TABLE OF CONTENTS

1. INTRODUCTION

This document describes the functionality and digital interface for the following 5G Tx/Rx dual-pol 8-channel beamformer ICs (BFICs):

- AWMF-0151 28 GHz BFIC.
- AWMF-0159 39 GHz BFIC
- AWMF-0165 26 GHz BFIC.

2. DEVICE OVERVIEW

The AWMF-0151, AWMF-0159 and AWMF-0165 are highly-integrated silicon 8-channel ASICs intended for 5G phased array applications. The devices support up to 8 radiating elements and include all requisite beam-steering controls for 6-bit phase and 5-bit gain adjustment. There are two quads (a quad is defined as a group of 4 channels connected to the same common port) referred to in this document as A and B. The two quads can be used to support either 4 dual-pol radiating elements or 8 single-pol radiating elements.

In the case when the device is connected to 4 dual-pol antennas, the element ports of quad A are connected to one polarization on these antennas and the element ports of quad B are connected to the other polarization on these same antennas. The common ports of the two quads can be stimulated with different uncorrelated data streams in this case.

In the case when the device is connected to 8 single-pol antennas, each of the 8 element ports is connected to a different antenna element. Both common ports are stimulated with the same data stream. This case requires an external power combiner (e.g., PCB Wilkinson) to feed the common ports A and B.

2.1 Design Features

Figure 1 shows a functional block diagram of the device.
Figure 2 identifies the WLBGA pin-out mapping.

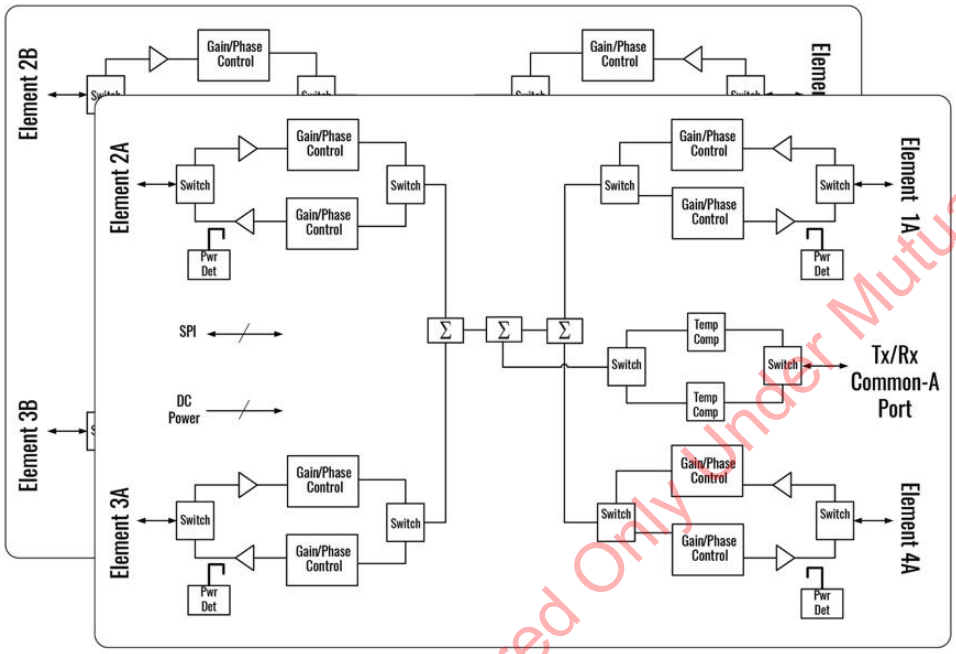

Figure 1: Simplified ASIC Block Diagram

These devices incorporate functions and features optimized for mmWave 5G phased-array radios:

- Serial daisy chain support for scalable phased arrays.
- Fast beam steering (FBS).
  - On-chip memory to store 512 beam angle settings.
  - Rapid broadcast commands enable symbol-by-symbol steering, during the cyclic prefix (CP).
- 3-D beam steering (TDBS).
  - On-chip memory to store 64 Tx and 64 Rx gain settings.
  - Fast power-down of selected channels, for beam-widening and DC power savings.
  - Dynamic gain reconfiguration for array taper.
  - Rapid broadcast commands enable symbol-by-symbol taper changes, during CP.
- Adjustment of Rx gain to achieve higher IIP3.
- 6-bit Tx power detector at each element port.
- 6-bit temperature sensor.
- ZERO-CAL feature eliminates or reduces array calibration requirements.
- Automatic temperature compensation of Tx/Rx gain.
  - Legacy manual mode also supported.
- Tx/Rx gain control to enable use of the ASIC as a driver IC.

	13	12	11	10	9	8	7	6	5	4	3	2	1	
A		GND	VDDPA_3A	RX_EN	SPI_SDO	GND	RF_COM_A	GND	SPI_CLK	TX_EN	VDDPA_2A	GND		A
B	RF3_A	GND	VDDPA_3A	GND	VDD_1P8	GND		GND	VDD_1P8	GND	VDDPA_2A	GND	RF2_A	B
C	GND	GND	VDDPA_3B		VDD_1P8	GND		GND	VDD_1P8		VDDPA_2B	GND	GND	C
D	RF3_B	GND	VDDPA_3B	GND	VDD_1P8	VDD_1P8		VDD_1P8	VDD_1P8	GND	VDDPA_2B	GND	RF2_B	D
E	GND	GND	GND	GND	GND				GND	GND	GND	GND	GND	E
F	GND	GND	GND		GND				VDD_DIG_1P8	GND_DIG	GND	GND	GND	F
G	RF4_B	GND	VDDPA_4B	GND	VDD_1P8	VDD_1P8		VDD_1P8	VDD_1P8	GND	VDDPA_1B	GND	RF1_B	G
H	GND	GND	VDDPA_4B		VDD_1P8	GND		GND	VDD_1P8		VDDPA_1B	GND	GND	H
J	RF4_A	GND	VDDPA_4A	GND	VDD_1P8	GND		GND	VDD_1P8	GND	VDDPA_1A	GND	RF1_A	J
K		GND	VDDPA_4A	SPI_CSB	SPI_SDI	GND	RF_COM_B	GND	SPI_PDI	SPI_LDB	VDDPA_1A	GND		K
	13	12	11	10	9	8	7	6	5	4	3	2	1	

Figure 2: WLBGA Ball Map Pin-Out (Bottom View)

3. Device Operating Modes

These devices have 4 operating modes defined by the states of the RX_EN and TX_EN digital input pins. Table 1 summarizes these modes.

Table 1: Device Operating Modes

MODE	RX_EN	TX_EN	Description
Standby	0	0	RF circuits powered down. Fast switching to Tx/Rx modes.
Tx Mode	0	1	Transmit mode.
Rx Mode	1	0	Receive mode.
Sleep	1	1	RF/bias circuits powered down. Low power dissipation. Slow switching to Tx/Rx modes.

4. Digital Interface And SPI Architecture

4.1 SPI Overview

These ASICs are digitally programmed via a 5-wire serial peripheral interface (SPI). A sixth signal (SPI_LDB) is also provided for backward compatibility. All digital I/O pins are compatible with 1.8V CMOS logic levels. Table 2 summarizes the digital interface pin-out.

Table 2: Digital I/O Pin Descriptions

Pin	Description
SPI_CLK	Clock input.
SPI_SDI	Serial data input. During broadcast commands using SPI_PDI pin, SPI_SDI pin should be driven low.
SPI_PDI	Broadcast data input. During serial commands using SPI_SDI pin, or if SPI_PDI is never used, it should be driven low.
SPI_SDO	Serial data output.
SPI_CSB	Chip select (active low) input to enable SPI transactions.
SPI_LDB	Latch data (active low) input (for backward compatibility). If SPI_LDB is not used, it should be driven low.

The SPI_SDI and SPI_PDI pins are physically OR'ed in the ASIC, and can be used interchangeably.

The SPI interface provides for write and read operations in either a **Serial Mode** or a **Broadcast Mode** for ICs in an array topology. All 6 pins should be connected to the controller on the schematic/PCB. The FPGA controller logic must implement serial mode and may implement broadcast mode to improve array steering speed.

In a **Serial Mode** transaction, data is shifted in on SPI_SDI and shifted out on SPI_SDO. The shifted out data during a serial write command comprises the selected readback register data followed by the shifted in data packet. SPI_PDI must be set to 0 during serial transactions.

In a **Broadcast Mode** transaction, data is shifted in on the SPI_PDI pin. This mode provides a parallel write command to a specific register in all ICs in the (selected) array. These commands may be thought of as "broadcast" writes to an array of ICs. SPI_SDI must be set to 0 during broadcast transactions.

The broadcast mode also provides a set of fast-access write commands that enable fast-beam steering (FBS) using phase controls and 3-D beam steering (TDBS) using gain controls. FBS/TDBS commands select a preset beam weight from internal memories in the ICs. The internal FBS/TDBS memories in each IC in the array are serially programmed with unique contents at startup. The contents of the FBS memory can be programmed so that for an FBS index N, the array beam will steer to a desired direction. The SPI_SDO pin is held in tri-state during broadcast mode transactions and telemetry/register readback is not supported in this mode.

The SPI interface differs from a traditional SPI interface in that it uses synchronous rising edge timing. Data at the SDO output is clocked out by the rising edge of SPI_CLK, and is latched at the SDI input of the next IC on the next rising edge of SPI_CLK. A transaction always begins with the falling edge of the chip select signal SPI_CSB. Note: The first SDO bit output from the controller (or daisy chained IC) is presented at the falling edge of SPI_CSB. For serial transactions, this first bit is always 0. Write data is latched into registers/memory on the rising edge of SPI_CSB. In a legacy mode, write data is latched into the target registers in the ICs synchronously on the falling edge of SPI_LDB after SPI_CSB is de-asserted (rises). Data is always presented MSbit first.

The serial data out SPI_SDO pin is held in a high impedance state, or tri-state, when the IC is not selected. It is recommended (for steering command speed) that the controller have dedicated SDO/SDI pins for each daisy chain to write all daisy chains simultaneously. However, if it is desired to save pins on the controller, SPI_SDO pins of multiple ICs can be connected together, as long as only one IC is selected at a time. A daisy chain length of 4-8 ICs is recommended.

Figure 3 illustrates, in a compact form, the format of the different command types. Details of each mode are provided in following sections.

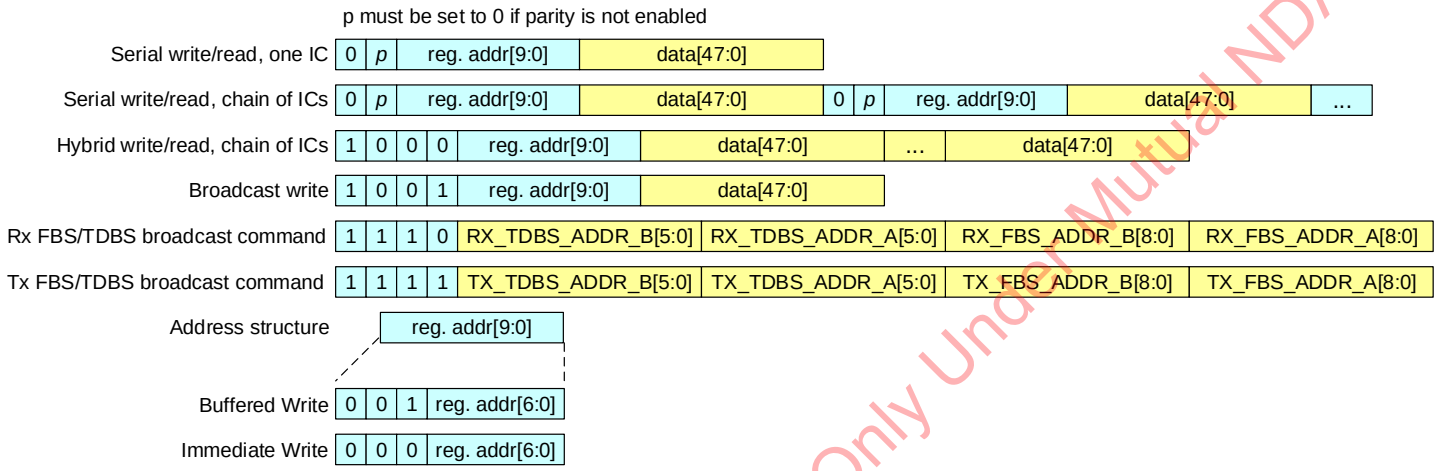

Figure 3: SPI Command Formats

Serial transactions start with two leading zero (0) bits followed by the 10-bit register address and the 48-bit data word. The Serial mode control word is always 12 bits long. Serial writes must always send exactly 60 bits per IC in the chain to match the 60-bit shift register length. Note: an optional odd parity mode can be enabled, in which case the 2nd MSbit of the control word implements odd **parity bit** over the (single target IC's) entire 60-bit transfer. Otherwise you must set the 2nd MSbit to 0. Write data is always assumed aligned such that the last received bit is shifted into the register LSB prior to the end of the transaction.

A **buffer bit** (3rd MSbit of Address) can be set in the Serial transaction control word. This indicates that the register write should be buffered and applied in the future. All pending buffered register writes are applied simultaneously, following: 1) a write where the buffer bit is **not** set or 2) a broadcast FBS/TDBS steer command. This feature enables multiple serial writes to take effect (steer) simultaneously (e.g. to configure the phase of A and B quads simultaneously). Otherwise set the upper 3 MSbits of address to 0. Note that each buffered write transaction still needs to be followed by a CSB rising edge or an LDB falling edge to indicate the end of each transaction.

