IMU-3000
Motion Processing Unit
Register Map and Register Descriptions
Revision 1.1

CONTENTS

1 Document Information

1.1 Revision History

Revision Date	Revision	Description	
05/19/2011	1.0		Initial Release
06/09/2011	1.1		Modified for clarity and readability
		Sec. 3, 3.8, 3.10, 3.12	Removed bits pertaining to the Digital Motion Processor
		Sec. 3.1	Clarified initial value and bit 0 of WHO_AM_I register
		Sec. 3.1, 3.2	Provided initial values of registers

1.2 Purpose and Scope

This document provides information regarding the register map and descriptions for the IMU-3000™ Inertial Measurement Unit (IMU™).

Electrical characteristics are based upon simulation results and limited characterization data of advanced samples only. Specifications are subject to change without notice. Final specifications will be updated based upon characterization of final silicon.

1.3 Product Overview

The IMU-3000 is the world's first IMU solution with 6-axis sensor fusion using its field-proven and proprietary MotionFusion™ engine for consumer applications. The IMU-3000 has an embedded 3-axis gyroscope and Digital Motion Processor™ (DMP™) hardware accelerator engine with a secondary I^2C port that interfaces to third party digital accelerometers to deliver a complete 6-axis MotionFusion output to its primary I^2C port. This combines both linear and rotational motion into a single data stream for the application. The device is ideally suited for a wide variety of consumer products requiring a rugged, low-cost MotionProcessing™ solution for applications in game controllers, remote controls for broadband connected TVs and set top boxes, sports, fitness, medical and other applications. By providing an integrated MotionFusion output, the IMU-3000 offloads the intensive MotionProcessing computation requirements from the host processor, reducing the need for frequent polling of the motion sensor output and enabling use of low cost, low power microcontrollers.

The IMU-3000 features a 3-axis digital gyro with programmable full-scale ranges of ±250, ±500, ±1000, and ±2000 degrees/sec (dps), which is useful for precision tracking of both fast and slow motions. Rate noise performance sets the industry standard at 0.01 dps/√Hz, providing the highest-quality user experience in pointing and gaming applications. Factory-calibrated initial sensitivity reduces production-line calibration requirements. The part's on-chip FIFO and dedicated I^2C-master accelerometer sensor bus simplify system timing and lower system power consumption; the sensor bus allows the IMU-3000 to directly acquire data from the off-chip accelerometer without intervention from an external processor, while the FIFO allows a system microcontrollers to burst read the sensor data and then go to sleep while the IMU collects more data. Other industry-leading features include on-chip 16-bit ADCs, programmable digital filters, a precision clock with 1% variation from -40°C to 85°C, an embedded temperature sensor, programmable interrupts, and a low 13mW power consumption. Parts are available with an I^2C serial interface, a VDD operating range of 2.1 to 3.6V, and a VLOGIC interface voltage from 1.71V to 3.6V.

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 IMU-3000 package size down to a revolutionary footprint of 4x4x0.9mm (QFN), while providing the highest performance, lowest noise, and the lowest cost semiconductor packaging to address a wide range of handheld consumer electronic devices. The device provides the highest robustness by supporting 10,000*g* shock in operation. The highest cross-axis isolation is achieved by design from its single silicon integration.

The IMU-3000 was designed to connect directly with a third-party 3-axis digital accelerometer, which slaves directly to the IMU-3000 master and can be clocked from the internal phase locked loop of the IMU-3000 device, providing highly accurate timing for a true 6-axis MotionProcessing solution previously only available in costly and bulky inertial measurement units.

1.4 Software Solutions

This section describes the MotionApps™ software solutions included with the InvenSense MPU (Motion Processing Unit) and IMU (Inertial Measurement Unit) product families. Please note that the products within the IDG, IXZ, and ITG families do not include these software solutions.

The MotionApps Platform is a complete software solution that in combination with the InvenSense IMU and MPU MotionProcessor families delivers robust, well-calibrated 6-axis and/or 9-axis sensor fusion data using its field proven and proprietary MotionFusion™ engine. Solution packages are available for smartphones and tablets as well as for embedded microcontroller-based devices.

The MotionApps Platform provides a turn-key solution for developers and accelerates time-to-market. It consists of complex 6/9-axis sensor fusion algorithms, robust multi-sensor calibration, a proven software architecture for Android and other leading operating systems, and a flexible power management scheme.

The MotionApps Platform is integrated within the middleware of the target OS (the sensor framework), and also provides a kernel device driver to interface with the physical device. This directly benefits application developers by providing a cohesive set of APIs and a well-defined sensor data path in the user-space.

The table below describes the MotionApps software solutions included with the InvenSense MPU and IMU product families.

