AN2041
APPLICATION NOTE

LIS3LV02DQ: 3-AXIS - ±2g/±6g DIGITAL OUTPUT LOW VOLTAGE LINEAR ACCELEROMETER

This document is intended to provide application note for the low-voltage 3-axis digital output linear MEMS accelerometer provided in QFN-28 package.

1 DESCRIPTION

The LIS3LV02DQ is a three axes digital output linear accelerometer that includes a sensing element and an IC interface able to take the information from the sensing element and to provide the measured acceleration signals to the external world through an I^{2}C/SPI serial interface.

The sensing element, capable of detecting the acceleration, is manufactured using a dedicated process developed by ST to produce inertial sensors and actuators in silicon.

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

The LIS3LV02DQ has a user selectable full scale of ±2g, ±6g and it is capable of measuring acceleration over a bandwidth of 640 Hz for all axes. The device bandwidth may be selected accordingly to the application requirements. A self-test capability allows the user to check the functioning of the system.

The device may be configured to generate an inertial wake-up/free-fall interrupt signal when a programmable acceleration threshold is crossed at least in one of the three axes.

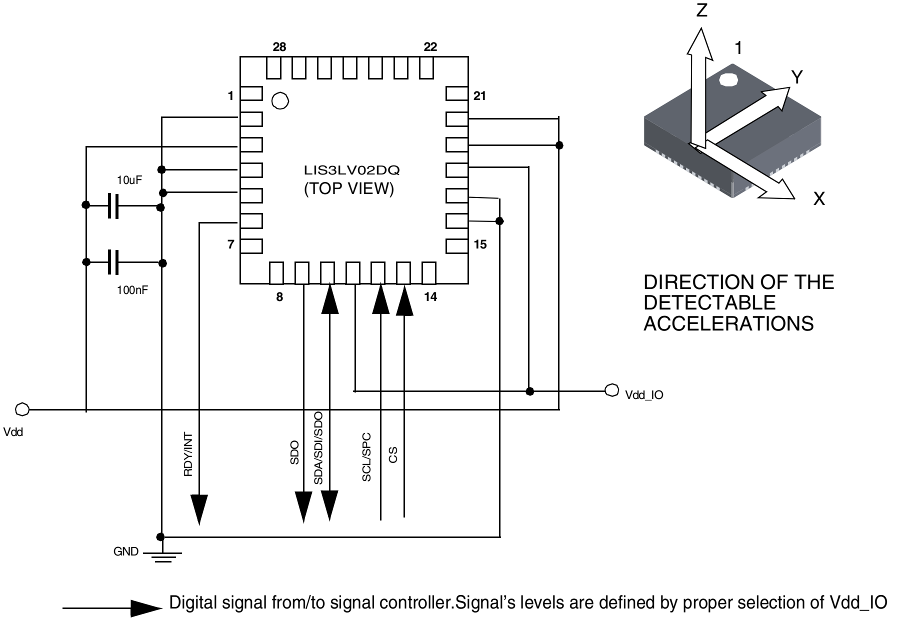
The LIS3LV02DQ is specified over a temperature range extending from -40°C to +85°C and it is provided in a plastic Quad Flat No-lead (QFN) package. This is a leadless package based on copper lead frame which exploits half-encapsulation technology to expose the rear side of the die pad and the tiny fingers for the connection with the PCB.

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

2 ELECTRICAL CONNECTION

The typical electrical connection of the LIS3LV02DQ is shown in Figure 1.

Figure 1. LIS3LV02DQ ELECTRICAL CONNECTION


28
22
1
21
7
15
8
14
LIS3LV02DQ
(TOP VIEW)
10uF
100nF
Vdd
Vdd_IO
GND
RDY/INT
SDO
SDA/SDI/SDO
SCL/SPC
CS
Z
1
Y
X
DIRECTION OF THE DETECTABLE ACCELERATIONS
Digital signal from/to signal controller.Signal's levels are defined by proper selection of Vdd_IO

The device core is supplied through Vdd line (Vdd typ=2.5V) while the I/O pads are supplied through Vdd_IO line. Power supply decoupling capacitors (100 nF ceramic, 10 µF Al) should be placed as near as possible to the pin 3 of the device.

Both the voltage supplies must be present at the same time to have proper behavior of the IC. It is possible to remove Vdd mantaining Vdd_IO without blocking the communication busses.

The central die pad and pin #1 indicator are physically connected to GND.

The functionality of the device and the measured acceleration data are selectable/accessible through the I^{2}C/SPI interface.

When using the I^{2}C, CS must be tied high while SDO must be left floating.

3 SOLDERING INFORMATION

The QFN-28 package is lead free and green package qualified for soldering heat resistance according to JEDEC J-STD-020C. Central die pad and pin #1 indicator are physically connected to GND. Land pattern and soldering recommendations are available upon request.

4 ABSOLUTE MAXIMUM RATINGS

Stresses above those listed as “absolute maximum ratings” may cause permanent damage to the device. This is a stress rating 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 pins Supply voltage[1]	-0.3 to Vdd +0.1	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)	3000g for 0.5 ms	
		10000g for 0.1 ms	
A_{UNP}	Acceleration (Any axis, Unpowered)	3000g for 0.5 ms	
		10000g 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	4.0 (HBM)	kV
		200 (MM)	V
		1.5 (CDM)	kV

Note: 1. Supply voltage on any pin should never exceed 6.0V

Warning: This is a ESD sensitive device, improper handling can cause permanent damages to the part.

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

5 THEORY OF OPERATION

The LIS3LV02DQ is a high performance, low-power, digital output 3-axis linear accelerometer packaged in a QFN package. The complete device includes a sensing element and an IC interface able to take the information from the sensing element and to provide a signal to the external world through an I^2C/SPI serial interface.

A proprietary process is used to create a surface micro-machined accelerometer. The technology allows to carry out suspended silicon structures which are attached to the substrate in a few points called anchors and are free to move in the direction of the sensed acceleration. To be compatible with the traditional packaging techniques a cap is placed on top of the sensing element to avoid blocking 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.

At steady state the nominal value of the capacitors are few pF and when an acceleration is applied the maximum variation of the capacitive load is up to 100fF.

The complete measurement chain is composed by a low-noise capacitive amplifier which converts into an analog voltage the capacitive unbalancing of the MEMS sensor and by three ΣΔ analog-to-digital converters, one for each axis, that translate the produced signal into a digital bitstream. The ΣΔ converters are tightly coupled with dedicated reconstruction filters which remove the high frequency components of the quantization noise and provide low rate and high resolution digital words. The charge amplifier and the ΣΔ converters are operated respectively at 61.5 kHz and 20.5 kHz. The data rate at the output of the reconstruction depends on the user selected Decimation Factor (DF) and spans from 40 Hz to 2560 Hz.

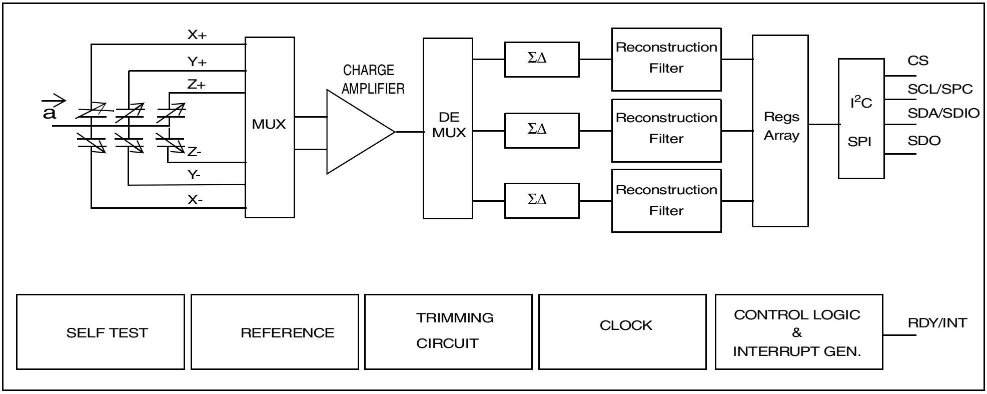
The acceleration data may be accessed through an I^{2}C/SPI interface thus making the device particularly suitable for direct interfacing with a microcontroller.

The LIS3LV02DQ features a Data-Ready signal (RDY) which indicates when a new set of measured acceleration data is available thus simplifying data synchronization in digital system employing the device itself. The LIS3LV02DQ may also be configured to generate an inertial Wake-Up, Direction Detection and Free-Fall interrupt signal accordingly to a programmed acceleration event along the enabled axes.

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

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

Figure 2. DEVICE BLOCK DIAGRAM



6 POWER SUPPLY AND BOARD LAYOUT HINTS

The LIS3LV02DQ is designed for a voltage supply spanning from 2.16V up to 3.6V.

The typical current consumption in normal mode at 2.5V is 600µA.

Adequate power supply decoupling is required to ensure IC performances. The 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 it is recommended the use of one 100nF ceramic or polyester capacitor which must be placed as close as possible to device Vdd lead. For filtering lower-frequency noise signals, a larger aluminum capacitor of 10µF or greater should be placed near the device in parallel to the former capacitor.

It is recommended that the afore capacitors are placed as close as possible to pin 3.

7 DIGITAL INTERFACES

The registers embedded inside the LIS3LV02DQ may be accessed through I^{2}C and SPI serial interfaces. They are mapped onto the same pads. To select/exploit the I^{2}C interface, CS line must be tied high.

Table 2. I^{2}C/SPI signals mapping

Pin Name	Description
CS	SPI chip select (CS) I^{2}C/SPI selector (1: I^{2}C mode; 0: SPI enabled)
SCL/SPC	SPI CK line (SPC) I^{2}C clock line (SCL)
SDI/SDA/SDO	SPI data in (SDI) I^{2}C serial data (SDA) SPI data out (SDO) -when in 3-wire mode-
SDO	SPI data out (SDO) -when not in 3-wire mode-

7.1 I^{2}C Bus Interface

The LIS3LV02DQ I^{2}C is a bus slave. The I^{2}C is employed to write/read the data into/from the registers.
The relevant I^{2}C terminology is shown 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: SCL and SDA.

