# AN3393
# Application note

## LIS3DSH: 3-axis digital output accelerometer

## Introduction

This document is intended to provide information on the use of and application hints related to ST's LIS3DSH 3-axial digital accelerometer.

The LIS3DSH is an ultra low-power high performance 3-axis linear accelerometer belonging to the "nano" family.

It has dynamically user selectable full scales of ±2*g*/±4*g*/±6*g*/±8*g*/±16*g* and is capable of measuring accelerations with output data rates from 3.125 Hz to 1.6 kHz.

The self-test capability allows the user to check the functioning of the sensor in the final application.

The LIS3DSH has an integrated first in, first out (FIFO) buffer allowing the user to store data for host processor intervention reduction.

The device can be configured to generate interrupt signals activated by user defined motion patterns. To do this, two embedded Finite State Machines can be programmed independently for motion detection. Each State Machine has 16 states.

The LIS3DSH is available in small thin plastic land grid array package (LGA), and it is guaranteed to operate over an extended temperature range from -40 °C to +85 °C.

# Contents


**1 Operating modes . . . . . 10**

1.1 Power-down mode . . . . . 11

1.2 Normal mode . . . . . 11

1.3 Switch mode timing . . . . . 11

**2 Startup sequence . . . . . 12**

2.1 Reading acceleration data . . . . . 12

2.1.1 Using the status register . . . . . 12

2.1.2 Using the data-ready (DRY) signal . . . . . 12

2.1.3 Using the block data update (BDU) feature . . . . . 13

2.2 Understanding acceleration data . . . . . 13

2.2.1 Data alignment . . . . . 14

2.2.2 Example of acceleration data . . . . . 14

**3 Interrupt generation . . . . . 15**

**4 Register description . . . . . 17**

4.1 Register table . . . . . 17

4.2 OUT_T (0x0C) . . . . . 19

4.3 INFO1 (0x0D) . . . . . 19

4.4 INFO2 (0x0E) . . . . . 20

4.5 WHO_AM_I (0x0F) . . . . . 20

4.6 OFF_X (0x10), OFF_Y (0x11), OFF_Z (0x12) . . . . . 20

4.7 CS_X (0x13), CS_Y (0x14), CS_Z (0x15) . . . . . 21

4.8 LC_L (0x16), LC_H (0x17) . . . . . 21

4.9 STAT (0x18) . . . . . 22

4.10 PEAK1 (0x19), PEAK2 (0x1A) . . . . . 23

4.11 Vector filter coefficients (0X1B-0X1E) . . . . . 23

4.12 THRS3 (0X1F) . . . . . 23

4.13 CTRL_REG4 (0X20) . . . . . 24

4.14 CTRL_REG1 (0X21) . . . . . 25

4.15 CTRL_REG2 (0X22) . . . . . 25

4.16 CTRL_REG3 (0X23) . . . . . 26
4.17 CTRL_REG5 (0X24) . . . . . 27
4.18 CTRL_REG6 (0X25) . . . . . 27
4.19 STATUS (0X27) . . . . . 28
4.20 OUT_X_L (0X28), OUT_X_H (0x29) . . . . . 29
4.21 OUT_Y_L (0X2A), OUT_Y_H (0x2B) . . . . . 29
4.22 OUT_Z_L (0X2C), OUT_Z_H (0x2D) . . . . . 29
4.23 FIFO_CTRL (0X2E) . . . . . 29
4.24 FIFO_SRC (0X2F) . . . . . 30
4.25 ST1_X (40h-4Fh) . . . . . 31
4.26 TIM4_1 (50h) . . . . . 31
4.27 TIM3_1 (51h) . . . . . 31
4.28 TIM2_1(52h - 53h) . . . . . 31
4.29 TIM1_1(54h - 55h) . . . . . 31
4.30 THRS2_1(56h) . . . . . 32
4.31 THRS1_1(57h) . . . . . 32
4.32 MASK1_B(59h) . . . . . 32
4.33 MASK1_A(5Ah) . . . . . 33
4.34 SETT1 (5Bh) . . . . . 33
4.35 PR1 (5Ch) . . . . . 34
4.36 TC1 (5Dh-5Eh) . . . . . 34
4.37 OUTS1 (5Fh) . . . . . 34
4.38 ST2_X (60h-6Fh) . . . . . 35
4.39 TIM4_2 (70h) . . . . . 35
4.40 TIM3_2 (71h) . . . . . 35
4.41 TIM2_2 (72h - 73h) . . . . . 36
4.42 TIM1_2 (74h - 75h) . . . . . 36
4.43 THRS2_2 (76h) . . . . . 36
4.44 THRS1_2 (77h) . . . . . 36
4.45 DES2 (78h) . . . . . 37
4.46 MASK2_B (79h) . . . . . 37
4.47 MASK2_A (7Ah) . . . . . 37
4.48 SETT2 (7Bh) . . . . . 38

4.49 PR2 (7Ch) . . . 38
4.50 TC2 (7Dh-7E) . . . 39
4.51 OUTS2 (7Fh) . . . 39

**5 State Machine . . . 41**
5.1 State Machine definition . . . 41
5.2 State Machine in LIS3DSH . . . 41
5.3 Signal block . . . 42
5.3.1 LSB cutter . . . 42
5.3.2 Vector calculation . . . 42
5.3.3 Vector filter . . . 43
5.4 State Machine blocks . . . 44
5.4.1 Decimator . . . 44
5.4.2 DIFF calculation . . . 45
5.5 State Machine description . . . 45

**6 Operation codes . . . 48**
6.1 Next/Reset conditions . . . 48
6.1.1 NOP (0h) . . . 49
6.1.2 TI1 (1h) . . . 49
6.1.3 TI2 (2h) . . . 49
6.1.4 TI3 (3h) . . . 49
6.1.5 TI4 (4h) . . . 50
6.1.6 GNTH1 (5h) . . . 50
6.1.7 GNTH2 (6h) . . . 50
6.1.8 LNTH1 (7h) . . . 51
6.1.9 LNTH2 (8h) . . . 51
6.1.10 GTTH1 (9h) . . . 52
6.1.11 LLTH2 (Ah) . . . 52
6.1.12 GRTH1 (Bh) . . . 52
6.1.13 LRTH1 (Ch) . . . 53
6.1.14 GRTH2 (Dh) . . . 53
6.1.15 LRTH2 (Eh) . . . 54
6.1.16 NZERO (Fh) . . . 54
6.2 Commands . . . 54
6.2.1 STOP (00h) . . . 56

6.2.2 CONT (11h) . . . 57
6.2.3 JMP (22h) . . . 57
6.2.4 SRP (33h) . . . 58
6.2.5 CRP (44h) . . . 58
6.2.6 SETP (55h) . . . 59
6.2.7 SETS1 (66h) . . . 59
6.2.8 STHR1 (77h) . . . 59
6.2.9 OUTC (88h) . . . 60
6.2.10 OUTW (99h) . . . 60
6.2.11 STHR2 (AAh) . . . 61
6.2.12 DEC (BBh) . . . 61
6.2.13 SISW (CCh) . . . 61
6.2.14 REL (DDh) . . . 62
6.2.15 STHR3 (EEh) . . . 62
6.2.16 SSYNC (FFh) . . . 62
6.2.17 SABS0 (12h) . . . 64
6.2.18 SABS1 (13h) . . . 65
6.2.19 SELMA (14h) . . . 65
6.2.20 SRADI0 (21h) . . . 65
6.2.21 SRADI1 (23h) . . . 66
6.2.22 SELSA (24h) . . . 66
6.2.23 SCS0 (31h) . . . 66
6.2.24 SCS1 (32h) . . . 66
6.2.25 SRTAM0 (34h) . . . 67
6.2.26 STIM3 (41h) . . . 67
6.2.27 STIM4 (42h) . . . 67
6.2.28 SRTAM1 (43h) . . . 68

**7 Axis mask filter . . . 69**

7.1 Mask registers . . . 69
7.2 Sign filter . . . 70
7.3 Temporary output mask . . . 71
7.4 Output register (OUTSy) . . . 72

**8 Peak detection . . . 73**

**9 Examples of State Machine configurations . . . 75**

9.1 Toggle . . . 75
9.2 Wake-up . . . 76
9.3 Freefall . . . 77
9.4 Double-turn . . . 79
9.5 Double-tap . . . 81
9.6 6D position recognition . . . 83

**10 First in first out (FIFO) buffer . . . 86**
10.1 FIFO description . . . 86
10.2 FIFO registers . . . 87
10.2.1 Control register 6 (0x25) . . . 87
10.2.2 FIFO control register (0x2E) . . . 88
10.2.3 FIFO source register (0x2F) . . . 89
10.3 FIFO modes . . . 90
10.3.1 Bypass mode . . . 90
10.3.2 FIFO mode . . . 91
10.3.3 Stream mode . . . 92
10.3.4 Stream-to-FIFO mode . . . 94
10.3.5 Bypass-to-Stream mode . . . 95
10.3.6 Bypass-to-FIFO mode . . . 96
10.4 Watermark . . . 96
10.5 Retrieve data from FIFO . . . 97

**11 Revision history . . . 99**

# List of tables

Table 1. Data rate configuration . . . 10
Table 2. Power consumption . . . 10
Table 3. Turn-on times . . . 11
Table 4. Output data registers content vs. acceleration (FS = 2 g) . . . 14
Table 5. Interrupt bits . . . 15
Table 6. Register table . . . 17
Table 7. OUT_T . . . 19
Table 8. INFO1 . . . 19
Table 9. INFO2 . . . 20
Table 10. WHO_AM_I . . . 20
Table 11. Offset axis . . . 20
Table 12. Constant shift for single axis . . . 21
Table 13. Status of long counter LSB (0x16)r . . . 21
Table 14. Status of long counter MSB (0x17) . . . 21
Table 15. Status of long counter values . . . 21
Table 16. STAT register . . . 22
Table 17. STAT register description . . . 22
Table 18. PEAK1, 2 register description . . . 23
Table 19. VFC register description . . . 23
Table 20. THRS3 register description . . . 23
Table 21. Control register 1 description . . . 24
Table 22. Description control register 5 . . . 24
Table 23. Data rate . . . 24
Table 24. Control register 2 description . . . 25
Table 26. Control register 3 description . . . 25
Table 27. Control register 3 bit description . . . 26
Table 28. Control register 4 description . . . 26
Table 30. Control register 5 description . . . 27
Table 31. Control register 5 bit description . . . 27
Table 32. Self-test mode . . . 27
Table 33. Control register 6 description . . . 27
Table 34. Control register 6 bit description . . . 27
Table 35. Status register description . . . 28
Table 36. Status register bit description . . . 28
Table 37. FIFO_CTRL description . . . 29
Table 38. FIFO_CTRL bit description . . . 29
Table 39. FIFO MODE description . . . 30
Table 40. FIFO_CTRL description . . . 30
Table 41. FIFO_SRC bit description . . . 30
Table 42. Timer4 default value . . . 31
Table 43. Timer3 default value . . . 31
Table 44. TIM2_1_L default value . . . 31
Table 45. TIM2_1_H default value . . . 31
Table 46. TIM1_1_L default value . . . 31
Table 47. TIM1_1_H default value . . . 32
Table 48. HRS2_1 default value . . . 32
Table 49. THRS1_1 default value . . . 32
Table 50. MASK1_B axis and sign mask register . . . 32

Table 51. MASK1_B register structure . . . 32
Table 52. MASK1_A axis and sign mask register . . . 33
Table 53. MASK1_A register structure . . . 33
Table 54. SETT1 register structure . . . 33
Table 55. SETT1 register description . . . 33
Table 56. PR1 register . . . 34
Table 57. PR1 register description . . . 34
Table 58. TC1_L default value . . . 34
Table 59. TC1_H default value . . . 34
Table 60. OUTS1 register . . . 35
Table 61. OUTS1 register description . . . 35
Table 62. Timer4 default value . . . 35
Table 63. Timer3 default value . . . 35
Table 64. TIM2_1_L default value . . . 36
Table 65. TIM2_1_H default value . . . 36
Table 66. TIM1_2_L default value . . . 36
Table 67. TIM1_2_H default value . . . 36
Table 68. THRS2_2 default value . . . 36
Table 69. THRS1_2 default value . . . 36
Table 70. DES2 default value . . . 37
Table 71. MASK2_B axis and sign mask register . . . 37
Table 72. MASK2_B register description . . . 37
Table 73. MASK2_A axis and sign mask register . . . 37
Table 74. MASK2_A register description . . . 37
Table 75. SETT2 register . . . 38
Table 76. SETT2 register description . . . 38
Table 77. PR2 register . . . 39
Table 78. PR2 register description . . . 39
Table 79. TC2_L default value . . . 39
Table 80. TC2_H default value . . . 39
Table 81. OUTS2 register . . . 39
Table 82. OUTS2 register description . . . 39
Table 83. Conditions . . . 48
Table 84. Commands (main set) . . . 55
Table 85. Commands (extended set) . . . 56
Table 86. Forbidden OP codes . . . 56
Table 87. MASKy register . . . 70
Table 88. Register configuration for toggle application . . . 75
Table 89. Register configuration for wake-up application . . . 76
Table 90. Register configuration for freefall application . . . 77
Table 91. Register configuration for double-turn application . . . 79
Table 92. Register configuration for double-tap application . . . 81
Table 93. OUTS1 (5Fh) register content in 6D position recognition . . . 84
Table 94. Register configuration for 6D position recognition . . . 84
Table 95. FIFO buffer full representation (32nd sample set stored) . . . 86
Table 96. FIFO overrun representation (33rd sample set stored and 1st sample discarded) . . . 87
Table 97. FIFO enable bit in CTRL_REG6 . . . 88
Table 98. FIFO_CTRL . . . 88
Table 99. FIFO buffer behavior selection . . . 88
Table 100. FIFO_SRC_REG . . . 89
Table 101. FIFO_SRC_REG behavior assuming FTH[4:0] = 15 . . . 89
Table 102. CNTRL6 (0x25) . . . 90
Table 103. Document revision history . . . 99

# List of figures

Figure 1. Data-ready signal (IA = 1) . . . 13
Figure 2. Interrupt signals and interrupt pins . . . 16
Figure 3. Generic State Machine . . . 41
Figure 4. State Machine in LIS3DSH . . . 42
Figure 5. Signal block . . . 42
Figure 6. Vector filter . . . 43
Figure 7. State Machine structure . . . 44
Figure 8. Simple State Machine . . . 46
Figure 9. Single state description . . . 46
Figure 10. SSYNC - SM1+SM2 for 32 states SM . . . 63
Figure 11. SSYNC - SM2 used as subroutine of SM1 . . . 64
Figure 12. Axis mask structure . . . 69
Figure 13. Example of signed and unsigned thresholds . . . 70
Figure 14. Temporary mask example . . . 71
Figure 15. Peak detection example . . . 73
Figure 16. Toggle State Machine . . . 75
Figure 17. Toggle output . . . 75
Figure 18. Wake-up State Machine . . . 76
Figure 19. Wake-up output . . . 77
Figure 20. Freefall State Machine . . . 78
Figure 21. Freefall Output . . . 79
Figure 22. Double-turn State Machine . . . 80
Figure 23. Double-turn output . . . 81
Figure 24. Double-tap State Machine . . . 82
Figure 25. Double-tap output . . . 83
Figure 26. 6D positions . . . 83
Figure 27. 6D State Machine . . . 85
Figure 28. FIFO_EN connection block diagram . . . 88
Figure 29. FIFO mode behavior . . . 91
Figure 30. Stream mode fast reading behavior . . . 92
Figure 31. Stream mode slow reading behavior . . . 93
Figure 32. Stream mode slow reading zoom . . . 93
Figure 33. Stream-to-FIFO mode: interrupt not latched . . . 94
Figure 34. Stream-to-FIFO mode: interrupt latched . . . 95
Figure 35. Bypass-to-Stream mode . . . 96
Figure 36. Watermark behavior - FTH[4:0] = 10 . . . 97
Figure 37. FIFO reading diagram - FTH[4:0] = 10 . . . 98

# 1 Operating modes

The LIS3DSH provides two different operating modes: Power-down mode and Normal mode.

After power supply is applied, the LIS3DSH performs a 10 ms boot procedure to load trimming parameters from internal Flash memory. After the boot is completed, the device is automatically configured in Power-down mode.

Referring to the LIS3DSH datasheet, output data rate (ODR) and Zen, Yen, Xen bits of the CTRL_REG4 register are used to select the operating modes (Power-down and Normal mode) and the output data rate (see *Table 1*).

**Table 1. Data rate configuration**

| ODR3 | ODR2 | ODR1 | ODR0 | ODR selection |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | Power-down |
| 0 | 0 | 0 | 1 | 3.125 Hz |
| 0 | 0 | 1 | 0 | 6.25 Hz |
| 0 | 0 | 1 | 1 | 12.5 Hz |
| 0 | 1 | 0 | 0 | 25 Hz |
| 0 | 1 | 0 | 1 | 50 Hz |
| 0 | 1 | 1 | 0 | 100 Hz |
| 0 | 1 | 1 | 1 | 400 Hz |
| 1 | 0 | 0 | 0 | 800 Hz |
| 0 | 0 | 0 | 1 | 1600 Hz |

**Table 2. Power consumption**

| ODR (Hz) | Current consumption (µA) @ Vdd=2.5 V [typ.] |
|---|---|
| Power-down | 2 |
| 3.125 | 11 |
| 6.25 | 19 |
| 12.5 | 35 |
| 25 | 67 |
| 50 | 119 |
| 100 | 225 |
| 400 | 225 |
| 800 | 225 |
| 1600 | 225 |

*Table 2.* shows typical values of power consumption for the different operating modes.

## 1.1 Power-down mode

When the device is in Power-down mode, almost all internal blocks are switched off to minimize power consumption. Digital interfaces ($I^2C$ and SPI) are still active to allow communication with the device. The content of the configuration registers is preserved and output data registers are not updated, therefore keeping the last data sampled in the memory before switching to Power-down mode.

## 1.2 Normal mode

In Normal mode, data are generated at the selected output data rate (ODR) through the ODR bits. Nine different ODR configurations are available in Normal mode, from 3.125 Hz to 1600 Hz.

## 1.3 Switch mode timing

Turn-on times of the LIS3DSH accelerometer are shown in *Table 3.* Their values depend on ODR and bandwidth selected.

**Table 3. Turn-on times**

| ODR [Hz] | Analog filter BW = 800 Hz | Analog filter BW = 400 Hz | Analog filter BW = 200 Hz | Analog filter BW = 50 Hz |
|---|---|---|---|---|
| 1600 | 3/ODR | 4/ODR | 8/ODR | 26/ODR |
| 800 | 2/ODR | 3/ODR | 5/ODR | 14/ODR |
| 400 | 2/ODR | 2/ODR | 3/ODR | 8/ODR |
| 100 | 2/ODR | 2/ODR | 2/ODR | 3/ODR |
| [50 ... 3.125] | 1/ODR | 1/ODR | 1/ODR | 1/ODR |

# 2 Startup 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. after approximately 5 milliseconds, the device automatically enters Power-down mode. To turn on the device and gather acceleration data, it is necessary to select one of the operating modes and enable at least one of the axes through the CTRL_REG4 register.

The following general purpose sequence can be used to configure the device:

1. Write CTRL_REG4 = 67h // X, Y, Z enabled, ODR = 100 Hz
2. Write CTRL_REG3 = C8h // DRY active high on INT1 pin

## 2.1 Reading acceleration data

### 2.1.1 Using the status register

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

1. Read STATUS
2. If STATUS(3) = 0, then go to 1
3. If STATUS(7) = 1, then some data have been overwritten
4. Read OUT_X_L
5. Read OUT_X_H
6. Read OUT_Y_L
7. Read OUT_Y_H
8. Read OUT_Z_L
9. Read OUT_Z_H
10. Data processing
11. Go to 1

The check performed at step 3 allows the user to understand whether the reading rate is adequate compared to the data generation rate. In the case one or more acceleration samples have been overwritten by new data, because of a too slow reading rate, the ZYXOR bit of STATUS is set to 1.

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

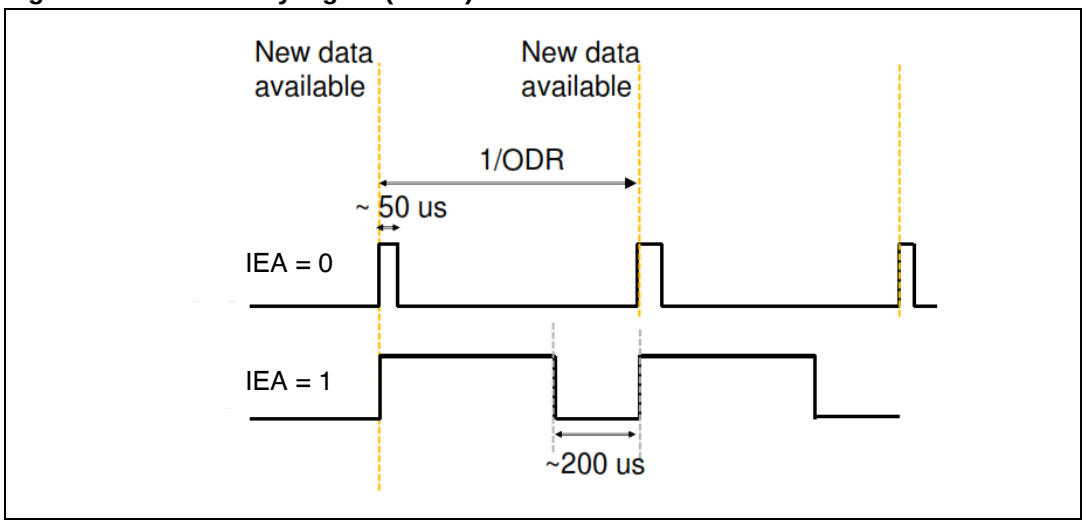
### 2.1.2 Using the data-ready (DRY) signal

The device may be configured to have one HW signal to determinate when a new set of measurement data is available for reading. The signal can be driven to the INT1 pin by

setting the DR_EN bit of CTRL_REG3. Signal polarity is set through the IEA bit and signal shape through the IEL bit of CTRL_REG3.

*Figure 1* shows the behavior of the data-ready when the IEA bit is set to 1 in combination with the setting of the IEL bit. The signal rises to 1 when a new set of angular rate data has been generated and it is available for reading. The interrupt is reset when the higher part of one of the enabled channels has been read (29h, 2Bh, 2Dh).

**Figure 1. Data-ready signal (IA = 1)**




### 2.1.3 Using the block data update (BDU) feature

If the reading of the angular rate data is particularly slow and cannot be synchronized (or it is not required) with either the XYZDA bit in the STATUS register or with the DRDY signal, it is strongly recommended to set the BDU (block data update) bit to 1 in the CTRL_REG4 register.

This feature avoids the reading of values (most significant and least significant parts of the angular rate data) related to different samples. In particular, when the BDU is activated, the data registers related to each channel always contain the most recent angular rate data produced by the device, but, if the reading of a given pair (i.e. OUT_X_H and OUT_X_L, OUT_Y_H and OUT_Y_L, OUT_Z_H and OUT_Z_L) is initiated, the refresh for that pair is blocked until both MSB and LSB parts of the data are read.

*Note: BDU only guarantees that OUT_X(Y, Z)_L and OUT_X(Y,Z)_H have been sampled at the same time. For example, if the reading speed is too low, it may read X and Y sampled at T1 and Z sampled at T2.*

## 2.2 Understanding acceleration data

The measured acceleration data are sent to OUT_X_H, OUT_X_L, OUT_Y_H, OUT_Y_L, OUT_Z_H, and OUT_Z_L registers. These 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 OUT_X_H & OUT_X_L (OUT_Y_H & OUT_Y_L, OUT_Z_H & OUT_Z_L) and it is expressed in 2's complement number.

### 2.2.1 Data alignment

Acceleration data are represented as 16-bit numbers.

### 2.2.2 Example of acceleration data

*Table 4* provides a few basic examples of the data that is read in the data registers when the device is subject to a given angular rate. The values listed in the table are given under the hypothesis of perfect device calibration (i.e. no offset, no gain error,....).

**Table 4. Output data registers content vs. acceleration (FS = 2 g)**

| Acceleration values | Register address | |
|---|---|---|
| | 28h | 29h |
| 2000 mg | 0xFFh | 0x7Fh |
| 1000 mg | 0x00h | 0x40h |
| 0 mg | 0x00h | 0x00h |
| -1000 mg | 0x00h | 0xC0h |
| -2000 mg | 0x00h | 0x80h |

# 3 Interrupt generation

The LIS3DSH can be configured to generate interrupt signals activated by user defined motion patterns. To do this, pins 9 and 11 are used respectively as INT2 and INT1.

Interrupt signals are the main results of the two State Machines; they are triggered when output/stop/continue states are reached in one of the two State Machines. When an interrupt occurs, the INT_SM1 or the INT_SM2 bit on the STAT register (18h) is updated.

Both State Machine 1 and State Machine 2 can be routed to INT1 and INT2, by setting SM1_INT and SM2_INT bits in the CTRL_REG1 and CTRL_REG2 registers.

Moreover, the device may be configured to have a HW signal to determine when a new set of measurement data is available for reading. By setting the DR_EN bit to '1' in the CTRL_REG3 register (23h), the data-ready signal is routed to INT1 and the DRDY bit in the STAT register (18h) is updated according to the status.

Interrupt signal polarity is set through the IEA bit while the signal shape (latched/pulsed) is set through the IEL bit in the CTRL_REG3 register (23h). When the interrupt is pulsed, it has a fixed duration of 50 $\mu$s.

An interrupt on the INT1 pin can also be generated when FIFO buffer is used, such as for a programmable Watermark level, FIFO empty or FIFO full events (see CTRL_REG6 register, 25h).

Finally, interrupts can be enabled/disabled by setting bits INT2_EN and INT1_EN in the CTRL_REG3 register (23h).

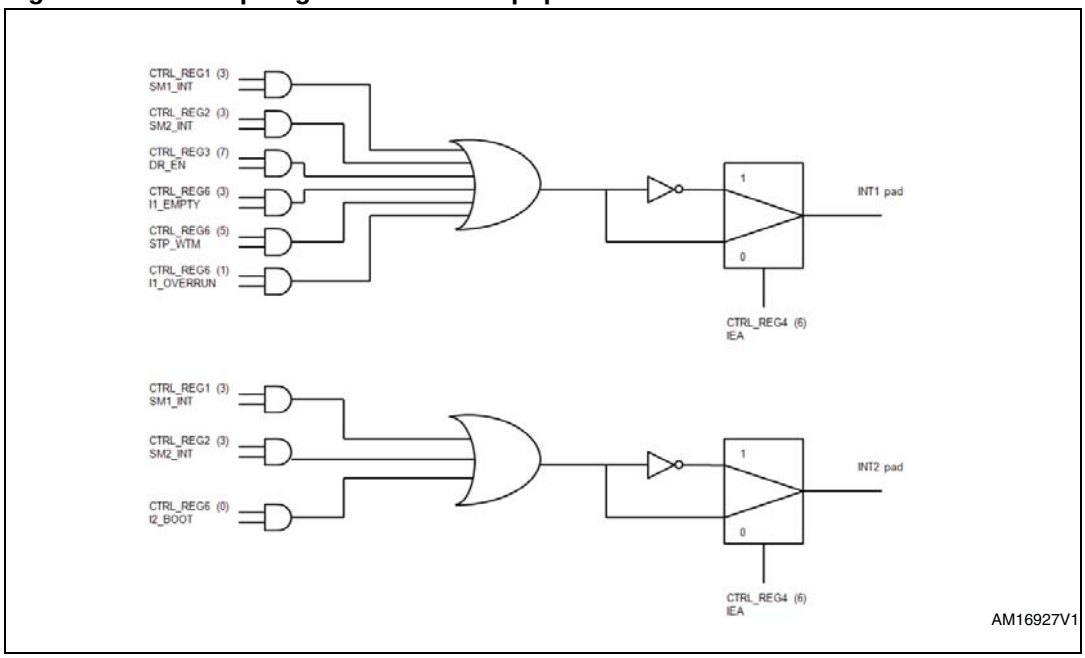
*Table 5* reports all the interrupt bits in the LIS3DSH.

*Figure 2* shows how the interrupt signals can be routed to the interrupt pins.

