interrupt pin by writing 011 in the cfg bit.

Figure 16. Interrupt and DataReady signal generation block diagram



The DataReady signal rises to 1 when a new set of acceleration data has been generated and is available for reading. The signal is reset after all the enabled channels are read through the serial interface.

Figure 17. DataReady signal



10 Revision history

Table 8. Document revision history

Date	Revision	Changes
12-Jun-2009	1	Initial release.

Please Read Carefully:

Information in this document is provided solely in connection with ST products. STMicroelectronics NV and its subsidiaries (“ST”) reserve the right to make changes, corrections, modifications or improvements, to this document, and the products and services described herein at any time, without notice.

All ST products are sold pursuant to ST’s terms and conditions of sale.

Purchasers are solely responsible for the choice, selection and use of the ST products and services described herein, and ST assumes no liability whatsoever relating to the choice, selection or use of the ST products and services described herein.

No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted under this document. If any part of this document refers to any third party products or services it shall not be deemed a license grant by ST for the use of such third party products or services, or any intellectual property contained therein or considered as a warranty covering the use in any manner whatsoever of such third party products or services or any intellectual property contained therein.

UNLESS OTHERWISE SET FORTH IN ST’S TERMS AND CONDITIONS OF SALE ST DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY WITH RESPECT TO THE USE AND/OR SALE OF ST PRODUCTS INCLUDING WITHOUT LIMITATION IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE (AND THEIR EQUIVALENTS UNDER THE LAWS OF ANY JURISDICTION), OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

UNLESS EXPRESSLY APPROVED IN WRITING BY AN AUTHORIZED ST REPRESENTATIVE, ST PRODUCTS ARE NOT RECOMMENDED, AUTHORIZED OR WARRANTED FOR USE IN MILITARY, AIR CRAFT, SPACE, LIFE SAVING, OR LIFE SUSTAINING APPLICATIONS, NOR IN PRODUCTS OR SYSTEMS WHERE FAILURE OR MALFUNCTION MAY RESULT IN PERSONAL INJURY, DEATH, OR SEVERE PROPERTY OR ENVIRONMENTAL DAMAGE. ST PRODUCTS WHICH ARE NOT SPECIFIED AS "AUTOMOTIVE GRADE" MAY ONLY BE USED IN AUTOMOTIVE APPLICATIONS AT USER’S OWN RISK.

Resale of ST products with provisions different from the statements and/or technical features set forth in this document shall immediately void any warranty granted by ST for the ST product or service described herein and shall not create or extend in any manner whatsoever, any liability of ST.

STMicroelectronics group of companies

Australia - Belgium - Brazil - Canada - China - Czech Republic - Finland - France - Germany - Hong Kong - India - Israel - Italy - Japan -
Malaysia - Malta - Morocco - Philippines - Singapore - Spain - Sweden - Switzerland - United Kingdom - United States of America

www.st.com