DUAL INTERMEDIATE FREQUENCY (IF) ANALOG FRONT-END FOR DIGITAL RADIO

FEATURES

- **Qualified for Automotive Applications**
- **Two Intermediate Frequency (IF) Analog-to-Digital Converters (ADCs)**
- **Two 12-Bit Auxiliary Digital-to-Analog Converters (DACs)**
- **8-Bit Auxiliary ADCs with Four-Channel Input Multiplexer (MUX)**
- **Integrated IF Digital Processing Core**
- **Integrated Circuitry for Third-Overtone Master Clock Oscillator**
- **Wakeup Circuit/Real-Time Clock With Separate Crystal Oscillator**
- **Flexible Data Interface Optimized for TMS Family of Digital Baseband Processors**
- **Pin-Selectable SPI™ and I^2C™ Control Interfaces**
- **3.3-V/1.8-V Supply (Integrated Regulator Available to Optionally Generate 1.8-V Supply)**
- **TQFP-144 (RFP) PowerPAD™ Package**

APPLICATIONS

- **IF-Sampled AM/FM Radio**
- **Hybrid Digital (HD) Digital Audio Broadcasting (DAB) Digital Radio**

DESCRIPTION

The AFE8221 implements the intermediate frequency (IF) sampling and processing functions of a digital radio receiver system. It is designed to be used with TI's digital radio baseband processors and AM/FM tuners. The AFE8221 can also be programmed by the baseband processor for use in conventional AM/FM and digital radio. This unit includes two IF inputs with associated filtering and digital processing circuitry.

The receive circuit oversamples the radio tuner IF output to reduce noise and improve dynamic range. The IF analog-to-digital converter (ADC) oversamples the IF input at rates up to 75 MHz. The AFE8221 then digitally mixes, filters, and decimates the signal to provide I and Q output signals to the baseband processor. A clock oscillator circuit is provided that can be used with an appropriate third-overtone crystal and external tank circuit to generate the sampling clock for the IF ADCs.

The AFE8221 also includes a real-time clock and associated low-power oscillator circuit. Two auxiliary digital-to-analog converters (DACs) are included for system control functions. An 8-bit auxiliary ADC and input multiplexer (MUX) can be used for system diagnostic functions. Other features include 12 general-purpose input/output (GPIO) lines, programmable interrupt generators, and an I^2C master for communication between the AFE and the tuner(s).

The AFE8221 is available in a TQFP-144 (20 mm × 20 mm) package and uses a 3.3-V and a 1.8-V power supply. An onboard voltage regulator is included to optionally generate the 1.8-V digital supply for the AFE8221.

ORDERING INFORMATION(1)

T_A	PACKAGE(2)		ORDERABLE PART NUMBER	TOP-SIDE MARKING
–40°C to 85°C	HTQFP – RFP	Tray of 60	AFE8221IRFPQ1	AFE8221Q

(1) For the most current package and ordering information, see the Package Option Addendum at the end of this document, or see the TI web site at www.ti.com.
(2) Package drawings, thermal data, and symbolization are available at www.ti.com/packaging.

This integrated circuit can be damaged by ESD. Texas Instruments recommends that all integrated circuits be handled with appropriate precautions. Failure to observe proper handling and installation procedures can cause damage.

ESD damage can range from subtle performance degradation to complete device failure. Precision integrated circuits may be more susceptible to damage because very small parametric changes could cause the device not to meet its published specifications.

ABSOLUTE MAXIMUM RATINGS(1)

over operating free-air temperature range (unless otherwise noted)

	Supply voltage range	AVDD	–0.5 V to 3.6 V
		DVDD	–0.5 V to 3.6 V
		IOVDD	–0.5 V to 3.6 V
	Voltage between	AGND to DGND	–0.3 V to 0.5 V
		AVDD to DVDD	–3.3 V to 3.3 V
V_{IN}	Digital input voltages(2)		–0.3 V to (DVDD + 0.3 V)
V_{OUT}	Digital data output voltage		–0.3 V to (DVDD + 0.3 V)
T_A	Operating free-air temperature range		–40°C to 85°C
T_{stg}	Storage temperature range		–55°C to 125°C

(1) Stresses beyond those listed under *absolute maximum ratings* may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under *recommended operating conditions* is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.
(2) Measured with respect to DGND.

RECOMMENDED OPERATING CONDITIONS

over operating free-air temperature range (unless otherwise noted)

	PARAMETER	MIN	TYP	MAX	UNIT
AVDD	Analog supply voltage	3.14	3.3	3.6	V
DVDD	Digital supply voltage	1.6	1.8	2.0	V
IOVDD	Output driver supply voltage	1.6		3.6	V
	Input common-mode voltage		VCM		V
	Differential input voltage		2		V_{PP}
V_{IH}	High-level input voltage, digital inputs	0.7 × IOVDD			V
V_{IL}	Low-level input voltage, digital inputs			0.25 × IOVDD	V
T_A	Operating free-air temperature	–40		+85	°C

POWER SUPPLY SPECIFICATIONS

T_A = 25°C, AVDD = IOVDD = 3.3 V, DVDD = 1.8 V (unless otherwise noted)

PARAMETER	TEST CONDITIONS	MIN	TYP	MAX	UNIT
Power Consumption					
Analog supply current			130	155	mA
Digital supply current	$\overline{\text{REG\_ENB}}$ disabled		65	85	mA
Digital I/O supply current	$\overline{\text{REG\_ENB}}$ disabled		35	50	mA
	$\overline{\text{REG\_ENB}}$ enabled		105	125	mA
Power dissipation	$\overline{\text{REG\_ENB}}$ disabled		660		mW
	$\overline{\text{REG\_ENB}}$ enabled		725		mW
Reduced-Power Modes					
Software power-down	Control register address 1 set to 0x0000		100		mW
Hardware power-down	PWD enabled		50		µW

IF ADC SPECIFICATIONS

T_A = 25°C, AVDD = IOVDD = 3.3 V, DVDD = 1.8 V, f_S = 75 MHz (unless otherwise noted)

PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
DC Accuracy						
Input impedance				2		kΩ
Offset error				3.0		mV
Gain error				1.0		%FS
Full-scale input level		Peak differential, 1x gain		2.0		V_{PP}
		Peak differential, 2x gain		1.0		V_{PP}
Power Supply						
PSRR	Power-supply rejection ratio	AVDD = 3.15 VDC to 3.6 VDC		72		dB
References						
REFP	Positive reference		1.9	2.0	2.1	V
REFN	Negative reference		0.9	1.0	1.1	V
VCM	Common-mode voltage		1.4	1.5	1.6	V
AC Performance						
Input sample rate			75			MHz
SNR	Signal-to-noise ratio within a limited passband	Input 10.7 MHz, –1 dBFS, in 3-kHz passband		105		dBc
		Input 10.7 MHz, –1 dBFS, in 100-kHz passband	85	90		
	Third-order intermodulation distortion	–7-dB signals at 10.656 MHz and 10.729 MHz		91		dB
		–10-dB signals at 10.656 MHz and 10.729 MHz		94		
SFDR	Spurious-free dynamic range	–1-dB input at 10.7 MHz, 100-kHz passband	88	96		dBc

AUXILIARY DAC SPECIFICATIONS

T_A = 25°C, AVDD = IOVDD = 3.3 V, DVDD = 1.8 V (unless otherwise noted)

PARAMETER	TEST CONDITIONS	MIN	TYP	MAX	UNIT
Resolution					
Resolution			12		Bits
Output Voltage Range					
Output voltage range	Input code 0x000		0		V
	Input code 0x3FF		2.7		V
Settling Time					
Settling time	0.1% of FSR			10	µs
DC Performance					
Offset			±1		% of FSR
Gain error			±5		% of FSR
DNL	Monotonic		±0.5		LSB
INL	Offset and gain errors removed		±3.0		LSB
PSRR	Input code 0x200, AVDD = 3.15 VDC to 3.6 VDC		30		dB

AUXILIARY ADC SPECIFICATIONS

T_A = 25°C, AVDD = IOVDD = 3.3 V, DVDD = 1.8 V (unless otherwise noted)

PARAMETER	TEST CONDITIONS	MIN	TYP	MAX	UNIT
Resolution					
Resolution			10		Bits
Input Voltage Range					
Input voltage range	Input code 0x00		0		V
	Input code 0xFF		3.0		V
Input Impedance					
Input impedance			30		kΩ
Conversion Time					
Conversion Time				8704	MCLK cycles
DC Performance					
Offset			±1.0		% of FSR
Gain error			±1.5		% of FSR
DNL	Monotonic	–1.0		1.5	LSB
INL	Offset and gain errors removed	–1.5	±0.5	1.5	LSB
PSRR	Midscale input, AVDD = 3.15 VDC to 3.6 VDC		30		dB

DIGITAL I/O SPECIFICATIONS

T_A = 25°C, IOVDD = 3.3 V (unless otherwise noted)

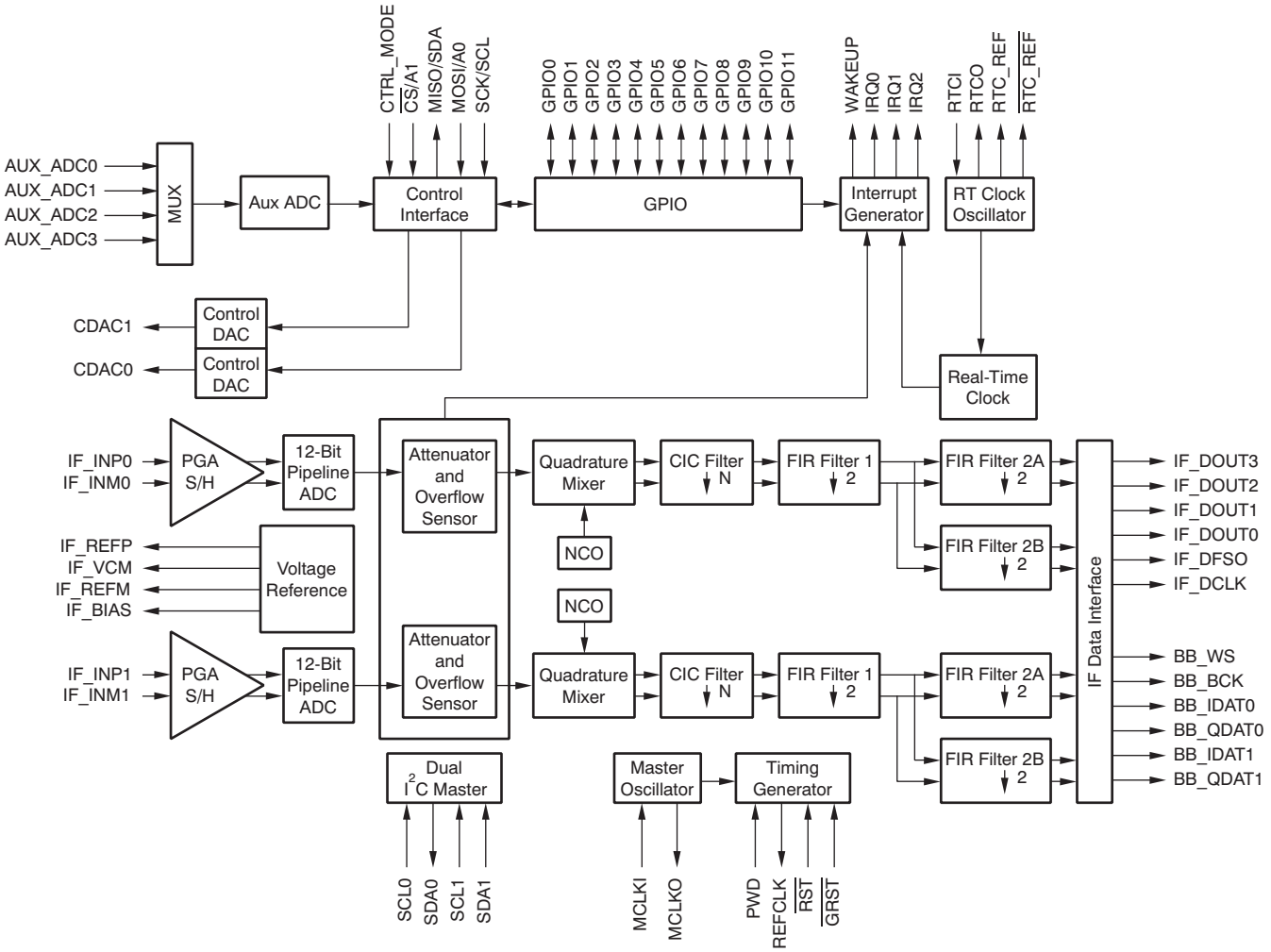
PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
I_{IH}	High-level input current	V_{IH} = 1.6 V to 3.6 V	–10		10	µA
I_{IL}	Low-level input current	V_{IL} = 0 V to 0.4 V	–10		10	µA
V_{OH}	High-level output voltage	I_{OH} = –50 µA	0.8 × IOVDD			V
V_{OL}	Low-level output voltage	I_{OL} = 50 µA			0.2 × IOVDD	V

CLOCK OSCILLATOR SPECIFICATIONS

T_A = 25°C, DVDD = 1.8 V, IOVDD = 3.3 V (unless otherwise noted)

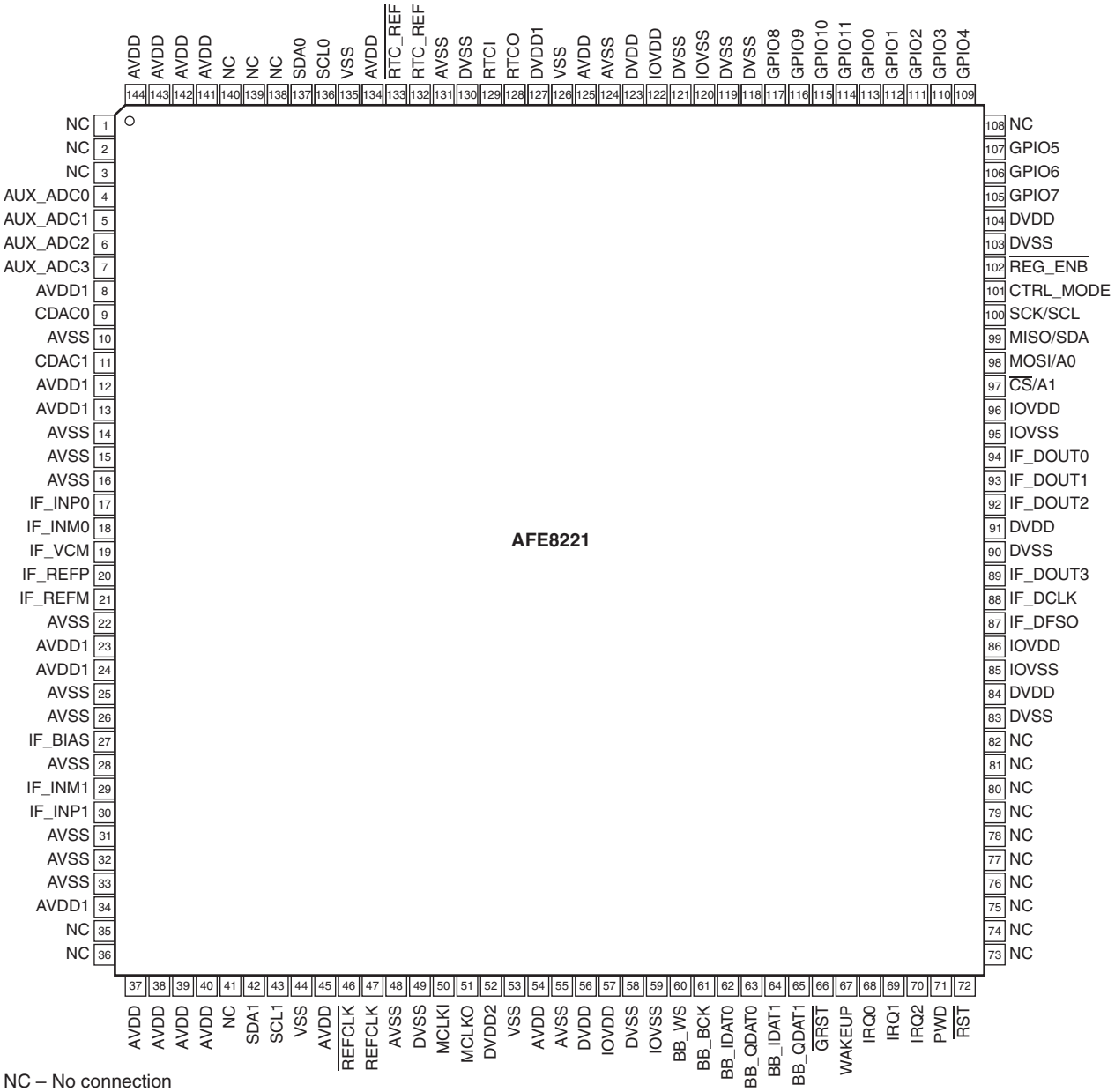
PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
f_{XTAL}	Crystal frequency	See the Master Clock Oscillator section	20		75	MHz

FUNCTIONAL BLOCK DIAGRAM



TERMINAL ASSIGNMENTS

TQFP-144
Top View



NC – No connection

TERMINAL FUNCTIONS

TERMINAL		FUNCTION	DESCRIPTION
NAME	NO.		
NC	1	Open	No connect
NC	2	Open	No connect
NC	3	Open	No connect
AUX_ADC0	4	Analog input	Auxiliary ADC channel 0
AUX_ADC1	5	Analog input	Auxiliary ADC channel 1
AUX_ADC2	6	Analog input	Auxiliary ADC channel 2
AUX_ADC3	7	Analog input	Auxiliary ADC channel 3
AVDD1	8	Supply	3.3-V analog supply (internally switched)
CDAC0	9	Output	Control DAC 0 output
AVSS	10	Ground	Analog ground
CDAC1	11	Output	Control DAC 1 output
AVDD1	12	Supply	3.3-V analog supply (internally switched)
AVDD1	13	Supply	3.3-V analog supply (internally switched)
AVSS	14	Ground	Analog ground
AVSS	15	Ground	Analog ground
AVSS	16	Ground	Analog ground
IF_INP0	17	Input	IF ADC channel 0 positive input
IF_INM0	18	Input	IF ADC channel 0 negative input
IF_VCM	19	Output	IF ADC common-mode voltage
IF_REFP	20	Output	IF ADC positive reference
IF_REFM	21	Output	IF ADC negative reference
AVSS	22	Ground	Analog ground
AVDD1	23	Supply	3.3-V analog supply (internally switched)
AVDD1	24	Supply	3.3-V analog supply (internally switched)
AVSS	25	Ground	Analog ground
AVSS	26	Ground	Analog ground
IF_BIAS	27	Input	IF ADC bias input
AVSS	28	Ground	Analog ground
IF_INM1	29	Input	IF ADC channel 1 negative input
IF_INP1	30	Input	IF ADC channel 1 positive input
AVSS	31	Ground	Analog ground
AVSS	32	Ground	Analog ground
AVSS	33	Ground	Analog ground
AVDD1	34	Supply	3.3-V analog supply (internally switched)
NC	35	Open	No connect
NC	36	Open	No connect
AVDD	37	Supply	3.3-V analog supply
AVDD	38	Supply	3.3-V analog supply
AVDD	39	Supply	3.3-V analog supply
AVDD	40	Supply	3.3-V analog supply
NC	41	Open	No connect
SDA1	42	Bidirectional	Channel 1 tuner I^2C data
SCL1	43	Output	Channel 1 tuner I^2C clock
AVSS	44	Ground	Analog ground
AVDD	45	Supply	3.3-V analog supply
$\overline{\text{REFCLK}}$	46	Output	Inverted reference clock output

TERMINAL FUNCTIONS (continued)

TERMINAL		FUNCTION	DESCRIPTION
NAME	NO.		
REFCLK	47	Output	Reference clock output
AVSS	48	Ground	Analog ground
DVSS	49	Ground	Digital ground (for MCLK oscillator)
MCLKI	50	Input	MCLK oscillator input
MCLKO	51	Output	MCLK oscillator output
DVDD2	52	Supply	1.8-V digital supply (for MCLK oscillator)
AVSS	53	Ground	Analog ground
AVDD	54	Supply	3.3-V analog supply
AVSS	55	Ground	Analog ground
DVDD	56	Supply	1.8-V digital supply
IOVDD	57	Supply	3.3-V digital I/O supply
DVSS	58	Ground	Digital ground
IOVSS	59	Ground	Digital I/O ground
BB_WS	60	Output	Secondary baseband word select
BB_BCK	61	Output	Secondary baseband word bit clock
BB_IDAT0	62	Output	Secondary baseband channel 0 output (I)
BB_QDAT0	63	Output	Secondary baseband channel 0 output (Q)
BB_IDAT1	64	Output	Secondary baseband channel 1 output (I)
BB_QDAT1	65	Output	Secondary baseband channel 1 output (Q)
$\overline{\text{GRST}}$	66	Input	Global reset (active low)
WAKEUP	67	Output	WAKEUP interrupt output
IRQ0	68	Output	Interrupt output 0
IRQ1	69	Output	Interrupt output 1
IRQ2	70	Output	Interrupt output 2
PWD	71	Input	Power-down pin (active high)
$\overline{\text{RST}}$	72	Input	Reset pin (active low)
NC	73	Open	No connect
NC	74	Open	No connect
NC	75	Open	No connect
NC	76	Open	No connect
NC	77	Open	No connect
NC	78	Open	No connect
NC	79	Open	No connect
NC	80	Open	No connect
NC	81	Open	No connect
NC	82	Open	No connect
DVSS	83	Ground	Digital ground
DVDD	84	Supply	1.8-V digital supply
IOVSS	85	Ground	Digital I/O ground
IOVDD	86	Supply	3.3-V digital I/O supply
IF_DFSO	87	Output	IF interface frame sync
IF_DCLK	88	Output	IF interface bit clock
IF_DOUT3	89	Output	IF interface data out 3
DVSS	90	Ground	Digital ground
DVDD	91	Supply	1.8-V digital supply
IF_DOUT2	92	Output	IF interface data out 2