**Table 5. Interrupt bits**

| Bit | Register | Behavior |
|---|---|---|
| INT_SM1 | STAT (18h) | Updated when INT1 occurs |
| INT_SM2 | STAT (18h) | Updated when INT2 occurs |
| SM1_INT | CTRL_REG1 (21h) | State Machine 1 interrupt routed to INT1/INT2 |
| SM2_INT | CTRL_REG2 (22h) | State Machine 2 interrupt routed to INT1/INT2 |
| DR_EN | CTRL_REG3 (23h) | Enable/disable data-ready signal (routed to INT1) |
| IEA | CTRL_REG3 (23h) | Define interrupt signal polarity (active LOW / active HIGH) |
| IEL | CTRL_REG3 (23h) | Define interrupt signal shape: latched / pulsed |
| I1_EMPTY | CTRL_REG6 (25h) | Enable FIFO Empty indication on INT1 pin |
| I1_WTM | CTRL_REG6 (25h) | Enable FIFO Watermark interrupt on INT1 pin |
| I1_OVERRUN | CTRL_REG6 (25h) | Enable FIFO Overrun interrupt on INT1 pin |
| I2_BOOT | CTRL_REG6 (25h) | Enable BOOT interrupt on INT2 pin |

**Figure 2. Interrupt signals and interrupt pins**



# 4 Register description

## 4.1 Register table

**Table 6. Register table**

| Register name | Address | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|---|---|---|---|---|---|---|---|---|---|
| OUT_T | 0Ch | Temp7 | Temp6 | Temp5 | Temp4 | Temp3 | Temp2 | Temp1 | Temp0 |
| INFO1 | 0Dh | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| INFO2 | 0Eh | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| WHO_AM_I | 0Fh | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| OFF_X | 10h | OFFx_7 | OFFx_6 | OFFx_5 | OFFx_4 | OFFx_3 | OFFx_2 | OFFx_1 | OFFx_0 |
| OFF_Y | 11h | OFFy_7 | OFFy_6 | OFFy_5 | OFFy_4 | OFFy_3 | OFFy_2 | OFFy_1 | OFFy_0 |
| OFF_Z | 12h | OFFz_7 | OFFz_6 | OFFz_5 | OFFz_4 | OFFz_3 | OFFz_2 | OFFz_1 | OFFz_0 |
| CS_X | 13h | CS_7 | CS_6 | CS_5 | CS_4 | CS_3 | CS_2 | CS_1 | CS_0 |
| CS_Y | 14h | CS_7 | CS_6 | CS_5 | CS_4 | CS_3 | CS_2 | CS_1 | CS_0 |
| CS_Z | 15h | CS_7 | CS_6 | CS_5 | CS_4 | CS_3 | CS_2 | CS_1 | CS_0 |
| LC_L | 16h | LC_L_7 | LC_L_6 | LC_L_5 | LC_L_4 | LC_L_3 | LC_L_2 | LC_L_1 | LC_L_0 |
| LC_H | 17h | LC_H_7 | LC_H_6 | LC_H_5 | LC_H_4 | LC_H_3 | LC_H_2 | LC_H_1 | LC_H_0 |
| STAT | 18h | LONG | SYNCW | SYNC1 | SYNC2 | INT_SM1 | INT_SM2 | DOR | DRDY |
| PEAK1 | 19h | PKx_7 | PKx_6 | PKx_5 | PKx_4 | PKx_3 | PKx_2 | PKx_1 | PKx_0 |
| PEAK2 | 1Ah | PKx_7 | PKx_6 | PKx_5 | PKx_4 | PKx_3 | PKx_2 | PKx_1 | PKx_0 |
| VFC_1 | 1Bh | VFC1_7 | VFC1_6 | VFC1_5 | VFC1_4 | VFC1_3 | VFC1_2 | VFC1_1 | VFC1_0 |
| VFC_2 | 1Ch | VFC2_7 | VFC2_6 | VFC2_5 | VFC2_4 | VFC2_3 | VFC2_2 | VFC2_1 | VFC2_0 |
| VFC_3 | 1Dh | VFC3_7 | VFC3_6 | VFC3_5 | VFC3_4 | VFC3_3 | VFC3_2 | VFC3_1 | VFC3_0 |
| VFC_4 | 1Eh | VFC4_7 | VFC4_6 | VFC4_5 | VFC4_4 | VFC4_3 | VFC4_2 | VFC4_1 | VFC4_0 |
| THRS3 | 1Fh | THRS3_7 | THRS3_6 | THRS3_5 | THRS3_4 | THRS3_3 | THRS3_2 | THRS3_1 | THRS3_0 |
| CTRL_REG4 | 20h | ODR3 | ODR2 | ODR1 | ODR0 | BDU | ZEN | YEN | XEN |
| CTRL_REG1 | 21h | HYST1_2 | HYST1_1 | HYST1_0 | - | SM1_INT | - | - | SM1_EN |
| CTRL_REG2 | 22h | HYST2_2 | HYST2_1 | HYST2_0 | - | SM2_INT | - | - | SM2_EN |
| CTRL_REG3 | 23h | DR_EN | IEA | IEL | INT2_EN | INT1_EN | VFLIT | Reserved | STRT |
| CTRL_REG5 | 24h | BW2 | BW1 | FSCALE2 | FSCALE1 | FSCALE0 | ST2 | ST1 | SIM |

Table 6. Register table (continued)

| Register name | Address | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|---|---|---|---|---|---|---|---|---|---|
| CTRL_REG6 | 25h | BOOT | FIFO_EN | STP_WTM | IF_ADD_INC | I1_EMPTY | I1_WTM | I1_OVERRUN | I2_BOOT |
| STATUS | 27h | ZYXOR | ZOR | YOR | XOR | ZYXDA | ZDA | YDA | XDA |
| OUT_X_L | 28h | XD7 | XD6 | XD5 | XD4 | XD3 | XD2 | XD1 | XD0 |
| OUT_X_H | 29h | XD15 | XD14 | XD13 | XD12 | XD11 | XD10 | XD9 | XD8 |
| OUT_Y_L | 2Ah | YD7 | YD6 | YD5 | YD4 | YD3 | YD2 | YD1 | YD0 |
| OUT_Y_H | 2Bh | YD15 | YD14 | YD13 | YD12 | YD11 | YD10 | YD9 | YD8 |
| OUT_Z_L | 2Ch | ZD7 | ZD6 | ZD5 | ZD4 | ZD3 | ZD2 | ZD1 | ZD0 |
| OUT_Z_H | 2Dh | ZD15 | ZD14 | ZD13 | ZD12 | ZD11 | ZD10 | ZD9 | ZD8 |
| FIFO_CTRL | 2Eh | FMODE2 | FMODE1 | FMODE0 | WTMP4 | WTMP3 | WTMP2 | WTMP1 | WTMP0 |
| FIFO_SRC | 2Fh | WTM | OVRN_FIFO | EMPTY | FSS4 | FSS3 | FSS2 | FSS1 | FSS0 |
| ST1_X | 40h - 4Fh | ST1_7 | ST1_6 | ST1_5 | ST1_4 | ST1_3 | ST1_2 | ST1_1 | ST1_0 |
| TIM4_1 | 50h | TM_7 | TM_6 | TM_5 | TM_4 | TM_3 | TM_2 | TM_1 | TM_0 |
| TIM3_1 | 51h | TM_7 | TM_6 | TM_5 | TM_4 | TM_3 | TM_2 | TM_1 | TM_0 |
| TIM2_1_L | 52h | TM_7 | TM_6 | TM_5 | TM_4 | TM_3 | TM_2 | TM_1 | TM_0 |
| TIM2_1_H | 53h | TM_15 | TM_14 | TM_13 | TM_12 | TM_11 | TM_10 | TM_9 | TM_8 |
| TIM1_1_L | 54h | TM_7 | TM_6 | TM_5 | TM_4 | TM_3 | TM_2 | TM_1 | TM_0 |
| TIM1_1_H | 55h | TM_15 | TM_14 | TM_13 | TM_12 | TM_11 | TM_10 | TM_9 | TM_8 |
| THRS2_1 | 56h | THS7 | THS6 | THS5 | THS4 | THS3 | THS2 | THS1 | THS0 |
| THRS1_1 | 57h | THS7 | THS6 | THS5 | THS4 | THS3 | THS2 | THS1 | THS0 |
| MASK1_B | 59h | P_X | N_X | P_Y | N_Y | P_Z | N_Z | P_V | N_V |
| MASK1_A | 5Ah | P_X | N_X | P_Y | N_Y | P_Z | N_Z | P_V | N_V |
| SETT1 | 5Bh | P_DET | THR3_SA | ABS | - | - | THR3_MA | R_TAM | SITR |
| PR1 | 5Ch | PP3 | PP2 | PP1 | PP0 | RP3 | RP2 | RP1 | RP0 |
| TC1_L | 5Dh | TC1_7 | TC1_6 | TC1_5 | TC1_4 | TC1_3 | TC1_2 | TC1_1 | ‘TC1_0 |
| TC1_H | 5Eh | TC1_15 | TC1_14 | TC1_13 | TC1_12 | TC1_11 | TC1_10 | TC1_9 | TC1_8 |
| OUTS1 | 5Fh | P_X | N_X | P_Y | N_Y | P_Z | N_Z | P_V | N_V |
| ST2_X | 60h - 6Fh | ST2_7 | ST2_6 | ST2_5 | ST2_4 | ST2_3 | ST2_2 | ST2_1 | ST2_0 |
| TIM4_2 | 70h | TM_7 | TM_6 | TM_5 | TM_4 | TM_3 | TM_2 | TM_1 | TM_0 |
| TIM3_2 | 71h | TM_7 | TM_6 | TM_5 | TM_4 | TM_3 | TM_2 | TM_1 | TM_0 |
| TIM2_2_L | 72h | TM_7 | TM_6 | TM_5 | TM_4 | TM_3 | TM_2 | TM_1 | TM_0 |
| TIM2_2_H | 73h | TM_15 | TM_14 | TM_13 | TM_12 | TM_11 | TM_10 | TM_9 | TM_8 |

Table 6. Register table (continued)

| Register name | Address | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|---|---|---|---|---|---|---|---|---|---|
| TIM1_2_L | 74h | TM_7 | TM_6 | TM_5 | TM_4 | TM_3 | TM_2 | TM_1 | TM_0 |
| TIM1_2_H | 75h | TM_15 | TM_14 | TM_13 | TM_12 | TM_11 | TM_10 | TM_9 | TM_8 |
| THRS2_2 | 76h | THS7 | THS6 | THS5 | THS4 | THS3 | THS2 | THS1 | THS0 |
| THRS1_2 | 77h | THS7 | THS6 | THS5 | THS4 | THS3 | THS2 | THS1 | THS0 |
| DES2 | 78h | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| MASK2_B | 79h | P_X | N_X | P_Y | N_Y | P_Z | N_Z | P_V | N_V |
| MASK2_A | 7Ah | P_X | N_X | P_Y | N_Y | P_Z | N_Z | P_V | N_V |
| SETT2 | 7Bh | P_DET | THR3_SA | ABS | RADI | D_CS | THR3_MA | R_TAM | SITR |
| PR2 | 7Ch | PP3 | PP2 | PP1 | PP0 | RP3 | RP2 | RP1 | RP0 |
| TC2_L | 7Dh | TC2_7 | TC2_6 | TC2_5 | TC2_4 | TC2_3 | TC2_2 | TC2_1 | TC2_0 |
| TC2_H | 7Eh | TC2_15 | TC2_14 | TC2_13 | TC2_12 | TC2_11 | TC2_10 | TC2_9 | TC2_8 |
| OUTS2 | 7Fh | P_X | N_X | P_Y | N_Y | P_Z | N_Z | P_V | N_V |

## 4.2 OUT_T (0x0C)

Table 7. OUT_T

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |

8-bit temperature output register. The value is expressed as 2’s complement.

The resolution is 1 LSB/deg and 00h corresponds to 25 degrees Celsius.

## 4.3 INFO1 (0x0D)

Table 8. INFO1

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |

Read-only information register. Its value is fixed at 0x21.

## 4.4 INFO2 (0x0E)

**Table 9. INFO2**

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Read-only information register. Its value is fixed at 0x00.

## 4.5 WHO_AM_I (0x0F)

**Table 10. WHO_AM_I**

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |

Device identification register. It is a read-only register.

## 4.6 OFF_X (0x10), OFF_Y (0x11), OFF_Z (0x12)

**Table 11. Offset axis**

| BIT | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| **Name** | OFFx_7 | OFFx_6 | OFFx_5 | OFFx_4 | OFFx_3 | OFFx_2 | OFFx_1 | OFFx_0 |
| **Default** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Offset compensation register for single axis. Default value is 0x00. The value is expressed in 2's complement.

Final acceleration output value is composed as:

Output(axis) = Measurement(axis) - OFFSET_x(axis) * 32

Where:

- x = X, Y, Z-axis
- Measurement(axis) = 16-bit raw data for X, Y, Z
- OFFSET_x(axis) = Compensation value from OFF_X, OFF_Y, OFF_Z registers
- OUTPUT(axis) = Acceleration value with offset compensation for output registers and State Machine.

According to the previous formula, the offset on each axis can be compensated from -4095 to 4096 LSB, with steps of 32 LSB.

## 4.7 CS_X (0x13), CS_Y (0x14), CS_Z (0x15)

**Table 12. Constant shift for single axis**

| BIT | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| Name | CS_7 | CS_6 | CS_5 | CS_4 | CS_3 | CS_2 | CS_1 | CS_0 |
| Default | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Constant shift value register for single axis. This value acts as a temporary offset in DIFF-Mode for State Machine 2 only (refer to *Section 5.4.2*). The default value is 0x00. The value is expressed in 2's complement.

## 4.8 LC_L (0x16), LC_H (0x17)

16-bit long-counter registers common for both State Machines.

**Table 13. Status of long counter LSB (0x16)r**

| BIT | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| Name | LC_L_7 | LC_L_6 | LC_L_5 | LC_L_4 | LC_L_3 | LC_L_2 | LC_L_1 | LC_L_0 |
| Default | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

**Table 14. Status of long counter MSB (0x17)**

| BIT | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| Name | LC_H_7 | LC_H_6 | LC_H_5 | LC_H_4 | LC_H_3 | LC_H_2 | LC_H_1 | LC_H_0 |
| Default | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Table 15. Status of long counter values**

| LC values | Condition |
|---|---|
| = -01h | Not valid value, counting Stopped |
| = 00h | Counter Full, interrupt happens and counter set to -01h |
| > 00h | Counting |

The value of the long counter is expressed in 2's complement.

This value is decreased whenever the DEC opcode is executed in the State Machine and the counter value is higher or equal to zero (see *Section 6.2.12*).

To stop counting, the value -01h must be written in these registers.

When the long counter is full (00h), the LONG bit is set to 1 in the STAT register (18h). The following state for the long counter is -01h (counter stopped).

Reading of LC registers resets the LONG bit in the STAT register (18h) to the default value (0).

## 4.9 STAT (0x18)

**Table 16. STAT register**

| BIT | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| **Name** | LONG | SYNCW | SYNC1 | SYNC2 | INT_SM1 | INT_SM2 | DOR | DRDY |
| **Default** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Table 17. STAT register description**

| Bit name | Description |
|---|---|
| **LONG** | 0= no interrupt, 1= LongCounter interrupt flag.<br>Common to both State Machines.<br>LONG flag is reset to default value by reading LC registers (16h and 17h). |
| **SYNCW** | Common information for OUTW host action waiting.<br>0 = no action waiting from Host.<br>1 = Host action is waiting after OUTW command.<br>This bit is reset to 0 whenever OUTS1/OUTS2 is read. |
| **SYNC1** | 0 = State Machine 1 running normally, 1 = State Machine 1 stopped and waiting for restart request from State Machine 2. |
| **SYNC2** | 0 = State Machine 2 running normally, 1 = State Machine 2 stopped and waiting for restart request from State Machine 1. |
| **INT_SM1** | 0 = no interrupt on State Machine 1, 1 = State Machine 1 interrupt happened. The interrupt signal is reset when the OUTS1 register is read. |
| **INT_SM2** | 0 = no interrupt on State Machine 2, 1 = State Machine 2 interrupt happened. The interrupt signal is reset when OUTS2 register is read. |
| **DOR** | The Data OverRun bit indicates when a new set of data has overwritten the previous one in output registers.<br>0 = no overrun, 1 = data overrun.<br>The overrun bit is automatically cleared when data are read and no new data have been produced in the meantime. |
| **DRDY** | 0 = data not ready, 1 = data ready.<br>New data are ready in output registers (refer to *Section 2.1*). |

## 4.10 PEAK1 (0x19), PEAK2 (0x1A)

**Table 18. PEAK1, 2 register description**

| BIT | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| Name | PKx_7 | PKx_6 | PKx_5 | PKx_4 | PKx_3 | PKx_2 | PKx_1 | PKx_0 |
| Default | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Peak 1 value for State Machine 1, default value: 0x00.

Peak 2 value for State Machine 2, default value: 0x00.

Peak register stores the highest absolute peak value detected.

Peak value is reset when the REL command occurs or new initial start occurs.

The value of the peak counter is expressed in 2's complement.

For more information about peak detection refer to *Section 8*.

## 4.11 Vector filter coefficients (0X1B-0X1E)

**Table 19. VFC register description**

| Add | Mnemonic | Definition | Default |
|---|---|---|---|
| 0X1B | VFC_1 | Coefficient 1 | 0x00 |
| 0X1C | VFC_2 | Coefficient 2 | 0x00 |
| 0X1D | VFC_3 | Coefficient 3 | 0x00 |
| 0X1E | VFC_4 | Coefficient 4 | 0x00 |

The vector filter is a $7^{th}$-order anti-symmetric FIR filter. The 8 taps have a 4x2 structure: VFC_1, VFC_2, VFC_3, VFC_4 and -VFC_1, -VFC_2, -VFC_3, -VFC_4.

The vector filter can be enabled or disabled by the VFILT bit in the CTRL_REG3 register.

For more information about the vector filter refer to *Section 5.3.3*.

## 4.12 THRS3 (0X1F)

**Table 20. THRS3 register description**

| BIT | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| Name | THRS3_7 | THRS3_6 | THRS3_5 | THRS3_4 | THRS3_3 | THRS3_2 | THRS3_1 | THRS3_0 |
| Default | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Common threshold for overrun detection. The value is always unsigned (ABS) regardless of ABS settings in the SETT1/SETT2 registers. So, the THRS3 value is symmetric to the zero level.

When the acceleration of any axis exceeds the THRS3 limit, State Machines are reset (PPx = RPx). Reset of State Machines is enabled through THR3_xA bits in the SETT1/SETT2 registers.

## 4.13 CTRL_REG4 (0X20)

**Table 21. Control register 1 description**

| BIT | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| Name | ODR3 | ODR2 | ODR1 | ODR0 | BDU | ZEN | YEN | XEN |
| Default | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |

**Table 22. Description control register 5**

| | |
|---|---|
| **ODR <3:0>** | Data rate selection. Default value: 0X00<br>(Other: refer to *Table 23*). |
| **BDU** | Block data update. Default value: 0<br>1: Continuous update<br>0: Output register not updated until MSB and LSB reading.<br>For more information about BDU, refer to *Section 2.1.3*. |
| **Zen** | Default value: 1<br>1: Z-axis enable<br>0: Z-axis disable |
| **Yen** | Default value: 1<br>1: Y-axis enable<br>0: Y-axis disable |
| **Xen** | Default value: 1<br>1: X-axis enable<br>0: X-axis disable |

**Table 23. Data rate**

| ODR3 | ODR2 | ODR1 | ODR0 | ODR selection |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | Power-down |
| 0 | 0 | 0 | 1 | 3.125 Hz |
| 0 | 0 | 1 | 0 | 6.25 Hz |
| 0 | 0 | 1 | 1 | 12.5 Hz |
| 0 | 1 | 0 | 0 | 25 Hz |
| 0 | 1 | 0 | 1 | 50 Hz |

Table 23. Data rate

| ODR3 | ODR2 | ODR1 | ODR0 | ODR selection |
|---|---|---|---|---|
| 0 | 1 | 1 | 0 | 100 Hz |
| 0 | 1 | 1 | 1 | 400 Hz |
| 1 | 0 | 0 | 0 | 800 Hz |
| 1 | 0 | 0 | 1 | 1600 Hz |

## 4.14 CTRL_REG1 (0X21)

State Machine 1 interrupt configuration register.

Table 24. Control register 2 description

| BIT | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| Name | HYST1_2 | HYST1_1 | HYST1_0 | - | SM1_INT | - | - | SM1_EN |
| Default | 0 | 0 | 0 | - | 0 | - | - | 0 |

Table 25. Control register 2 bit description

| Bit name | Description |
|---|---|
| **HYST1<2:0>** | Hysteresis which is added or subtracted from the threshold values (THRS1_1 and THRS2_1) of State Machine 1.<br>000 = 0 (default)<br>111 = 7 (maximum Hysteresis)<br>Hysteresis value is unsigned.<br>The Hysteresis value is added or subtracted according to the condition to evaluate (see *Section 6.1*). |
| **SM1_INT** | 0 = State Machine 1 interrupt routed to INT1.<br>1 = State Machine 1 interrupt routed to INT2. |
| **SM1_EN** | 0 = State Machine 1 disabled. Temporary memories and registers related to this State Machine are left intact.<br>1 = State Machine 1 enabled. |

## 4.15 CTRL_REG2 (0X22)

State Machine 2 interrupt configuration register.

Table 26. Control register 3 description

| BIT | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| Name | HYST2_2 | HYST2_1 | HYST2_0 | - | SM2_INT | - | - | SM2_EN |
| Default | 0 | 0 | 0 | - | 0 | - | - | 0 |

Table 27. Control register 3 bit description

| Bit name | Description |
|---|---|
| HYST2<2:0> | Hysteresis which is added or subtracted from the threshold values (THRS1_2 and THRS2_2) of State Machine 2.<br>000 = 0 (default)<br>111 = 7 (maximum Hysteresis)<br>Hysteresis value is unsigned.<br>The Hysteresis value is added or subtracted according to the condition to evaluate (see *Section 6.1*). |
| SM2_INT | 0 = State Machine 2 interrupt routed to INT1.<br>1 = State Machine 2 interrupt routed to INT2. |
| SM2_EN | 0 = State Machine 2 disabled. Temporary memories and registers related to this State Machine are left intact.<br>1 = State Machine 2 enabled. |

## 4.16 CTRL_REG3 (0X23)

Table 28. Control register 4 description

| Bit | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| Name | DR_EN | IEA | IEL | INT2_EN | INT1_EN | VFLIT | Reserved | STRT |
| Default | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Table 29. Control register 4 bit description

| Bit name | Description |
|---|---|
| DR_EN | 0 = Data-ready interrupt disabled.<br>1 = Data-ready interrupt enabled and routed to INT1. |
| IEA | 0 = Interrupt signal active LOW.<br>1 = Interrupt signal active HIGH. |
| IEL | 0 = Interrupt latched.<br>1 = Interrupt pulsed (refer to *Section 2.1.2*). |
| INT2_EN | 0 = INT2 signal disabled (High-Z state).<br>1 = INT2 signal enabled (signal pin fully functional). |
| INT1_EN | 0 = INT1 (DRDY) signal disabled (High-Z state).<br>1 = INT1 (DRDY) signal enabled (signal pin fully functional).<br>Note: DR_EN bit in CTRL_REG3 register should be taken into account too. |
| VFLIT | 0 = Vector filter disabled.<br>1 = Vector filter enabled. |
| STRT | Soft Reset: it resets the whole internal logic circuitry when set to 1. It automatically returns to 0. |

## 4.17 CTRL_REG5 (0X24)

**Table 30. Control register 5 description**

| Bit | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| Name | BW2 | BW1 | FSCALE2 | FSCALE1 | FSCALE0 | ST2 | ST1 | SIM |
| Default | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Table 31. Control register 5 bit description**

| Bit name | Description |
|---|---|
| **BW<2:1>** | Anti aliasing filter bandwidth.<br>Default value: 00 (00: 800 Hz; 01: 40 Hz; 10: 200 Hz; 11: 50 Hz) |
| **FSCALE<2:0>** | Full scale selection. Default value: 000<br>(000: +/-2G; 001 +/- 4G; 0010: +/- 6G; 011: +/- 8G; 100: +/- 16G) |
| **ST<2:1>** | Self-test Enable. Default value: 00.<br>(00: Self-test Disabled; Other: see *Table 32* . |
| **SIM** | SPI serial internal interface mode selection. Default value: 0<br>(0: 4-wire interface; 1: 3-wire interface). |

**Table 32. Self-test mode**

| ST2 | ST1 | Self-test mode |
|---|---|---|
| 0 | 0 | Normal mode |
| 0 | 1 | Positive sign self-test |
| 1 | 0 | Negative sign-test |
| 1 | 1 | Not Allowed |

## 4.18 CTRL_REG6 (0X25)

**Table 33. Control register 6 description**

| Bit | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| Name | BOOT | FIFO_EN | STP_WTM | IF_ADD_INC | I1_EMPTY | I1_WTM | I1_OVERRUN | I2_BOOT |
| Default | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Table 34. Control register 6 bit description**

| Bit name | Description |
|---|---|
| **BOOT** | Force reboot, cleared as soon as the reboot is finished. Active High |

Table 34. Control register 6 bit description

| | |
|---|---|
| **FIFO_EN** | FIFO Enable. Default value: 0.<br>(0: disable; 1: enable) |
| **STP_WTM** | Stop on Watermark - FIFO depth can be limited at the Watermark value, by setting to “1” the STP_WTM bit. Default value: 0.<br>(0: disable; 1: enable) |
| **IF_ADD_INC** | Register address automatically increased during a multiple byte access with a serial interface (I2C or SPI)<br>(0: disable; 1: enable) |
| **I1_EMPTY** | Enable FIFO Empty indication on INT1 pin. Default value 0.<br>(0: disable; 1: enable) |
| **I1_WTM** | FIFO Watermark interrupt on INT1 pin. Default value 0.<br>(0: disable; 1: enable) |
| **I1_OVERRUN** | FIFO Overrun interrupt on INT1 pin. Default value 0.<br>(0: disable; 1: enable) |
| **I2_BOOT** | BOOT interrupt on INT2 pin. Default value 0.<br>(0: disable; 1: enable) |

## 4.19 STATUS (0X27)

Table 35. Status register description

| BIT | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| Name | ZYXOR | ZOR | YOR | XOR | ZYXDA | ZDA | YDA | XDA |
| Default | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Table 36. Status register bit description

| Bit name | Description |
|---|---|
| **ZYXOR** | X, Y and Z-axis Data Overrun. Default value: 0<br>(O: no Overrun has occurred; 1: a new set of data has overwritten the previous ones |
| **ZOR** | Z-axis Data Overrun. Default value: 0<br>(0: no Overrun has occurred; 1: a new data for the Z-axis has overwritten the previous ones) |
| **YOR** | Y-axis Data Overrun. Default value: 0<br>(0: no Overrun has occurred; 1: a new data for the Y-axis has overwritten the previous ones) |
| **XOR** | X-axis Data Overrun. Default value: 0<br>(0: no Overrun has occurred; 1: a new data for the X-axis has overwritten the previous ones) |
| **ZYXDA** | X, Y and Z-axis new Data Available. Default value: 0<br>(0: a new set of data is not yet available; 1: a new set of data is available) |

Table 36. Status register bit description (continued)

| | |
|---|---|
| **ZDA** | Z-axis new data available. Default value: 0<br>(0: a new data for the Z-axis is not yet available; 1: a new data for Z-axis is available) |
| **YDA** | Y-axis new data available. Default value: 0<br>(0: a new data for the Y-axis is not yet available; 1: a new data for Y-axis is available) |
| **XDA** | X-axis new data available. Default value: 0<br>(0: a new data for the X-axis is not yet available; 1: a new data for X-axis is available) |

## 4.20 OUT_X_L (0X28), OUT_X_H (0x29)

X-axis acceleration data (16bit), MSB values are in OUT_X_H, LSB values are in OUT_X_L. The value is expressed in 2's complement.

## 4.21 OUT_Y_L (0X2A), OUT_Y_H (0x2B)

Y-axis acceleration data (16bit), MSB values are in OUT_Y_H, LSB values are in OUT_Y_L. The value is expressed in 2's complement.

## 4.22 OUT_Z_L (0X2C), OUT_Z_H (0x2D)

Z-axis acceleration data (16bit), MSB values are in OUT_Z_H, LSB values are in OUT_Z_L. The value is expressed in 2's complement.

## 4.23 FIFO_CTRL (0X2E)

Table 37. FIFO_CTRL description

| Bit | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| Name | FMODE2 | FMODE1 | FMODE0 | WTMP4 | WTMP3 | WTMP2 | WTMP1 | WTMP0 |
| Default | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Table 38. FIFO_CTRL bit description

| Bit name | Description |
|---|---|
| **FMODE<2:0>** | FIFO mode. Default value: 0<br>(see *Table 38* for FIFO modality) |
| **WTMP<4:0>** | FIFO Watermark pointer. It is the FIFO depth when the Watermark is enabled (see *Section 10.4*). |

Table 39. FIFO MODE description

| FMODE2 | FMODE1 | FMODE0 | Mode description |
|---|---|---|---|
| 0 | 0 | 0 | Bypass mode. FIFO turned off. |
| 0 | 0 | 1 | FIFO mode. Stop collecting data when FIFO is full. |
| 0 | 1 | 0 | Stream mode. If the FIFO is full, the new sample overwrites the older one (circular buffer). |
| 0 | 1 | 1 | Stream mode until trigger is de-asserted, then FIFO mode. |
| 1 | 0 | 0 | Bypass mode until trigger is de-asserted, then Stream mode. |
| 1 | 0 | 1 | Not to use. |
| 1 | 1 | 0 | Not to use. |
| 1 | 1 | 1 | Bypass mode until trigger is de-asserted, then FIFO mode. |

The FIFO trigger is the INT2 source.

