AM9017 Interface API Rev 1.02

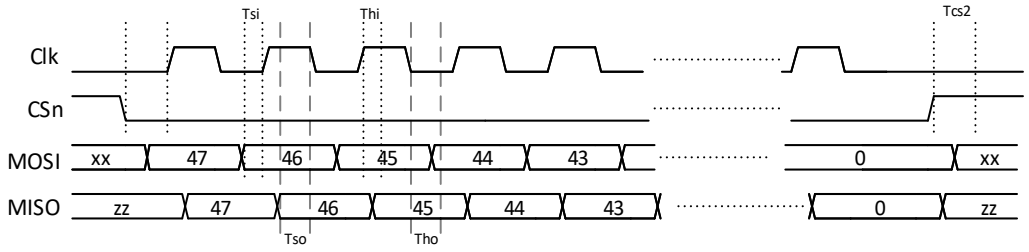
Contents

1. Description

The AM9017 is a miniature 0.1 to 18-GHz tuner module that contains a MachXO3 6900E controller. The MachXO3 6900E control interface consists of five lines: Clock, CMD_CSn, PROG_CSn, MOSI (Master Out Slave In) and MISO (Master In Slave Out). The system controlling the tuner is always the Master device in this configuration and the tuner is the Slave device. The SPI interface uses 48 bit commands and data is transmitted and received by the master device (the system controller) during each transaction. SPI sequences of less than 48 bits can be used in order to read back time sensitive status indicators (tuner busy bit, tuning LO lock status). Commands will only be correctly translated if 48 bits are transmitted.

2. Interface Specifications

A typical SPI transaction is shown below. The new command goes out on the MOSI line with data being read in by the slave device on each positive edge. The MISO line contains the desired Tuner output (different outputs can be requested using the Read_Mask command) and its data will be valid on every negative edge of the clock. To initiate a transaction the CSn line is pulled low and after a time of T_{cs} the first positive edge of the clock can be transmitted (MOSI data should already be stable). In order for commands to be correctly received the timing constraints in the table below must be observed. These constraints are referenced to the signals as they arrive at the Tuner I/O pins. The CSn pin used for Tuner Control is on FPGA pin CMD_CSn.



Parameter	Description	Min	Max
SPI Clock Frequency	Maximum frequency of SPI Clock		20 MHz
Tcs	Delay between CSn going low and first positive clock edge	16ns	
Tsi	Time data on MOSI must be stable preceding the sampling SPI clock positive edge	7ns	
Thi	Time data on MOSI must remain stable following the sampling SPI clock positive edge	2ns	

Tso	Time until data on MISO will be stable after negative SPI clock edge	9ns	
Tho	Time data on MISO will be stable following the sampling SPI clock positive edge	8ns	
Tcs2	Time CSn must be high between SPI transactions	65ns	

3. Tuner Operation

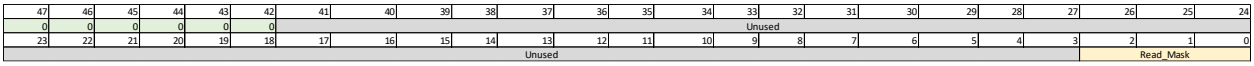
The AM9017 is primarily controlled by the Tuner_Setup, Set_Freq, and Set_Atten commands. The Set_Config and Manual_Set_Band commands are used to overwrite the calibration settings stored in the FPGA. The Set_Config command controls the states of amplifiers and the power settings for the different sections of the tuner. The Manual_Set_Band command controls the filter band selected as well as the tunable filter settings for that band.

After power is applied to the tuner, the Power On Pin is toggled, or the FPGA is reset, a Tuner_Setup message must be sent before the FPGA will respond to any of the other tuner control commands. A Tuner_Read can be performed before the Tuner_Setup is sent.

Each Tuner command has the same structure and uses the SPI bus selected by bringing CMD_CSn low. The most significant 6 bits contain the command code and the following 42 bits contain the command parameters.

The SPI bus selected by bringing PROG_CSn low is used to control the FPGA itself. Detailed command descriptions for the FPGA are in section 5. A useful command that uses this interface is the Refresh command (5.1.11). It forces the FPGA to reinitialize. This command can be used to solve issues encountered during initial power up.