TERMINAL FUNCTIONS (continued)

TERMINAL		FUNCTION	DESCRIPTION
NAME	NO.		
IF_DOUT1	93	Output	IF interface data out 1
IF_DOUT0	94	Output	IF interface data out 0
IOVSS	95	Ground	Digital I/O ground
IOVDD	96	Supply	3.3-V digital I/O supply
$\overline{CS}$/A1	97	Input	SPI chip select (active low) / I^2C address bit 1
MOSI/A0	98	Input	SPI data in / I^2C address bit 0
MISO/SDA	99	Bidirectional	SPI data out / I^2C SDA
SCK/SCL	100	Input	SPI clock / I^2C SCL
CTRL_MODE	101	Input	Control interface mode select (SPI = 0, I^2C = 1)
$\overline{REG\_ENB}$	102	Input	Enable onboard DVDD regulator (active low)
DVSS	103	Ground	Digital ground
DVDD	104	Supply	1.8-V digital supply
GPIO7	105	Bidirectional	GPIO 7
GPIO6	106	Bidirectional	GPIO 6
GPIO5	107	Bidirectional	GPIO 5
NC	108	Open	No connect
GPIO4	109	Bidirectional	GPIO 4
GPIO3	110	Bidirectional	GPIO 3
GPIO2	111	Bidirectional	GPIO 2
GPIO1	112	Bidirectional	GPIO 1
GPIO0	113	Bidirectional	GPIO 0
GPIO11	114	Bidirectional	GPIO 11
GPIO10	115	Bidirectional	GPIO 10
GPIO9	116	Bidirectional	GPIO 9
GPIO8	117	Bidirectional	GPIO 8
DVSS	118	Ground/input	Digital ground/Test1
DVSS	119	Ground/input	Digital ground/Test0
IOVSS	120	Ground	Digital I/O ground
DVSS	121	Ground	Digital ground
IOVDD	122	Supply	3.3-V digital I/O supply
DVDD	123	Supply	1.8-V digital supply
AVSS	124	Ground	Analog ground
AVDD	125	Supply	3.3-V analog supply
AVSS	126	Ground	Analog ground
DVDD1	127	Supply	1.8-V digital supply (for RTC oscillator)
RTCO	128	Output	RTC oscillator output
RTCI	129	Input	RTC oscillator input
DVSS	130	Ground	Digital ground (for RTC oscillator)
AVSS	131	Ground	Analog ground
RTC_REF	132	Output	RTC output
$\overline{RTC\_REF}$	133	Output	Inverted RTC output
AVDD	134	Supply	3.3-V analog supply
AVSS	135	Ground	Analog ground
SCL0	136	Output	Channel 0 tuner I^2C clock
SDA0	137	Bidirectional	Channel 0 tuner I^2C data
NC	138	Open	No connect

TERMINAL FUNCTIONS (continued)

TERMINAL		FUNCTION	DESCRIPTION
NAME	NO.		
NC	139	Open	No connect
NC	140	Open	No connect
AVDD	141	Supply	3.3-V analog supply
AVDD	142	Supply	3.3-V analog supply
AVDD	143	Supply	3.3-V analog supply
AVDD	144	Supply	3.3-V analog supply
AVSS	—	Analog ground	Center pad

TIMING DIAGRAMS

Output Data Interface Timing

PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
t_{D1}	DCLK to DFSO delay		–2.9		3.7	ns
t_{D2}	DCLK to DOUTx delay		–3.1		3.8	ns

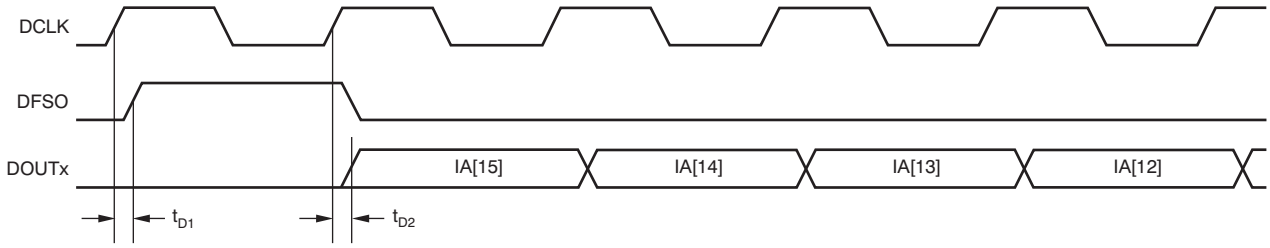


Figure 1. Output Data Interface Timing

Primary Data Interface Timing

PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
t_{D1}	BCLK to WS delay		–2.9		3.7	ns
t_{D2}	BCLK to DOUTx delay		–3.1		3.8	ns

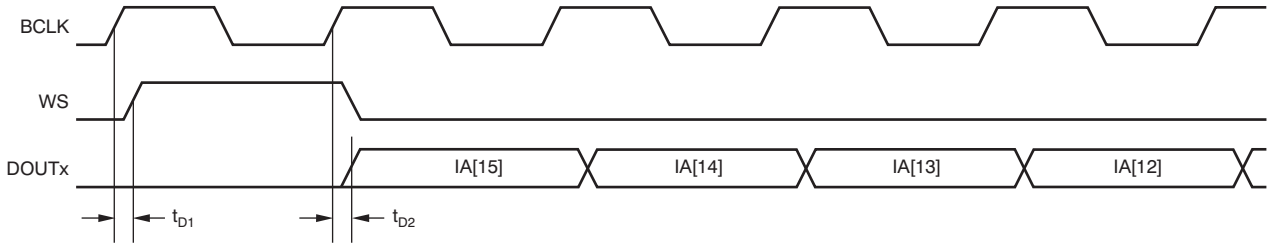


Figure 2. Primary Data Interface Timing

SPI Control Interface Timing

PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
f_{SCK}	Maximum SCK frequency				20	MHz
t_L	$\overline{CS}$ lead time	Trailing $\overline{CS}$ to leading SCK	5.0			ns
t_T	$\overline{CS}$ trail time	Trailing SCK to leading $\overline{CS}$	5.0			ns
t_I	$\overline{CS}$ idle time	Leading $\overline{CS}$ to trailing $\overline{CS}$	5.0			ns
t_{SU3}	MOSI to SCK setup time		5.0			ns
t_{H3}	MOSI to SCK hold time		1.0			ns
t_{D4}	SCK to MISO delay		1.0		10.4	ns

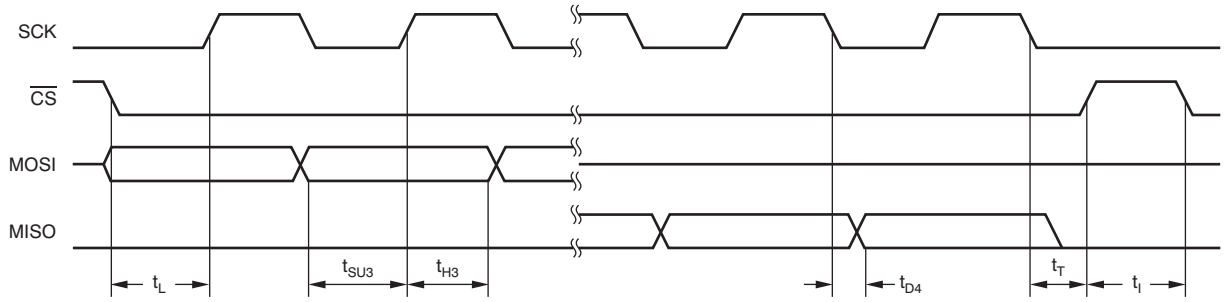


Figure 3. SPI Control Interface Timing

I²C Bus Interface Timing

PARAMETER		MIN	TYP	MAX	UNIT
f_{SCK}	SCK clock frequency	0		400	kHz
V_{IL}	Input voltage, low			0.3 × V_{DD}	V
V_{IH}	Input voltage, high	0.7 × V_{DD}			V
t_{START}	Setup time for START or repeated START condition	0.6			µs
t_{STOP}	Setup time for STOP condition	0.6			µs
t_{LOW}	LOW period of SCK clock	1.3			µs
t_{HIGH}	HIGH period of SCK clock	0.6			µs
t_{HD_DAT}	Data hold time from SCK falling	100(1)			ns
		250			
t_{SU_DAT}	Data setup time to SCK rising	100(1)			ns
		250			
t_{BUF}	Bus free time between a STOP and START condition	4.7			µs

(1) Valid when MCLK > 20 MHz; otherwise, is 250 ns.

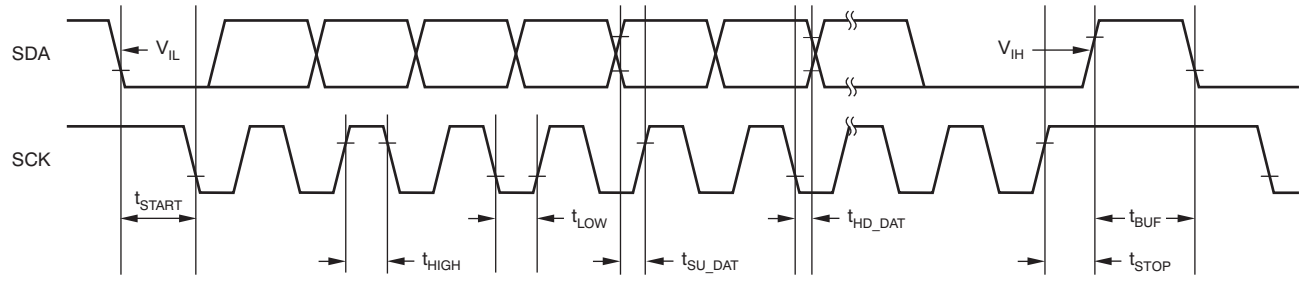


Figure 4. I²C Bus Interface Timing

DETAILED DESCRIPTION

Reset Pins

The AFE8221 has two active-low reset pins, $\overline{\text{GRST}}$ and $\overline{\text{RST}}$. When $\overline{\text{GRST}}$ is brought low, all registers on the chip are brought to default values (0, unless otherwise specified). When $\overline{\text{RST}}$ is brought low, all registers are brought to default values except for:

- Real-time clock registers (counters and alarms)
- Registers to configure the WAKEUP interrupt
- Registers controlling the GPIO pins

These registers are left in the previously programmed states.

Analog Supply Connections

A clean 3.3-V analog supply should be connected to all AVDD pins (37–40, 45, 54, 125, 134, and 141–144). Limited decoupling is required on the AVDD pins; a 0.1-μF capacitor near pins 45 and 54 and another capacitor near pins 125 and 134 should suffice.

The AFE8221 contains an internal analog switch that is used to disconnect power from the major analog blocks when the PWD pin is high. When the PWD pin is low, the AVDD1 pins (8, 12, 13, 23, 24, and 34) are internally connected to the AVDD pins (37–40 and 141–144). Since the AVDD1 pins are actually the active supply pins for the IF ADC and other analog components, the AVDD1 pins should be heavily bypassed with a minimum of parallel 0.1-μF and 0.01-μF ceramic capacitors at each pin (or pin pair).

Digital Supply Connections

The digital supply connections depend on whether the onboard regulators are used to generate the 1.8-V digital core voltage ($\overline{\text{REG\_ENB}}$ low); or if the digital core voltage comes from a system-level supply ($\overline{\text{REG\_ENB}}$ high). In either case, all IOVDD pins should be connected to the 3.3-V I/O supply and appropriately bypassed. If the internal regulators are used, this supply also sources the current drawn by the digital core.

External 1.8-V Core Supply

If an external 1.8-V supply is used, all DVDD pins should be connected to the 1.8-V supply and appropriately bypassed with 0.1-μF and 0.01-μF capacitors. DVDD1 and DVDD2 pins may also be connected directly to the 1.8-V supply or may be optionally connected through a small (1 Ω to 10 Ω) series resistor to reduce supply noise coupling into the MCLK oscillator (powered through DVDD1) or the RTC oscillator (powered through DVDD2).

When using an external supply, the PWD pin disables the MCLK oscillator when high, shutting off the clock to most of the digital core. As long as the external 1.8-V supply is maintained, all register settings in the digital core are maintained when PWD is high.

Internal 1.8-V Regulator

If the internal 1.8-V regulator is used, then 0.1-μF and 0.01-μF decoupling capacitors should still be put at the DVDD, DVDD1, and DVDD2 pins. DVDD2 should still be connected to the DVDD pins either directly or through a small series resistor. DVDD1 must be isolated from DVDD and DVDD2.

While using the internal regulators, the MCLK oscillator and the internal regulators are disabled when the PWD pin is high. This condition causes most of the register settings to be lost, except for the registers associated with the real-time clock, GPIO, and WAKEUP interrupt. For this reason, the $\overline{\text{RST}}$ pin should be brought low prior to bringing the PWD pin low (to come out of power-down). The $\overline{\text{RST}}$ pin should be held low for at least 10 ms after PWD goes low to allow the internal regulators to stabilize.

Note that the internal regulators are linear regulators, and therefore are relatively inefficient. Power dissipation as a result of the digital core almost doubles when the internal regulators are used (same core current, but drawn from a 3.3-V supply instead of a 1.8-V supply). Whenever possible, the use of a more efficient external switching regulator is encouraged in order to minimize overall system power as well as to reduce the thermal stress on the AFE8221.

Control Interface

Configuration and control data are written to the AFE8221 via the control interface. The control interface supports two protocols, SPI and I^2C. If the CTRL_MODE pin is tied low, then an SPI interface is implemented. If CTRL_MODE is tied high, then an I^2C protocol-compatible interface is implemented.

SPI Interface

The SPI interface consists of four signals: a serial clock (SCK), an active-low chip select ($\overline{CS}$), a serial data input (MOSI—master out, slave in), and a serial data output (MISO—master in, slave out). Data are transferred in groups of 32 bits. The first 16 bits are the instruction, which indicates:

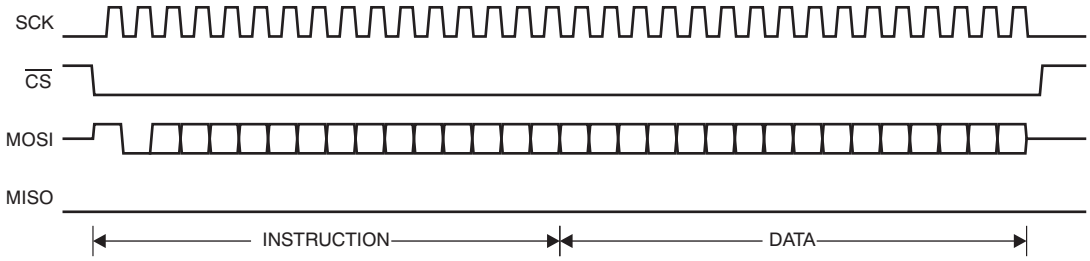
1. If data are to be written or to be read;
2. If the data target is a register or RAM; and
3. The address of the data target.

The second 16 bits are the data transfer, which is input on MOSI for a write cycle or output on MISO for a read cycle.

Figure 5 shows an SPI write cycle. The cycle is initiated by the high-to-low transition of the $\overline{CS}$ line. 32 SCK pulses clock the instruction and the data into the MOSI line. Data are clocked in MSB first. The first 16 bits are the instruction. There are two possible write cycle instructions: register write and memory write. The formats for these instructions are shown in Figure 6 and Figure 7, respectively.

The only information required for a register write is the seven-bit register address (REG_ADDR). For a memory write, both the five-bit memory select (MEM) plus the six-bit memory address (MEM_ADDR) are required.

Following the 16-bit instruction, the 16-bit data word is clocked in, again MSB first. At the end of the write cycle, this data word is written to the appropriate register or memory location in the AFE8221.



NOTE: To terminate a Write/Read cycle, $\overline{CS}$ must be brought high.

Figure 5. SPI Control Interface Write Cycle

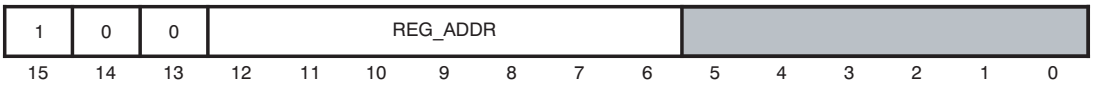


Figure 6. Register Write Instruction Format

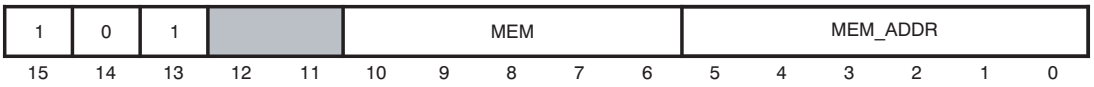


Figure 7. Memory Write Instruction Format

Figure 8 shows the SPI interface read cycle. It is similar to the write cycle, except that instead of the data word being clocked into MOSI during the second half of the cycle, the data word is clocked out of MISO. Note that only register reads are permitted; RAM reads cannot be read back.

For reading and writing, data block transfers are supported. For a block transfer, multiple data words are transmitted following the memory read or write instruction. The data words are read from or written sequentially starting at the address contained in the instruction. The sequential access terminates when the $\overline{CS}$ line goes high. Figure 9 shows a register block read cycle. In the illustration, three succeeding register locations are read starting at address N. The block write cycle is similar except, of course, data are clocked into MOSI.

In all cases, the control interface is reset when $\overline{CS}$ goes high. If the final SCK is not received before $\overline{CS}$ goes high, then the cycle ends prematurely. For a read cycle, data transfer terminates; for a write cycle, no data are written to either a register or to memory.

I²C Slave Interface

The AFE8221 control interface can be configured to provide I²C slave operation. It has a 10-bit slave address of *00010010AB* and complies with the Philips I²C specification. Note that address bits A and B are determined by the state of the I²C address pins A1 and A0. The mapping of SPI pins to I²C pins is shown in Table 1.

Table 1. SPI/I²C Pin Mapping

CTRL_MODE = 0 (SPI)	CTRL_MODE = 1 (I²C)
Chip select ($\overline{CS}$)	I²C address bit (A1)
Master out slave in (MOSI)	I²C address bit (A0)
Master in slave out (MISO)	Serial data line (SDA)
SPI clock (SCK)	Serial clock line (SCL)

The AFE8221 I²C interface supports both fast mode (400K bits/sec) and standard mode (100K bits/sec) operation. However, if the master crystal frequency is less than 20 MHz, then only standard mode is supported.

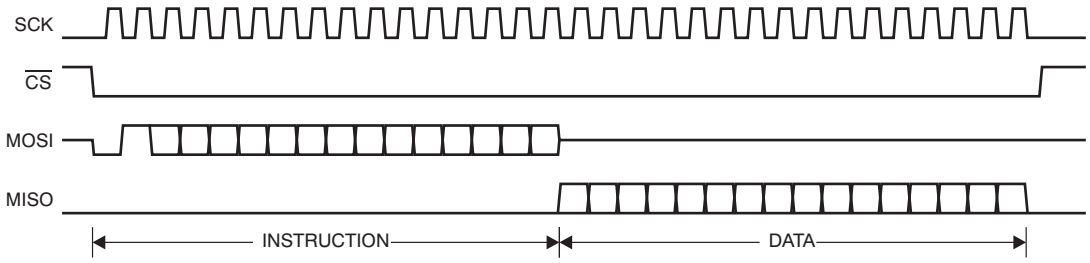


Figure 8. SPI Control Interface Read Cycle

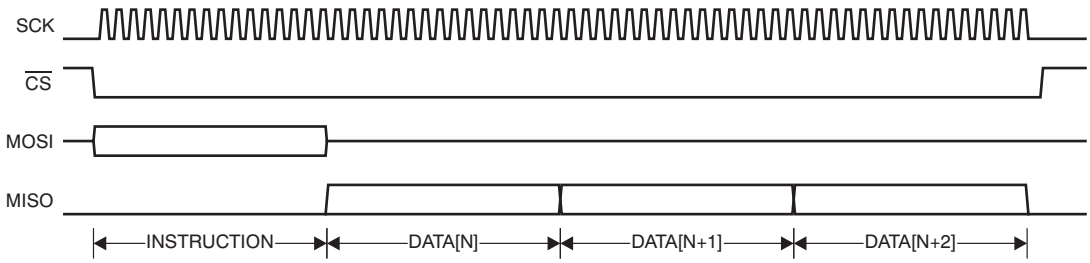


Figure 9. SPI Control Interface Block Read Cycle

As a reference, a typical data transfer on the I²C bus is described in Figure 10. Each data byte is eight bits long and must be followed by an Acknowledge bit. Start and stop conditions are defined as a transition of the SDA signal with SCL high. A pulse of the SCL clock signal indicates the transfer of data or an Acknowledge bit on the SDA pin. The transmitting device drives SDA data during clock periods 1–8. The receiving device acknowledges by driving SDA low during clock 9. Master devices always generate the SCL clock and initiate transactions. Refer to the Philips I²C Bus Specification for further details.