Broadcast transactions start with a leading one (1) bit. A broadcast write to any register has a 14-bit control word including the 10-bit register address, which is followed by the 48-bit data word. Broadcast-mode register writes are 62 bits long. Set SPI_SDI=0.

The broadcast **TDBS/FBS** write transactions are 34 bits long. They are sent on the SPI_PDI pin. Set SPI_SDI=0. The command consists of a 4-bit opcode, separate 6-bit gain memory addresses (TDBS) for A and B quads, and separate 9-bit phase memory addresses (FBS) for A and B quads. A single

broadcast command selects the addresses to be used for both quads; separate commands are needed for changing TX and RX beamweights. The FBS and TDBS memories must be programmed using serial transactions, ahead of time. The TDBS and FBS address contents can be thought of as 'presets', where a preset represents a steer-angle and taper for the whole phased array.

4.1.1 Serial Mode Operation

The Serial Mode implements a combined write and read transaction for either a single IC or any number of ICs in a series daisy chain. Before the SPI transaction starts, the controller shift register contains write data for the ICs, and each IC's shift register contains its readback data. After the SPI transaction completes, the controller shift register contains readback data from the ICs, and each IC's shift register contains its write data. For a daisy chain with *N* ICs, a serial command will have 60**N* spi_clk cycles, and the controller shift register length must be 60**N*. A typical steer command will target the same register address, with different values per IC. But a single SPI command could target different register addresses in different ICs. Refer to the Memory Map or Register Map for register details.

4.1.1.1 Serial Write/Read, Two IC Chain

A Serial Write/Read transaction with a controller and two ICs is illustrated in Figure 4 and Figure 5. At the start of the transaction, the controller IC has the address and data ready to shift to each IC. Each IC has read data from the preselected register ready to shift to the controller. When the transaction is complete, each IC shift register contains the address and data from the controller, and the controller shift register contains the read data from the ICs. Both ICs latch (and act on) the control word and data at the completion of the transaction.

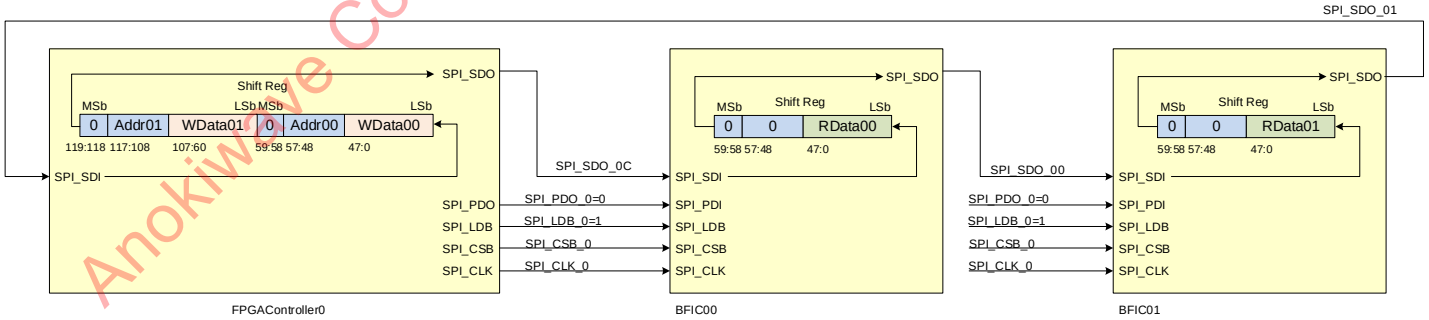

Figure 4: Shift register contents before Serial SPI Transaction

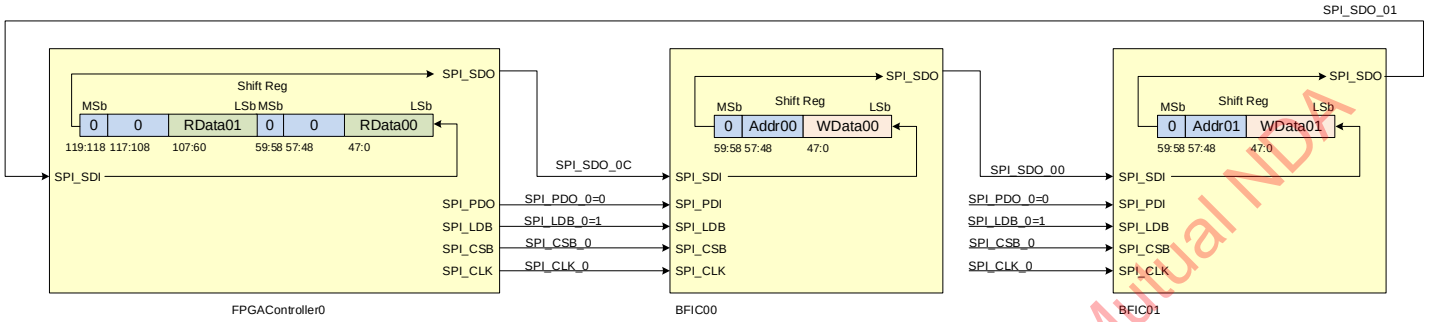

Figure 5: Shift register contents after Serial SPI Transaction (120 SPI_CLKs)

A timing diagram for the single IC serial-mode transaction is shown in Figure 6. The write starts with the falling edge of SPI_CSB. Data is shifted in MSbit first, starting with the control word followed by the data. The control word and data are provided for each IC in the serial chain, allowing unique values for each IC. A normal write concludes with SPI_CSB returning high (de-asserting). Data is latched following the rising edge of SPI_CSB. Additionally, the figure shows the optional SPI_LDB signal used in legacy applications. In this mode, a falling edge on SPI_LDB is used to latch the data into the target register, and then a rising edge of SPI_LDB to return it to its high idle level. Note: MODE register bit 22 (spi_ldb_en) controls LDB behavior. By default, a signal INT_LDB is internally generated following the rising edge of SPI_CSB, and the SPI_LDB pin is ignored. If spi_ldb_en is set to 1, INT_LDB is ignored, and SPI_LDB is used to latch the register data. There is typically 40ns settling time for the steer data to become valid, after the rising edge of SPI_CSB, when INT_LDB is used, indicated by the arrows in Figure 6. SPI_LDB should be connected on the PCB, for backward/forward compatibility, and must be driven to 0 when spi_ldb_en=0. If SPI_LDB is used, there is typically a 30ns settling time.

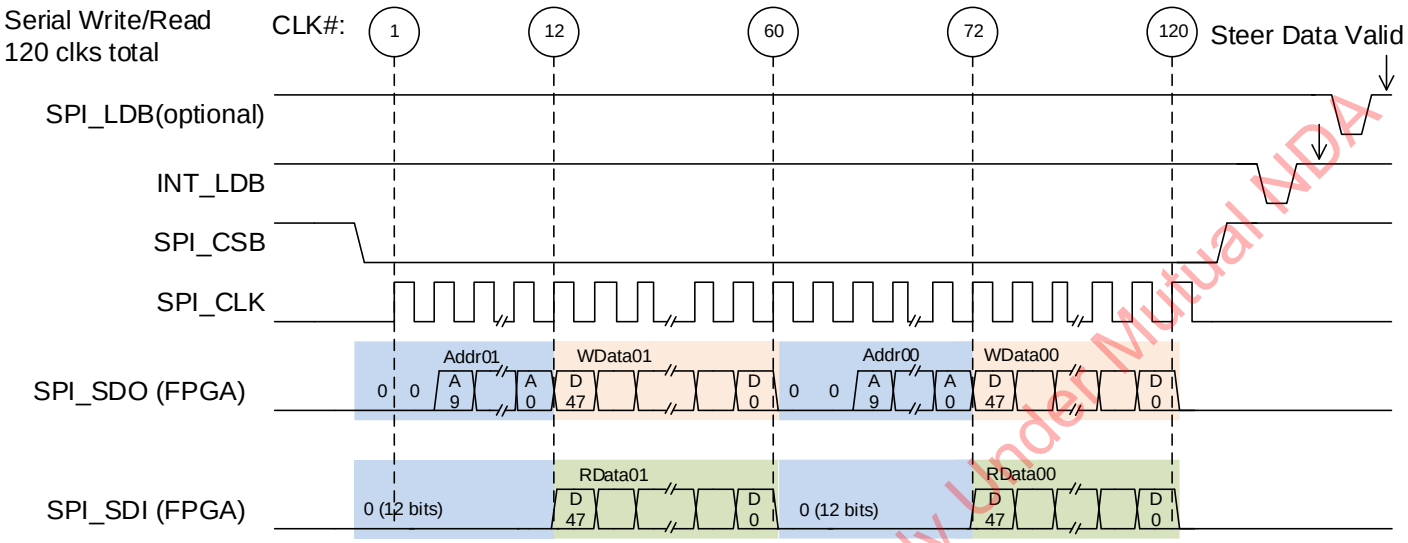

Figure 6: Timing diagram for a two IC Serial Write/Read transaction

Shift Example:

Refer to Figure 4, Figure 6 above. There are two BFICs in the daisy chain in this example, so the FPGA shift register length = 2 * (12 + 48) = 120 bits.

1) The initial state of the shift registers is shown above in Figure 4.
2) FPGAController0 loads its shift register with write data for BFIC00, BFIC01. Note BFIC00 and BFIC01 will have their shift registers loaded with the read data at the falling edge of SPI_CSB.
3) When SPI_CSB goes low (before CLK1), FPGAController0 outputs the first bit of the header (MSbit[119] of shift register = 0). Each BFIC (00 and 01) also outputs the first bit of the header (MSbit[59] of shift register = 0). No clock is required for this operation.
4) On the rising edge of CLK1:
    a. FPGAController0 shift register shifts left. FPGAController0 latches (in LSbit of its shift register) the MSbit of read data from BFIC01 (0) from SPI_SDO_01. After tPDO (6ns max), SPI_SDO_0C settles to the value of bit 118 (MSbit-1).
    b. BFIC00 shift register shifts left. BFIC00 latches the MSbit of write data from FPGAController0 (0) from SPI_SDO_0C. After tPDO (6ns max), SPI_SDO_00 settles to the value of bit 58 (MSbit-1).
    c. BFIC01 shift register shifts left. BFIC01 latches the MSbit of read data from BFIC00 (0) from SPI_SD0_00. After tPDO (6ns max), SPI_SDO_01 settles to the value of bit 58 (MSbit-1).
5) On the rising edge of CLK61:

a. FPGAController0 shift register shifts left. FPGAController0 latches the MSbit of read data from BFIC01 (which was originally BFIC00 bit 59) from SPI_SDO_01. After tPDO (6ns max), SPI_SDO_0C settles to the value of bit 58 (=0 2nd bit of 2nd header).
b. BFIC00 shift register shifts left. BFIC00 latches (initial FPGA shift register) bit 59 of write data from FPGAController0 (0) from SPI_SDO_0C. After tPDO (6ns max), SPI_SDO_00 settles to the value of (initial FPGA shift register) bit 118.
c. BFIC01 shift register shifts left. BFIC01 latches (initial FPGA shift register) bit 119 of write data from FPGAController0 (0) from SPI_SDO_00. After tPDO (6ns max), SPI_SDO_01 settles to the value of (initial BFIC00 shift register) bit 58 (MSbit-1 = 0).

4.1.1.2 Readback

Note that every serial mode transaction returns readback data, shifted out on the SPI_SDO pin. The readback data (ex. temperature) is captured internally by the IC at the start of the transaction, then shifted out. The readback word does not contain the read address, so the first 12 bits of the readback word are all 0's. By default, register TELEM0 containing version and telemetry data is read back. Each IC has a register (RADDR) that contains the register address that will be read, allowing the host to read other data besides the default telemetry data. The register contents selected by RADDR will be shifted out on every SPI transaction. The first time you wish to read a particular address, *N*, there will be two steps:

1) Write RADDR = *N*
2) Perform a second SPI transaction to shift out the read data.

If you do not need to write any data, but need to shift out readback data you can:

a) Write to a read only register (such as VERSION)
b) Write to a test register (SPARE_TST)
c) Write **valid** data to any register (ex. repeat previous command). Be careful not to write invalid data (e.g, all 0's).

Every subsequent SPI transaction will shift out data read from register *N* until RADDR is changed.

4.1.2 Broadcast Mode

Broadcast write transactions transfer identical control and data words to all ICs in a group simultaneously. Ensure SPI_SDI=0 during parallel writes. Figure 7 shows a timing diagram for broadcast mode write to a register.