These pins are described in the subsequent table:

Table 4. I^{2}C Pin Description

Term	Description
SCL	**S**erial **CL**ock Line
SDA	**S**erial **DA**ta Line

SDA is a bidirectional line. Both SCL and SDA are connected to a positive supply voltage via a pull-up resistor. When the bus is free both lines are HIGH.

7.1.1 I^{2}C Operation

The transaction on the bus is started through a START signal. A START condition is defined as a HIGH to LOW transition on the data line while the SCL line is held HIGH (refer to ST condition in the following paragraph). After this has been transmitted by the Master, the bus is considered busy. The next byte of data transmitted after the start condition contains the address of the slave in the first 7 bits and the eighth bit tells 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.

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 (SAK subsequence). A receiver which has been addressed is obliged to generate an acknowledge after each byte of data has been received.

The I^{2}C embedded inside the LIS3LV02DQ behaves like a slave device and the following protocol must be adhered to. After the start condition (ST) a salve address is sent, once a slave acknowledge has been returned (SAK), a 8-bit sub-address will be transmitted (SUB): the 7 LSB represent the actual register address while the MSB enables address autoincrement.

If the MSB of the SUB field is ‘1’, the SUB (register address) will be automatically incremented to allow multiple data read/write at increasing addresses.

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.

If the LSB of the slave address was ‘1’ (read), a repeated START (SR) condition will have to be issued after the sub-address byte; if the LSB is ‘0’ (write) the Master will transmit to the slave with direction unchanged.

7.1.2 I^{2}C Subsequences

In order to better define subsequences and to clarify line SCL and SDA behavior, a description containing discrete value of SCL and SDA will follow. In column there is the value present on line SCL and SDA in discrete timing. These simple subsequences are used to realize complex commands described in the following paragraph.

Table 5. I^{2}C Subsequences

ST: START condition

SCL	1111
SDA	1100

SR: Repeated START condition

SCL	1111
SDA	1100

SAD: Slave address (binary address: abcdefgh. In LIS3LV02DQ “abcdefg” = “0011101”)

Table 5. I^{2}C Subsequences

SCL	00110	0110	0110	0110	0110	0110	0110	0110
SDA	0aaaa	bbbb	cccc	dddd	eeee	f f f f	gggg	hhhh

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: abcdefgh)

SCL	00110	0110	0110	0110	0110	0110	0110	0110
SDA	0aaaa	bbbb	cccc	dddd	eeee	f f f f	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
SDA	0aaaa	bbbb	cccc	dddd	eeee	f f f f	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

7.1.3 I^{2}C Read and Write Sequences

The previous subsequences are used to realize actual write and read sequences described in the tables below.

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		MAK		NMAK	SP
Slave			SAK		SAK			SAK	DATA		DATA		DATA		

Data are transmitted in byte format. Each data transfer contains 8 bits. The number of bytes transferred per transfer is unlimited. Data is transferred with the Most Significant Bit (MSB) first. If a receiver can't 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 only continues when the receiver is ready for another byte and releases the data line. If a slave receiver doesn't acknowledge the slave address (i.e. it is not able to receive because it is performing some 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 (SP). Each data transfer must be terminated by the generation of a STOP condition.

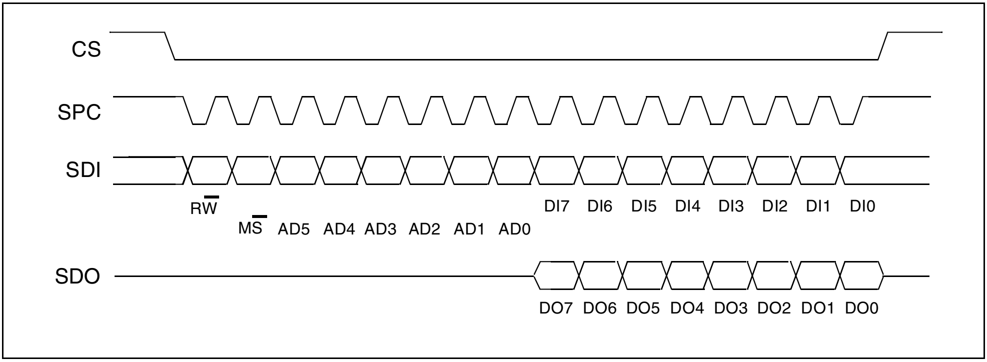
7.2 SPI Bus Interface

The LIS3LV02DQ SPI is a bus slave. The SPI allows to write and read the registers of the device.

The Serial Interface interacts with the outside world with 4 wires: **CS**, **SPC**, **SPDI** and **SPDO**.

7.2.1 Read & Write Protocol

Figure 3. Read & Write Protocol



CS is the Chip Select and it is controlled by the SPI master. It goes low at the start of the transmission and goes back high at the end. **SPC** is the Serial Port Clock and it 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. Those 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 clocks pulses or in multiple of 8 in case 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, ...) starts at the last falling edge of SPC just before the rising edge of **CS**.

bit 0: RW 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 latter case, the chip will drive **SDO** at the start of bit 8.

bit 1: M$\overline{S}$ bit. When 0, the address will remain unchanged in multiple read/write commands. When 1, the address will be 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 will be written into the device (MSb first).

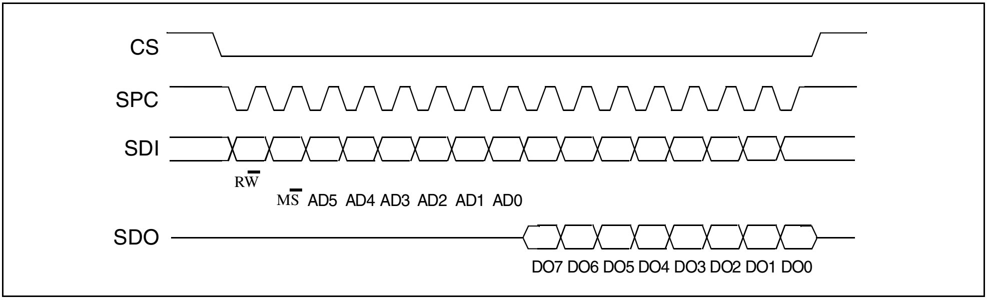
bit 8-15: data DO(7:0) (read mode). This is the data that will be read from the device (MSb first).

In multiple read/write commands further blocks of 8 clock periods will be added. When M$\overline{S}$ bit is 0 the address used to read/write data remains the same for every block. When M$\overline{S}$ 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.

7.2.2 SPI Read

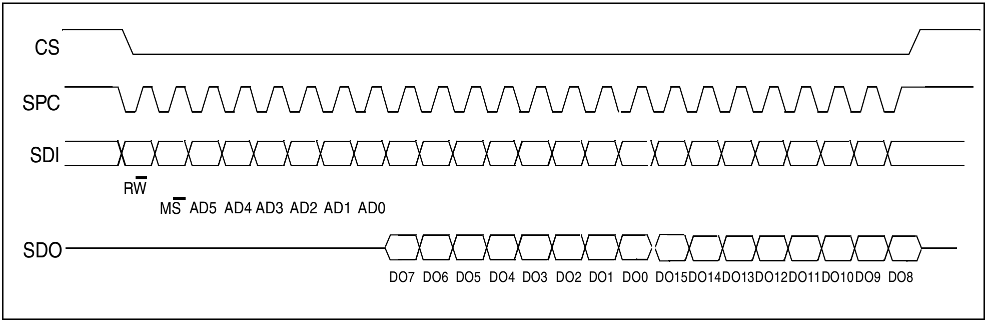
Figure 4. SPI Read Protocol



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

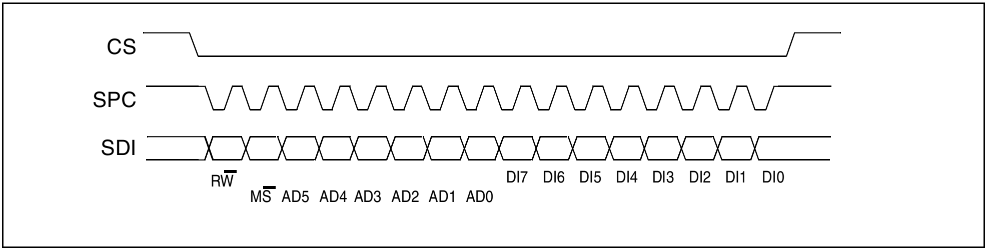
bit 0: READ bit. The value is 1.
bit 1: M$\overline{\text{S}}$ bit. When 0 do not increment address, when 1 increment address in multiple reading.
bit 2-7: address AD(5:0). This is the address field of the indexed register.
bit 8-15: data DO(7:0) (read mode). This is the data that will be read from the device (MSb first).
bit 16-... : data DO(...-8). Further data in multiple byte reading.

Figure 5. Multiple Bytes SPI Read Protocol (2 bytes example)


CS
SPC
SDI
RW
MS AD5 AD4 AD3 AD2 AD1 AD0
SDO
DO7 DO6 DO5 DO4 DO3 DO2 DO1 DO0 DO15 DO14 DO13 DO12 DO11 DO10 DO9 DO8

7.2.3 SPI Write

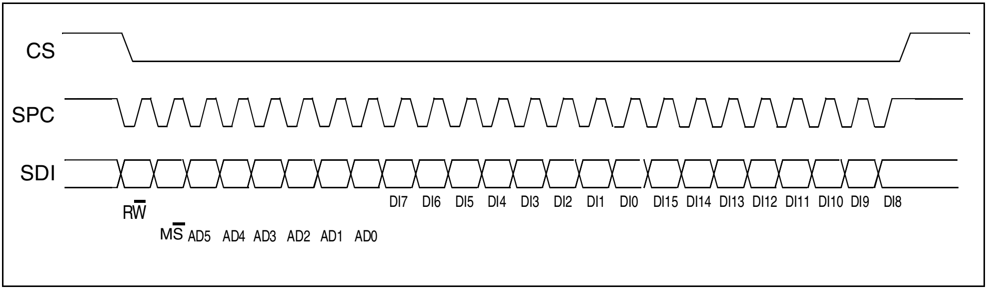
Figure 6. SPI Write Protocol



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

bit 0: WRITE bit. The value is 0.
bit 1: M$\overline{\text{S}}$ bit. When 0 do not increment address, when 1 increment address in multiple writing.
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 will be written inside the device (MSb first).
bit 16-... : data DI(...-8). Further data in multiple byte writing.

