# MPU-3300
# Register Map and Descriptions
# Revision 1.0

# CONTENTS

# 1 Revision History

| Revision Date | Revision | Description |
|---|---|---|
| 5/31/2012 | 1.0 | Initial Release |

## 2 Purpose and Scope

This document provides preliminary information regarding the register map and descriptions for the Motion Processing Units™ MPU-3300™.

The MPU-3300 is the world's first integrated 3-axis gyroscope for Industrial applications. The MPU-3300 features three 16-bit analog-to-digital converters (ADCs) for digitizing the gyroscope outputs. For precision tracking of motion, the parts feature a user-programmable gyroscope full-scale range of ±225 and ±450°/sec (dps).

An on-chip 1024 Byte FIFO buffer helps lower system power consumption by allowing the system processor to read the sensor data in bursts and then enter a low-power mode as the MPU collects more data. Communication with all registers of the device is performed using either $I^2C$ at 400kHz or SPI at 1MHz. For applications requiring faster communications, the sensor and interrupt registers may be read using SPI at 20MHz. Additional features include an embedded temperature sensor and an on-chip oscillator with ±1% variation over the operating temperature range.

By leveraging its patented and volume-proven Nasiri-Fabrication platform, which integrates MEMS wafers with companion CMOS electronics through wafer-level bonding, InvenSense has driven the MPU-3300 package size down to a revolutionary footprint of 4x4x0.9mm (QFN), while providing the highest performance, lowest noise, and the lowest cost semiconductor packaging required for industrial electronic devices. The part features a robust 10,000*g* shock tolerance, and has programmable low-pass filters for the gyroscopes, and the on-chip temperature sensor.

For power supply flexibility, the MPU-3300 operates from VDD power supply voltage range of 2.375V-3.46V.

For more detailed information for the MPU-3300 devices, please refer to the "MPU-3300 Product Specification".

# 3 Register Map

The register map for the MPU-3300 is listed below.

| Addr (Hex) | Addr (Dec.) | Register Name | Serial I/F | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0D | 13 | SELF_TEST_X | R/W | - | | | XG_TEST[4-0] | | | | |
| 0E | 14 | SELF_TEST_Y | R/W | - | | | YG_TEST[4-0] | | | | |
| 0F | 15 | SELF_TEST_Z | R/W | - | | | ZG_TEST[4-0] | | | | |
| 19 | 25 | SMPLRT_DIV | R/W | SMPLRT_DIV[7:0] | | | | | | | |
| 1A | 26 | CONFIG | R/W | - | - | EXT_SYNC_SET[2:0] | | | DLPF_CFG[2:0] | | |
| 1B | 27 | GYRO_CONFIG | R/W | XG_ST | YG_ST | ZG_ST | FS_SEL [1:0] | | - | - | - |
| 23 | 35 | FIFO_EN | R/W | TEMP _FIFO_EN | XG _FIFO_EN | YG _FIFO_EN | ZG _FIFO_EN | - | SLV2 _FIFO_EN | SLV1 _FIFO_EN | SLV0 _FIFO_EN |
| 24 | 36 | I2C_MST_CTRL | R/W | MULT _MST_EN | WAIT _FOR_ES | SLV_3 _FIFO_EN | I2C_MST _P_NSR | I2C_MST_CLK[3:0] | | | |
| 25 | 37 | I2C_SLV0_ADDR | R/W | I2C_SLV0 _RW | I2C_SLV0_ADDR[6:0] | | | | | | |
| 26 | 38 | I2C_SLV0_REG | R/W | I2C_SLV0_REG[7:0] | | | | | | | |
| 27 | 39 | I2C_SLV0_CTRL | R/W | I2C_SLV0 _EN | I2C_SLV0 _BYTE_SW | I2C_SLV0 _REG_DIS | I2C_SLV0 _GRP | I2C_SLV0_LEN[3:0] | | | |
| 28 | 40 | I2C_SLV1_ADDR | R/W | I2C_SLV1 _RW | I2C_SLV1_ADDR[6:0] | | | | | | |
| 29 | 41 | I2C_SLV1_REG | R/W | I2C_SLV1_REG[7:0] | | | | | | | |
| 2A | 42 | I2C_SLV1_CTRL | R/W | I2C_SLV1 _EN | I2C_SLV1 _BYTE_SW | I2C_SLV1 _REG_DIS | I2C_SLV1 _GRP | I2C_SLV1_LEN[3:0] | | | |
| 2B | 43 | I2C_SLV2_ADDR | R/W | I2C_SLV2 _RW | I2C_SLV2_ADDR[6:0] | | | | | | |
| 2C | 44 | I2C_SLV2_REG | R/W | I2C_SLV2_REG[7:0] | | | | | | | |
| 2D | 45 | I2C_SLV2_CTRL | R/W | I2C_SLV2 _EN | I2C_SLV2 _BYTE_SW | I2C_SLV2 _REG_DIS | I2C_SLV2 _GRP | I2C_SLV2_LEN[3:0] | | | |
| 2E | 46 | I2C_SLV3_ADDR | R/W | I2C_SLV3 _RW | I2C_SLV3_ADDR[6:0] | | | | | | |
| 2F | 47 | I2C_SLV3_REG | R/W | I2C_SLV3_REG[7:0] | | | | | | | |
| 30 | 48 | I2C_SLV3_CTRL | R/W | I2C_SLV3 _EN | I2C_SLV3 _BYTE_SW | I2C_SLV3 _REG_DIS | I2C_SLV3 _GRP | I2C_SLV3_LEN[3:0] | | | |
| 31 | 49 | I2C_SLV4_ADDR | R/W | I2C_SLV4 _RW | I2C_SLV4_ADDR[6:0] | | | | | | |
| 32 | 50 | I2C_SLV4_REG | R/W | I2C_SLV4_REG[7:0] | | | | | | | |
| 33 | 51 | I2C_SLV4_DO | R/W | I2C_SLV4_DO[7:0] | | | | | | | |
| 34 | 52 | I2C_SLV4_CTRL | R/W | I2C_SLV4 _EN | I2C_SLV4 _INT_EN | I2C_SLV4 _REG_DIS | I2C_MST_DLY[4:0] | | | | |
| 35 | 53 | I2C_SLV4_DI | R | I2C_SLV4_DI[7:0] | | | | | | | |
| 36 | 54 | I2C_MST_STATUS | R | PASS_ THROUGH | I2C_SLV4 _DONE | I2C_LOST _ARB | I2C_SLV4 _NACK | I2C_SLV3 _NACK | I2C_SLV2 _NACK | I2C_SLV1 _NACK | I2C_SLV0 _NACK |
| 37 | 55 | INT_PIN_CFG | R/W | INT_LEVEL | INT_OPEN | LATCH _INT_EN | INT_RD _CLEAR | FSYNC_ INT_LEVEL | FSYNC _INT_EN | I2C _BYPASS _EN | - |
| 38 | 56 | INT_ENABLE | R/W | - | - | - | FIFO _OFLOW _EN | I2C_MST _INT_EN | - | - | DATA _RDY_EN |
| 3A | 58 | INT_STATUS | R | - | - | - | FIFO _OFLOW _INT | I2C_MST _INT | - | - | DATA _RDY_INT |
| 41 | 65 | TEMP_OUT_H | R | TEMP_OUT[15:8] | | | | | | | |
| 42 | 66 | TEMP_OUT_L | R | TEMP_OUT[7:0] | | | | | | | |