InvenSense MotionProcessor Devices and Included MotionApps Software

Feature	Included Software				Notes
	MotionApps	Embedded MotionApps	MotionApps Lite	Embedded MotionApps Lite	
Part Number	MPU-3050™ MPU-6050™		IMU-3000™		
Processor Type	Mobile Application Processor	8/16/32-bit Microcontroller	Mobile Application Processor	8/16/32-bit Microcontroller	
Applications	Smartphones, tablets	TV remotes, health/fitness, toys, other embedded	Smartphones, tablets	TV remotes, health/fitness, toys, other embedded	
6-Axis MotionFusion	Yes		Yes		< 2% Application Processor load using on-chip Digital Motion Processor (DMP). Reduces processing requirements for embedded applications
9-Axis MotionFusion	Yes		No		
Gyro Bias Calibration	Yes		Yes		No-Motion calibration and temperature calibration
3rd Party Compass Cal API	Yes		No		Integrates 3rd party compass libraries
Gyro-Assisted Compass Calibration (Fast Heading)	Yes		No		Quick compass calibration using gyroscope
Magnetic Anomaly Rejection (Improved Heading)	Yes		No		Uses gyro heading data when magnetic anomaly is detected

The table below lists recommended documentation for the MotionApps software solutions.

Software Documentation

Platform	MotionApps and MotionApps Lite	Embedded MotionApps and Embedded MotionApps Lite
Software Documentation	• Installation Guide for Linux and Android MotionApps Platform, v1.9 or later • MPL Functional Specifications	• Embedded MotionApps Platform User Guide, v3.0 or later • Embedded MPL Functional Specifications

For more information about the InvenSense MotionApps Platform, please visit the Developer's Corner or consult your local InvenSense Sales Representative.

2 Register Map

Addr (Hex)	Addr (Decimal)	Register Name	R/W	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0	0	WHO_AM_I	R/W	-	ID						-
C	12	X_OFFS_USRH	R/W	X_OFF_H							
D	13	X_OFFS_USRL	R/W	X_OFF_L							
E	14	Y_OFFS_USRH	R/W	Y_OFFS_H							
F	15	Y_OFFS_USRL	R/W	Y_OFFS_L							
10	16	Z_OFFS_USRH	R/W	Z_OFFS_H							
11	17	Z_OFFS_USRL	R/W	Z_OFFS_L							
12	18	FIFO_EN	R/W	TEMP_ OUT	GYRO_ XOUT	GYRO_ YOUT	GYRO_ ZOUT	AUX_ XOUT	AUX_ YOUT	AUX_ ZOUT	FIFO_ FOOTER
13	19	AUX_VDDIO	R/W	-	-	-	-	-	AUX_ VDDIO	-	-
14	20	AUX_SLV_ ADDR	R/W	CLKOUT EN	AUX_ID						
15	21	SMPLRT_DIV	R/W	SMPLRT_DIV							
16	22	DLPF_FS	R/W	-	-	-	FS_SEL		DLPF_CFG		
17	23	INT_CFG	R/W	ACTL	OPEN	LATCH_ INT_EN	INT_ ANYRD_ 2CLEAR	I2C_MST _ERR_E N	IMU_ RDY_ EN	-	RAW_ RDY_ EN
18	24	AUX_BURST_AD DR	R/W	BURST_ADDR							
1A	26	INT_STATUS	R	FIFO_FU LL	-	-		I2C_MST _ERR	IMU_ RDY	-	RAW_ DATA_ RDY
1B	27	TEMP_OUT_H	R	TEMP_OUT_H							
1C	28	TEMP_OUT_L	R	TEMP_OUT_L							
1D	29	GYRO_XOUT_H	R	GYRO_XOUT_H							
1E	30	GYRO_XOUT_L	R	GYRO_XOUT_L							
1F	31	GYRO_YOUT_H	R	GYRO_YOUT_H							
20	32	GYRO_YOUT_L	R	GYRO_YOUT_L							
21	33	GYRO_ZOUT_H	R	GYRO_ZOUT_H							
22	34	GYRO_ZOUT_L	R	GYRO_ZOUT_L							
23	35	AUX_XOUT_H	R	AUX_XOUT_H							
24	36	AUX_XOUT_L	R	AUX_XOUT_L							
25	37	AUX_YOUT_H	R	AUX_YOUT_H							
26	38	AUX_YOUT_L	R	AUX_YOUT_L							
27	39	AUX_ZOUT_H	R	AUX_ZOUT_H							
28	40	AUX_ZOUT_L	R	AUX_ZOUT_L							
3A	58	FIFO_COUNTH	R	-	-	-	-	-	-	FIFO_COUNT_H	
3B	59	FIFO_COUNTL	R	FIFO_COUNT_L							
3C	60	FIFO_R	R	FIFO_DATA							
3D	61	USER_CTRL	R/W	DMP_EN	FIFO_EN	AUX_IF_EN	-	AUX_IF_RST	DMP_RST	FIFO_RST	GYRO_RST
3E	62	PWR_MGM	R/W	H_RESET	SLEEP	STBY_XG	STBY_YG	STBY_ZG	CLK_SEL		

3 Register Description

This section details each register within the InvenSense IMU-3000 gyroscope. Note that any bit that is not defined should be set to zero in order to be compatible with future InvenSense devices.