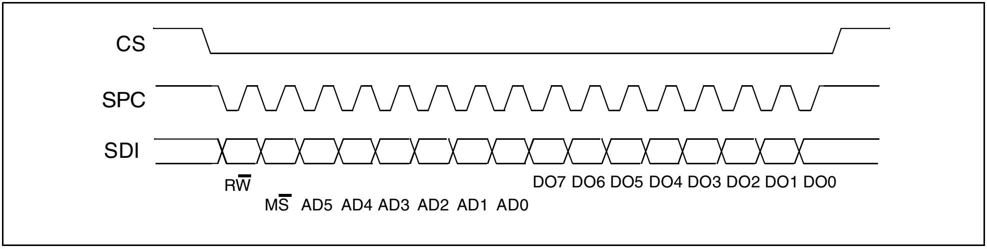
Figure 7. Multiple Bytes SPI Write Protocol (2 bytes example)


CS
SPC
SDI
R$\overline{\text{W}}$
M$\overline{\text{S}}$ AD5 AD4 AD3 AD2 AD1 AD0
DI7 DI6 DI5 DI4 DI3 DI2 DI1 DI0 DI15 DI14 DI13 DI12 DI11 DI10 DI9 DI8

7.2.4 SPI Read in 3-wires Mode

3-wires mode is entered by setting to 1 bit SIM (SPI Serial Interface Mode selection) in CTRL_REG2.

Figure 8. SPI Read Protocol In 3-wires Mode


CS
SPC
SDI
R$\overline{\text{W}}$
M$\overline{\text{S}}$ AD5 AD4 AD3 AD2 AD1 AD0
DO7 DO6 DO5 DO4 DO3 DO2 DO1 DO0

The SPI Read command is performed with 16 clocks pulses:

bit 0: READ bit. The value is 1.
bit 1: M$\overline{\text{S}}$ bit. When 0 do not increment address, when 1 increment address in multiple reading.
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 will be read from the device (MSb first).

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

8 REGISTERS MAPPING

The table given below provides a listing of the registers embedded in the device and the related address

.

Table 6. Registers address map

Reg. Name	Type	Register Address		Default	Comment
		Binary	Hex		
		0000000 - 0001110	00 - 0E		Reserved
WHO_AM_I	1111	0F	3A		
		0010000 - 0010101	15-Oct		Reserved
OFFSET_X	rw	10110	16	Calibration	Loaded at boot
OFFSET_Y	rw	10111	17	Calibration	Loaded at boot
OFFSET_Z	rw	11000	18	Calibration	Loaded at boot
GAIN_X	rw	11001	19	Calibration	Loaded at boot
GAIN_Y	rw	11010	1A	Calibration	Loaded at boot
GAIN_Z	rw	11011	1B	Calibration	Loaded at boot
		0011100 -0011111	1C-1F		Reserved
CTRL_REG1	rw	100000	20	111	
CTRL_REG2	rw	100001	21	0	
CTRL_REG3	rw	100010	22	100	
HP_FILTER RESET	r	100011	23	dummy	Dummy register
		0100100-0100110	24-26		Not Used
STATUS_REG	rw	100111	27	0	
OUTX_L	r	101000	28	output	
OUTX_H	r	101001	29	output	
OUTY_L	r	101010	2A	output	
OUTY_H	r	101011	2B	output	
OUTZ_L	r	101100	2C	output	
OUTZ_H	r	101101	2D	output	
	r	101110	2E		Reserved
		101111	2F		Not Used

Table 6. Registers address map

Reg. Name	Type	Register Address		Default	Comment
		Binary	Hex		
FF_WU_CFG	rw	110000	30	0	
FF_WU_SRC	rw	110001	31	0	
FF_WU_ACK	r	110010	32	dummy	Dummy register
		110011	33		Not Used
FF_WU_THS_L	rw	110100	34	0	
FF_WU_THS_H	rw	110101	35	0	
FF_WU_DURATION	rw	110110	36	0	
		110111	37		Not Used
DD_CFG	rw	111000	38	0	
DD_SRC	rw	111001	39	0	
DD_ACK	r	111010	3A	dummy	Dummy register
		111011	3B		Not Used
DD_THSI_L	rw	111100	3C	0	
DD_THSI_H	rw	111101	3D	0	
DD_THSE_L	rw	111110	3E	0	
DD_THSE_H	rw	111111	3F	0	
		1000000-1111111	40-7F		Reserved

8.1 Reserved Registers

Registers marked as reserved must not be changed. Random changes of the content of those registers might cause permanent damages to the device.

8.2 Registers loaded at Boot

The LIS3LV02DQ is factory trimmed. The content of the registers that are loaded at boot must not be changed. Their content is automatically restored when the device is powered-up.

9 ABOUT CONTROL REGISTERS

9.1 CTRL_REG1 (20h)

PD1	PD0	DF1	DF0	ST	Zen	Yen	Xen

PD1, PD0	Power Down Control. Default value: 00 (00: power down mode; x1, 1x: normal mode)
DF1, DF0	Decimation Factor Control. Default value: 00 (00: decimate by 512 01: decimate by 128 10: decimate by 32 11: decimate by 8)
ST	Self Test Enable. Default value: 0 (0: normal mode; 1: self test enabled)
Zen	Z-axis enable. Default value: 1 (0: channel disabled; 1: channel enabled)
Yen	Y-axis enable. Default value: 1 (0: channel disabled; 1: channel enabled)
Xen	X-axis enable. Default value: 1 (0: channel disabled; 1: channel enabled)

PD1, PD0 bit allows to turn on the device out of power-down mode. The device is in power-down mode when PD1, PD0= “00” (default value after boot). The device is in normal mode when either PD1 or PD0 is set to 1.

DF1, DF0 bit allows to select the data rate at which acceleration samples are produced. The default value is 00 which corresponds to a data-rate of 40Hz. By changing the content of DF1, DF0 to “01”, “10” and “11” the selected data-rate will be set respectively equal to 160Hz, 640Hz and to 2560Hz.

ST bit is used to activate the self test function. When the bit is set to one, an output change will occur to the device outputs thus allowing to check the functionality of the whole measurement chain.

Zen bit enables the Z-axis measurement channel when set to 1. The default value is 1.

Yen bit enables the Y-axis measurement channel when set to 1. The default value is 1.

Xen bit enables the X-axis measurement channel when set to 1. The default value is 1.

9.2 CTRL_REG2 (21h)

FS	BDU	BLE	BOOT	IEN	DRDY	SIM	DAS

FS	Full Scale selection. Default value: 0 (0: +/- 2g; 1: +/- 6g)
BDU	Block Data Update. Default value: 0 (0: continuous update; 1: output registers not updated until MSB and LSB reading)
BLE	Big/Little Endian selection. Default value: 0 (0: little endian; 1: big endian)
BOOT	Reboot memory content. Default value: 0 (0: normal mode; 1: reboot memory content)
IEN	Interrupt Enable. Default value: 0 (0: data-ready on RDY pad; 1: interrupt signal on RDY pad)
DRDY	Enable Data-Ready generation. Default value: 0 (0: data-ready gen. disabled; 1: enable data-ready generation)
SIM	SPI Serial Interface Mode selection. Default value: 0 (0: 4-wire interface; 1: 3-wire interface)
DAS	Data Alignment Selection. Default value: 0 (0: 12 bit right justified; 1: 16 bit left justified)

FS bit is used to select Full Scale value. After the device power-up the default full scale value is +/-2g. In order to obtain a +/-6g full scale it is necessary to set FS bit to ‘1’.

BDU bit is used to inhibit output registers update until both upper and lower register parts are read. In default mode (BDU= ‘0’) the output register values are updated continuously. If for any reason it is not sure to read faster than output data rate it is recommended to set BDU bit to ‘1’. In this way the content of output registers is not updated until both MSB and LSB are read avoiding to read values related to different sample time.

BLE bit is used to select Big Endian or Little Endian representation for output registers. In Big Endian’s one MSB acceleration value is located at addresses 28h (X-axis), 2Ah (Y-axis) and 2Ch (Z-axis) while LSB acceleration value is located at addresses 29h (X-axis), 2Bh (Y-axis) and 2Dh (Z-axis). In Little Endian representation the order is inverted (refer to data register description for more details).

BOOT bit is used to refresh the content of internal registers stored in the flash memory block. At the device power up the content of the flash memory block is transferred to the internal registers related to trimming functions to permit a good behavior of the device itself. If for any reason the content of trimming registers was changed it is sufficient to use this bit to restore correct values. When BOOT bit is set to ‘1’ the content of internal flash is copied inside corresponding internal registers and it is used to calibrate the device. These values are factory

trimmed and they are different for every accelerometer. They permit a good behavior of the device and normally they have not to be changed.
At the end of the boot process the BOOT bit is set again to ‘0’.

IEN bit is used to switch the value present on data-ready pad between Data-ready signal and Interrupt signal. At power up the Data-ready signal is chosen. It is however necessary to modify DRDY bit to enable Data-ready signal generation.