For more information about FIFO refer to *Section 10*.

## 4.24 FIFO_SRC (0X2F)

Table 40. FIFO_CTRL description

| Bit | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| Name | WTM | OVRN_FIFO | EMPTY | FSS4 | FSS3 | FSS2 | FSS1 | FSS0 |
| Default | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Table 41. FIFO_SRC bit description

| Bit name | Description |
|---|---|
| **WTM** | Watermark status.<br>(0: FIFO filling is lower than WTM level; 1: FIFO filling is equal or higher than WTM level) |
| **OVRN_FIFO** | Overrun bit status.<br>(0: FIFO is not completely filled; 1: FIFO is completely filled) |
| **EMPTY** | FIFO empty bit status.<br>(0: FIFO not empty; 1: FIFO empty) |
| **FSS<4:0>** | Number of samples stored in the FIFO - 1 |

For more information about FIFO refer to *Section 10*.

## 4.25 ST1_X (40h-4Fh)

State Machine 1 code register ST1_X (X = 1-16).

State Machine 1 system register is composed of sixteen 8-bit registers. Each register can contain an operational code, as described in *Section 6*.

## 4.26 TIM4_1 (50h)

8-bit unsigned initial value for Timer Counter 1 (5Dh-5Eh).

1LSb = 1/ODR.

**Table 42. Timer4 default value**

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

## 4.27 TIM3_1 (51h)

8-bit unsigned initial value for Timer Counter 1 (5Dh-5Eh).

1LSb = 1/ODR.

**Table 43. Timer3 default value**

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

## 4.28 TIM2_1(52h - 53h)

16-bit unsigned initial value for Timer Counter 1 (5Dh-5Eh).

1LSb = 1/ODR.

**Table 44. TIM2_1_L default value**

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Table 45. TIM2_1_H default value**

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

## 4.29 TIM1_1(54h - 55h)

16-bit unsigned initial value for Timer Counter 1 (5Dh-5Eh).

1LSb = 1/ODR.

**Table 46. TIM1_1_L default value**

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Table 47. TIM1_1_H default value**

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

## 4.30 THRS2_1(56h)

Threshold value for State Machine 1 conditions. Data are in 2's complement.

1LSb = FS/2^7.

**Table 48. HRS2_1 default value**

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

## 4.31 THRS1_1(57h)

Threshold value for State Machine 1 conditions. Data are in 2's complement.

1LSb = FS/2^7.

**Table 49. THRS1_1 default value**

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

## 4.32 MASK1_B(59h)

Swap axis and sign mask for State Machine 1 motion detection operations. For more information refer to *Section 7*.

**Table 50. MASK1_B axis and sign mask register**

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| P_X | N_X | P_Y | N_Y | P_Z | N_Z | P_V | N_V |

**Table 51. MASK1_B register structure**

| | |
|---|---|
| P_X | 0 = X+ disabled, 1 = X+ enabled |
| N_X | 0 = X- disabled, 1 = X- enabled |
| P_Y | 0 = Y+ disabled, 1 = Y+ enabled |
| N_Y | 0 = Y- disabled, 1 = Y- enabled |
| P_Z | 0 = Z+ disabled, 1 = Z+ enabled |
| N_Z | 0 = Z- disabled, 1 = Z- enabled |
| P_V | 0 = V+ disabled, 1 = V+ enabled |
| N_V | 0 = V- disabled, 1 = V- enabled |

## 4.33 MASK1_A(5Ah)

Default axis and sign mask for State Machine 1 motion detection operations. For more information refer to *Section 7*.

**Table 52. MASK1_A axis and sign mask register**

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| P_X | N_X | P_Y | N_Y | P_Z | N_Z | P_V | N_V |

**Table 53. MASK1_A register structure**

| | |
|---|---|
| P_X | 0 = X+ disabled, 1 = X+ enabled |
| N_X | 0 = X- disabled, 1 = X- enabled |
| P_Y | 0 = Y+ disabled, 1 = Y+ enabled |
| N_Y | 0 = Y- disabled, 1 = Y- enabled |
| P_Z | 0 = Z+ disabled, 1 = Z+ enabled |
| N_Z | 0 = Z- disabled, 1 = Z- enabled |
| P_V | 0 = V+ disabled, 1 = V+ enabled |
| N_V | 0 = V- disabled, 1 = V- enabled |

## 4.34 SETT1 (5Bh)

Setting of threshold, peak detection, and flags for State Machine 1 motion detection operations. For more information refer to *Section 7*.

**Table 54. SETT1 register structure**

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| P_DET | THR3_SA | ABS | - | - | THR3_MA | R_TAM | SITR |

**Table 55. SETT1 register description**

| | |
|---|---|
| P_DET | SM1 peak detection bit. Default value: 0<br>0 = peak detection disabled, 1 = peak detection enabled<br>For more information about peak detection refer to *Section 8*. |
| THR3_SA | Default value:0<br>0 = no action, 1 = threshold 3 enabled for axis and sign mask reset (MASKB_1) |
| ABS | Default value:0<br>0 = unsigned thresholds THRSx, 1 = signed thresholds THRSx<br>For more details refer to *Section 7.2*. |
| THR3_MA | Default value:0<br>0 = no action, 1 = threshold 3 enabled for axis and sign mask reset (MASKA_1) |

**Table 55. SETT1 register description**

| | |
|---|---|
| R_TAM | Next condition validation flag. Default value:0<br>0 = mask frozen on the axis that triggers the condition, 1 = standard mask always evaluated.<br>For more details about the temporary axis mask refer to *Section 7.3*. |
| SITR | Default value:0<br>0 = no actions, 1 = STOP and CONT commands generate an interrupt and perform output actions as OUTC command. |

## 4.35 PR1 (5Ch)

Program and reset pointers for State Machine 1.

**Table 56. PR1 register**

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| PP3 | PP2 | PP1 | PP0 | RP3 | RP2 | RP1 | RP0 |

**Table 57. PR1 register description**

| | |
|---|---|
| PP3-PP0 | SM1 program pointer address |
| RP3-RP0 | SM1 reset pointer address |

## 4.36 TC1 (5Dh-5Eh)

16-bit general timer counter for State Machine 1.

**Table 58. TC1_L default value**

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Table 59. TC1_H default value**

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Registers are read-only.

The TC1 counter can be used in State Machine 1 through the conditions defined in *Section 6.1*. Registers TIM1_1 (54h-55h), TIM2_1 (52h-53h), TIM3_1 (51h), and TIM4_1 (50h) define the initial value of the Timer Counter 1.

## 4.37 OUTS1 (5Fh)

Output flags on axis for State Machine 1 management.

Table 60. OUTS1 register

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| P_X | N_X | P_Y | N_Y | P_Z | N_Z | P_V | N_V |

Read action of this register affects interrupt release function.

After reading OUTS1, the value is set to default (00h).

Table 61. OUTS1 register description

| | |
|---|---|
| P_X | 0 = X+ noshow, 1 = X+ show |
| N_X | 0 = X- noshow, 1 = X- show |
| P_Y | 0 = Y+ noshow, 1 = Y+ show |
| N_Y | 0 = Y- noshow, 1 = Y- show |
| P_Z | 0 = Z+ noshow, 1 = Z+ show |
| N_Z | 0 = Z- noshow, 1 = Z- show |
| P_V | 0 = V+ noshow, 1 = V+ show |
| N_V | 0 = V- noshow, 1 = V- show |

For more information about output registers refer to *Section 7.4*.

## 4.38 ST2_X (60h-6Fh)

State Machine 2 code register ST2_X (X = 1-16).

State Machine 2 system register is composed of sixteen 8-bit registers. Each register can contain an operational code, as described in *Section 6*.

## 4.39 TIM4_2 (70h)

8-bit unsigned initial value for Timer Counter 2 (7Dh-7Eh).

1LSb = 1/ODR.

Table 62. Timer4 default value

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

## 4.40 TIM3_2 (71h)

8-bit unsigned initial value for Timer Counter 2 (7Dh-7Eh).

1LSb = 1/ODR.

Table 63. Timer3 default value

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

## 4.41 TIM2_2 (72h - 73h)

16-bit unsigned initial value for Timer Counter 2 (7Dh-7Eh).

1LSb = 1/ODR.

**Table 64. TIM2_1_L default value**

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Table 65. TIM2_1_H default value**

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

## 4.42 TIM1_2 (74h - 75h)

16-bit unsigned initial value for Timer Counter 2 (7Dh-7Eh).

1LSb = 1/ODR.

**Table 66. TIM1_2_L default value**

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Table 67. TIM1_2_H default value**

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

## 4.43 THRS2_2 (76h)

Threshold value for State Machine 2 conditions. Data are in 2’s complement.

1LSb = FS/2^7.

**Table 68. THRS2_2 default value**

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

## 4.44 THRS1_2 (77h)

Threshold value for State Machine 2 conditions. Data are in 2’s complement.

1LSb = FS/2^7.

**Table 69. THRS1_2 default value**

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

## 4.45 DES2 (78h)

Decimation counter value for State Machine 2. More information in *Section 5.4*.

**Table 70. DES2 default value**

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

## 4.46 MASK2_B (79h)

Axis and sign mask (swap) for State Machine 2 motion detection operation. For more information refer to *Section 7*.

**Table 71. MASK2_B axis and sign mask register**

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| P_X | N_X | P_Y | N_Y | P_Z | N_Z | P_V | N_V |

**Table 72. MASK2_B register description**

| | |
|---|---|
| P_X | 0 = X+ disabled, 1 = X+ enabled |
| N_X | 0 = X- disabled, 1 = X- enabled |
| P_Y | 0 = Y+ disabled, 1 = Y+ enabled |
| N_Y | 0 = Y- disabled, 1 = Y- enabled |
| P_Z | 0 = Z+ disabled, 1 = Z+ enabled |
| N_Z | 0 = Z - disabled, 1 = Z- enabled |
| P_V | 0 = V+ disabled, 1 = V+ enabled |
| N_V | 0 = V- disabled, 1 = V- enabled |

## 4.47 MASK2_A (7Ah)

Axis and sign mask (default) for State Machine 2 motion detection operation. For more information refer to *Section 7*.

**Table 73. MASK2_A axis and sign mask register**

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| P_X | N_X | P_Y | N_Y | P_Z | N_Z | P_V | N_V |

**Table 74. MASK2_A register description**

| | |
|---|---|
| P_X | 0 = X+ disabled, 1 =X+ enabled |
| N_X | 0 = X- disabled, 1 = X- enabled |
| P_Y | 0 = Y+ disabled, 1 = Y+ enabled |
| N_Y | 0 = Y- disabled, 1 = Y- enabled |

Table 74. MASK2_A register description (continued)

| | |
|---|---|
| P_Z | 0 = Z+ disabled, 1 = Z+ enabled |
| N_Z | 0 = Z- disabled, 1 = Z- enabled |
| P_V | 0 = V+ disabled, 1 = V+ enabled |
| N_V | 0 = V- disabled, 1 = V- enabled |

## 4.48 SETT2 (7Bh)

Setting of threshold, peak detection, and flags for State Machine 2 motion detection operations.

Table 75. SETT2 register

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| P_DET | THR3_SA | ABS | RADI | D_CS | THR3_MA | R_TAM | SITR |

Table 76. SETT2 register description

| | |
|---|---|
| P_DET | SM2 peak detection. Default value:0<br>0 = peak detection disabled, 1 = peak detection enabled.<br>For more information about peak detection refer to *Section 8*. |
| THR3_SA | Default value:0<br>0 = no action, 1 = threshold 3 limit value for axis and sign mask reset (MASK2_B) |
| ABS | Default value:0<br>0 = unsigned thresholds, 1 = signed thresholds<br>For more details refer to *Section 7.2*. |
| RADI | 0 = raw data; 1 = diff data for State Machine 2 |
| D_CS | 0 = DIFF2 enabled (difference between current data and previous data),<br>1 = constant shift enabled (difference between current data and constant values) |
| THR3_MA | Default value:0<br>0 = no action, 1 = threshold 3 enabled for axis and sign mask reset (MASK2_A) |
| R_TAM | Next condition validation flag. Default value:0<br>0 = mask frozen on the axis that triggers the condition, 1 = standard mask always evaluated.<br>For more details about the temporary axis mask refer to *Section 7.3*. |
| SITR | Default value:0<br>0 = no actions, 1 = STOP and CONT commands generate an interrupt and perform output actions as OUTC command. |

## 4.49 PR2 (7Ch)

Program and reset pointers for State Machine 2.

**Table 77. PR2 register**

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| PP3 | PP2 | PP1 | PP0 | RP3 | RP2 | RP1 | RP0 |

**Table 78. PR2 register description**

| | |
|---|---|
| PP3-PP0 | SM2 program pointer address |
| RP3-RP0 | SM2 reset pointer address |

## 4.50 TC2 (7Dh-7E)

16-bit general Timer Counter for State Machine 2.

**Table 79. TC2_L default value**

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Table 80. TC2_H default value**

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Registers are read-only.

The TC2 counter can be used in State Machine 2 through the conditions defined in *Section 6.1*. Registers TIM1_2 (74h-75h), TIM2_2 (72h-73h), TIM3_2 (71h), and TIM4_2 (70h) define the initial value of the Timer Counter 2.

## 4.51 OUTS2 (7Fh)

Output flags on axis for State Machine 1 management.

**Table 81. OUTS2 register**

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| P_X | N_X | P_Y | N_Y | P_Z | N_Z | P_V | N_V |

Read action of this register affects interrupt release function.

After reading OUTS2, the value is set to default (00h).

**Table 82. OUTS2 register description**

| | |
|---|---|
| P_X | 0 = X+ noshow, 1 = X+ show |
| N_X | 0 = X- noshow, 1 = X- show |
| P_Y | 0 = Y+ noshow, 1 = Y+ show |
| N_Y | 0 = Y- noshow, 1 = Y- show |
| P_Z | 0 = Z+ noshow, 1 = Z+ show |

Table 82. OUTS2 register description (continued)

| | |
|---|---|
| N_Z | 0 = Z- noshow, 1 = Z- show |
| P_V | 0 = V+ noshow, 1 = V+ show |
| N_V | 0 = V- noshow, 1 = V- show |

For more information about output registers refer to *Section 7.4*.

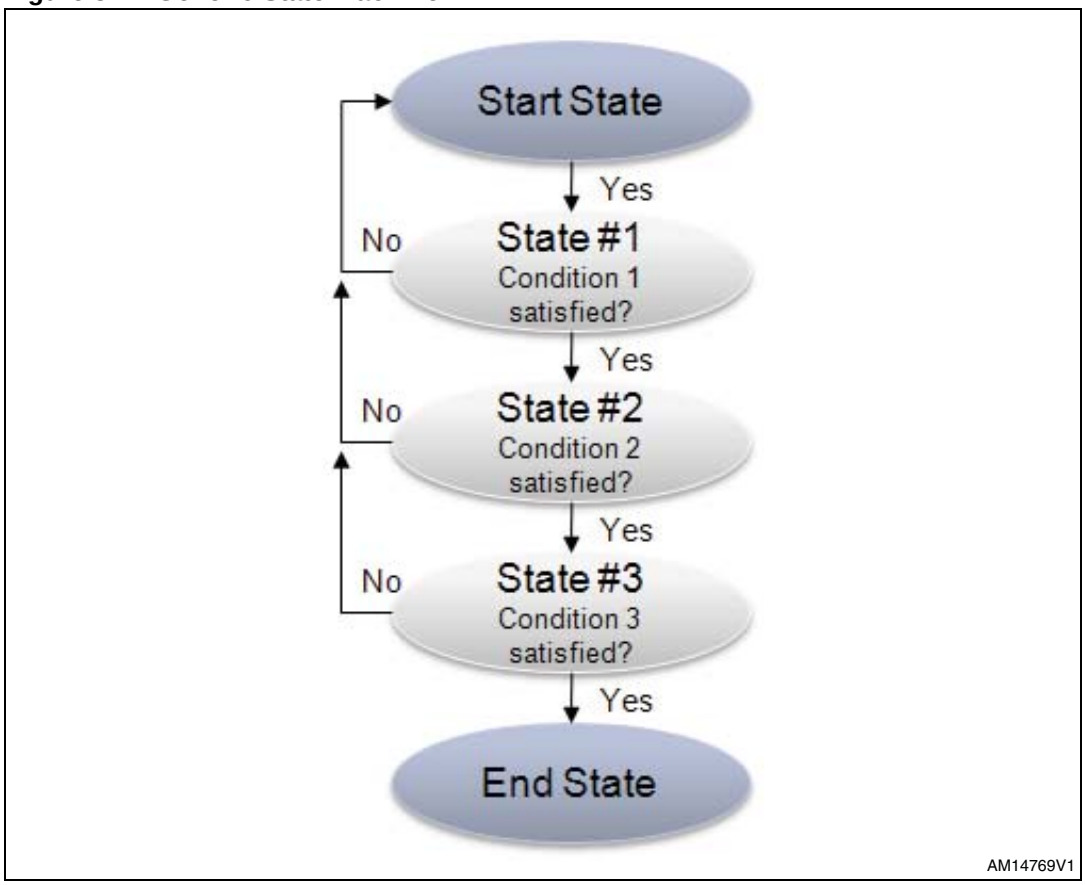
# 5 State Machine

## 5.1 State Machine definition

State Machine is a mathematical abstraction used to design logic connections. It is a behavioral model composed of a finite number of states and transitions between states, similar to a flow chart in which one can inspect the way logic runs when certain conditions are met. The State Machine begins with a Start state (or 0 state), goes to different states through transitions dependent on the inputs, and can finally end in a specific state (called Stop state). The current state is determined by the past states of the system.

*Figure 3* shows a generic State Machine.

**Figure 3. Generic State Machine**



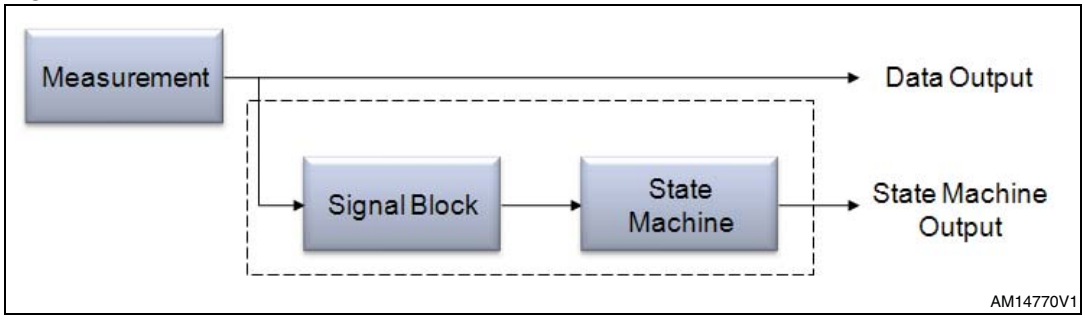

## 5.2 State Machine in LIS3DSH

The LIS3DSH works as a normal accelerometer, generating acceleration output data. However, these data can be used to perform a program in the embedded State Machine (*Figure 4*).

In the LIS3DSH accelerometer there are two different and independent finite State Machines, each one composed of 16 states. The two State Machines can be programmed

independently. An interrupt is generated when the End state is reached or when some specific command is performed.

**Figure 4. State Machine in LIS3DSH**




## 5.3 Signal block

Referring to *Figure 5*, while the measurement chain of LIS3DSH generates 16-bit wide data, State Machine inputs can be selected between:

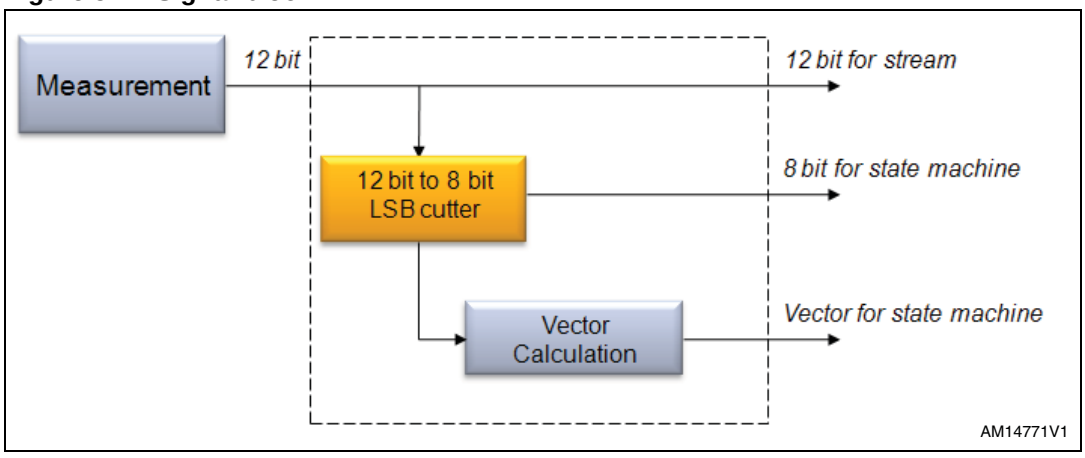
1. 8-bit wide acceleration data produced by LSB cutter.
2. 8-bit wide acceleration vector amplitude (V), calculated and filtered (if enabled).

### 5.3.1 LSB cutter

8-bit input data to State Machine are generated by dividing sensor output data by 256:

*8 bit data = 16 bit data* / 256.

**Figure 5. Signal block**




### 5.3.2 Vector calculation

Vector values (V) are in 8-bit format as well and their range is limited from -127 to +127.

Acceleration vector amplitude is only available inside the two State Machines, but cannot be read outside.

The vector value is calculated by means of an approximation formula:

Vrow = (45*a1+77*a2) / 256

where:

- X, Y, Z are axes 8-bit measured raw input values.

- a1 and a2 are temporary maximum 16-bit values, defined as follows:
  - a1 = abs(X)+abs(Y)+abs(Z)
  - a2 = max (abs(X), abs(Y), abs(Z))

- 45 and 77 are 8-bit fixed constants.

The calculated vector (Vrow) can be filtered by a 7th-order FIR filter.

### 5.3.3 Vector filter

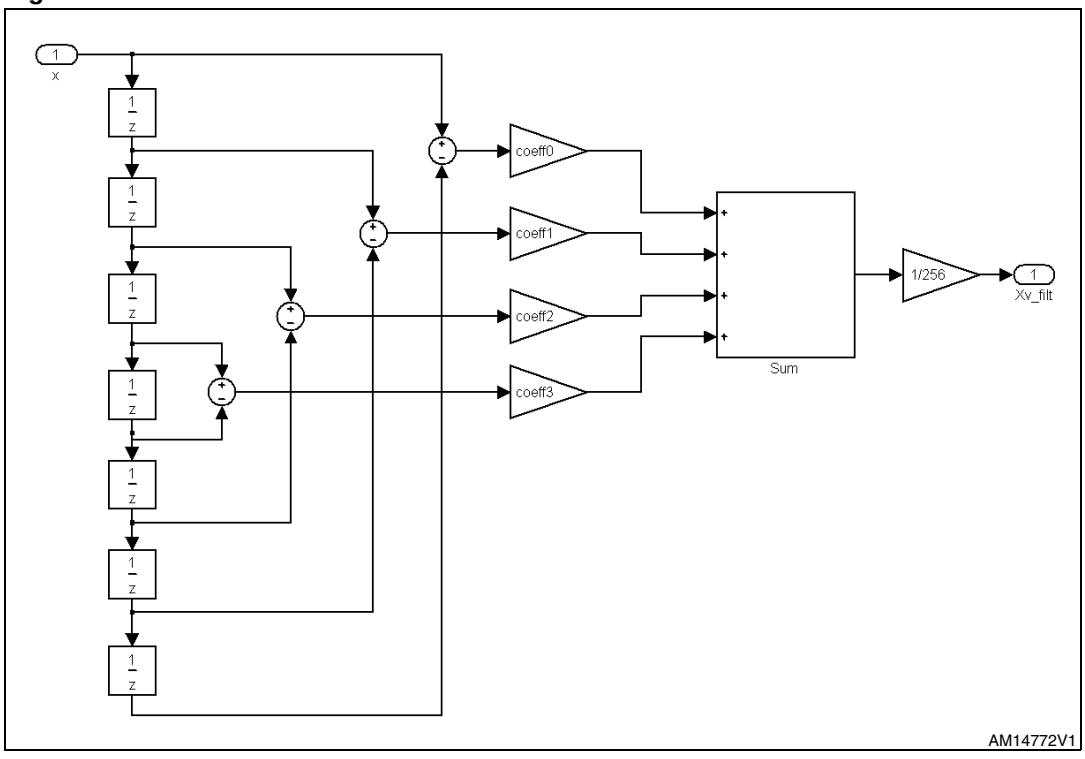
The vector filter is a 7th-order anti-symmetric FIR filter.

The transfer function of this filter is the following:

Xv_filt = (x0- x7) coeff0 + (x1-x6) coeff1+ (x2-x5) coeff2 + (x3-x4) coeff3.

*Figure 6.* shows the structure of the 7th-order anti-symmetric FIR filter.

**Figure 6. Vector filter**




The four coefficients can be chosen by using registers 1Bh, 1Ch, 1Dh and 1Eh (vector filter coefficient registers). In this way, different filter configurations can be implemented. For example, a band-pass filter can be obtained by choosing the coefficients: 53, 127, 127, 53.

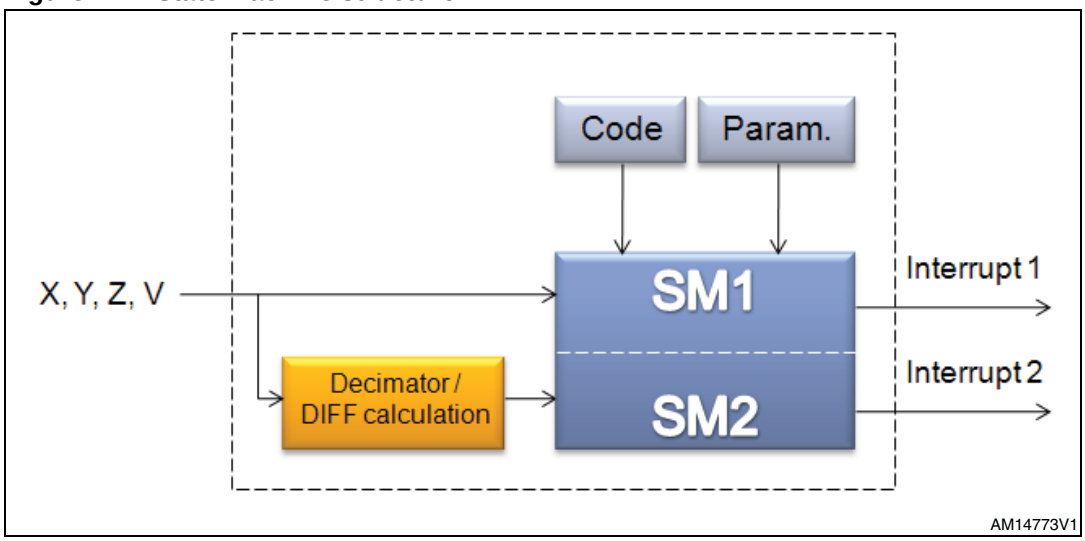
To enable the vector filter, the VFILT bit in the CTRL_REG3 register (23h) must be set to “1”.

## 5.4 State Machine blocks

Output data coming from the Signal Block are sent to the State Machine Block composed of the two State Machines. There are some differences in terms of functionality between the two State Machines:

1. State Machine 2 has decimator functionality, according to DES2 factor (DES2 register, 78h).
2. State Machine 2 has DIFF functionality/filter. The DIFF filter can be configured in two different ways:
   a) DIFF filtering with previous data values (X, Y, Z) as diff
   b) DIFF filtering with Constant Shift register values (X, Y, Z) as cs
3. When DIFF functionality is selected in State Machine 2, the vector value calculated (V) is left intact.

**Figure 7. State Machine structure**




### 5.4.1 Decimator

The decimation function is a method to reduce the sample rate of the data going to State Machine 2.

The decimation function is based on the initial value of the DES2 register (78h) and DCC (decimation counter register) according to the selected ODRx factor.

ODR_SM2 = ODR / (DES + 1)

At startup:

DCC = DES2 (initial decimation value)

when sample clock occurs:

DCC = DCC -1

When DCC is equal to 0, the current sample is used as new input for State Machine 2.

DCC = DES2 (initial decimation value)

### 5.4.2 DIFF calculation

This function is available only in State Machine 2. It is a data process method which calculates:

1. $diff_2$ - difference between current data (X, Y, Z) and previous data.
2. cs - difference between current data (X, Y, Z) and Constant Shift registers.