The register space allows single-byte reads and writes, as well as burst reads and writes. When performing burst reads or writes, the memory pointer will increment until either (1) reading or writing is terminated by the master, or (2) the memory pointer reaches registers 57 or 60.

3.1 Register 0 – Who Am I

Type: Read/Write

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Default Value
0	0	-	ID						-	68h or 69h

Description:

This register is used to verify the identity of the device.

Bit 7 is reserved.

Parameters:

ID Contains the 6-bit I^2C address of the device. The Power-On-Reset value of Bit6: Bit1 is 110 100.

Bit0 is reserved. (May be 0 or 1)

3.2 Registers 12 to 17 – Gyro Offsets

Type: Read/Write

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Default Value
C	12	X_OFFS_H								00h
D	13	X_OFFS_L								00h
E	14	Y_OFFS_H								00h
F	15	Y_OFFS_L								00h
10	16	Z_OFFS_H								00h
11	17	Z_OFFS_L								00h

Description:

These registers are used to remove DC bias from the sensor outputs. The values in these registers are subtracted from the gyro sensor values before going into the sensor registers (see registers 29 to 34).

Parameters:

X_OFFS_H/L 16-bit offset (high and low bytes) of X gyro offset (2's complement)
Y_OFFS_H/L 16-bit offset (high and low bytes) of Y gyro offset (2's complement)
Z_OFFS_H/L 16-bit offset (high and low bytes) of Z gyro offset (2's complement)

3.3 Register 18 – FIFO Enable

Type: Read/Write

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Default Value
12	18	TEMP_ OUT	GYRO_ XOUT	GYRO_ YOUT	GYRO_ ZOUT	AUX_ XOUT	AUX_ YOUT	AUX_ ZOUT	FIFO_ FOOTER	00h

Description:

This register determines what data goes into the IMU-3000 FIFO, which is a 512 byte First-In-First-Out buffer (see register 60). Sensor data is automatically placed into the FIFO after each ADC sampling period is complete. The ADC sample rate is controlled by register 21.

The order at which the data is put into the FIFO is from MSB to LSB, which means that it will match the order shown in the parameter detail below. Two bytes are used for each reading. For example, if Gyro X, Gyro Y, Gyro Z, and FIFO_FOOTER are configured to go into the FIFO, then each sample period the following 8 bytes would be inserted into the FIFO, as shown below:

Gyro X High byte	Gyro X Low byte	Gyro Y High byte	Gyro Y Low byte	Gyro Z High byte	Gyro Z Low byte	FIFO_FOOTER High byte	FIFO_FOOTER Low byte

Parameters:

TEMP_OUT	Setting this inserts the Temperature reading into FIFO
GYRO_XOUT	Setting this inserts the X Gyro reading into FIFO
GYRO_YOUT	Setting this inserts the Y Gyro reading into FIFO
GYRO_ZOUT	Setting this inserts the Z Gyro reading into FIFO
AUX_XOUT	Setting this inserts the X Accelerometer reading into FIFO
AUX_YOUT	Setting this inserts the Y Accelerometer reading into FIFO
AUX_ZOUT	Setting this inserts the Z Accelerometer reading into FIFO
FIFO_FOOTER	Last word (2 bytes) for FIFO read. Described in more detail in Register 60

3.4 Registers 19 – AUX (Accel) VDDIO

Type: Read/Write

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Default Value
13	19	-	-	-	-	-	AUX_ VDDIO	-	-	00h

Description:

This register determines the I/O logic levels for the secondary I^2C bus clock and data lines (AUX_CL, AUX_DA). 1=VDD, 0=VLOGIC.

Bits 7 through 3, 1, and 0 are reserved.

Parameters:

AUX_VDDIO	I/O logic levels for the secondary I^2C bus clock and data lines (AUX_CL, AUX_DA). 1=VDD, 0=VLOGIC.

3.5 Register 20 – AUX (Accel) Slave Address

Type: Read/Write

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Default Value
14	20	CLKOUTEN	AUX_ID							00h

<u>Description:</u>

This register contains the enable bit for the reference clock output and the 7-bit slave address of the external 3rd party accelerometer.