DRDY bit is used to enable DataReady pad activation. If DRDY bit is ‘0’ (default value) on DataReady pad a ‘0’ value is present. If a DataReady signal is desired it is necessary to set to ‘1’ DRDY bit. DataReady signal goes to ‘1’ whenever a new data is available for all the enabled axes. For example if Z-axis is disabled, DataReady signal goes to ‘1’ when new values are available for both X and Y axes. DataReady signal comes back to ‘0’ when all the registers containing values of the enabled axes are read. To be sure not to loose any data coming from the accelerometer data registers must be read before a new DataReady rising edge is generated. In this case DataReady signal will have the same frequency of the data rate chosen.

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 SDO pad. In 3-wire interface mode output data are sent to SDA_SDI pad.

DAS bit selects between 12 bit right justified and 16 bit left justified data representation. The first case is the default case and the most significant bits are replaced by the bit representing the sign.

9.3 CTRL_REG3 (22h)

ECK	HPDD	HPFF	FDS	res	res	CFS1	CFS0

ECK	External Clock. Default value: 0 (0: clock from internal oscillator; 1: clock from external pad)
HPDD	High Pass filter enabled for Direction Detection. Default value: 0 (0: filter bypassed; 1: filter enabled)
HPFF	High Pass filter enabled for Free-Fall. Default value: 0 (0: filter bypassed; 1: filter enabled)
FDS	Filtered Data Selection. Default value: 0 (0: internal filter bypassed; 1: data from internal filter)
CFS1, CFS0	High-pass filter Cut-off Frequency coefficient Selection (00: HPc = 512 01: HPc = 1024 10: HPc = 2048 11: HPc = 4096)
res	Reserved. These bit are reserved for future enhancements. Best to be left to their default value (10 binary)

ECK bit selects whether the clock used in the core comes from internal oscillator or from external pad OSC_IN. In the latter case internal oscillator is switched off and the pin Pull-Down is disabled thus reducing power consumption. External clock must be in the range of 1.045 MHz +/- 10% and must have a duty cycle of 50%.

HPDD bit permits to select either filtered or not filtered data to feed the Signal Processing (Direction Detection) block as described in Figure 10. When HPDD is set to ‘0’ (default mode) the data used to generate DD interrupt come directly from digital block or temperature compensation block while if HPDD is set to ‘1’ the interrupt signals are based on High-Pass filtered data. For additional details about the data flow, please refer to Figure 10.

HPFF bit allows to select between filtered or not filtered data to be processed by the Signal Processing (WakeUp or FreeFall) block as described in Figure 15. If HPFF is set ‘0’ (default mode) data used to generate WU or FF interrupt come directly from digital block or temperature compensation block while if HPFF is set ‘1’ the interrupt signals are based on High-Pass filtered data. For additional details about the data flow, please refer to Figure 10.

FDS bit decides whether data stored in output registers are High Pass filtered or not. In default mode (FDS bit set to ‘0’) signal are not filtered while it is possible to access data coming from HP filter setting FDS bit to ‘1’. For additional details about the data flow, please refer to Figure 10.

CFS<1:0> bit select High-pass filter Cut-off Frequency coefficient. Increasing this number makes Cut-off Frequency (@-3dB) move to lower values.

The Cut-off Frequency behavior is described by the following equation:

$$f_{Cut-off} = \left(\frac{0.318}{HPc}\right)\left(\frac{ODR}{2}\right)$$

Where HPc = 512, 1024, 2048, 4096 when CFS<1:0>= 00, 01, 10, 11 respectively and ODR represents the Output Data Rate selectable through the DF1, DF0 bits in CTRL_REG1.
The figure below gives a representation of the possible transfer functions obtained modifying Output Data Rate and High Pass Filter coefficient (HPc).

Figure 9. HP Filter Transfer Function

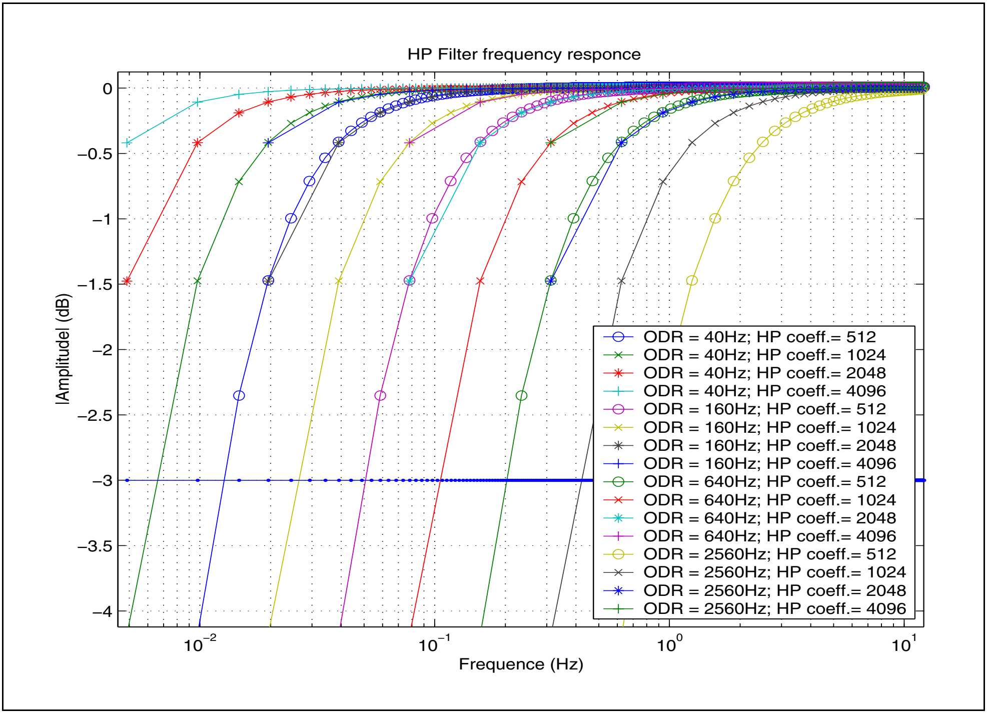
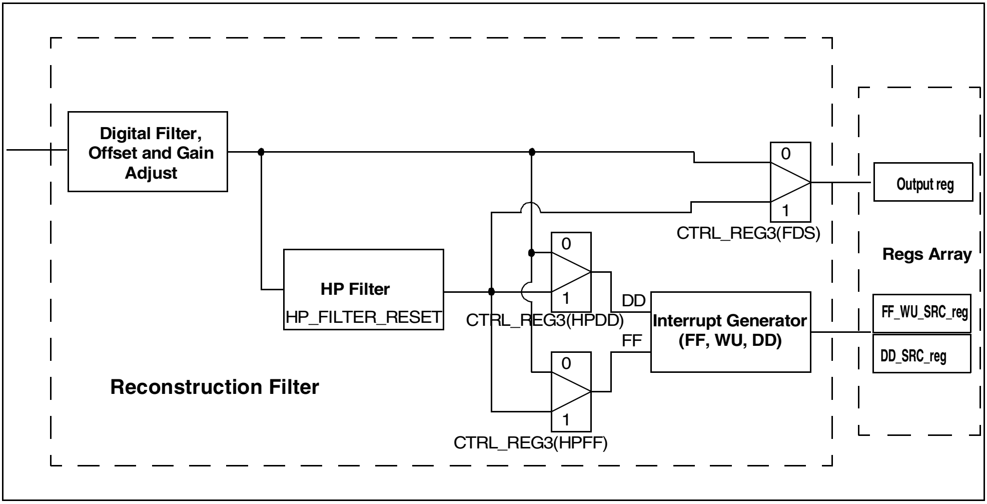


Figure 10. gives a representation of a reconstruction filter with its control signals.

Figure 10. Reconstruction Filter Block Diagram



The data coming from Offset and Gain adjust block go directly to register array for reading when FDS is set to ’0’ (default value)

Instead when the FDS bit in CTRL_REG3 is set to ’1’, the path that involves the use of the embedded HP filter is activated.

HP filtered data can be selected for further processing by the Interrupt Generator block setting the desired values of HPDD and HPFF bit. Default value is ‘0’ and corresponds to the use of non filtered data. The output of Interrupt Generator block is used to load FF_WU_SRC and DD_SRC registers.

10 DATA REGISTERS

10.1 WHO_AM_I (0Fh)

W7	W6	W5	W4	W3	W2	W1	W0

W7, W0	LIS3LV02DQ Physical Address equal to 3Ah

Addressing this register the physical address of the device is returned.
For LIS3LV02DQ the physical address assigned in factory is 3Ah.

10.2 STATUS_REG (27h).

ZYXOR	ZOR	YOR	XOR	ZYXDA	ZDA	YDA	XDA

ZYXOR	X, Y and Z axis Data Overrun. Default value: 0 (0: no overrun has occurred; 1: a new set of data has overwritten the previous ones)
ZOR	Z axis Data Overrun. Default value: 0 (0: no overrun has occurred; 1: a new data for the Z-axis has overwritten the previous one)
YOR	Y axis Data Overrun. Default value: 0 (0: no overrun has occurred; 1: a new data for the Y-axis has overwritten the previous one)
XOR	X axis Data Overrun. Default value: 0 (0: no overrun has occurred; 1: a new data for the X-axis has overwritten the previous one)
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: a new data for the Z-axis is not yet available; 1: a new data for the Z-axis is available)
YDA	Y axis new Data Available. Default value: 0 (0: a new data for the Y-axis is not yet available; 1: a new data for the Y-axis is available)
XDA	X axis new Data Available. Default value: 0 (0: a new data for the X-axis is not yet available; 1: a new data for the X-axis is available)

ZYXOR is set to one whenever a new set of 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_L, OUTX_H, OUTY_L, OUTY_H, OUTZ_L, OUTZ_H) has been overwritten unless BDU bit in CTRL_REG2 is ‘1’. ZYXOR is cleared when the upper parts (OUTX_H, OUTY_H, OUTZ_H) of the active channel registers 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 and the BDU bit in CTRL_REG2 is ‘0’, the previous sample is overwritten. When BDU bit is set to ‘1’ no update is performed until both upper and lower part of OUTZ register are read. ZOR is cleared anytime OUTZ_H 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 and the BDU bit in CTRL_REG2 is ‘0’, the previous sample is overwritten. When BDU bit is set to ‘1’ no update is performed until both upper and lower part of OUTY register are read. YOR is cleared anytime 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 and the BDU bit in CTRL_REG2 is ‘0’, the previous sample is overwritten. When BDU bit is set to ‘1’ no update is performed until both upper and lower part of OUTX register are read. XOR is cleared anytime 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 MSBs 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 OUTZ_H register is read. In order to trigger, the ZDA bit requires the Z axis to be enabled (bit Zen=1 inside CTRL_REG1).