| Addr (Hex) | Addr (Dec.) | Register Name | Serial I/F | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 43 | 67 | GYRO_XOUT_H | R | GYRO_XOUT[15:8] | | | | | | | |
| 44 | 68 | GYRO_XOUT_L | R | GYRO_XOUT[7:0] | | | | | | | |
| 45 | 69 | GYRO_YOUT_H | R | GYRO_YOUT[15:8] | | | | | | | |
| 46 | 70 | GYRO_YOUT_L | R | GYRO_YOUT[7:0] | | | | | | | |
| 47 | 71 | GYRO_ZOUT_H | R | GYRO_ZOUT[15:8] | | | | | | | |
| 48 | 72 | GYRO_ZOUT_L | R | GYRO_ZOUT[7:0] | | | | | | | |
| 49 | 73 | EXT_SENS_DATA_00 | R | EXT_SENS_DATA_00[7:0] | | | | | | | |
| 4A | 74 | EXT_SENS_DATA_01 | R | EXT_SENS_DATA_01[7:0] | | | | | | | |
| 4B | 75 | EXT_SENS_DATA_02 | R | EXT_SENS_DATA_02[7:0] | | | | | | | |
| 4C | 76 | EXT_SENS_DATA_03 | R | EXT_SENS_DATA_03[7:0] | | | | | | | |
| 4D | 77 | EXT_SENS_DATA_04 | R | EXT_SENS_DATA_04[7:0] | | | | | | | |
| 4E | 78 | EXT_SENS_DATA_05 | R | EXT_SENS_DATA_05[7:0] | | | | | | | |
| 4F | 79 | EXT_SENS_DATA_06 | R | EXT_SENS_DATA_06[7:0] | | | | | | | |
| 50 | 80 | EXT_SENS_DATA_07 | R | EXT_SENS_DATA_07[7:0] | | | | | | | |
| 51 | 81 | EXT_SENS_DATA_08 | R | EXT_SENS_DATA_08[7:0] | | | | | | | |
| 52 | 82 | EXT_SENS_DATA_09 | R | EXT_SENS_DATA_09[7:0] | | | | | | | |
| 53 | 83 | EXT_SENS_DATA_10 | R | EXT_SENS_DATA_10[7:0] | | | | | | | |
| 54 | 84 | EXT_SENS_DATA_11 | R | EXT_SENS_DATA_11[7:0] | | | | | | | |
| 55 | 85 | EXT_SENS_DATA_12 | R | EXT_SENS_DATA_12[7:0] | | | | | | | |
| 56 | 86 | EXT_SENS_DATA_13 | R | EXT_SENS_DATA_13[7:0] | | | | | | | |
| 57 | 87 | EXT_SENS_DATA_14 | R | EXT_SENS_DATA_14[7:0] | | | | | | | |
| 58 | 88 | EXT_SENS_DATA_15 | R | EXT_SENS_DATA_15[7:0] | | | | | | | |
| 59 | 89 | EXT_SENS_DATA_16 | R | EXT_SENS_DATA_16[7:0] | | | | | | | |
| 5A | 90 | EXT_SENS_DATA_17 | R | EXT_SENS_DATA_17[7:0] | | | | | | | |
| 5B | 91 | EXT_SENS_DATA_18 | R | EXT_SENS_DATA_18[7:0] | | | | | | | |
| 5C | 92 | EXT_SENS_DATA_19 | R | EXT_SENS_DATA_19[7:0] | | | | | | | |
| 5D | 93 | EXT_SENS_DATA_20 | R | EXT_SENS_DATA_20[7:0] | | | | | | | |
| 5E | 94 | EXT_SENS_DATA_21 | R | EXT_SENS_DATA_21[7:0] | | | | | | | |
| 5F | 95 | EXT_SENS_DATA_22 | R | EXT_SENS_DATA_22[7:0] | | | | | | | |
| 60 | 96 | EXT_SENS_DATA_23 | R | EXT_SENS_DATA_23[7:0] | | | | | | | |
| 63 | 99 | I2C_SLV0_DO | R/W | I2C_SLV0_DO[7:0] | | | | | | | |
| 64 | 100 | I2C_SLV1_DO | R/W | I2C_SLV1_DO[7:0] | | | | | | | |
| 65 | 101 | I2C_SLV2_DO | R/W | I2C_SLV2_DO[7:0] | | | | | | | |
| 66 | 102 | I2C_SLV3_DO | R/W | I2C_SLV3_DO[7:0] | | | | | | | |
| 67 | 103 | I2C_MST_DELAY_CTRL | R/W | DELAY_ES_SHADOW | - | - | I2C_SLV4_DLY_EN | I2C_SLV3_DLY_EN | I2C_SLV2_DLY_EN | I2C_SLV1_DLY_EN | I2C_SLV0_DLY_EN |
| 68 | 104 | SIGNAL_PATH_RESET | R/W | - | - | - | - | - | GYRO_RESET | - | TEMP_RESET |
| 6A | 106 | USER_CTRL | R/W | - | FIFO_EN | I2C_MST_EN | I2C_IF_DIS | - | FIFO_RESET | I2C_MST_RESET | SIG_COND_RESET |
| 6B | 107 | PWR_MGMT_1 | R/W | DEVICE_RESET | SLEEP | CYCLE | - | TEMP_DIS | CLKSEL[2:0] | | |
| 6C | 108 | PWR_MGMT_2 | R/W | - | - | - | - | - | STBY_XG | STBY_YG | STBY_ZG |
| 72 | 114 | FIFO_COUNTH | R/W | FIFO_COUNT[15:8] | | | | | | | |

| Addr (Hex) | Addr (Dec.) | Register Name | Serial I/F | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 73 | 115 | FIFO_COUNTL | R/W | FIFO_COUNT[7:0] | | | | | | | |
| 74 | 116 | FIFO_R_W | R/W | FIFO_DATA[7:0] | | | | | | | |
| 75 | 117 | WHO_AM_I | R | - | WHO_AM_I[6:1] | | | | | | - |

Note: Register Names ending in _H and _L contain the high and low bytes, respectively, of an internal register value.

In the detailed register tables that follow, register names are in capital letters, while register values are in capital letters and italicized.

The reset value is 0x00 for all registers other than the registers below.

- Register 13 (SELF_TEST_X): Reset value is non-zero
- Register 14 (SELF_TEST_Y): Reset value is non-zero
- Register 15 (SELF_TEST_Z): Reset value is non-zero
- Register 117: 0x68

# 4 Register Descriptions

This section describes the function and contents of each register within the MPU-3300.

Note: The device will come up in awake mode upon power-up.

## 4.1 Registers 13 to 15 – Self Test Registers SELF_TEST_X, SELF_TEST_Y, and SELF_TEST_Z

**Type: Read/Write**

| Register (Hex) | Register (Decimal) | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|---|---|---|---|---|---|---|---|---|---|
| 0D | 13 | - | | | XG_TEST[4-0] | | | | |
| 0E | 14 | - | | | YG_TEST[4-0] | | | | |
| 0F | 15 | - | | | ZG_TEST[4-0] | | | | |

**Description:**

These registers are used for gyroscope self-tests that permit the user to test the mechanical and electrical portions of the gyroscope. The following sections describe the self-test process.

**1. Gyroscope Hardware Self-Test: Relative Method**

Gyroscope self-test permits users to test the mechanical and electrical portions of the gyroscope. Code for operating self-test is included within the MotionApps™ software provided by InvenSense. Please refer to the next section (*Obtaining the Gyroscope Factory Trim (FT) Value*) if not using MotionApps software.

When self-test is activated, the on-board electronics will actuate the appropriate sensor. This actuation will move the sensor's proof masses over a distance equivalent to a pre-defined Coriolis force. This proof mass displacement results in a change in the sensor output, which is reflected in the output signal. The output signal is used to observe the self-test response.

The self-test response (STR) is defined as follows:

$$SelfTest\ Response = Gyroscope\ Output\ with\ Self\text{-}Test\ Enabled - Gyroscope\ Output\ with\ Self\text{-}Test\ Disabled$$

This self test-response is used to determine whether the part has passed or failed self-test by finding the change from factory trim of the self-test response as follows:

$$Change\ from\ Factory\ Trim\ of\ the\ Self\text{-}Test\ Response(\%) = \frac{(STR - FT)}{FT}$$

where,

$FT = Factory\ trim\ value\ of\ selftest\ response, available\ via\ MotionApps\ software$

This change from factory trim of the self-test response must be within the limits provided in the MPU-3300 Product Specification document for the part to pass self-test. Otherwise, the part is deemed to have failed self-test.

Obtaining the Gyroscope Factory Trim (FT) Value

If InvenSense MotionApps software is not used, the procedure detailed below should be followed to obtain the Factory trim value of the self test response (FT) mentioned above. For the specific registers mentioned below, please refer to registers 13-15.

The Factory trim value of the self test response (FT) is calculated as shown below. FT[Xg], FT[Yg], and FT[Zg] refer to the factory trim (FT) values for the gyroscope X, Y, and Z axes, respectively. XG_TEST is the decimal version of XG_TEST[4-0], YG_TEST is the decimal version of YG_TEST[4-0], and ZG_TEST is the decimal version of ZG_TEST[4-0].

**When performing self test for the gyroscope, the full-scale range should be set to ±225dps.**

$$\begin{cases} FT[Xg] = 25 * 145.6 * 1.046^{(XG\_TEST-1)} & \text{if } XG\_TEST \neq 0 \\ FT[Xg] = 0 & \text{if } XG\_TEST = 0 \end{cases}$$

$$\begin{cases} FT[Yg] = -25 * 145.6 * 1.046^{(YG\_TEST-1)} & \text{if } YG\_TEST \neq 0 \\ FT[Yg] = 0 & \text{if } YG\_TEST = 0 \end{cases}$$

$$\begin{cases} FT[Zg] = 25 * 145.6 * 1.046^{(ZG\_TEST-1)} & \text{if } ZG\_TEST \neq 0 \\ FT[Zg] = 0 & \text{if } ZG\_TEST = 0 \end{cases}$$

**Parameters:**

| | |
|---|---|
| *XG_TEST* | 5-bit unsigned value. FT[Xg] is determined by using this value as explained above. |
| *YG_TEST* | 5-bit unsigned value. FT[Yg] is determined by using this value as explained above. |
| *ZG_TEST* | 5-bit unsigned value. FT[Zg] is determined by using this value as explained above. |

Bits 7 through 5 are reserved.

## 4.2 Register 25 – Sample Rate Divider SMPRT_DIV

**Type: Read/Write**

| Register (Hex) | Register (Decimal) | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|---|---|---|---|---|---|---|---|---|---|
| 19 | 25 | SMPLRT_DIV[7:0] | | | | | | | |

**Description:**

This register specifies the divider from the gyroscope output rate used to generate the Sample Rate for the MPU-3300.

The sensor register output, and FIFO output sampling are based on the Sample Rate.

The Sample Rate is generated by dividing the gyroscope output rate by *SMPLRT_DIV*:

Sample Rate = Gyroscope Output Rate / (1 + *SMPLRT_DIV*) where Gyroscope Output Rate = 8kHz when the DLPF is disabled (*DLPF_CFG* = 0 or 7), and 1kHz when the DLPF is enabled (see Register 26).

**Parameters:**

| | |
|---|---|
| *SMPLRT_DIV* | 8-bit unsigned value. The Sample Rate is determined by dividing the gyroscope output rate by this value. |

## 4.3 Register 26 – Configuration
**CONFIG**

**Type: Read/Write**

| Register (Hex) | Register (Decimal) | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|---|---|---|---|---|---|---|---|---|---|
| 1A | 26 | - | - | EXT_SYNC_SET[2:0] | | | DLPF_CFG[2:0] | | |

**Description:**

This register configures the external Frame Synchronization (FSYNC) pin sampling and the Digital Low Pass Filter (DLPF) setting for the gyroscopes.

An external signal connected to the FSYNC pin can be sampled by configuring *EXT_SYNC_SET*.