The AFE8221 has 16-bit internal registers and operates on 16-bit instructions. Because the I²C interface is inherently an 8-bit interface, special formats are required to send instructions and data between an I²C Master and the AFE8221. The I²C Write Operation and I²C Read Operation sections describe these formats in detail.

I²C Write Operation

Write operations require a start condition followed by two bytes describing both a 10-bit address format and the AFE8221 10-bit slave address. The next two bytes must contain the 16-bit instruction word format described previously in Figure 6 or Figure 7, depending on the internal resource being addressed. Finally, a pair of bytes containing the 16-bit write data must be provided. If additional 16-bit writes are required, further pairs of bytes may be used as part of a block transfer. After the final pair of write data bytes, an I²C stop condition must be provided to terminate the transaction. Figure 12 illustrates a block write transfer of *N* 16-bit data words. Gray areas denote slave-driven SDA cycles; white areas are master-driven.

I²C Read Operation

Read operations require a start condition followed by two bytes describing both a 10-bit address format and the AFE8221 10-bit slave address. The next two bytes must contain the 16-bit instruction word format, as illustrated in Figure 11. A repeated start followed by the first byte of the slave address is then required to create a combined transaction. Note that the R/$\overline{W}$ bit is set to 1 (read), indicating that subsequent bytes are to be read from the slave. The AFE8221 presents addressed 16-bit data words in 8-bit pairs until a NACK (N) is provided by the master. After the final pair of read data bytes, an I²C stop condition must be provided to terminate the transaction. Figure 13 illustrates a block read transfer of *N* 16-bit data words. Gray areas denote slave-driven SDA cycles; white areas are master-driven.

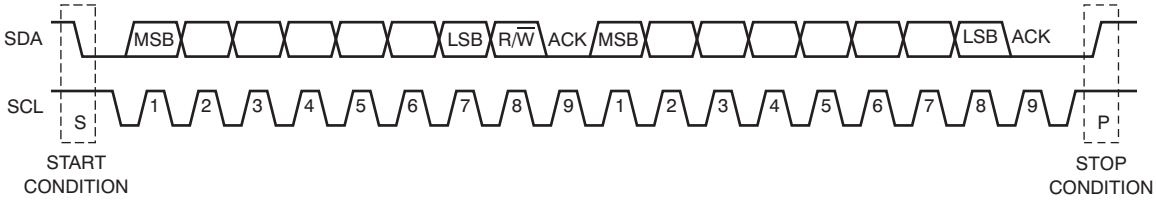


Figure 10. Example Data Transfer on the I²C Bus

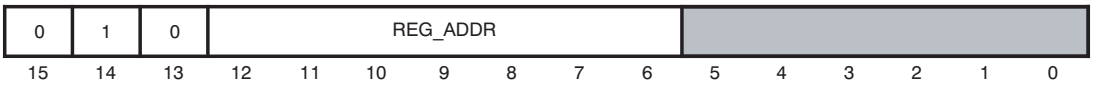


Figure 11. Register Read Instruction Format

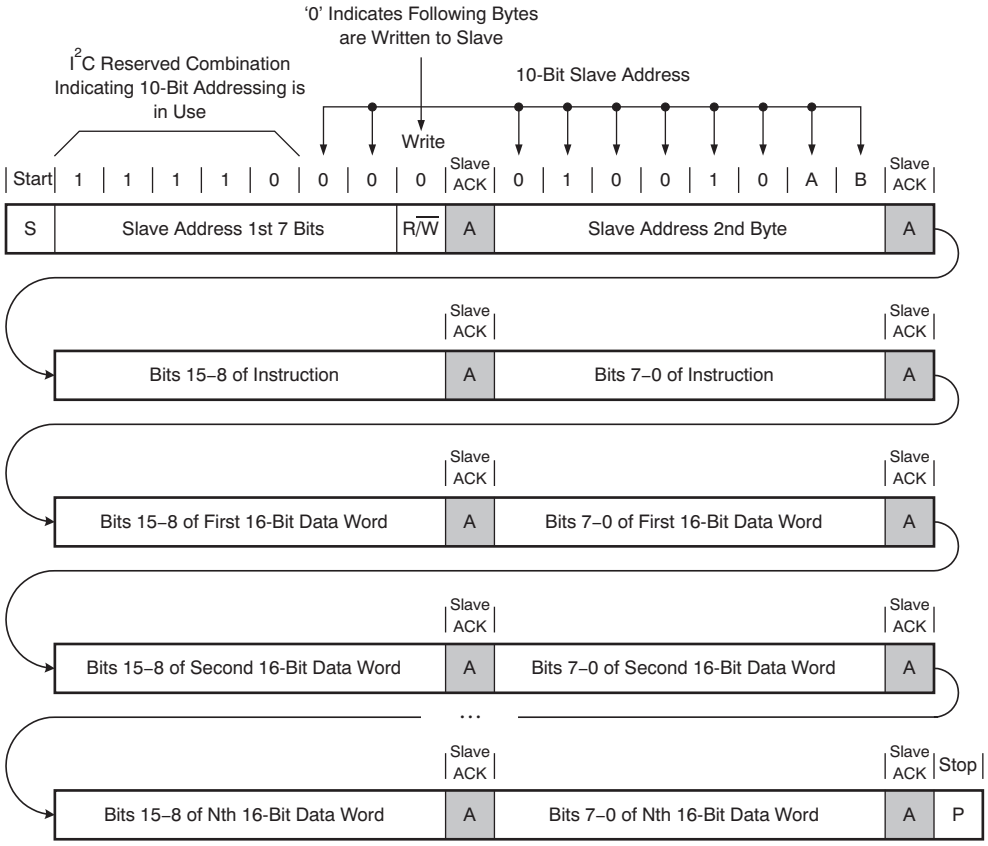


Figure 12. Example I²C Write Operation

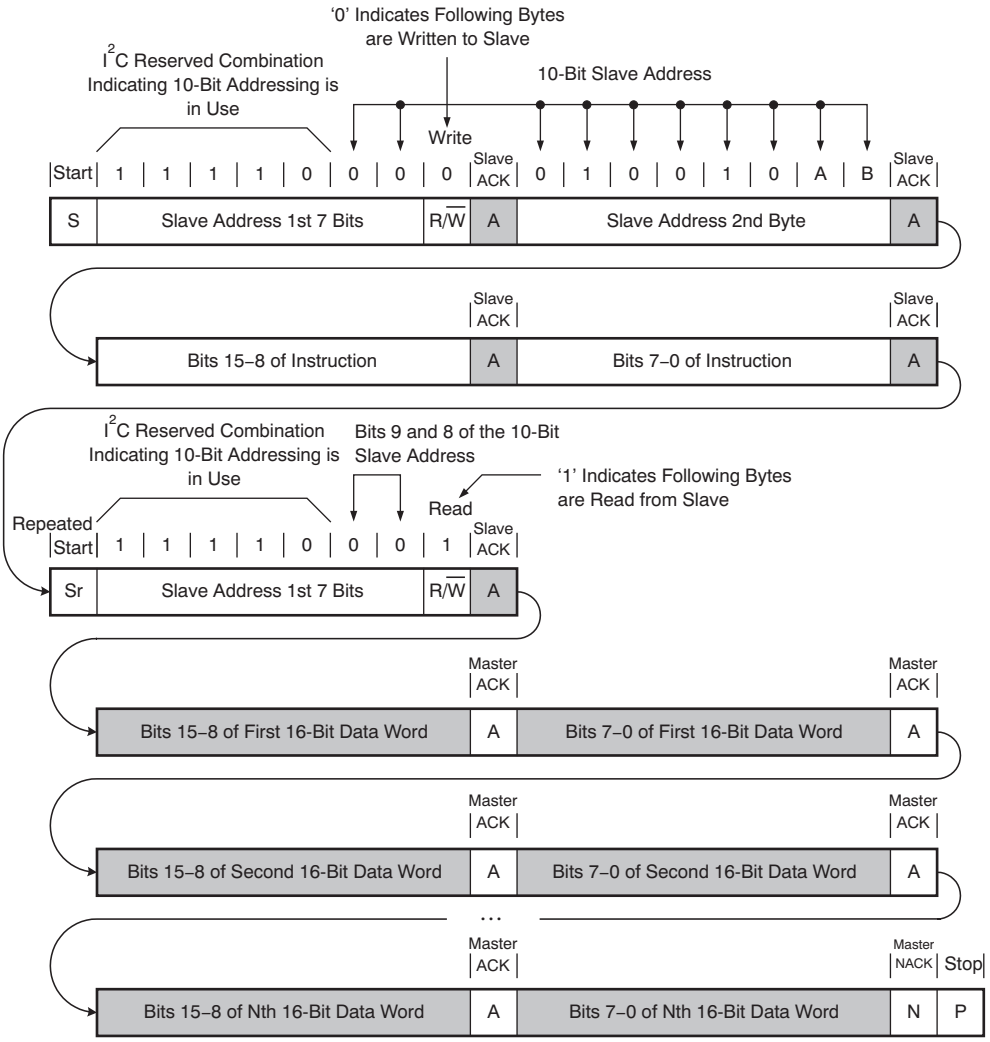


Figure 13. Example I²C Read Operation

IF Analog-to-Digital Converters (IF_ADC0 and IF_ADC1)

IF_ADC0 and IF_ADC1 are 12-bit pipeline ADCs that are used to sample the output of the tuner(s). Figure 14 shows recommended connections for the IF ADCs.

The IF ADCs have three power modes controlled by *ifadc_en[0]* and *ifadc_en[1]*. Full-power mode occurs when both *ifadc_en[0]* and *ifadc_en[1]* are high. In this case, both ADCs are biased to the highest levels and are ready to operate. If only *ifadc_en[0]* or *ifadc_en[1]* is high, then the converters are operating in reduced-power mode, where the enabled ADC is fully biased and ready to operate while the second ADC is in a low (but not zero) bias state (a minimum bias current is necessary to maintain safe voltages within the ADC core). In low-power mode, both *ifadc_en[0]* and *ifadc_en[1]* are low. In this case, all IF ADC circuits are in the minimum bias mode. Note that to reach a true sleep mode, the analog supply to the IF ADC block must be turned off.

When *ifadc_gain0* is low, IF_ADC0 is in its normal 1x gain operating state. If *ifadc_gain0* is high, then the gain of IF_ADC0 is changed to 2x. In a similar fashion, *ifadc_gain1* controls the gain of IF_ADC1. Table 2 shows the *ifadc_en* and *ifadc_gain* control variable parameters.

Table 2. IF_ADC Control Register Settings

PARAMETER	ADDRESS	BITS
ifadc_en[0]	1	0
ifadc_en[1]	1	1
ifadc_gain0	1	2
ifadc_gain1	1	3

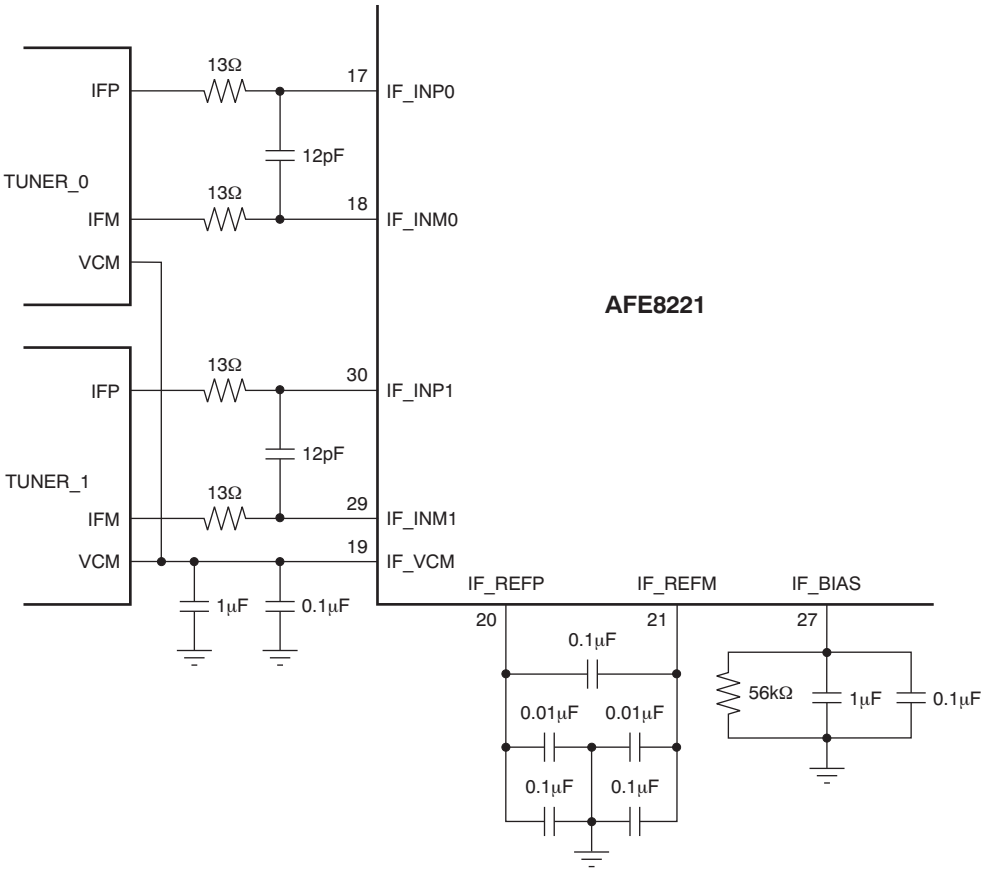


Figure 14. IF ADC Connections

IF ADC Alarm/Attenuator

The output of each IF ADC is monitored to ensure that the full-scale input range is not exceeded. If an ADC over-range condition occurs, an overflow signal is generated that may be used to generate an interrupt on the IRQ line, depending on the settings in the IRQ interrupt generator.

In addition, programmable limits may be set for each IF ADC. If the absolute value of IF_ADC0 exceeds *if_adc_limit0* or the absolute value of IF_ADC1 exceeds *ifadc_limit1*, then an interrupt may be generated on IRQ again depending on the settings in the IRQ interrupt generator.

In the case of an IF ADC event, the IRQ status register can be read back to determine the type of event and on which ADC channel it occurred. The IRQ status register can be polled to determine if an IF ADC event has occurred in the case where IF ADC events are masked from generating an interrupt.

The control variable *ddc0_atten* causes an attenuation of the IF_ADC0 output prior to the DDC. The attenuation ranges in 3-dB steps from 0 dB (for *ddc0_atten* = 0) to –18 dB (for *ddc0_atten* = 6). *ddc1_atten* has the same effect on the output of IF_ADC1.

To better synchronize the IF ADC attenuator with the tuner automatic gain control (AGC), a delay may be programmed between when a new value of *ddc_atten* is written and when it takes effect. When a new value of *ddc0_atten* is written, a counter (driven by MCLK) is initialized to *ddc0_delay*. When the counter reaches zero, the actual attenuation change occurs. Likewise, *ddc1_delay* affects *ddc1_atten*. Note that if a new *ddc0_atten* is written before the delay counter has reached zero from the previous write, the previous write is discarded. Table 3 shows the attenuator, delay, and limit control variables.

Table 3. IF ADC Control Register Settings

PARAMETER	ADDRESS	BITS
ddc0_atten	3	2:0
ddc1_atten	15	2:0
ddc0_delay	4	15:0
ddc1_delay	16	15:0
ifadc_limit0	46	11:0
ifadc_limit1	47	11:0

Digital Downconverter 0 (DDC0)

DDC0 operation is controlled by *ddc_en[0]*. When *ddc_en[0]* is 1, operation of DDC0 is enabled. If *ddc_en[0]* is 0, operation of DDC0 is disabled. Table 4 shows the DDC0 operation control settings.

Table 4. DDC Control Register Settings

PARAMETER	ADDRESS	BITS
ddc0_cic_dec_rate	9	8:0
ddc0_cic_scale	10	11:6
ddc0_cic_shift	10	5:0
ddc0_demod_freq[31:16]	5	15:0
ddc0_demod_freq[15:0]	6	15:0
ddc0_demod_phase[31:16]	7	15:0
ddc0_demod_phase[15:0]	8	15:0
ddc0_fir1_base_address	11	13:8
ddc0_fir1_mode	11	1:0
ddc0_fir1_ncoeffs	11	7:2
ddc0_fir1_nodec	14	9
ddc0_fir2_nodec	14	10
ddc0_fir2a_base_address	12	15:9
ddc0_fir2a_mode	12	1:0
ddc0_fir2a_ncoeffs	12	8:2
ddc0_fir2a_shift	14	3:0
ddc0_fir2b_base_address	13	15:9
ddc0_fir2b_mode	13	1:0
ddc0_fir2b_ncoeffs	13	8:2
ddc0_fir2b_shift	14	7:4
ddc0_interleave	14	8
ddc_en[0]	1	4
ddc_sync	1	6

Quadrature Mixer/NCO

The NCO frequency and initial phase are set by the 32-bit unsigned variables *ddc0_demod_freq* and *ddc0_demod_phase*. The I and Q outputs of the mixer can be calculated by Equation 1 and Equation 2.

$$I = ADC \times \cos(2\pi ft + \phi) \quad (1)$$

$$Q = ADC \times \sin(2\pi ft + \phi) \quad (2)$$

where ADC is the output of the IF analog-to-digital converter, f is the NCO phase offset (in radians) given by Equation 3, and ϕ is the NCO phase offset (in radians) given by Equation 4.

$$f = f_{MCLK} \frac{ddc0\_demod\_freq}{2^{32}} \quad (3)$$

$$\phi = 2\pi \frac{ddc0\_demod\_phase}{2^{32}} \quad (4)$$

The *ddc_sync* signal can be used to control the phase of the mixer. While the *ddc_sync* signal is high, the phase accumulator is held to a constant value *ddc0_demod_phase*, essentially holding it to 0 in Equation 1 and Equation 2. When the *ddc_sync* signal is brought low, the phase accumulator is incremented by the value *ddc0_demod_freq* once per MCLK cycle.

CIC Filter

The first stage of decimation filtering is provided by a fifth-order CIC filter. The operation of the CIC filter is controlled by the unsigned variables *ddc0_cic_dec_rate*, *ddc0_cic_scale*, and *ddc0_cic_shift*. The valid range for *ddc0_cic_dec_rate* is from 4 to 256.

The inherent dc gain of the CIC filter is *ddc0_cic_dec_rate.* The control variables *ddc0_cic_shift* and *ddc0_cic_scale* are used to reduce this very high gain before the signal is output to the next stage of the decimation filter. The combined effect of *ddc0_cic_dec_rate*, *ddc0_cic_shift*, and *ddc0_cic_scale* produces an overall dc gain for the CIC filter of Equation 5.

$$GAIN = ddc0\_cic\_dec\_rate^{5} \frac{ddc0\_cic\_scale/32}{2_{ddc0\_cic\_shift}} \quad (5)$$

In general, *ddc0_cic_shift* and *ddc0_cic_scale* should be chosen to make GAIN as close to 1 as possible. For example, if *ddc0_cic_dec_rate* is 20, setting *ddc0_cic_shift* to 22 and *ddc0_cic_scale* to 41 results in a GAIN of 0.9775.

First FIR Filter

The block following the CIC filter is a decimate-by-two finite impulse response (FIR) filter with programmable coefficients. *ddc0_fir1_mode* sets the type of filter response—ODD (MODE = 00: symmetric impulse response, odd number of taps), EVEN (MODE = 01: symmetric impulse response, even number of taps), HALFBAND (MODE = 10), and ARBITRARY (MODE = 11: non-symmetric impulse response).

The 16-bit wide filter coefficients are stored in memory bank 0. Up to 64 coefficients can be stored in this memory. Depending on the types of filters desired and the number of taps, coefficients for multiple filter responses may be stored in the memory bank. The filter response may be changed simply by updating the control register with new values for *ddc0_fir1_mode*, *ddc0_fir1_ncoeff*, and *ddc0_fir1_base_addr*.

ddc0_fir1_ncoeff defines the number of unique filter coefficients that make up the filter response. *ddc0_fir1_base_addr* defines the memory location where the first filter coefficient is stored. The actual filter length is a function of the *ddc0_fir1_mode* and *ddc0_fir1_ncoeff*, as shown in Equation 6.

Filter Length = 2 × (ddc0_fir1_ncoeff – 1) + 1 for ODD
Filter Length = 2 × ddc0_fir1_ncoeff for EVEN
Filter Length = 4 × (ddc0_fir1_ncoeff – 1) + 1 for HALFBAND
Filter Length = ddc0_fir1_ncoeff for ARBITRARY (6)

The maximum filter length that can be realized is limited by two factors. First, the number of clock cycles between successive filter outputs limits the number of coefficients that can be processed, as shown in Equation 7.

$$ddc0\_fir1\_ncoeff \le 2 \times ddc0\_cic\_dec\_rate \quad (7)$$

where *ddc0_cic_dec_rate* is the decimation ration of the CIC filter.