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

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

10.3 OUTX_L (28h)

XD7	XD6	XD5	XD4	XD3	XD2	XD1	XD0

XD7, XD0	X axis acceleration data LSb (bit BLE in CTRL_REG2 reg set to ‘0’)

In Big Endian Mode (bit BLE in CTRL_REG2 set to ‘1’) the content of this register is the MSb acceleration data and depends by bit DAS in CTRL_REG2 reg as described in the following sub-section.

10.4 OUTX_H (29h)

When reading the register in “12 bit right justified” mode (bit DAS in CTRL_REG2 reg set to 0) the most significant bits (7:4) are replaced with XD11 (i.e. XD15-XD12=XD11, XD11, XD11, XD11).

XD15	XD14	XD13	XD12	XD11	XD10	XD9	XD8

XD15, XD8	X axis acceleration data MSb (bit BLE in CTRL_REG2 reg set to ‘0’)

In Big Endian Mode (bit BLE in CTRL_REG2 set to ‘1’) the content of this register is the LSb acceleration data.

10.5 OUTY_L (2Ah)

YD7	YD6	YD5	YD4	YD3	YD2	YD1	YD0

YD11, YD8	Y axis acceleration data LSb (bit BLE in CTRL_REG2 reg set to ‘0’)

In Big Endian Mode (bit BLE in CTRL_REG2 set to ‘1’) the content of this register is the MSb acceleration data and depends by bit DAS in CTRL_REG2 reg as described in the following section.

10.6 OUTY_H (2Bh)

When reading the register in “12 bit right justified” mode (bit DAS in A_IF_CTRL2 reg set to 0) the most significant bits (7:4) are replaced with YD11 (i.e. YD15-YD12=YD11, YD11, YD11, YD11)

YD15	YD14	YD13	YD12	YD11	YD10	YD9	YD8

YD15, YD8	Y axis acceleration data MSb (bit BLE in CTRL_REG2 reg set to ‘0’)

In Big Endian Mode (bit BLE in CTRL_REG2 set to ‘1’) the content of this register is the LSb acceleration data.

10.7 OUTZ_L (2Ch)

ZD7	ZD6	ZD5	ZD4	ZD3	ZD2	ZD1	ZD0

ZD11, ZD8	Z axis acceleration data LSb (bit BLE in CTRL_REG2 reg set to ‘0’)

In Big Endian Mode (bit BLE in CTRL_REG2 set to ‘1’) the content of this register is the MSb acceleration data and depends by bit DAS in CTRL_REG2 reg as described in the following section.

10.8 OUTZ_H (2Dh)

When reading the register in “12 bit right justified” mode (bit DAS in A_IF_CTRL2 reg set to 0) the most significant bits (7:4) are replaced with ZD11 (i.e. ZD15-ZD12=ZD11, ZD11, ZD11, ZD11).

ZD15	ZD14	ZD13	ZD12	ZD11	ZD10	ZD9	ZD8

ZD15, ZD8	Z axis acceleration data MSb (bit BLE in CTRL_REG2 reg set to ‘0’)

In Big Endian Mode (bit BLE in CTRL_REG2 set to ‘1’) the content of this register is the LSb acceleration data.

11 FREE-FALL AND WAKE-UP REGISTERS

The following sections describes the registers that are involved in the generation of the interrupt signal associated to the inertial wake-up and free-fall events.

11.1 HP_FILTER_RESET (23h)

Dummy register. A read at this address set the content of the internal High-Pass filter to the actual X, Y and Z acceleration values.

X	X	X	X	X	X	X	X

11.2 FF_WU_CFG (30h)

Free-fall and wake-up configuration register.

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 into FF_WU_SRC reg with the FF_WU_SRC reg cleared by reading FF_WU_ACK 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)

AOI bit allows to select between Wake-Up (OR combination of interrupt events) and Free-Fall (AND combination of interrupt events) detection

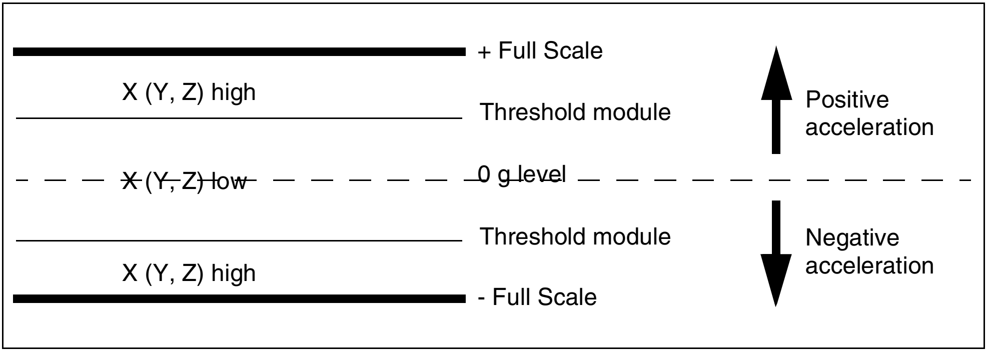
LIR defines whether the configured interrupt event has to be latched by the device once it has happened. In order to clear the request and the related source reg (FF_WU_SRC) it is necessary to perform a reading at the dummy register FF_WU_ACK reg.

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

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

The threshold module which is used by the system to detect any free-fall or inertial wake-up event is defined by FF_WU_THS_H and FF_WU_THS_L registers and it is given by their concatenation FF_WU_THS_H & FF_WU_THS_L. The threshold value is expressed over 16 bit 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 FF_WU_THS_H & FF_WU_THS_L. Similarly, X, (Y, Z) low is true when the unsigned acceleration value of the X (Y, Z) channel is lower than FF_WU_THS_H & FF_WU_THS_L. Refer to Figure 11 for more details.

Figure 11. FF_WU_CFG High and Low



11.3 FF_WU_SRC (31h)

Free-fall and wake-up source register. 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 which is being triggering (or has triggered, in case of LIR bit in FF_WU_SRC reg set to 1) the interrupt signal. In particular IA is equal to 1 when the combination of acceleration events specified in FF_WU_CFG register is true. This bit is used for the generation of the interrupt signal associated to the free-fall/wake-up events (see Figure 13 for any additional detail).

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_H & FF_WU_THS_L.

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_H & FF_WU_THS_L. Refer to Figure 11 for more details.

11.4 FF_WU_ACK (32h)

Dummy register. A read at this address clears the FF_WU_SRC reg and the FF, WU interrupt if the latched option was chosen (LIR bit in FF_WU_CFG reg set to 1).

X	X	X	X	X	X	X	X

11.5 FF_WU_THS_L (34h)

THS7	THS6	THS5	THS4	THS3	THS2	THS1	THS0

THS7, THS0	Free-fall / wake-up Threshold Lsb

11.6 FF_WU_THS_H (35h)

THS15	THS14	THS13	THS12	THS11	THS10	THS9	THS8

THS15, THS8	Free-fall / wake-up Threshold Msb

FF_WU_THS_H and FF_WU_THS_L registers define the threshold which is used by the system to detect any free-fall or inertial wake-up event. The threshold value is expressed over 16 bit as an unsigned number. In particular 7FFFh corresponds to full-scale acceleration (i.e. either 2g or 6g depending on the value of FS bit in CTRL_REG2).

11.7 FF_WU_DURATION (36h)

Set the minimum duration of the free-fall/wake-up event that must be recognized by the LIS3LV02DQ.

FWD7	FWD6	FWD5	FWD4	FWD3	FWD2	FWD1	FWD0

FWD7, FWD0	Free-fall / wake-up minimum duration threshold (default value 00h)

The number contained in the register represents the number of samples produced at the sampling rate set by DF1, DF0 bits in CTRL_REG1.

Events having a duration shorter than the value specified in the FF_WU_DURATION register are filtered out by the device. A value of 00h in this register means that the interrupt is generated as soon as the free-fall/wake-up event configured in FF_WU_CFG reg occurs without waiting for any confirmation. See Figure 13 for additional details.

12 DIRECTION-DETECTION REGISTERS

The LIS3LV02DQ allows the implementation of motion-controlled functions such as gaming and terminal control while requiring reduced computational power to the application controller. The following sections describes the functionality of the registers that are involved in the recognition of the direction in which the terminal has been tilted.

12.1 DD_CFG (38h)

Direction Detector configuration register.

IEND	LIR	ZHIE	ZLIE	YHIE	YLIE	XHIE	XLIE

IEND	Interrupt enable on Direction change. Default value: 0 (0: disabled 1: interrupt signal enabled)
LIR	Latch Interrupt request into DD_SRC reg with the DD_SRC reg cleared by reading DD_ACK 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)

IEND enables the generation of an interrupt signal when a change in DD_SRC reg occurs, i.e. when the tilt of terminal has changed with respect to the previous one.

LIR defines whether the current direction/tilt status has to be latched inside the DD_SRC register. In order to clear the content of the DD_SRC register and the related interrupt request (if enabled through the IEND bit) it is necessary to perform a reading at the dummy register DD_ACK. For any additional detail refer to Figure 13.