Signal changes to the FSYNC pin are latched so that short strobes may be captured. The latched FSYNC signal will be sampled at the Sampling Rate, as defined in register 25. After sampling, the latch will reset to the current FSYNC signal state.

The sampled value will be reported in place of the least significant bit in a sensor data register determined by the value of *EXT_SYNC_SET* according to the following table.

| EXT_SYNC_SET | FSYNC Bit Location |
|---|---|
| 0 | Input disabled |
| 1 | TEMP_OUT_L[0] |
| 2 | GYRO_XOUT_L[0] |
| 3 | GYRO_YOUT_L[0] |
| 4 | GYRO_ZOUT_L[0] |

The DLPF is configured by *DLPF_CFG*. The gyroscope data is filtered according to the value of *DLPF_CFG* as shown in the table below.

| DLPF_CFG | Gyroscope | | |
|---|---|---|---|
| | Bandwidth (Hz) | Delay (ms) | Fs (kHz) |
| 0 | 256 | 0.98 | 8 |
| 1 | 188 | 1.9 | 1 |
| 2 | 98 | 2.8 | 1 |
| 3 | 42 | 4.8 | 1 |
| 4 | 20 | 8.3 | 1 |
| 5 | 10 | 13.4 | 1 |
| 6 | 5 | 18.6 | 1 |
| 7 | RESERVED | | 8 |

Bits 7 and 6 are reserved.

**Parameters:**

*EXT_SYNC_SET* 3-bit unsigned value. Configures the FSYNC pin sampling.
*DLPF_CFG* 3-bit unsigned value. Configures the DLPF setting.

## 4.4 Register 27 – Gyroscope Configuration GYRO_CONFIG

**Type: Read/Write**

| Register (Hex) | Register (Decimal) | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|---|---|---|---|---|---|---|---|---|---|
| 1B | 27 | XG_ST | YG_ST | ZG_ST | FS_SEL[1:0] | | - | - | - |

**Description:**

This register is used to trigger gyroscope self-test and configure the gyroscopes' full scale range.

Gyroscope self-test permits users to test the mechanical and electrical portions of the gyroscope. The self-test for each gyroscope axis can be activated by controlling the *XG_ST*, *YG_ST*, and *ZG_ST* bits of this register. Self-test for each axis may be performed independently or all at the same time.

When self-test is activated, the on-board electronics will actuate the appropriate sensor. This actuation will move the sensor's proof masses over a distance equivalent to a pre-defined Coriolis force. This proof mass displacement results in a change in the sensor output, which is reflected in the output signal. The output signal is used to observe the self-test response.

The self-test response is defined as follows:

Self-test response = Sensor output with self-test enabled – Sensor output without self-test enabled

The self-test limits for each gyroscope axis is provided in the electrical characteristics tables of the MPU-3300 Product Specification document. When the value of the self-test response is within the min/max limits of the product specification, the part has passed self test. When the self-test response exceeds the min/max values specified in the document, the part is deemed to have failed self-test.

*FS_SEL* selects the full scale range of the gyroscope outputs according to the following table.

| FS_SEL | Full Scale Range |
|---|---|
| 0 | ± 225 °/s |
| 1 | ± 450 °/s |

Bits 2 through 0 are reserved.

**Parameters:**

| | |
|---|---|
| *XG_ST* | Setting this bit causes the X axis gyroscope to perform self test. |
| *YG_ST* | Setting this bit causes the Y axis gyroscope to perform self test. |
| *ZG_ST* | Setting this bit causes the Z axis gyroscope to perform self test. |
| *FS_SEL* | 2-bit unsigned value. Selects the full scale range of gyroscopes. |

## 4.5 Register 35 – FIFO Enable
**FIFO_EN**

**Type: Read/Write**

| Register (Hex) | Register (Decimal) | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|---|---|---|---|---|---|---|---|---|---|
| 23 | 35 | TEMP_ FIFO_EN | XG_ FIFO_EN | YG_ FIFO_EN | ZG_ FIFO_EN | - | SLV2 _FIFO_EN | SLV1 _FIFO_EN | SLV0 _FIFO_EN |

**Description:**

This register determines which sensor measurements are loaded into the FIFO buffer.

Data stored inside the sensor data registers (Registers 65 to 96) will be loaded into the FIFO buffer if a sensor's respective FIFO_EN bit is set to 1 in this register.

When a sensor's FIFO_EN bit is enabled in this register, data from the sensor data registers will be loaded into the FIFO buffer. The sensors are sampled at the Sample Rate as defined in Register 25. For further information regarding sensor data registers, please refer to Registers 65 to 96

When an external Slave's corresponding FIFO_EN bit (*SLVx_FIFO_EN,* where x=0, 1, or 2) is set to 1, the data stored in its corresponding data registers (EXT_SENS_DATA registers, Registers 73 to 96) will be written into the FIFO buffer at the Sample Rate. EXT_SENS_DATA register association with $I^2C$ Slaves is determined by the I2C_SLVx_CTRL registers (where x=0, 1, or 2; Registers 39, 42, and 45). For information regarding EXT_SENS_DATA registers, please refer to Registers 73 to 96.

Note that the corresponding FIFO_EN bit (*SLV3_FIFO_EN)* is found in I2C_MST_CTRL (Register 36). Also note that Slave 4 behaves in a different manner compared to Slaves 0-3. Please refer to Registers 49 to 53 for further information regarding Slave 4 usage.

**Parameters:**

| | |
|---|---|
| *TEMP_FIFO_EN* | When set to 1, this bit enables TEMP_OUT_H and TEMP_OUT_L (Registers 65 and 66) to be written into the FIFO buffer. |
| *XG_ FIFO_EN* | When set to 1, this bit enables GYRO_XOUT_H and GYRO_XOUT_L (Registers 67 and 68) to be written into the FIFO buffer. |
| *YG_ FIFO_EN* | When set to 1, this bit enables GYRO_YOUT_H and GYRO_YOUT_L (Registers 69 and 70) to be written into the FIFO buffer. |
| *ZG_ FIFO_EN* | When set to 1, this bit enables GYRO_ZOUT_H and GYRO_ZOUT_L (Registers 71 and 72) to be written into the FIFO buffer. |
| *SLV2_ FIFO_EN* | When set to 1, this bit enables EXT_SENS_DATA registers (Registers 73 to 96) associated with Slave 2 to be written into the FIFO buffer. |
| *SLV1_ FIFO_EN* | When set to 1, this bit enables EXT_SENS_DATA registers (Registers 73 to 96) associated with Slave 1 to be written into the FIFO buffer. |
| *SLV0_ FIFO_EN* | When set to 1, this bit enables EXT_SENS_DATA registers (Registers 73 to 96) associated with Slave 0 to be written into the FIFO buffer. |

<u>Note</u>: For further information regarding the association of EXT_SENS_DATA registers to particular slave devices, please refer to Registers 73 to 96.

Bit 3 is reserved.

## 4.6 Register 36 – I²C Master Control
**I2C_MST_CTRL**

**Type: Read/Write**

| Register (Hex) | Register (Decimal) | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|---|---|---|---|---|---|---|---|---|---|
| 24 | 36 | MULT _MST_EN | WAIT _FOR_ES | SLV_3 _FIFO_EN | I2C_MST _P_NSR | I2C_MST_CLK[3:0] | | | |

**Description:**

This register configures the auxiliary I²C bus for single-master or multi-master control. In addition, the register is used to delay the Data Ready interrupt, and also enables the writing of Slave 3 data into the FIFO buffer. The register also configures the auxiliary I²C Master's transition from one slave read to the next, as well as the MPU-3300's 8MHz internal clock.

Multi-master capability allows multiple I²C masters to operate on the same bus. In circuits where multi-master capability is required, set *MULT_MST_EN* to 1. This will increase current drawn by approximately 30µA.

In circuits where multi-master capability is required, the state of the I²C bus must always be monitored by each separate I²C Master. Before an I²C Master can assume arbitration of the bus, it must first confirm that no other I²C Master has arbitration of the bus. When *MULT_MST_EN* is set to 1, the MPU-3300's bus arbitration detection logic is turned on, enabling it to detect when the bus is available.

When the *WAIT_FOR_ES* bit is set to 1, the Data Ready interrupt will be delayed until External Sensor data from the Slave Devices are loaded into the EXT_SENS_DATA registers. This is used to ensure that both the internal sensor data (i.e. from gyro) and external sensor data have been loaded to their respective data registers (i.e. the data is synced) when the Data Ready interrupt is triggered.

When the Slave 3 FIFO enable bit (*SLV_3_FIFO_EN*) is set to 1, Slave 3 sensor measurement data will be loaded into the FIFO buffer each time. EXT_SENS_DATA register association with I²C Slaves is determined by I2C_SLV3_CTRL (Register 48).

For further information regarding EXT_SENS_DATA registers, please refer to Registers 73 to 96.

The corresponding FIFO_EN bits for Slave 0, Slave 1, and Slave 2 can be found in Register 35.

The *I2C_MST_P_NSR* bit configures the I²C Master's transition from one slave read to the next slave read. If the bit equals 0, there will be a restart between reads. If the bit equals 1, there will be a stop followed by a start of the following read. When a write transaction follows a read transaction, the stop followed by a start of the successive write will be always used.