Second, the size of the data memory (which stores incoming data samples) limits filter length to 62 taps. Note that two data memory locations are required to filter processing.

The dc gain of the FIR filter depends on the coefficient values and the filter mode. For ODD mode and HALFBAND mode, the dc gain is given by Equation 8:

$$GAIN = \left(\frac{h_{NCOEFF} + \sum_{n=1}^{NCOEFF-1} 2h_n}{2^{15} - 1} \right) \quad (8)$$

where h_n is the n^{th} of NCOEFF filter coefficients stored in memory.

For EVEN mode the, dc gain is shown by Equation 9:

$$GAIN = \left(\frac{\sum_{n=1}^{NCOEFF} 2h_n}{2^{15} - 1} \right) \quad (9)$$

while for ARBITRARY mode the gain is shown by Equation 10:

$$GAIN = \left(\frac{\sum_{n=1}^{NCOEFF} h_n}{2^{15} - 1} \right) \quad (10)$$

Second FIR Filters

The first FIR filter is followed by two parallel second FIR filters, FIR2A and FIR2B. Duplicate filters allow the output of two I and Q output streams with different bandwidths. For example, the bandwidth of FIR2A may be set wide to accommodate reception of digital broadcasts, while FIR2B may be set narrower to receive an analog broadcast sharing the same band. Coefficients for FIR2A are stored in memory bank 1 (MEM = 1) and coefficients for FIR2B are stored in memory bank 2 (MEM = 2).

The operation of the second FIR filter is similar to the first FIR filter with several notable exceptions. First, the depths of the coefficient and data memories are doubled to 128. This size increase allows for filters up to 126 taps to be realized without running out of data memory. It also allows longer sets of filter coefficients to be stored in coefficient memory.

Second, because of the additional decimation by two from the first FIR filter, twice as many MCLK cycles are available to process coefficients, increasing the maximum allowable value of NCOEFF, as shown in Equation 11 and Equation 12.

$$ddc0\_fir2a\_ncoeff \le 4 \times ddc0\_cic\_dec\_rate \quad (11)$$

$$ddc0\_fir2b\_ncoeff \le 4 \times ddc0\_cic\_dec\_rate \quad (12)$$

Third, in the first FIR filter the total of all the filter tap weights must add up to (2^{15} – 1) to achieve unity gain through the filter. With longer filters (and therefore, smaller coefficients), frequency response errors may be introduced as a result of coefficient truncation. A Shift parameter has been added to the second FIR filter to alleviate this problem. The total of all filter tap weights must add up to ($2^{15+ddc0\_fir2a\_shift}$ – 1) to achieve unity gain through FIR2A (similarly for *ddc0_fir2b_shift* and FIR2B). Note that shift values for FIR2A and FIR2B can be set separately.

Extended-Length Filter Mode

If FIR2A or FIR2B cannot provide enough filter taps to achieve the desired frequency response, setting control bit *ddc0_interleave* puts the two filters into an interleaved mode that doubles the length of the filter that can be realized. However, there are several limitations:

1. Only odd symmetrical filters may be realized;
2. The filter length M must be such that (M + 1)/4 is an integer; and
3. Only one filter can be realized (in *ddc_interleave* mode the A and B outputs are identical: IB = IA and QB = QA).

In addition to setting the *ddc0_interleave* bit, FIR2A must be set to EVEN mode and FIR2A must be set to ODD mode. *ddc0_fir2a_ncoeff* and *ddc0_fir2b_ncoeff* are both set to (M + 1)/4. *ddc0_fir2a_shift* and *ddc0_fir2b_shift* should be identical. There are no restrictions on *ddc0_fir2a_base_addr* or *ddc0_fir2b_base_addr*.

The M-tap filter has (M + 1)/2 unique coefficients. The first, third, fifth, etc. coefficients are loaded into the FIR2A coefficient memory; the second, fourth, sixth, etc. coefficients are loaded into the FIR2B memory. The center coefficients of the filter end up as the last coefficient loaded into FIR2B.

FIR Filter Transfer Functions

Equation 13 to Equation 21 show transfer functions and dc gain for the various filter modes. Generic names for the control variables are used; just substitute the appropriate variable (that is, *ddc0_fir2a_ncoeff* for NCOEFF) as necessary. Also, note that SHIFT has a value of 0 for FIR1.

Basic Filter Modes

$$H_{EVEN}(z) = \sum_{n=0}^{NCOEFF-1} COEFF_{BASE\_ADDR+n} \times (z^{-n} + z^{-(2 \times NCOEFF - 1 - n)}) \quad (13)$$

$$H_{ODD}(z) = \sum_{n=0}^{NCOEFF-2} COEFF_{BASE\_ADDR+n} \times (z^{-n} + z^{-(2 \times NCOEFF - 2 - n)}) + COEFF_{BASE\_ADDR+NCOEFF-1} \times z^{NCOEFF-1} \quad (14)$$

$$H_{HALFBAND}(z) = \sum_{n=0}^{NCOEFF-2} COEFF_{BASE\_ADDR+n} \times (z^{-2n} + z^{-(4 \times NCOEFF - 6 - 2n)}) + COEFF_{BASE\_ADDR+NCOEFF-1} \times z^{2 \times NCOEFF - 3} \quad (15)$$

$$H_{ARBITRARY}(z) = \sum_{n=0}^{NCOEFF-1} COEFF_{BASE\_ADDR+n} \times z^{-n} \quad (16)$$

$$GAIN_{EVEN}(z) = 2^{-SHIFT} \times \frac{2 \times \sum_{n=0}^{NCOEFF-1} COEFF_{BASE\_ADDR+n}}{2^{15} - 1} \quad (17)$$

$$GAIN_{ODD} = GAIN_{HALFBAND} = 2^{-SHIFT} \times \frac{2 \times \sum_{n=0}^{NCOEFF-2} COEFF_{BASE\_ADDR+n} + COEFF_{BASE\_ADDR+NCOEFF-1}}{2^{15} - 1} \quad (18)$$

$$GAIN_{ARBITRARY}(z) = 2^{-SHIFT} \times \frac{\sum_{n=0}^{NCOEFF-1} COEFF_{BASE\_ADDR+n}}{2^{15} - 1} \quad (19)$$

Extended-Length Filter Mode

$$H_{EXTENDED}(z) = 2^{-SHIFT} \times \left(\begin{array}{l} \sum_{n=0}^{NCOEFF-1} COEFF\_A_{BASE\_ADDR\_A+n} \times (z^{-2\times n} + z^{-2\times(2\times NCOEFF-1-n)}) \\ + \sum_{n=0}^{NCOEFF-2} COEFF\_B_{BASE\_ADDR\_B+n} \times (z^{-2\times n+1} + z^{-2\times(2\times NCOEFF-2-n)+1}) \\ + COEFF\_B_{BASE\_ADDR\_B+NCOEFF-1} \times z^{2\times NCOEFF} \end{array} \right) \quad (20)$$

$$GAIN_{EXTENDED} = \frac{2^{-SHIFT}}{2^{15}-1} \times \left(\begin{array}{l} 2 \times \sum_{n=0}^{NCOEFF-1} COEFF\_A_{BASE\_ADDR\_A+n} \\ + 2 \times \sum_{n=0}^{NCOEFF-2} COEFF\_B_{BASE\_ADDR\_B+n} \\ + COEFF\_B_{BASE\_ADDR\_B+NCOEFF-1} \end{array} \right) \quad (21)$$

Digital Downconverter 1 (DDC1)

The description of DDC1 is identical to the description of DDC0, with the following exceptions:

1. DDC1 is enabled by *ddc_en[1]*.
2. Control variables are prefixed with *ddc1* instead of *ddc0*.
3. FIR coefficients are stored in memory banks 3, 4, and 5 instead of 0, 1, and 2.

Table 5 shows the DDC1 operation control settings.

Primary IF Data Interface

The two DDCs produce a total of eight 16-bit output values (I and Q from each of four final-stage FIR filters). The IF data interface time-multiplexes these eight values onto four serial lines. The IF data interface also generates the necessary clock and frame sync signals to complete the interface to the DSP. The general timing of the IF data interface is shown in Figure 15.

Note that each serial line (IF_DOUT0 through IF_DOUT3) can carry up to four time-multiplexed 16-bit signals. The actual number of signals per line is limited by:

a. the frequency of IF_DCLK, which can be programmed to be the same as the IF sampling clock (MCLK), one-half the IF sampling frequency, or one-fourth the IF sampling frequency; and

b. the overall decimation ratio of the DDC that determines the frequency of IF_DFSO pulses and therefore the number of IF_DCLK cycles available to clock out data.

Table 5. IF Control Register Settings

PARAMETER	ADDRESS	BITS
ddc1_cic_dec_rate	21	8:0
ddc1_cic_scale	22	11:6
ddc1_cic_shift	22	5:0
ddc1_demod_freq[31:16]	17	15:0
ddc1_demod_freq[15:0]	18	15:0
ddc1_demod_phase[31:16]	19	15:0
ddc1_demod_phase[15:0]	20	15:0
ddc1_fir1_base_address	23	13:8
ddc1_fir1_mode	23	1:0
ddc1_fir1_ncoeffs	23	7:2
ddc1_fir1_nodec	26	9
ddc1_fir2_nodec	26	10
ddc1_fir2a_base_address	24	15:9
ddc1_fir2a_mode	24	1:0
ddc1_fir2a_ncoeffs	24	8:2

Table 5. IF Control Register Settings (continued)

PARAMETER	ADDRESS	BITS
ddc1_fir2a_shift	26	3:0
ddc1_fir2b_base_address	25	15:9
ddc1_fir2b_mode	25	1:0
ddc1_fir2b_ncoeffs	25	8:2
ddc1_fir2b_shift	26	7:4
ddc1_interleave	26	8
ddc_en[1]	1	5
ddc_sync	1	6

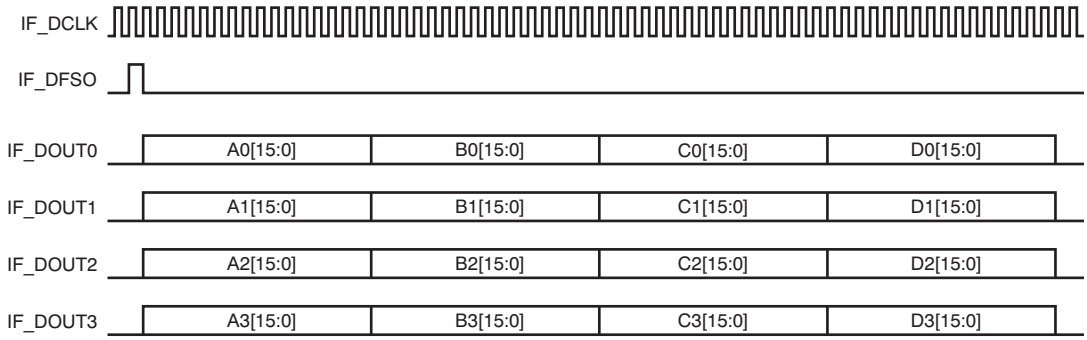


Figure 15. IF Data General Timing

Control register variables *dout0_config*, *dout1_config*, *dout2_config*, and *dout3_config*, are used to assign specific output data streams to particular time slots in the IF interface output frame. Each register is broken into four 4-bit values, each of which is used to assign the source for a given time slot according to Table 6.

Table 6. Time Slot Sources

VALUE	SOURCE
0	No source assigned
1	DDC0, FIR2A, I
2	DDC0, FIR2A, Q
3	DDC0, FIR2B, I
4	DDC0, FIR2B, Q
5	DDC1, FIR2A, I
6	DDC1, FIR2A, Q
7	DDC1, FIR2B, I
8	DDC1, FIR2B, Q

dout0_config controls the four time slots of IF_DOUT0, register 24 controls the four time slots of IF_DOUT1, and so on. The mapping of register bits to time slots is summarized in Table 7.

Table 7. Register Bit Mapping

PARAMETER	[15:12]	[11:8]	[7:4]	[3:0]
dout0_config	D0	C0	B0	A0
dout1_config	D1	C1	B1	A1
dout2_config	D2	C2	B2	A2
dout3_config	D3	C3	B3	A3

For example, bits [11:8] of *dout2_config* set the source assignment for time slot C2 of IF_DOUT2.

The control variable *if_dclk_div* sets the frequency of IF_DCLK, as shown in Equation 22 and Equation 23.

$$f_{IF\_DCLK} = \frac{f_{MCLK}}{if\_dclk\_div} \qquad if\_dclk\_div > 1 \tag{22}$$

$$f_{IF\_DCLK} = f_{MCLK} \qquad if\_dclk\_div \le 1 \tag{23}$$

Normally the data and the frame sync change on the rising edge of IF_DCLK. If *if_dclk_edge* is set to 1 then IF_DCLK is inverted so that data and frame sync change on the falling edge of IF_DCLK.

The control value *if_dfso_select* determines which DDC is responsible for generating IF_DFSO. If *if_dfso_select* is 0, then an IF_DFSO pulse is generated each time a new output is ready from DDC0. Similarly, if *if_dfso_select* is 1, then an IF_DFSO pulse is generated each time a new output is ready from DDC1. If the decimation rates of DDC0 and DDC1 are identical, then it does not matter which DDC initiates the IF_DFSO pulse. If the decimation rates are different, then the DDC with the smaller decimation ratio (higher output rate) should be chosen to generate the IF_DFSO pulse. Note that in this case, outputs from the slower DDC are repeated for multiple frames and it is the responsibility of the DSP software to compensate. This compensation is easiest to do if the higher decimation rate is an integer multiple of the lower decimation rate.

Finally, *if_dfso_mode* is used to select alternate forms of frame sync. In the default case (*if_dfso_mode* = 0), the frame sync is a high pulse one clock period wide that occurs the clock cycle before the first data bit of the serial output. If *if_dfso_mode* is set to 1, then the frame sync changes polarity once per frame; again, one clock cycle before the first data bit of the frame. If *if_dfso_mode* is set to 2, then the frame sync behaves like the default frame sync except that the sync pulse is 16 clock periods wide. The three frame sync modes are illustrated in Figure 16 and Figure 17. Table 8 shows the detailed timing conditions for Figure 17.

It is recommended that the DSP interface be configured to sample IF_DFSO and the four IF_DOUT lines on the trailing edge of IF_DCLK. Table 9 shows the *dout*, *if_dclk*, *if_dfso*, and *if_dout* operation control settings.

Table 8. Detailed Timing Conditions

PARAMETER	DESCRIPTION	MIN	TYP	MAX	UNIT
t_{D1}	IF_DCLK0 to IF_DFS0 delay	–2.9	0	3.7	ns
t_{D2}	IF_DCLK0 to IF_DOUTx delay	–3.1	0	3.8	ns

Table 9. Primary IF Control Register Settings

PARAMETER	ADDRESS	BITS
dout_en	1	7
if_dclk_div	31	4:0
if_dclk_edge	31	5
if_dfso_mode	31	8:7
if_dfso_select	31	6
if_dout0_config	27	15:0
if_dout1_config	28	15:0
if_dout2_config	29	15:0
if_dout3_config	30	15:0

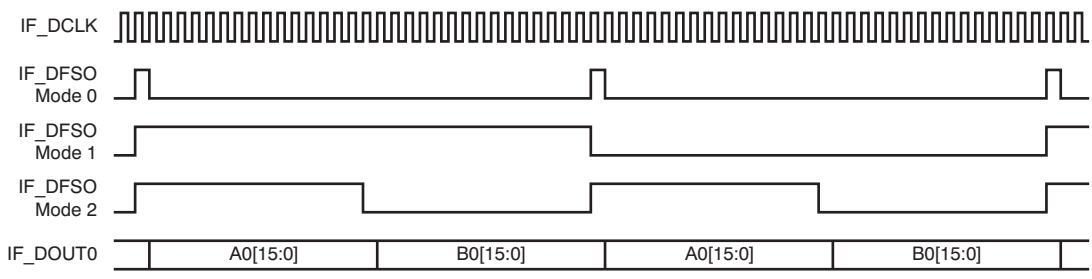


Figure 16. Frame Sync Modes

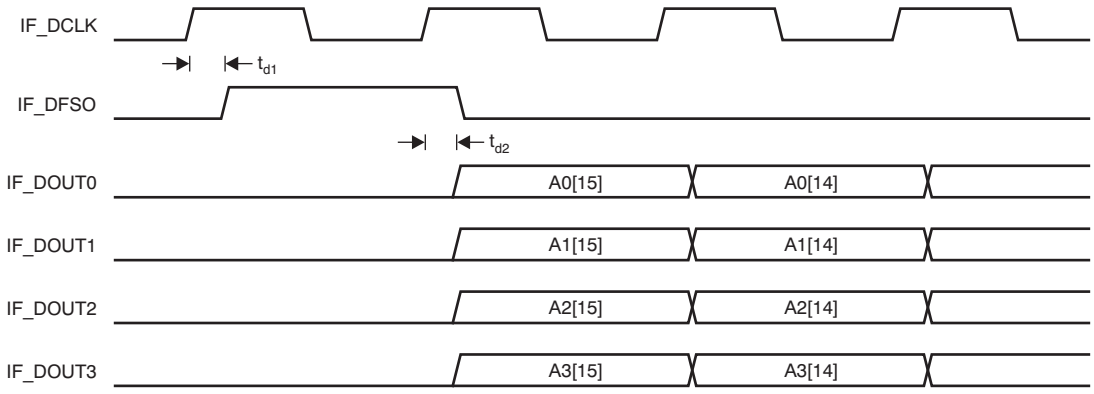


Figure 17. Detailed Timing

Alternate IF Data Interface

The operation and timing of the alternate IF data interface are identical to the primary IF data interface. Pin names are changed such that BB_BCK is equivalent to IF_DCLK; BB_WS is equivalent to IF_DFSO;` and BB_IOUT0, BB_IOUT1, BB_QOUT0, and BB_QOUT1 are each equivalent to any IF_DOUTx pins. The parameter names are also changed to reflect the different interface pin names. Table 10 shows the BB operation control settings.

Table 10. Alternate IF Control Register Settings

PARAMETER	ADDRESS	BITS
bb_dclk_edge	36	5
bb_dclk_div	36	4:0
bb_dout0_config	32	15:0
bb_dout1_config	33	15:0
bb_dout2_config	34	15:0
bb_dout3_config	35	15:0
bb_en	1	8
bb_ws_mode	36	10:7
bb_ws_select	36	6

Auxiliary DACs

CDAC0 is enabled by a high value set for *cdac_en[0]*. Similarly, CDAC1 is enabled by a high value set for *cdac_en[1]*. A control DAC that is disabled is put into a low-power state.

The control DAC outputs are set by the control variable *cdac0_out* for CDAC0 and *CDAC1_OUT* for CDAC1. A value of zero generates a 0 output from the control DAC while a value of *4095* generates a full-scale output from the control DAC. Table 11 shows the CDAC operation control settings.

Table 11. CDAC Control Register Settings

PARAMETER	ADDRESS	BITS
cdac_en[0]	1	9
cdac_en[1]	1	10
cdac0_out	37	11:0
cdac1_out	38	11:0

Auxiliary ADC

The auxiliary ADC is an 8-bit successive approximation converter that is intended for low-speed, low-accuracy tasks such as system diagnostics. Any one of four input pins can be connected to the auxiliary ADC. The parameter *aux_adc_sel* is used to connect a particular input pin to the converter. This input multiplexer operates according to the following sequence:

- aux_adc_sel = 0:
  - No aux ADC inputs are connected, all inputs high impedance
- aux_adc_sel = 1:
  - AUX_ADC0 pin connected to aux ADC
- aux_adc_sel = 2:
  - AUX_ADC1 pin connected to aux ADC
- aux_adc_sel = 3:
  - AUX_ADC2 pin connected to aux ADC
- aux_adc_sel = 4:
  - AUX_ADC3 pin connected to aux ADC

A conversion in initiated by writing to register 39 with bit 15 (*aux_adc_trig*) high. The conversion time is 8704 MCLK cycles. At the end of the conversion *auc_adc_done* goes high and the result is returned in *aux_adc_out*. As an alternative to polling *aux_adc_done*, the AFE8221 can be configured to generate an interrupt when an auxiliary ADC conversion is completed. Table 12 shows the *aux_adc* operation control settings.

Table 12. AUX_ADC Control Register Settings

PARAMETER	ADDRESS	BITS
aux_adc_done	39	15
aux_adc_out	39	7:0
aux_adc_sel	39	11:8
aux_adc_trig	39	15

Master Clock Oscillator

The master clock oscillator supports third-overtone designs from 55 MHz to 75 MHz. It can also support fundamental operations in the 20-MHz to 30-MHz range. The recommended third-overtone circuit for third-overtone operation is shown in Figure 18 and Table 13.