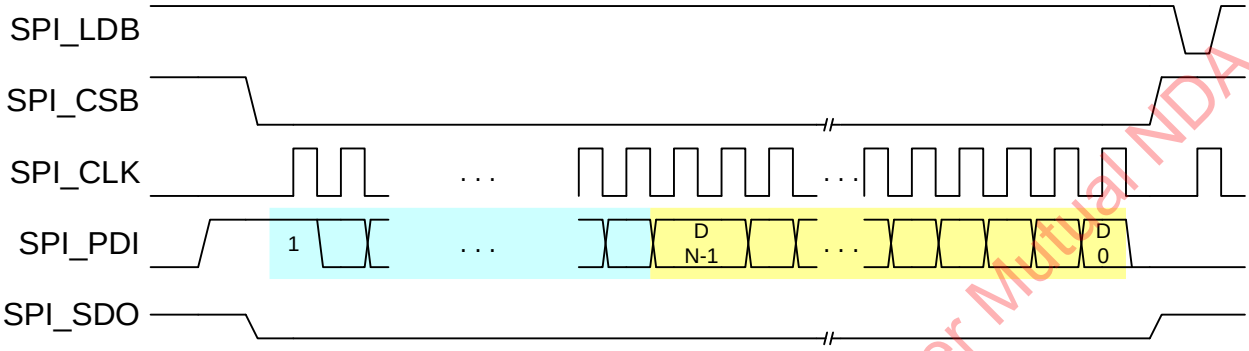

Figure 7: Timing Diagram for Broadcast Write

4.2 Memory/Register Map

The IC address space is shared between programmable registers, special-purpose registers, and memories for fast-beam steering, 3-D beam steering and temperature compensation. The addresses are 10-bit long, and all registers are 48 bits in length.

As shown in Table 3, the first part of the register map is dedicated to programmable registers in the serial interface, from addresses 0x000 to 0x03F. The second part of the register map, from addresses 0x100 to 0x2FF, is dedicated to the memories that can be accessed using broadcast-mode commands (FBS/TDBS). These addresses provide direct serial write access to the memories, with two words/entries per register.

Memory tables including fast-beam steering and 3-D beam steering will be populated with ‘unknown’ entries on power up. The user should not rely on memories defaulting to all 0s state on power up. Also, note that issuing an RFIC reset does not affect the contents of the FBS and TDBS memory tables.

Table 3: Memory Map

Address [9:0]	Register Name	Register Description	Type
0x00	MODE	IC configuration	r/w
0x01	BW_TX_A	Beam weight controls for Tx channels in quad A	r/w
0x02	BW_RX_A	Beam weight controls for Rx channels in quad A	r/w
0x03	RX_MEMADDR	Readback Rx FBS/TDBS memory pointers.	r
0x04	TX_MEMADDR	Readback Tx FBS/TDBS memory pointers.	r
0x05	ATTTC	Gain Controls for Manual Temperature Compensation	r/w
0x0D	ENGR0	Miscellaneous ATC & Tx/Rx override controls.	r/w
0x16	ATC_CALC	Readback ATC algorithm outputs	r
0x1F-0x21	ATC_CONFIG	ATC configuration registers	r/w
0x22	BW_TX_B	Beam weight controls for Tx channels in quad B	r/w
0x23	BW_RX_B	Beam weight controls for Rx channels in quad B	r/w
0x30	TELEM0	Telemetry 0 (default)	r
0x31	TELEM1	Telemetry 1	r
0x32	TELEM2	Telemetry 2	r
0x33	TELEM3	Telemetry 3	r
0x3A	SPARE	Spare/Dummy Register	r/w
0x3D	RADDR	Stores the address of readback register	r/w
0x3E	PROD_ID	Part number and version information	r
0x100	Rx TDBS Memory	Rx TDBS entries 1, 0	r/w
…		…	
0x11F		Rx TDBS entries 63, 62	r/w
0x140	Tx TDBS Memory	Tx TDBS entries 1, 0	r/w
…		…	
0x15F		Tx TDBS entries 63, 62	r/w
0x200	FBS Memory	FBS entries 1, 0	r/w
…		…	
0x2FF		FBS entries 511, 510	r/w

Details of the register map can be found in Appendix A1.

4.2.1 MODE Register

The MODE register provides control of several aspects of the chip configuration and options, including:

- software reset
- SPI options
- fast beam steering memory selection options
- beam weight selection options
- Zcal initialization

4.2.1.1 Software Reset (sw_reset)

The MODE/sw_reset bit resets all logic and programmable registers in the IC to their default values. To use this bit, program the corresponding MODE register bit (bit 22) to 1, and then program the MODE register again to the MODE register default value, which has bit 22 cleared (0). SPI serial/parallel modes may be used to issue the reset command. Always issue a sw_reset after power is applied to the IC.

4.2.1.2 Enable ZCal (zcal_init)

The zcal_init is a 1-bit field that applies internally stored calibration factors to the BFIC circuitry. To apply the calibration factors, a 1 must be written to this field. Successful loading of the calibration factors is indicated by the status field in TELEM0 [30], zcal_applied=1. The zcal_init bit is self-clearing. To disable the Zcal engine, the BFIC should either be power cycled, or a sw_reset=1 issued. Querying the TELEM0 [30] shall then indicate zcal_applied=0.

4.3 Fast Beam Steering (FBS) Memories and Operation

Fast beam steering phase values for each beam direction are stored in Fast Beam Steering memory, which are addressable through normal SPI operations in the shared address space. FBS values may be selected for the array using the broadcast mode writes for Rx/Tx FBS. Setting the MODE/spi_fbs_sel bit allows the selected FBS memory data to be presented to the Rx and Tx elements.

The structures of the FBS memories are shown in Figure 8. Note that each 48-bit register in the FBS memories stores two FBS states, hence 256 registers are used to store 512 FBS states. The FBS memory can be programmed using serial mode SPI transactions; each transaction will consist of data

for two FBS states. Also note that the format of each 24-bit FBS state is the same as the bits 23:0 in the beamweight registers (registers 0x01, 0x02, 0x22, 0x23), and includes 6-bit phase-shift control for each channel in a quad (a.k.a. pol).

Note: while switching between different FBS states using broadcast FBS write, the FBS memory index is used and not the SPI register address. See Figure 3.

SPI address	FBS row	column1	column0
0x200	0	FBS1[23:0]	FBS0[23:0]
0x201	1	FBS3[23:0]	FBS2[23:0]
0x202	2	FBS5[23:0]	FBS4[23:0]
…	…	…	…
0x2FF	255	FBS511[23:0]	FBS510[23:0]

Figure 8: FBS memory map

4.3.1 Writing to FBS Memory

The ASICs are equipped with independent memory locations in which to store the fast beam steering (FBS) phase states for transmit and receive modes. There are 256 registers available for FBS storage, as illustrated below.

FBS Registers (256 registers)	0x200-0x2ff

There are 512 states available. Two phase states are stored per register, where a phase state comprises of the four element phase code values. Each phase code is 6 bits, yielding 24 bits of information per phase state. Thus two phase states occupy the complete 48 bits of a register.

If the 6-bit phase code for channel *M* is denoted as PMn[5:0] and *n* denotes the n^{th} location in the FBS memory, then the FBS register can be represented as shown below.

48 bit register containing two FBS states	
$P4_{n+1}[5:0]$ $P3_{n+1}[5:0]$ $P2_{n+1}[5:0]$ $P1_{n+1}[5:0]$	$P4_n[5:0]$ $P3_n[5:0]$ $P2_n[5:0]$ $P1_n[5:0]$

In order to illustrate the process of programming the FBS registers, consider the example where the 10^{th} and 11^{th} FBS memories of the transmit phase shifters are to be loaded. The phases/phase codes for the two phase states are shown below.

FBS #	Element 1	Element 2	Element 3	Element 4
10	11.25°/2	22.5°/4	45°/8	90°/16
11	180°/32	270°/48	0°/0	270°/48

Thus register 0x205 should be written with the following:

Register	Data
0x205	c0-0c-20-40-81-02

In order to read the contents of 0x205, the following command would be sent twice: once to set the address to read from, and again to read back the data.

Register	Data
0x03d	00-0-00-00-02-05

4.4 3-D Beam Steering (TDBS) Memories and Operation

The 3-D Beam Steering amplitude values are stored in the TDBS memory, which is addressable through normal SPI operations in the shared address space. TDBS values may be selected for the array using the broadcast mode writes for TDBS. Setting the MODE/spi_tdbs_sel bit high allows the selected TDBS data to be presented to the Tx/Rx elements.

The structures of the TDBS memories are shown in Figure 9. Note that each 48-bit register in the TDBS memories stores two TDBS states, hence 32 registers are used to store 64 TDBS states (each, for Rx and Tx). The TDBS memory can be programmed using serial mode SPI transactions; each transaction will consist of data for two TDBS states. Also note that the format of each 24-bit TDBS state is the same as the bits 47:24 in the beamweight registers (registers 0x01, 0x02, 0x22, 0x23), and includes 4-bit gain control for each channel in a quad (a.k.a. pol), 4-bit gain control for the common arm in the quad and disable bits for each channel in a quad.

Note: while switching between different TDBS states using broadcast TDBS write, the TDBS memory index is used and not the SPI register address. See Figure 3.

SPI address	RX_TDBS row	column 1	column 0
0x100	0	RX_TDBS1[23:0]	RX_TDBS0[23:0]
0x101	1	RX_TDBS3[23:0]	RX_TDBS2[23:0]
0x102	2	RX_TDBS5[23:0]	RX_TDBS4[23:0]
. . .	. . .	. . .	. . .
0x11F	31	RX_TDBS63[23:0]	RX_TDBS62[23:0]

SPI address	TX_TDBS row	column 1	column 0
0x140	0	TX_TDBS1[23:0]	TX_TDBS0[23:0]
0x141	1	TX_TDBS3[23:0]	TX_TDBS2[23:0]
0x142	2	TX_TDBS5[23:0]	TX_TDBS4[23:0]
. . .	. . .	. . .	. . .
0x15F	31	TX_TDBS63[23:0]	TX_TDBS62[23:0]

Figure 9: TDBS Memory Map

4.4.1 Writing to TDBS Memory

The ASICs are equipped with independent memory locations in which to store the three-dimensional beam steering (TDBS=attenuation) states for transmit and receive modes. Each mode has 32 registers available for TDBS storage, as illustrated below.

RX TDBS Registers (32 registers)	0x100-0x11f
TX TDBS Registers (32 registers)	0x140-0x15f

Each mode has 64 TDBS states available, where a TDBS state comprises of the element arm disables (4 bits), the common arm taper code (4 bits) and the element arm taper codes (4 channels each of 4 bits), yielding a total of 24 bits. Two TDBS states are stored per register. Organization of a TDBS register is illustrated below, where *n* denotes the n^{th} TDBS state, Dm denotes the 1-bit disable state of element *m*, Am denotes the 4-bit taper code of the m^{th} element and AC denotes the 4-bit common arm taper code.

48 bit register containing two TDBS states	
$D4_{n+1}[0]$ $D3_{n+1}[0]$ $D2_{n+1}[0]$ $D1_{n+1}[0]$ $AC_{n+1}[3:0]$ $A4_{n+1}[3:0]$ $A3_{n+1}[3:0]$ $A2_{n+1}[3:0]$ $A1_{n+1}[3:0]$	$D4_n[0]$ $D3_n[0]$ $D2_n[0]$ $D1_n[0]$ $AC_n[3:0]$ $A4_n[3:0]$ $A3_n[3:0]$ $A2_n[3:0]$ $A1_n[3:0]$

To illustrate the process of programming the TDBS registers, consider the example where the 10th and 11th TDBS memories of the receiver are to be loaded. The states for the two TDBS entries are shown below.

TDBS#	Dis D1..D4	AC[3:0]	A1[3:0]	A2[3:0]	A3[3:0]	A4[3:0]
10	1010	5	0	1	2	3
11	0101	15	8	4	2	1

Thus, register 0x105 should be written with the following:

Register	Data
0x105	af-12-48-55-32-10

To read the content of 0x105, the following command would be sent twice. Once to set the address to read from, and again to read back the data.

Register	Data
0x03d	00-0-00-00-01-05

4.5 ZCal Feature

The ZCal feature reduces the part to part variation for the BFICs. This feature can correct for variations in RF gain and RF phase, as well as improve accuracy of the power detector and temperature sensor. This feature is enabled by setting the MODE/Zcal_init bit to a 1. This bit must be set high to initiate the Zcal engine. Verification that the ZCal feature is operational can then be observed in the "ZCal Applied" bit in the TELEM0 register.

***Note**: All BFIC parts in this family have the Zcal option, although the initial silicon for advance access may not have this feature enabled. For production parts with Zcal enabled, please use the instructions in this document to utilize Zcal feature for your system design. Please contact your local sales contact to confirm the Zcal availability on your parts.*

4.6 Temperature Compensation

The Tx/Rx gain of these devices can be kept relatively constant by using the temperature compensation (temp-comp) features. Two temp-comp modes are available: manual and automatic. Many options and controls are provided to allow flexibility in system-level implementation, as described below.

4.6.1 Manual Temperature Compensation

In the manual approach, the temp-comp settings in Register 0x005 can be changed by the host controller using SPI transactions. In an array, the same settings could be written to all the ICs using broadcast mode transactions. Tables 4 and 5 list the recommended settings across temperature for AWMF-0151 Tx and Rx, respectively. Tables 6 and Table 7 list the recommended settings across temperature for AWMF-0165 Tx and Rx, respectively. Tables 8 and 9 list the recommended settings across temperature for AWMF-0159 Tx and Rx, respectively.