The CLKOUTEN bit is used to enable (1) or disable (0) the reference clock output at the CLKOUT pin.

AUX_ID, the 7-bit accelerometer slave address, is used to access the accelerometer so that its sensor reading can be automatically read during each sample period at the same time as the gyro sensors.

When reading the external accelerometer registers, the IMU-3000 takes over the secondary I^2C bus, as a master to the accel, performing a burst read of the sensor registers. For this interface to be active, the AUX_*IF_EN* flag in the User Control register (61) must be set (set to 1).

Whenever changing this register, the secondary accel interface must be reset with AUX_*IF_RST* to take effect. Refer to the User Control register (61).

<u>Parameters:</u>

CLKOUTEN The enable bit for the reference clock output that is provided at the CLKOUT pin. 1=clock output enabled; 0=clock output disabled.

AUX_ID Contains the I^2C address of the external accelerometer. The address can be changed by writing to this register.

3.6 Register 21 – Sample Rate Divider

Type: Read/Write

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Default Value
15	21	SMPLRT_DIV								00h

<u>Description:</u>

This register determines the sample rate of the IMU-3000 gyros. The analog gyros are sampled internally at either 1kHz or 8kHz, determined by the *DLPF_CFG* setting (see register 22). This sampling is then filtered digitally and delivered into the sensor registers after the number of cycles determined by this register. The sample rate is given by the following formula:

$F_{sample} = F_{internal}$ / (divider+1), where $F_{internal}$ is either 1kHz or 8kHz

As an example, if the internal sampling is at 1kHz, then setting this register to 7 would give the following:

F_{sample} = 1kHz / (7 + 1) = 125Hz, or 8ms per sample

<u>Parameters:</u>

SMPLRT_DIV Sample rate divider: 0 to 255

3.7 Register 22 – DLPF, Full Scale

Type: Read/Write

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Default Value
16	22	-	-	-	FS_SEL		DLPF_CFG			00h

Description:

This register configures parameters related to the sensor acquisition.

The *FS_SEL* parameter allows setting the full-scale range of the gyro sensors, as described in the table below.

FS_SEL

FS_SEL	Gyro Full-Scale Range
0	±250°/sec
1	±500°/sec
2	±1000°/sec
3	±2000°/sec

The *DLPF_CFG* parameter sets the digital low pass filter configuration. It also determines the internal analog sampling rate used by the device as shown in the table below.

DLPF_CFG

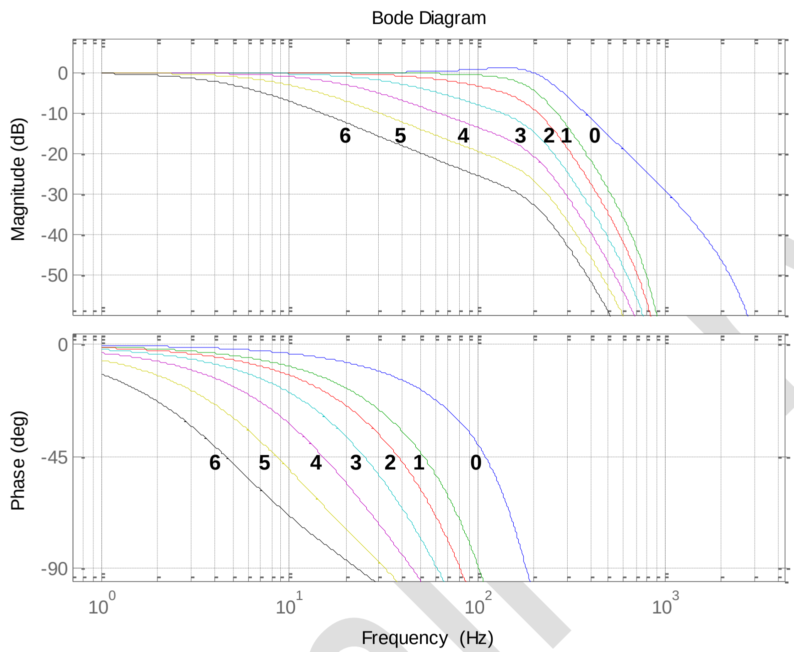
DLPF_CFG	Low Pass Filter Bandwidth	Analog Sample Rate
0	256Hz	8kHz
1	188Hz	1kHz
2	98Hz	1kHz
3	42Hz	1kHz
4	20Hz	1kHz
5	10Hz	1kHz
6	5Hz	1kHz
7	Reserved	Reserved

Bits 7 through 5 are reserved.