*I2C_MST_CLK* is a 4 bit unsigned value which configures a divider on the MPU-3300 internal 8MHz clock. It sets the I²C master clock speed according to the following table:

| I2C_MST_CLK | I²C Master Clock Speed | 8MHz Clock Divider |
|---|---|---|
| 0 | 348 kHz | 23 |
| 1 | 333 kHz | 24 |
| 2 | 320 kHz | 25 |
| 3 | 308 kHz | 26 |
| 4 | 296 kHz | 27 |
| 5 | 286 kHz | 28 |
| 6 | 276 kHz | 29 |
| 7 | 267 kHz | 30 |
| 8 | 258 kHz | 31 |
| 9 | 500 kHz | 16 |
| 10 | 471 kHz | 17 |
| 11 | 444 kHz | 18 |
| 12 | 421 kHz | 19 |
| 13 | 400 kHz | 20 |
| 14 | 381 kHz | 21 |
| 15 | 364 kHz | 22 |

**Parameters:**

| | |
|---|---|
| *MUL_MST_EN* | When set to 1, this bit enables multi-master capability. |
| *WAIT_FOR_ES* | When set to 1, this bit delays the Data Ready interrupt until External Sensor data from the Slave devices have been loaded into the EXT_SENS_DATA registers. |
| *SLV3_FIFO_EN* | When set to 1, this bit enables EXT_SENS_DATA registers associated with Slave 3 to be written into the FIFO. The corresponding bits for Slaves 0-2 can be found in Register 35. |
| *I2C_MST_P_NSR* | Controls the I²C Master's transition from one slave read to the next slave read.<br>When this bit equals 0, there is a restart between reads.<br>When this bit equals 1, there is a stop and start marking the beginning of the next read.<br><br>When a write follows a read, a stop and start is always enforced. |
| *I2C_MST_CLK* | 4 bit unsigned value. Configures the I²C master clock speed divider. |

Note: For further information regarding the association of EXT_SENS_DATA registers to particular slave devices, please refer to Registers 73 to 96.

## 4.7 Registers 37 to 39 – I$^2$C Slave 0 Control
## I2C_SLV0_ADDR, I2C_SLV0_REG, and I2C_SLV0_CTRL

**Type: Read/Write**

| Register (Hex) | Register (Decimal) | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|---|---|---|---|---|---|---|---|---|---|
| 25 | 37 | I2C_SLV0_RW | I2C_SLV0_ADDR[6:0] | | | | | | |
| 26 | 38 | I2C_SLV0_REG[7:0] | | | | | | | |
| 27 | 39 | I2C_SLV0_EN | I2C_SLV0_BYTE_SW | I2C_SLV0_REG_DIS | I2C_SLV0_GRP | I2C_SLV0_LEN[3:0] | | | |

<u>**Description:**</u>

These registers configure the data transfer sequence for Slave 0. Slaves 1, 2, and 3 also behave in a similar manner to Slave 0. However, Slave 4's characteristics differ greatly from those of Slaves 0-3. For further information regarding Slave 4, please refer to registers 49 to 53.

I$^2$C slave data transactions between the MPU-3300 and Slave 0 are set as either read or write operations by the *I2C_SLV0_RW* bit. When this bit is 1, the transfer is a read operation. When the bit is 0, the transfer is a write operation.

*I2C_SLV0_ADDR* is used to specify the I$^2$C slave address of Slave 0.

Data transfer starts at an internal register within Slave 0. This address of this register is specified by *I2C_SLV0_REG*.

The number of bytes transferred is specified by *I2C_SLV0_LEN.* When more than 1 byte is transferred (*I2C_SLV0_LEN* > 1), data is read from (written to) sequential addresses starting from *I2C_SLV0_REG*.

In read mode, the result of the read is placed in the lowest available EXT_SENS_DATA register. For further information regarding the allocation of read results, please refer to the EXT_SENS_DATA register description (Registers 73 – 96).

In write mode, the contents of *I2C_SLV0_DO* (Register 99) will be written to the slave device.

*I2C_SLV0_EN* enables Slave 0 for I$^2$C data transaction. A data transaction is performed only if more than zero bytes are to be transferred (*I2C_SLV0_LEN* > 0) between an enabled slave device (*I2C_SLV0_EN* = 1).

*I2C_SLV0_BYTE_SW* configures byte swapping of word pairs. When byte swapping is enabled, the high and low bytes of a word pair are swapped. Please refer to *I2C_SLV0_GRP* for the pairing convention of the word pairs. When this bit is cleared to 0, bytes transferred to and from Slave 0 will be written to EXT_SENS_DATA registers in the order they were transferred.

When *I2C_SLV0_REG_DIS* is set to 1, the transaction will read or write data only. When cleared to 0, the transaction will write a register address prior to reading or writing data. This bit should equal 0 when specifying the register address within the Slave device to/from which the ensuing data transaction will take place.

*I2C_SLV0_GRP* specifies the grouping order of word pairs received from registers. When cleared to 0, bytes from register addresses 0 and 1, 2 and 3, etc (even, then odd register addresses) are paired to form a word. When set to 1, bytes from register addresses are paired 1 and 2, 3 and 4, etc. (odd, then even register addresses) are paired to form a word.

I$^2$C data transactions are performed at the Sample Rate, as defined in Register 25. The user is responsible for ensuring that I$^2$C data transactions to and from each enabled Slave can be completed within a single period of the Sample Rate.

The I$^2$C slave access rate can be reduced relative to the Sample Rate. This reduced access rate is determined by *I2C_MST_DLY* (Register 52). Whether a slave's access rate is reduced relative to the Sample Rate is determined by I2C_MST_DELAY_CTRL (Register 103).

The processing order for the slaves is fixed. The sequence followed for processing the slaves is Slave 0, Slave 1, Slave 2, Slave 3 and Slave 4. If a particular Slave is disabled it will be skipped.

Each slave can either be accessed at the sample rate or at a reduced sample rate. In a case where some slaves are accessed at the Sample Rate and some slaves are accessed at the reduced rate, the sequence of accessing the slaves (Slave 0 to Slave 4) is still followed. However, the reduced rate slaves will be skipped if their access rate dictates that they should not be accessed during that particular cycle. For further information regarding the reduced access rate, please refer to Register 52. Whether a slave is accessed at the Sample Rate or at the reduced rate is determined by the Delay Enable bits in Register 103.

**Parameters:**

| | |
|---|---|
| *I2C_SLV0_RW* | When set to 1, this bit configures the data transfer as a read operation.<br>When cleared to 0, this bit configures the data transfer as a write operation. |
| *I2C_SLV0_ADDR* | 7-bit I$^2$C address of Slave 0. |
| *I2C_SLV0_REG* | 8-bit address of the Slave 0 register to/from which data transfer starts. |
| *I2C_SLV0_EN* | When set to 1, this bit enables Slave 0 for data transfer operations.<br>When cleared to 0, this bit disables Slave 0 from data transfer operations. |
| *I2C_SLV0_BYTE_SW* | When set to 1, this bit enables byte swapping. When byte swapping is enabled, the high and low bytes of a word pair are swapped. Please refer to *I2C_SLV0_GRP* for the pairing convention of the word pairs.<br>When cleared to 0, bytes transferred to and from Slave 0 will be written to EXT_SENS_DATA registers in the order they were transferred. |
| *I2C_SLV0_REG_DIS* | When set to 1, the transaction will read or write data only.<br>When cleared to 0, the transaction will write a register address prior to reading or writing data. |
| *I2C_SLV0_GRP* | 1-bit value specifying the grouping order of word pairs received from registers. When cleared to 0, bytes from register addresses 0 and 1, 2 and 3, etc (even, then odd register addresses) are paired to form a word. When set to 1, bytes from register addresses are paired 1 and 2, 3 and 4, etc. (odd, then even register addresses) are paired to form a word. |
| *I2C_SLV0_LEN* | 4-bit unsigned value. Specifies the number of bytes transferred to and from Slave 0.<br>Clearing this bit to 0 is equivalent to disabling the register by writing 0 to *I2C_SLV0_EN*. |

**Byte Swapping Example**

The following example demonstrates byte swapping for *I2C_SLV0_BYTE_SW* = 1, *I2C_SLV0_GRP* = 0, *I2C_SLV0_REG* = 0x01, and *I2C_SLV0_LEN* = 0x4:

1. The first byte, read from Slave 0 register 0x01, will be stored at EXT_SENS_DATA_00. Because *I2C_SLV0_GRP* = 0, bytes from even, then odd register addresses will be paired together as word pairs. Since the read operation started from an odd register address instead of an even address, only one byte is read.
2. The second and third bytes will be swapped, since *I2C_SLV0_BYTE_SW* = 1 and *I2C_SLV0_REG[0]* = 1. The data read from 0x02 will be stored at EXT_SENS_DATA_02, and the data read from 0x03 will be stored at EXT_SENS_DATA_01.
3. The last byte, read from address 0x04, will be stored at EXT_SENS_DATA_03. Because there is only one byte remaining in the read operation, byte swapping will not occur.

**Slave Access Example**

Slave 0 is accessed at the Sample Rate, while Slave 1 is accessed at half the Sample Rate. The other slaves are disabled. In the first cycle, both Slave 0 and Slave 1 will be accessed. However, in the second cycle, only Slave 0 will be accessed. In the third cycle, both Slave 0 and Slave 1 will be accessed. In the fourth cycle, only Slave 0 will be accessed. This pattern continues.

## 4.8 Registers 40 to 42 – I²C Slave 1 Control I2C_SLV1_ADDR, I2C_SLV1_REG, and I2C_SLV1_CTRL

**Type: Read/Write**

| Register (Hex) | Register (Decimal) | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|---|---|---|---|---|---|---|---|---|---|
| 28 | 40 | I2C_SLV1_RW | I2C_SLV1_ADDR[6:0] | | | | | | |
| 29 | 41 | I2C_SLV1_REG[7:0] | | | | | | | |
| 2A | 42 | I2C_SLV1_EN | I2C_SLV1_BYTE_SW | I2C_SLV1_REG_DIS | I2C_SLV1_GRP | I2C_SLV1_LEN[3:0] | | | |

**Description:**

These registers describe the data transfer sequence for Slave 1. Their functions correspond to those described for the Slave 0 registers (Registers 37 to 39).