$diff_2$:

- Previous samples (X,Y,Z) selected for calculations
- Diff2 = *current measured - previous sample*

After calculation, the new samples are moved to the previous samples and the “old” previous samples are discarded.

cs:

Constant Shift acts like temporary offset shift. Constant Shift initial values (stored in registers 13h, 14h and 15h) are used to calculate DIFF results:

- cs = *current measured - constant shift*

To enable DIFF calculation on State Machine 2, bits RADI and D_CS must be set in the SETT2 register (7Bh).

## 5.5 State Machine description

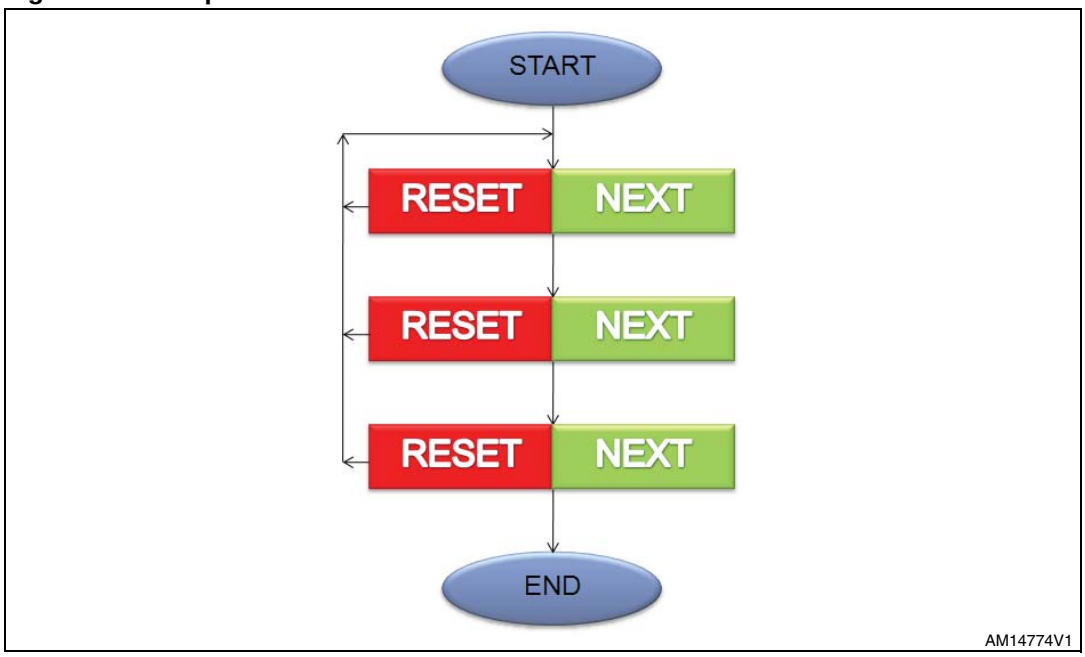
State Machine is a set of defined states, with inputs, outputs and transitions between states. In the LIS3DSH accelerometer, two State Machines are available. They can run either independently or synchronized, but always using the same input data.

For each new data sample, the State Machine 1 is performed first, then State Machine 2 with the same common internal input data.

Input data are not changed during State Machine executions. Calculation results are stored in temporary parameters.

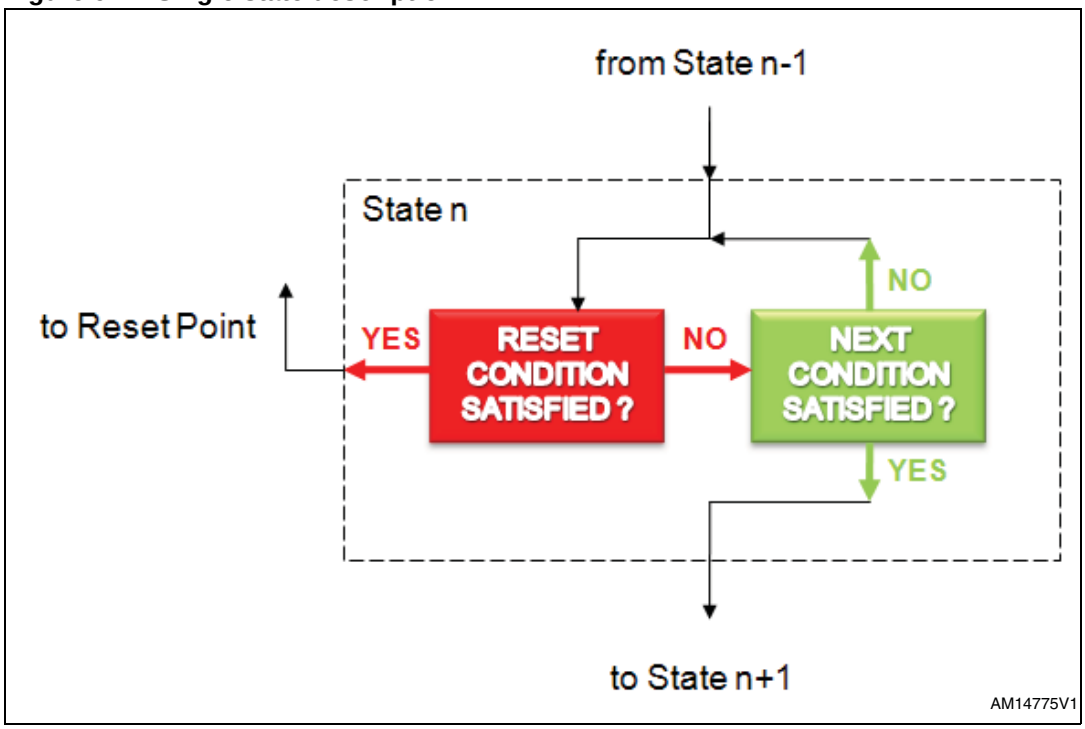
A simple State Machine is shown in *Figure 8*:

**Figure 8. Simple State Machine**




Each state includes NEXT/RESET conditions. The RESET condition is defined in the MSB part while the NEXT condition is defined in the LSB part of the ST1_X and ST2_X registers. As shown in *Figure 9*, the RESET condition is performed first, the NEXT condition is performed only when the RESET condition is not satisfied. When both conditions (NEXT and RESET) are not satisfied, the State Machine waits for a new sample and starts the evaluation again in the same state.

**Figure 9. Single state description**



Commands and their parameters are executed as one single step command.

From state (n) it is only possible to have a transition either to the next state (n+1), or to the state pointed by the Reset Point, or to continue in the same state (n).

Transition to the Reset Point happens whenever the “RESET condition” is true.

Transition to the next step happens whenever the “NEXT condition” is true and “RESET condition” is false.

An interrupt is generated whenever the End state is reached.

# 6 Operation codes

Operation (OP) codes can be divided into two groups: NEXT/RESET conditions and COMMANDS.

1. NEXT/RESET conditions are a combination of two conditions. All NEXT/RESET conditions can be applied at any state and their evaluation occurs when a new sample set (X, Y, Z, V) is generated. The two comparisons are executed in one single state; NEXT/RESET conditions belonging to the same state are synchronized to the sample clock.
2. COMMANDS have special tasks for flow control, output and synchronization. There are three types of commands, depending on execution timing:
   – Immediately executed: commands executed immediately as they appear;
   – Executed after trigger: wait for internal or external trigger to continue;
   – Special command (JMP command): conditional jump command.

The OP codes have a direct effect on registers and internal memories. For some OP codes, additional side-effects can occur (such as update of status information).

## 6.1 Next/Reset conditions

*Note: Character "y" in the text is used to refer to State Machine 1 or State Machine 2.*

**Table 83. Conditions**

| OPCODE | Mnemonic | Description | Note |
|---|---|---|---|
| 0h | NOP | No operation | Execution moves to another condition |
| 1h | TI1 | Timer 1 (16-bit value) valid | No evaluation of data samples |
| 2h | TI2 | Timer 2 (16-bit value) valid | No evaluation of data samples |
| 3h | TI3 | Timer 3 (8-bit value) valid | No evaluation of data samples |
| 4h | TI4 | Timer 4 (8-bit value) valid | No evaluation of data samples |
| 5h | GNTH1 | Any/triggered axis greater than THRS1 | First axis triggers |
| 6h | GNTH2 | Any/triggered axis greater than THRS2 | First axis triggers |
| 7h | LNTH1 | Any/triggered axis less than or equal to THRS1 | First axis triggers |
| 8h | LNTH2 | Any/triggered axis less than or equal to THRS2 | First axis triggers |
| 9h | GTTH1 | Any/triggered axis greater than THRS1 but using always standard Axis mask (MASK1) | First axis triggers |
| Ah | LLTH2 | All axis less than or equal to THRS2 | First masked axis triggers |
| Bh | GRTH1 | Any/triggered axis greater than reversed THRS1 | First axis triggers |

Table 83. Conditions (continued)

| OPCODE | Mnemonic | Description | Note |
|---|---|---|---|
| Ch | LRTH1 | Any/triggered axis less than or equal to reversed THRS1 | First axis triggers |
| Dh | GRTH2 | Any/triggered axis greater than reversed THRS2 | First axis triggers |
| Eh | LRTH2 | Any/triggered axis less than or equal to reversed THRS2 | First axis triggers |
| Fh | NZERO | Any axis zero crossed | First axis triggers |

### 6.1.1 NOP (0h)

NOP (no operation) is used as filler for the NEXT/RESET pair for some particular conditions which don't need an active opposite condition.

- If NOP is in RESET condition: do nothing and move to NEXT condition;
- If NOP is in NEXT condition: do nothing and stay in the same state re-evaluating the RESET condition when a new sample arrives;
- If NOP opcode in NEXT condition is not real: use case;
- No outputs.

### 6.1.2 TI1 (1h)

TI1 condition counts and evaluates the counter value of the TCy register.

If a new sample set (X, Y, Z, V) occurs, then TCy = TCy - 1:

- If TCy > 0 (counter is not full): continue comparisons in the current state (wait for new samples);
- If TCy == 0 (counter is full): Timer TI1 condition (NEXT or RESET) is valid.

This condition affects or is affected by the following registers:

- TIM1_y is 16 bits unsigned initial value;
- PRy: Program and Reset pointer addresses.

### 6.1.3 TI2 (2h)

TI2 condition counts and evaluates the counter value of the TCy register.

If a new sample set (X,Y,Z,V) occurs, then TCy = TCy - 1:

- If TCy > 0 (counter is not full): continue comparisons in the current state (wait for new samples);
- If TCy == 0 (counter is full): Timer TI2 condition (NEXT or RESET) is valid.

This condition affects or is affected by the following registers:

- TIM2_y is 16 bits unsigned initial value;
- PRy: Program and Reset pointer addresses.

### 6.1.4 TI3 (3h)

TI3 condition counts and evaluates the counter value of the TCy register.

If a new sample set (X, Y, Z, V) occurs, then TCy = TCy - 1:

- If TCy > 0 (counter is not full): continue comparisons in the current state (wait for new samples);
- If TCy == 0 (counter is full): Timer TI1 condition (NEXT or RESET) is valid.

This condition affects or is affected by the following registers:

- TIM3_y is 8 bits unsigned initial value;
- PRy: Program and Reset pointer addresses.

### 6.1.5 TI4 (4h)

TI4 condition counts and evaluates the counter value of the TCy register.

If a new sample set (X, Y, Z, V) occurs, then TCy = TCy - 1:

- If TCy > 0 (counter is not full): continue comparisons in the current state (wait for new samples);
- If TCy == 0 (counter is full): Timer TI1 condition (NEXT or RESET) is valid.

This condition affects or is affected by the following registers:

- TIM4_y is 8 bits unsigned initial value;
- PRy: Program and Reset pointer addresses.

### 6.1.6 GNTH1 (5h)

The GNTH1 condition is valid if any/triggered axis of the data sample set (X, Y, Z, V) is greater than threshold 1 level.

Threshold is: THRS1_y + Hysteresis.

Hysteresis is:

- State Machine 1: CTRL_REG1, bits HYST2_1, HYST1_1 and HYST0_1;
- State Machine 2: CTRL_REG2, bits HYST2_2, HYST1_2 and HYST0_2.

This condition affects or is affected by the following registers:

- THRS1_y: Threshold 1 value;
- MASKy_A and MASKy_B: Axis mask filter values;
- SETTy, bit ABS: Unsigned/signed settings;
- SETTy, bit R_TAM: Release temporary output mask settings;
- SETTy, bit P_DET: Peak detection settings;
- PEAKy: Peak output value;
- PRy: Program and Reset pointer addresses.

### 6.1.7 GNTH2 (6h)

The GNTH2 condition is valid if any/triggered axis of the data sample set (X, Y, Z, V) is greater than threshold 2 level.

Threshold is: THRS2_y + Hysteresis.

Hysteresis is:

- State Machine 1: CTRL_REG1, bits HYST2_1, HYST1_1 and HYST0_1;
- State Machine 2: CTRL_REG2, bits HYST2_2, HYST1_2 and HYST0_2.

This condition affects or is affected by the following registers:

- THRS2_y: Threshold 2 value;
- MASKy_A and MASKy_B: Axis mask filter values;
- SETTy, bit ABS: Unsigned/signed settings;
- SETTy, bit R_TAM: Release temporary output mask settings;
- SETTy, bit P_DET: Peak detection settings;
- PEAKy: Peak output value;
- PRy: Program and Reset pointer addresses.

### 6.1.8 LNTH1 (7h)

The LNTH1 condition is valid if any/triggered axis of data sample set (X, Y, Z, V) is less than or equal to threshold 1 level.

Threshold is: THRS1_y - Hysteresis.

Hysteresis is:

- State Machine 1: CTRL_REG1, bits HYST2_1, HYST1_1 and HYST0_1;
- State Machine 2: CTRL_REG2, bits HYST2_2, HYST1_2 and HYST0_2.

This condition affects or is affected by the following registers:

- THRS1_y: Threshold 1 value;
- MASKy_A and MASKy_B: Axis mask filter values;
- SETTy, bit ABS: Unsigned/signed settings;
- SETTy, bit R_TAM: Release temporary output mask settings;
- SETTy, bit P_DET: Peak detection settings;
- PEAKy: Peak output value;
- PRy: Program and Reset pointer addresses.

### 6.1.9 LNTH2 (8h)

The LNTH2 condition is valid if any/triggered axis of data sample set (X, Y, Z, V) is less than or equal to threshold 2 level.

Threshold is: THRS2_y - Hysteresis.

Hysteresis is:

- State Machine 2: CTRL_REG1, bits HYST2_1, HYST1_1 and HYST0_1;
- State Machine 3: CTRL_REG2, bits HYST2_2, HYST1_2 and HYST0_2.

This condition affects or is affected by the following registers:

- THRS2_y: Threshold 2 value;
- MASKy_A and MASKy_B: Axis mask filter values;
- SETTy, bit ABS: Unsigned/signed settings;
- SETTy, bit R_TAM: Release temporary output mask settings;
- SETTy, bit P_DET: Peak detection settings;
- PEAKy: Peak output value;
- PRy: Program and Reset pointer addresses.

### 6.1.10 GTTH1 (9h)

The GTTH1 condition is valid if any/triggered axis of data sample set (X, Y, Z, V) is greater than threshold 1 level. The GTTH1 condition always evaluates the standard Axis mask (MASK1).

Threshold is: THRS1_y + Hysteresis.

Hysteresis is:

- State Machine 2: CTRL_REG1, bits HYST2_1, HYST1_1 and HYST0_1;
- State Machine 3: CTRL_REG2, bits HYST2_2, HYST1_2 and HYST0_2.

This condition affects or is affected by the following registers:

- THRS1_y: Threshold 1 value;
- MASK1_A and MASK1_B: Axis mask filter values;
- SETT1, bit ABS: Unsigned/signed settings;
- SETT1, bit P_DET: Peak detection settings;
- PEAK1: Peak output value;
- PR1: Program and Reset pointer addresses;
- Note: R_TAM bit in SETT1 register does not affect this condition, since the standard mask is always evaluated.

### 6.1.11 LLTH2 (Ah)

The LLTH2 condition is valid if all axes in data sample set (X, Y, Z, V) are less than or equal to threshold 2 level.

Threshold is: THRS2_y - Hysteresis.

Hysteresis is:

- State Machine 1: CTRL_REG1, bits HYST2_1, HYST1_1 and HYST0_1;
- State Machine 2: CTRL_REG2, bits HYST2_2, HYST1_2 and HYST0_2.

This condition affects or is affected by the following registers:

- THRS2_y: Threshold 2 value;
- MASKy_A and MASKy_B: Axis mask filter values;
- SETTy, bit ABS: Unsigned/signed settings;
- SETTy, bit R_TAM: Release temporary output mask settings;
- SETTy, bit P_DET: Peak detection settings;
- PEAKy: Peak output value;
- PRy: Program and Reset pointer addresses.

### 6.1.12 GRTH1 (Bh)

The GRTH1 condition is valid if any/triggered axis of data sample set (X, Y, Z, V) is greater than reversed threshold 1 level.

Threshold is: THRS1_y + Hysteresis.

Hysteresis is:

- State Machine 1: CTRL_REG1, bits HYST2_1, HYST1_1 and HYST0_1;
- State Machine 2: CTRL_REG2, bits HYST2_2, HYST1_2 and HYST0_2;

This condition affects or is affected by the following registers:

- THRS1_y: Threshold 1 value;
- MASKy_A and MASKy_B: Axis mask filter values;
- SETTy, bit ABS: Unsigned/signed settings;
- SETTy, bit R_TAM: Release temporary output mask settings;
- SETTy, bit P_DET: Peak detection settings;
- PEAKy: Peak output value;
- PRy: Program and Reset pointer addresses.

### 6.1.13 LRTH1 (Ch)

The LRTH1 condition is valid if any/triggered axis of data sample set (X, Y, Z, V) is less than or equal to reversed threshold 1 level.

Threshold is: THRS1_y - Hysteresis.

Hysteresis is:

- State Machine 1: CTRL_REG1, bits HYST2_1, HYST1_1 and HYST0_1;
- State Machine 2: CTRL_REG2, bits HYST2_2, HYST1_2 and HYST0_2.

This condition affects or is affected by the following registers:

- THRS1_y: Threshold 1 value;
- MASKy_A and MASKy_B: Axis mask filter values;
- SETTy, bit ABS: Unsigned/signed settings;
- SETTy, bit R_TAM: Release temporary output mask settings;
- SETTy, bit P_DET: Peak detection settings;
- PEAKy: Peak output value;
- PRy: Program and Reset pointer addresses.

### 6.1.14 GRTH2 (Dh)

The GRTH2 condition is valid if any/triggered axis of data sample set (X, Y, Z, V) is greater than revered threshold 2 level.

Threshold is: THRS2_y + Hysteresis.

Hysteresis is:

- State Machine 1: CTRL_REG1, bits HYST2_1, HYST1_1 and HYST0_1;
- State Machine 2: CTRL_REG2, bits HYST2_2, HYST1_2 and HYST0_2.

This condition affects or is affected by the following registers:

- THRS2_y: Threshold 2 value;
- MASKy_A and MASKy_B: Axis mask filter values;
- SETTy, bit ABS: Unsigned/signed settings;
- SETTy, bit R_TAM: Release temporary output mask settings;
- SETTy, bit P_DET: Peak detection settings;
- PEAKy: Peak output value;
- PRy: Program and Reset pointer addresses.

### 6.1.15 LRTH2 (Eh)

The LRTH2 condition is valid if any/triggered axis of data sample set (X, Y, Z, V) is less than or equal to reversed threshold 2 level.

Threshold is: THRS2_y - Hysteresis.

Hysteresis is:

- State Machine 1: CTRL_REG1, bits HYST2_1, HYST1_1 and HYST0_1;
- State Machine 2: CTRL_REG2, bits HYST2_2, HYST1_2 and HYST0_2.

This condition affects or is affected by the following registers:

- THRS2_y: Threshold 2 value;
- MASKy_A and MASKy_B: Axis mask filter values;
- SETTy, bit ABS: Unsigned/signed settings;
- SETTy, bit R_TAM: Release temporary output mask settings;
- SETTy, bit P_DET: Peak detection settings;
- PEAKy: Peak output value;
- PRy: Program and Reset pointer addresses.

### 6.1.16 NZERO (Fh)

The NZERO condition is valid if any axis of data sample set (X, Y, Z, V) changes the sign. No Hysteresis is considered for this condition.

This condition affects or is affected by the following registers:

- MASKy_A and MASKy_B: Axis mask filter values;
- SETTy, bit ABS: Unsigned/signed settings;
- SETTy, bit R_TAM: Release temporary output mask settings;
- SETTy, bit P_DET: Peak detection settings;
- PEAKy: Peak output value;
- PRy: Program and Reset pointer addresses.

## 6.2 Commands

COMMANDS have special tasks for flow control, output and synchronization. There are three types of commands, depending on execution timing:

- Immediately executed: commands executed without waiting for a new sample;
- Executed after trigger: wait for an internal or external trigger to proceed. The internal trigger may be: wait for the new sample. The external trigger may be: wait for reading of the OUTS register;
- Special commands (JMP commands): special conditions comparison for conditional jump commands.

*Note: Character “y” in the text refers to State Machine 1 or State Machine 2.*

**Table 84. Commands (main set)**

| OPCODE | Mnemonic | Description | Note |
|---|---|---|---|
| 00h | STOP | Stops execution, and resets RESET-POINT to start | All other registers and internal memories are left intact |
| 11h | CONT | Continues execution from RESET-POINT | |
| 22h | JMP | Conditional jump. It includes:<br>- two conditions (1st parameter)<br>- two addresses for valid conditions (2nd parameter) | |
| 33h | SRP | Sets RESET-POINT to next step address | |
| 44h | CRP | Sets RESET-POINT to address 0 (start position) | |
| 55h | SETP | Sets parameter in register memory. It includes:<br>- register address (1st parameter)<br>- new parameter to be set (2nd parameter) | Address parameter is direct absolute pointer to register memory |
| 66h | SETS1 | Sets new value (1st parameter) to SETTy register | |
| 77h | STHR1 | Sets new value (1st parameter) to THRS1y register | |
| 88h | OUTC | Sets outputs to output registers | |
| 99h | OUTW | Sets outputs to output registers and waits for host actions for output latch release | Host driven event |
| AAh | STHR2 | Sets new value (1st parameter) to THRS2y register | |
| BBh | DEC | Decreases long counter (LC) value and validate counter | |
| CCh | SISW | Swaps temporary axis mask sign to opposite sign | |
| DDh | REL | Releases temporary axis mask information | |
| EEh | STHR3 | Sets new value (1st parameter) to THRS3y register | |
| FFh | SSYNC | Toggles execution from one State Machine to the other one | Affects both State Machines.<br>Immediate execution and Wait (sync) |

**Table 85. Commands (extended set)**

| OPCODE | Mnemonic | Description | Note |
|---|---|---|---|
| 12h | SABS0 | Sets ABS=0 in SETTy register (unsigned thrs) | |
| 13h | SABS1 | Sets ABS=1 in SETTy register (signed thrs) | |
| 14h | SELMA | Sets mask pointer to MASKy_A | |
| 21h | SRADI0 | Sets RADI=0 in SETT2 register (raw data mode) | Only for State Machine 2 |
| 23h | SRADI1 | Sets RADI=1 in SETT2 register (difference data mode) | Only for State Machine 2 |
| 24h | SELSA | Sets mask pointer to MASKy_B | |
| 31h | SCS0 | Sets D_CS=0 in SETT2 register (DIFF data form ON for State Machine 2) | Only for State Machine 2. |
| 32h | SCS1 | Sets D_CS=1 in SETT2 register (Constant Shift data form ON for State Machine 2) | Only for State Machine 2 |
| 34h | SRTAM0 | Sets R_TAM=0 in SETTy register (Temporary Axis mask is kept intact) | |
| 41h | STIM3 | Sets a new value (1st parameter) to TIM3y register | |
| 42h | STIM4 | Sets a new value (1st parameter) to TIM4y register | |
| 43h | SRTAM1 | Sets R_TAM=1 in SETTy register (Temporary Axis mask is released to default after every valid NEXT condition) | |

**Table 86. Forbidden OP codes**

| OPCODE | Note |
|---|---|
| 21h | Forbidden in State Machine 1. When it exists in State Machine 1, it immediately stops |
| 23h | Forbidden in State Machine 1. When it exists in State Machine 1, it immediately stops |
| 31h | Forbidden in State Machine 1. When it exists in State Machine 1, it immediately stops |
| 32h | Forbidden in State Machine 1. When it exists in State Machine 1, it immediately stops |

### 6.2.1 STOP (00h)

The STOP command halts execution and waits for host restart.

This command has no parameters and it is an “Immediately executed” type.

Actions:

1. STOP command halts State Machine y execution:
   – CTRL_REG1, bit SM1_EN is set to 0 for State Machine 1;
   – Or CTRL_REG2, bit SM2_EN is set to 0 for State Machine 2;
   – Set STAT, bit SYNC1 = 0 and STAT, bit SYNC2 = 0 (synchronization is not allowed).
2. If SETTy, bit SITR = 1:
   – OUTSy is updated to selected temporary mask value;
   – Set output signal: STAT, bit INT_SMy = 1.
3. Wait for restart from host. When restart occurs:
   – CTRL_REG1, bit SM1_EN is set to 0 for State Machine 1;
   – Or CTRL_REG2, bit SM2_EN is set to 0 for State Machine 2.

This command affects or is affected by the following registers:

– State Machine y is enabled/disabled: CTRL_REG1, bit SM1_EN is set to 0/1 for State Machine 1 or CTRL_REG2, bit SM2_EN is set to 0/1 for State Machine 2;
– SETTy, bit SITR: defines output functionality of STOP command;
– OUTSy: output value of State Machine y;
– STAT, bit INT_SMy: indicator of valid interrupt action;
– PRy: Program and Reset pointer addresses.

### 6.2.2 CONT (11h)

The CONT command loops execution to the beginning.

This command has no parameters and it is an “Immediately executed” type.

Actions:

1. If SETTy, bit SITR = 1:
   – OUTSy is updated to selected temporary mask value;
   – Set output register (and signal if selected): STAT, bit INT_SMy = 1.
2. Default initial start executed
3. Continue execution from step address PPy = 0

This command affects or is affected by the following registers:

– State Machine y is enabled/disabled: CTRL_REG1, bit SM1_EN is set to 0/1 for State Machine 1. CTRL_REG2, bit SM2_EN is set to 0/1 for State Machine 2;
– SETTy, bit SITR: Defines output functionality of STOP command;
– OUTSy: Output value of State Machine y;
– STAT, bit INT_SMy: Indicator of valid interrupt action;
– PRy: Program and Reset pointer addresses.

### 6.2.3 JMP (22h)

JMP is the conditional jump command. It has two conditions (in the 1st parameter), these two conditions refer to different jump addresses (available in the 2nd parameter):

– 1st parameter: COND1 (4 bits - MSB part); condition COND2 (4 bits - LSB part);
– 2nd parameter: ADD1 (4 bits - MSB part), address ADD2 (4 bits - LSB part).

The two conditions are evaluated as two NEXT conditions in state.

Conditions inside JMP command:

- are using only the selected MASKy_A / MASKy_B axis mask;
- do not affect temporary axis mask pointer;
- have no peak detection.

The two conditions can be any condition described in this document.

The first step and second step are executed immediately.

Rules for condition validation:

- COND1 is validated first with new data sample set;
- If COND1 is not valid, COND2 is validated next;
- If COND2 is not valid, wait for a new sample set and restart conditions validation;
- If COND1 is valid, jump to ADD1;
- If COND2 is valid, jump to ADD2.

This command affects or is affected by the following registers:

- THRS1_y, THRS2_y: Thresholds limit value;
- TIM1_y, TIM2_y, TIM3_y, TIM4_y: Timers initial value;
- MASKy_A and MASKy_B: Axis mask filter values;
- SETTy, bit ABS: Unsigned/signed settings;
- SETTy, bit T_SIGN: Temporary output mask filter settings;
- PRy: Program and Reset pointer addresses.

### 6.2.4 SRP (33h)

The SRP command sets the Reset Point to the next address/state.

This command has no parameters and it is an “Immediately executed” type.

Actions:

SRP command sets the Reset Point (RPy) to the next step address: RPy = PPy + 1.

This command affects or is affected by the following registers:

- PRy: Program and Reset pointer addresses.

### 6.2.5 CRP (44h)

The CRP command clears Reset Point to the start position (at the beginning of the program code). This command has no parameters and it is an “Immediately executed” type.

Actions:

CRP command sets Reset Point (RPy) to the beginning of program code: RPy = 0.

This command affects or is affected by the following registers:

- PRy: Program and Reset pointer addresses.

### 6.2.6 SETP (55h)

The SETP command allows the configuration of the State Machine currently used to be modified.

It sets a register address (1st parameter) belonging to the current State Machine area to a new value (8 bits - 2nd parameter).

This command is an “Immediately executed” type and it takes two parameters:

- 1st parameter: Address (8 bits);
- 2nd parameter: New value (8 bits) to write in the address specified by the first parameter.