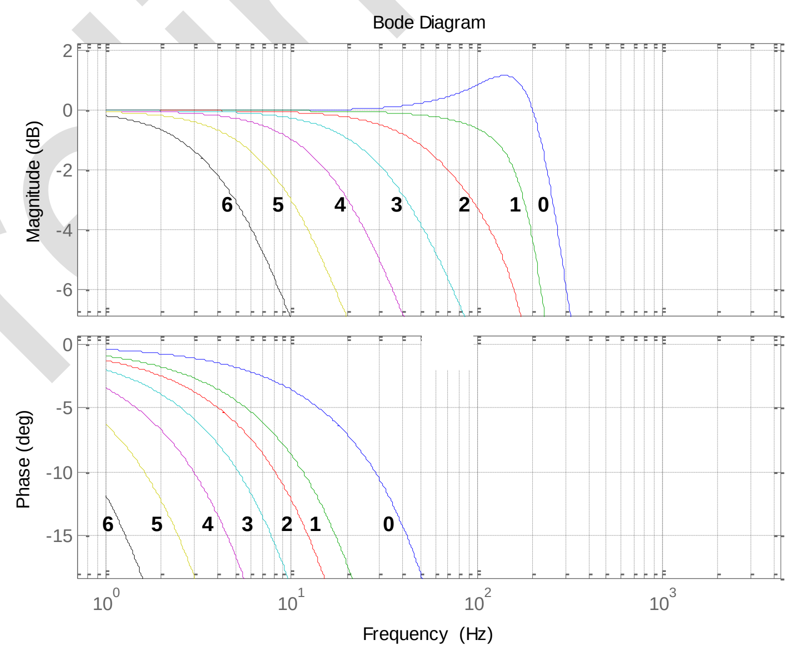
Parameters:

FS_SEL Full scale selection for gyro sensor data
DLPF_CFG Digital low pass filter configuration

DLPF Characteristics: The gain and phase responses of the digital low pass filter settings (*DLPF_CFG*) are shown below:



Gain and Phase vs. Digital Filter Setting



Gain and Phase vs. Digital Filter Setting, Showing Passband Details

3.8 Register 23 – Interrupt Configuration

Type: Read/Write

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Default Value
17	23	ACTL	OPEN	LATCH_ INT_EN	INT_ ANYRD_ 2CLEAR	I2C_MST _ERR_E N	IMU_ RDY_ EN	-	RAW_ RDY_ EN	00h

Description:
This register configures the interrupt operation of the IMU-3000. The interrupt output pin (INT) configuration can be set, the interrupt latching/clearing method can be set, and the triggers for the interrupt can be set. If LATCH_INT_EN = 1, the INT pin is held active until the interrupt status register is cleared.

Note that if the application requires reading every sample of data from the IMU-3000, it is best to enable the raw data ready interrupt (*RAW_RDY_EN*). This allows the application to know when new sample data is available.

Bit 1 is reserved.

Parameters:

ACTL	Logic level for INT output pin – 1=active low, 0=active high
OPEN	Drive type for INT output pin – 1=open drain, 0=push-pull
LATCH_INT_EN	Latch mode – 1=latch until interrupt is cleared, 0=50µs pulse
INT_ANYRD_2CLEAR	Interrupt status register clear method – 1=clear by reading any register, 0=clear by reading interrupt status register (26) only
I2C_MST_ERR_EN	Enable interrupt when accelerometer on secondary I^2C bus does not acknowledge IMU-3000
IMU_RDY_EN	Enable interrupt when device is ready (PLL ready after changing clock source)
RAW_RDY_EN	Enable interrupt when data is available

3.9 Register 24 – AUX (Accel) Burst Read Address / Secondary I^2C Bus I/O Level

Type: Read/Write

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Default Value
18	24	BURST_ADDR								00h

Description:
This register configures the burst-mode-read starting address for an accelerometer attached to the secondary I^2C bus of the IMU-3000, and determines the I/O logic levels of the secondary I^2C bus clock and data lines (AUX_CL, AUX_DA).

Parameters:

AUX_VDDIO	I/O logic levels for the secondary I^2C bus clock and data lines (AUX_CL, AUX_DA; 1=VDD, 0=VLOGIC)
BURST_ADDR	Burst-mode-read starting address for external accelerometer attached to secondary I^2C bus of the IMU-3000.

3.10 Register 26 – Interrupt Status

Type: Read only

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Default Value
1A	26	FIFO_FULL	-	-	-	I2C_MST _ERR	IMU_ RDY	-	RAW_ DATA_ RDY	00h

Description:

This register is used to determine the status of the IMU-3000 interrupt. Whenever one of the interrupt sources is triggered, the corresponding bit will be set. The polarity of the interrupt pin (active high/low) and the latch type (pulse or latch) has no affect on these status bits.