## 4.9 Registers 43 to 45 – I²C Slave 2 Control I2C_SLV2_ADDR, I2C_SLV2_REG, and I2C_SLV2_CTRL

**Type: Read/Write**

| Register (Hex) | Register (Decimal) | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|---|---|---|---|---|---|---|---|---|---|
| 2B | 43 | I2C_SLV2_RW | I2C_SLV2_ADDR[6:0] | | | | | | |
| 2C | 44 | I2C_SLV2_REG[7:0] | | | | | | | |
| 2D | 45 | I2C_SLV2_EN | I2C_SLV2_BYTE_SW | I2C_SLV2_REG_DIS | I2C_SLV2_GRP | I2C_SLV2_LEN[3:0] | | | |

**Description:**

These registers describe the data transfer sequence for Slave 2. Their functions correspond to those described for the Slave 0 registers (Registers 37 to 39).

## 4.10 Registers 46 to 48 – I²C Slave 3 Control I2C_SLV3_ADDR, I2C_SLV3_REG, and I2C_SLV3_CTRL

**Type: Read/Write**

| Register (Hex) | Register (Decimal) | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|---|---|---|---|---|---|---|---|---|---|
| 2E | 46 | I2C_SLV3_RW | I2C_SLV3_ADDR[6:0] | | | | | | |
| 2F | 47 | I2C_SLV3_REG[7:0] | | | | | | | |
| 30 | 48 | I2C_SLV3_EN | I2C_SLV3_BYTE_SW | I2C_SLV3_REG_DIS | I2C_SLV3_GRP | I2C_SLV3_LEN[3:0] | | | |

**Description:**

These registers describe the data transfer sequence for Slave 3. Their functions correspond to those described for the Slave 0 registers (Registers 37 to 39).

## 4.11 Registers 49 to 53 – $I^2C$ Slave 4 Control
## I2C_SLV4_ADDR, I2C_SLV4_REG, I2C_SLV4_DO, I2C_SLV4_CTRL, and I2C_SLV4_DI

**Type: Read/Write**

| Register (Hex) | Register (Decimal) | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|---|---|---|---|---|---|---|---|---|---|
| 31 | 49 | I2C_SLV4_RW | I2C_SLV4_ADDR[6:0] | | | | | | |
| 32 | 50 | I2C_SLV4_REG[7:0] | | | | | | | |
| 33 | 51 | I2C_SLV4_DO[7:0] | | | | | | | |
| 34 | 52 | I2C_SLV4_EN | I2C_SLV4_INT_EN | I2C_SLV4_REG_DIS | I2C_MST_DLY[4:0] | | | | |
| 35 | 53 | I2C_SLV4_DI[7:0] | | | | | | | |

**Description:**

These registers describe the data transfer sequence for Slave 4. The characteristics of Slave 4 differ greatly from those of Slaves 0-3. For further information regarding the characteristics of Slaves 0-3, please refer to Registers 37 to 48.

$I^2C$ slave data transactions between the MPU-3300 and Slave 4 are set as either read or write operations by the *I2C_SLV4_RW* bit. When this bit is 1, the transfer is a read operation. When the bit is 0, the transfer is a write operation.

*I2C_SLV4_ADDR* is used to specify the $I^2C$ slave address of Slave 4.

Data transfer starts at an internal register within Slave 4. This register address is specified by *I2C_SLV4_REG*.

In read mode, the result of the read will be available in *I2C_SLV4_DI*. In write mode, the contents of *I2C_SLV4_DO* will be written into the slave device.

A data transaction is performed only if the *I2C_SLV4_EN* bit is set to 1. The data transaction should be enabled once its parameters are configured in the _ADDR and _REG registers. For write, the _DO register is also required. *I2C_SLV4_EN* will be cleared after the transaction is performed once.

An interrupt is triggered at the completion of a Slave 4 data transaction if the interrupt is enabled . The status of this interrupt can be observed in Register 54.

When *I2C_SLV4_REG_DIS* is set to 1, the transaction will read or write data instead of writing a register address. This bit should equal 0 when specifying the register address within the Slave device to/from which the ensuing data transaction will take place.

*I2C_MST_DLY* configures the reduced access rate of $I^2C$ slaves relative to the Sample Rate. When a slave's access rate is decreased relative to the Sample Rate, the slave is accessed every

1 / (1 + *I2C_MST_DLY*) samples

This base Sample Rate in turn is determined by *SMPLRT_DIV* (register 25) and *DLPF_CFG* (register 26). Whether a slave's access rate is reduced relative to the Sample Rate is determined by I2C_MST_DELAY_CTRL (register 103).

For further information regarding the Sample Rate, please refer to register 25.

Slave 4 transactions are performed after Slave 0, 1, 2 and 3 transactions have been completed. Thus the maximum rate for Slave 4 transactions is determined by the Sample Rate as defined in Register 25.

**Parameters:**

| | |
|---|---|
| *I2C_SLV4_RW* | When set to 1, this bit configures the data transfer as a read operation. When cleared to 0, this bit configures the data transfer as a write operation. |
| *I2C_SLV4_ADDR* | 7-bit $I^2C$ address for Slave 4. |
| *I2C_SLV4_REG* | 8-bit address of the Slave 4 register to/from which data transfer starts. |
| *I2C_SLV4_DO* | This register stores the data to be written into the Slave 4. If *I2C_SLV4_RW* is set 1 (set to read), this register has no effect. |
| *I2C_SLV4_EN* | When set to 1, this bit enables Slave 4 for data transfer operations. When cleared to 0, this bit disables Slave 4 from data transfer operations. |
| *I2C_SLV4_INT_EN* | When set to 1, this bit enables the generation of an interrupt signal upon completion of a Slave 4 transaction. When cleared to 0, this bit disables the generation of an interrupt signal upon completion of a Slave 4 transaction. The interrupt status can be observed in Register 54. |
| *I2C_SLV4_REG_DIS* | When set to 1, the transaction will read or write data. When cleared to 0, the transaction will read or write a register address. |
| *I2C_MST_DLY* | Configures the decreased access rate of slave devices relative to the Sample Rate. |
| *I2C_SLV4_DI* | This register stores the data read from Slave 4. This field is populated after a read transaction. |

## 4.12 Register 54 – I²C Master Status I2C_MST_STATUS

**Type: Read Only**

| Register (Hex) | Register (Decimal) | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|---|---|---|---|---|---|---|---|---|---|
| 36 | 54 | PASS_ THROUGH | I2C_SLV4 _DONE | I2C_LOST _ARB | I2C_SLV4 _NACK | I2C_SLV3 _NACK | I2C_SLV2 _NACK | I2C_SLV1 _NACK | I2C_SLV0 _NACK |

**Description:**

This register shows the status of the interrupt generating signals in the I²C Master within the MPU-3300. This register also communicates the status of the FSYNC interrupt to the host processor.

Reading this register will clear all the status bits in the register.

**Parameters:**

*PASS_THROUGH* This bit reflects the status of the FSYNC interrupt from an external device into the MPU-3300. This is used as a way to pass an external interrupt through the MPU-3300 to the host application processor. If enabled in the INT_PIN_CFG register by asserting bit FSYNC_INT_EN, this will cause an interrupt.

*I2C_SLV4_DONE* Automatically sets to 1 when a Slave 4 transaction has completed. This triggers an interrupt if the *I2C_MST_INT_EN* bit in the INT_ENABLE register (Register 56) is asserted and if the *SLV_4_DONE_INT* bit is asserted in the I2C_SLV4_CTRL register (Register 52).

*I2C_LOST_ARB* This bit automatically sets to 1 when the I²C Master has lost arbitration of the auxiliary I²C bus (an error condition). This triggers an interrupt if the *I2C_MST_INT_EN* bit in the INT_ENABLE register (Register 56) is asserted.

*I2C_SLV4_NACK* This bit automatically sets to 1 when the I²C Master receives a NACK in a transaction with Slave 4. This triggers an interrupt if the *I2C_MST_INT_EN* bit in the INT_ENABLE register (Register 56) is asserted.

*I2C_SLV3_NACK* This bit automatically sets to 1 when the I²C Master receives a NACK in a transaction with Slave 3. This triggers an interrupt if the *I2C_MST_INT_EN* bit in the INT_ENABLE register (Register 56) is asserted.

*I2C_SLV2_NACK* This bit automatically sets to 1 when the I²C Master receives a NACK in a transaction with Slave 2. This triggers an interrupt if the *I2C_MST_INT_EN* bit in the INT_ENABLE register (Register 56) is asserted.

*I2C_SLV1_NACK* This bit automatically sets to 1 when the I²C Master receives a NACK in a transaction with Slave 1. This triggers an interrupt if the *I2C_MST_INT_EN* bit in the INT_ENABLE register (Register 56) is asserted.

*I2C_SLV0_NACK* This bit automatically sets to 1 when the I²C Master receives a NACK in a transaction with Slave 0. This triggers an interrupt if the *I2C_MST_INT_EN* bit in the INT_ENABLE register (Register 56) is asserted.

## 4.13 Register 55 – INT Pin / Bypass Enable Configuration INT_PIN_CFG

**Type: Read/Write**

| Register (Hex) | Register (Decimal) | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|---|---|---|---|---|---|---|---|---|---|
| 37 | 55 | INT_LEVEL | INT_OPEN | LATCH _INT_EN | INT_RD _CLEAR | FSYNC_ INT_LEVEL | FSYNC_ INT_EN | I2C _BYPASS _EN | - |

**Description:**

This register configures the behavior of the interrupt signals at the INT pins. This register is also used to enable the FSYNC Pin to be used as an interrupt to the host application processor, as well as to enable Bypass Mode on the $I^2C$ Master.