Table 4: AWMF-0151 Tx Temperature Compensation Settings

Die Backside Temperature (°C)	spi_tc_txc[4:0]	spi_tc_txq[4:0]
-35	22	18
-30	22	18
-25	21	17
-20	20	16
-15	20	16
-10	19	15
-5	18	14
0	18	14
5	17	13
10	17	13
15	16	12
20	15	11
25	15	11
30	14	10
35	13	9
40	13	9
45	12	8
50	11	7
55	11	7
60	10	6
65	10	6
70	9	5
75	8	4
80	8	4
85	7	3
90	6	2
95	6	2
100	5	1
105	5	1

Table 5: AWMF-0151 Rx Temperature Compensation Settings

Die Backside Temperature (°C)	spi_tc_rxc[4:0]	spi_tc_rxq[4:0]
-35	23	18
-30	22	17
-25	21	17
-20	20	16
-15	20	16
-10	19	15
-5	18	15
0	18	14
5	17	13
10	16	13
15	15	12
20	15	12
25	14	11
30	13	10
35	13	10
40	12	9
45	11	9
50	10	8
55	10	7
60	9	7
65	8	6
70	8	6
75	7	5
80	6	5
85	5	4
90	5	3
95	4	3
100	3	2
105	3	2

Table 6: AWMF-0165 Tx Temperature Compensation Settings

Die Backside Temperature (°C)	spi_tc_txc[4:0]	spi_tc_txq[4:0]
-35	24	15
-30	24	15
-25	23	14
-20	22	14
-15	21	13
-10	21	13
-5	20	12
0	19	11
5	19	11
10	18	10
15	17	10
20	16	9
25	16	9
30	15	8
35	14	7
40	14	7
45	13	6
50	12	6
55	11	5
60	11	5
65	10	4
70	9	3
75	9	3
80	8	2
85	7	2
90	6	1
95	6	1
100	5	0
105	4	0

Table 7: AWMF-0165 Rx Temperature Compensation Settings

Die Backside Temperature (°C)	spi_tc_rxc[4:0]	spi_tc_rxq[4:0]
-35	21	16
-30	20	15
-25	20	15
-20	19	14
-15	18	14
-10	18	13
-5	17	12
0	16	12
5	16	11
10	15	11
15	14	10
20	14	10
25	13	9
30	12	8
35	12	8
40	11	7
45	10	7
50	10	6
55	9	6
60	8	5
65	8	4
70	7	4
75	6	3
80	6	3
85	5	2
90	4	1
95	4	1
100	3	0
105	2	0

Table 8: AWMF-0159 Tx Temperature Compensation Settings

Die Backside Temperature (°C)	spi_tc_txc[4:0]	spi_tc_txq[4:0]
-35	31	19
-30	31	17
-25	30	17
-20	29	16
-15	28	16
-10	27	15
-5	26	14
0	26	14
5	25	13
10	24	12
15	23	12
20	22	11
25	21	11
30	20	10
35	19	9
40	18	9
45	17	8
50	16	8
55	15	7
60	14	6
65	14	6
70	13	5
75	12	4
80	11	4
85	10	3
90	9	3
95	8	2
100	7	1
105	6	1

Table 9: AWMF-0159 Rx Temperature Compensation Settings

Die Backside Temperature (°C)	spi_tc_rxc[4:0]	spi_tc_rxq[4:0]
-35	24	18
-30	24	18
-25	23	17
-20	23	17
-15	22	16
-10	21	15
-5	21	15
0	20	14
5	19	13
10	19	13
15	18	12
20	18	12
25	17	11
30	16	10
35	16	10
40	15	9
45	15	9
50	14	8
55	13	7
60	13	7
65	12	6
70	11	5
75	11	5
80	10	4
85	10	4
90	9	3
95	8	2
100	8	2
105	7	1

4.6.2 Automatic Temperature Compensation

These ICs include an Automatic Temperature Compensation (ATC) feature to simplify system-level temp-comp implementation. No user intervention is typically required when using the ATC feature, but several options are provided to enable application-specific scenarios. By default, ATC is disabled; the *atc_bypass* bit in Register 0x0D (ENGR0) must be set to 0 and the on-chip clock oscillator for ATC algorithm must be turned on by setting the *dig_osc_en* bit in Register 0x0D high. The *atc_mode* bit in Register 0x000 (MODE) controls the temperature compensation mode; when set to 0, the ATC algorithm calculates the temp comp values but doesn't apply them automatically and waits for the host to apply the correction. When the bit is set to 1, the ATC algorithm is fully autonomous and the temp comp corrections are applied automatically. The *atc_reset* bit in Register 0x0D should also be set to 0 to synchronously reset the ATC state machine; this bit should be set to 1 before changing any of the ATC_CONFIG registers. The ATC algorithm operates by performing the following steps:

(a) estimate the IC temperature,
(b) calculate the temp-comp gain settings based on IC temperature, and
(c) update the IC gain by applying the temp-comp settings.

After measuring the temperature and calculating the temp-comp settings, the settings need to be applied by the ATC in order for the gain change to take effect at a safe time. This change can be applied immediately after the ATC calculation is complete or triggered by an event. The different trigger options are set in Register 0x021 (ATC_CONFIG5). Furthermore, Register 0x016 (ATC_CALC) is provided to read back the status of the ATC algorithm; the IC temperature and temp-comp settings calculated by ATC can be read back from this register.

4.7 ASIC Initialization Sequence

To ensure proper operation, the power-up state of the device needs to be adjusted via a SPI initialization sequence. Tables 10, 11 and 12 list the initialization SPI words to be written to the AWMF-0151, AWMF-0165 and AWMF-0159 respectively.

Table 10: AWMF-0151 Initialization Command Sequence(A1_EVM5)

Register Address	Data (48 bit Hex)
0x0000	00-00-00-01-00-00
0x0001	00-00-00-00-00-00
0x0002	00-00-00-00-00-00
0x0005	00-00-00-07-29-cb
0x0006	00-00-04-18-61-86
0x0007	00-00-04-18-61-86
0x0008	00-00-00-44-42-33
0x0009	00-00-00-b8-55-84
0x000a	00-00-05-d1-b2-92
0x000b	00-00-33-49-29-e0
0x000c	00-00-08-44-02-00
0x000d	00-00-00-a0-00-00
0x000e	00-00-28-14-0a-05
0x000f	00-00-28-14-0a-05
0x0010	00-00-00-00-00-00
0x0011	00-00-00-00-00-00
0x0012	00-00-00-00-00-00
0x0017	00-00-04-18-61-86
0x0018	00-00-04-18-61-86
0x0019	00-00-28-14-0a-05
0x001a	00-00-28-14-0a-05
0x001b	00-00-00-00-00-00
0x001c	00-00-00-00-00-00
0x001d	00-00-22-08-87-de
0x001e	00-00-24-3c-24-3f
0x001f	00-00-00-00-0f-a0
0x0020	00-00-1f-c0-00-04
0x0021	00-00-00-00-77-5f
0x0022	00-00-00-00-00-00
0x0023	00-00-00-00-00-00
0x003d	00-00-00-00-00-30
0x004e	00-00-00-00-00-00

Table 11: AWMF-0165 Initialization Command Sequence(A2_EVM6L)

Register Address	Data (48 bit Hex)
0x0000	00-00-00-01-00-00
0x0001	00-00-00-00-00-00
0x0002	00-00-00-00-00-00
0x0005	00-00-00-07-a1-a9
0x0006	00-00-04-18-61-86
0x0007	00-00-04-18-61-86
0x0008	00-00-00-43-33-33
0x0009	00-00-00-55-55-84
0x000a	00-00-03-85-b2-92
0x000b	00-00-00-49-23-e0
0x000c	00-00-10-44-04-00
0x000d	00-00-00-a8-00-00
0x000e	00-00-28-14-0a-05
0x000f	00-00-28-14-0a-05
0x0010	00-00-00-00-00-00
0x0011	00-00-00-00-00-00
0x0012	00-00-00-00-00-00
0x0017	00-00-04-18-61-86
0x0018	00-00-04-18-61-86
0x0019	00-00-28-14-0a-05
0x001a	00-00-28-14-0a-05
0x001b	00-00-00-00-00-00
0x001c	00-00-00-00-00-00
0x001d	00-00-22-08-87-de
0x001e	00-00-24-3c-24-3f
0x001f	00-00-00-00-0f-a0
0x0020	00-00-1f-c0-00-04
0x0021	00-00-00-00-77-5f
0x0022	00-00-00-00-00-00
0x0023	00-00-00-00-00-00
0x003d	00-00-00-00-00-30
0x004e	00-00-ff-c0-0f-7f

Table 12: AWMF-0159 Initialization Command Sequence(A1_EVM4H)

Register Address	Data (48 bit Hex)
0x0000	00-00-00-01-00-00
0x0001	00-00-00-00-00-00
0x0002	00-00-00-00-00-00
0x0005	00-00-00-0a-2a-2b
0x0006	00-00-04-18-61-86
0x0007	00-00-04-18-61-86
0x0008	00-00-00-44-34-33
0x0009	00-00-00-33-44-94
0x000a	00-00-09-4d-b2-db
0x000b	00-00-00-6d-b0-18
0x000c	00-00-00-44-06-00
0x000d	00-00-00-a8-00-00
0x000e	0b-b4-ca-65-32-99
0x000f	0b-b4-ca-65-32-99
0x0010	00-00-00-00-00-00
0x0011	00-00-00-00-00-00
0x0012	00-00-00-00-00-00
0x0017	00-00-04-18-61-86
0x0018	00-00-04-18-61-86
0x0019	0b-b4-ca-65-32-99
0x001a	0b-b4-ca-65-32-99
0x001b	00-00-00-00-00-00
0x001c	00-00-00-00-00-00
0x001d	00-00-22-08-87-de
0x001e	00-00-24-3c-24-3f
0x001f	00-00-00-00-0f-a0
0x0020	00-00-1f-c0-00-04
0x0021	00-00-00-00-77-5f
0x0022	00-00-00-00-00-00
0x0023	00-00-00-00-00-00
0x003d	00-00-00-00-00-30
0x004e	00-00-00-00-00-00

5. Telemetry Operation and Readback

5.1 Tx Power Detector Operation

Each of the 8 Tx elements has a power detector to provide an estimate of transmit output power. On the falling edge of SPI_CSB, the power detector ADC conversion begins. After 10 SPI_CLK cycles, readback data is latched into holding registers (for all power detectors and temperature sensor). Figure 10 shows the typical case. Note that the readback data is latched *before* the ADC conversion completes. It represents the measured power, triggered at the falling edge of SPI_CSB of the *previous* SPI transfer. In many applications, this delay does not matter. The FPGA or controller code can account for the latency by issuing two SPI transfers back-to-back, and taking the 2nd readback data, or by averaging over multiple measurements.

Typical ADC Read with latency on Telemetry
(ADC ready signal after readback data is latched)

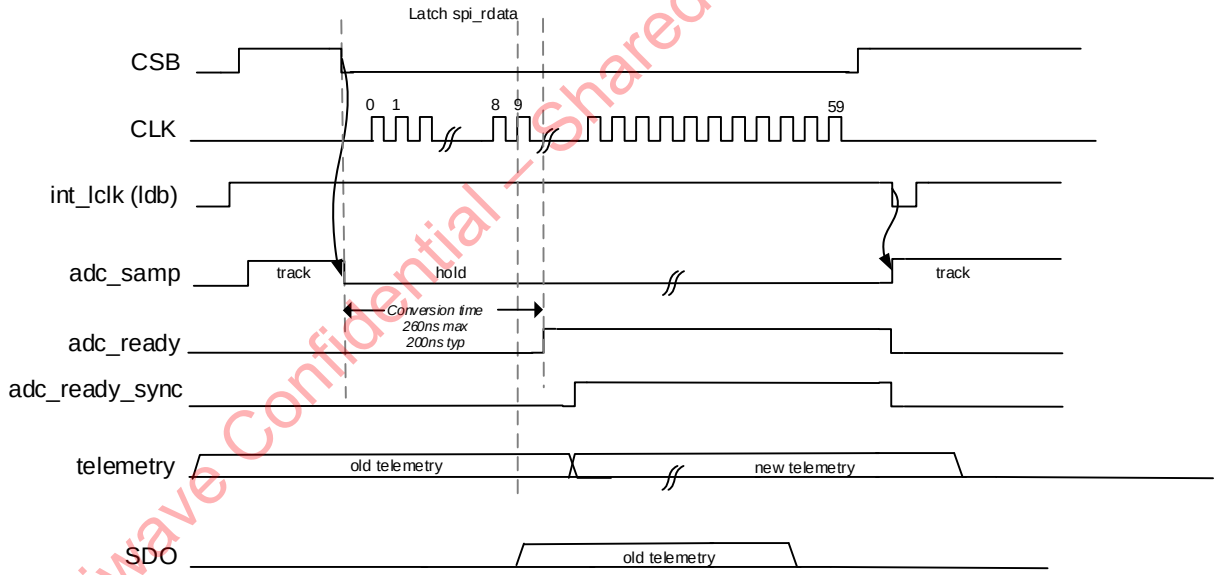

Figure 10: Typical SPI transfer with latency on Telemetry

If you wish to read back the PDET values, without an extra SPI transfer latency, see Figure 11. The controller provides a gap between the falling edge of CSB and the first SPI_CLK. If the gap plus 10 SPI_CLK cycles is greater than 260ns, the ADC acquired at the falling edge of CSB (for the *current* SPI transfer) will be read back. Note that a 2ms PDET settling time (t_{ps}) is required between the rising edge of TX_EN (or a step change in RF) and the falling edge of SPI_CSB which starts the ADC conversion.