Actions:

1. SETP command sets one byte (2nd parameter) to any register address (1st parameter):
   - Address must be a “Write” or “Read-Write” type register;
   - Program pointer is increased by 3 units: PPy = PPy +3.

This command affects or is affected by the following registers:

- PRy: Program and Reset pointer addresses;
- Any register address indicated by the first parameter.

### 6.2.7 SETS1 (66h)

The SETS1 command sets the content of the SETTy register (on the current State Machine) to a new value (1st parameter).

This command is an “Immediately executed” type and it takes one parameter:

- The new value to be set in SETTy register (8 bits).

Actions:

1. SETS1 command sets the value of the SETTy register (on current State Machine) to a new value (8 bits - 1st parameter):
   - The new value of the SETTy register is valid when the next step starts;
   - Program pointer is increased by 2 units: PPy = PPy +2.

This command affects or is affected by the following registers:

- PRy: Program and Reset pointer addresses;
- SETTy: Program flow control register for State Program y.

### 6.2.8 STHR1 (77h)

The STHR1 command sets the threshold1 register to a new value (1st parameter).

This commands is an “Immediately executed” type and it takes one parameter:

- The new value to be set in the THRS1_y register (8 bits).

Actions:

1. STHR1 command sets the value of the THRS1_y register (on the current State Machine) to a new value (8 bits - 1st parameter):
   - The new value of the THRS1_y register is valid when the next step starts;
   - Program pointer is increased by 2 units: PPy = PPy +2.

This command affects or is affected by the following registers:

- PRy: Program and Reset pointer addresses;
- THRS1_y: Threshold 1 limit value.

### 6.2.9 OUTC (88h)

OUTC stands for Output Command. This command sets the outputs to the output registers.

The OUTC command has no parameters and it is an “Immediately executed” type.

An interrupt is triggered when the OUTC command is performed.

Actions:

1. OUTSy is updated to the selected temporary mask value.
2. Set output signal: STAT, bit INT_SMy = 1.
3. If SETTy, bit P_DET = 1: PEAKy = 0.

This command affects or is affected by the following registers:

- MASKy_A and MASKy_B: Axis mask filter values;
- OUTSy: Output value of State Machine y;
- STAT, bit INT_SMy: Interrupt indicator of State Program y;
- SETTy, bit P_DET: Peak detection settings;
- PEAKy: Peak output value;
- PRy: Program and Reset pointer addresses.

### 6.2.10 OUTW (99h)

OUTW stands for “output command and acknowledge from host”.

This command performs output actions (like the OUTC command) generating an interrupt. After that, the command waits for a host action before continuing the State Machine execution.

The host action is the reading of register OUTSy (5Fh / 7Fh).

This command has no parameters and it is an “Executed after trigger” type.

Actions;

1. OUTSy is updated to selected temporary mask value.
2. Set output signals:
   - STAT, bit INT_SMy = 1;
   - STAT, bit SYNCW = 1.
3. Stop and wait\;
4. Waits for reading of OUTSy register for release of interrupt information:
   - If OUTSy > 0 then wait for the releasing of the OUTSy register (State Machine y waits for host actions).
5. If OUTSy released, OUTSy == 0:
   - STAT, bit SYNCW = 0;
   - STAT, bit INT_SMy = 0.
6. If SETTy, bit P_DET = 1: PEAKy = 0.
7. Continue State Machine y execution.

This command affects or is affected by the following registers:

- MASKy_A and MASKy_B: Axis mask filter values
- OUTSy: Output value of State Machine y
- STAT, bit INT_SMy: Interrupt indicator of State Machine y
- PRy: Program and Reset pointer addresses.

### 6.2.11 STHR2 (AAh)

The STHR2 command sets the threshold1 register to a new value (1st parameter).

This command is an “Immediately executed” type and it takes one parameter:

- new value to be set in the THRS2_y register (8 bits).

Actions:

1. STHR2 command sets the value of the THRS2_y register (on current State Machine) to one new value (8 bits - 1st parameter):
   - New value of THRS2_y register is valid when next step starts;
   - Program pointer is increased by 2 units: PPy = PPy +2.

This command affects or is affected by the following registers:

- PRy: Program and Reset pointer addresses;
- THRS2_y: Threshold 2 limit value.

### 6.2.12 DEC (BBh)

The DEC command decreases the long counter (LC) value and evaluates the result.

This command has no parameters and it is an “Immediately executed” type.

Actions:

1. If LC > 0: LC = LC – 1.
2. If LC == 0, long counter value is valid:
   - STAT, bit LONG = 1;
   - OUTSy is updated to selected temporary mask value;
   - Set output register (and signal if selected): STAT, bit INT_SMy = 1;
   - LC = -1 (inactive long counter).

This command affects or is affected by the following registers:

- PRy: Program and Reset pointer addresses;
- LC: Long counter register;
- STAT, bit LONG: Indicator flag of valid long counter;
- STAT, bit INT_SMy: Interrupt indicator of State Machine y.

### 6.2.13 SISW (CCh)

The SISW command swaps the temporary axis mask sign to the opposite sign.

This command has no parameters and it is an “Immediately executed” type.

Actions:

1. Change selected temporary mask one axis sign to the opposite:
   – If sign(axis) is positive, new sign(axis) is negative;
   – If sign(axis) is negative, new sign(axis) is positive;
   – If axis information is zero, no changes.

This command affects or is affected by the following registers:
   – PRy: Program and Reset pointer addresses.

### 6.2.14 REL (DDh)

The REL command releases the temporary axis mask information.

This command has no parameters and it is an “Immediately executed” type.

Actions:
1. Reset temporary masks to default value.
2. If SETTy, bit P_DET == 1:
   – PEAKy = 0;
   – Reset peak detection to initial phase.

This command affects or is affected by the following registers:
   – PRy: Program and Reset pointer addresses;
   – SETTy, bit P_DET: Peak detection settings;
   – PEAKy: Peak output value.

### 6.2.15 STHR3 (EEh)

The STHR3 command sets the threshold 3 register to a new value (1st parameter).

This command is an “Immediately executed” type and it takes one parameter:
   – New value to be set in THRS3_y register (8 bits).

Actions:
1. The STHR3 command sets the value of the THRS3_y register to a new value (8 bits - 1st parameter):
   – The new value of THRS3_y register is valid when the next step starts;
   – Program pointer is increased by 2 units: PPy = PPy +2.

This command affects or is affected by the following registers:
   – PRy: Program and Reset pointer addresses;
   – THRS3_y: Threshold 3 limit value.

### 6.2.16 SSYNC (FFh)

The SYNC command switches execution from one State Machine to the other one. This command takes effect only when both State Machine 1 and State Machine 2 are enabled (through SM1_EN and SM2_EN bits in CTRL_REG1 and CTRL_REG2 registers). Execution waits for halt release from the other State Machine.

The SYNC command can be used in two ways:

1. Combining State Machine 1 and State Machine 2 as one single State Machine with a maximum of 32 states (as shown in *Figure 10.*);
2. Using State Machine 2 as a sub-routine of State Machine 1, which can be executed multiple times (as shown in *Figure 11*).

The SYNC command has no parameters and it is an “Executed after trigger” type.

**Figure 10. SSYNC - SM1+SM2 for 32 states SM**

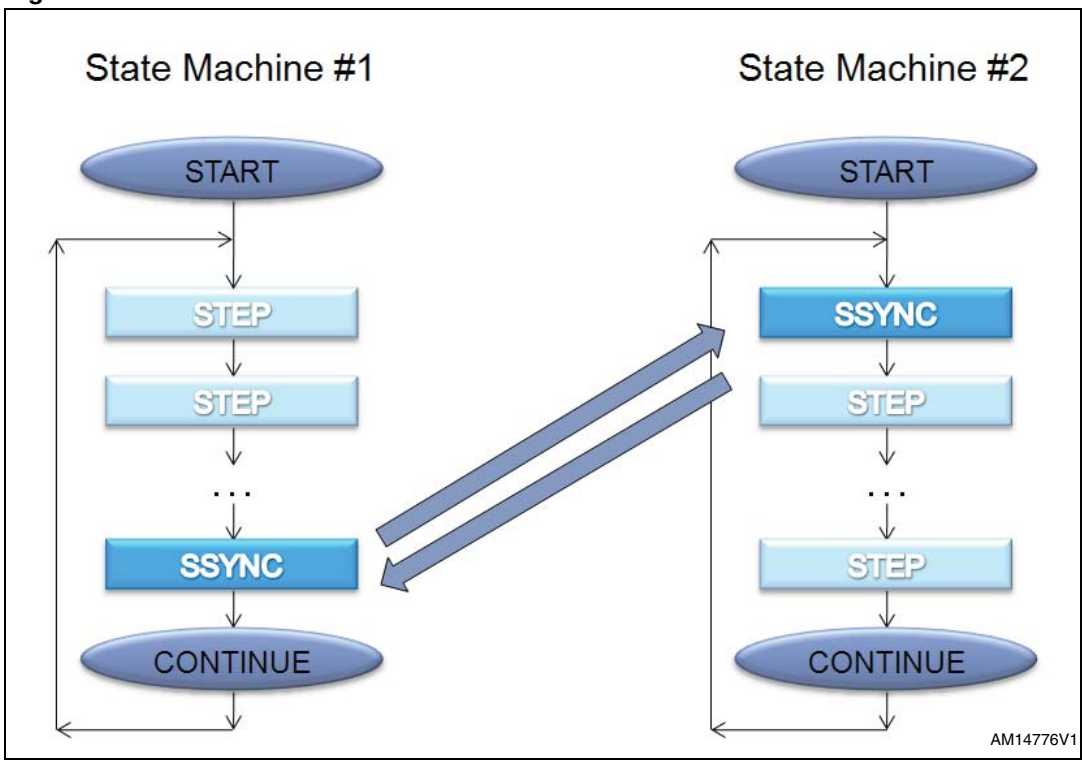
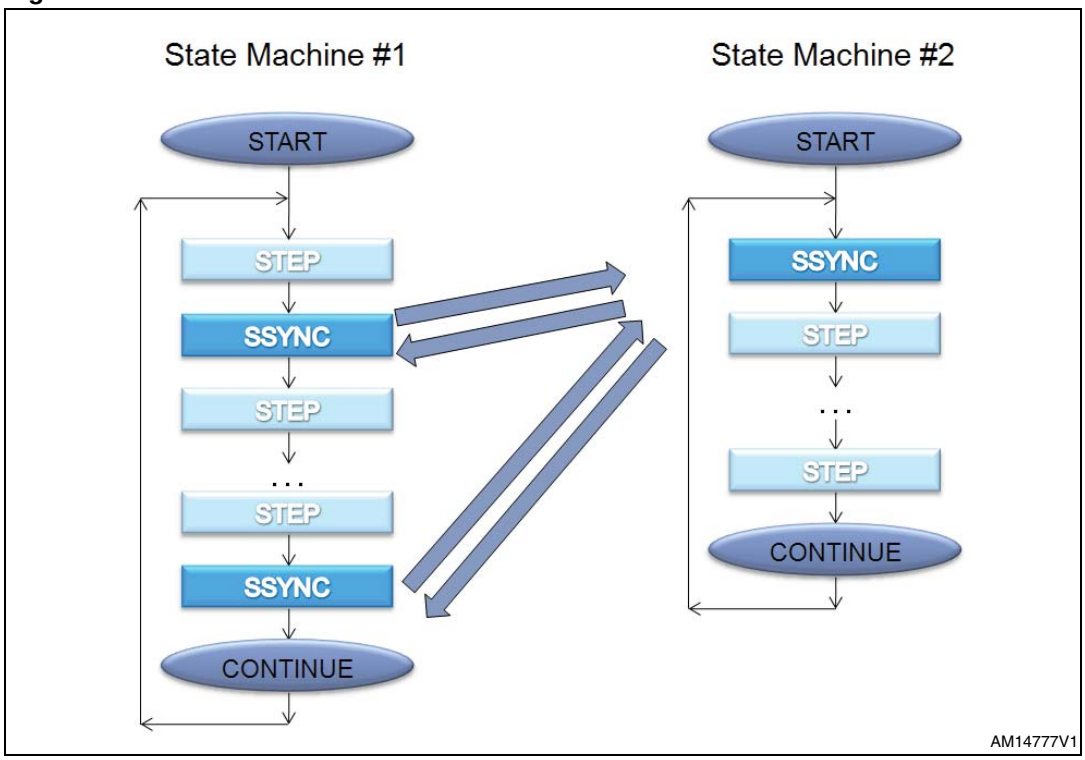


**Figure 11. SSYNC - SM2 used as subroutine of SM1**




Actions:

1. When both State Machine 1 and State Machine 2 are enabled:
   - If State Machine 1 is executed, then STAT, bit SYNC1 = 1 and STAT, bit SYNC2 = 0: State Machine 1 is stopped and it waits for synchronization release/restart from State Machine 2;
   - If State Machine 2 is executed, then STAT, bit SYNC2 = 1 and STAT, bit SYNC1 = 0: State Machine 2 is stopped and it waits for synchronization release/restart from State Machine 1.

This command affects or is affected by the following registers:

- PRy: Program and Reset pointer addresses;
- STAT, bit SYNC1: Sync flag for State Machine 1;
- STAT, bit SYNC2: Sync flag for State Machine 2.

### 6.2.17 SABS0 (12h)

The SABS0 command sets ABS setting to unsigned.

This command has no parameters and it is an “Immediately executed” type.

Actions:

1. The SABS0 command sets register SETTy, bit ABS to 0:
   - Sign filter is not sign dependent;
   - The new value of the SETTy register is valid starting from the next step.

This command affects or is affected by the following registers:

- PRy: Program and Reset pointer addresses;
- SETTy, bit ABS: Unsigned/signed settings.

### 6.2.18 SABS1 (13h)

The SABS1 command sets ABS setting to signed.

This command has no parameters and it is an “Immediately executed” type.

Actions:

1. SABS1 command sets register SETTy, bit ABS to 1:
   - Sign filter is sign dependent;
   - The new value of the SETTy register is valid starting from the next step.

This command affects or is affected by the following registers:

- PRy: Program and Reset pointer addresses;
- SETTy, bit ABS: Unsigned/signed settings.

### 6.2.19 SELMA (14h)

The SELMA command sets the axis mask pointer to MASKy_A.

This command has no parameters and it is an “Immediately executed” type.

Actions:

1. MASKy_A is selected.
2. Reset peak detection to initial phase.

This command affects or is affected by the following registers:

- PRy: Program and Reset pointer addresses;
- MASKy_A and MASKy_B: Axis mask filter values.

### 6.2.20 SRADI0 (21h)

The SRADI0 command disables the difference mode on input data for State Machine 2 (raw data mode). The SRADI0 command is effective only for State Machine 2.

This command has no parameters and it is an “Immediately executed” type.

Actions:

1. SRADI0 command sets register SETT2, bit RADI to 0:
   - Raw data mode is selected for State Machine 2;
   - The new value of the SETT2 register is valid starting from the next step;
   - Program pointer 2 is increased by 1 unit: PP2 = PP2 +1.

This command affects or is affected by the following registers:

- PR2: Program and Reset pointer addresses for State Machine 2;
- SETT2, bit RADI: RAW data / DIFF data input type selector for State Machine 2.

### 6.2.21 SRADI1 (23h)

The SRADI1 command enables the difference mode on input data for State Machine 2. The SRADI1 command is effective only for State Machine 2.

This command has no parameters and it is an “Immediately executed” type.

Actions:

1. The SRADI0 command sets to 1 the bit RADI, in register SETT2:
   - DIFF mode is selected for State Machine 2;
   - The new value of the SETT2 register is valid starting from the next step;
   - Program pointer 2 is increased by 1 unit: PP2 = PP2 +1.

This command affects or is affected by the following registers:

- PR2: Program and Reset pointer addresses for State Machine 2;
- SETT2, bit RADI: RAW data / DIFF data input type selector for State Machine 2.

### 6.2.22 SELSA (24h)

The SELSA command sets the axis mask pointer to MASKy_B.

This command has no parameters and it is an “Immediately executed” type.

Actions:

1. MASKy_B is selected.
2. Reset peak detection to initial phase.

This command affects or is affected by the following registers:

- PRy: Program and Reset pointer addresses;
- MASKy_A and MASKy_B: Axis mask filter values.

### 6.2.23 SCS0 (31h)

The SCS0 command sets the DIFF2 difference mode for State Machine 2. After this command, input data is the difference between current data and the previous data.

The SCS0 command is effective only for State Machine 2.

This command has no parameters and it is an “Immediately executed” type.

Actions:

1. SCS0 command sets to 0 the bit D_CS, in register SETT2:
   - DIFF calculated data input type is selected for State Machine 2;
   - The new value of the SETT2 register is valid starting from the next step.

This command affects or is affected by the following registers:

- PR2: Program and Reset pointer addresses for State Machine 2;
- SETT2, bit RADI: RAW data / DIFF data input type selector for State Machine 2.

### 6.2.24 SCS1 (32h)

The SCS1 command sets the Constant Shift difference mode for State Machine 2: input data is the difference between the current data and the value contained in Constant Shift registers. SCS1 command is effective only for State Machine 2.

This command has no parameters and it is an “Immediately executed” type.

Actions:

1. SCS1 command sets register SETT2, bit D_CS to 1:
   – Constant Shift calculated data input type is selected for State Machine 2;
   – The new value of the SETT2 register is valid starting from the next step.

This command affects or is affected by the following registers:

– PR2: Program and Reset pointer addresses for State Machine 2;
– SETT2, bit RADI: RAW data / DIFF data input type selector for State Machine 2;
– CS_X / CS_Y / CS_Z: Constant Shift registers.

### 6.2.25 SRTAM0 (34h)

The SRTAM0 command configures the R_TAM bit to preserve temporary axis mask.

This command has no parameters and it is an “Immediately executed” type.

Actions:

1. SRTAM0 command sets register SETTy, bit RTAM to 0:
   – Temporary axis mask value does not change after valid NEXT condition;
   – The new value of the SETTy register is valid starting from the next step.

This command affects or is affected by the following registers:

– PRy: Program and Reset pointer addresses;
– SETTy, bit R_TAM: Temporary axis mask and peak state flag release.

### 6.2.26 STIM3 (41h)

The STIM3 command sets Timer 3 belonging to the current State Machine to a new value (8 bits - 1st parameter).

This command is an “Immediately executed” type and it takes one parameter:

– new value to be set in the TIM3_y register (8 bits).

Actions:

1. The STIM3 command replaces current value of the TIM3_y register address to one new byte (1st parameter) on the current State Machine:
   – The new value of the TIM3_y register is valid starting from the next step;
   – Program pointer is increased by 2 units: PPy = PPy +2.

This command affects or is affected by the following registers:

– PRy: Program and Reset pointer addresses;
– TIM3_y: Timer 3 limit value.

### 6.2.27 STIM4 (42h)

The STIM4 command sets Timer 4 of the current State Machine to a new value (8 bits - 1st parameter).

This command is an “Immediately executed” type and it takes one parameter:

– New value to be set in the TIM4_y register (8 bits).

Actions:

1. The STIM4 command replaces the current value of the TIM4_y register address to a new byte (1st parameter) on the current State Machine:
   - The new value of the TIM4_y register is valid starting from the next step;
   - Program pointer is increased by 2 units: PPy = PPy +2.

This command affects or is affected by the following registers:

- PRy: Program and Reset pointer addresses;
- TIM4_y: Timer 4 limit value.

### 6.2.28 SRTAM1 (43h)

The SRTAM1 command configures the R_TAM bit to release the temporary axis mask after every valid NEXT condition.

This command has no parameters and it is an “Immediately executed” type.

Actions:

1. SRTAM1 command sets register SETTy, bit RTAM to 1:
   - The temporary axis mask value is set to default after every valid NEXT condition;
   - The new value of the SETTy register is valid starting from the next step.

This command affects or is affected by the following registers:

- PRy: Program and Reset pointer addresses;
- SETTy, bit R_TAM: Temporary axis mask and peak state flag release.

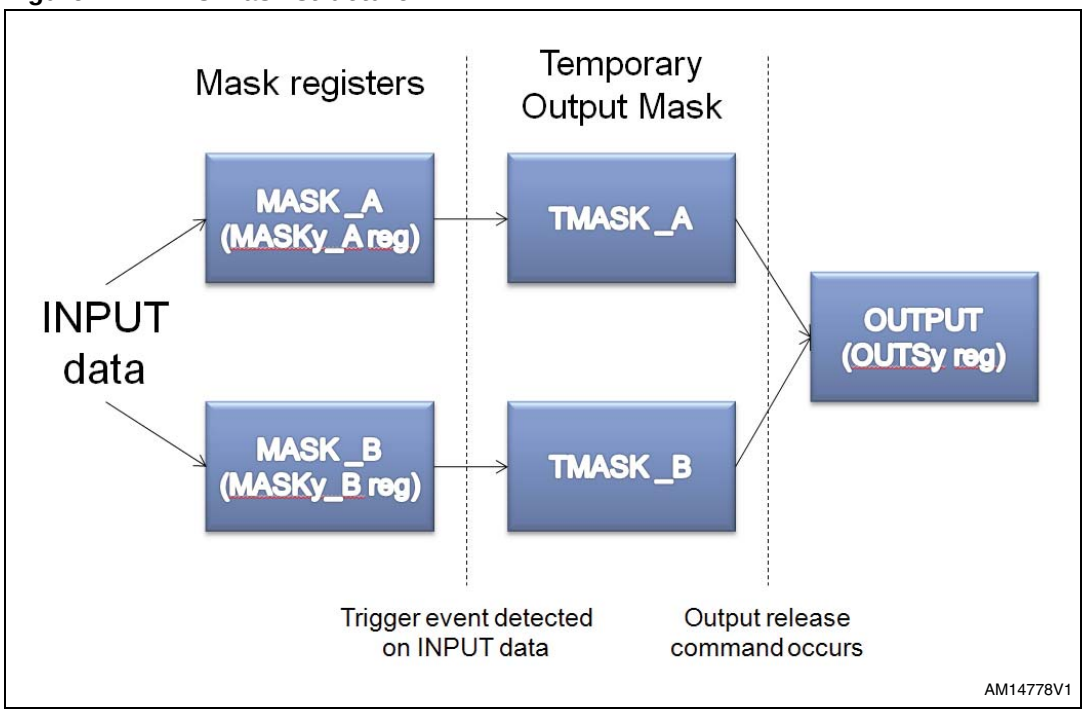
# 7 Axis mask filter

The axis mask filter is used to allow or prevent axes and sign triggers. It is possible to mask in or out all the 3 axes, X, Y and Z, both in a positive and negative direction. Moreover, it is also possible to mask in or out the vector value V (defined in *Section 5.3*).

There are two independent mask registers (MASKA_y and MASKB_y) and two temporary output axis masks; they are connected to the output register (OUTSy) as shown in *Figure 12*: both mask registers have a corresponding temporary output mask in the internal memory area (this area is not accessible by the user), which contains current results of the trigger event used as the axis mask for the next comparison or output commands. The temporary mask value is set as output when one of the output commands (such as CONT, OUTC, OUTW, etc....) is performed. Finally, the output register is cleared by reading OUTSy itself.

The ABS bit (Unsigned/signed setting) and R_TAM bit (Release temporary output mask settings) in SETTy affect trigger events and temporary mask release.

**Figure 12. Axis mask structure**




## 7.1 Mask registers

Mask registers MASKy_A and MASKy_B are used to enable or disable mask action on the input data (X,Y,Z,V). Data is filtered through mask: if a mask bit is set to 1, then the corresponding axis and sign is enabled.

The default mask register is MASKy_A. However, it is possible to change the current active mask by using the dedicated commands SELMA (MASKy_A selected) and SELSA (MASKy_B selected).

For each axis, four different mask settings are possible:

1. Positive axis bit = 0 / Negative axis bit = 0, axis is then disabled.
2. Positive axis bit = 0 / Negative axis = 1, negative data of axis is then enabled.
3. Positive axis bit = 1 / Negative axis = 0, positive data of axis is then enabled.
4. Positive axis bit = 1 / Negative axis = 1, full axis is then enabled.

*Table 87* shows the content of a MASKy register.

**Table 87. MASKy register**

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| P_X | N_X | P_Y | N_Y | P_Z | N_Z | P_V | N_V |

The order of axis mask evaluation is the following:

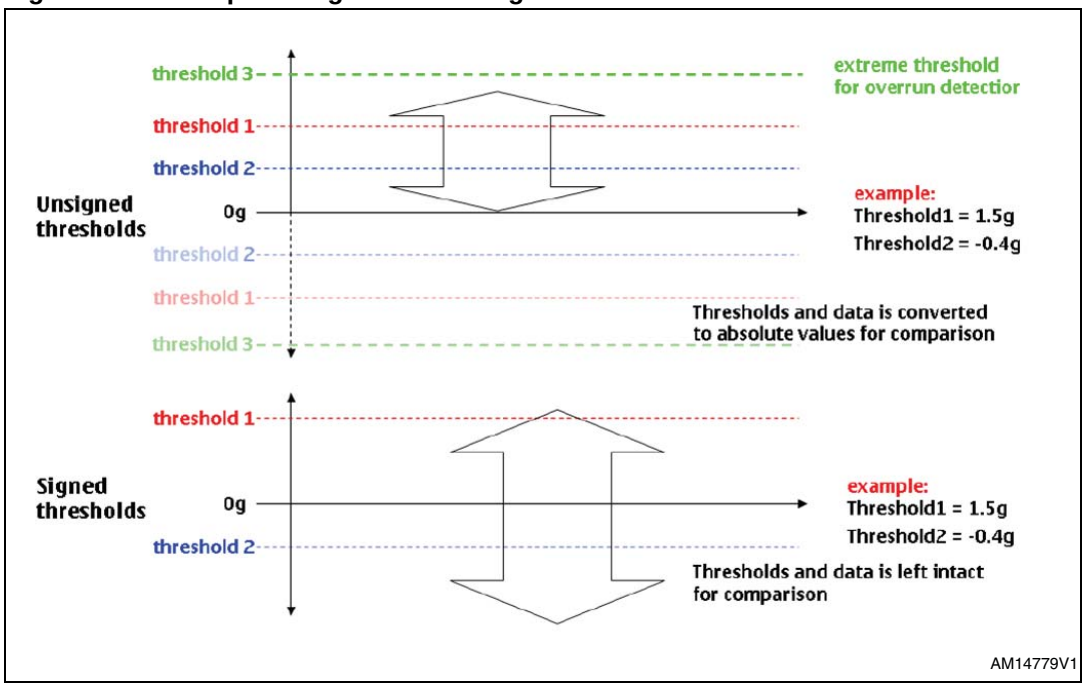
+X, -X, +Y, -Y, +Z, -Z, +V, -V.

## 7.2 Sign filter

The sign filter function is used to define whether the thresholds are signed (ABS bit set to 1 in the SETTy register) or unsigned (SETTy, bit ABS set to 0).

*Figure 13* shows how the bit ABS in the SETTy register affects the thresholds.

**Figure 13. Example of signed and unsigned thresholds**




When the ABS bit is set to 0 in the SETTy register, thresholds are symmetric to the zero level (refer to the first plot of *Figure 13*).

When the ABS bit is set to 1 in the SETTy register, thresholds are sign dependent (refer to the second plot of *Figure 13*).

## 7.3 Temporary output mask

The temporary output masks (TMASK_A or TMASK_B) contain temporary results of trigger events; basically, they represent the current active masks. The temporary output mask value is set to the output register when one of the output commands (such as CONT, OUTC, OUTW, etc....) is performed.