3.1. Tuner_Read (Command Code 000000)



Parameter	Bit(s)	Description	Possible Values
Read_Mask	2:0	Determines the content of the Tuner MISO stream during SPI transactions	000: Tuner Status 001: Serial Number and HW Rev 010: FPGA Revision 011: PLL Read

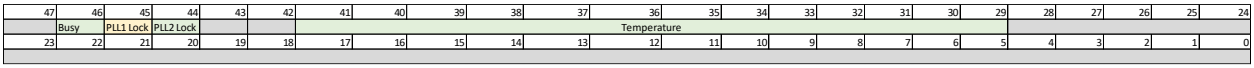
The Tuner_Read command is primarily used to check the Tuner status message but it can also be used to read the Tuner's serial number, hardware revision, and FPGA revision by changing the Read Mask. Two Tuner_Read commands need to be sent when the Read Mask is changed in order to actually have the MISO data reflect the changed mask. It is important to reset the mask to 0x00 in the second Tuner_Read command so the tuner will resume sending out tuner status on subsequent Tuner_Reads. An example transaction to read the FPGA revision is below.

First Command (set mask to read FPGA rev): 0x000000000002
Second Command (read FPGA rev and reset mask to Status): 0x000000000000

After sending the second command all subsequent SPI transactions will have Tuner Status on the MISO stream.

The Tuner defaults to a Read_Mask of 001 on startup. The first SPI transaction will have the Serial Number and HW Rev on the MISO stream. If a Tuner_Setup command is used the Read_Mask will be set to 000.

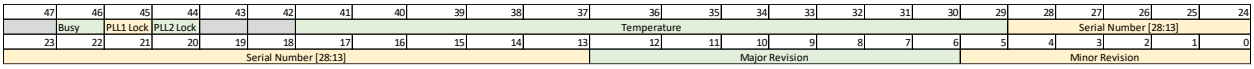
3.1.1. MISO Format – Tuner Status (Read_Mask 000)



Parameter	Bit(s)	Description	Possible Values
Unused Bit	47	Spare bit	N/A
Busy Bit	46	Indicates whether the Tuner is ready to receive a new command. Commands received while this bit is high will be ignored.	0: Tuner is ready to receive commands 1: Tuner is busy, new commands will be ignored
PLL1Lock	45	Indicates whether the tuning LO is locked	0: Tuning LO1 is unlocked 1: Tuning LO is locked
PLL2 Lock	44	Indicates whether the fixed LO is locked	0: Fixed LO2 is unlocked 1: Fixed LO2 is locked
Unused	43:42	Spare bit	N/A
Temperature	41:29	Temperature reading from the tuner, requires TC77 temperature sensor	0-2047: multiply by 0.0625 to get the temperature in degrees C 2048 – 4097: 2's complement format for negative temperatures. Convert with the formula: $-(Value - 2^{13}) * 0.0625$
Unused	28:0	Spare bits	N/A

When Read_Mask is set to 000 the MISO stream will contain Tuner Status information. Read_Mask will be set to 000 whenever a Tuner_Setup command is sent in addition to being controlled by the Tuner_Read command.

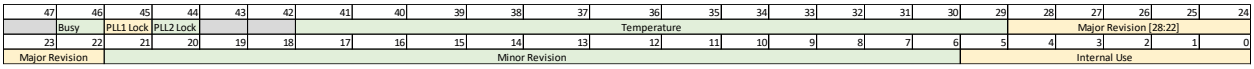
3.1.2. MISO Format – Tuner Serial Number and HW Revision (Read_Mask 001)