*FSYNC_INT_EN* enables the FSYNC pin to be used as an interrupt to the host application processor. A transition to the active level specified in *FSYNC_INT_LEVEL* will trigger an interrupt. The status of this interrupt is read from the *PASS_THROUGH* bit in the $I^2C$ Master Status Register (Register 54).

When *I2C_BYPASS_EN* is equal to 1 and *I2C_MST_EN* (Register 106 bit[5]) is equal to 0, the host application processor will be able to directly access the auxiliary $I^2C$ bus of the MPU-3300. When this bit is equal to 0, the host application processor will not be able to directly access the auxiliary $I^2C$ bus of the MPU-3300 regardless of the state of *I2C_MST_EN*.

For further information regarding Bypass Mode, please refer to the MPU-3300 Product Specification document.

Bit 0 is reserved.

**Parameters:**

*INT_LEVEL* — When this bit is equal to 0, the logic level for the INT pin is active high.
When this bit is equal to 1, the logic level for the INT pin is active low.

*INT_OPEN* — When this bit is equal to 0, the INT pin is configured as push-pull.
When this bit is equal to 1, the INT pin is configured as open drain.

*LATCH_INT_EN* — When this bit is equal to 0, the INT pin emits a 50us long pulse.
When this bit is equal to 1, the INT pin is held high until the interrupt is cleared.

*INT_RD_CLEAR* — When this bit is equal to 0, interrupt status bits are cleared only by reading INT_STATUS (Register 58)

When this bit is equal to 1, interrupt status bits are cleared on any read operation.

*FSYNC_INT_LEVEL* — When this bit is equal to 0, the logic level for the FSYNC pin (when used as an interrupt to the host processor) is active high.
When this bit is equal to 1, the logic level for the FSYNC pin (when used as an interrupt to the host processor) is active low.

*FSYNC_INT_EN* — When equal to 0, this bit disables the FSYNC pin from causing an interrupt to the host processor.

When equal to 1, this bit enables the FSYNC pin to be used as an interrupt to the host processor.

*I2C_BYPASS_EN* When this bit is equal to 1 and *I2C_MST_EN* (Register 106 bit[5]) is equal to 0, the host application processor will be able to directly access the auxiliary $I^2C$ bus of the MPU-3300.

When this bit is equal to 0, the host application processor will not be able to directly access the auxiliary $I^2C$ bus of the MPU-3300 regardless of the state of *I2C_MST_EN (*Register 106 bit[5]).

## 4.14 Register 56 – Interrupt Enable

**INT_ENABLE**

**Type: Read/Write**

| Register (Hex) | Register (Decimal) | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|---|---|---|---|---|---|---|---|---|---|
| 38 | 56 | - | - | - | FIFO _OFLOW _EN | I2C_MST _INT_EN | - | - | DATA _RDY_EN |

**Description:**

This register enables interrupt generation by interrupt sources.

For information regarding the interrupt status for each interrupt generation source, please refer to Register 58. Further information regarding $I^2C$ Master interrupt generation can be found in Register 54.

Bits 7, 6, 5, 2 and 1 are reserved.

**Parameters:**

*FIFO_OFLOW_EN* When set to 1, this bit enables a FIFO buffer overflow to generate an interrupt.

*I2C_MST_INT_EN* When set to 1, this bit enables any of the $I^2C$ Master interrupt sources to generate an interrupt.

*DATA_RDY_EN* When set to 1, this bit enables the Data Ready interrupt, which occurs each time a write operation to all of the sensor registers has been completed.

### 4.15 Register 58 – Interrupt Status
### INT_STATUS

**Type: Read Only**

| Register (Hex) | Register (Decimal) | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|---|---|---|---|---|---|---|---|---|---|
| 3A | 58 | - | - | - | FIFO _OFLOW _INT | I2C_MST _INT | - | - | DATA _RDY_INT |

**Description:**

This register shows the interrupt status of each interrupt generation source. Each bit will clear after the register is read.

For information regarding the corresponding interrupt enable bits, please refer to Register 56.

For a list of $I^2C$ Master interrupts, please refer to Register 54.

Bits 7, 6, 5, 2 and 1 are reserved.

**Parameters:**

*FIFO_OFLOW_INT* — This bit automatically sets to 1 when a FIFO buffer overflow interrupt has been generated.

The bit clears to 0 after the register has been read.

*I2C_MST_INT* — This bit automatically sets to 1 when an $I^2C$ Master interrupt has been generated. For a list of $I^2C$ Master interrupts, please refer to Register 54.

The bit clears to 0 after the register has been read.

*DATA_RDY_INT* — This bit automatically sets to 1 when a Data Ready interrupt is generated.

The bit clears to 0 after the register has been read.

## 4.16 Registers 65 and 66 – Temperature Measurement TEMP_OUT_H and TEMP_OUT_L

**Type: Read Only**

| Register (Hex) | Register (Decimal) | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|---|---|---|---|---|---|---|---|---|---|
| 41 | 65 | TEMP_OUT[15:8] | | | | | | | |
| 42 | 66 | TEMP_OUT[7:0] | | | | | | | |

**Description:**

These registers store the most recent temperature sensor measurement.

Temperature measurements are written to these registers at the Sample Rate as defined in Register 25.

These temperature measurement registers, along with the gyroscope measurement registers, and external sensor data registers, are composed of two sets of registers: an internal register set and a user-facing read register set.

The data within the temperature sensor's internal register set is always updated at the Sample Rate. Meanwhile, the user-facing read register set duplicates the internal register set's data values whenever the serial interface is idle. This guarantees that a burst read of sensor registers will read measurements from the same sampling instant. Note that if burst reads are not used, the user is responsible for ensuring a set of single byte reads correspond to a single sampling instant by checking the Data Ready interrupt.

The scale factor and offset for the temperature sensor are found in the Electrical Specifications table of the MPU-3300 Product Specification document.

The temperature in degrees C for a given register value may be computed as:

Temperature in degrees C = (TEMP_OUT Register Value as a signed quantity)/340 + 36.53

Please note that the math in the above equation is in decimal.

**Parameters:**

*TEMP_OUT* 16-bit signed value.

Stores the most recent temperature sensor measurement.

## 4.17 Registers 67 to 72 – Gyroscope Measurements

## GYRO_XOUT_H, GYRO_XOUT_L, GYRO_YOUT_H, GYRO_YOUT_L, GYRO_ZOUT_H, and GYRO_ZOUT_L

**Type: Read Only**

| Register (Hex) | Register (Decimal) | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|---|---|---|---|---|---|---|---|---|---|
| 43 | 67 | GYRO_XOUT[15:8] | | | | | | | |
| 44 | 68 | GYRO_XOUT[7:0] | | | | | | | |
| 45 | 69 | GYRO_YOUT[15:8] | | | | | | | |
| 46 | 70 | GYRO_YOUT[7:0] | | | | | | | |
| 47 | 71 | GYRO_ZOUT[15:8] | | | | | | | |
| 48 | 72 | GYRO_ZOUT[7:0] | | | | | | | |

**Description:**

These registers store the most recent gyroscope measurements.

Gyroscope measurements are written to these registers at the Sample Rate as defined in Register 25.

These gyroscope measurement registers, along with the temperature measurement registers, and external sensor data registers, are composed of two sets of registers: an internal register set and a user-facing read register set.

The data within the gyroscope sensors' internal register set is always updated at the Sample Rate. Meanwhile, the user-facing read register set duplicates the internal register set's data values whenever the serial interface is idle. This guarantees that a burst read of sensor registers will read measurements from the same sampling instant. Note that if burst reads are not used, the user is responsible for ensuring a set of single byte reads correspond to a single sampling instant by checking the Data Ready interrupt.

Each 16-bit gyroscope measurement has a full scale defined in *FS_SEL* (Register 27). For each full scale setting, the gyroscopes' sensitivity per LSB in *GYRO_xOUT* is shown in the table below:

| FS_SEL | Full Scale Range | LSB Sensitivity |
|---|---|---|
| 0 | ± 225 °/s | 145.6 LSB/°/s |
| 1 | ± 450 °/s | 72.8 LSB/°/s |

**Parameters:**

*GYRO_XOUT* 16-bit 2's complement value.

Stores the most recent X axis gyroscope measurement.

*GYRO_YOUT* 16-bit 2's complement value.

Stores the most recent Y axis gyroscope measurement.

*GYRO_ZOUT* 16-bit 2's complement value.

Stores the most recent Z axis gyroscope measurement.