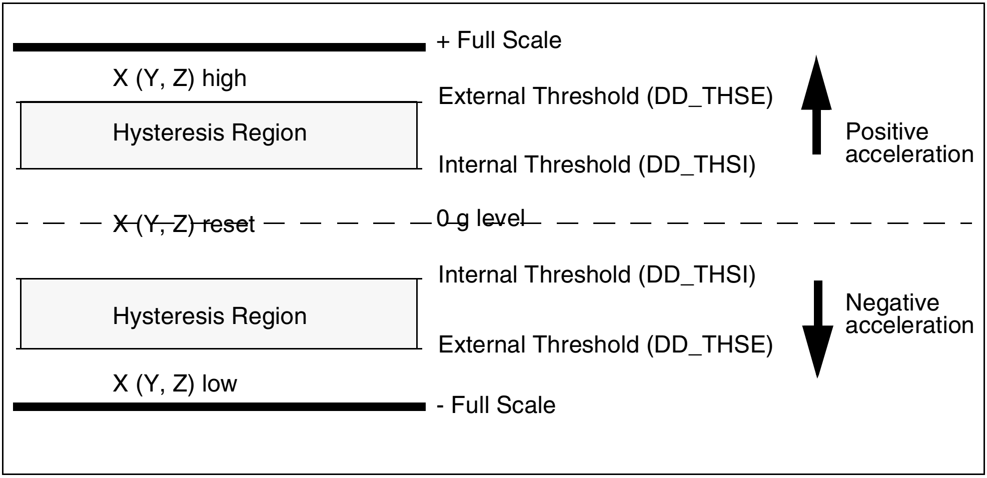
XHIE (YHIE, ZHIE) configure the device to recognize whether the acceleration value along the X axis has exceeded the preset threshold module in the positive direction of the related acceleration axis.

XLIE (YLIE, ZLIE) configure the device to recognize whether the acceleration value along the X axis has exceeded the preset threshold module in the negative direction of the related acceleration axis.

In order to avoid false detections and/or bouncing produced for example by either spurious vibration or tremor, one inner/internal and one outer/external thresholds is defined.
Both the thresholds are expressed over 16 bit as an unsigned number and are given respectively by the concatenation of DD_THSI_H & DD_THSI_L and DD_THSE_H & DD_THSE_L.
Whenever the inner thresholds is greater than the outer one, the hysteresis region will be null and the threshold used to detect the tilt direction is DD_THSE_H & DD_THSE_L.

X, (Y, Z) acceleration data is considered high when the acceleration value of the X (Y, Z) channel is higher than the outer DD_THSE_H & DD_THSE_L threshold (i.e. acceleration exceeding the threshold in module with the terminal tilted in the positive acceleration direction).
Similarly, X, (Y, Z) acceleration data is considered low when the acceleration value of the X (Y, Z) channel is lower than the outer DD_THSE_H & DD_THSE_L threshold (i.e. acceleration exceeding the threshold in module but terminal tilted in the negative acceleration direction).
Both X (Y, Z) high and X (Y, Z) low are reset when the absolute acceleration data on X (Y, Z) channel is lower than the inner DD_THSI_H & DD_THSI_L threshold. Refer to Figure 12 for more details

Figure 12. DD_CFG High and Low Description



12.2 DD_SRC (39h)

Direction Detector source 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 direction in which the device is being tilted. In particular IA is set to one when the tilt of terminal has changed with respect to the previous one. The axes directions that are enabled to set the IA bit to 1 are specified by DD_CFG register. This bit is used for the generation of the interrupt signal associated to the direction-detector (see Figure 13 for any additional detail).

X, (Y, Z) acceleration direction is considered high when the acceleration value of the X (Y, Z) channel is higher than the outer DD_THSE_H & DD_THSE_L threshold (i.e. acceleration exceeding the threshold in module with the terminal tilted in the positive acceleration direction).
Similarly, X, (Y, Z) acceleration data is considered low when the acceleration value of the X (Y, Z) channel is lower than the outer DD_THSE_H & DD_THSE_L threshold (i.e. acceleration exceeding the threshold in module but terminal tilted in the negative acceleration direction).
Both X (Y, Z) high and X (Y, Z) low are reset when the absolute acceleration data on X (Y, Z) channel is lower than the inner DD_THSI_H & DD_THSI_L threshold. Refer to Figure 12 for more details.

When the LIR bit of DD_CFG register is set to 1, the device stores the current direction/tilt status inside the DD_SRC register any time it has changed from the previous state.
In order to refresh the content of the DD_SRC register and to any interrupt request it is necessary to perform a reading at the dummy register DD_ACK.

12.3 DD_ACK (3Ah)

Dummy register. A read at this address allows the refresh of the DD_SRC reg and clears any DD interrupt signal.

x	x	x	x	x	x	x	x

12.4 DD_THSI_L (3Ch)

THSI7	THSI6	THSI5	THSI4	THSI3	THSI2	THSI1	THSI0

THSI7, THSI0	Direction detection Internal Threshold Lsb

12.5 DD_THSI_H (3Dh)

THSI15	THSI14	THSI13	THSI12	THSI11	THSI10	THSI9	THSI8

THSI15, THSI8	Direction detection Internal Threshold Msb

The concatenation of DD_THSI_H and DD_THSI_L registers defines the inner/internal threshold which is used by the LIS3LV02DQ device for direction-detection (see Figure 12). The threshold value is expressed over 16 bit as an unsigned number. In particular 7FFFh corresponds to full-scale acceleration (i.e. either 2g or 6g depending on the value of FS bit in CTRL_REG2). In case no hysteresis is desired, the inner and outer threshold must be equal. The inner threshold should be lower than or equal to the outer one. Whenever the inner thresholds is greater than the outer one, the hysteresis region will be null and the threshold used to detect the tilt direction is DD_THSE_H & DD_THSE_L.

12.6 DD_THSE_L (3Eh)

THSE7	THSE6	THSE5	THSE4	THSE3	THSE2	THSE1	THSE0

THSE7, THSE0	Direction detection External Threshold Lsb

12.7 DD_THSE_H (3Fh)

THSE15	THSE14	THSE13	THSE12	THSE11	THSE10	THSE9	THSE8

THSE15, THSE8	Direction detection External Threshold Msb

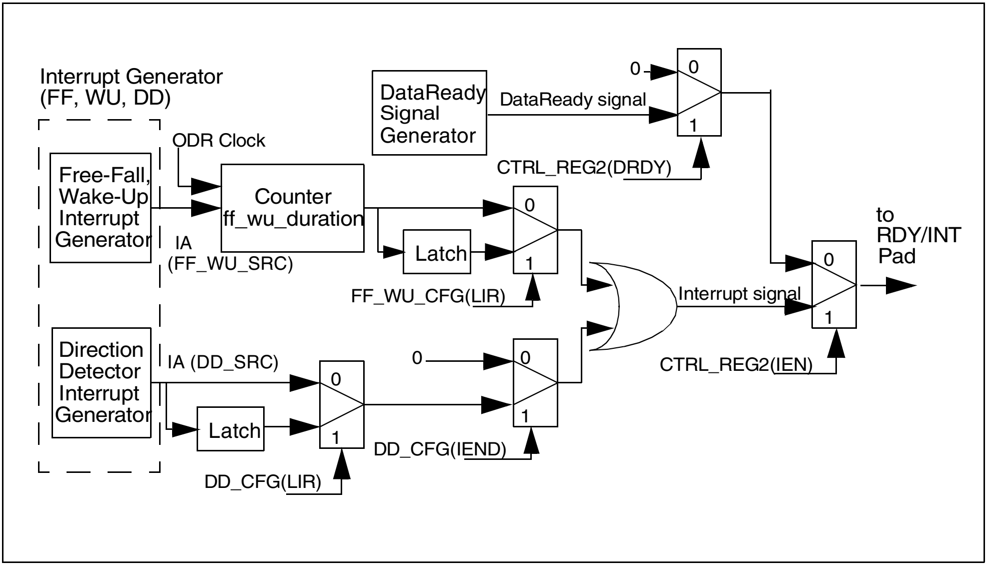
The concatenation of DD_THSE_H and DD_THSE_L registers defines the outer/external threshold which is used by the LIS3LV02DQ device for direction-detection (see Figure 12). The threshold value is expressed over 16 bit as an unsigned number. In particular 7FFFh corresponds to full-scale acceleration (i.e. either 2g or 6g depending on the value of FS bit in CTRL_REG2). In case no hysteresis is desired, the inner and outer threshold must be equal. The inner threshold should be lower than or equal to the outer one. Whenever the inner thresholds is greater than the outer one, the hysteresis region will be null and the threshold used to detect the tilt direction is DD_THSE_H & DD_THSE_L.

13 DATA READY VS. INTERRUPT SIGNAL

The device is provided with a pin which can be activated to generate either the data-ready or the interrupt signal. The functionality of the pin is selected acting on bit IEN and DRDY of CTRL_REG2 accordingly to the block diagram given in Figure 13.

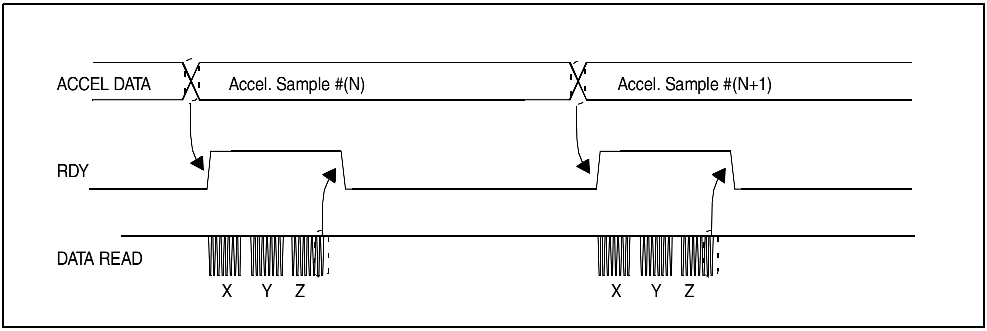
In particular the DataReady signal, stored in STATUS_REG(3), which indicates when a new set of acceleration data is ready, is made available by setting DRDY bit to 1 and IEN bit to 0, while the interrupt signal is carried out when IEN bit is set to 1. Being the pad shared for the both DataReady and Interrupt, one signal at a time may be generated by the device. In case both DRDY and IEN are set to 1, the pad will generate the Interrupt signal which is given an higher priority.