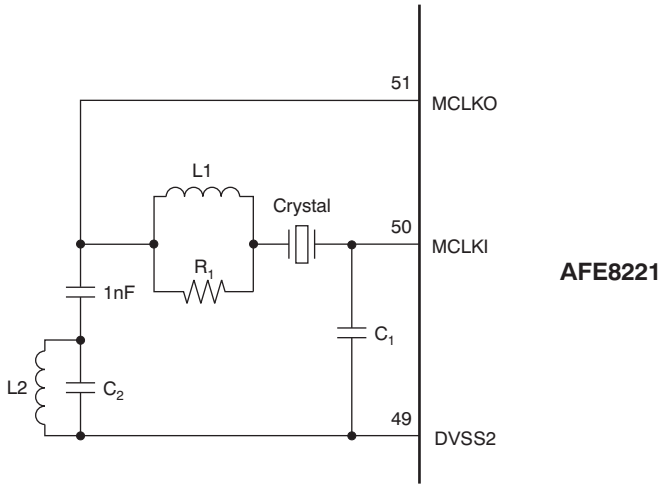


Figure 18. Third-Overtone Operation

Table 13. Third-Overtone Operation Recommendations

FREQUENCY (MHz)	C_1 (pF)	C_2 (pF)	L1 (μH)	L2 (μH)	R_1 (kΩ)
55	3	10	0.1	4.7	6.8
60	5	10	0.82	3.3	4.7
65	4	10	0.68	2.7	3.3
70	5	10	0.56	2.7	3.3
75	3	10	0.56	2.2	3.3

The master clock oscillator may be optionally divided down to provide a reference clock on the REFCLK pin. Control variable *refclk_en* enables the generation of the reference clock when high. Two variables, *refclk_hi* and *refclk_lo*, define the high and low periods of REFCLK in terms of MCLK cycles. REFCLK is high for *refclk_hi* cycles of MCLK, then low for *refclk_lo* periods of MCLK. REFCLK frequency is limited to integer submultiples of MCLK. Table 14 shows the *refclk* operation control settings.

Table 14. REFCLK Control Register Settings

PARAMETER	ADDRESS	BITS
refclk_en	1	13
refclk_hi	41	15:0
refclk_lo	40	15:0

Real-Time Clock Oscillator

The real-time clock oscillator supports crystals in the frequency range of 32.768 kHz through 150 kHz. The real-time clock module can be programmed to operate accurately with crystals in this frequency range.

The real-time clock oscillator output may be optionally output on the RTC_OUT pin when *rtc_oe* is set high. This option allows the real-time clock oscillator to be used as an alternate reference clock in the event that an acceptable frequency cannot be derived from MCLK. Table 15 shows the *rtc_oe* control setting.

Table 15. RTC Control Register Setting

PARAMETER	ADDRESS	BITS
rtc_oe	1	12

I^2C Master

The I^2C Master interface uses control variables (as shown in Table 16) and two 16-byte buffers to create I^2C bus transactions compliant with the Philips I^2C-Bus Specification Version 2.1. Both 7- and 10-bit addressing schemes are supported. Control variables supply address, data transfer direction, data burst length, and transaction control information to an I^2C master engine. This engine handles the details of the I^2C signaling and uses two 16-byte buffers to store data transferred during the transaction. A block diagram for this interface is illustrated in Figure 19.

SCL clock rates are controlled using the *i2cm_clk_cycles* control variable given by Equation 24.

$$f_{SCL} = \frac{f_{MCLK}}{4 \times i2cm\_clk\_cycles} \quad (24)$$

The interface supports both standard and fast-mode clock rates of 100 kHz and 400 kHz, respectively. Although two pairs of SCL and SDA pins are provided, the pins share a common master function. Reprogramming of the *i2cm_if_select* variable should only be performed when the *i2cm_done* status is 1, indicating that all pending I^2C transactions have completed and that it is safe to change the selected pair.

Table 16. I²C Control Register Settings

PARAMETER	ADDRESS	BITS
i2cm_10b_addressing	124	12
i2cm_allow_slave_nack	124	10
i2cm_clear_slave_nack	124	11
i2cm_clk_cycles	124	7:0
i2cm_done	124	15
i2cm_holding	124	14
i2cm_if_sel	121	14
i2cm_multimaster	124	8
i2cm_read_auto_inc	123	15
i2cm_read_byte	123	7:0
i2cm_read_byte_ptr	123	11:8
i2cm_restart_data_length	121	12:8
i2cm_restart_rw	121	13
i2cm_scl_sync_en	124	9
i2cm_slave_addr	120	14:0
i2cm_start_data_length	121	4:0
i2cm_start_rw	121	5
i2cm_use_sr	121	15
i2cm_use_stop	124	13
i2cm_write_auto_inc	122	15
i2cm_write_byte	122	7:0
i2cm_write_byte_ptr	122	11:8

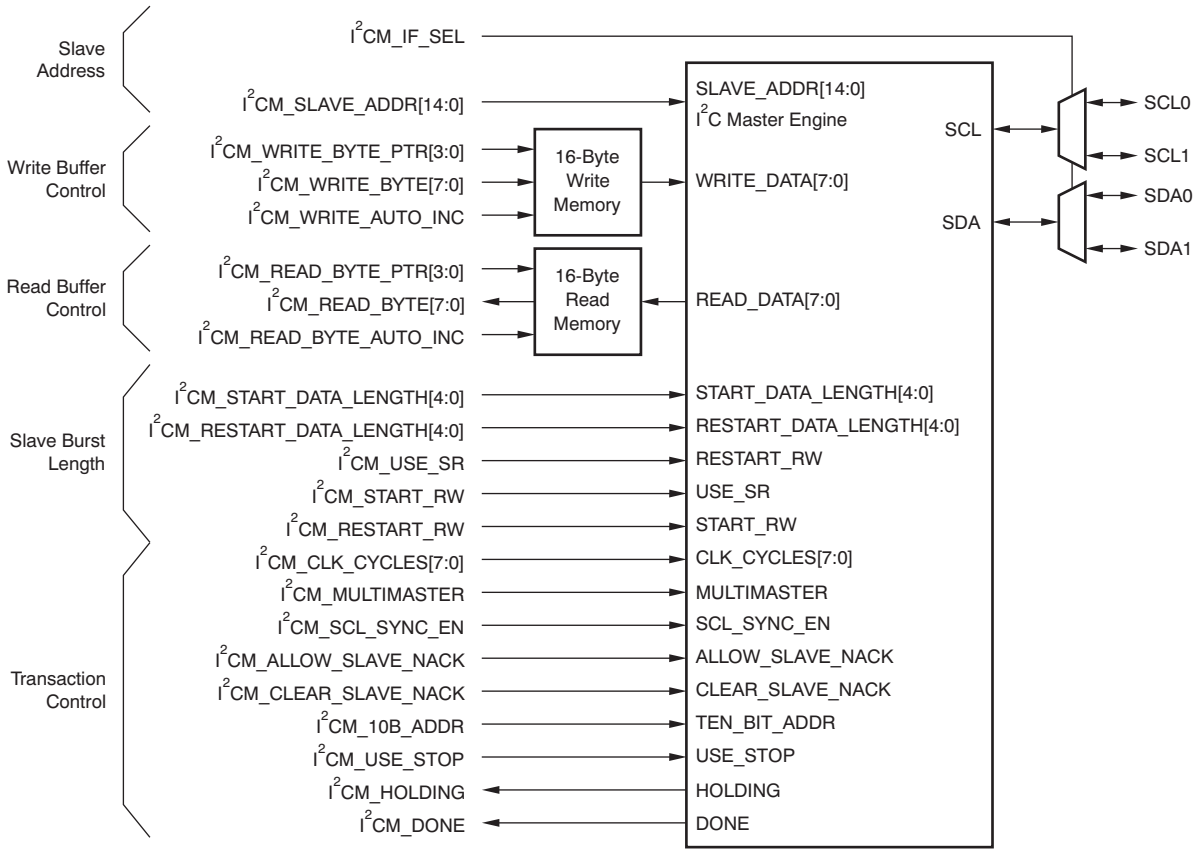


Figure 19. I²C Master Block Diagram

I²C Write Transactions

Write data must be stored in sequential locations in the write buffer starting at location zero. *i2cm_write_byte_ptr[3:0]* specifies one of the 16 memory locations where *i2cm_write_byte[7:0]* data will be written. An auto-increment feature permits the internal update of this pointer without specifying an offset for each byte after the first byte.

Once the desired write data are loaded into this memory, *i2cm_start_data_length[4:0]* must specify the number of bytes to write and *i2cm_start_rw* should be set to 0, indicating that write data will follow the address. *i2cm_10b_addr* should be set to select the desired 7- or 10-bit addressing scheme as described in the Control Register Assignments section of this document.

The write transaction is initiated by writing the slave address to *i2cm_slave_addr*. The host controller should poll the *i2cm_done* bit for a 1, indicating that the transaction has completed. The sequence of actions generated on the I²C bus are:

Start → Slave Addr → Write Data Burst → Stop

I²C Read Transactions

i2cm_start_data_length[4:0] must specify the number of bytes to read and *i2cm_start_rw* should be 1, indicating that read data will follow the address. *i2cm_10b_addr* specifies the addressing scheme.

The read transaction is initiated by writing the slave address to *i2cm_slave_addr*. The host controller should poll the *i2cm_done* bit for a 1, indicating the transaction has completed. Once completed, the read data can be extracted from the read buffer using the control variables *i2cm_read_ptr[3:0]* and *i2cm_read_byte[7:0]*. The sequence of actions generated on the bus are:

Start → Slave Addr → Read Data Burst → Stop

I²C Combined Format Transactions

The I²C specification describes combined write/read formats where a master initially transmits data to a slave and then reads data from the same slave. The *i2cm_use_sr* parameter is used to create a repeated START condition to support this format. By setting the *i2cm_use_sr* parameter to 1, the master interface can create the following sequence of actions:

Start → Slave Addr → Data Burst 1 →

Start → Slave Addr → Data Burst 2 → Stop

i2cm_start_data_length[4:0] and *i2cm_start_rw* control the data burst length and direction for DATA BURST 1. *i2cm_restart_data_length[4:0]* and *i2cm_restart_rw* control the data burst length and direction for DATA BURST 2. If the data direction is the same for both halves of the combined transaction, data are stored sequentially in the 16-byte buffer. Writing *i2cm_slave_addr* initiates the transaction.

I²C Data Bursts Greater than 16 Bytes

To create an I²C read or write burst greater than 16 bytes, the *i2cm_use_stop* parameter should be set to 0, causing the interface to pause between each burst of bytes transferred. This pause allows the host to either reload or empty the buffers, depending on the direction of data transfer.

After starting the transaction by writing *i2cm_slave_addr*, the *i2cm_holding* status bit should be monitored for a logic 1, indicating that the interface has completed the current set of byte transfers and is waiting for the host to continue. After reloading or emptying the buffers as needed, the host should rewrite *i2cm_slave_address* to continue the transfer for the next block of up to 16 bytes. For the final transfer of the long data burst, *i2cm_use_stop* must be set to 1 prior to re-writing the *i2cm_slave_address*. This configuration creates a normal STOP condition to properly terminate the transfer.

Interrupt Operation

As an alternative to polling the values of *i2cm_done* or *i2cm_use_stop*, the AFE8221 can be programmed to generate an interrupt when either of these values goes high.

Real-Time Clock

The real-time clock (RTC) is enabled by setting *rtc_en* to 1. While *rtc_en* is 0, the RTC oscillator continues to run but the RTC registers do not advance.

The RTC can operate with a range of oscillator frequencies up to 100 kHz. At the beginning of each second, 2x the value of *rtc_max_count* is loaded into the RTC crystal counter. This counter is decremented at the rate of the RTC oscillator until it hits zero, which generates a strobe that increments the seconds counter as well as re-initializes the RTC crystal counter. For a nominal 32.768-kHz clock crystal, *rtc_max_count* should be set to 16,384 (the default value); for a nominal 100-kHz crystal, *rtc_max_count* should be set to 50,000. Table 18 illustrates the RTC control variable settings.

The RTC can be coarsely calibrated by adjusting the *rtc_max_count* to an appropriate value other than half the nominal crystal frequency. If finer calibration is required, compensation mode can be enabled by setting *rtc_comp_en* to 1. In compensation mode, the two's-complement value stored in *rtc_comp_val* is added to the one-second counter when it is re-initialized at the beginning of each hour; thus, the first second of each hour is lengthened or shortened depending on the sign of *rtc_comp_val*. The compensation can be applied to several seconds at the beginning of each hour; *rtc_comp_cnt* holds the number of seconds per hour to which the compensation is applied. By spreading the compensation out over a number of seconds, the impact on the length of any given second is minimized.

Setting and Reading the RTC

Because of the need to carefully synchronize any update of the RTC time registers (*rtc_seconds*, *rtc_minutes*, etc.), they must be written in a slightly different manner than the other control registers. Time registers must be written individually; after a particular register address is written, at least two clock cycles of the RTC oscillator must pass before another register write occurs. The MSB of each time register address can be polled to determine if it is safe to make another write: if the MSB is 1, the interface is still busy and a new write should not be initiated. If the MSB is 0, then the interface is ready to accept another write. There is no limitation on reading the time registers.

Note that all time register values are BCD-encoded. Also note that the *rtc_day_of_week* is a read-only value that is internally calculated from the *rtc_day*, *rtc_month*, and *rtc_year* registers. Ranges on the various time registers are shown in Table 17. When the rtc_mode changes, the real-time clock alarm settings should also be changed to reflect the new time format. For instance, an alarm setting of 1300 hours never generates an interrupt in 12-hour mode. This setting should be reset to 1:00 PM when the mode is changed to 12-hour mode.

Table 17. Time Register Ranges

PARAMETER	RANGE
rtc_seconds	0 to 59
rtc_minutes	0 to 59
rtc_hours	1 to 12 (12-hour mode); 0 to 23 (24-hour mode)
rtc_ampm	0 (AM) or 1 (PM) 12-hour mode only
rtc_day	1 to 31, depending on month
rtc_month	1 to 12
rtc_year	0 to 99 (for years 2000 to 2099)
rtc_day_of_week	0 (Sunday) to 6 (Saturday)

Invalid combinations of *rtc_day* and *rtc_month* (trying to set February 30, for example) cause unpredictable behavior and should be avoided. The February 28/29 rollover variation based on leap year is automatically corrected for.

The RTC defaults to operate in 12-hour plus AM/PM mode. To operate in 24-hour mode (where the AM/PM bits are disabled) set *rtc_mode* to 1. Care must be taken when switching between AM/PM mode and 24-hour mode to avoid setting the time to a invalid value. See Figure 20 and Figure 21 for the proper procedures.

Real-Time Clock Alarm

The real-time clock alarm function can be used to generate an interrupt (or a wakeup interrupt) at a pre-programmed time. If the appropriate bit in an interrupt enable register is set, an interrupt will be generated when the values in the RTC time registers become equal to the values in the RTC alarm registers. The register settings are shown in Table 18.

Table 18. RTC Alarm Control Register Settings

PARAMETER	ADDRESS	BITS
rtc_seconds_alarm[6:0]	67	6:0
rtc_minutes_alarm[6:0]	68	6:0
rtc_hours_alarm[5:0]	69	5:0
rtc_ampm_alarm	69	7
rtc_day_alarm[5:0]	70	5:0
rtc_month_arlarm[4:0]	71	4:0
rtc_year_alarm[7:0]	72	7:0

GPIO

12 general-purpose I/O pins are provided, labeled GPIO0 through GPIO11. The direction of the 12 GPIO pins can be independently set through control variable *gpio_oe(11:0)*. A pin is an input if the corresponding bit of *gpio_oe* is 0; a pin is an output if the corresponding bit of *gpio_oe* is 1.

The control variable *gpio(11:0)* serves different functions, depending on whether it is read from or written to. A read operation from *gpio* returns the logic state of the eight GPIO pins regardless of their direction. A write to *gpio* sets the output state of the GPIO pins if they are configured as outputs; there is no effect if the pin is configured as an input. Note that the write value of *gpio* is stored in a register, so that if a GPIO pin is changed from an input to an output its logic state is set by the stored value of *gpio*. Table 21 shows the *gpio* control variable settings.

The GPIO inputs can be optionally debounced if an RTC oscillator is running. Debouncing is controlled by *gpio_delay*, which is divided into 12 2-bit fields, each controlling a particular GPIO input according to Table 19.

Table 19. gpio_delay

[23:22]	[21:20]	[19:18]	[17:16]	[15:14]	[13:12]	[11:10]	[9:8]	[7:6]	[5:4]	[3:2]	[1:0]
GPIO11	GPIO10	GPIO9	GPIO8	GPIO7	GPIO6	GPIO5	GPIO4	GPIO3	GPIO2	GPIO1	GPIO0

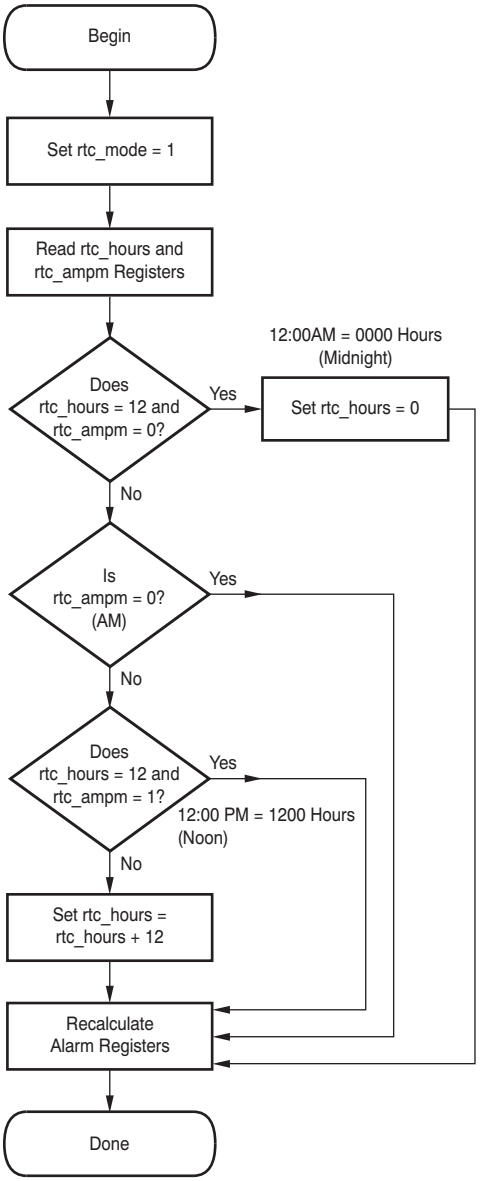


Figure 20. Procedure for Updating RTC Hour When Going from 12-Hour Mode to 24-Hour Mode

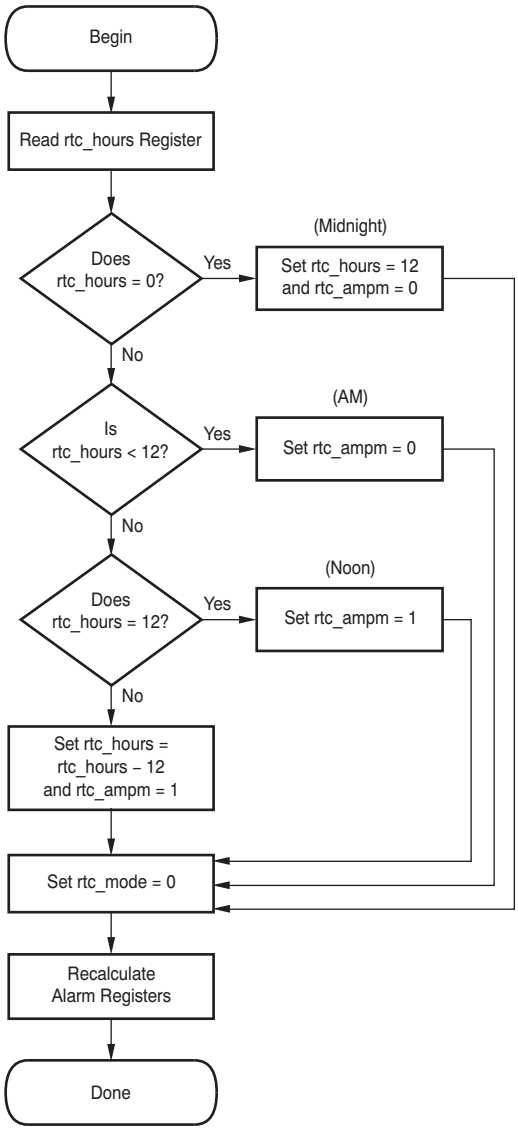


Figure 21. Procedure for Updating RTC Hour When Going from 24-Hour Mode to 12-Hour Mode

The debounce circuitry uses a clock divided from the RTC oscillator, with a debounce clock frequency given by Equation 25.

$$f_{DEBOUNCE} = \frac{f_{RTC}}{2^{2 \times GPIO\_DEBOUNCE\_FREQ + 1}} \quad (25)$$

If debounce is enabled, then in order for a GPIO input to change value (and possibly generate an interrupt if so programmed) it must remain stable for the number of debounce clock cycles (zero to three) given in the appropriate field of *gpio_delay.*

Table 20. General RTC Control Register Settings

PARAMETER	ADDRESS	BITS
rtc_ampm	75	7
rtc_comp_cnt[5:0]	64	12:7
rtc_comp_en	64	2
rtc_comp_val	66	15:0
rtc_day[5:0]	76	5:0
rtc_day_of_week[2:0]	79	2:0
rtc_en	64	0
rtc_hours[5:0]	75	5:0
rtc_max_count[15:0]	65	15:0
rtc_minutes[6:0]	74	6:0
rtc_mode	64	1
rtc_month[4:0]	77	4:0
rtc_seconds[6:0]	73	6:0
rtc_year[7:0]	78	7:0

Table 21. GPIO Control Register Settings

PARAMETER	ADDRESS	BITS
gpio	43	11:0
gpio_delay[15:0]	44	15:0
gpio_oe	42	11:0

Alternate Registers (GPIO and Input Attenuator)

If some of the GPIO pins on the AFE8221 are to be used to control the gain of a tuner, it may be desirable to change the GPIO values at the same time as the input attenuation to the DDC. To make this process more deterministic, the control parameters *gpio*, *ddc0_atten*, and *ddc1_atten* can be accessed through the alternate control register addresses of 96 and 97. By writing to register 96, *gpio* and *ddc0_atten* can be changed in a single register write; by writing to register 97, *gpio* and *ddc1_atten* can be changed in a single register write. Table 22 shows the operation control settings for these parameters.