Parameter	Bit(s)	Description	Possible Values
Unused	47	Spare Bit	N/A
Busy Bit	46	Indicates whether the Tuner is ready to receive a new command. Commands received while this bit is high will be ignored.	0: Tuner is ready to receive commands 1: Tuner is busy, new commands will be ignored
PLL1Lock	45	Indicates whether the tuning LO is locked	0: Tuning LO1 is unlocked 1: Tuning LO is locked
PLL2 Lock	44	Indicates whether the fixed LO is locked	0: Fixed LO2 is unlocked 1: Fixed LO2 is locked
Unused	43:42	Spare bit	N/A
Temperature	41:29	Temperature reading from the tuner, requires TC77 temperature sensor	0-2047: multiply by 0.0625 to get the temperature in degrees C 2048 – 4097: 2's complement format for negative temperatures. Convert with the formula: $-(Value - 2^{13}) * 0.0625$
Serial Number	28:13	Serial Number for the AM9017	0-65535
Major Revision	12:6	Major HW Revision of the AM9017	0-127
Minor Revision	5:0	Minor HW Revision of the AM9017	0-127

When Read_Mask is set to 001 the MISO stream will contain Tuner serial number and hardware revision information. The busy bit is also contained in this message (it is always bit 46 regardless of Read_Mask).

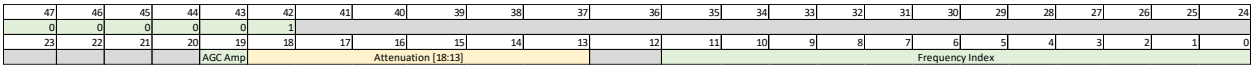
3.1.3. MISO Format – FPGA Revision (Read_Mask 010)



Parameter	Bit(s)	Description	Possible Values
Unused	47		
Busy Bit	46	Indicates whether the Tuner is ready to receive a new command. Commands received while this bit is high will be ignored.	0: Tuner is ready to receive commands 1: Tuner is busy, new commands will be ignored
PLL1Lock	45	Indicates whether the tuning LO is locked	0: Tuning LO1 is unlocked 1: Tuning LO is locked
PLL2 Lock	44	Indicates whether the fixed LO is locked	0: Fixed LO2 is unlocked 1: Fixed LO2 is locked
Unused	43:42	Spare bit	N/A
Temperature	41:29	Temperature reading from the tuner, requires TC77 temperature sensor	0-2047: multiply by 0.0625 to get the temperature in degrees C 2048 – 4097: 2's complement format for negative temperatures. Convert with the formula: $-(Value - 2^{13}) * 0.0625$
Major Revision	28:22	Major FPGA HW Revision	0-127
Minor Revision	21:6	Major HW Revision of the AM9017	0-65535
Internal Use	5:0		N/A

When Read_Mask is set to 010 the MISO stream will contain Tuner FPGA revision information. The busy bit is also contained in this message (it is always bit 46 regardless of Read_Mask).

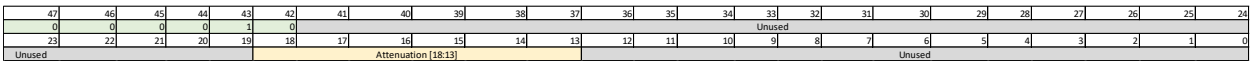
3.2. Tuner_Setup (Command Code 000001)



Parameter	Bit(s)	Description	Possible Values
Unused	41:20	Unused	
AGC Amp	19	Controls the front end amplifier in the RF path	0: Amp bypassed 1: Amp engaged
Attenuation	18:13	Sets the attenuation to apply in addition to the calibrated 0dB state	0-38 dB in 1 dB steps
Frequency Index	11:0	Index for the desired center frequency. Calculated using the formula: $(CF - 350)/5$ where CF is the desired center frequency in MHz in 5 MHz steps.	0 – 3480: 0 is for 350 MHz center, 3480 is for 17750 MHz center

The Tuner_Setup command is the primary interface command for the Tuner. When this command is used the FPGA will automatically calculate the values for all of the configurable settings on the tuner not explicitly controlled in the command. Examples of automatically calculated values are preselector settings, calibrated attenuator settings, PLL settings, frequency dependent amplifier power, etc. This command is all that is needed to tune from one frequency to another. The first command sent to the tuner after a Reset or application of power must be either a Tuner Setup command or a Tuner Read command.