Figure 13. Interrupt and DataReady Signal Generation Block Diagram



In particular the data-ready (RDY) is rised to 1 when a new set of acceleration data has been generated and it is availale for reading. The signal is reset after all the enabled channels are read through the serial interface.

Figure 14. Data Ready Signal



14 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 completed, i.e. approximately after 5 milli-seconds, the device automatically enters power-down mode.

To turn-on the device and gather acceleration data it is necessary to write C7h inside the CTRL_REG1. With this command the three acceleration channels (i.e. X, Y and Z axis) are enabled.

15 READING ACCELERATION DATA

15.1 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_L
5     read OUTX_H
6     read OUTY_L
7     read OUTY_H
8     read OUTZ_L
9     read OUTZ_H
10    data processing
11    goto 1
```

The check performed at step 3 allows to understand 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 reading rate too slow, the bit STATUS_REG(7) will be set to 1.

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

15.2 Using the Data-Ready Signal

The device may be configured to have one HW signal (RDY, pin #6) to determinate when a new set of measurement data is available for reading. This signal is represented by STATUS_REG(3) content. The signal is active high (data available) and it returns to logic zero when the higher part of the data of all the enabled channels have been read. To use the RDY signal, it is necessary to set the CTRL_REG2 to xxxx 01xx.

15.3 Using the Block Data Update feature

In case the reading of the acceleration data is particularly slow and it can not be (or it is not required to have it) synchronized with either the XYZDA bit present inside the STATUS_REG or with the RDY signal, it is strongly recommended to set the BDU (Block Data Update) bit in CTRL_REG2 to 1.

This feature avoids the reading of values (most significant and least significant parts of the acceleration data) related to different samples. In particular, when the BDU is activated, the data registers related to each channel always contain the most recent acceleration data produced by the device but, in case the reading of a given pair (i.e. OUTX_H and OUTX_L, OUTY_H and OUTY_L, OUTZ_H and OUTZ_L) is initiated, the refresh for that pair is blocked until both MSB and LSB parts of the data are read.

16 UNDERSTANDING ACCELERATION DATA

The measured acceleration data are sent into OUTX_H, OUTX_L, OUTY_H, OUTY_L, OUTZ_H and OUTZ_L registers. Those registers contain respectively the most significant part and the least significant part of the acceleration signals acting on the X, Y and Z axes.

The complete acceleration data for the X (Y, Z) channel is given by the concatenation OUTX_H & OUTX_L (OUTY_H & OUTY_L, OUTZ_H & OUTZ_L) and it is expressed as a 2’s complement number.

16.1 Data Alignment: 12 and 16 bit Modes

In 12 bit mode (bit DAS in CTRL_REG2 set to 0 -default configuration-) the acceleration data are right justified and the most significant bits represent a replica of the sign bit while in 16 bit mode (bit DAS in CTRL_REG2 set to 1) the data produced by the device are left justified. In the latter mode, a few of the least significant bit stored inside OUTX_L, OUTY_L and OUTZ_L registers might assume random values accordingly to the SNR performances of the device.

16.2 Big-Little Endian Selection

The LIS3LV02DQ allows to swap the content of the lower and the upper part of the acceleration registers (i.e. OUTX_H with OUTX_L) so to be compliant with both little-endian and big-endian data representations.

"Little Endian" means that the low-order byte of the number is stored in memory at the lowest address, and the high-order byte at the highest address. (The little end comes first). This mode corresponds to bit BLE in CTRL_REG2 reset to 0 -default configuration-.

On the contrary "Big Endian" means that the high-order byte of the number is stored in memory at the lowest address, and the low-order byte at the highest address.

In order to activate big-endian mode it is necessary to set bit DAS in CTRL_REG2 to 1.

16.3 Example of Acceleration Data

The table below provides few basic examples of the data that will be 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, ...) and show practically the effect of DAS and BLE bit.

Table 7. Output Data Registers Content vs. Acceleration (FS= 2g)

Acceleration Values	DAS=0; BLE=0		DAS=0; BLE=1		DAS=1; BLE=0		DAS=1; BLE=1	
	Register Address							
	28h	29h	28h	29h	28h	29h	28h	29h
0 g	00h	00h	00h	00h	0xh	00h	00h	0xh
350 mg	66h	01h	01h	66h	6xh	16h	16h	6xh
1 g	00h	04h	04h	00h	0xh	40h	40h	0xh
-350 mg	9Ah	FEh	FEh	9Ah	Axh	E9h	E9h	Axh
-1g	00h	FCh	FCh	00h	0xh	C0h	C0h	0xh

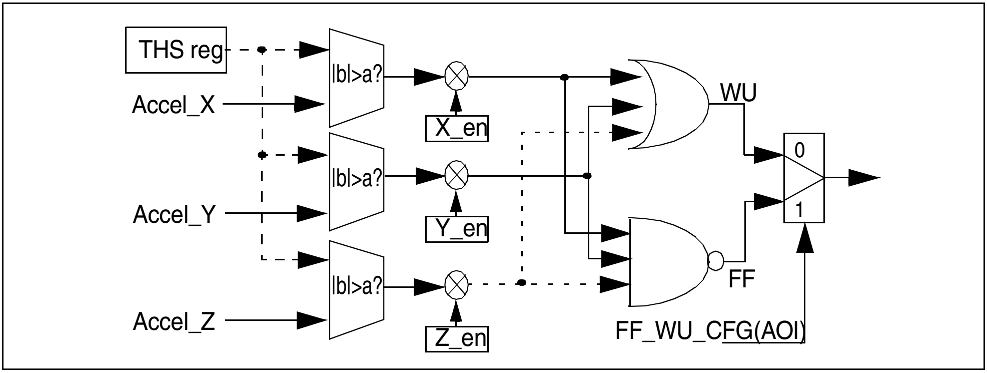
17 INTERRUPT GENERATION DESCRIPTION

The LIS3LV02DQ can provide an interrupt signal and offers several possibilities to personalize this signal. The registers involved in the interrupt generation behavior are CTRL_REG2, FF_WU_CFG, FF_WU_THS, FF_WU_DURATION, DD_CFG, DD_THSI, DD_THSE. In this section a brief description of the capability of the interrupt generation is provided.

The LIS3LV02DQ interrupt signal can behave as Free-Fall, Wake-Up and/or Direction Detector. Whenever an interrupt condition is verified the interrupt signal goes high and reading FF_WU_SRC and DD_SRC registers it is possible to understand which condition happened.

Free-Fall signal (FF) and Wake-Up signal (WU) interrupt generation block is represented by the diagram below:

Figure 15. Free-Fall, Wake-Up Interrupt Generator



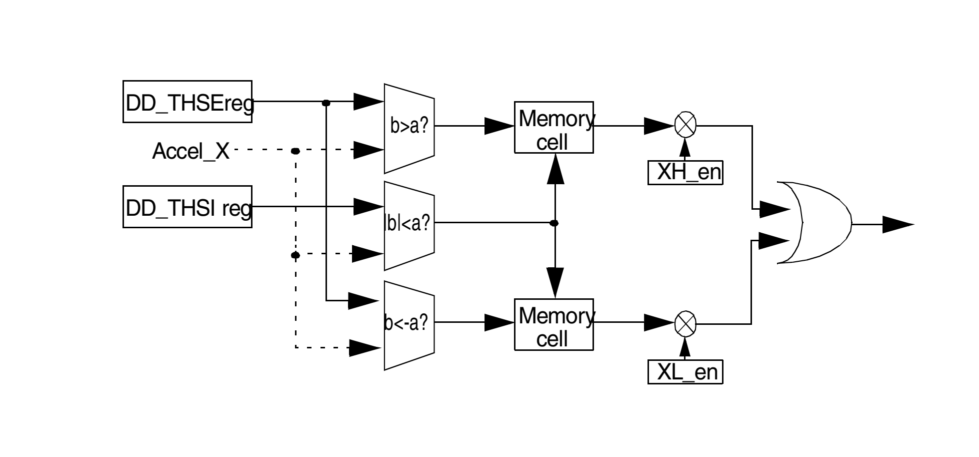
FF or WU interrupt generation is selected through AOI bit in FF_WU_CFG register (address 30 hexadecimal). If AOI bit is ‘0’ signals coming from comparators are put in logical or. Depending on values written in FF_WU_CFG register every time the value of almost one of the enabled axes exceeds the threshold written in module in FF_WU_THS registers a FF, WU interrupt is generated. Otherwise if AOI bit is ‘1’ signals coming from comparators are going into a “NAND” port. In this case an interrupt signal is generated only if all the enabled axes are passing the threshold written in FF_WU_THS registers.

FF_WU_CFG(LIR) bit permits to decide if the interrupt request has to be latched or not. If LIR bit is ‘0’ (default value) interrupt signal goes high when the interrupt condition is satisfied and comes back low immediately if the interrupt condition is no more verified. Otherwise if LIR bit is ‘1’ whenever a interrupt condition is applied the interrupt signal remains high even if the condition comes back to non-interrupt status until a reading to FF_WU_ACK register is performed.

The remaining bits of FF_WU_CFG register permits to decide on which axis the interrupt decision has to be performed and on which direction the threshold has to be passed to generate the interrupt request.

Figure 16 is a representation of Direction Detector Interrupt Generator for one of the axes. The data coming from comparators are compared with the one of the previous acquisition. If a difference is found it means that the threshold written in the DD_THS registers was exceeded and an interrupt signal is generated. If different values are used for DD_THSE and DD_THSI registers it is possible to introduce an hysteresis in the interrupt generation. The difference between WU and DD signals is that the first one gives information if the device reads statically an acceleration in a chosen range while the second one provides an information on the device movement from a position. Reading Direction Detector Source Register is then possible to understand in which direction and in which versus the movement has happened.

Figure 16. Direction Detector Interrupt Generator for X axis



These Interrupt generator blocks can be combined with the embedded High Pass filter in order to obtain further feature.

Furthermore, as described in Figure 13, FF_WU and DD can be used together to generate an interrupt signal that is the logic "OR" of the two interrupts.

18 INERTIAL WAKE-UP

18.1 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 (100mg used in the example). The event which triggers the interrupt is latched inside the device and its occurrence is signalled through the usage of the RDY/INT pin.