Depending on the triggered event, four different kinds of temporary mask are possible for each axis:

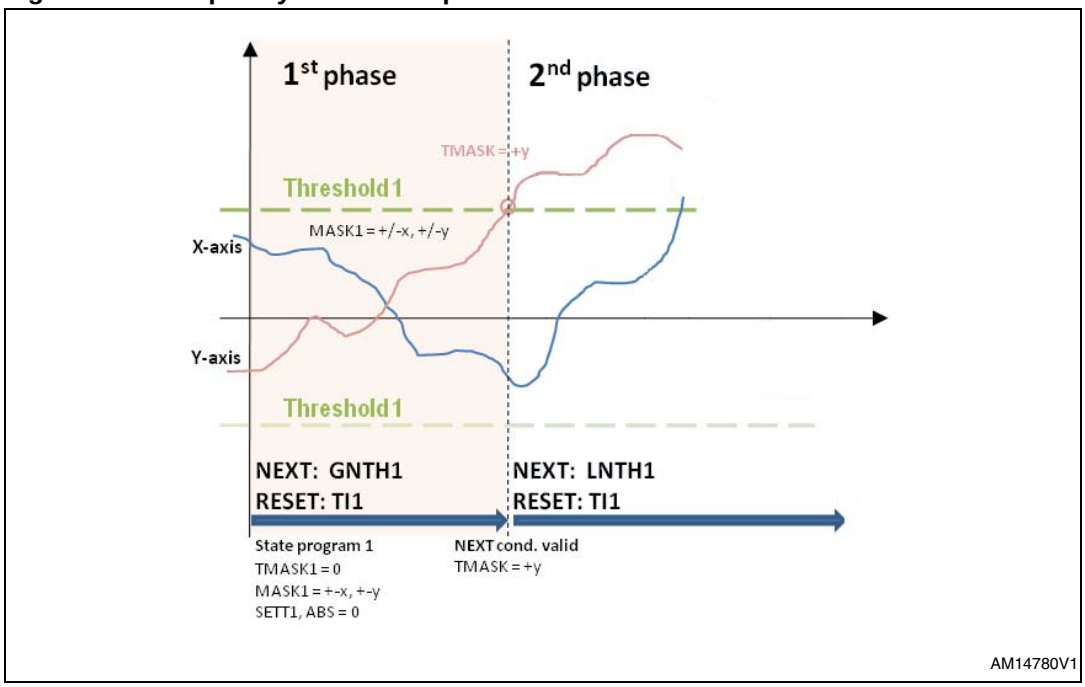
1. Positive axis bit = 0 / Negative axis bit = 0, axis is not then triggered.
2. Positive axis bit = 0 / Negative axis bit = 1, negative data of axis is then triggered.
3. Positive axis bit = 1 / Negative axis bit = 0, positive data of axis is then triggered.
4. Positive axis bit = 1 / Negative axis bit = 1, not possible case.

The temporary output mask is cleared in the following cases:

- when the program starts;
- by using the REL command;
- when the R_TAM bit in the SETTy register is set to 1. In this case, the temporary output mask is set to default after every NEXT command.

As described in *Section 7.1*, MASKx registers contain information about the axes to evaluate. However, when R_TAM = 0 (in SETTy register), only the first triggered axis is placed on the temporary mask (TMASKx) and considered during program execution. In the case of two axes triggering the State Machine at the same time, the temporary output mask selects only the first axis by following the order: X, Y, Z and V. Once the axis has been selected it remains the only one considered until the State Machine program restarts (after the END state or a RESET condition). An example of how the temporary mask is updated and how its value affects the State Machine is shown in *Figure 14*.

**Figure 14. Temporary mask example**



The example of *Figure 14* starts with the following settings:

- MASK1 = F0h (+-x enabled, +-y enabled);
- ABS = 0 in SETT1 register (Unsigned threshold);
- TMASK1 = 0 (Temporary mask default value).

Moreover, the R_TAM bit in the SETT1 register is set to 0, so that the mask is frozen on the axis that triggers the condition.

In the first phase (State 1), the NEXT condition (GNTH1 in this case) is evaluated on all the axes specified by MASK1 (in this case: +-x and +-y). At the beginning, both X and Y are lower than threshold 1, but after a while, acceleration on the Y-axis exceeds threshold 1. So, the value +y is stored in the Temporary Mask, and the State Machine is triggered on this axis in the following conditions.

In the second phase (State 2), a “lower than threshold 1” condition is evaluated. This condition is evaluated just on the positive Y-axis regardless of whether the acceleration on the X-axis is lower than threshold 1. This is because TMASK = +y and the R_TAM bit was set to 0 in the SETT1 register.

Note that if the R_TAM bit was set to 1 (instead of 0), then the condition LNTH1 would be valid on the first sample evaluated in the second phase (State 2). In fact, when the R_TAM bit is 1, the standard mask (MASK1) is always evaluated regardless of the value in the temporary mask.

## 7.4 Output register (OUTSy)

The output register is updated with the value stored in the Temporary Output Mask (see *Section 7.3*) when one of the output commands (such as CONT, OUTC, OUTW, etc....) is performed.

The OUTSy register is cleared after reading itself. Moreover, by reading this register the interrupt status changes:

- when the interrupt is latched (IEL bit = 0 in CTRL_REG3 register), it becomes low (if it is active high) or high (if it is active low);
- when the interrupt is pulsed (IEL bit = 1 in CTRL_REG3 register), a new interrupt pulse can be generated. In fact, until register OUTSy has not been read, no more interrupt can be generated.

# 8 Peak detection

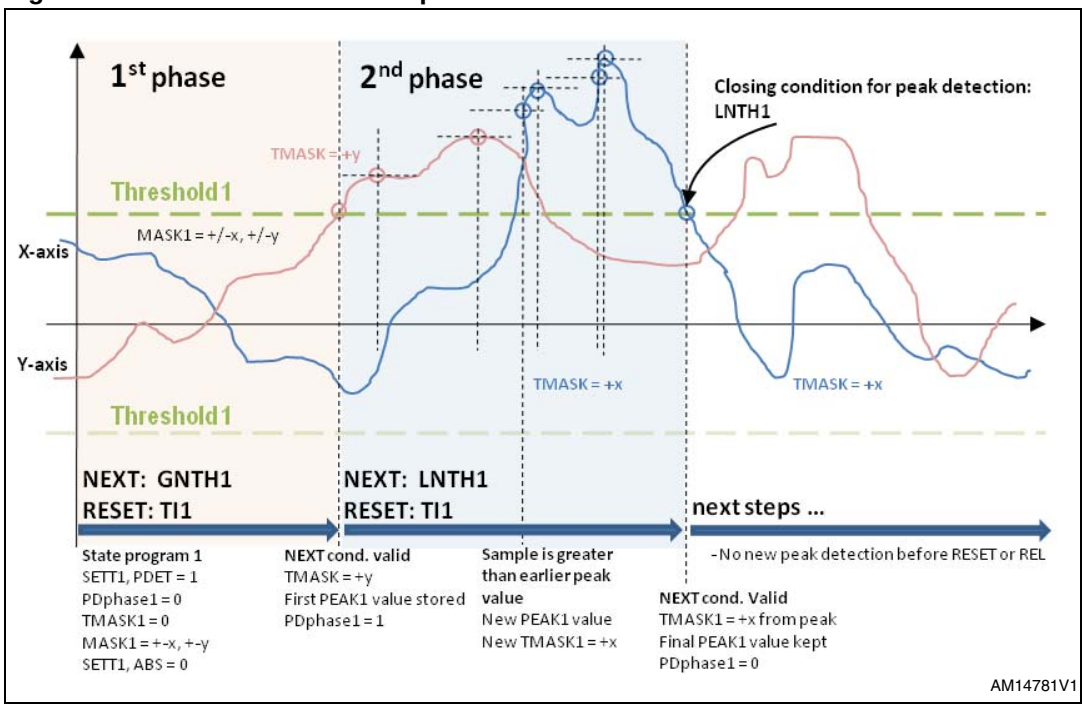
The purpose of this function is to memorize the absolute highest peak value reached by any axis during the State Machine execution and set the temporary axis mask (TMASK described in *Section 7.3*) to follow the axis with the highest peak value.

The peak detection function is available both in State Machine 1 and in State Machine 2. The two different absolute peak values are stored respectively in PEAK1 (19h) and PEAK2 (1Ah) registers. To enable peak detection, the P_DET bit in the SETTy register must be set to 1.

Peak detection is a separate and optional function of the available opcode commands. In order to find out the peak value, this function always implements a “greater than” condition and converts the measured value to an absolute number.

*Figure 15* shows an example of peak detection:

**Figure 15. Peak detection example**




The example starts with the following settings:

- MASK1 = F0h (+-x enabled, +-y enabled);
- P_DET = 1 in the SETT1 register (Peak detection enabled on State Machine 1);
- ABS = 0 in the SETT1 register (Unsigned threshold);
- TMASK1 = 0 (Temporary mask default value).

In the first phase (State 1), the NEXT condition (GNTH1 in this case) is evaluated. Since the positive Y-axis shows the absolute maximum value, the temporary mask (TMASK1) contains +y and the absolute peak value is stored in the PEAK1 register.

In the second phase (State 2), a higher absolute peak value occurs on the positive X-axis. So, the new temporary mask is TMASK1 = +x, and the PEAK1 register is updated with the new peak value.

Note also that the LNTH1 condition has not been validated when the acceleration on the Y-axis becomes lower than threshold 1, because in this second phase the temporary mask (TMASK) has been changed to +x (axis with highest peak).

In the following states, there are no more higher absolute peak values on any axis. So, TMASK1 still contains +x axis, and the peak value is the one stored in the PEAK1 register previously.

# 9 Examples of State Machine configurations

## 9.1 Toggle

Toggle is a simple State Machine configuration that generates an interrupt every n sample.

The idea is to use a timer to count n samples. To do this, in the first state of the machine the RESET condition is ignored (NOP) and the NEXT condition contains a simple timer (TI3) which counts three samples.

**Table 88. Register configuration for toggle application**

| Register | Address | Value |
|---|---|---|
| CTRL_REG1 | 0x21 | 0x01 |
| CTRL_REG3 | 0x23 | 0x48 |
| CTRL_REG4 | 0x20 | 0x67 |
| CTRL_REG5 | 0x24 | 0x00 |
| TIM3_1 | 0x51 | 0x03 |
| ST1_1 | 0x40 | 0x03 |
| ST1_2 | 0x41 | 0x11 |

**Figure 16. Toggle State Machine**

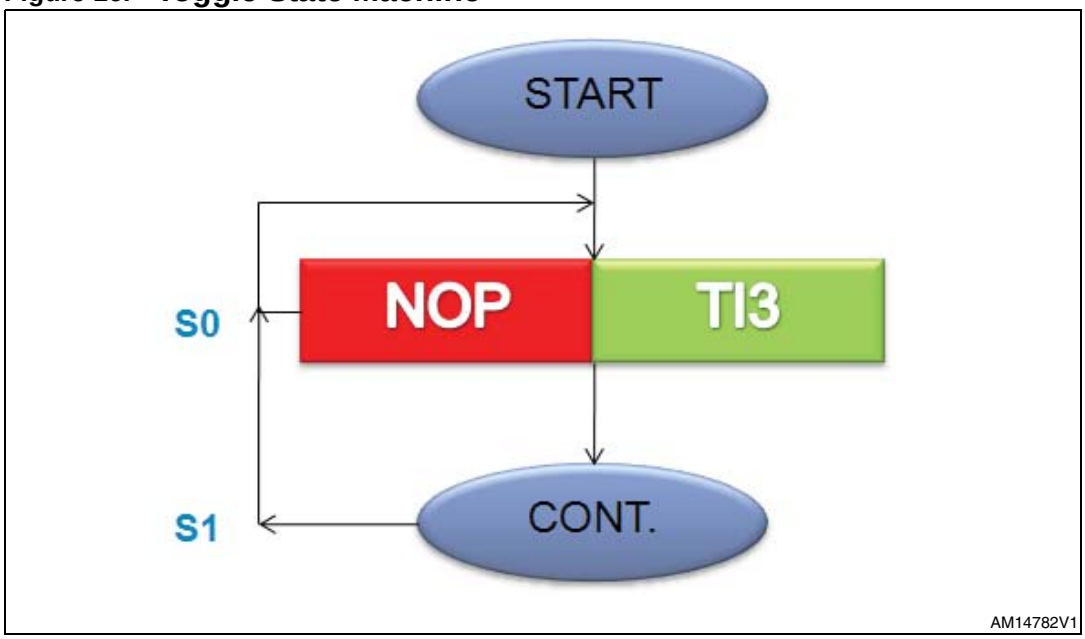
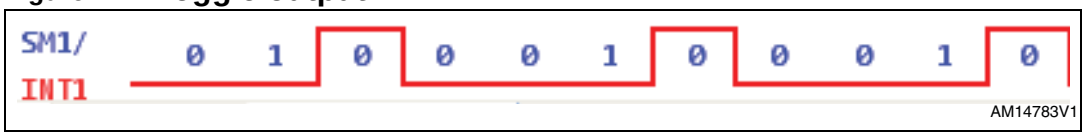



**Figure 17. Toggle output**



## 9.2 Wake-up

For an ultra low-power application it is desirable to have an interrupt signal that wakes up the system after a movement.

This application can also be done with a State Machine with just one state.

In this case the RESET condition is ignored (NOP) and the NEXT condition is a “any/triggered axis greater than threshold 1” (GNTH1). So, the next condition is satisfied when one or more axes exceeds threshold 1 value.

**Table 89. Register configuration for wake-up application**

| Register | Address | Value |
|---|---|---|
| CTRL_REG1 | 0x21 | 0x01 |
| CTRL_REG3 | 0x23 | 0x48 |
| CTRL_REG4 | 0x20 | 0x67 |
| CTRL_REG5 | 0x24 | 0x00 |
| THRS1_1 | 0x57 | 0x55 |
| ST1_1 | 0x40 | 0x05 |
| ST1_2 | 0x41 | 0x11 |

**Figure 18. Wake-up State Machine**

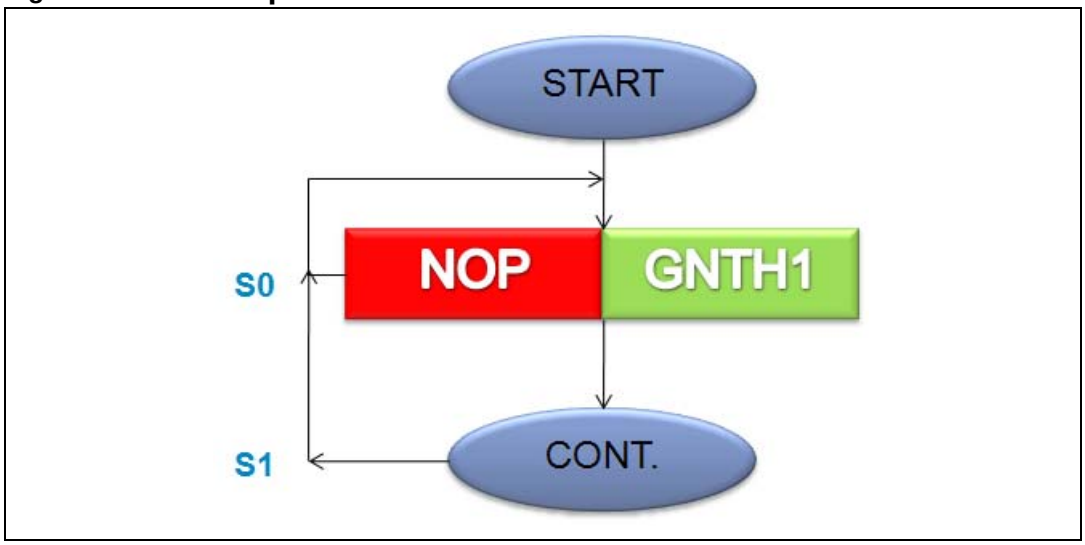
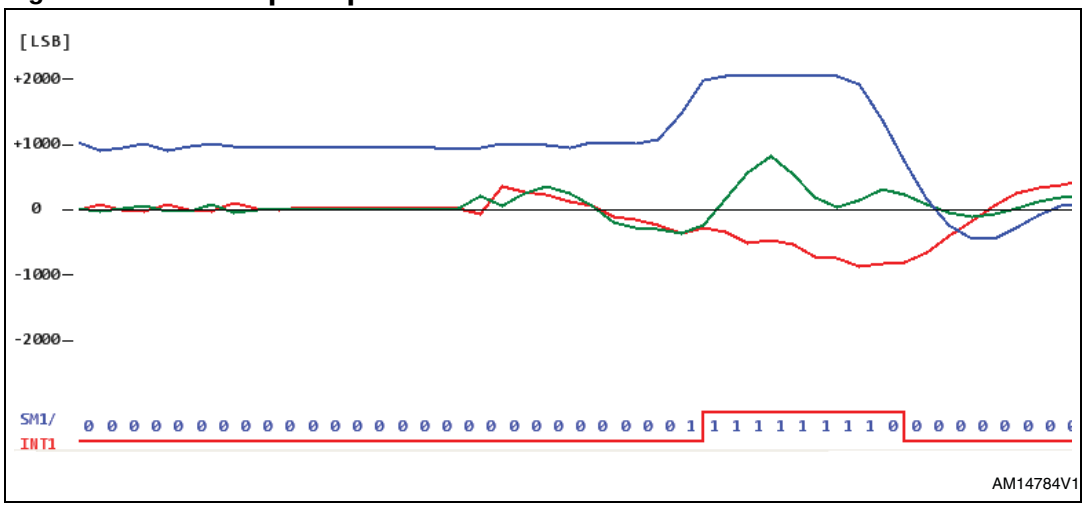


Figure 19. **Wake-up output**




## 9.3 Freefall

This feature is used to detect when a system is dropping down just to protect data on the hard drive. If the object is in freefall, the X-axis, Y-axis and Z-axis have zero acceleration.

To implement this function, the first state of the State Machine (ST0_1, 0x40) has an NOP in the RESET condition and “all axes less than or equal to threshold2” (LLTH2) in the NEXT condition.

An additional state (ST1_1, 0x41) checks whether the axes remain under the previous condition for 100 ms by using a GNTH2 command in the RESET condition and a T1 command in the NEXT condition.

In this way the freefall is detected only when all three axes are below the threshold for at least 100 ms.

An interrupt on INT1 is generated when the CONT state is reached (that is when a freefall event has been detected).

Table 90. **Register configuration for freefall application**

| Register | Address | Value | Comments |
|---|---|---|---|
| CTRL_REG3 | 0X23 | 0x48 | -INT1 enabled<br>-Interrupt active HIGH<br>-Interrupt latched (for Pulsed set Value to 0x68) |
| CTRL_REG4 | 0x20 | 0x77 | ODR = 400 Hz |
| TIM1_1L | 0x55 | 0x28 | Freefall duration (= 100 ms) |
| THRS2_1 | 0x56 | 0x18 | Freefall threshold (= 375 mg) |
| MASK1_B | 0x59 | 0xA8 | Enable positive X, Y, Z masks |

Table 90. Register configuration for freefall application (continued)

| Register | Address | Value | Comments |
|---|---|---|---|
| MASK1_A | 0x5A | 0xA8 | Enable positive X, Y, Z masks |
| SETT1 | 0x5B | 0x03 | -Unsigned threshold<br>-Standard mask always evaluated |
| ST1_1 | 0x40 | 0x0A | Reset: NOP / Next: LLTH2 |
| ST1_2 | 0x41 | 0x61 | Reset: GNTH2 / Next: TI1 |
| ST1_3 | 0x42 | 0x11 | CONTt |
| CTRL_REG1 | 0x21 | 0x01 | State Machine 1 enabled |

Figure 20. Freefall State Machine

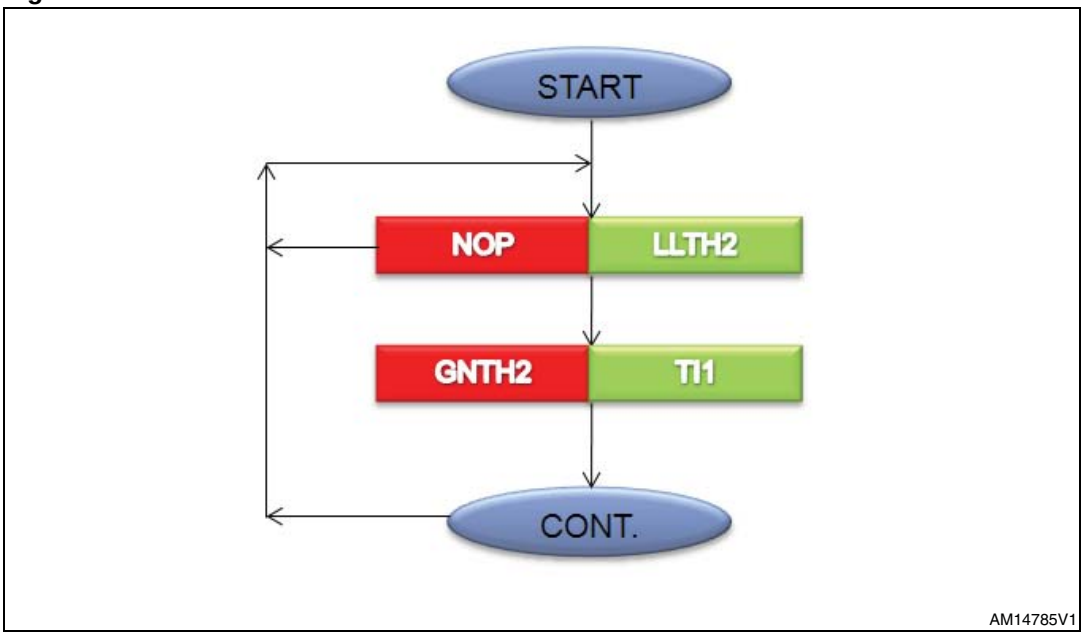
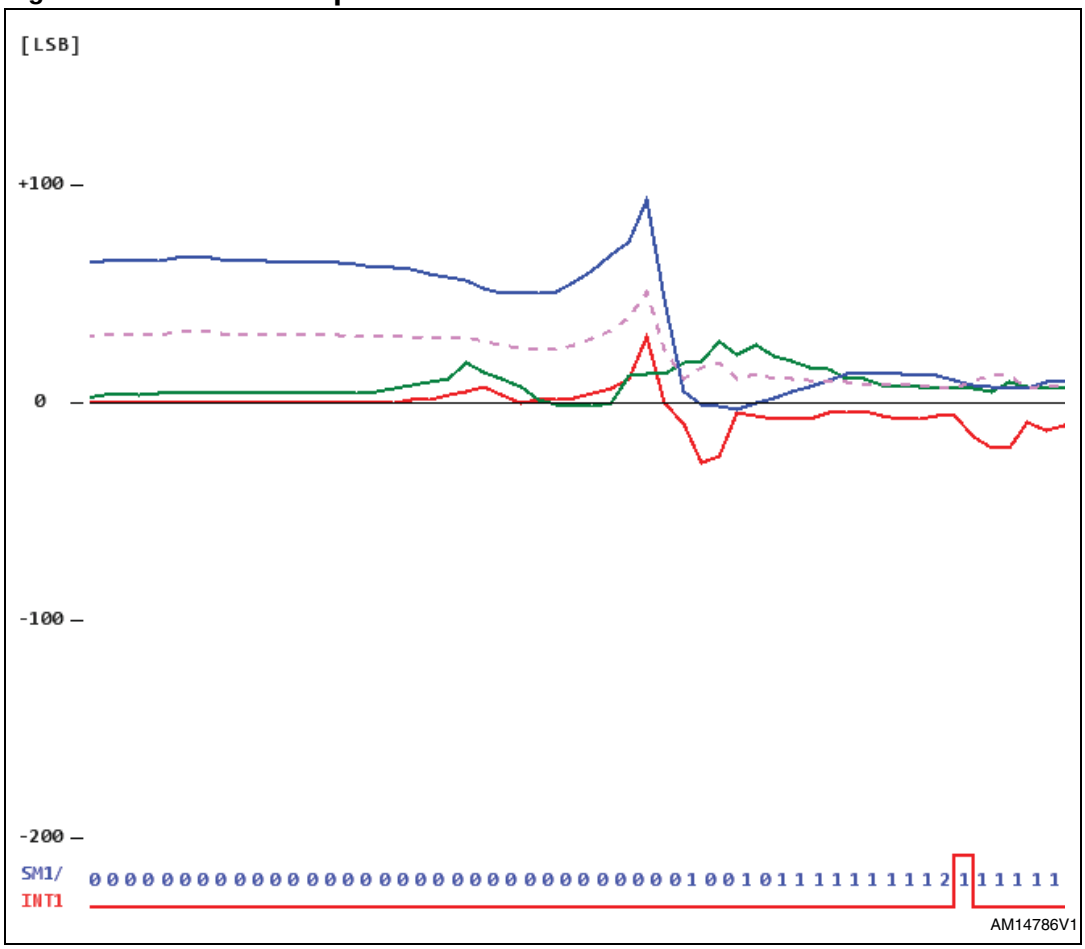


Figure 21. **Freefall Output**




## 9.4 Double-turn

The idea may be to use this function in a mobile phone to switch on or switch off the ring tone by recognizing a gesture like Face Up - Face Down - Face Up.

The function is performed using 4 states:

- check if the acceleration on the Z-axis is lower than threshold 1
- check if the acceleration on the Z-axis is lower than threshold 2
- check if the acceleration on the Z-axis is higher than threshold 2
- check if the acceleration on the Z-axis is higher than threshold 1.

**Table 91. Register configuration for double-turn application**

| Register | Address | Value |
|---|---|---|
| CTRL_REG1 | 0x21 | 0x01 |
| CTRL_REG3 | 0x23 | 0x48 |
| CTRL_REG4 | 0x20 | 0x67 |
| CTRL_REG5 | 0x24 | 0x00 |

Table 91. Register configuration for double-turn application (continued)

| Register | Address | Value |
|---|---|---|
| THRS2_1 | 0x56 | 0xD0 |
| THRS1_1 | 0x57 | 0x30 |
| ST1_1 | 0x40 | 0x07 |
| ST1_2 | 0x41 | 0x08 |
| ST1_3 | 0x42 | 0x06 |
| ST1_4 | 0x43 | 0x05 |
| ST1_5 | 0x44 | 0x11 |
| MASK1_B | 0x59 | 0x08 |
| MASK1_A | 0x5A | 0x08 |
| SETT1 | 0x5B | 0x23 |

Figure 22. Double-turn State Machine

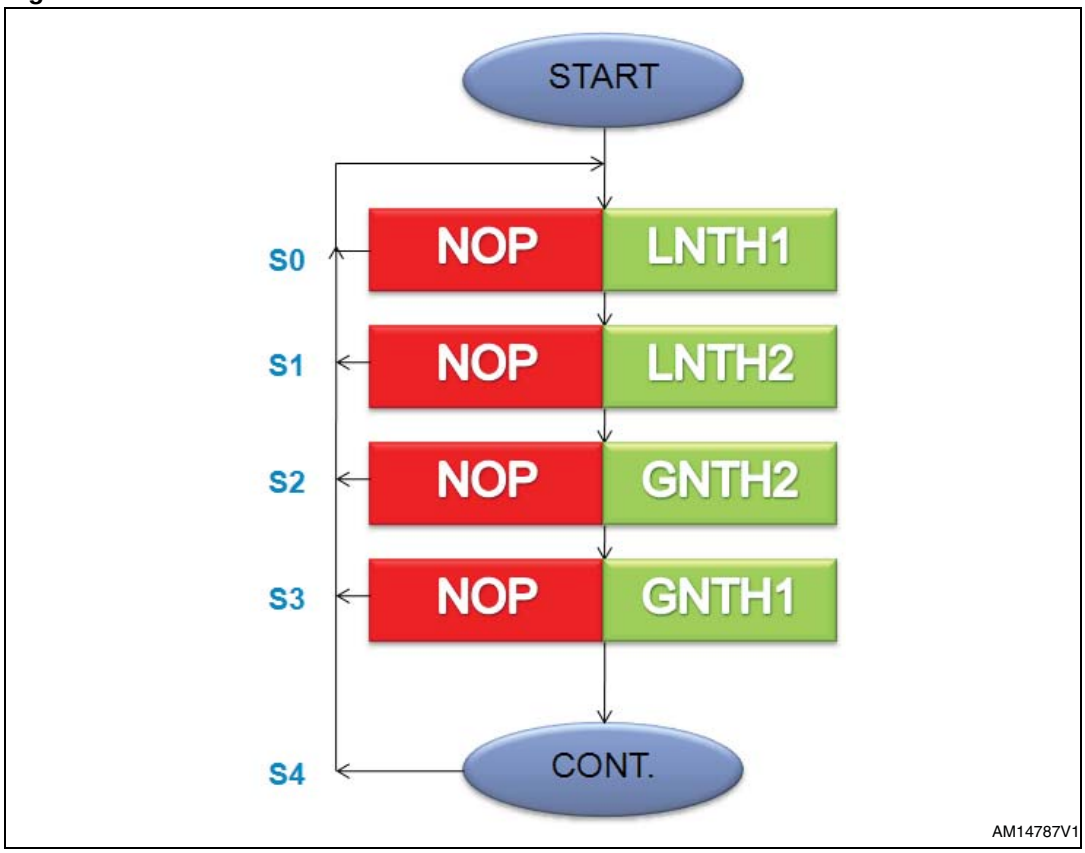
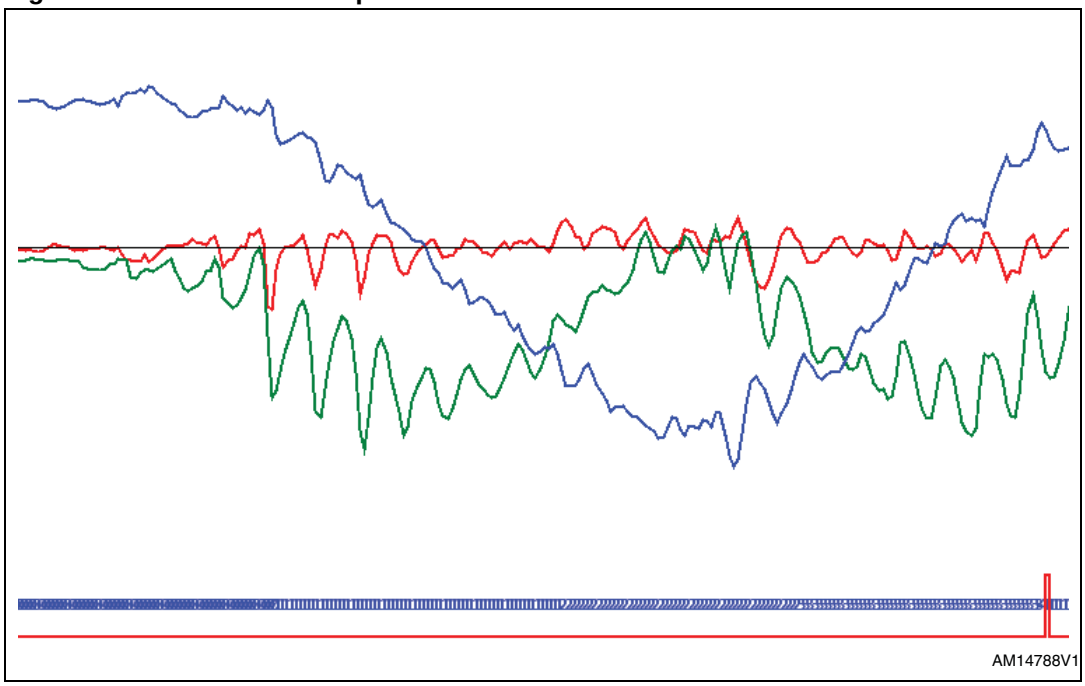


**Figure 23. Double-turn output**




## 9.5 Double-tap

The implementation is intended to recognize the double-tap on the device.