In normal use, the *RAW_DATA_RDY* interrupt is used to determine when new sensor data is available in either the sensor registers (27 to 34) or in the FIFO (60).

Interrupt Status bits get cleared as determined by INT_ANYRD_2CLEAR in the interrupt configuration register (23).

Bits 6 through 4 and 1 are reserved.

Parameters:

FIFO_FULL	FIFO has overflowed. Cleared when Register 26 is read and when *FIFO_RST* (register 61) is set.
I2C_MST_ERR	The IMU-3000 did not receive an acknowledge from the accelerometer on the secondary I^2C bus when the IMU-3000 was acting as a master
IMU_RDY	PLL ready
RAW_DATA_RDY	Raw data or FIFO data is ready

3.11 Registers 27 to 40 – Sensor Registers

Type: Read only

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Default Value*
1B	27	TEMP_OUT_H								00h
1C	28	TEMP_OUT_L								00h
1D	29	GYRO_XOUT_H								00h
1E	30	GYRO_XOUT_L								00h
1F	31	GYRO_YOUT_H								00h
20	32	GYRO_YOUT_L								00h
21	33	GYRO_ZOUT_H								00h
22	34	GYRO_ZOUT_L								00h
23	35	AUX_XOUT_H								00h
24	36	AUX_XOUT_L								00h
25	37	AUX_YOUT_H								00h
26	38	AUX_YOUT_L								00h
27	39	AUX_ZOUT_H								00h
28	40	AUX_ZOUT_L								00h

*Default Value applies if sensor is disabled.

Description:

These registers contain the gyro, temperature sensor, and accelerometer (aux) data for the IMU-3000. At any time, these values can be read from the device; however it is best to use the interrupt function to determine when new data is available.

If the FIFO is used, the contents of these registers are automatically copied into the FIFO at each sample period, and then this data can be read from the FIFO (register 60).

Before being placed into these registers, the gyro sensor data is first manipulated by the full scale setting (register 22) and the offset settings (registers 12 to 17).

Parameters:

TEMP_OUT_H/L 16-bit temperature data (2's complement data format)
GYRO_XOUT_H/L 16-bit X gyro output data (2's complement data format)
GYRO_YOUT_H/L 16-bit Y gyro output data (2's complement data format)
GYRO_ZOUT_H/L 16-bit Z gyro output data (2's complement data format)
AUX_XOUT_H/L 16-bit X aux (accel) output data (as available from aux)
AUX_YOUT_H/L 16-bit Y aux (accel) output data (as available from aux)
AUX_ZOUT_H/L 16-bit Z aux (accel) output data (as available from aux)

3.12 Registers 58 to 59 – FIFO Count

Type: Read only

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Default Value
3A	58	-	-	-	-	-	-	FIFO_COUNT_H		00h
3B	59	FIFO_COUNT_L								00h

Description:
These registers indicate how many bytes of valid data are contained in the FIFO. The FIFO can contain up to 512 bytes of data

If the FIFO gets filled up completely, the length will read 512. In this state, the IMU-3000 continues to put new sensor data into the FIFO, thus overwriting old FIFO data. Note, however, that the alignment of sensor data can change in this overflow condition. InvenSense recommends resetting the FIFO if an overflow condition occurs (use register 61), which will clear out the FIFO.

Bits 7 through 2 in Register 58 are reserved.

Parameters:
FIFO_COUNT_H/L Number of bytes currently in FIFO

3.13 Register 60 – FIFO Data

Type: Read only

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Default Value
3C	60	FIFO_DATA								00h

Parameters:

FIFO_DATA Contains the FIFO data

Description:

This is the output register of the FIFO. Each read of this register gets the oldest contents of the IMU-3000 FIFO buffer; thus the data is read out in the same order that the IMU-3000 put the data in. If the FIFO operation is enabled, the IMU-3000 puts new data into the FIFO at each sample interval. The data that goes in is determined by the FIFO enable register (18).

A burst read or write is required for reading or writing *multiple* bytes to or from this register, since any read or write on this register causes an auto increment and a prefetch to occur.