Table 22. Alternate GPIO and DDC Control Register Settings

PARAMETER	ADDRESS	BITS
gpio	96	11:0
	97	
ddc0_atten	96	14:12
ddc1_atten	97	

Interrupt Generators

There are three programmable interrupt pins; IRQ0, IRQ1, and IRQ2. Only the operation of IRQ0 is described here; IRQ1 and IRQ2 are programmed in the same way, using different control variables.

Interrupts can be generated from various sources. Interrupt generation is enabled through *irq0_en*, as Table 23 shows.

Table 23. Interrupt Generation

BIT POSITION	SOURCE	BIT POSITION	SOURCE
0	GPIO	8	RTC alarm
1	None	9	RTC seconds rollover
2	I^2C Master done	10	RTC minutes rollover
3	Aux ADC done	11	RTC hours rollover
4	IFADC0 over-range	12	RTC months rollover
5	IFADC1 over-range	13	RTC day rollover
6	IFADC0 limit	14	RTC year rollover
7	IFADC1 limit	15	—

Setting a bit of *irq0_en* allows the generation of an interrupt for the corresponding event. All three IRQ generators run on the master clock (MCLK). When an interrupt event occurs on a given source signal, a value of 1 is written to the corresponding bit of *irq0_status*. This value is held in *irq0_status* until it is explicitly cleared by writing a 0 to the appropriate bit of *irq0_status*. A typical sequence upon receipt of an interrupt would be to poll *irq0_status* to determine the source of the interrupt, take whatever system action is appropriate, and then clear *irq0_status*.

Changes to any of the GPIO pins can also be programmed as interrupts. GPIO pin events are defined as changes from low to high or from high to low, depending on whether the corresponding bit in *irq0_gpio_edge* is high or low. GPIO interrupts are enabled by setting the corresponding bit in *irq0_gpio_en*; they are identified and cleared by reading and writing the corresponding bit in *irq0_gpio_status*.

The behavior of the IRQ0 pin is determined by *irq0_sense*. When *irq0_sense* is 0, IRQ0 is normally low and goes high on an unmasked interrupt event. When *irq0_sense* is 1, IRQ0 is normally high and goes low on an unmasked interrupt event. Table 24 shows the *irq0*, *irq1*, and *irq2* operations control settings.

Table 24. IRQ Control Register Settings

PARAMETER	ADDRESS	BITS
irq0_en	50	15:0
irq1_en	55	15:0
irq2_en	60	15:0
irq0_gpio_edge	48	11:0
irq1_gpio_edge	53	11:0
irq2_gpio_edge	58	11:0
irq0_gpio_en	49	11:0
irq1_gpio_en	54	11:0
irq2_gpio_en	59	11:0
irq0_gpio_status	52	11:0
irq1_gpio_status	57	11:0
irq2_gpio_status	62	11:0
irq0_sense	2	1
irq1_sense	2	2
irq2_sense	2	3
irq0_status	51	15:0
irq1_status	56	15:0
irq2_status	61	15:0

Wakeup Interrupt Generator

The WAKEUP interrupt generator functions in the same way as the IRQ generators with the following exceptions:

1. The WAKEUP generator runs on the RTC clock instead of MCLK;
2. The WAKEUP generator operates when the AFE is in low-power mode, whereas the IRQ generators do not; and
3. The interrupt sources for the WAKEUP interrupt generator are slightly different.

Table 25 shows the wakeup control settings. Table 26 shows the generator functions.

Table 25. Wakeup Control Register Settings

PARAMETER	ADDRESS	BITS
wakeup_sense	2	0
wakeup_gpio_edge	80	11:0
wakeup_gpio_en	81	11:0
wakeup_en	82	15:0
wakeup_status	83	15:0
wakeup_gpio_status	84	11:0

Table 26. WAKEUP Interrupt Generator

BIT POSITION	SOURCE
0	GPIO
1	None
2	None
3	None
4	None
5	None
6	None
7	None
8	RTC alarm
9	RTC seconds rollover
10	RTC minutes rollover
11	RTC hours rollover
12	RTC months rollover
13	RTC day rollover
14	RTC year rollover

Control Register Assignments

Table 27. Control Registers

	Address:	**1**	
	Description:	**Functional Block Enables**	
Bits	**Range**	**Action**	**Parameter Name**
1:0	0..3	Enable IFADC converters	ifadc_en(1:0)
2	0/1	Gain control for IF_ADC0	ifadc_gain0
3	0/1	Gain control for IF_ADC1	ifadc_gain1
5:4	0..3	Enable DDCs	ddc_en(1:0)
6	0/1	Synchronize DDC0 and DDC1	ddc_sync
7	0/1	Enable primary IF data interface	dout_en
8	0/1	Enable secondary IF data interface	bb_en
10:9	0..3	Enable auxiliary DACs	cdac_en(1:0)
11	0/1	Enable auxiliary ADC	aux_adc_en
12	0/1	Enable RTC output pins	rtc_oe
13	0/1	Enable reference clock output pins	refclk_en
	Address:	**2**	
	Description:	**Interrupt Output Level Configuration**	
Bits	**Range**	**Action**	**Parameter Name**
0	0/1	0 = Active high WAKEUP interrupt	wakeup_sense
		1 = Active low WAKEUP interrupt	
1	0/1	0 = Active high IRQ0 interrupt	irq0_sense
		1 = Active low IRQ0 interrupt	
2	0/1	0 = Active high IRQ1 interrupt	irq1_sense
		1 = Active low IRQ1 interrupt	
3	0/1	0 = Active high IRQ2 interrupt	irq2_sense
		1 = Active low IRQ2 interrupt	
	Address:	**3**	
	Description:	**DDC0 Input Attenuator**	
Bits	**Range**	**Action**	**Parameter Name**
2:0	0..6	Attenuation setting for DDC0	ddc_atten(2:0)
	Address:	**4**	
	Description:	**DDC0 Input Attenuator**	
Bits	**Range**	**Action**	**Parameter Name**
15:0	0..65535	Delay setting for DDC0 attenuator	ddc0_delay(15:0)
	Address:	**5**	
	Description:	**DDC0 NCO Frequency**	
Bits	**Range**	**Action**	**Parameter Name**
15:0	0..65535	Upper bytes of DDC0 NCO frequency	ddc0_demod_freq(31:16)
	Address:	**6**	
	Description:	**DDC0 NCO Frequency**	
Bits	**Range**	**Action**	**Parameter Name**
15:0	0..65535	Lower bytes of DDC0 NCO frequency	ddc0_demod_freq(15:0)
	Address:	**7**	
	Description:	**DDC0 NCO Phase**	
Bits	**Range**	**Action**	**Parameter Name**
15:0	0..65535	Upper bytes of DDC0 NCO phase	ddc0_demod_phase(31:16)

Table 27. Control Registers (continued)

	Address:	**8**	
	Description:	**DDC0 NCO Phase (continued)**	
Bits	**Range**	**Action**	**Parameter name**
15:0	0..65535	Lower bytes of DDC0 NCO phase	ddc0_demod_phase(15:0)
	Address:	**9**	
	Description:	**DDC0 CIC Filter**	
Bits	**Range**	**Action**	**Parameter Name**
8:0	4..256	CIC filter decimation rate	ddc0_cic_dec_rate(8:0)
	Address:	**10**	
	Description:	**DDC0 CIC Filter**	
Bits	**Range**	**Action**	**Parameter Name**
5:0	0..63	CIC filter post-filter shift	ddc0_cic_shift(5:0)
11:6	0..32	CIC filter post-filter scale	ddc0_cic_scale(5:0)
	Address:	**11**	
	Description:	**DDC0 FIR Filter 1**	
Bits	**Range**	**Action**	**Parameter Name**
1:0	0..3	FIR filter mode	ddc0_fir1_mode(1:0)
7:2	0..63	Number of coefficients to process	ddc0_fir1_ncoeffs(5:0)
13:8	0..63	Coefficient base address	ddc0_fir1_base_addr(5:0)
	Address:	**12**	
	Description:	**DDC0 FIR Filter 2A**	
Bits	**Range**	**Action**	**Parameter Name**
1:0	0..3	FIR filter mode	ddc0_fir2a_mode(1:0)
8:2	0..127	Number of coefficients to process	ddc0_fir2a_ncoeffs(6:0)
15:9	0..127	Coefficient base address	ddc0_fir2a_base_addr(6:0)
	Address:	**13**	
	Description:	**DDC0 FIR Filter 2B**	
Bits	**Range**	**Action**	**Parameter Name**
1:0	0..3	FIR filter mode	ddc0_fir2b_mode(1:0)
8:2	0..127	Number of coefficients to process	ddc0_fir2b_ncoeffs(6:0)
15:9	0..127	Coefficient base address	ddc0_fir2b_base_addr(6:0)
	Address:	**14**	
	Description:	**DDC0 FIR Filter Extended Features**	
Bits	**Range**	**Action**	**Parameter Name**
3:0	0..15	Post-filter shift for FIR filter 2A	ddc0_fir2a_shift(3:0)
7:4	0..15	Post-filter shift for FIR filter 2B	ddc0_fir2b_shift(3:0)
8	0/1	Enable interleave mode for FIR filter 2A and FIR filter 2B	ddc0_interleave
9	0/1	Disable decimation for FIR filter 1	ddc0_fir1_nodec
10	0/1	Disable decimation for FIR filter 2A and FIR filter 2B	ddc0_fir2_nodec
	Address:	**15**	
	Description:	**DDC1 Input Attenuator**	
Bits	**Range**	**Action**	**Parameter Name**
2:0	0..6	Attenuation setting for DDC1	ddc1_atten(2:0)
	Address:	**16**	
	Description:	**DDC1 Input Attenuator**	
Bits	**Range**	**Action**	**Parameter Name**
15:0	0..65535	Delay setting for DDC1 attenuator	ddc1_delay(15:0)

Table 27. Control Registers (continued)

	Address:	**17**	
	Description:	**DDC1 NCO Frequency**	
Bits	**Range**	**Action**	**Parameter Name**
15:0	0..65535	Upper bytes of DDC1 NCO frequency	ddc1_demod_freq(31:16)
	Address:	**18**	
	Description:	**DDC1 NCO Frequency**	
Bits	**Range**	**Action**	**Parameter Name**
15:0	0..65535	Lower bytes of DDC1 NCO frequency	ddc1_demod_freq(15:0)
	Address:	**19**	
	Description:	**DDC1 NCO Phase**	
Bits	**Range**	**Action**	**Parameter Name**
15:0	0..65535	Upper bytes of DDC1 NCO phase	ddc1_demod_phase(31:16)
	Address:	**20**	
	Description:	**DDC1 NCO Phase**	
Bits	**Range**	**Action**	**Parameter Name**
15:0	0..65536	Lower bytes of DDC1 NCO phase	ddc1_demod_phase(15:0)
	Address:	**21**	
	Description:	**DDC1 CIC Filter Decimation**	
Bits	**Range**	**Action**	**Parameter Name**
8:0	4..256	CIC filter decimation rate	ddc1_cic_dec_rate(8:0)
	Address:	**22**	
	Description:	**DDC1 CIC Filter**	
Bits	**Range**	**Action**	**Parameter Name**
5:0	0..63	CIC filter post-filter shift	ddc1_cic_shift(5:0)
11:6	0..32	CIC filter post-filter scale	ddc1_cic_scale(5:0)
	Address:	**23**	
	Description:	**DDC1 FIR Filter 1**	
Bits	**Range**	**Action**	**Parameter Name**
1:0	0..3	FIR filter mode	ddc1_fir1_mode(1:0)
7:2	0..63	Number of coefficients to process	ddc1_fir1_ncoeffs(5:0)
13:8	0..63	Coefficient base address	ddc1_fir1_base_addr(5:0)
	Address:	**24**	
	Description:	**DDC1 FIR Filter 2A**	
Bits	**Range**	**Action**	**Parameter Name**
1:0	0..3	FIR filter mode	ddc1_fir2a_mode(1:0)
8:2	0..127	Number of coefficients to process	ddc1_fir2a_ncoeffs(6:0)
15:9	0..127	Coefficient base address	ddc1_fir2a_base_addr(6:0)
	Address:	**25**	
	Description:	**DDC1 FIR Filter 2B**	
Bits	**Range**	**Action**	**Parameter Name**
1:0	0..3	FIR filter mode	ddc1_fir2b_mode(1:0)
8:2	0..127	Number of coefficients to process	ddc1_fir2b_ncoeffs(6:0)
15:9	0..127	Coefficient base address	ddc1_fir2b_base_addr(6:0)

Table 27. Control Registers (continued)

	Address:	**26**	
	Description:	**DDC1 FIR Filter Extended Features**	
Bits	**Range**	**Action**	**Parameter Name**
3:0	0..15	Post-filter shift for FIR filter 2A	ddc1_fir2a_shift(3:0)
7:4	0..15	Post-filter shift for FIR filter 2B	ddc1_fir2b_shift(3:0)
8	0/1	Enable interleave mode for FIR filter 2A and FIR filter 2B	ddc1_interleave
9	0/1	Disable decimation for FIR filter 1	ddc1_fir1_nodec
10	0/1	Disable decimation for FIR filter 2A and FIR filter 2B	ddc1_fir2_nodec
	Address:	**27**	
	Description:	**Data Interface Configuration**	
Bits	**Range**	**Action**	**Parameter Name**
15:0	0..65535	Configuration for IF_DOUT0	if_dout0_config
	Address:	**28**	
	Description:	**Data Interface Configuration**	
Bits	**Range**	**Action**	**Parameter Name**
15:0	0..65535	Configuration for IF_DOUT1	if_dout1_config
	Address:	**29**	
	Description	**Data Interface Configuration**	
Bits	**Range**	**Action**	**Parameter Name**
15:0	0..65535	Configuration for IF_DOUT2	if_dout2_config
	Address:	**30**	
	Description:	**Data Interface Configuration**	
Bits	**Range**	**Action**	**Parameter Name**
15:0	0..65535	Configuration for IF_DOUT3	if_dout3_config
	Address:	**31**	
	Description:	**Data Interface Configuration**	
Bits	**Range**	**Action**	**Parameter Name**
4:0	0..16	Divide factor to derive IF_DCLK from MCLK	if_dclk_div(4:0)
5	0/1	0: IF_DFSO and IF_DOUTx change on rising edge of IF_DCLK	if_dclk_edge
		1: IF_DFSO and IF_DOUTx change on falling edge of IF_DCLK	
6	0/1	0: IF_DFSO generated by DDC0	if_dfso_select
		1: IF_DFSO generated by DDC1	
8:7	0..2	0: IF_DFSO one IF_DCLK cycle wide	if_dfso_mode(1:0)
		1: IF_DFSO toggles once per frame	
		2: IF_DFSO 16 IF_DCLK cycles wide	
	Address:	**32**	
	Description:	**Data Interface Configuration**	
Bits	**Range**	**Action**	**Parameter Name**
15:0	0..65535	Configuration for BB_DOUT0	bb_dout0_config
	Address:	**33**	
	Description:	**Data Interface Configuration**	
Bits	**Range**	**Action**	**Parameter Name**
15:0	0..65535	Configuration for BB_DOUT1	bb_dout1_config
	Address:	**34**	
	Description:	**Data Interface Configuration**	
Bits	**Range**	**Action**	**Parameter Name**
15:0	0..65535	Configuration for BB_DOUT2	bb_dout2_config

Table 27. Control Registers (continued)

	Address:	**35**	
	Description:	**Data Interface Configuration (continued)**	
Bits	**Range**	**Action**	**Parameter Name**
15:0	0..65535	Configuration for BB_DOUT3	bb_dout3_config
	Address:	**36**	
	Description:	**Data Interface Configuration**	
Bits	**Range**	**Action**	**Parameter Name**
4:0	0..16	Divide factor to derive BB_BCK from MCLK	bb_bck_div(4:0)
5	0/1	0: BB_WS and BB_DOUTx change on rising edge of BB_BCK	bb_bck_edge
		1: BB_WS and BB_DOUTx change on falling edge of BB_BCK	
6	0/1	0: BB_WS generated by DDC0	bb_ws_select
		1: BB_WS generated by DDC1	
8:7	0..2	0: BB_WS one BB_BCK cycle wide	bb_ws_mode(1:0)
		1: BB_WS toggles once per frame	
		2: BB_WS 16 BB_BCK cycles wide	
	Address:	**37**	
	Description:	**CDAC0 Output**	
Bits	**Range**	**Action**	**Parameter Name**
11:0	0..4095	Output value for CDAC0	cdac0_out(11:0)
	Address:	**38**	
	Description:	**CDAC1 Output**	
Bits	**Range**	**Action**	**Parameter Name**
11:0	0..4095	Output value for CDAC1	cdac1_out(11:0)
	Address:	**39**	
	Description:	**Aux ADC**	
Bits	**Range**	**Action**	**Parameter Name**
7:0	0..255	Register read: Conversion result for auxiliary ADC (read only)	aux_adc_out(7:0)
11:8	0, 1, 2, 4, or 8	0: No aux ADC inputs connected	aux_adc_sel(3:0)
		1: AUX_ADC0 pin connected to aux ADC	
		2: AUX_ADC1 pin connected to aux ADC	
		4: AUX_ADC2 pin connected to aux ADC	
		8: AUX_ADC3 pin connected to aux ADC	
14:12		Not used	
15	0/1	Register write: Starts a conversion when 1 is written	aux_adc_done
		Register read: Returns 0 while conversion is in progress, 1 when conversion is finished	
	Address:	**40**	
	Description:	**Reference Clock Configuration**	
Bits	**Range**	**Action**	**Parameter Name**
15:0	0..4095	Low period (in units of MCLK cycles) for reference clock output	refclk_lo(15:0)
	Address:	**41**	
	Description:	**Reference Clock Configuration**	
Bits	**Range**	**Action**	**Parameter Name**
15:0	0..4095	High period (in units of MCLK cycles) for reference clock output	refclk_hi(15:0)

Table 27. Control Registers (continued)

Address: 42 Description: GPIO Configuration			
Bits	**Range**	**Action**	**Parameter Name**
0	0/1	0 = GPIO0 set as input 1 = GPIO0 set as output	gpio_oe(0)
1	0/1	0 = GPIO1 set as input 1 = GPIO1 set as output	gpio_oe(1)
2	0/1	0 = GPIO2 set as input 1 = GPIO2 set as output	gpio_oe(2)
3	0/1	0 = GPIO3 set as input 1 = GPIO3 set as output	gpio_oe(3)
4	0/1	0 = GPIO4 set as input 1 = GPIO4 set as output	gpio_oe(4)
5	0/1	0 = GPIO5 set as input 1 = GPIO5 set as output	gpio_oe(5)
6	0/1	0 = GPIO6 set as input 1 = GPIO6 set as output	gpio_oe(6)
7	0/1	0 = GPIO7 set as input 1 = GPIO7 set as output	gpio_oe(7)
8	0/1	0 = GPIO8 set as input 1 = GPIO8 set as output	gpio_oe(8)
9	0/1	0 = GPIO9 set as input 1 = GPIO9 set as output	gpio_oe(9)
10	0/1	0 = GPIO10 set as input 1 = GPIO10 set as output	gpio_oe(10)
11	0/1	0 = GPIO11 set as input 1 = GPIO11 set as output	gpio_oe(11)
Address: 43 **Description: GPIO Configuration (continued)**			
Bits	**Range**	**Action**	**Parameter Name**
0	0/1	Register write: drives value on GPIO0 pin if enabled as output Register read: returns value on GPIO0 pin	gpio(0)
1	0/1	Register write: drives value on GPIO1 pin if enabled as output Register read: returns value on GPIO1 pin	gpio(1)
2	0/1	Register write: drives value on GPIO2 pin if enabled as output Register read: returns value on GPIO2 pin	gpio(2)
3	0/1	Register write: drives value on GPIO3 pin if enabled as output Register read: returns value on GPIO3 pin	gpio(3)
4	0/1	Register write: drives value on GPIO4 pin if enabled as output Register read: returns value on GPIO4 pin	gpio(4)
5	0/1	Register write: drives value on GPIO5 pin if enabled as output Register read: returns value on GPIO5 pin	gpio(5)
6	0/1	Register write: drives value on GPIO6 pin if enabled as output Register read: returns value on GPIO6 pin	gpio(6)
7	0/1	Register write: drives value on GPIO7 pin if enabled as output Register read: returns value on GPIO7 pin	gpio(7)
8	0/1	Register write: drives value on GPIO8 pin if enabled as output Register read: returns value on GPIO8 pin	gpio(8)