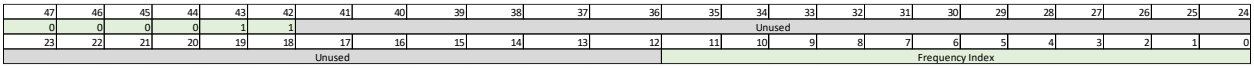
3.3. Set_Atten (Command Code 000010)



Parameter	Bit(s)	Description	Possible Values
RFAttenuation	18:13	Sets the attenuation of the tuner without changing any other tuner parameters	0-38 dB in 1dB steps

The Set_Atten command is used when the attenuation is the only Tuner parameter that needs to change.

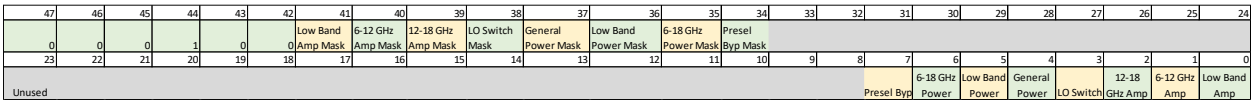
3.4. Set_Freq (Command Code 000011)



Parameter	Bit(s)	Description	Possible Values
Unused	41:12	Unused	
Frequency Index	11:0	Index for the desired center frequency. Calculated using the formula: $(CF - 350)/5$ where CF is the desired center frequency in MHz in 5 MHz steps.	0 – 3480: 0 is for 350 MHz center, 3480 is for 17750 MHz center

The Set_Freq command is used when the frequency index is the only parameter that needs to change. All other parameters retain their current values. The attenuation will adjust to the new Frequency Index's calibration settings but the net result will be the current attenuation set by the last Tuner_Setup or Set_Atten command. AGC_Amp status will also not change.

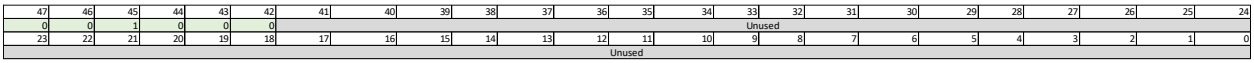
3.5. Set_Config (Command Code 000100)



Parameter	Bit(s)	Description	Possible Values
Low Band Amp Mask	41	Determines whether Low Band Amp setting is applied	0: Ignore Low Band Amp setting 1: Apply Low Band Amp setting
6-12 GHz Amp Mask	40	Determines whether 6-12 GHz Amp setting is applied	0: Ignore 6-12 GHz Amp setting 1: Apply 6-12 GHz Amp setting
12-18 GHz Amp Mask	39	Determines whether 12-18 GHz Amp setting is applied	0: Ignore 12-18 GHz Amp setting 1: Apply 12-18 GHz Amp setting
LO Switch Mask	38	Determines whether LO Switch setting is applied	0: Ignore LO Switch 1: Apply LO Switch setting
General Power Mask	37	Determines whether General Power Setting is applied	0: Ignore General Power setting 1: Apply General Power setting
Low Band Power Mask	36	Determines whether Low Band Power setting is applied	0: Ignore Low Band Power setting 1: Apply Low Band Power setting
6-18 GHz Power Mask	35	Determines whether 6-18 GHz Power setting is applied	0: Ignore 6-18 GHz Power setting 1: Apply 6-18 GHz Power setting
Presel Byp Mask	34	Determines whether the Presel Byp setting is applied	0: Ignore Presel Byp setting 1: Apply Presel Byp setting
Unused	33:8	Unused	
Presel Byp	7	If center frequency is <= 6000 MHz, determines whether the preselector filters are bypassed	0: Preselectors engaged 1: Preselectors bypassed
6-18 GHz Power	6	Controls power to amplifiers in the path used by center frequencies > 6000 MHz that aren't controlled by another control line. 6-18 GHz Power can also be turned off with the General Power setting.	0: 6-18 GHz Power Off 1: 6-18 GHz Power On
Low Band Power	5	Controls power to amplifiers in the path used by center frequencies <= 6000 MHz that aren't controlled by another control line. Not affected by General Power setting	0: Low Band Power Off 1: Low Band Power On
General Power	4	Controls power to circuitry that is not frequency specific. Turns off power to 6-18 GHz circuitry.	0: General Power Off 1: General Power On
LO Switch	3	Manually selects direction of LO distribution switch	0: > 6 GHz RF path 1: <= 6 GHz RF path
12-18 GHz Amp	2	Manually controls 12-18 GHz amplifier	0: 12-18 GHz amplifier bypassed 1: 12-18 GHz amplifier on
6-12 GHz Amp	1	Manually controls 6-12 GHz amplifier	0: 6-12 GHz amplifier bypassed 1: 6-12 GHz amplifier on
Low Band Amp	0	Manually controls the <= 6000 MHz amplifier	0: Low Band Amp bypassed 1: Low Band Amp On