```
1    write C7h into CTRL_REG1                // Turn on the sensor and set DF=512
2    write 08h into CTRL_REG2                // Enables interrupt generation
3    write 08h into CTRL_REG3                // Set CTRL_REG3 to its default state
4    write 60h into FF_WU_THS_L reg          // Set the lower part of wake-up threshold
5    write 06h into FF_WU_THS_H reg          // Set the higher part of wake-up threshold
6    write 00h into FF_WU_DURATION reg       // No filtering/confirmation on the event
7    write 4Ah into FF_WU_CFG                // Configure desired wake-up event
```

```
8     poll RDY/INT pad; if RDY/INT=0 then goto 8         // Poll RDY/INT pin waiting for the
                                                         // wake-up event

9     read FF_WU_SRC reg                                 // Return the event that has triggered the
                                                         // interrupt

10    (Wake-up event has occurred; insert your code here) // Event handling

11    read FF_WU_ACK register                            // Clear interrupt request

12    goto 8
```

18.2 Using the HP Filter

The code provided below gives a basic routine which shows the practical use of the inertial wake-up feature performed onto high-pass filtered data. In particular the device is configured to recognize when the high-frequency component of the acceleration applied along either X, Y or Z axis exceeds a preset threshold (100mg used in the example). The event which triggers the interrupt is latched inside the device and its occurrence is signalled through the usage of the RDY/INT pin.

```
1     write C7h into CTRL_REG1                           // Turn on the sensor and set DF=512

2     write 08h into CTRL_REG2                           // Enables interrupt generation

3     write 28h into CTRL_REG3                           // Enable the HP filter and force HPc=512

4     write 60h into FF_WU_THS_L reg                     // Set the lower part of wake-up threshold

5     write 06h into FF_WU_THS_H reg                     // Set the higher part of wake-up threshold

6     write 00h into FF_WU_DURATION reg                  // No filtering/confirmation on the event

7     read HP_FILTER_RESET register                      // Dummy read to force the HP filter to
                                                         // actual acceleration value
                                                         // (i.e. set reference acceleration/tilt value)

8     write 6Ah into FF_WU_CFG                           // Configure desired wake-up event

9     poll RDY/INT pad; if RDY/INT=0 then goto 9         // Poll RDY/INT pin waiting for the
                                                         // wake-up event

10    (Wake-up event has occurred; insert your code here) // Event handling

11    read FF_WU_SRC reg                                 // Return the event that has triggered the
                                                         // interrupt

12    (Insert your code here)                            // Event handling

13    read FF_WU_ACK register                            // Clear interrupt request

14    goto 9
```

At step 7, a dummy read at 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 it is required to set the orientation/tilt of the device as a reference state without waiting for the filter to settle.

19 FREE FALL DETECTION

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

```
1    write D7h into CTRL_REG1                          // Turn on the sensor and set DF=128

2    write 08h into CTRL_REG2                          // Enables interrupt generation

3    write 08h into CTRL_REG3                          // Set CTRL_REG3 to its default state

4    write 60h into FF_WU_THS_L reg                    // Set the lower part of free-fall threshold

5    write 16h into FF_WU_THS_H reg                    // Set the higher part of free-fall threshold

6    write 05h into FF_WU_DURATION reg                 // Set minimum event duration

7    write D5h into FF_WU_CFG                          // Configure free-fall recognition and latch
                                                       // interrupt request

8    poll RDY/INT pad; if RDY/INT=0 then goto 8        // Poll RDY/INT pin waiting for the free-fall
                                                       // event

9    (Free-fall event has occurred; insert your code here)   // Event handling

10   read FF_WU_ACK register                           // Clear interrupt request

11   goto 8
```

The code sample exploits a threshold set at 350mg for free-fall recognition and the event is notified by the hardware signal RDY/INT. At step 6, the FF_WU_DURATION register is configured so to ignore events that are shorter than 5/DR=5/160~30msec (DR=data rate; see Table 8.) in order to avoid false detections.

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

20 DIRECTION DETECTION

The code disclosed below illustrates the ability of the LIS3LV02DQ device to recognize in which direction it has been tilted and to flag a change in its orientation/tilt with respect to the previous state through the hardware RDY/INT signal. In the example below the device is configured to detect tilt which exceed 260mg (set as external/outer threshold), while the internal/ inner threshold is set, as an example, to 150mg.

```
1    write C7h into CTRL_REG1                               // Turn on the sensor and set DF=512

2    write 08h into CTRL_REG2                               // Enables interrupt generation

3    write 4Bh into CTRL_REG3                               // Enable the HP filter and force HPc=4096

4    write A3h into DD_THSE_L reg                           // Set the lower part of outer threshold

5    write 10h into DD_THSE_H reg                           // Set the higher part of outer threshold

6    write 99h into DD_THSI_L reg                           // Set the lower part of inner threshold

7    write 09h into DD_THSI_H reg                           // Set the higher part of inner threshold

8    read HP_FILTER_RESET register                          // Dummy read to force the HP filter to
                                                            // actual acceleration value
                                                            // (i.e. set reference acceleration/tilt value)

9    write CFh into DD_CFG reg                              // Configure direction detection along
                                                            // X and Y axes

10   poll RDY/INT pad; if RDY/INT=0 then goto 10            // Poll RDY/INT pin waiting for a direction
                                                            // change event

11   (Direction/tilt has changed; insert your code here)    // Event handling

12   read DD_SRC reg                                        // Return the direction in which the tilt
                                                            // has occurred

13   (Insert your code here)                                // Event handling

14   read DD_ACK register                                   // Clear interrupt request (on direction
                                                            // change)

15   goto 10
```

At step 8, a dummy read at HP_FILTER_RESET register is performed to set the current acceleration/tilt as a reference state value for threshold comparison. This read may be performed any time it is required to set the orientation/tilt of the device as a reference state without waiting for the filter to settle.

21 OUTPUT DATA RATE SELECTION AND READING TIMING

The output data rate is user selectable through Decimation Factor Control bit (DF1, DF0) stored in CTRL_REG1 (20h) register. At power-on-reset DF1, DF0 are reset to 0 thus providing a default decimation factor set to 512.

The selectable decimation factor are given in table below together with the output data rate.

Table 8. Output Data Rate

DF1, DF0	Decimation Factor	Output data rate	Digital Filter cut-off freq. (-3dB)
00	512	40 Hz	10 Hz
01	128	160 Hz	42 Hz
10	32	640 Hz	168 Hz
11	8	2560 Hz	675 Hz

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

A minimum reading period 150 µs shorter than the output data rate period is defined not to loose any data produced. During this time period the reading of the data must be performed and DataReady signal can be used as a trigger to begin the reading sequence. At the end of the complete sequence DataReady signal goes down and the following rise edge advise that new data are available. If this minimum reading frequency is not observed it is possible to loose some data and the DataReady signal have no more the meaning of a trigger signal. The status register can be used to infer if an overrun happened.

Figure 17. Reading Timing

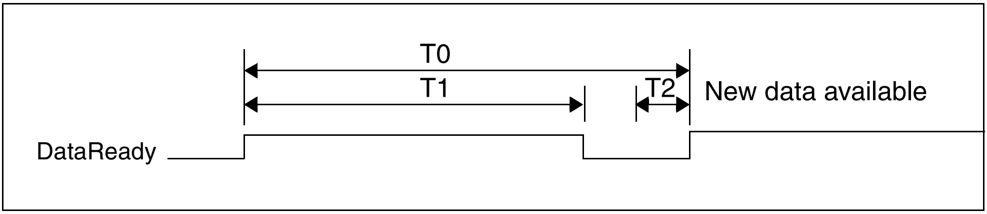


Table 9. Timing Value to Avoid Loosing the Data

Time	Description	Min	Typ	Max
T0	Data rate	350µs		25 ms
T1	Reading period			T0-T2
T2	New data generation	150µs		

22 USING THE EXTERNAL CLOCK

The device provides the capability to drive the internal circuitry through an external clock.
In order to switch to the external clock it is necessary to set to 1 the ECK bit stored inside CTRL_REG3.
The external clock must have a frequency of 1.045 MHZ (+/-10%) and a duty cycle of 50%.

23 REVISION HISTORY

Table 10. Revision History

Date	Revision	Description of Changes
July 2005	1	First Issue.
12-Oct-2005	2	Updated registers description.

The present note which is for guidance only, aims at providing customers with information regarding their products in order for them to save time. As a result, STMicroelectronics shall not be held liable for any direct, indirect or consequential damages with respect to any claims arising from the content of such a note and/or the use made by customers of the information contained herein in connection with their products.

Information furnished is believed to be accurate and reliable. However, STMicroelectronics assumes no responsibility for the consequences of use of such information nor for any infringement of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of STMicroelectronics. Specifications mentioned in this publication are subject to change without notice. This publication supersedes and replaces all information previously supplied. STMicroelectronics products are not authorized for use as critical components in life support devices or systems without express written approval of STMicroelectronics.

STMicroelectronics group of companies
Australia - Belgium - Brazil - Canada - China - Czech Republic - Finland - France - Germany - Hong Kong - India - Israel - Italy - Japan - Malaysia - Malta - Morocco - Singapore - Spain - Sweden - Switzerland - United Kingdom - United States of America
www.st.com