The first two states (GNTH1 | TI1) are used to create a “pre-silence” time window, the NEXT condition is the Timer1. If one or more axes become greater than threshold 1, the RESET condition is satisfied and the program pointer is reset. After that there is the real first tap detection composed of two states, the first one is composed of (NOP | GNTH2), where the next condition is satisfied when one or more triggered axes exceed threshold 2. In the second state (TI3 | LNTH2), it is checked whether the acceleration value on the axis becomes lower than threshold 2 within a time defined by Timer 3. The result of these states is to create a time window (TI3) where the acceleration is first higher, and then lower, than threshold 2.

When the first tap has been detected, the system waits 20 ms (TI4), again doing a pre-silence time window (GNTH1 | TI1) and starting with the second tap detection.

**Table 92. Register configuration for double-tap application**

| Register | Address | Value |
|---|---|---|
| CTRL_REG1 | 0x21 | 0x01 |
| CTRL_REG3 | 0x23 | 0x48 |
| CTRL_REG4 | 0x20 | 0x67 |
| CTRL_REG5 | 0x24 | 0x00 |
| TIM4_1 | 0x50 | 0x02 |
| TIM3_1 | 0x51 | 0x01 |

Table 92. Register configuration for double-tap application (continued)

| Register | Address | Value |
|---|---|---|
| TIM2_1L | 0x52 | 0x32 |
| TIM1_1L | 0x54 | 0x07 |
| THRS2_1 | 0x56 | 0x55 |
| THRS1_1 | 0x57 | 0x55 |
| ST1_1 | 0x40 | 0x51 |
| ST1_2 | 0x41 | 0x51 |
| ST1_3 | 0x42 | 0x06 |
| ST1_4 | 0x43 | 0x38 |
| ST1_5 | 0x44 | 0x04 |
| ST1_6 | 0x45 | 0x91 |
| ST1_7 | 0x46 | 0x26 |
| ST1_8 | 0x47 | 0x38 |
| ST1_9 | 0x48 | 0x04 |
| ST1_10 | 0x49 | 0x91 |
| ST1_11 | 0x4A | 0x11 |

Figure 24. Double-tap State Machine

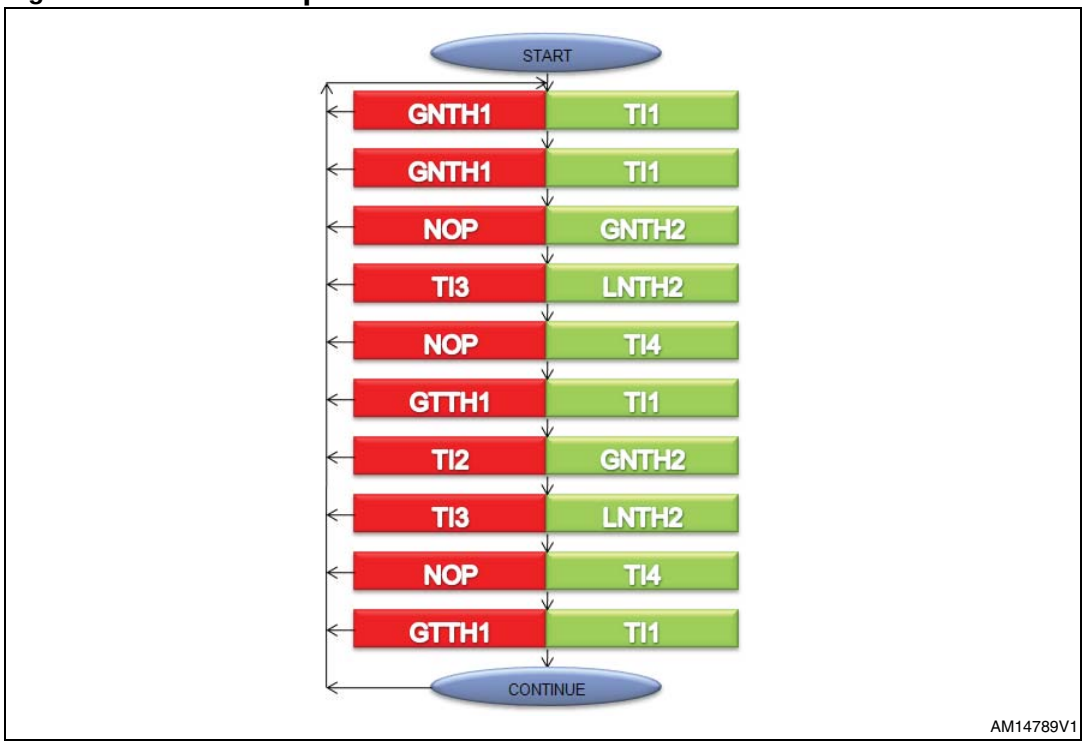
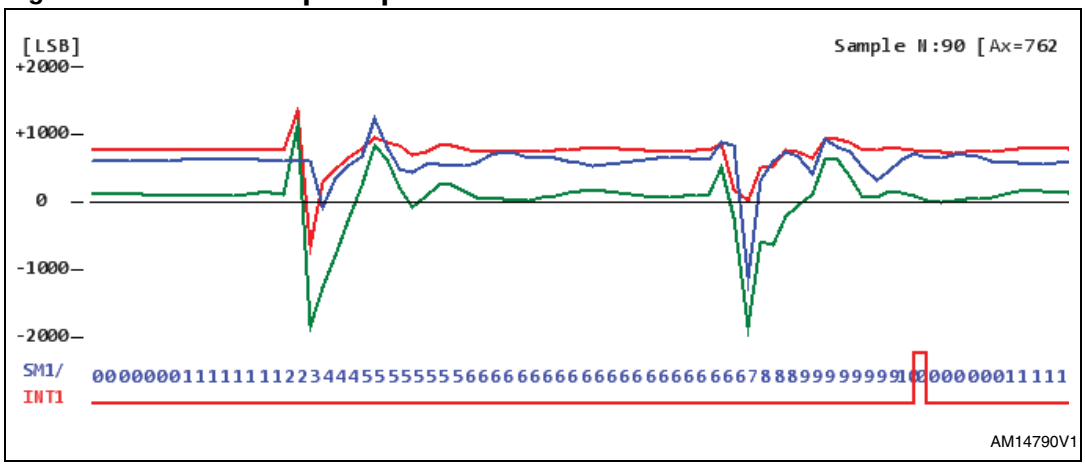


**Figure 25. Double-tap output**




## 9.6 6D position recognition

In this example, the 6D position is recognized by using the LIS3DSH. An interrupt (INT1) is generated when one of the 6 positions has been detected. When the interrupt occurs, the new position is available in the OUTS1 (5Fh) register. *Figure 26* and *Table 93* show the content of the OUTS1 register in the 6 different positions.

**Figure 26. 6D positions**

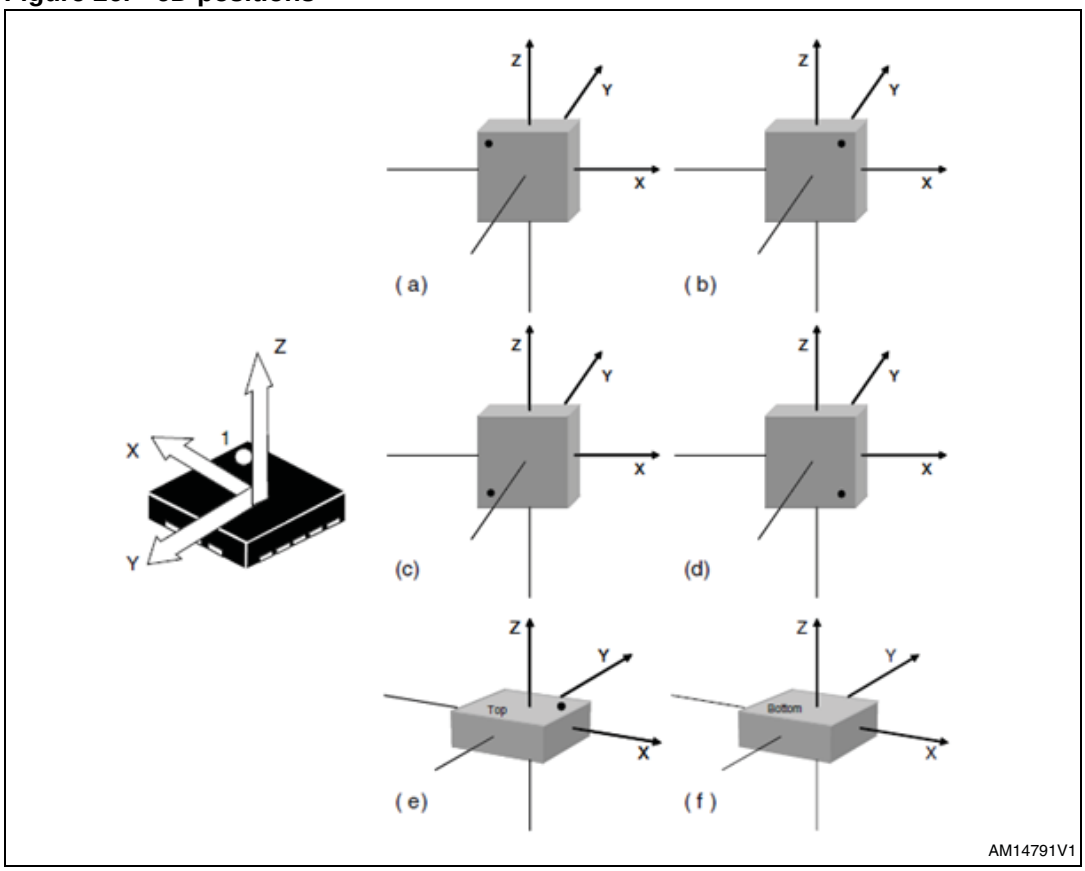


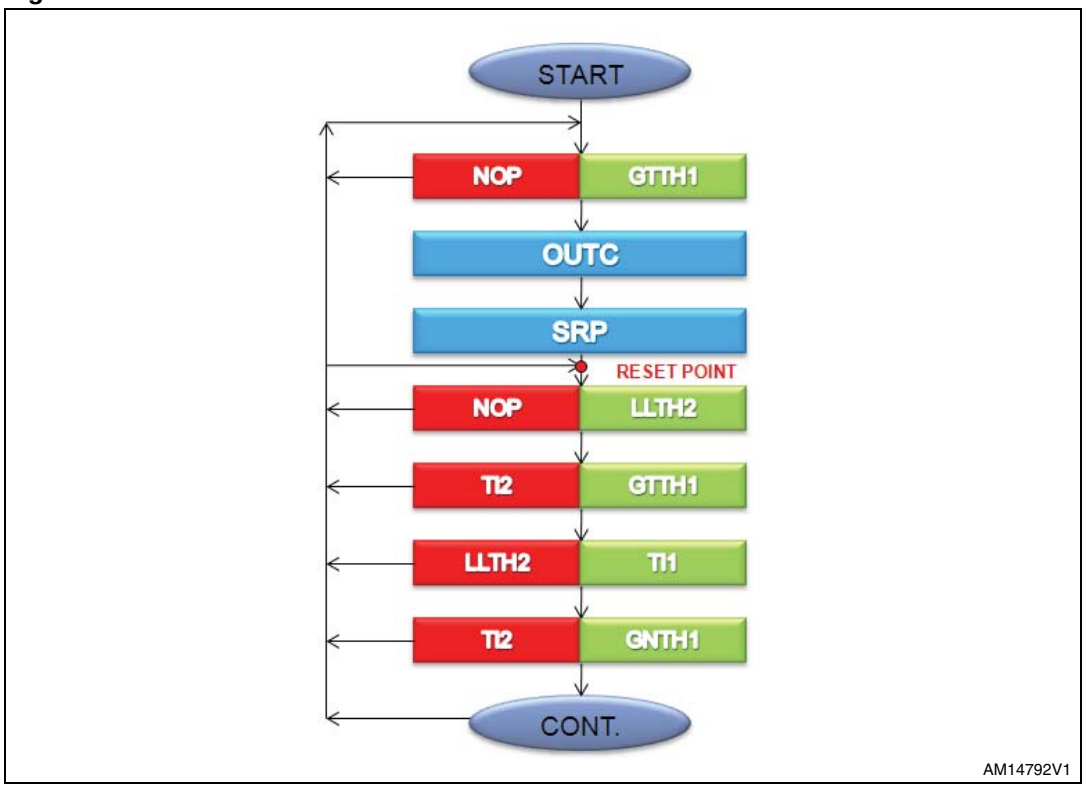
**Table 93. OUTS1 (5Fh) register content in 6D position recognition**

| Case | P_X | N_X | P_Y | N_Y | P_Z | N_Z | P_V | N_V |
|---|---|---|---|---|---|---|---|---|
| (a) | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| (b) | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| (c) | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| (d) | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| (e) | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| (f) | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |

**Table 94. Register configuration for 6D position recognition**

| Register | Address | Value | Comments |
|---|---|---|---|
| CTRL_REG3 | 0X23 | 0x48 | -INT1 enabled<br>-Interrupt active HIGH<br>-Interrupt latched (for Pulsed set Value to 0x68) |
| CTRL_REG4 | 0x20 | 0x97 | ODR = 1600 Hz |
| TIM2_1L | 0x52 | 0x00 | Timer 2 = 1280 ms |
| TIM2_1H | 0x53 | 0x08 | Timer 2 = 1280 ms |
| TIM1_1L | 0x54 | 0x80 | Timer 1 = 80 ms |
| TIM1_1H | 0x55 | 0x00 | Timer 1 = 80 ms |
| THRS2_1 | 0x56 | 0x30 | Threshold 2 = 750 mg |
| THRS1_1 | 0x57 | 0x32 | Threshold 1 = 781 mg |
| MASK1_B | 0x59 | 0xFC | Enable positive and negative X, Y, Z masks |
| MASK1_A | 0x5A | 0xFC | Enable positive and negative X, Y, Z masks |
| SETT1 | 0x5B | 0x23 | -Signed threshold<br>-Standard mask always evaluated |
| ST1_1 | 0x40 | 0x09 | Reset: NOP / Next: GTTH1 |
| ST1_2 | 0x41 | 0x88 | OUTC |
| ST1_3 | 0x42 | 0x33 | SRP |
| ST1_4 | 0x43 | 0x0A | Reset: NOP / Next: LLTH2 |
| ST1_5 | 0x44 | 0x29 | Reset: TI2 / Next: GTTH1 |
| ST1_6 | 0x45 | 0xA1 | Reset: LLTH2 / Next: TI1 |
| ST1_7 | 0x46 | 0x25 | Reset: TI2 / Next: GTTH1 |
| ST1_8 | 0x47 | 0x11 | CONT |
| CTRL_REG1 | 0x21 | 0x01 | State Machine 1 enabled |

**Figure 27. 6D State Machine**



# 10 First in first out (FIFO) buffer

In order to decrease the host processor interaction and facilitate post processing data for events recognition, the LIS3DSH embeds a first in, first out buffer (FIFO) for each of the three output channels, X, Y, and Z.

FIFO use allows a consistent power saving for the system; it can wake up only when needed and burst the significant data out from the FIFO.

The FIFO buffer can work according to six different modes that guarantee a high-level of flexibility during application development: Bypass mode, FIFO mode, Stream mode, Stream-to-FIFO mode, Bypass-to-Stream mode and Bypass-to-FIFO mode.

The programmable Watermark level, FIFO overrun and FIFO empty events can be enabled to generate dedicated interrupts on the INT1 pin.

## 10.1 FIFO description

The FIFO buffer is able to store up to 32 angular rate samples of 16 bits for each channel; data are stored in the 16-bit 2's complement left justified representation.

The data samples set consists of 6 bytes (Xl, Xh, Yl, Yh, Zl, and Zh) and they are released to the FIFO at the selected output data rate (ODR).

The new sample set is placed in the first empty FIFO slot until the buffer is full, therefore, the oldest value is overwritten.

**Table 95. FIFO buffer full representation (32nd sample set stored)**

| Output registers | 0x28h | 0x29h | 0x2Ah | 0x2Bh | 0x2Ch | 0x2Dh |
|---|---|---|---|---|---|---|
| | Xl(0) | Xh(0) | Yl(0) | Yh(0) | Zl(0) | Zh(0) |
| FIFO index | FIFO sample set | | | | | |
| FIFO(0) | Xl(0) | Xh(0) | Yl(0) | Yh(0) | Zl(0) | Zh(0) |
| FIFO(1) | Xl(1) | Xh(1) | Yl(1) | Yh(1) | Zl(1) | Zh(1) |
| FIFO(2) | Xl(2) | Xh(2) | Yl(2) | Yh(2) | Zl(2) | Zh(2) |
| FIFO(3) | Xl(3) | Xh(3) | Yl(3) | Yh(3) | Zl(3) | Zh(3) |
| ... | ... | ... | ... | ... | ... | ... |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| ... | ... | ... | ... | ... | ... | ... |
| FIFO(30) | Xl(30) | Xh(30) | Yl(30) | Yh(30) | Zl(30) | Zh(30) |
| FIFO(31) | Xl(31) | Xh(31) | Yl(31) | Yh(31) | Zl(31) | Zh(31) |

Table 96. FIFO overrun representation (33rd sample set stored and 1st sample discarded)

| Output registers | 0x28h | 0x29h | 0x2Ah | 0x2Bh | 0x2Ch | 0x2Dh |
|---|---|---|---|---|---|---|
| | XI(1) | XH(1) | YI(1) | YH(1) | ZI(1) | ZH(1) |
| FIFO index | Sample set | | | | | |
| FIFO(0) | XI(1) | XH(1) | YI(1) | YH(1) | ZI(1) | ZH(1) |
| FIFO(1) | XI(2) | XH(2) | YI(2) | YH(2) | ZI(2) | ZH(2) |
| FIFO(2) | XI(3) | XH(3) | YI(3) | YH(3) | ZI(3) | ZH(3) |
| FIFO(3) | XI(4) | XH(4) | YI(4) | YH(4) | ZI(4) | ZH(4) |
| ... | ... | ... | ... | ... | ... | ... |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| ... | ... | ... | ... | ... | ... | ... |
| FIFO(30) | XI(31) | XH(31) | YI(31) | YH(31) | ZI(31) | ZH(31) |
| FIFO(31) | XI(32) | XH(32) | YI(32) | YH(32) | ZI(32) | ZH(32) |

*Table 95* represents the FIFO full status when 32 samples are stored in the buffer while *Table 96* represents the next step when the 33rd sample is inserted into FIFO and the 1st sample is overwritten. The new oldest sample set is made available in the output registers.

When FIFO is enabled and the mode is different from Bypass, the output registers (28h to 2Dh) always contain the oldest FIFO sample set.

## 10.2 FIFO registers

The FIFO buffer is managed by three different accelerometer registers, two of these allow the FIFO behavior to be enabled and configured, the third provides information about the buffer status.

### 10.2.1 Control register 6 (0x25)

The FIFO_EN bit in CTRL_REG6 must be set to 1 in order to enable the internal first in, first out buffer; while this bit is set, the LIS3DSH output registers (28h to 2Dh) do not contain the current acceleration value but they always contain the oldest value stored in FIFO.

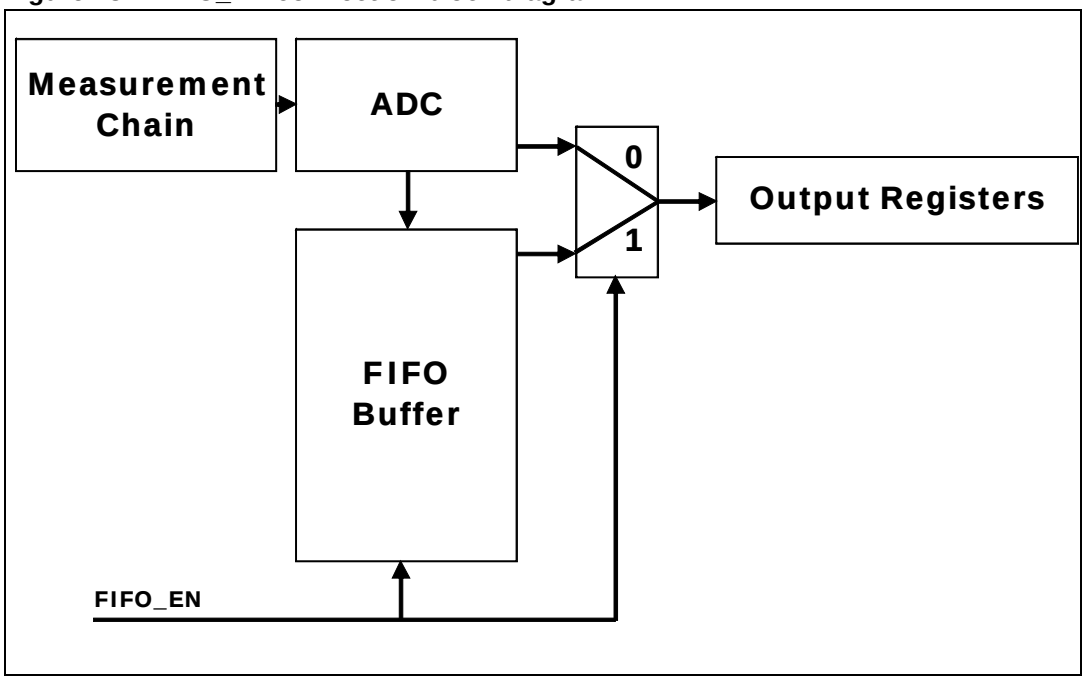
The STP_WTM bit can be used to limit the maximum FIFO buffer depth to the actual Watermark level set in the FIFO_CTRL register.

The ADD_INC bit (default “1”) enables the Address auto-increment during a serial interface multiple byte access.

Table 97. FIFO enable bit in CTRL_REG6

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| X | FIFO_EN | STP_WTM | ADD_INC | X | X | X | X |

Figure 28. FIFO_EN connection block diagram




### 10.2.2 FIFO control register (0x2E)

This register is dedicated to FIFO mode selection and Watermark configuration.

Table 98. FIFO_CTRL

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| FMODE2 | FMODE1 | FMODE0 | WTMP4 | WTMP3 | WTMP2 | WTMP1 | WTMP0 |

FMODE[1:0] bits are dedicated to define the FIFO buffer behavior selection:

Table 99. FIFO buffer behavior selection

| FMODE2 | FMODE1 | FMODE0 | FIFO buffer behavior |
|---|---|---|---|
| 0 | 0 | 0 | Bypass mode |
| 0 | 0 | 1 | FIFO mode |
| 0 | 1 | 0 | Stream mode |
| 0 | 1 | 1 | Stream-to-FIFO mode |
| 1 | 0 | 0 | Bypass-to-Stream mode |

**Table 99. FIFO buffer behavior selection (continued)**

| FMODE2 | FMODE1 | FMODE0 | FIFO buffer behavior |
|---|---|---|---|
| 1 | 0 | 1 | Not used |
| 1 | 1 | 0 | Not used |
| 1 | 1 | 1 | Bypass-to-FIFO mode |

The trigger used to activate Stream-to-FIFO, Bypass-to-Stream and Bypass-to-FIFO modes is related to the INT_SM2 bit value of the STAT register and does not depend on the interrupt pin value and polarity. The trigger is generated also if the selected interrupt is not driven to an interrupt pin.

WTMP[4:0] bits are intended to define the Watermark level; when FIFO content exceeds this value, the WTM bit is set to “1” in the FIFO_SRC register.

### 10.2.3 FIFO source register (0x2F)

This register is updated at every ODR and provides information about the FIFO buffer status.

**Table 100. FIFO_SRC_REG**

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| WTM | OVRN | EMPTY | FSS4 | FSS3 | FSS2 | FSS1 | FSS0 |

- WTM bit is set high when FIFO content exceeds Watermark level.
- OVRN bit is set high when FIFO buffer is full; this means that the FIFO buffer contains 32 unread samples. At the following ODR a new sample set replaces the oldest FIFO value. The OVRN bit is reset when the first sample set has been read.
- EMPTY flag is set high when all FIFO samples have been read and FIFO is empty.
- FSS[4:0] field always contains the current number of unread samples stored in the FIFO buffer. When FIFO is enabled, this value increases at ODR frequency until the buffer is full, whereas, it decreases every time that one sample set is retrieved from FIFO.

Register content is updated synchronous to the FIFO write and read operation.

**Table 101. FIFO_SRC_REG behavior assuming FTH[4:0] = 15**

| WTM | OVRN | Empty | FSS[4:1] | Unread FIFO samples | Timing |
|---|---|---|---|---|---|
| 0 | 0 | 1 | 00000 | 0 | t0 |
| 0 | 0 | 0 | 00001 | 1 | t0 + 1/ODR |
| 0 | 0 | 0 | 00010 | 2 | t0 + 2/ODR |
| ... | ... | ... | ... | ... | ... |
| 0 | 0 | 0 | 01111 | 15 | t0 + 15/ODR |
| 1 | 0 | 0 | 10000 | 16 | t0 + 16/ODR |
| ... | ... | ... | ... | ... | ... |

**Table 101. FIFO_SRC_REG behavior assuming FTH[4:0] = 15 (continued)**

| WTM | OVRN | Empty | FSS[4:1] | Unread FIFO samples | Timing |
|---|---|---|---|---|---|
| 1 | 0 | 0 | 11110 | 30 | t0 + 30/ODR |
| 1 | 0 | 0 | 11111 | 31 | t0 + 31/ODR |
| 1 | 1 | 0 | 11111 | 32 | t0 + 32/ODR |

The Watermark flag, the FIFO overrun and FIFO empty event can be enabled to generate a dedicated interrupt on the INT1 pin by configuring the CTRL_REG6 register.

**Table 102. CNTRL6 (0x25)**

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| X | FIFO_EN | STP_WTM | IF_ADD_INC | I1_EMPTY | I1_WTM | I1_OVERRUN | X |

- I2_WTM bit drives Watermark flag (WTM) on the INT1 pin
- I1_OVRRUN bit drives overrun event (OVRN) on the INT1 pin
- I1_EMPTY bit drives empty event (EMPTY) on the INT1 pin.

If one or more bits are set to “1”, the INT1 pin status is the logical OR combination of the selected signals.

## 10.3 FIFO modes

The LIS3DSH FIFO buffer can be configured to operate in six different modes selectable by the FM[2:0] field in the FIFO_CTRL register. Available configurations ensure a high-level of flexibility and extend the number of functions usable in application development.

Bypass, FIFO, Stream, Stream-to-FIFO, Bypass-to-Stream and Bypass-to-FIFO modes are described in the following paragraphs.

### 10.3.1 Bypass mode

When bypass mode is enabled, FIFO is not operational: buffer content is cleared, output registers (0x28 to 0x2D) are frozen at the last value loaded, and the FIFO buffer remains empty until another mode is selected.

Follow these steps for Bypass mode configuration:

1. Turn on FIFO by setting the FIFO_En bit to “1” in control register 6 (0x25). After this operation the FIFO buffer is enabled but isn’t collecting data, output registers are frozen to the last samples set loaded.
2. Activate Bypass mode by setting the FN[2:0] field to “000” in the FIFO control register (0x2E). If this mode is enabled, the FIFO source register (0x2F) is forced equal to 0x20.

Bypass mode must be used in order to stop and reset the FIFO buffer when a different mode is operating. Note that placing the FIFO buffer into Bypass mode clears the whole buffer content.