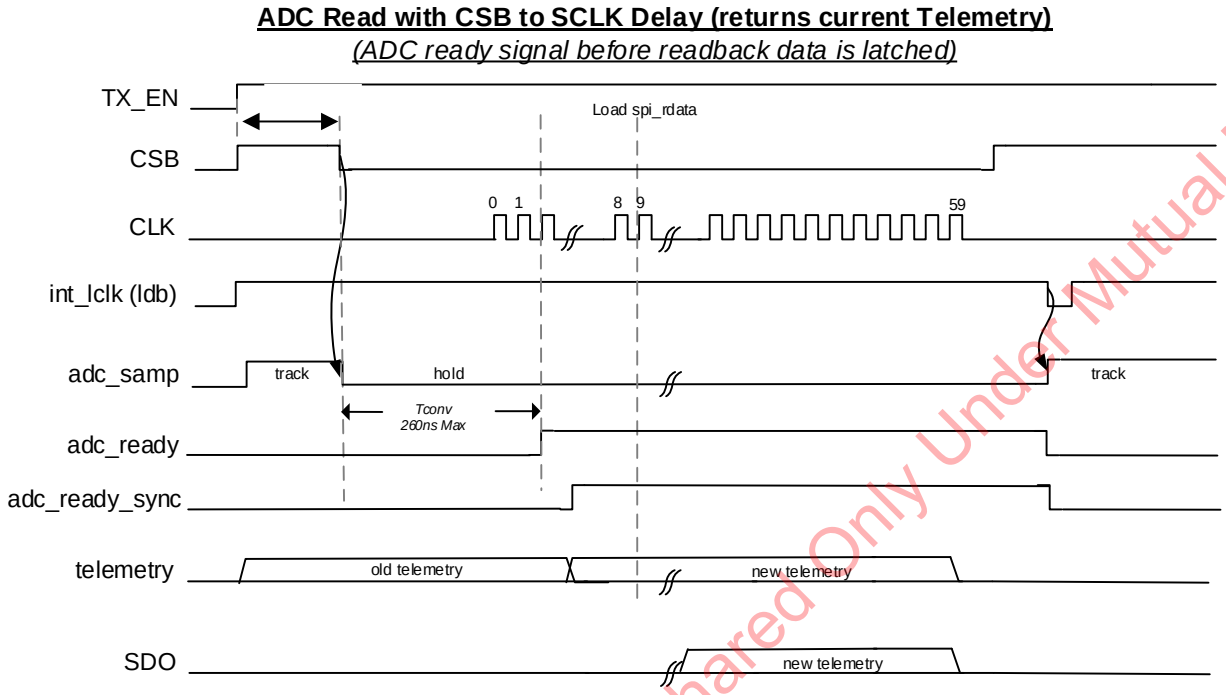

Figure 11: Gap between CSB/CLK to read current Telemetry

Example data for power detector output vs transmit output power is shown in Figure 12, 13 & 14. The ADC value is returned by the Telemetry read and can be converted to power using this graph.

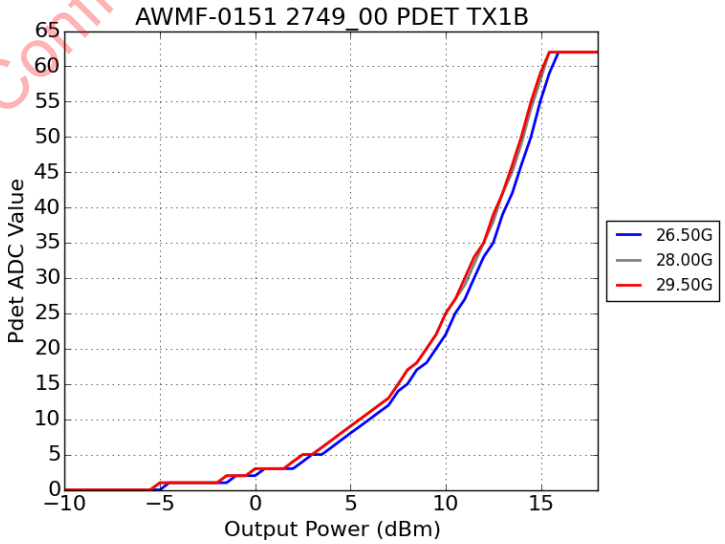

Figure 12: Example AWMF-0151 Power Detector Characteristics

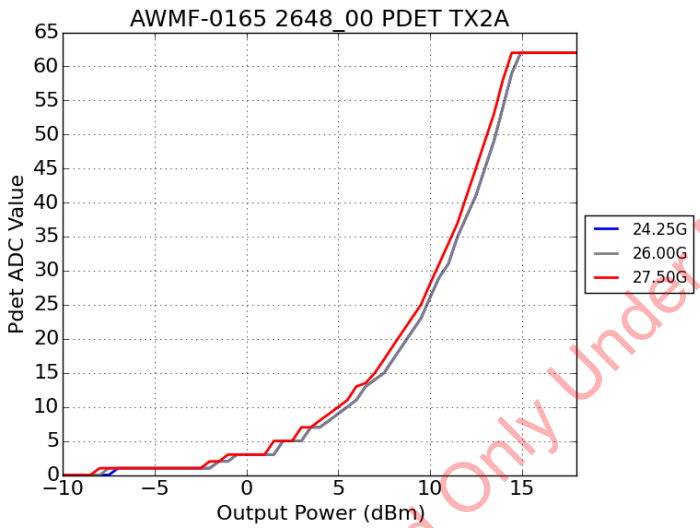

Figure 13: Example AWMF-0165 Power Detector Characteristics

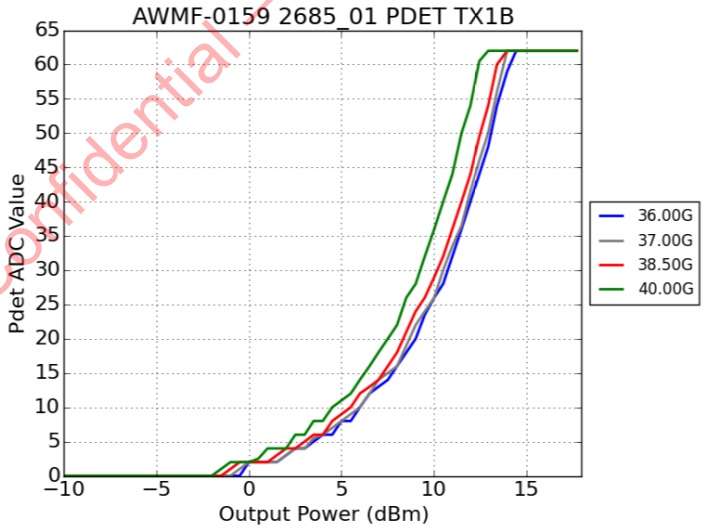

Figure 14: Example AWMF-0159 Power Detector Characteristics

5.2 Temperature Sensor

The temperature sensor is triggered to run at every falling edge of SPI_CSB. The previous temperature sensor data is latched at every falling edge of SPI_CSB for read back via the telemetry word. In order to obtain the most recent temperature data, first toggle SPI_CSB (low then high), then wait at least the minimum ADC conversion time, and then read back the telemetry data with a separate SPI transaction.

The temperature sensor output consists of a 6-bit digital word.

5.3 Telemetry Readback

Four different methods of telemetry readback are available on these devices for reporting the product identification, parity error, power detector outputs, and temperature sensor outputs. The telemetry data resides in registers 0x030 to 0x033 and details of these registers are included below in Table 13. The user can select (by setting RADDR) the desired telemetry that will be read back as part of every SPI transfer. See the register map in the appendix A1 for the detailed TELEM0-3 register contents.

Table 13: Contents of Telemetry Registers

Reg. Address	Reg Name	Contents
0x030	TELEM0	Product identification, 6-bit Quad-A power detector output, temperature sensor output, ZCal Applied
0x031	TELEM1	6-bit Quad-A power detector output, 6-bit Quad-B power detector output
0x032	TELEM2	Parity error, 5-bit Quad-A power detector output, 5-bit Quad-B power detector output, temperature sensor output
0x033	TELEM3	Product identification, 6-bit Quad-B power detector output, temperature sensor output

6. IC TIMING SPECIFICATION

6.1 SPI Timing

The SPI interface timing diagram is shown in Figure 15:

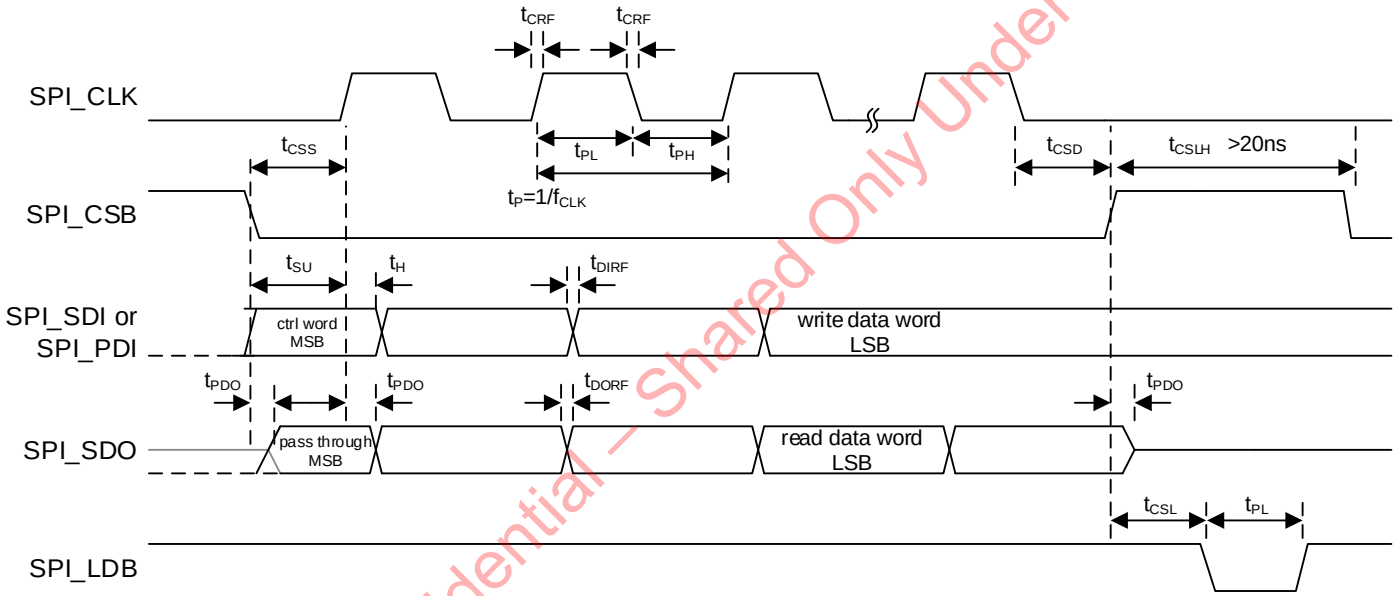

Figure 15: IC interface timing diagram

The timing constraints and electrical specifications for the SPI digital interface are listed in Table 14.

Note that the host should capture (SPI_SDI) the first bit (MSb) at the first SPI_CLK rising edge, and output (SDO) the first bit at the falling edge of SPI_CSB.

Table 14: SPI Interface Electrical and Timing Specifications

Parameter	Symbol	Units	Min	Typ	Max	Comment
Interface supply voltage	V_{IO}	V	1.62	1.8	1.98	VDD_{dig} 1P8 pin
SDO load capacitance	C_{SDO}	pF			20	
SDO load resistance	R_{SDO}	ohm	100k			Shunt
Output rise or fall time	$T_{RF,OUT}$	ns	1		5	SDO
Output high voltage	V_{OH}	V	V_{IO} - 0.45			SDO JESD8-7A Compliant
Output low voltage	V_{OL}	V			0.45	
Input rise or fall time	$T_{RF,IN}$	ns			5	All inputs
Input positive going threshold voltage	V_{IH}	V	0.4*V_{IO}		0.7*V_{IO}	JESD8-7A
Input negative going threshold voltage	V_{IL}	V	0.3*V_{IO}		0.6*V_{IO}	
Input hysteresis voltage	V_{HYS}	V	0.1*V_{IO}		0.4*V_{IO}	
Supply voltage rise time	TR_{VIO}	us	2			
SPI Clock (CLK) frequency	f_{CLK}	MHz			100	
Input setup time	t_{SU}	ns	3			For all inputs. This must be met at all chips in a daisy chain for the first bit in serial transactions
Input hold time	t_H	ns	1			All Inputs
CLK rise and fall time	t_{CRF}	ns			5	
CLK high duration	t_{PH}	ns	5			
CLK and LDB low duration	t_{PL}	ns	5			
CSB falling edge to CLK rising edge	t_{CSS}	ns	3			Select setup time. This must be met at all chips in a daisy chain for serial transactions
CLK falling edge to CSB rising edge	t_{CSD}	ns	3			Deselect time
CLK to SDO delay	t_{PDO}	ns	1.4		6	Clocked serial output delay
CSB to SDO release	t_{SDO_REL}	ns			6	Time for SDO to return to tri-state after CSB is de-asserted (high)
CSB rising edge to LDB falling edge	t_{CSL}	ns	3			
LDB/CSB to data latch delay	$t_{LDB,OUT}$	ns			5	
LDB/CSB to RF settling time	$t_{LDB,RF}$	ns			100	RF<10% from final value. Valid for beam-steering commands. Not valid for SPI commands that switch on/off active circuitry, resulting in bias current change.