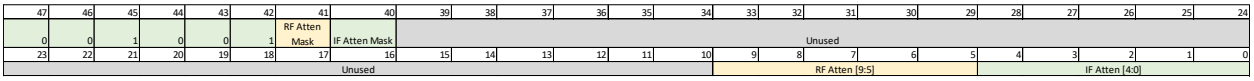
The Set_Config command is used to override the settings for tuner elements automatically determined by Tuner_Setup. This command utilizes mask bits to determine if the setting contained in the command will be applied to the tuner. This way any individual setting can be changed without having to know what any of the other settings are (just set their mask setting to 0). This command does not change the frequency or attenuation of the tuner.

3.6. Reset_Tuner (001000)



The Reset_Tuner command resets the Tuner's FPGA and puts it into its power up configuration. This command has no parameters. When this command is used the Tuner's Read_Mask is set to 001 (from the Tuner_Read command) and the serial number and HW Rev will be contained in the MISO stream during the next SPI transaction.

3.7. Manual Set Atten (001010)



Parameter	Bit(s)	Description	Possible Values
RF Atten Mask	41	Determines whether the setting in RF Atten is applied	0 = Ignore setting 1 = Apply setting
IF Atten Mask	40	Determines whether the setting in RF Atten is applied	0 = Ignore setting 1 = Apply setting
RF Atten	9:5	RF (front end) attenuator setting. Controls the first attenuator in the downconverter and 6 GHz paths	0-31 dB in 1 dB steps
IF Atten	4:0	IF attenuator setting. Controls the second attenuator in the downconverter and 6 GHz paths	0-31 dB in 1 dB steps

The Manual Set Atten command is used to override the calibration settings for the attenuators in the FPGA flash.

3.8. Manual Set Band (Command Code 001011)

47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32	31	30	29	28	27	26	25	24
0	0	1	0	1	1	Band Mask	LPFA Mask	HPFA Mask	LPFB Mask	HPFB Mask	Unused												
23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Unused	HPFB					LPFB					HPFA					LPFA					Band		

Parameter	Bit(s)	Description	Possible Values
Band Mask	41	Determines whether the setting in Band is applied	0 = Ignore setting 1 = Apply setting
LPFA Mask	40	Determines whether the setting in LPFA is applied	0 = Ignore setting 1 = Apply setting
HPFA Mask	39	Determines whether the setting in HPFA is applied	0 = Ignore setting 1 = Apply setting
LPFB Mask	38	Determines whether the setting in LPFB is applied	0 = Ignore setting 1 = Apply setting
HPFB Mask	37	Determines whether the setting in HPFB is applied	0 = Ignore setting 1 = Apply setting
Unused	36:23	Unused	
HPFB	22:18	Sets the tune word for the second HPF (if applicable)	0-31, for some filters only 0-15 may be valid
LPFB	17:13	Sets the tune word for the second LPF (if applicable)	0-31, for some filters only 0-15 may be valid
HPFA	12:8	Sets the tune word for the first HPF	0-31, for some filters only 0-15 may be valid
LPFA	7:3	Sets the tune word of the tunable LPF in band 1.	0-31, for some filters only 0-15 may be valid
Band	2:0	Manually chooses the band	0-4 (0 = Band 1, 4 = Band 5) Note: any setting higher than 4 will select Band 1.

The Manual Set Band command allows the user to override the default/calibrated filter settings and choose their own.