### 10.3.2 FIFO mode

In FIFO mode, the buffer continues filling until full (32 sample set stored,) then it stops collecting data and the FIFO content remains unchanged until a different mode is selected.

Follow these steps for FIFO mode configuration:

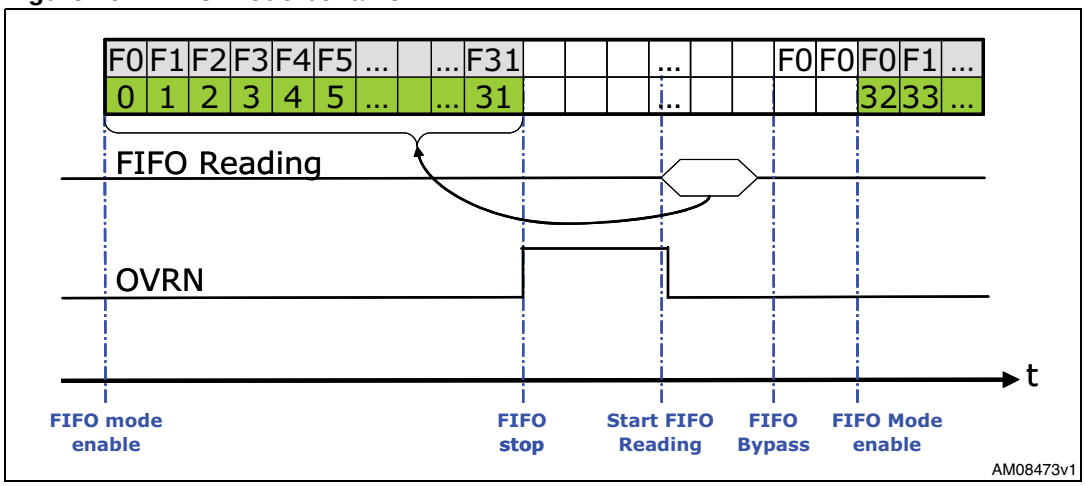
1. Turn on FIFO by setting the FIFO_En bit to “1” in the control register 6 (0x25). After this operation the FIFO buffer is enabled but isn’t collecting data, output registers are frozen to the last samples set loaded.
2. Activate FIFO mode by setting the FN[2:0] field to “001” in the FIFO control register (0x2E).

By selecting this mode, FIFO starts data collection and source register (0x2F) changes according to the number of samples stored. At the end of the procedure, the source register is set to 0xDF and the OVRN flag generates an interrupt if the I1_OVERRUN bit is selected in control register 6. Data can be retrieved when OVRN is set to”1”, performing a 32-sample set reading from the output registers, data can be retrieved also on the WTM flag instead of OVRN if the application requires a lower number of samples. Communication speed is not so important in FIFO mode because data collection is stopped and there is no risk of overwriting acquired data. Before restarting FIFO mode, at the end of the reading procedure it is necessary to transit from Bypass mode.

A FIFO mode application hint is reported below:

1. Set FIFO_En = 1: Enable FIFO.
2. Set FN[2:0] = (0,0,1): Enable FIFO mode.
3. Wait for OVRN or WTM interrupt.
4. Read data from gyroscope output registers.
5. Set FN[2:0] = (0,0,0): Enable bypass mode.
6. Repeat from point 2.

**Figure 29. FIFO mode behavior**




If FIFO mode is enabled, the buffer starts to collect data and fill all the 32 slots (from F0 to F31) at the selected output data rate. When the buffer is full, the OVRN bit goes up and data collection is permanently stopped; the user can decide to read FIFO content at any time because it is maintained unchanged until Bypass mode is selected. The reading procedure is composed of a 32-sample set of 6 bytes for a total of 192 bytes and retrieves data starting from the oldest sample stored in FIFO (F0). The OVRN bit is reset when the first sample set

has been read. The Bypass mode setting resets FIFO and allows the user to enable FIFO mode again.

### 10.3.3 Stream mode

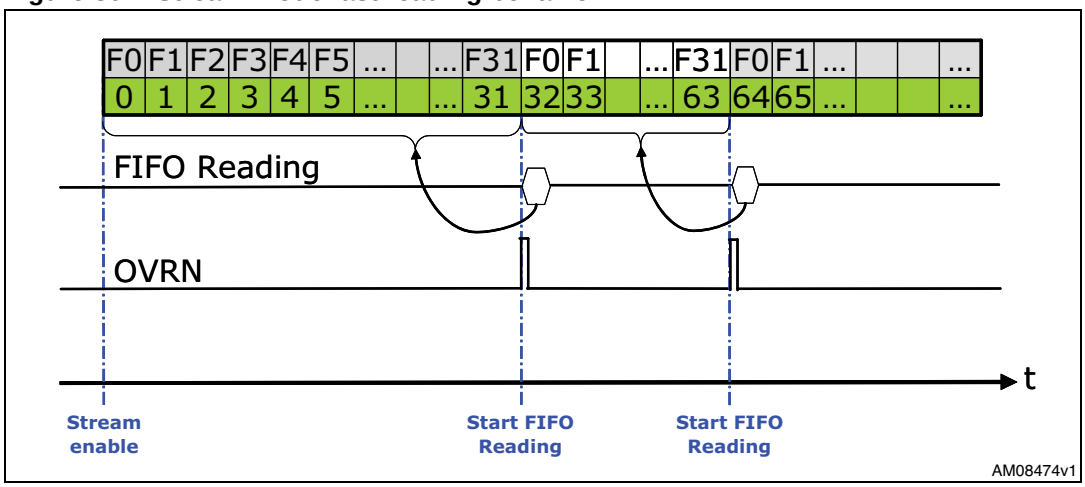
In Stream mode FIFO continues filling, when the buffer is full, the FIFO index restarts from the beginning and older data is replaced by the current. The oldest values continue to be overwritten until a read operation makes free FIFO slots available. Host processor reading speed is most important in order to free slots faster than new data is made available. FM[2:0] bypass configuration is used to stop this mode.

Follow these steps for FIFO mode configuration:

1. Turn on FIFO by setting the FIFO_En bit to “1” in the control register 6 (0x25). After this operation the FIFO buffer is enabled but isn’t collecting data, output registers are frozen to the last samples set loaded.
2. Activate Stream mode by setting the FN[2:0] field to “010” in the FIFO control register (0x2E).

As described, for FIFO mode, data can be retrieved when OVRN is set to”1” performing a 32-sample set reading from output registers, data can be retrieved also on the WTM flag if the application requires a smaller number of samples.

**Figure 30. Stream mode fast reading behavior**



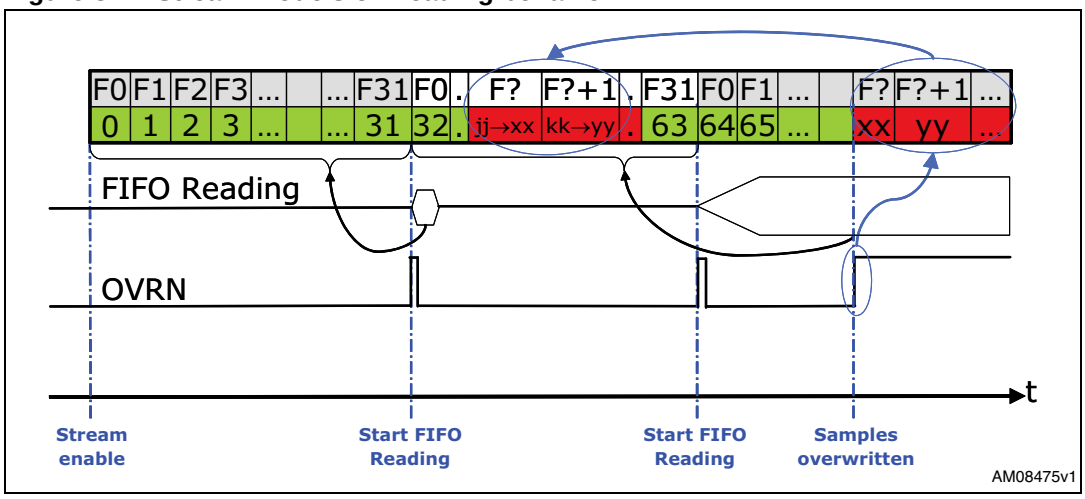

In Stream mode, the FIFO buffer is continuously filling (from F0 to F31) at the selected output data rate. When the buffer is full the OVRN flag goes up and the suggested solution is to read all FIFO samples (192 bytes) faster then 1*ODR, in order to make free FIFO slots available for the new accelerations. This allows a loss of data to be avoided and to decrease the host processor interaction increasing system efficiency. If the reading procedure is not fast enough, three different cases can be observed:

1. FIFO sample set (6 bytes) reading faster than 1*ODR: data are correctly retrieved because a free slot is made available before new data is generated.
2. FIFO sample set (6 bytes) reading synchronous3 to 1*ODR: data are correctly retrieved because a free slot is made available before new data is generated but FIFO benefits are not exploited. This case is equivalent to read data on data-ready interrupt

and does not reduce the host processor interaction compared to the standard accelerometer reading.

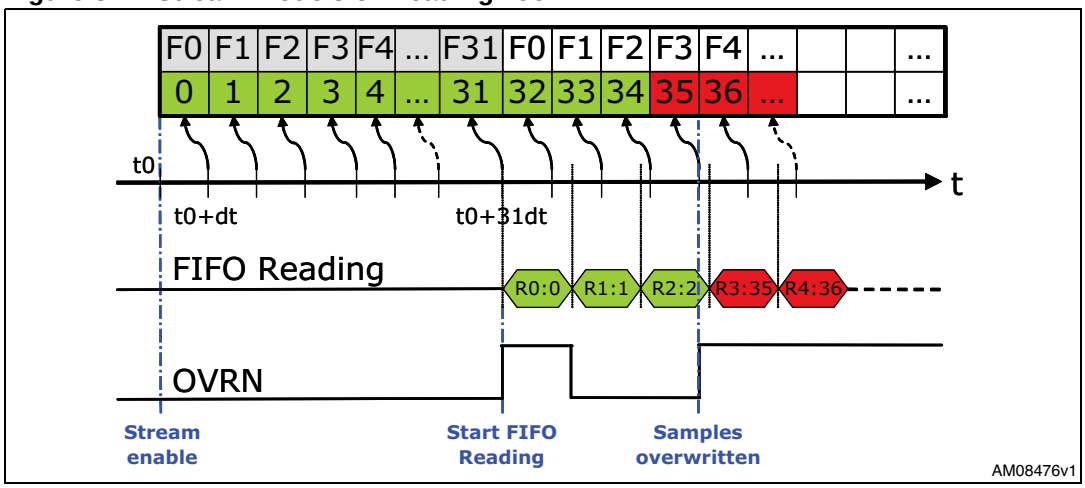
3. FIFO sample set (6 bytes) reading slower than 1*ODR: in this case some data is lost because data recovery is not fast enough to free slots for new angular rate data, *Figure 31*. The number of correctly recovered samples is related to the difference between the current ODR and the FIFO sample set reading rate.

**Figure 31. Stream mode slow reading behavior**




In *Figure 31*, due to slow reading, data from “jj” are not retrieved because they are replaced by the new acceleration samples generated by the system.

**Figure 32. Stream mode slow reading zoom**




After Stream mode enable, FIFO slots are filled at the end of each ODR time frame. The reading procedure must start as soon as the OVRN flag is set to “1”, data are retrieved from FIFO at the beginning of the reading operation. When a read command is sent to the device, the output registers content is moved to the SPI/$I^2C$ register and the current oldest FIFO value is shifted into the output registers in order to allow the next read operation. In the case of a reading slower than 1*ODR, some data can be retrieved from FIFO after that new sample is inserted into the addressed location. In *Figure 32* the fourth read command starts after the refresh of the F3 index and this generates a disconnect in the reading data. The OVRN flag advises the user that this event has taken place. In this example, three correct

samples have been read, the number of correctly recovered samples is dependent on the difference between the current ODR and the FIFO sample set reading timeframe.

### 10.3.4 Stream-to-FIFO mode

This mode is a combination of the Stream and FIFO modes described above. In Stream-to-FIFO mode, the FIFO buffer starts operating in Stream mode and switches to FIFO mode when INT2 occurs.

Follow these steps for Stream-to-FIFO mode configuration:

1. Configure State Machine 2 in order to generate interrupt on INT2.
2. Turn on FIFO by setting the FIFO_En bit to “1” in control register 6 (0x25). After this operation the FIFO buffer is enabled but isn’t collecting data, output registers are frozen to the last samples set loaded.
3. Activate Stream-to-FIFO mode by setting the FN[2:0] field to “011” in the FIFO control register (0x2E).

The interrupt trigger is related to the INT_SM2 bit in the STAT register and it is generated even if the interrupt signal is not driven to an interrupt pad. Mode switch is performed if both INT_SM2 and OVRN bits are set high. Stream-to-FIFO mode is sensitive to the trigger level and not to the trigger edge; this means that if Stream-to-FIFO is in FIFO mode and the interrupt condition disappears, the FIFO buffer returns to Stream mode because the IA bit becomes zero. It is suggested to latch the interrupt signal used as the FIFO trigger in order to avoid losing interrupt events. If the selected interrupt is latched, it is necessary to read the OUTS2 register to clear the INT_SM2 bit; after reading, the INT_SM2 bit takes 2*ODR to go low.

In Stream mode the FIFO buffer continues filling, when the buffer is full, the OVRN bit is set high and the next samples overwrite the oldest. When trigger occurs, two different cases can be observed:

1. If the FIFO buffer is already full (OVRN = “1”), it stops collecting data at the first sample after trigger. FIFO content is composed of #30 samples collected before the trigger event, the sample that has generated the interrupt event and one sample after trigger.
2. If FIFO isn’t yet full (initial transient), it continues filling until it is full (OVRN = “1”) and then, if trigger is still present, it stops collecting data.

**Figure 33. Stream-to-FIFO mode: interrupt not latched**

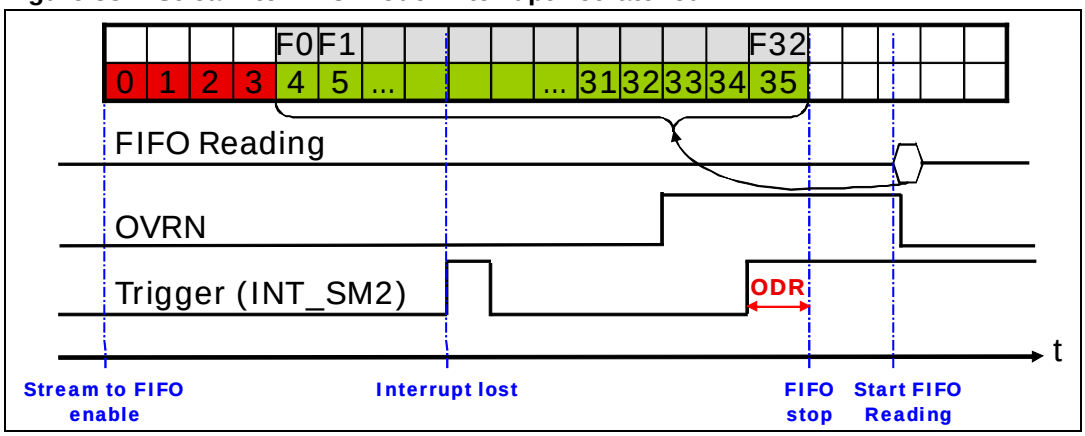
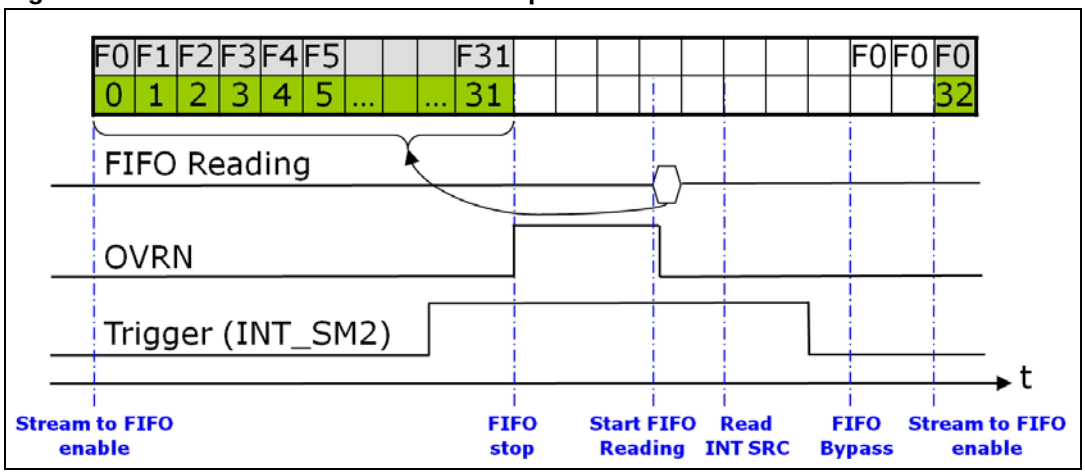


**Figure 34. Stream-to-FIFO mode: interrupt latched**




Stream-to-FIFO can be used in order to analyze the samples history that generates an interrupt; the standard operation is to read FIFO content when FIFO mode is triggered and FIFO buffer is full and stopped.

### 10.3.5 Bypass-to-Stream mode

This mode is a combination of the Bypass and Stream modes described above. In Bypass-to-Stream mode, the FIFO buffer starts operating in Bypass mode and switches to Stream mode when INT2 occurs.

Follow these steps for Bypass-to-Stream mode configuration:

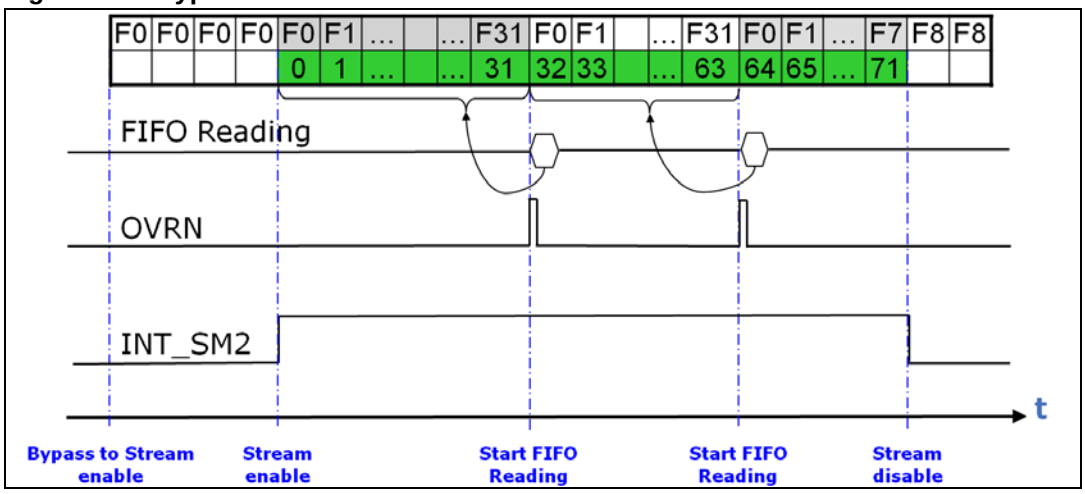
1. Configure State Machine 2 in order to generate interrupt on INT2.
2. Turn on FIFO by setting the FIFO_En bit to “1” in control register 6 (0x25). After this operation the FIFO buffer is enabled but isn’t collecting data, output registers are frozen to the last samples set loaded.
3. Activate Bypass-To-Stream mode by setting the FN[2:0] field to “100” in the FIFO control register (0x2E).

The interrupt trigger is related to the INT_SM2 bit in the STAT register and it is generated even if the interrupt signal is not driven to an interrupt pad. Bypass-to-Stream mode is sensitive to the trigger level and not to the trigger edge, this means that if Bypass-to-Stream is in Stream mode and the interrupt condition disappears, the FIFO buffer returns to Bypass mode because the INT_SM2 bit becomes zero.

It is suggested to latch the interrupt signal used as the stream trigger in order to avoid losing interrupt events. If the selected interrupt is latched, it is needed to read the register OUTS2 to clear the INT_SM2 bit; after reading, the INT_SM2 bit takes 2*ODR to go low.

In Stream mode the FIFO buffer continues filling. When the buffer is full, the OVRN bit is set high and the next samples overwrite the oldest.

**Figure 35. Bypass-to-Stream mode**




Bypass-to-Stream can be used in order to start the acquisition when the configured interrupt is generated.

### 10.3.6 Bypass-to-FIFO mode

This mode is a combination of the bypass and FIFO modes described above. In Bypass-to-FIFO mode, the FIFO buffer starts operating in Bypass mode and switches to FIFO mode when INT2 occurs.

Follow these steps for Bypass-to-FIFO mode configuration:

1. Configure State Machine 2 in order to generate interrupt on INT2.
2. Turn on FIFO by setting the FIFO_En bit to “1” in control register 6 (0x25). After this operation the FIFO buffer is enabled but isn’t collecting data, output registers are frozen to the last samples set loaded.
3. Activate Bypass-to-FIFO mode by setting the FN[2:0] field to “111” in the FIFO control register (0x2E).

The interrupt trigger is related to the INT_SM2 bit in the STAT register and it is generated even if the interrupt signal is not driven to an interrupt pad. Bypass-to-FIFO mode is sensitive to the trigger level and not to the trigger edge, this means that if Bypass-to-FIFO is in FIFO mode and the interrupt condition disappears, the FIFO buffer returns to Bypass mode because the INT_SM2 bit becomes zero.

It is suggested to latch the interrupt signal used as the stream trigger in order to avoid losing interrupt events. If INT2 is latched, it is necessary to read the register OUTS2 to clear the INT_SM2 bit; after reading, the INT_SM2 bit takes 2*ODR to go low.

In FIFO mode the FIFO buffer collects data until it is full and then stops acquisition.

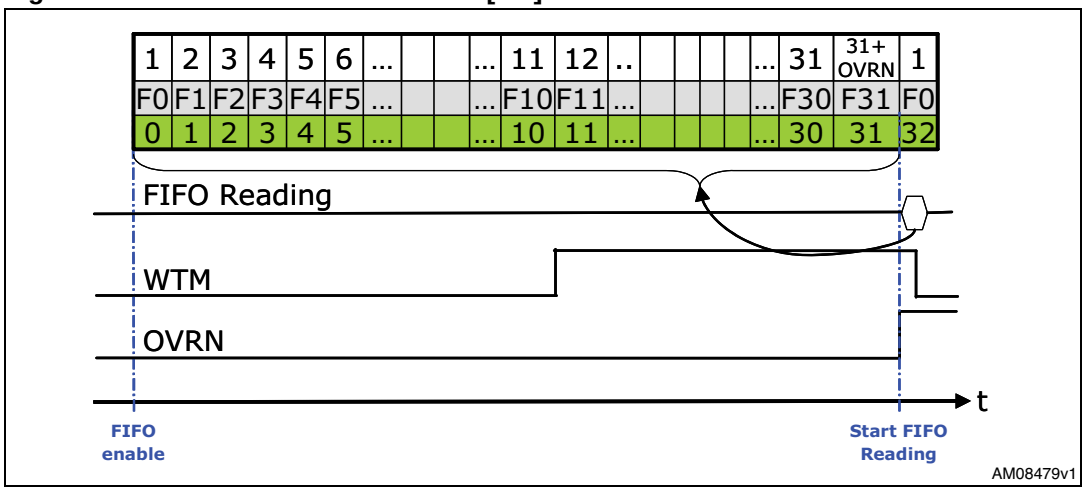
## 10.4 Watermark

Watermark is a configurable flag that can be used to generate specific interrupt in order to know when the FIFO buffer contains at least the number of samples defined as the Watermark level. The user can select the desired level in a range from 0 to 31 using the WTM[4:0] field in the FIFO control register while the FIFO source register FSS[4:0] always contains the number of samples stored in FIFO. If FSS[4:0] is greater than WTM[4:0], the

WTM bit is set high in the FIFO source register; on the contrary, WTM is driven low when the FSS[4:0] field becomes lower than WTM[4:0]. FSS[4:0] increases by one step at the ODR frequency and decreases by one step every time that a sample set reading is performed by the user.

FIFO depth can be limited at the Watermark value by setting to “1” the WTM_EN bit in the CTRL_REG6 register (25h); when this feature is activated both WTM and OVERRUN flags in the FIFO_SRC register have the same behavior.

**Figure 36. Watermark behavior - FTH[4:0] = 10**




In *Figure 36*, the first row indicates the FSS[4:0] value, the second row indicates the relative FIFO slot and the last row shows the incremental FIFO data. Assuming WTM[4:0] = 10, the WTM flag changes from “0” to “1” when the eleventh FIFO slot is filled (F10). *Figure 37* shows that the WTM flag goes down when the FIFO content is less than WTM[4:0], it means that nine unread sample sets remain in FIFO.

The Watermark flag (WTM) can be enabled to generate a dedicated interrupt on the INT1 pin by setting the I1_WTM bit high in CTRL_REG6 (0x25).

## 10.5 Retrieve data from FIFO

When FIFO is enabled and the mode is different from Bypass, reading the output registers (28h to 2Dh) return the oldest FIFO sample set.

Whenever output registers are read, their content is moved to the SPI/$I^2C$ output buffer. FIFO slots are ideally shifted up one level in order to release room for a new sample reception and output registers load the current oldest value stored in the FIFO buffer.

The whole FIFO content is retrieved performing thirty two read operations from output registers, every other read operation returns the same last value until a new sample set is available in the FIFO buffer.

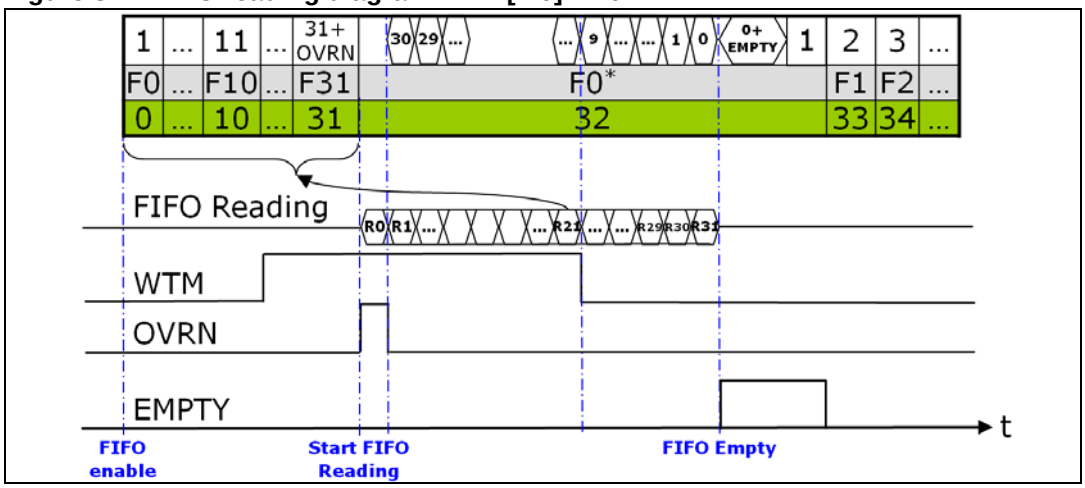
Data can be retrieved from FIFO using every reading byte combination in order to increase the application flexibility (ex: 196 single-byte reading, 32 readings of 6 bytes, 1 multiple reading of 196 bytes, etc.). To perform correct data reading, bit IF_ADD_INC in the CTRL_REG6 register must be set to “1”.

It is suggested to read all FIFO slots in a multiple byte reading of 196 bytes (6 output registers by 32 slots) faster than 1*ODR. In order to minimize communication between

master and slave the reading address is automatically updated by the device (ADD_INC = 1). Address rolls back to 0x28 when register 0x2D is reached.

In order to avoid losing data, the right ODR must be selected according to the serial communication rate available. In the case of standard $I^2C$ mode being used (max. rate 100 kHz), a single sample set reading takes 830 $\mu$s while total FIFO download is about 17.57 ms. $I^2C$ speed is lower than SPI and it needs about 29 clock pulses to start communication (start, slave address, device address+write, restart, device address+read) plus an additional 9 clock pulses for every byte to read. If this suggestion were followed, the complete FIFO reading would be performed faster than 1*ODR, this means that using a standard $I^2C$, the selectable ODR must be lower than 57 Hz. If a fast $I^2C$ mode is used (max. rate 400 kHz), the selectable ODR must be lower than 228 Hz.

**Figure 37. FIFO reading diagram - FTH[4:0] = 10**




In *Figure 37* “Rx” indicates a 6-byte reading operation and “F0*” represents a single ODR slot stretched for diagram convenience.

# 11 Revision history

Table 103. Document revision history

| Date | Revision | Changes |
|---|---|---|
| 21-Dec-2011 | 1 | Initial release. |
| 14-Dec-2012 | 2 | The entire document revised. |