6.2 TDD Switching

The switching time performance of the IC for different modes is summarized in Table 15.

As an example, measurement #1 shows the time delay for the part to transition from the Standby mode into the Rx mode. The switching time delay is measured/defined between the 50% threshold crossing of the rising RX_EN signal (RX_EN r50%) to the output RF signal reaching 90% of its level (RF r90%). Similarly, measurement #10 shows the time delay for the IC to switch between the Rx mode to Tx mode (i.e. TX_EN reaching the 50% threshold crossing level (TX_EN r50%) to the Tx path reaching 90% of its RF level (RF r90%). The other measurements follow a similar notation.

A 25ns guard period is recommended between TX_EN and RX_EN transitions to avoid both signals being high at the same time during switching (which would inadvertently result in the IC going into sleep mode).

Note that the time for the part to transition between the Sleep Mode and an operational condition is much longer than the time taken to transition from the Standby mode to the same condition.

Table 15: TDD Switching Performance

#	Start State	Finish State	Signal Level	Typ (ns)
1	Standby	Rx	RX_EN r50% -> RF r90%	112
2	Rx	Standby	RX_EN f50% -> RF f10%	7
3	Sleep	Rx	RX_EN r50% -> RF r90%	444
4	Rx	Sleep	RX_EN f50% -> RF f10%	5
5	Tx	Rx	RX_EN r50% -> RF r90%	110
6	Standby	Tx	TX_EN r50% -> RF r90%	179
7	Tx	Standby	TX_EN f50% -> RF f10%	8
8	Sleep	Tx	TX_EN r50% -> RF r90%	356
9	Tx	Sleep	TX_EN f50% -> RF f10%	8
10	Rx	Tx	TX_EN r50% -> RF r90%	180

7. Multi-Device Topology Arrangements

These ASICs can be accessed as a stand-alone device by the master, but the SPI is also designed to allow multiple devices to be arranged in various ways for large antenna arrays.

7.1 Daisy-Chain Topology

Multiple devices can be configured serially so that the master clocks the data into the SPI_SDI pin of the first device in the chain and the SPI_SDO pin of the device feeds the SPI_SDI pin of the next device in the chain successively until the SPI_SDO pin of the final device in the chain is read back by the master. The recommended schematic for a 64 element (4x4 BFIC) array is shown in Figure 16:. Four daisy chains are formed by the rows: IC0x, IC1x, IC2x, and IC3x. The serial command length will be 60x4 = 240 clocks. To steer in 240 clocks, we shift all 4 daisy chains at the same time. In general (ex. during steering) the data in each daisy chain is unique. However, the control signals: spi_clk, spi_csb, spi_ldb, and spi_pdo are identical for each daisy chain. To ensure good signal integrity (by reducing fanout/loading/length), these signals are duplicated for each daisy chain, rather than routing ex. spi_clk to all 16 BFICs. With this arrangement, SPI_CLK speeds of 50-100MHz are achievable. It may be possible to share control signals between multiple daisy chains to reduce pin count on FPGA_Controller0. The maximum clock speed of this approach is highly dependent on the PCB design.

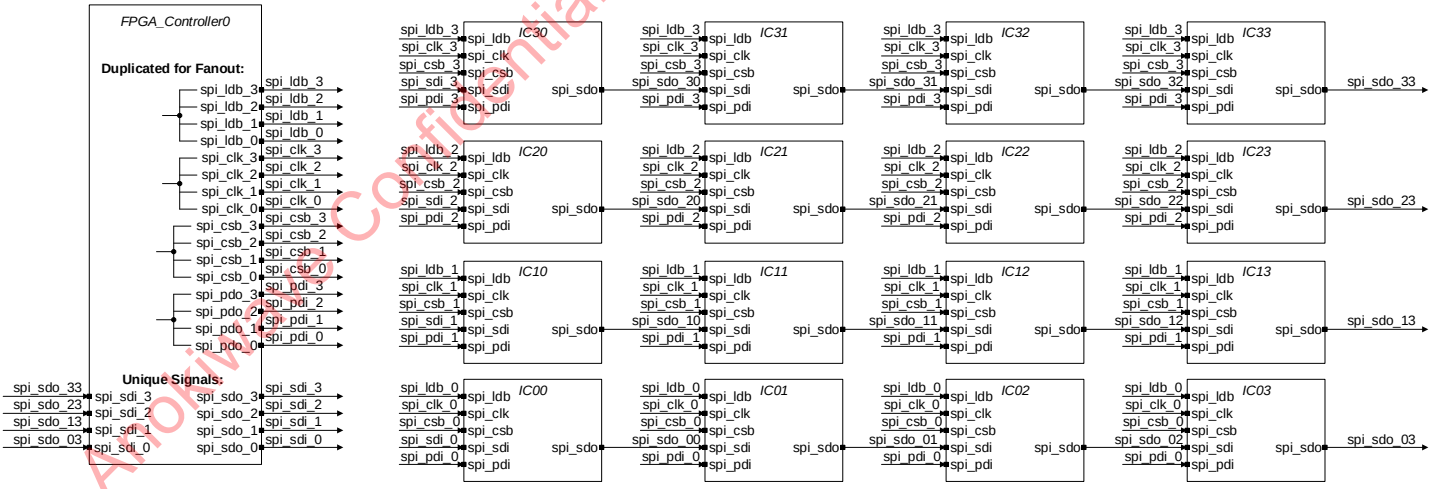

Figure 16: Example Daisy-Chain Arrangement (64 element array)

Microcontroller Notes:

1) It is also possible to control an array from a microcontroller SPI interface. An FPGA is ideal for the tight timing requirements on beam steering in infrastructure products. When controlling the array from a SPI microcontroller, the SDO pins can be connected together (ex spi_sdo_33, spi_sdo_23, spi_sdo_13, spi_sdo_03), as long as only a single daisy chain chip select is active at a time (ex. one of spi_csb_0, spi_csb_1, spi_csb_2, spi_csb_3).
2) The serial command can be zero padded in the MSbits to make the total command (for the daisy chain) a multiple of 8 bits if needed. For example, in a hypothetical daisy chain of length 3, take the 60*3=180bit command word and prepend 4ea. 0's in the MSbits.

APPENDICES

A1. Register Map

Table 16: Mode Register Contents

Reg. Addr.	Reg. Name	Type	MSB:LSB	Field Name	Default	Description
0x000	MODE	r/w	47:42	reserved	0	
0x000	MODE	r/w	41	reserved	0	Set to 0
0x000	MODE	r/w	40	parity_discard_en	0	1=enable parity discard. If parity_en==1, and there is a parity error, and parity discard is enabled, the write will not take effect. 0=report parity errors (if parity_en==1), but accept all write commands.
0x000	MODE	r/w	39	parity_en	0	1=enable parity checking and generation. If parity checking is enabled, the controller should set the parity bit (MSbit-1) such that the entire 60 bit serial message (including parity bit) has an odd number of 1's (odd parity). Parity checking is performed on serial messages. Outbound messages will have the 2nd bit (MSbit-1) modified by the BFIC to generate odd parity over the message.
0x000	MODE	r/w	38:35	rxip3_b[3:0]	0	RXIP3 linearity setting for all 4 Rx elements in quad B. Each bit independently controls gain.
0x000	MODE	r/w	34:31	rxip3_a[3:0]	0	RXIP3 linearity setting for all 4 Rx elements in quad A. Each bit independently controls gain.
0x000	MODE	r/w	30	atc_mode	0	1 = Automatic temperature compensation 0 = Host-controller-assisted temperature compensation
0x000	MODE	r/w	29:27	reserved	0	Set to 0
0x000	MODE	r/w	26:25	reserved	0	
0x000	MODE	r/w	24	sw_reset	0	Soft reset, write a 1, then write a 0, to this location to reset the SPI registers. The bit is not self-clearing. Note: issue a soft reset after applying power to IC. 0 = inactive 1 = reset internal registers
0x000	MODE	r/w	23	reserved	0	Set to 0
0x000	MODE	r/w	22	spi_ldb_en	0	Select whether the external LDB pin is used for latching registers 0 = LDB is not required and can be grounded or tied high 1 = LDB is required for latching write commands
0x000	MODE	r/w	21	reserved	0	Reserved set to 0
0x000	MODE	r/w	20:17	reserved	0	Reserved set to 0
0x000	MODE	r/w	16	reserved	1	Reserved set to 1
0x000	MODE	r/w	15:10	pdet_bw[5:0]	0	TX Power detector bandwidth. Setting to 0 gives highest bandwidth.
0x000	MODE	r/w	9	reserved	0	Reserved set to 0
0x000	MODE	r/w	8:5	reserved	0	Reserved set to 0
0x000	MODE	r/w	4	reserved	0	Reserved set to 0
0x000	MODE	r/w	3	zcal_init	0	0 = no action 1 = initiate the Zcal engine
0x000	MODE	r/w	2	spi_fbs_sel	0	FBS / Phase value select 0 = use phase values in BWx registers 1 = use phase values in Fast Beam Steering memory
0x000	MODE	r/w	1	spi_tdbs_sel	0	3-D Beam Steering gain value select 0 = use taper values in BWx registers 1 = use taper values in TDBS (3D Beam Steering) memory
0x000	MODE	r/w	0	reserved	0	Reserved set to 0

Table 17: BW_TX_A Register Contents

Reg. Addr.	Reg. Name	Type	MSB:LSB	Field Name	Default	Description
0x001	BW_TX_A	r/w	47	tx_rf4a_dis	0	1=disable Tx channel 4A
0x001	BW_TX_A	r/w	46	tx_rf3a_dis	0	1=disable Tx channel 3A
0x001	BW_TX_A	r/w	45	tx_rf2a_dis	0	1=disable Tx channel 2A
0x001	BW_TX_A	r/w	44	tx_rf1a_dis	0	1=disable Tx channel 1A
0x001	BW_TX_A	r/w	43:40	tx_coma_gain [3:0]	0	Tx A Common Arm Gain/Attenuation Attenuation = 0.5dB * tx_coma_gain
0x001	BW_TX_A	r/w	39:36	tx_rf4a_gain [3:0]	0	Tx channel 4A Gain/Attenuation Attenuation = 0.5dB * tx_rf4a_gain
0x001	BW_TX_A	r/w	35:32	tx_rf3a_gain [3:0]	0	Tx channel 3A Gain/Attenuation Attenuation = 0.5dB * tx_rf3a_gain
0x001	BW_TX_A	r/w	31:28	tx_rf2a_gain [3:0]	0	Tx channel 2A Gain/Attenuation Attenuation = 0.5dB * tx_rf2a_gain
0x001	BW_TX_A	r/w	27:24	tx_rf1a_gain [3:0]	0	Tx channel 1A Gain/Attenuation Attenuation = 0.5dB * tx_rf1a_gain
0x001	BW_TX_A	r/w	23:18	tx_rf4a_phase [5:0]	0	Tx channel 4A Phase-Shift Phase-shift = 5.625deg * tx_rf4a_phase
0x001	BW_TX_A	r/w	17:12	tx_rf3a_phase [5:0]	0	Tx channel 3A Phase-Shift Phase-shift = 5.625deg * tx_rf3a_phase
0x001	BW_TX_A	r/w	11:6	tx_rf2a_phase [5:0]	0	Tx channel 2A Phase-Shift Phase-shift = 5.625deg * tx_rf2a_phase
0x001	BW_TX_A	r/w	5:0	tx_rf1a_phase [5:0]	0	Tx channel 1A Phase-Shift Phase-shift = 5.625deg * tx_rf1a_phase

Table 18: BW_RX_A Register Contents

Reg. Addr.	Reg. Name	Type	MSB:LSB	Field Name	Default	Description
0x002	BW_RX_A	r/w	47	rx_rf4a_dis	0	1=disable Rx channel 4A
0x002	BW_RX_A	r/w	46	rx_rf3a_dis	0	1=disable Rx channel 3A
0x002	BW_RX_A	r/w	45	rx_rf2a_dis	0	1=disable Rx channel 2A
0x002	BW_RX_A	r/w	44	rx_rf1a_dis	0	1=disable Rx channel 1A
0x002	BW_RX_A	r/w	43:40	rx_coma_gain [3:0]	0	Rx A Common Arm Gain/Attenuation Attenuation = 0.5dB * rx_coma_gain
0x002	BW_RX_A	r/w	39:36	rx_rf4a_gain [3:0]	0	Rx channel 4A Gain/Attenuation Attenuation = 0.5dB * rx_rf4a_gain
0x002	BW_RX_A	r/w	35:32	rx_rf3a_gain [3:0]	0	Rx channel 3A Gain/Attenuation Attenuation = 0.5dB * rx_rf3a_gain
0x002	BW_RX_A	r/w	31:28	rx_rf2a_gain [3:0]	0	Rx channel 2A Gain/Attenuation Attenuation = 0.5dB * rx_rf2a_gain
0x002	BW_RX_A	r/w	27:24	rx_rf1a_gain [3:0]	0	Rx channel 1A Gain/Attenuation Attenuation = 0.5dB * rx_rf1a_gain
0x002	BW_RX_A	r/w	23:18	rx_rf4a_phase [5:0]	0	Rx channel 4A Phase-Shift Phase-shift = 5.625deg * rx_rf4a_phase
0x002	BW_RX_A	r/w	17:12	rx_rf3a_phase [5:0]	0	Rx channel 3A Phase-Shift Phase-shift = 5.625deg * rx_rf3a_phase
0x002	BW_RX_A	r/w	11:6	rx_rf2a_phase [5:0]	0	Rx channel 2A Phase-Shift Phase-shift = 5.625deg * rx_rf2a_phase
0x002	BW_RX_A	r/w	5:0	rx_rf1a_phase [5:0]	0	Rx channel 1A Phase-Shift Phase-shift = 5.625deg * rx_rf1a_phase

Table 19: RX_MEMADDR Register Contents

Reg. Addr.	Reg. Name	Type	MSB:LSB	Field Name	Default	Description
0x003	RX_MEMADDR	r	47:39	reserved	0	
0x003	RX_MEMADDR	r	38:33	rx_tdbs_addr_b [5:0]	0	Rx TDBS memory pointer for B pol.
0x003	RX_MEMADDR	r	32:24	rx_fbs_addr_b [8:0]	0	Rx FBS memory pointer for B pol.
0x003	RX_MEMADDR	r	23:15	reserved	0	
0x003	RX_MEMADDR	r	14:9	rx_tdbs_addr_a [5:0]	0	Rx TDBS memory pointer for A pol.
0x003	RX_MEMADDR	r	8:0	rx_fbs_addr_a [8:0]	0	Rx FBS memory pointer for A pol.