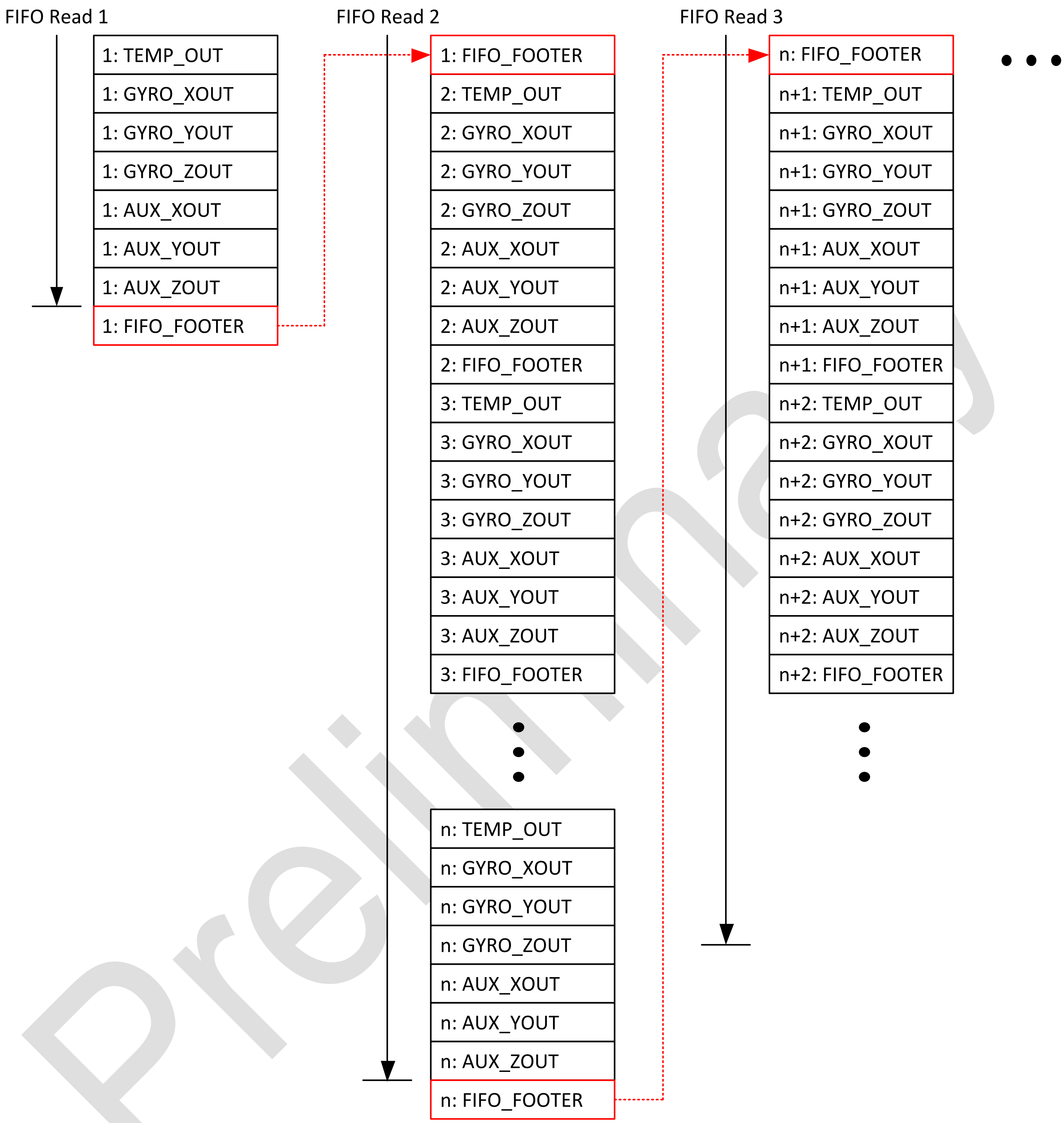
Proper operation of the FIFO requires that at least one word (2 bytes) of data be left in the FIFO during any read operation. To implement this, it is recommended that one extra two byte word (FIFO_FOOTER) be added to the end of the FIFO data so that all desired data can be read at each cycle, leaving the extra word remaining in the FIFO. This extra word will be read out (first) during the next read operation on the FIFO.
Data is read into the FIFO in the following order:

TEMP_OUT	Temperature high and low bytes (2 bytes)
GYRO_XOUT	X Gyro high and low bytes (2 bytes)
GYRO_YOUT	Y Gyro high and low bytes (2 bytes)
GYRO_ZOUT	Z Gyro high and low bytes (2 bytes)
AUX_XOUT	X Accelerometer high and low bytes (2 bytes)
AUX_YOUT	Y Accelerometer high and low bytes (2 bytes)
AUX_ZOUT	Z Accelerometer high and low bytes (2 bytes)
FIFO_FOOTER	*Last word for FIFO read (2 bytes)*

For example, if it is desired to obtain temperature, gyro, and accelerometer data from the FIFO, then one should also add FIFO_FOOTER into the FIFO enable register (18) in addition to the desired data. As shown in the figure below, the first time data is written to the FIFO, the FIFO will contain: *TEMP_OUT, GYRO_XOUT, GYRO_YOUT, GYRO_ZOUT, AUX_XOUT, AUX_YOUT, AUX_ZOUT,* and *FIFO_FOOTER*. The first FIFO read will read all but the *FIFO_FOOTER* data, which will be read in the 2nd FIFO read. In the 2nd FIFO read, the *FIFO_FOOTER* data that was left over from the previous read is read out first, followed by all but the last *FIFO_FOOTER* data in the FIFO. This pattern of reading is continued, as shown in the figure.