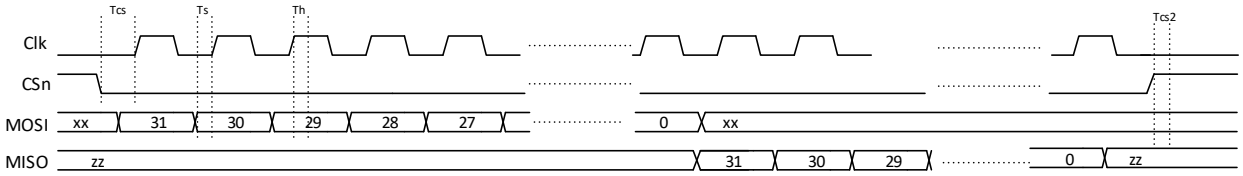
4. Tuner Initialization

The Tuner requires a simple initialization sequence to begin operation. When 3.3V is applied to the Tuner's 3.3V input pin the FPGA will boot. The boot up sequence takes approximately 3ms to complete. Once the sequence is complete and the Tuner's On/Off pin is high (it can be set to high before the FPGA boot starts) the Tuner will be awaiting initial configuration. At this point there are only 3 commands that will produce meaningful results, Tuner_Read, Tuner_Reset, and Tuner_Setup. A Tuner_Setup command needs to be sent before the other commands can be used to configure the Tuner. This is the case after initial FPGA boot up, after a Tuner_Reset command, and after the Tuner's On/Off pin is toggled from Off to On. Once the initial Tuner_Setup command is complete the Tuner can then be controlled with the commands Set_Atten, Set_Freq, Set_Config, as well as with the Tuner_Setup command.

5. Tuner FPGA Update

The Tuner's onboard FPGA can be updated via the same CLK, MOSI, and MISO lines used for Tuner Control. The SSN PROG (pin 28) pin is used as the CSn line instead of the SSN CMD pin. The Timing and sampling edges are different from the Tuner Control SPI interface. The Tuner Control interface is

designed to have transmit data (MOSI) sampled on the positive edge of the clock and receive data (MISO) sampled on the negative edge while the programming interface sample both the MISO and MOSI lines on the positive edge. Additionally, when data is read back on the MISO line that data will begin being sent after the MOSI transmission is complete. The transmitting of MOSI data and receiving of MISO data is all part of a single SPI transaction. A timing diagram along with a table of timing parameters is shown below. Programming the USERCODE and SECURITY settings will not be done so that step should be skipped and this device is not being configured via the internal WISHBONE interface. The process is complete once the REFRESH command has been sent.