## 4.18 Registers 73 to 96 – External Sensor Data EXT_SENS_DATA_00 through EXT_SENS_DATA_23

**Type: Read Only**

| Register (Hex) | Register (Decimal) | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|---|---|---|---|---|---|---|---|---|---|
| 49 | 73 | EXT_SENS_DATA_00[7:0] | | | | | | | |
| 4A | 74 | EXT_SENS_DATA_01[7:0] | | | | | | | |
| 4B | 75 | EXT_SENS_DATA_02[7:0] | | | | | | | |
| 4C | 76 | EXT_SENS_DATA_03[7:0] | | | | | | | |
| 4D | 77 | EXT_SENS_DATA_04[7:0] | | | | | | | |
| 4E | 78 | EXT_SENS_DATA_05[7:0] | | | | | | | |
| 4F | 79 | EXT_SENS_DATA_06[7:0] | | | | | | | |
| 50 | 80 | EXT_SENS_DATA_07[7:0] | | | | | | | |
| 51 | 81 | EXT_SENS_DATA_08[7:0] | | | | | | | |
| 52 | 82 | EXT_SENS_DATA_09[7:0] | | | | | | | |
| 53 | 83 | EXT_SENS_DATA_10[7:0] | | | | | | | |
| 54 | 84 | EXT_SENS_DATA_11[7:0] | | | | | | | |
| 55 | 85 | EXT_SENS_DATA_12[7:0] | | | | | | | |
| 56 | 86 | EXT_SENS_DATA_13[7:0] | | | | | | | |
| 57 | 87 | EXT_SENS_DATA_14[7:0] | | | | | | | |
| 58 | 88 | EXT_SENS_DATA_15[7:0] | | | | | | | |
| 59 | 89 | EXT_SENS_DATA_16[7:0] | | | | | | | |
| 5A | 90 | EXT_SENS_DATA_17[7:0] | | | | | | | |
| 5B | 91 | EXT_SENS_DATA_18[7:0] | | | | | | | |
| 5C | 92 | EXT_SENS_DATA_19[7:0] | | | | | | | |
| 5D | 93 | EXT_SENS_DATA_20[7:0] | | | | | | | |
| 5E | 94 | EXT_SENS_DATA_21[7:0] | | | | | | | |
| 5F | 95 | EXT_SENS_DATA_22[7:0] | | | | | | | |
| 60 | 96 | EXT_SENS_DATA_23[7:0] | | | | | | | |

**Description:**

These registers store data read from external sensors by the Slave 0, 1, 2, and 3 on the auxiliary $I^2C$ interface. Data read by Slave 4 is stored in I2C_SLV4_DI (Register 53).

External sensor data is written to these registers at the Sample Rate as defined in Register 25. This access rate can be reduced by using the Slave Delay Enable registers (Register 103).

External sensor data registers, along with the gyroscope measurement registers, and temperature measurement registers, are composed of two sets of registers: an internal register set and a user-facing read register set.

The data within the external sensors' internal register set is always updated at the Sample Rate (or the reduced access rate) whenever the serial interface is idle. This guarantees that a burst read of sensor registers will read measurements from the same sampling instant. Note that if burst reads are not used, the user is responsible for ensuring a set of single byte reads correspond to a single sampling instant by checking the Data Ready interrupt.

Data is placed in these external sensor data registers according to I2C_SLV0_CTRL, I2C_SLV1_CTRL, I2C_SLV2_CTRL, and I2C_SLV3_CTRL (Registers 39, 42, 45, and 48). When more than zero bytes are read (*I2C_SLVx_LEN* > 0) from an enabled slave (*I2C_SLVx_EN* = 1), the slave is read at the Sample Rate (as defined in Register 25) or delayed rate (if specified in Register 52 and 103). During each Sample cycle, slave reads are performed in order of Slave number. If all slaves are enabled with more than zero bytes to be read, the order will be Slave 0, followed by Slave 1, Slave 2, and Slave 3.

Each enabled slave will have EXT_SENS_DATA registers associated with it by number of bytes read (*I2C_SLVx_LEN)* in order of slave number, starting from EXT_SENS_DATA_00. Note that this means enabling or disabling a slave may change the higher numbered slaves' associated registers. Furthermore, if fewer total bytes are being read from the external sensors as a result of such a change, then the data remaining in the registers which no longer have an associated slave device (i.e. high numbered registers) will remain in these previously allocated registers unless reset.

If the sum of the read lengths of all SLVx transactions exceed the number of available EXT_SENS_DATA registers , the excess bytes will be dropped. There are 24 EXT_SENS_DATA registers and hence the total read lengths between all the slaves cannot be greater than 24 or some bytes will be lost.

Note: Slave 4's behavior is distinct from that of Slaves 0-3. For further information regarding the characteristics of Slave 4, please refer to Registers 49 to 53.

**Example:**

Suppose that Slave 0 is enabled with 4 bytes to be read (*I2C_SLV0_EN* = 1 and *I2C_SLV0_LEN* = 4) while Slave 1 is enabled with 2 bytes to be read, (*I2C_SLV1_EN*=1 and *I2C_SLV1_LEN* = 2). In such a situation, EXT_SENS_DATA _00 through _03 will be associated with Slave 0, while EXT_SENS_DATA _04 and 05 will be associated with Slave 1.

If Slave 2 is enabled as well, registers starting from EXT_SENS_DATA_06 will be allocated to Slave 2.

If Slave 2 is disabled while Slave 3 is enabled in this same situation, then registers starting from EXT_SENS_DATA_06 will be allocated to Slave 3 instead.

**Register Allocation for Dynamic Disable vs. Normal Disable**

If a slave is disabled at any time, the space initially allocated to the slave in the EXT_SENS_DATA register, will remain associated with that slave. This is to avoid dynamic adjustment of the register allocation.

The allocation of the EXT_SENS_DATA registers is recomputed only when (1) all slaves are disabled, or (2) the *I2C_MST_RST* bit is set (Register 106).

This above is also true if one of the slaves gets NACKed and stops functioning.

### 4.19 Register 99 – I²C Slave 0 Data Out
### I2C_SLV0_DO

**Type: Read/Write**

| Register (Hex) | Register (Decimal) | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|---|---|---|---|---|---|---|---|---|---|
| 63 | 99 | I2C_SLV0_DO[7:0] | | | | | | | |

**Description:**

This register holds the output data written into Slave 0 when Slave 0 is set to write mode.

For further information regarding Slave 0 control, please refer to Registers 37 to 39.

**Parameters:**

*I2C_SLV0_DO* 8 bit unsigned value that is written into Slave 0 when Slave 0 is set to write mode.

### 4.20 Register 100 – I²C Slave 1 Data Out
### I2C_SLV1_DO

**Type: Read/Write**

| Register (Hex) | Register (Decimal) | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|---|---|---|---|---|---|---|---|---|---|
| 64 | 100 | I2C_SLV1_DO[7:0] | | | | | | | |

**Description:**

This register holds the output data written into Slave 1 when Slave 1 is set to write mode.

For further information regarding Slave 1 control, please refer to Registers 40 to 42.

**Parameters:**

*I2C_SLV1_DO* 8 bit unsigned value that is written into Slave 1 when Slave 1 is set to write mode.

## 4.21 Register 101 – I²C Slave 2 Data Out I2C_SLV2_DO

**Type: Read/Write**

| Register (Hex) | Register (Decimal) | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|---|---|---|---|---|---|---|---|---|---|
| 65 | 101 | I2C_SLV2_DO[7:0] | | | | | | | |

**Description:**

This register holds the output data written into Slave 2 when Slave 2 is set to write mode.

For further information regarding Slave 2 control, please refer to Registers 43 to 45.

**Parameters:**

*I2C_SLV2_DO* 8 bit unsigned value that is written into Slave 2 when Slave 2 is set to write mode.

## 4.22 Register 102 – I²C Slave 3 Data Out I2C_SLV3_DO

**Type: Read/Write**

| Register (Hex) | Register (Decimal) | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|---|---|---|---|---|---|---|---|---|---|
| 66 | 102 | I2C_SLV3_DO[7:0] | | | | | | | |

**Description:**

This register holds the output data written into Slave 3 when Slave 3 is set to write mode.

For further information regarding Slave 3 control, please refer to Registers 46 to 48.

**Parameters:**

*I2C_SLV3_DO* 8 bit unsigned value that is written into Slave 3 when Slave 3 is set to write mode.

## 4.23 Register 103 – I²C Master Delay Control
## I2C_MST_DELAY_CTRL

**Type: Read/Write**

| Register (Hex) | Register (Decimal) | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|---|---|---|---|---|---|---|---|---|---|
| 67 | 103 | DELAY_ES_SHADOW | - | - | I2C_SLV4_DLY_EN | I2C_SLV3_DLY_EN | I2C_SLV2_DLY_EN | I2C_SLV1_DLY_EN | I2C_SLV0_DLY_EN |

**Description:**

This register is used to specify the timing of external sensor data shadowing. The register is also used to decrease the access rate of slave devices relative to the Sample Rate.

When *DELAY_ES_SHADOW* is set to 1, shadowing of external sensor data is delayed until all data has been received.

When *I2C_SLV4_DLY_EN, I2C_SLV3_DLY_EN, I2C_SLV2_DLY_EN, I2C_SLV1_DLY_EN,* and *I2C_SLV0_DLY_EN* are enabled, the rate of access for the corresponding slave devices is reduced.

When a slave's access rate is decreased relative to the Sample Rate, the slave is accessed every

1 / (1 + *I2C_MST_DLY*) samples.

This base Sample Rate in turn is determined by *SMPLRT_DIV* (register 25) and *DLPF_CFG* (register 26).

For further information regarding *I2C_MST_DLY*, please refer to register 52.

For further information regarding the Sample Rate, please refer to register 25.

Bits 6 and 5 are reserved.

**Parameters:**

| | |
|---|---|
| *DELAY_ES_SHADOW* | When set, delays shadowing of external sensor data until all data has been received. |
| *I2C_SLV4_DLY_EN* | When enabled, slave 4 will only be accessed at a decreased rate. |
| *I2C_SLV3_DLY_EN* | When enabled, slave 3 will only be accessed at a decreased rate. |
| *I2C_SLV2_DLY_EN* | When enabled, slave 2 will only be accessed at a decreased rate. |
| *I2C_SLV1_DLY_EN* | When enabled, slave 1 will only be accessed at a decreased rate. |
| *I2C_SLV0_DLY_EN* | When enabled, slave 0 will only be accessed at a decreased rate. |

## 4.24 Register 104 – Signal Path Reset SIGNAL_PATH_RESET

**Type: Write Only**

| Register (Hex) | Register (Decimal) | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|---|---|---|---|---|---|---|---|---|---|
| 68 | 104 | - | - | - | - | - | GYRO _RESET | - | TEMP _RESET |

**Description:**

This register is used to reset the analog and digital signal paths of the gyroscope, and temperature sensors.