Note that the first FIFO read is similar to the subsequent reads in that one word of data is always left in the FIFO. It differs, though, in that the in subsequent reads the leftover data from the previous read is read first; however, for the first read there is no leftover data from a previous read.

If the FIFO is allowed to overflow, it operates as a circular buffer in which at any time it contains the most recent 512 bytes. Recommended operation in this mode is to disable data going into the FIFO prior to reading the FIFO to avoid pointer conflicts. After halting the FIFO input, the 512 bytes in the FIFO should be read out in a single burst read. The first byte read will not be valid.



Reading from the FIFO

(Note that AUX_XOUT, AUX_YOUT, and AUX_ZOUT are the X, Y, and Z accelerometer outputs, respectively.)

3.14 Register 61 – User Control

Type: Read/Write

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Default Value
3D	61	DMP_ EN	FIFO_ EN	AUX_IF_ EN	-	AUX_IF_ RST	DMP_ RST	FIFO_ RST	GYRO_ RST	00h

Description:

This register is used to enable various modes on the IMU-3000, as well as reset these functions.

For each of the functions that can be enabled, the function should be reset at the same time to assure it works properly. Note that the reset bits in the register are automatically cleared after the function is reset.

For example, to enable the FIFO set both the *FIFO_EN* and the *FIFO_RST* bits. This will start the FIFO storage on the next sample period.

As an additional example, for an external processor to communicate directly to the external accelerometer (i.e. have the secondary I^2C bus be directly connected to the primary I^2C bus), the AUX*_IF_EN* bit should be cleared and the AUX*_IF_RST* bit should be set. This will allow the I^2C bus to pass through the IMU-3000 and allow the processor to control the accelerometer device (as well as the IMU). Pass through mode is useful for allowing the processor to configure the accelerometer, since the IMU-3000 can perform burst reads on the accelerometer, but is not set up to configure the device.

Bit 4 is reserved.

Parameters:

DMP_EN	Enable Digital Motion Processor (DMP)
FIFO_EN	Enable FIFO operation for sensor data
AUX_IF_EN	Enable IMU as master to accelerometer interface via secondary I^2C (clear bit to configure primary I^2C bus to pass through directly to the secondary I^2C bus)
AUX_IF_RST	Reset secondary accelerometer interface function; set this whenever changing AUX*_IF_EN*
DMP_RST	Reset DMP function; set this whenever changing *DMP_EN*
FIFO_RST	Reset FIFO function; set this to clear FIFO or when changing *FIFO_EN*
GYRO_RST	Reset gyro analog and digital functions

3.15 Register 62 – Power Management

Type: Read/Write

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Default Value
3E	62	H_RESET	SLEEP	STBY _XG	STBY _YG	STBY _ZG	CLK_SEL			00h

Description:

This register is used to manage the power control, select the clock source, and to issue a master reset to the device.

H_RESET is used to reset the device and set the internal registers to the power-up default settings.

STBY_XG, STBY_YG, and *STBY_ZG* are used to place the gyros into a standby or active mode (1=standby; 0=normal operating mode).

Setting the *SLEEP* bit in the register puts the device into a low power sleep mode. In this mode, only the serial interface and internal registers remain active, allowing for a very low standby current. Clearing this bit puts the device back into normal mode. The individual standby selections for each of the gyros should be used if any of them are not used by the application.

The *CLK_SEL* setting determines the device clock source as follows:

CLK_SEL

CLK_SEL	Clock Source
0	Internal oscillator
1	PLL with X Gyro reference
2	PLL with Y Gyro reference
3	PLL with Z Gyro reference
4	PLL with external 32.768kHz reference
5	PLL with external 19.2MHz reference
6	Reserved
7	Stop clock and synchronous reset clock state

On power up, the IMU-3000 defaults to the internal oscillator. It is highly recommended that the device is configured to use one of the gyros (or an external clock) as the clock reference, due to the improved stability.

Parameters:

H_RESET	Reset device and internal registers to the power-up-default settings
SLEEP	Enable low power sleep mode
STBY_XG	Put gyro X in standby mode (1=standby, 0=normal)
STBY_YG	Put gyro Y in standby mode (1=standby, 0=normal)
STBY_ZG	Put gyro Z in standby mode (1=standby, 0=normal)
CLK_SEL	Select device clock source