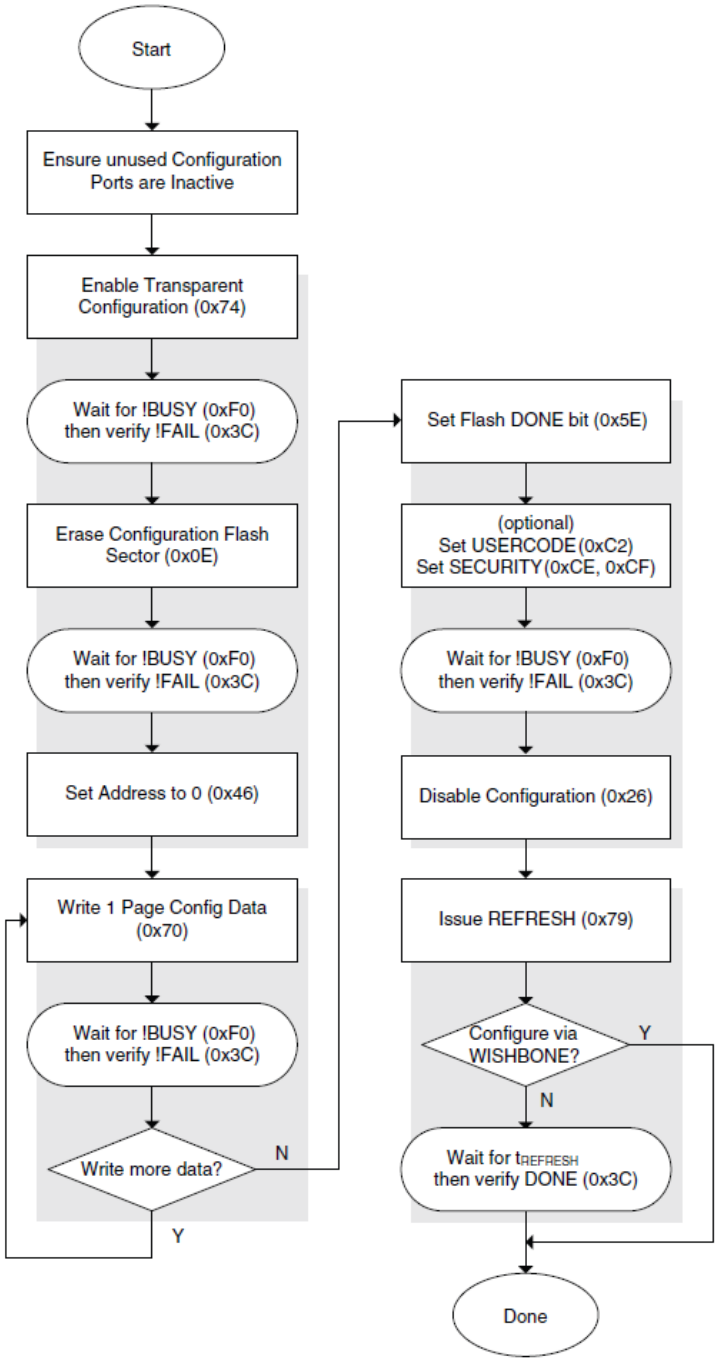


Parameter	Description	Min	Max
SPI Clock Frequency	Maximum frequency of SPI Clock		66 MHz
Tcs	Delay between CSn going low and first positive clock edge	15ns	
Ts	Time data on MOSI and MISO must be stable preceding the sampling SPI clock positive edge	7ns	
Th	Time data on MOSI and MISO must remain stable following the sampling SPI clock positive edge	2ns	
Tcs2	Time CSn must be high between SPI transactions	25ns	

5.1. FPGA Image Update Procedure

The FPGA image is stored in Flash Memory on the FPGA itself. The Flash Memory (called Configuration Flash) consists of 9211 pages, each 128 bits in length. The procedure for updating the FPGA's Configuration Flash requires erasing the flash and then writing all of the necessary pages that contain program data, up to 9211 pages. It involves enabling the FPGA's configuration mode, erasing the flash, writing the data, disabling the config interface, and setting the DONE bit. During the process there are multiple checks for the previous command's completion/success. Once the process is complete a REFRESH command can be applied and the FPGA will "reboot" using the new configuration flash image.

5.1.1. FPGA Update Flow Chart



5.1.2. Check Device ID (0xE0)

Reads the FPGA Device ID. The command has 3 operands (1 byte each) and 32 bits of data read from the FPGA for an overall length of 64 bits. The operands are fixed values. In the example below the 'Z's are 'don't care' states for when the MISO is not reporting valid data or the MOSI data doesn't matter and the 'x's are the 4 byte device ID. This command can be used to verify the communication interface is working.

Example Command (MOSI): 0xE0 00 00 00 ZZ ZZ ZZ ZZ
Example Command (MISO): 0xZZ ZZ ZZ ZZ xx xx xx xx

The 4 byte device ID should be 0x61 2B 50 43.

5.1.3. Enable Transparent Configuration Command (0x74)

Places the FPGA in a state where it can write or read to the Configuration Flash. Transparent Configuration Mode allows the FPGA to maintain its current GPIO states while the Flash is being updated. The command has 3 operands (1 byte each) resulting in a command length of 32 bits. The operands are fixed values. The Busy Flag must be polled after this command is sent.

Example Command: 0x74 08 00 00

5.1.4. Check Busy Flag Command (0xF0)

Causes the FPGA to output 8 bits of data containing the busy flag. The command has 3 operands (1 byte each) and reads back 1 byte of output data resulting in an overall command length of 40 bits. The operands are fixed values.