Table 27. Control Registers (continued)

9	0/1	Register write: drives value on GPIO9 pin if enabled as output Register read: returns value on GPIO9 pin	gpio(9)
10	0/1	Register write: drives value on GPIO10 pin if enabled as output Register read: returns value on GPIO10 pin	gpio(10)
11	0/1	Register write: drives value on GPIO11 pin if enabled as output Register read: returns value on GPIO11 pin	gpio(11)
	Address:	**44**	
	Description:	**GPIO Configuration**	
Bits	**Range**	**Action**	**Parameter Name**
1:0	0..3	GPIO0 debounce setting	gpio_delay(1:0)
3:2	0..3	GPIO1 debounce setting	gpio_delay(3:2)
5:4	0..3	GPIO2 debounce setting	gpio_delay(5:4)
7:6	0..3	GPIO3 debounce setting	gpio_delay(7:6)
9:8	0..3	GPIO4 debounce setting	gpio_delay(9:8)
11:10	0..3	GPIO5 debounce setting	gpio_delay(11:10)
13:12	0..3	GPIO6 debounce setting	gpio_delay(13:12)
15:14	0..3	GPIO7 debounce setting	gpio_delay(15:14)
	Address:	**45**	
	Description:	**GPIO Configuration**	
Bits	**Range**	**Action**	**Parameter Name**
1:0	0..3	GPIO8 debounce setting	gpio_delay(17:16)
3:2	0..3	GPIO9 debounce setting	gpio_delay(19:18)
5:4	0..3	GPIO10 debounce setting	gpio_delay(21:20)
7:6	0..3	GPIO11 debounce setting	gpio_delay(23:22)
	Address:	**46**	
	Description:	**IF ADC Alarm**	
Bits	**Range**	**Action**	**Parameter Name**
11:0	0..2047	Alarm limit for IF_ADC0	ifadc0_limit(11:0)
	Address:	**47**	
	Description:	**IF ADC Alarm**	
Bits	**Range**	**Action**	**Parameter Name**
11:0	0..2047	Alarm limit for IF_ADC1	ifadc1_limit(11:0)
	Address:	**48**	
	Description:	**IRQ0 Configuration**	
Bits	**Range**	**Action**	**Parameter Name**
11:0	0..4095	GPIO input edge select for IRQ0	irq0_gpio_edge(11:0)
	Address:	**49**	
	Description:	**IRQ0 Configuration**	
Bits	**Range**	**Action**	**Parameter Name**
11:0	0..4095	IRQ0 GPIO enable	irq0_gpio_en(11:0)
	Address:	**50**	
	Description:	**IRQ0 Configuration**	
Bits	**Range**	**Action**	**Parameter Name**
15:0	0..65535	IRQ0 enable	irq0_en(15:0)

Table 27. Control Registers (continued)

	Address:	**51**	
	Description:	**IRQ0 Status**	
Bits	**Range**	**Action**	**Parameter Name**
15:0	0..65535	Register read: returns IRQ0 status Register write: clears interrupt bit if 1 is written	irq0_status(15:0)
	Address:	**52**	
	Description:	**IRQ0 GPIO Status**	
Bits	**Range**	**Action**	**Parameter Name**
11:0	0..4095	Register read: returns IRQ0 status Register write: clears interrupt bit if 1 is written	irq0_gpio_status(11:0)
	Address:	**53**	
	Description:	**IRQ1 Configuration**	
Bits	**Range**	**Action**	**Parameter Name**
11:0	0..4095	GPIO input edge select for IRQ1	irq1_gpio_edge(11:0)
	Address:	**54**	
	Description:	**IRQ1 Configuration**	
Bits	**Range**	**Action**	**Parameter Name**
11:0	0..4095	IRQ1 GPIO enable	irq1_gpio_en(11:0)
	Address:	**55**	
	Description:	**IRQ1 Configuration**	
Bits	**Range**	**Action**	**Parameter Name**
15:0	0..65535	IRQ1 enable	irq1_en(15:0)
	Address:	**56**	
	Description:	**IRQ1 Status**	
Bits	**Range**	**Action**	**Parameter Name**
15:0	0..65535	Register read: returns IRQ1 status Register write: clears interrupt bit if 1 is written	irq1_status(15:0)
	Address:	**57**	
	Description:	**IRQ1 GPIO Status (continued)**	
Bits	**Range**	**Action**	**Parameter Name**
11:0	0..4095	Register read: returns IRQ1 status Register write: clears interrupt bit if 1 is written	irq1_gpio_status(11:0)
	Address:	**58**	
	Description:	**IRQ2 Configuration**	
Bits	**Range**	**Action**	**Parameter Name**
11:0	0..4095	GPIO input edge select for IRQ2	irq2_gpio_edge(11:0)
	Address:	**59**	
	Description:	**IRQ2 Configuration**	
Bits	**Range**	**Action**	**Parameter Name**
11:0	0..4095	IRQ2 GPIO enable	irq2_gpio_en(11:0)
	Address:	**60**	
	Description:	**IRQ2 Configuration**	
Bits	**Range**	**Action**	**Parameter Name**
15:0	0..65535	IRQ2 enable	irq2_en(15:0)

Table 27. Control Registers (continued)

	Address:	**61**	
	Description:	**IRQ2 Status**	
Bits	**Range**	**Action**	**Parameter Name**
15:0	0..65535	Register read: returns IRQ2 status	irq2_status(15:0)
		Register write: clears interrupt bit if 1 is written	
	Address:	**62**	
	Description:	**IRQ2 GPIO Status**	
Bits	**Range**	**Action**	**Parameter Name**
11:0	0..4095	Register read: returns IRQ2 status	irq2_gpio_status(11:0)
		Register write: clears interrupt bit if 1 is written	
	Address:	**63**	
	Description:	**Not Used**	
Bits	**Range**	**Action**	**Parameter Name**
—	—	—	—
	Address:	**64**	
	Description:	**Real-Time Clock Configuration**	
Bits	**Range**	**Action**	**Parameter Name**
0	0/1	0 = Freeze real-time clock	rtc_en
		1 = Enable real-time clock operation	
1	0/1	0:12 hour mode	rtc_mode
		1:24 hour mode	
2	0/1	Enable clock compensation	rtc_comp_en
3	0/1	Enable clock test mode	rtc_test_en
6:4	0..4	Clock test mode selection	rtc_test_mode(2:0)
12:7	0..31	Compensation count	rtc_comp_cnt(5:0)
15:13	0..7	Frequency select for GPIO debounce	gpio_debounce_freq(2:0)
	Address:	**65**	
	Description:	**Real-Time Clock Configuration**	
Bits	**Range**	**Action**	**Parameter Name**
15:0	0..32767	Real-time one second terminal count. Default = 16384 (for 32.768-kHz crystal)	rtc_max_count(15:0)
	Address:	**66**	
	Description:	**Real-Time Clock Configuration (continued)**	
Bits	**Range**	**Action**	**Parameter Name**
15:0	–32768..32767	Real-time clock compensation value. Default = 16384	rtc_comp_val(15:0)
	Address:	**67**	
	Description:	**Real-Time Clock Alarm**	
Bits	**Range**	**Action**	**Parameter Name**
6:0	0..59	Seconds alarm setting	rtc_seconds_alarm(6:0)
	Address:	**68**	
	Description:	**Real-Time Clock Alarm**	
Bits	**Range**	**Action**	**Parameter Name**
6:0	0..59	Minutes alarm setting	rtc_minutes_alarm(6:0)

Table 27. Control Registers (continued)

	Address:	**69**	
	Description:	**Realtime Clock Alarm**	
Bits	**Range**	**Action**	**Parameter Name**
5:0	1..12	Hour alarm setting, 12-hour mode	rtc_hours_alarm(5:0)
	0..23	Hour alarm setting, 24-hour mode	
6	—	Not used	—
7	0/1	12-hour mode: 0 = AM, 1 = PM	rtc_ampm_alarm
		24-hour mode: not used	
	Address:	**70**	
	Description:	**Real-Time Clock Alarm**	
Bits	**Range**	**Action**	**Parameter Name**
5:0	1..31	Day of the month alarm setting	rtc_day_alarm(5:0)
	Address:	**71**	
	Description:	**Real-Time Clock Alarm**	
Bits	**Range**	**Action**	**Parameter Name**
4:0	1..12	Month alarm setting	rtc_months_alarm(4:0)
	Address:	**72**	
	Description:	**Real-Time Clock Alarm**	
Bits	**Range**	**Action**	**Parameter Name**
7:0	0..99	Year alarm setting	rtc_year_alarm(7:0)
	Address:	**73**	
	Description:	**Real-Time Clock Current Time**	
Bits	**Range**	**Action**	**Parameter Name**
6:0	0..59	Seconds register	rtc_seconds(6:0)
14:7	—	Not used	—
15	0/1	Real-time clock busy (read only)	rtc_busy
	Address:	**74**	
	Description:	**Real-Time Clock Current Time**	
Bits	**Range**	**Action**	**Parameter Name**
6:0	0..59	Minutes register	rtc_minutes(6:0)
14:7	—	Not used	—
15	0/1	Real-time clock busy (read only)	rtc_busy
	Address:	**75**	
	Description:	**Real-Time Clock Current Time (continued)**	
Bits	**Range**	**Action**	**Parameter Name**
5:0	1..12	Hour register, 12-hour mode	rtc_hours(5:0)
	0..23	Hour register, 24-hour mode	
6	—	Not used	—
7	0/1	12-hour mode: 0 = AM, 1 = PM	rtc_ampm
		24-hour mode: not used	
14:8	—	Not used	—
15	0/1	Real-time clock busy (read only)	rtc_busy
	Address:	**76**	
	Description:	**Real-Time Clock Current Time**	
Bits	**Range**	**Action**	**Parameter Name**
5:0	1..31	Day of month register	rtc_day(5:0)
14:6	—	Not used	—
15	0/1	Real-time clock busy (read only)	rtc_busy

Table 27. Control Registers (continued)

	Address:	**77**	
	Description:	**Real-Time Clock Current Time**	
Bits	**Range**	**Action**	**Parameter Name**
4:0	1..12	Month register	rtc_month(4:0)
14:5	—	Not used	—
15	0/1	Real-time clock busy (read only)	rtc_busy
	Address:	**78**	
	Description:	**Real-Time Clock Current Time**	
Bits	**Range**	**Action**	**Parameter Name**
7:0	0..99	Year register	rtc_year(7:0)
14:8	—	Not used	—
15	0/1	Real-time clock busy (read only)	rtc_busy
	Address:	**79**	
	Description:	**Real-Time Clock Current Time**	
Bits	**Range**	**Action**	**Parameter Name**
2:0	0..6	Day of week	rtc_day_of_week(2:0)
14:3	—	Not used	—
15	0/1	Real-time clock busy (read only)	rtc_busy
	Address:	**80**	
	Description:	**WAKEUP Configuration**	
Bits	**Range**	**Action**	**Parameter Name**
11:0	0..4095	GPIO input edge select for WAKEUP	wakeup_gpio_edge(11:0)
	Address:	**81**	
	Description:	**WAKEUP Configuration**	
Bits	**Range**	**Action**	**Parameter Name**
11:0	0..4095	WAKEUP GPIO enable	wakeup_gpio_en(11:0)
	Address:	**82**	
	Description:	**WAKEUP Configuration**	
Bits	**Range**	**Action**	**Parameter Name**
15:0	0..65535	WAKEUP enable	wakeup_en(15:0)
	Address:	**83**	
	Description:	**WAKEUP Status**	
Bits	**Range**	**Action**	**Parameter Name**
15:0	0..65535	Register read: returns WAKEUP status	wakeup_status(15:0)
		Register write: clears interrupt bit if 1 is written	
	Address:	**84**	
	Description:	**WAKEUP GPIO Status**	
Bits	**Range**	**Action**	**Parameter Name**
11:0	0..4095	Register read: returns WAKEUP status	wakeup_gpio_status(11:0)
		Register write: clears interrupt bit if 1 is written	
	Address:	**85–119**	
	Description:	**Not Used**	
Bits	**Range**	**Action**	**Parameter Name**
—	—	—	—

Table 27. Control Registers (continued)

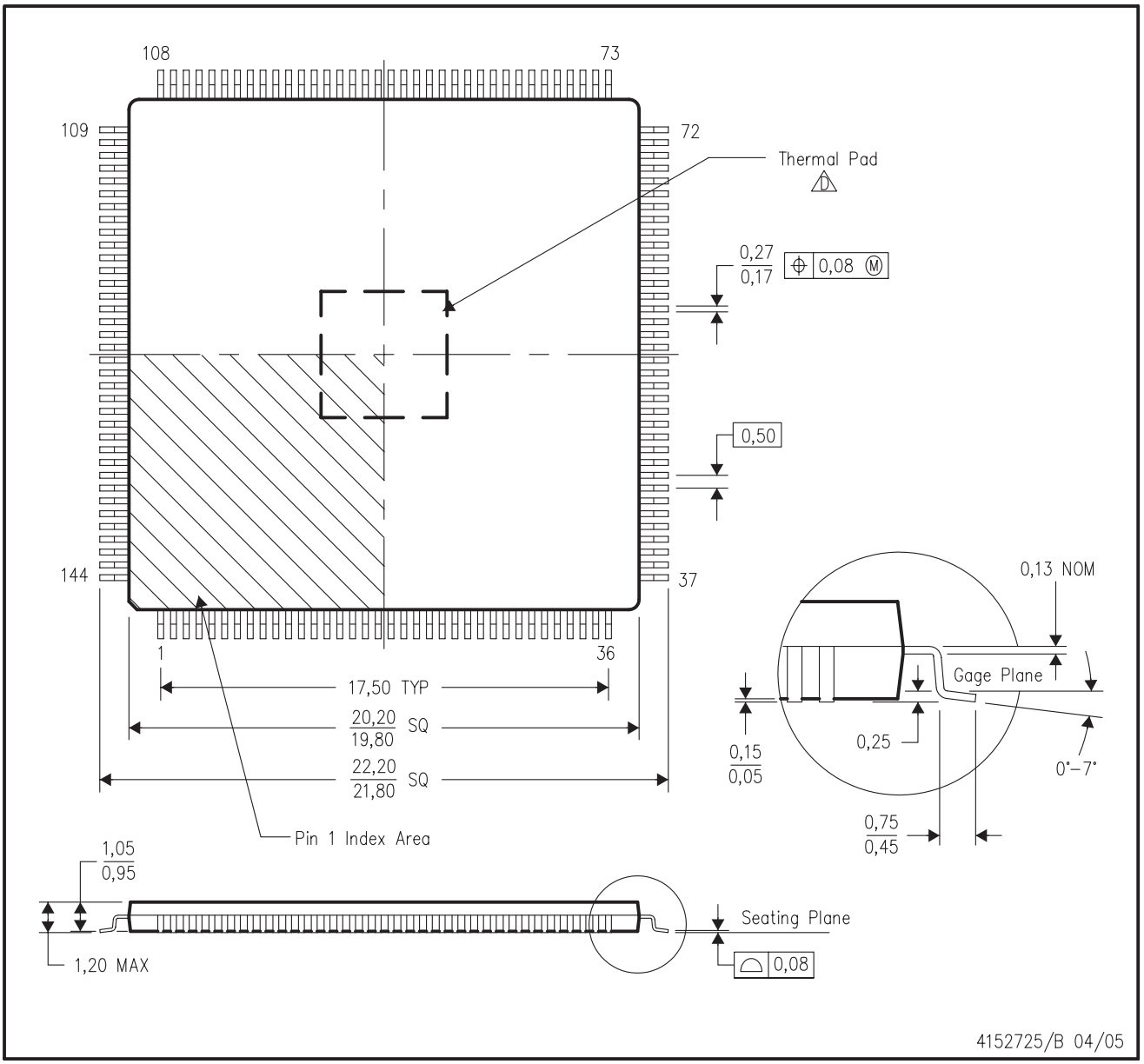
	Address:	**120**	
	Description:	**I²C Master—Slave Address**	
Bits	**Range**	**Action**	**Parameter Name**
14:0	0..32767	Slave address used for master transactions. Also starts transaction. 7-bit addressing: bits 6:0 are the slave address. Bits 14:7 are ignored. 10-bit addressing: bits 9:0 are the slave address. Bits 14:10 are the upper five bits for the slave address first byte	i2cm_slave_addr(14:0)
15	—	Not used	—
	Address:	**121**	
	Description:	**I²C Master—Slave Burst length**	
Bits	**Range**	**Action**	**Parameter Name**
4:0	1..16	Number of bytes to transfer for the first data burst	i2cm_start_data_length(4:0)
5	0/1	Selects the Read/Write bit value used with the slave address following START. 0 selects Write 1 selects Read	i2cm_start_rw
7:6	—	Not used	—
12:8	1..16	Number of bytes to transfer for the second data burst in a combined format transfer. This parameter is used only if i2cm_use_sr = 1.	i2cm_restart_data_length(4:0)
13	0/1	Selects the Read/Write bit value used with the slave address following RESTART. 0 selects Write 1 selects Read	i2cm_restart_rw
14	0/1	0 = SDA0/SCL0 interface 1 = SDA1/SCL1 interface	i2xm_if_sel
15	0/1	0 selects I²C transactions without a repeated start. 1 selects combined transactions with a repeated start.	i2cm_use_sr
	Address:	**122**	
	Description:	**I²C Master—Write Buffer Control**	
Bits	**Range**	**Action**	**Parameter Name**
7:0	0..255	Stores data in the write buffer at the location specified by i2cm_write_byte_ptr(3:0)	i2cm_write_byte(7:0)
11:8	0..15	Buffer location where the i2cm_write_byte should be placed	i2cm_write_byte_ptr(3:0)
14:12	—	Not used	—
15	0/1	Auto increment 0 = i2cm_write_byte_ptr is used for storing the i2cm_write_byte value in the write buffer memory 1 = i2cm_write_byte_ptr is ignored from the host and it is auto-incremented for writing the i2cm_write_byte to the buffer	i2cm_write_auto_inc
	Address:	**123**	
	Description:	**I²C Master—Read Buffer Control**	
Bits	**Range**	**Action**	**Parameter Name**
7:0	0..255	Read only - retrieves data read by the I²C master from the read buffer	i2cm_read_byte(7:0)
11:8	0..15	Buffer location where the i2cm_read_byte should be retrieved	i2cm_read_byte_ptr(3:0)
14:12	—	Not used	—
15	0/1	Auto increment 0 = i2cm_read_byte_ptr is used for retrieving the i2cm_write_byte value in the read buffer memory 1 = i2cm_read_byte_ptr is ignored from the host and it is auto-incremented for retrieving the i2cm_read_byte from the buffer	i2cm_read_auto_inc

Table 27. Control Registers (continued)

Address: 124 Description: I²C Master—Read Buffer Control			
Bits	**Range**	**Action**	**Parameter Name**
7:0	5 to 200	Controls the I²C SCL clock rate	i2cm_clk_cycles(7:0)
8	0/1	Multi-master	i2cm_multimaster
		0 = single I²C master on SDA and SCL signals	
		1 = multiple I²C masters present on SDA and SCL signals	
9	0/1	SCL sync enable	i2cm_scl_sync_en
		0 = prohibit SCL stretching by slave	
		1 = permit SCL stretching by slave	
10	0/1	Allow slave NACK	i2cm_allow_slave_nack
		0 = require slave to ACK transfers	
		1 = permit slave to not-acknowledge (NACK)	
11	0/1	Clear slave NACK. if i2cm_allow_slave_nack is zero and the slave fails to acknowledge, this bit when read will be set. No further I²C transactions are allowed until this bit is written as a 1 to clear the slave NACK condition.	i2cm_clear_slave_nack
12	0/1	0 = Use 7 bit addressing	i2cm_10b_addressing
		1 = Use 10 bit addressing	
13	0/1	End transfer with stop	i2cm_use_stop
		0 = do not issue a stop after last byte transferred and pause transaction	
		1 = issue a stop after the last byte transferred	
14	0/1	Holding. Read-only. Used when i2cm_use_stop is set to zero.	i2cm_holding
15	0/1	Done. Read-only. When set, the I²C master has completed any pending transactions.	i2cm_done

MECHANICAL DATA

RFP (S–PQFP–G144) PowerPAD™ PLASTIC QUAD FLATPACK



NOTES:
A. All linear dimensions are in millimeters.
B. This drawing is subject to change without notice.
C. Body dimensions do not include mold flash or protrusion
1 This package is designed to be soldered to a thermal pad on the board. Refer to Technical Brief, PowerPad Thermally Enhanced Package, Texas Instruments Literature No. SLMA002 for information regarding recommended board layout. This document is available at www.ti.com <http://www.ti.com>.
E. Falls within JEDEC MS–026

PowerPAD is a trademark of Texas Instruments.