The reset will revert the signal path, analog to digital converters, and filters to their power up configurations.

Note: This register does not clear the sensor registers.

Bits 7, 6, 5, 4, 3, and 1 are reserved.

**Parameters:**

*GYRO_RESET* When set to 1, this bit resets the gyroscope analog and digital signal paths.

*TEMP_RESET* When set to 1, this bit resets the temperature sensor analog and digital signal paths.

## 4.25 Register 106 – User Control
**USER_CTRL**

**Type: Read/Write**

| Register (Hex) | Register (Decimal) | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|---|---|---|---|---|---|---|---|---|---|
| 6A | 106 | - | FIFO_EN | I2C_MST _EN | I2C_IF _DIS | - | FIFO _RESET | I2C_MST _RESET | SIG_COND _RESET |

**Description:**

This register allows the user to enable and disable the FIFO buffer, I²C Master Mode, and primary I²C interface. The FIFO buffer, I²C Master, sensor signal paths and sensor registers can also be reset using this register.

When *I2C_MST_EN* is set to 1, I²C Master Mode is enabled. In this mode, the MPU-3300 acts as the I²C Master to the external sensor slave devices on the auxiliary I²C bus. When this bit is cleared to 0, the auxiliary I²C bus lines (AUX_DA and AUX_CL) are logically driven by the primary I²C bus (SDA and SCL). This is a precondition to enabling Bypass Mode. For further information regarding Bypass Mode, please refer to Register 55.

The primary SPI interface will be enabled in place of the disabled primary I²C interface when *I2C_IF_DIS* is set to 1.

When the reset bits (*FIFO_RESET, I2C_MST_RESET, and SIG_COND_RESET*) are set to 1, these reset bits will trigger a reset and then clear to 0.

Bits 7 and 3 are reserved.

**Parameters:**

*FIFO_EN* — When set to 1, this bit enables FIFO operations.

When this bit is cleared to 0, the FIFO buffer is disabled. The FIFO buffer cannot be written to or read from while disabled.

The FIFO buffer's state does not change unless the MPU-3300 is power cycled.

*I2C_MST_EN* — When set to 1, this bit enables I²C Master Mode.

When this bit is cleared to 0, the auxiliary I²C bus lines (AUX_DA and AUX_CL) are logically driven by the primary I²C bus (SDA and SCL).

*I2C_IF_DIS* — When set to 1, this bit disables the primary I²C interface and enables the SPI interface instead.

*FIFO_RESET* — This bit resets the FIFO buffer when set to 1 while *FIFO_EN* equals 0. This bit automatically clears to 0 after the reset has been triggered.

*I2C_MST_RESET* — This bit resets the I²C Master when set to 1 while *I2C_MST_EN* equals 0. This bit automatically clears to 0 after the reset has been triggered.

*SIG_COND_RESET* When set to 1, this bit resets the signal paths for all sensors (gyroscopes, and temperature sensor). This operation will also clear the sensor registers. This bit automatically clears to 0 after the reset has been triggered.

When resetting only the signal path (and not the sensor registers), please use Register 104, SIGNAL_PATH_RESET.

## 4.26 Register 107 – Power Management 1 PWR_MGMT_1

**Type: Read/Write**

| Register (Hex) | Register (Decimal) | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|---|---|---|---|---|---|---|---|---|---|
| 6B | 107 | DEVICE _RESET | SLEEP | - | - | TEMP_DIS | CLKSEL[2:0] | | |

**Description:**

This register allows the user to configure the power mode and clock source. It also provides a bit for resetting the entire device, and a bit for disabling the temperature sensor.

By setting *SLEEP* to 1, the MPU-3300 can be put into low power sleep mode.

An internal 8MHz oscillator, gyroscope based clock, or external sources can be selected as the MPU-3300 clock source. When the internal 8 MHz oscillator or an external source is chosen as the clock source, the MPU-3300 can operate in low power modes with the gyroscopes disabled.

Upon power up, the MPU-3300 clock source defaults to the internal oscillator. However, it is highly recommended that the device be configured to use one of the gyroscopes (or an external clock source) as the clock reference for improved stability. The clock source can be selected according to the following table.

| CLKSEL | Clock Source |
|---|---|
| 0 | Internal 8MHz oscillator |
| 1 | PLL with X axis gyroscope reference |
| 2 | PLL with Y axis gyroscope reference |
| 3 | PLL with Z axis gyroscope reference |
| 4 | PLL with external 32.768kHz reference |
| 5 | PLL with external 19.2MHz reference |
| 6 | Reserved |
| 7 | Stops the clock and keeps the timing generator in reset |

For further information regarding the MPU-3300 clock source, please refer to the MPU-3300 Product Specification document.

Bits 5 and 4 are reserved.

**Parameters:**

*DEVICE_RESET* When set to 1, this bit resets all internal registers to their default values.

The bit automatically clears to 0 once the reset is done.

The default values for each register can be found in Section 3.

*SLEEP* When set to 1, this bit puts the MPU-3300 into sleep mode.

*TEMP_DIS* When set to 1, this bit disables the temperature sensor.

*CLKSEL* 3-bit unsigned value. Specifies the clock source of the device.

## 4.27 Register 108 – Power Management 2 PWR_MGMT_2

**Type: Read/Write**

| Register (Hex) | Register (Decimal) | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|---|---|---|---|---|---|---|---|---|---|
| 6C | 108 | - | - | - | - | - | STBY_XG | STBY_YG | STBY_ZG |

**Description:**

The user can put individual gyroscope axes into standby mode by using this register. If the device is using a gyroscope axis as the clock source and this axis is put into standby mode, the clock source will automatically be changed to the internal 8MHz oscillator.

Bits 7, 6, 5, 4, and 3 are reserved.

**Parameters:**

| | |
|---|---|
| *STBY_XG* | When set to 1, this bit puts the X axis gyroscope into standby mode. |
| *STBY_YG* | When set to 1, this bit puts the Y axis gyroscope into standby mode. |
| *STBY_ZG* | When set to 1, this bit puts the Z axis gyroscope into standby mode. |

## 4.28 Register 114 and 115 – FIFO Count Registers FIFO_COUNT_H and FIFO_COUNT_L

**Type: Read Only**

| Register (Hex) | Register (Decimal) | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|---|---|---|---|---|---|---|---|---|---|
| 72 | 114 | FIFO_COUNT[15:8] | | | | | | | |
| 73 | 115 | FIFO_COUNT[7:0] | | | | | | | |

**Description:**

These registers keep track of the number of samples currently in the FIFO buffer.

These registers shadow the FIFO Count value. Both registers are loaded with the current sample count when FIFO_COUNT_H (Register 72) is read.

Note: Reading only FIFO_COUNT_L will not update the registers to the current sample count. FIFO_COUNT_H must be accessed first to update the contents of both these registers.

*FIFO_COUNT* should always be read in high-low order in order to guarantee that the most current FIFO Count value is read.

**Parameters:**

| | |
|---|---|
| *FIFO_COUNT* | 16-bit unsigned value. Indicates the number of bytes stored in the FIFO buffer. This number is in turn the number of bytes that can be read from the FIFO buffer and it is directly proportional to the number of samples available given the set of sensor data bound to be stored in the FIFO (register 35 and 36). |

## 4.29 Register 116 – FIFO Read Write
**FIFO_R_W**

**Type: Read/Write**

| Register (Hex) | Register (Decimal) | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|---|---|---|---|---|---|---|---|---|---|
| 74 | 116 | FIFO_DATA[7:0] | | | | | | | |

<u>**Description:**</u>

This register is used to read and write data from the FIFO buffer.

Data is written to the FIFO in order of register number (from lowest to highest). If all the FIFO enable flags (see below) are enabled and all External Sensor Data registers (Registers 73 to 96) are associated with a Slave device, the contents of registers 59 through 96 will be written in order at the Sample Rate.

The contents of the sensor data registers (Registers 59 to 96) are written into the FIFO buffer when their corresponding FIFO enable flags are set to 1 in FIFO_EN (Register 35). An additional flag for the sensor data registers associated with $I^2C$ Slave 3 can be found in I2C_MST_CTRL (Register 36).

If the FIFO buffer has overflowed, the status bit *FIFO_OFLOW_INT* is automatically set to 1. This bit is located in INT_STATUS (Register 58). When the FIFO buffer has overflowed, the oldest data will be lost and new data will be written to the FIFO.

If the FIFO buffer is empty, reading this register will return the last byte that was previously read from the FIFO until new data is available. The user should check *FIFO_COUNT* to ensure that the FIFO buffer is not read when empty.

<u>**Parameters:**</u>

*FIFO_DATA* 8-bit data transferred to and from the FIFO buffer.

## 4.30 Register 117 – Who Am I
**WHO_AM_I**

**Type: Read Only**

| Register (Hex) | Register (Decimal) | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|---|---|---|---|---|---|---|---|---|---|
| 75 | 117 | - | WHO_AM_I[6:1] | | | | | | - |

**Description:**

This register is used to verify the identity of the device. The contents of *WHO_AM_I* are the upper 6 bits of the MPU-3300's 7-bit $I^2C$ address. The least significant bit of the MPU-3300's $I^2C$ address is determined by the value of the AD0 pin. The value of the AD0 pin is not reflected in this register.

The default value of the register is 0x68.

Bits 7 and 0 are reserved. (Hard coded to 0)

**Parameters:**

*WHO_AM_I* Contains the 6-bit $I^2C$ address of the MPU-3300.

The Power-On-Reset value of Bit6:Bit1 is 110 100.