Table 20: TX_MEMADDR Register Contents

Reg. Addr.	Reg. Name	Type	MSB:LSB	Field Name	Default	Description
0x004	TX_MEMADDR	r	47:39	reserved	0	
0x004	TX_MEMADDR	r	38:33	tx_tdbs_addr_b [5:0]	0	Tx TDBS memory pointer for B pol.
0x004	TX_MEMADDR	r	32:24	tx_fbs_addr_b [8:0]	0	Tx FBS memory pointer for B pol.
0x004	TX_MEMADDR	r	23:15	reserved	0	
0x004	TX_MEMADDR	r	14:9	tx_tdbs_addr_a [5:0]	0	Tx TDBS memory pointer for A pol.
0x004	TX_MEMADDR	r	8:0	tx_fbs_addr_a [8:0]	0	Tx FBS memory pointer for A pol.

Table 21: ATTTC Register Contents

Reg. Addr.	Reg. Name	Type	MSB:LSB	Field Name	Default	Description
0x005	ATTTC	r/w	47:20	reserved	0	
0x005	ATTTC	r/w	19:15	spi_tc_txc [4:0]	4	Temp Comp setting for Tx common arm
0x005	ATTTC	r/w	14:10	spi_tc_txq [4:0]	8	Temp Comp setting for all Tx element arms
0x005	ATTTC	r/w	9:5	spi_tc_rxc [4:0]	4	Temp Comp setting for Rx common arm
0x005	ATTTC	r/w	4:0	spi_tc_rxq [4:0]	8	Temp Comp setting for all Rx element arms

Table 22: ENGR0 Register Contents

Reg. Addr.	Reg. Name	Type	MSB:LSB	Field Name	Default	Description
0x00D	ENGR0	r/w	47:24	reserved	-	Use value from initialization sequence.
0x00D	ENGR0	r/w	23	atc_bypass	1	1=Ignore ATC logic and use legacy open-loop temp comp. 0=atc_mode bit in MODE register determines ATC operation.
0x00D	ENGR0	r/w	22	atc_disable	0	Disable ATC circuitry.
0x00D	ENGR0	r/w	21	atc_reset	1	1=synchronous reset of the ATC circuitry, not self-clearing. Set this bit to 1, then 0 to reset the ATC state machine. Always set atc_reset=1 before changing ATC_CONFIG registers, then set to 0 to avoid unpredictable behavior.
0x00D	ENGR0	r/w	20:19	reserved	1	Use value from initialization sequence.
0x00D	ENGR0	r/w	18	dig_osc_en	0	1=enable oscillator needed for ATC algorithm.
0x00D	ENGR0	r/w	17	tx_comb_dis	0	1=disable circuitry in Tx B common arm.
0x00D	ENGR0	r/w	16	rx_comb_dis	0	1=disable circuitry in Rx B common arm.
0x00D	ENGR0	r/w	15	tx_coma_dis	0	1=disable circuitry in Tx A common arm.
0x00D	ENGR0	r/w	14	rx_coma_dis	0	1=disable circuitry in Rx A common arm.
0x00D	ENGR0	r/w	13	reserved	0	Use value from initialization sequence.
0x00D	ENGR0	r/w	12	spi_rx_en	0	Rx is enabled if either the RX_EN pin or spi_tx_en are set high.
0x00D	ENGR0	r/w	11	spi_tx_en	0	Tx is enabled if either the TX_EN pin or spi_tx_en are set high.
0x00D	ENGR0	r/w	10:0	reserved	0	Use value from initialization sequence.

Table 23: ATC_CALC Register Contents

Reg. Addr.	Reg. Name	Type	MSB:LSB	Field Name	Default	Description
0x016	ATC_CALC	r	47:41	reserved	0	
0x016	ATC_CALC	r	40:35	tsensor_last [5:0]	0	Most recent output of temp sensor ADC.
0x016	ATC_CALC	r	34:29	tsensor_avg [5:0]	0	Averaged output of temp sensor ADC.
0x016	ATC_CALC	r	28:20	temp_deg [8:0]	0	IC temperature in degrees Celsius (based on tsensor_avg). Signed 2's complement format.
0x016	ATC_CALC	r	19:15	atc_txc [4:0]	0	Temp Comp setting for Tx common arm, calculated by ATC.
0x016	ATC_CALC	r	14:10	atc_txq [4:0]	0	Temp Comp setting for all Tx element arms, calculated by ATC.
0x016	ATC_CALC	r	9:5	atc_rxc [4:0]	0	Temp Comp setting for Rx common arm, calculated by ATC.
0x016	ATC_CALC	r	4:0	atc_rxq [4:0]	0	Temp Comp setting for all Rx element arms, calculated by ATC.

Table 24: ATC_CONFIG3 Register Contents

Reg. Addr.	Reg. Name	Type	MSB:LSB	Field Name	Default	Description
0x01F	ATC_CONFIG3	r/w	47:22	Reserved	0	
0x01F	ATC_CONFIG3	r/w	21:2	atc_meas_period [19:0]	1000	The rate at which ATC state machine samples the temp sensor ADC output, specified as period in milliseconds. Note that the on-chip clock is not very precise.
0x01F	ATC_CONFIG3	r/w	1:0	atc_meas_trig [1:0]	0	Trigger source that triggers ATC state machine. 0 = No trigger. ATC state machine measures continuously. 1 = ATC state machine starts at CSB falling edge. 2 = ATC state machine starts at TX_EN rising edge. 3 = ATC state machine starts at RX_EN rising edge.

Table 25: ATC_CONFIG5 Register Contents

Reg. Addr.	Reg. Name	Type	MSB:LSB	Field Name	Default	Description
0x021	ATC_CONFIG5	r/w	47:29	Reserved	0	
0x021	ATC_CONFIG5	r/w	28:9	Reserved	59	Should be set to the default value.
0x021	ATC_CONFIG5	r/w	8:7	atc_update_trig_tx [1:0]	2	Event that triggers Tx gain update by ATC state machine. 0 = Update immediately after ATC calculation is done. 1 = Update at the end of any SPI transaction. 2 = Update on falling edge of TX_EN (i.e., when exiting Tx mode). 3 = Update at the end of a beamweight or FBS SPI transaction.
0x021	ATC_CONFIG5	r/w	6:5	atc_update_trig_rx [1:0]	2	Event that triggers Rx gain update by ATC state machine. 0 = Update immediately after ATC calculation is done. 1 = Update at the end of any SPI transaction. 2 = Update on falling edge of RX_EN (i.e., when exiting Rx mode). 3 = Update at the end of a beamweight or FBS SPI transaction.
0x021	ATC_CONFIG5	r/w	4:0	Reserved	31	

Table 26: BW_TX_B Register Contents

Reg. Addr.	Reg. Name	Type	MSB:LSB	Field Name	Default	Description
0x022	BW_TX_B	r/w	47	tx_rf4b_dis	0	1=disable Tx channel 4B
0x022	BW_TX_B	r/w	46	tx_rf3b_dis	0	1=disable Tx channel 3B
0x022	BW_TX_B	r/w	45	tx_rf2b_dis	0	1=disable Tx channel 2B
0x022	BW_TX_B	r/w	44	tx_rf1b_dis	0	1=disable Tx channel 1B
0x022	BW_TX_B	r/w	43:40	tx_comb_gain [3:0]	0	Tx B Common Arm Gain/Attenuation Attenuation = 0.5dB * tx_comb_gain
0x022	BW_TX_B	r/w	39:36	tx_rf4b_gain [3:0]	0	Tx channel 4B Gain/Attenuation Attenuation = 0.5dB * tx_rf4b_gain
0x022	BW_TX_B	r/w	35:32	tx_rf3b_gain [3:0]	0	Tx channel 3B Gain/Attenuation Attenuation = 0.5dB * tx_rf3b_gain
0x022	BW_TX_B	r/w	31:28	tx_rf2b_gain [3:0]	0	Tx channel 2B Gain/Attenuation Attenuation = 0.5dB * tx_rf2b_gain
0x022	BW_TX_B	r/w	27:24	tx_rf1b_gain [3:0]	0	Tx channel 1B Gain/Attenuation Attenuation = 0.5dB * tx_rf1b_gain
0x022	BW_TX_B	r/w	23:18	tx_rf4b_phase [5:0]	0	Tx channel 4B Phase-Shift Phase-shift = 5.625deg * tx_rf4b_phase
0x022	BW_TX_B	r/w	17:12	tx_rf3b_phase [5:0]	0	Tx channel 3B Phase-Shift Phase-shift = 5.625deg * tx_rf3b_phase
0x022	BW_TX_B	r/w	11:6	tx_rf2b_phase [5:0]	0	Tx channel 2B Phase-Shift Phase-shift = 5.625deg * tx_rf2b_phase
0x022	BW_TX_B	r/w	5:0	tx_rf1b_phase [5:0]	0	Tx channel 1B Phase-Shift Phase-shift = 5.625deg * tx_rf1b_phase

Table 27: BW_RX_B Register Contents

Reg. Addr.	Reg. Name	Type	MSB:LSB	Field Name	Default	Description
0x023	BW_RX_B	r/w	47	rx_rf4b_dis	0	1=disable Rx channel 4B
0x023	BW_RX_B	r/w	46	rx_rf3b_dis	0	1=disable Rx channel 3B
0x023	BW_RX_B	r/w	45	rx_rf2b_dis	0	1=disable Rx channel 2B
0x023	BW_RX_B	r/w	44	rx_rf1b_dis	0	1=disable Rx channel 1B
0x023	BW_RX_B	r/w	43:40	rx_comb_gain [3:0]	0	Rx B Common Arm Gain/Attenuation Attenuation = 0.5dB * rx_comb_gain
0x023	BW_RX_B	r/w	39:36	rx_rf4b_gain [3:0]	0	Rx channel 4B Gain/Attenuation Attenuation = 0.5dB * rx_rf4b_gain
0x023	BW_RX_B	r/w	35:32	rx_rf3b_gain [3:0]	0	Rx channel 3B Gain/Attenuation Attenuation = 0.5dB * rx_rf3b_gain
0x023	BW_RX_B	r/w	31:28	rx_rf2b_gain [3:0]	0	Rx channel 2B Gain/Attenuation Attenuation = 0.5dB * rx_rf2b_gain
0x023	BW_RX_B	r/w	27:24	rx_rf1b_gain [3:0]	0	Rx channel 1B Gain/Attenuation Attenuation = 0.5dB * rx_rf1b_gain
0x023	BW_RX_B	r/w	23:18	rx_rf4b_phase [5:0]	0	Rx channel 4B Phase-Shift Phase-shift = 5.625deg * rx_rf4b_phase
0x023	BW_RX_B	r/w	17:12	rx_rf3b_phase [5:0]	0	Rx channel 3B Phase-Shift Phase-shift = 5.625deg * rx_rf3b_phase
0x023	BW_RX_B	r/w	11:6	rx_rf2b_phase [5:0]	0	Rx channel 2B Phase-Shift Phase-shift = 5.625deg * rx_rf2b_phase
0x023	BW_RX_B	r/w	5:0	rx_rf1b_phase [5:0]	0	Rx channel 1B Phase-Shift Phase-shift = 5.625deg * rx_rf1b_phase