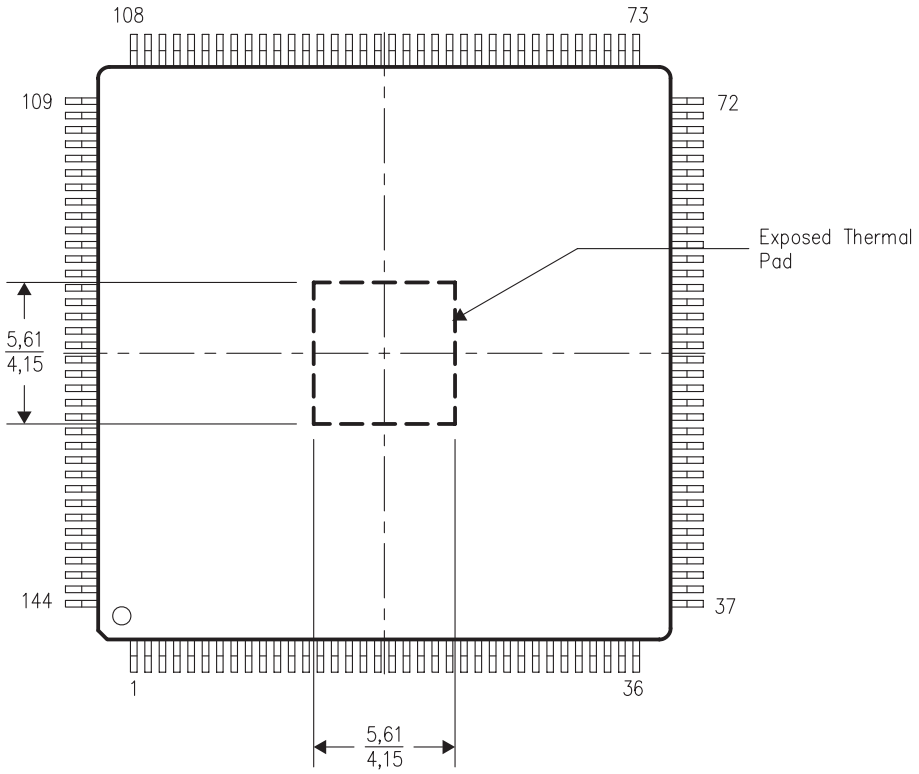
MECHANICAL DATA

THERMAL INFORMATION

This PowerPAD™ package incorporates an exposed thermal pad that is designed to be attached directly to an external heatsink. The thermal pad must be soldered directly to the printed circuit board (PCB). After soldering, the PCB can be used as a heatsink. In addition, through the use of thermal vias, the thermal pad can be attached directly to the appropriate copper plane shown in the electrical schematic for the device, or alternatively, can be attached to a special heatsink structure designed into the PCB. This design optimizes the heat transfer from the integrated circuit (IC).

For additional information on the PowerPAD package and how to take advantage of its heat dissipating abilities, refer to Technical Brief, PowerPAD Thermally Enhanced Package, Texas Instruments Literature No. SLMA002 and Application Brief, PowerPAD Made Easy, Texas Instruments Literature No. SLMA004. Both documents are available at www.ti.com.

The exposed thermal pad dimensions for this package are shown in the following illustration.



Top View

NOTE: All linear dimensions are in millimeters

Exposed Thermal Pad Dimensions

4206900/B 10/07

PACKAGING INFORMATION

Orderable Device	Status (1)	Package Type	Package Drawing	Pins	Package Qty	Eco Plan (2)	Lead finish/ Ball material (6)	MSL Peak Temp (3)	Op Temp (°C)	Device Marking (4/5)	Samples
AFE8221IRFPQ1	ACTIVE	HTQFP	RFP	144	60	RoHS & Green	NIPDAU	Level-3-260C-168 HR	-40 to 85	AFE8221Q	Samples

(1) The marketing status values are defined as follows:
ACTIVE: Product device recommended for new designs.
LIFEBUY: TI has announced that the device will be discontinued, and a lifetime-buy period is in effect.
NRND: Not recommended for new designs. Device is in production to support existing customers, but TI does not recommend using this part in a new design.
PREVIEW: Device has been announced but is not in production. Samples may or may not be available.
OBSOLETE: TI has discontinued the production of the device.

(2) **RoHS:** TI defines "RoHS" to mean semiconductor products that are compliant with the current EU RoHS requirements for all 10 RoHS substances, including the requirement that RoHS substance do not exceed 0.1% by weight in homogeneous materials. Where designed to be soldered at high temperatures, "RoHS" products are suitable for use in specified lead-free processes. TI may reference these types of products as "Pb-Free".
RoHS Exempt: TI defines "RoHS Exempt" to mean products that contain lead but are compliant with EU RoHS pursuant to a specific EU RoHS exemption.
Green: TI defines "Green" to mean the content of Chlorine (Cl) and Bromine (Br) based flame retardants meet JS709B low halogen requirements of <=1000ppm threshold. Antimony trioxide based flame retardants must also meet the <=1000ppm threshold requirement.

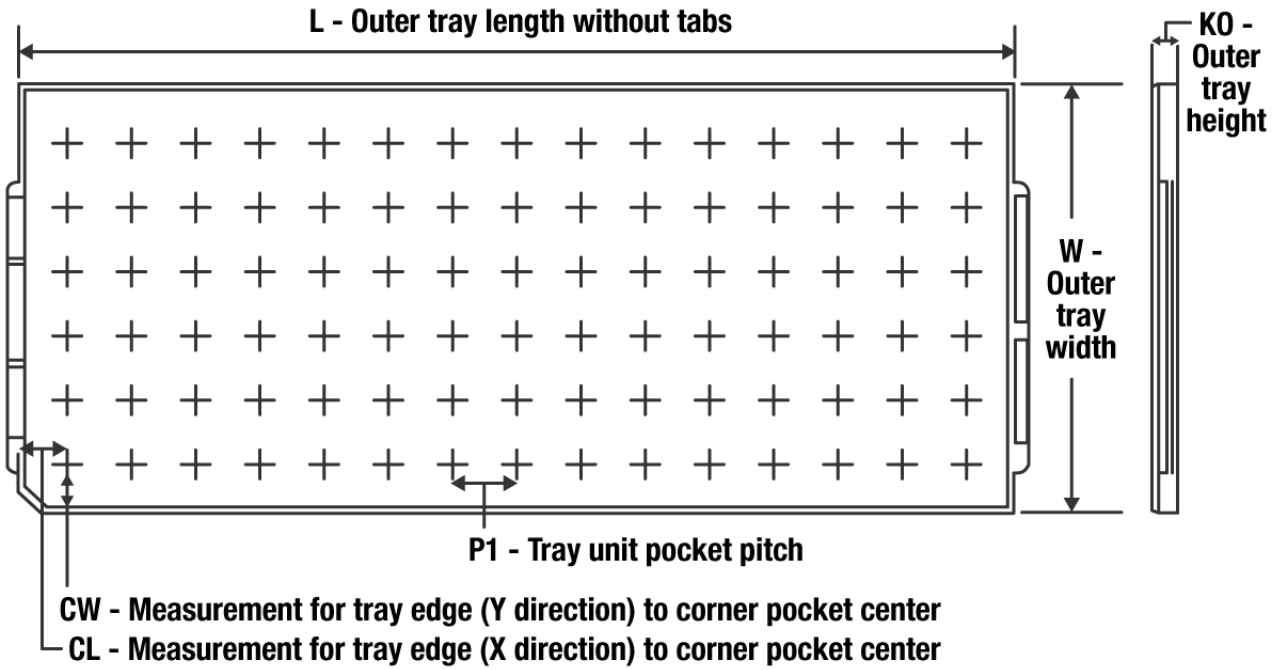
(3) MSL, Peak Temp. - The Moisture Sensitivity Level rating according to the JEDEC industry standard classifications, and peak solder temperature.

(4) There may be additional marking, which relates to the logo, the lot trace code information, or the environmental category on the device.

(5) Multiple Device Markings will be inside parentheses. Only one Device Marking contained in parentheses and separated by a "~" will appear on a device. If a line is indented then it is a continuation of the previous line and the two combined represent the entire Device Marking for that device.

(6) Lead finish/Ball material - Orderable Devices may have multiple material finish options. Finish options are separated by a vertical ruled line. Lead finish/Ball material values may wrap to two lines if the finish value exceeds the maximum column width.

TRAY

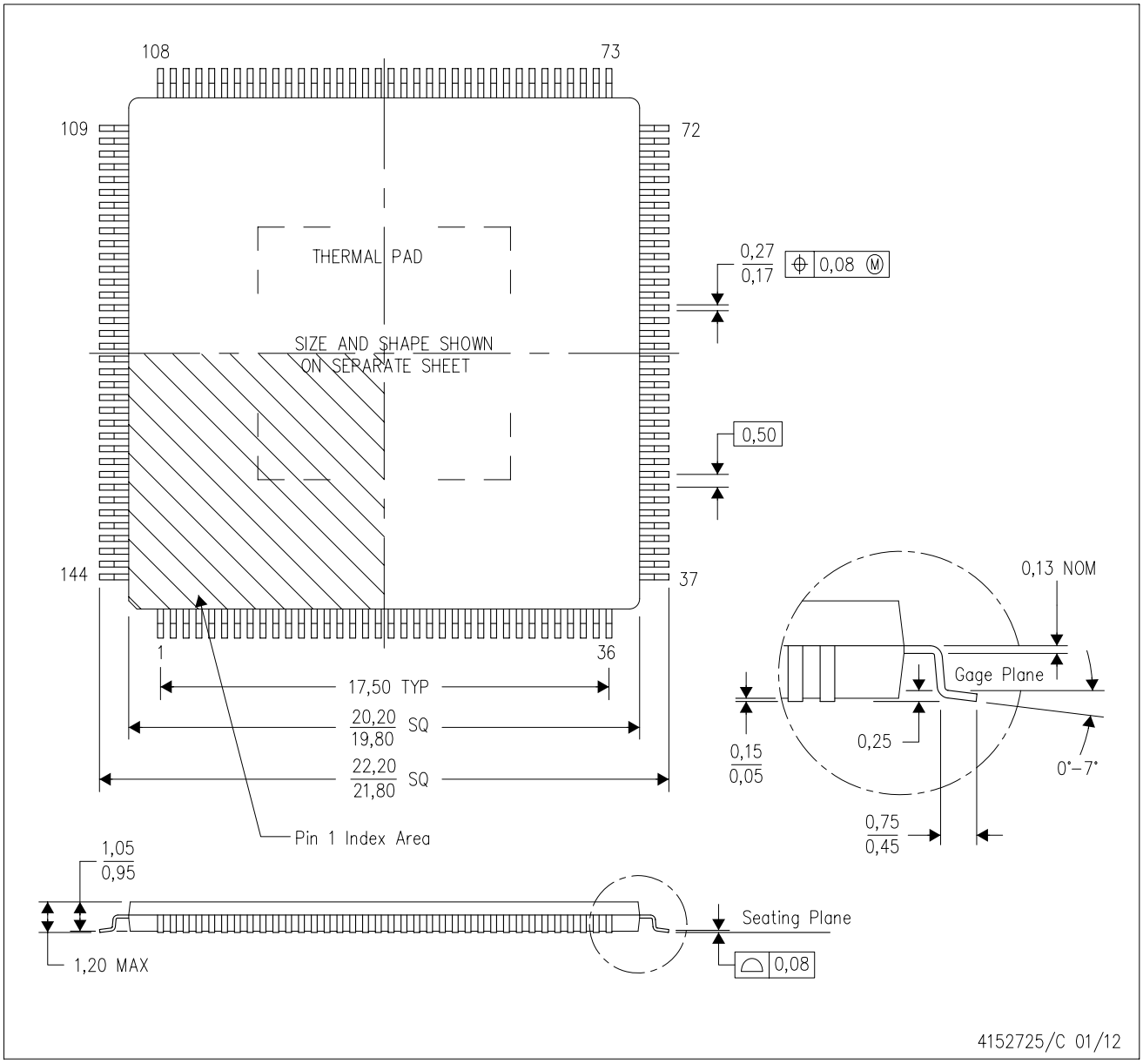


Chamfer on Tray corner indicates Pin 1 orientation of packed units.

*All dimensions are nominal

Device	Package Name	Package Type	Pins	SPQ	Unit array matrix	Max temperature (°C)	L (mm)	W (mm)	K0 (µm)	P1 (mm)	CL (mm)	CW (mm)
AFE8221IRFPQ1	RFP	HTQFP	144	60	5 x 12	150	315	135.9	7620	25.4	17.8	17.55

RFP (S–PQFP–G144) PowerPAD™ PLASTIC QUAD FLATPACK



NOTES:
A. All linear dimensions are in millimeters.
B. This drawing is subject to change without notice.
C. Body dimensions do not include mold flash or protrusion
D. This package is designed to be soldered to a thermal pad on the board. Refer to Technical Brief, PowerPad Thermally Enhanced Package, Texas Instruments Literature No. SLMA002 for information regarding recommended board layout. This document is available at www.ti.com <http://www.ti.com>.
E. See the additional figure in the Product Data Sheet for details regarding the exposed thermal pad features and dimensions.
F. Falls within JEDEC MS–026

PowerPAD is a trademark of Texas Instruments.

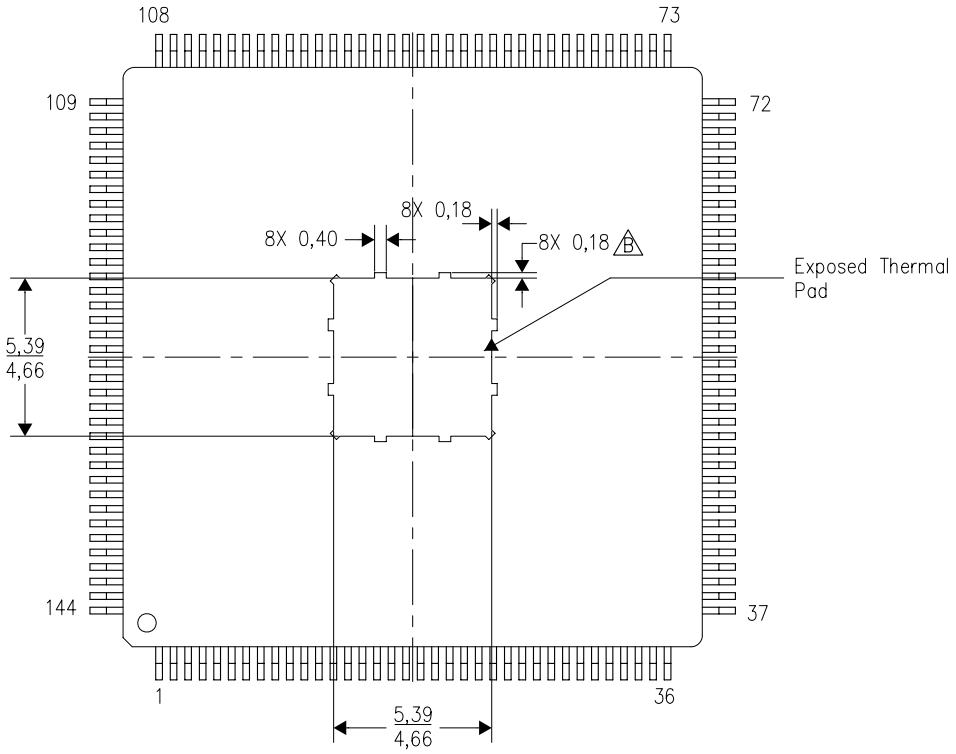
RFP (S–PQFP–G144) PowerPAD™ PLASTIC QUAD FLATPACK

THERMAL INFORMATION

This PowerPAD™ package incorporates an exposed thermal pad that is designed to be attached to a printed circuit board (PCB). The thermal pad must be soldered directly to the PCB. After soldering, the PCB can be used as a heatsink. In addition, through the use of thermal vias, the thermal pad can be attached directly to the appropriate copper plane shown in the electrical schematic for the device, or alternatively, can be attached to a special heatsink structure designed into the PCB. This design optimizes the heat transfer from the integrated circuit (IC).

For additional information on the PowerPAD package and how to take advantage of its heat dissipating abilities, refer to Technical Brief, PowerPAD Thermally Enhanced Package, Texas Instruments Literature No. SLMA002 and Application Brief, PowerPAD Made Easy, Texas Instruments Literature No. SLMA004. Both documents are available at www.ti.com.

The exposed thermal pad dimensions for this package are shown in the following illustration.



Top View

Exposed Thermal Pad Dimensions

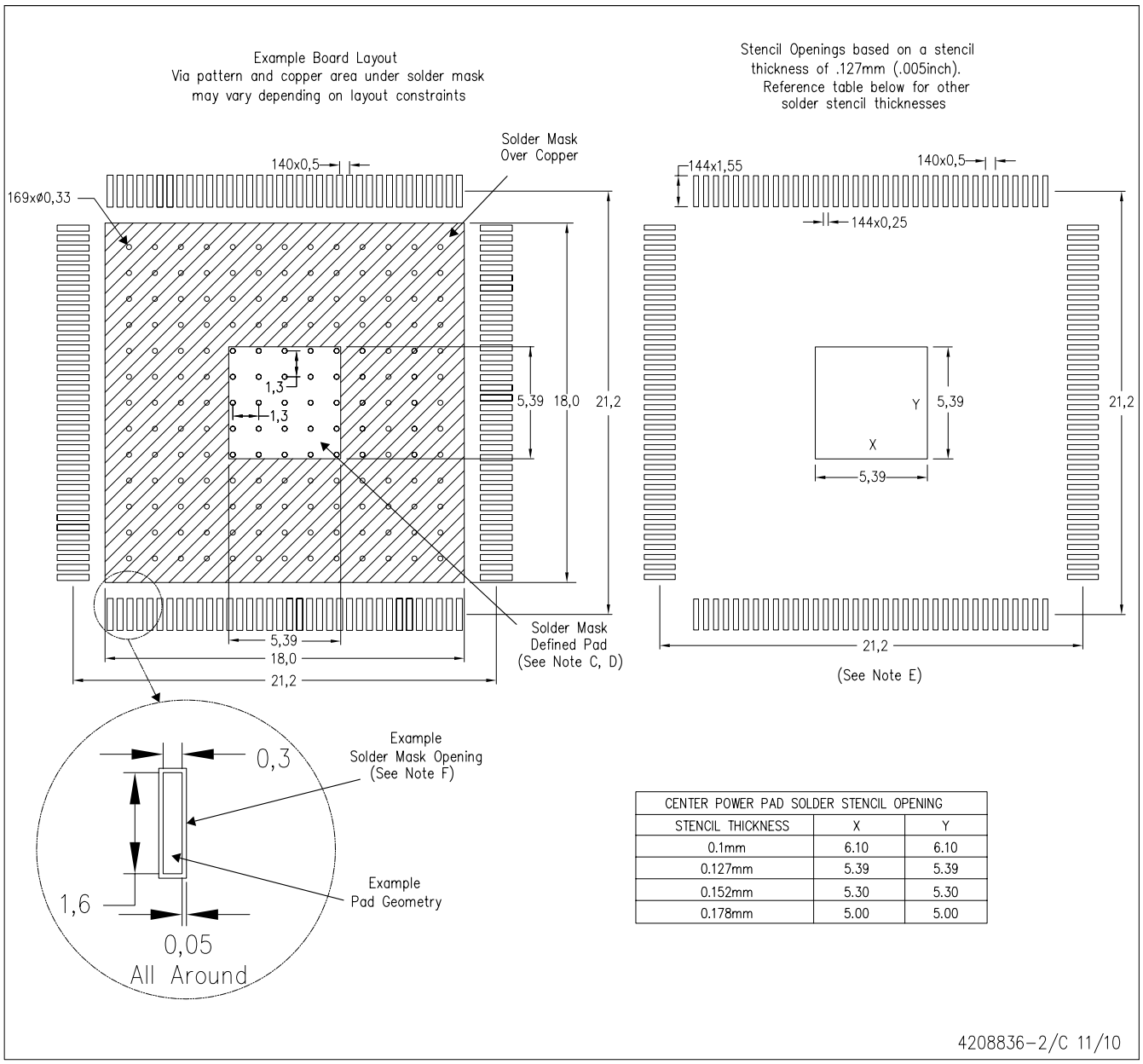
4206900–2/E 01/12

NOTES: A. All linear dimensions are in millimeters

B. Tie strap features may not be present

PowerPAD is a trademark of Texas Instruments.

RFP (S–PQFP–G144) PowerPAD™ PLASTIC QUAD FLATPACK



CENTER POWER PAD SOLDER STENCIL OPENING		
STENCIL THICKNESS	X	Y
0.1mm	6.10	6.10
0.127mm	5.39	5.39
0.152mm	5.30	5.30
0.178mm	5.00	5.00

NOTES:
A. All linear dimensions are in millimeters.
B. This drawing is subject to change without notice.
C. Customers should place a note on the circuit board fabrication drawing not to alter the center solder mask defined pad.
D. This package is designed to be soldered to a thermal pad on the board. Refer to Technical Brief, PowerPad Thermally Enhanced Package, Texas Instruments Literature No. SLMA002, SLMA004, and also the Product Data Sheets for specific thermal information, via requirements, and recommended board layout. These documents are available at www.ti.com <http://www.ti.com>. Publication IPC–7351 is recommended for alternate designs.
E. Laser cutting apertures with trapezoidal walls and also rounding corners will offer better paste release. Customers should contact their board assembly site for stencil design recommendations. Example stencil design based on a 50% volumetric metal load solder paste. Refer to IPC–7525 for other stencil recommendations.
F. Customers should contact their board fabrication site for solder mask tolerances between and around signal pads.

PowerPAD is a trademark of Texas Instruments.

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATA SHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products.

TI objects to and rejects any additional or different terms you may have proposed.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265