Table 28: Telemetry Registers Contents

Reg. Addr.	Reg. Name	Type	MSB:LSB	Field Name	Default	Description
0x030	TELEM0	r	47:39	product_id [8:0]	-	Product ID code from PROD_ID register
0x030	TELEM0	r	38:31	product_version[7:0]	-	Product version code from PROD_ID register
0x030	TELEM0	r	30	zcal_applied	0	0=internal calibration factors not applied 1=internal calibration factors applied
0x030	TELEM0	r	29:24	tx4a_pdet [5:0]	0	Power Detector Output from Channel 4A
0x030	TELEM0	r	23:18	tx3a_pdet [5:0]	0	Power Detector Output from Channel 3A
0x030	TELEM0	r	17:12	tx2a_pdet [5:0]	0	Power Detector Output from Channel 2A
0x030	TELEM0	r	11:6	tx1a_pdet [5:0]	0	Power Detector Output from Channel 1A
0x030	TELEM0	r	5:0	temp_sensor [5:0]	0	Temperature Sensor Output
0x031	TELEM1	r	47:42	tx4b_pdet [5:0]	0	Power Detector Output from Channel 4B
0x031	TELEM1	r	41:36	tx3b_pdet [5:0]	0	Power Detector Output from Channel 3B
0x031	TELEM1	r	35:30	tx2b_pdet [5:0]	0	Power Detector Output from Channel 2B
0x031	TELEM1	r	29:24	tx1b_pdet [5:0]	0	Power Detector Output from Channel 1B
0x031	TELEM1	r	23:18	tx4a_pdet [5:0]	0	Power Detector Output from Channel 4A
0x031	TELEM1	r	17:12	tx3a_pdet [5:0]	0	Power Detector Output from Channel 3A
0x031	TELEM1	r	11:6	tx2a_pdet [5:0]	0	Power Detector Output from Channel 2A
0x031	TELEM1	r	5:0	tx1a_pdet [5:0]	0	Power Detector Output from Channel 1A
0x032	TELEM2	r	47	parity_error	0	0=no parity error on previous received serial message. 1=parity error. Errors are only reported if parity_en==1.
0x032	TELEM2	r	46	reserved	0	
0x032	TELEM2	r	45:41	tx4b_pdet [5:1]	0	Power Detector Output from Channel 4B
0x032	TELEM2	r	40:36	tx3b_pdet [5:1]	0	Power Detector Output from Channel 3B
0x032	TELEM2	r	35:31	tx2b_pdet [5:1]	0	Power Detector Output from Channel 2B
0x032	TELEM2	r	30:26	tx1b_pdet [5:1]	0	Power Detector Output from Channel 1B
0x032	TELEM2	r	25:21	tx4a_pdet [5:1]	0	Power Detector Output from Channel 4A
0x032	TELEM2	r	20:16	tx3a_pdet [5:1]	0	Power Detector Output from Channel 3A
0x032	TELEM2	r	15:11	tx2a_pdet [5:1]	0	Power Detector Output from Channel 2A
0x032	TELEM2	r	10:6	tx1a_pdet [5:1]	0	Power Detector Output from Channel 1A
0x032	TELEM2	r	5:0	temp_sensor [5:0]	0	Temperature Sensor Output
0x033	TELEM3	r	47:39	product_id [8:0]	-	Product ID code from PROD_ID register
0x033	TELEM3	r	38:31	product_version[7:0]	-	Product version code from PROD_ID register
0x033	TELEM3	r	30	reserved	0	
0x033	TELEM3	r	29:24	tx4b_pdet [5:0]	0	Power Detector Output from Channel 4B
0x033	TELEM3	r	23:18	tx3b_pdet [5:0]	0	Power Detector Output from Channel 3B
0x033	TELEM3	r	17:12	tx2b_pdet [5:0]	0	Power Detector Output from Channel 2B
0x033	TELEM3	r	11:6	tx1b_pdet [5:0]	0	Power Detector Output from Channel 1B
0x033	TELEM3	r	5:0	temp_sensor [5:0]	0	Temperature Sensor Output

Table 29: SPARE Register Contents

Reg. Addr.	Reg. Name	Type	MSB:LSB	Field Name	Default	Description
0x03A	SPARE	r/w	47:0	spare [47:0]	0	Writeable register for testing, or to write dummy data to get readback data

Table 30: RADDR Register Contents

Reg. Addr.	Reg. Name	Type	MSB:LSB	Field Name	Default	Description
0x03D	RADDR	r/w	47:10	reserved	0	
0x03D	RADDR	r/w	9:0	raddr [9:0]	0x30	Serial Read Address During serial write/read transactions, contents of the register pointed to by raddr are shifted out on SPI_SDO. The default read address is TELEM0 (default Telemetry).

Table 31: PROD_ID Register Contents

Reg. Addr.	Reg. Name	Type	MSB:LSB	Field Name	Default	Description
0x03E	PROD_ID	r	47:17	reserved	0	
0x03E	PROD_ID	r	16:8	product_id [8:0]	0x151 / 0x159 / 0x165	Product part number.
0x03E	PROD_ID	r	7:0	product_version[7:0]	0xA0/A1/A2/B0	Product version (respin code).

Table 32: RX_TDBS Register Contents

Reg. Addr.	Reg. Name	Type	MSB:LSB	Field Name
0x100	RX_TDBS row 0	r/w	47	RX_TDBS1 / rx_rf4_dis
0x100	RX_TDBS row 0	r/w	46	RX_TDBS1 / rx_rf3_dis
0x100	RX_TDBS row 0	r/w	45	RX_TDBS1 / rx_rf2_dis
0x100	RX_TDBS row 0	r/w	44	RX_TDBS1 / rx_rf1_dis
0x100	RX_TDBS row 0	r/w	43:40	RX_TDBS1 / rx_com_gain [3:0]
0x100	RX_TDBS row 0	r/w	39:36	RX_TDBS1 / rx_rf4_gain [3:0]
0x100	RX_TDBS row 0	r/w	35:32	RX_TDBS1 / rx_rf3_gain [3:0]
0x100	RX_TDBS row 0	r/w	31:28	RX_TDBS1 / rx_rf2_gain [3:0]
0x100	RX_TDBS row 0	r/w	27:24	RX_TDBS1 / rx_rf1_gain [3:0]
0x100	RX_TDBS row 0	r/w	23	RX_TDBS0 / rx_rf4_dis
0x100	RX_TDBS row 0	r/w	22	RX_TDBS0 / rx_rf3_dis
0x100	RX_TDBS row 0	r/w	21	RX_TDBS0 / rx_rf2_dis
0x100	RX_TDBS row 0	r/w	20	RX_TDBS0 / rx_rf1_dis
0x100	RX_TDBS row 0	r/w	19:16	RX_TDBS0 / rx_com_gain [3:0]
0x100	RX_TDBS row 0	r/w	15:12	RX_TDBS0 / rx_rf4_gain [3:0]
0x100	RX_TDBS row 0	r/w	11:8	RX_TDBS0 / rx_rf3_gain [3:0]
0x100	RX_TDBS row 0	r/w	7:4	RX_TDBS0 / rx_rf2_gain [3:0]
0x100	RX_TDBS row 0	r/w	3:0	RX_TDBS0 / rx_rf1_gain [3:0]

Table 33: TX_TDBS Register Contents

Reg. Addr.	Reg. Name	Type	MSB:LSB	Field Name
0x140	TX_TDBS row 0	r/w	47	TX_TDBS1 / tx_rf4_dis
0x140	TX_TDBS row 0	r/w	46	TX_TDBS1 / tx_rf3_dis
0x140	TX_TDBS row 0	r/w	45	TX_TDBS1 / tx_rf2_dis
0x140	TX_TDBS row 0	r/w	44	TX_TDBS1 / tx_rf1_dis
0x140	TX_TDBS row 0	r/w	43:40	TX_TDBS1 / tx_com_gain [3:0]
0x140	TX_TDBS row 0	r/w	39:36	TX_TDBS1 / tx_rf4_gain [3:0]
0x140	TX_TDBS row 0	r/w	35:32	TX_TDBS1 / tx_rf3_gain [3:0]
0x140	TX_TDBS row 0	r/w	31:28	TX_TDBS1 / tx_rf2_gain [3:0]
0x140	TX_TDBS row 0	r/w	27:24	TX_TDBS1 / tx_rf1_gain [3:0]
0x140	TX_TDBS row 0	r/w	23	TX_TDBS0 / tx_rf4_dis
0x140	TX_TDBS row 0	r/w	22	TX_TDBS0 / tx_rf3_dis
0x140	TX_TDBS row 0	r/w	21	TX_TDBS0 / tx_rf2_dis
0x140	TX_TDBS row 0	r/w	20	TX_TDBS0 / tx_rf1_dis
0x140	TX_TDBS row 0	r/w	19:16	TX_TDBS0 / tx_com_gain [3:0]
0x140	TX_TDBS row 0	r/w	15:12	TX_TDBS0 / tx_rf4_gain [3:0]
0x140	TX_TDBS row 0	r/w	11:8	TX_TDBS0 / tx_rf3_gain [3:0]
0x140	TX_TDBS row 0	r/w	7:4	TX_TDBS0 / tx_rf2_gain [3:0]
0x140	TX_TDBS row 0	r/w	3:0	TX_TDBS0 / tx_rf1_gain [3:0]

Table 34: FBS Register Contents

Reg. Addr.	Reg. Name	Type	MSB:LSB	Field Name
0x200	FBS row 0	r/w	47:42	FBS1 / rf4_phase [5:0]
0x200	FBS row 0	r/w	41:36	FBS1 / rf3_phase [5:0]
0x200	FBS row 0	r/w	35:30	FBS1 / rf2_phase [5:0]
0x200	FBS row 0	r/w	29:24	FBS1 / rf1_phase [5:0]
0x200	FBS row 0	r/w	23:18	FBS0 / rf4_phase [5:0]
0x200	FBS row 0	r/w	17:12	FBS0 / rf3_phase [5:0]
0x200	FBS row 0	r/w	11:6	FBS0 / rf2_phase [5:0]
0x200	FBS row 0	r/w	5:0	FBS0 / rf1_phase [5:0]

A2. Hybrid Mode

An additional SPI **Hybrid Mode** transaction is supported for writing or reading the same register in a chain of ICs, allowing a slight (~15%) increase in throughput by using a single control word, sent in parallel, followed by unique data for each IC.

The Hybrid Write/Read transaction uses a parallel mode to shift a single control word into each IC, followed by serial mode shift of the register data through the ICs. The timing diagram, in Figure 17 illustrates the transaction for two ICs (IC0 followed by IC1).

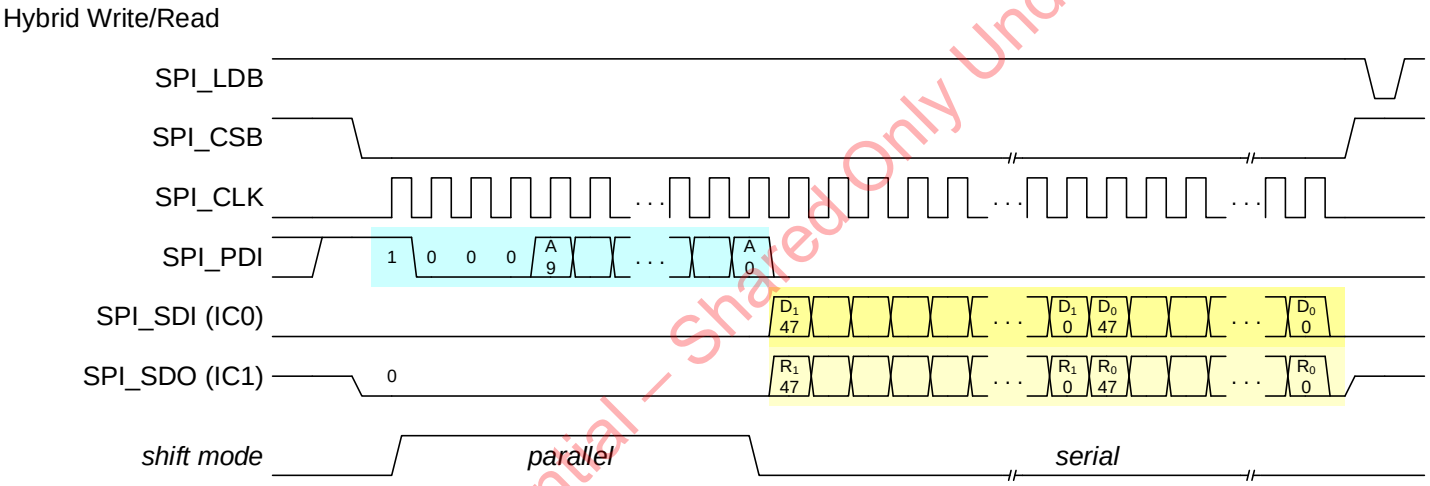

Figure 17: Timing diagram in Hybrid Write/Read mode for two daisy-chained ICs

REVISION RECORD

Revision	Date	Description of Change
1.0	05/09/2019	Initial Release
2.0	07/30/2019	Added Table 14, updated some text descriptions for clarity.
2.3	08/19/2019	Added AWMF-0159 PDET vs. ADC curves
2.5	09/24/2019	Updated AWMF-0151 and AWMF-0165 TCOMP tables and initialization sequences.
2.6	09/24/2019	Updated AWMF – 0159 TCOMP table and initialization sequence
2.7	09/26/2019	Updated the pinout diagram
3.0	12/16/2019	Updated descriptions in register maps and cleaned up typos.
4.0	02/04/2020	Added details of automatic temperature compensation feature.
5.0	02/28/2020	Added details of registers 0x03,0x04 in register map.
5.2	03/19/2020	Added details of registers 0x0D in register map. Updated description of FBS/TDBS register formats in Section 4.3 & 4.4. Added TDD switching time performance to Section 6. Updated SPI command format description and improved ATC operation description. Added LDB to RF settling time.
6.0	05/20/2020	Clean up some descriptions and update register map.
7.0	06/12/2020	Update all Initialization and TCOMP tables based on ETRR for all 3 products.
8.0	07/31/2020	Added details on ZCal feature and updated register map accordingly.