Example Command (MOSI): 0xF0 00 00 00 ZZ
Example Command (MISO): 0xZZ ZZ ZZ ZZ 80 (Busy bit is contained in the MSB of the data returned)

Output data format:

7	6	5	4	3	2	1	0
Busy Flag	Reserved						

Parameter	Bit(s)	Description	Possible Values
Busy Bit	7	Indicates whether the FPGA is ready to receive commands.	0: FPGA is ready to receive commands 1: FPGA is busy

5.1.5. Erase Flash (0x0E)

Erases the specified Flash sector of the FPGA. The command has 3 operands (1 byte each) resulting in a command length of 32 bits. The operands control which flash sector gets erased. It is very important to correctly send the command exactly as shown in the example or unintended Flash sectors could be

erased. Only the first operand affects the commands operation. The Busy Flag must be polled after this command is sent.

Example Command: 0x0E 04 00 00 (Erase Configuration Flash ONLY)

Operand #1 Format:

7	6	5	4	3	2	1	0
Reserved				Erase UFM	Erase CFG	Erase Feature	Erase SRAM

Parameter	Bit(s)	Description	Possible Values
Erase UFM	3	If set the User Flash will be erased. SET THIS TO 0.	0: No action 1: Erase UFM
Erase CFG	2	If set the Configuration Flash will be erased. SET THIS TO 1.	0: No action 1: Erase Configuration Flash
Erase Feature	1	If set the Feature Row will be erased. SET THIS TO 0.	0: No action 1: Erase Feature Row
Erase SRAM	0	If set the SRAM will be erased. SET THIS TO 0.	0: No action 1: Erase SRAM

5.1.6. Check Status Command (0x3C)

Reports the general status of the FPGA. Can be used in place of 0xF0 to check the busy bit. The command has 3 operands (1 byte each) and 32 bits of data read from the FPGA for an overall length of 64 bits. The operands are fixed values. In the example below the 'Z's are 'don't care' states for when the MISO is not reporting valid data or the MOSI data doesn't matter and the 'x's are the 4 byte status message from the FPGA.

Example Command (MOSI): 0x3C 00 00 00 ZZ ZZ ZZ ZZ
Example Command (MISO): 0xZZ ZZ ZZ ZZ xx xx xx xx

Output Data Format:

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Unused				RESERVED	Unused	RESERVED			Unused						
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Unused		Fail	Busy	Unused		Cfg Intfc	RESERVED	Unused							

Parameter	Bit(s)	Description	Possible Values
RESERVED	27, 25:23, 8	Used by FPGA to indicate certain status that is not applicable to FPGA updates	N/A
Fail	13	Fail flag indicates if the previous operation succeeded.	0: Operation Succeeded 1: Operation Failed
Busy	12	Indicates whether the FPGA is ready to receive commands.	0: FPGA is ready to receive commands 1: FPGA is busy
Cfg Intfc	9	Indicates whether the FPGA is in Configuration Mode (if 0x74 has been sent this should be a 1)	0: Configuration Mode Disabled 1: Configuration Mode Enabled

5.1.7. Set FPGA Flash Address to 0 (0x46)

Sets the address where data will be read from/written to 0 (start of flash sector). The command has 3 parameters (1 byte each) and no output data for an overall length of 32 bits. The operands are fixed values.

Example Command: 0x46 00 00 00

5.1.8. Write Page of Flash Data (0x70)

Writes the 128 bit data section of the command into the current Flash address. The Flash address is incremented by 1 after the completion of this command. The command has 3 operands and 128 bits of data to be written to the FPGA for a command length of 160 bits. The operands have fixed values. The Busy Flag must be polled after this command is sent.

Example Command: 0x70 00 00 01 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F

The data in the above example is 0x00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F

5.1.9. Set Flash DONE Bit (0x5E)

Sets the DONE bit indicating the end of the data transfer. The command has 3 operands and no data for a command length of 32 bits. The Busy Flag must be polled after this command is sent.

Example Command: 0x5E 00 00 00

5.1.10. Disable Configuration Interface (0x26)

Disables the Configuration Interface putting the FPGA back into user mode. The command only has 2 operands and no data for a command length of 24 bits. The operands are fixed values.

Example Command: 0x26 00 00

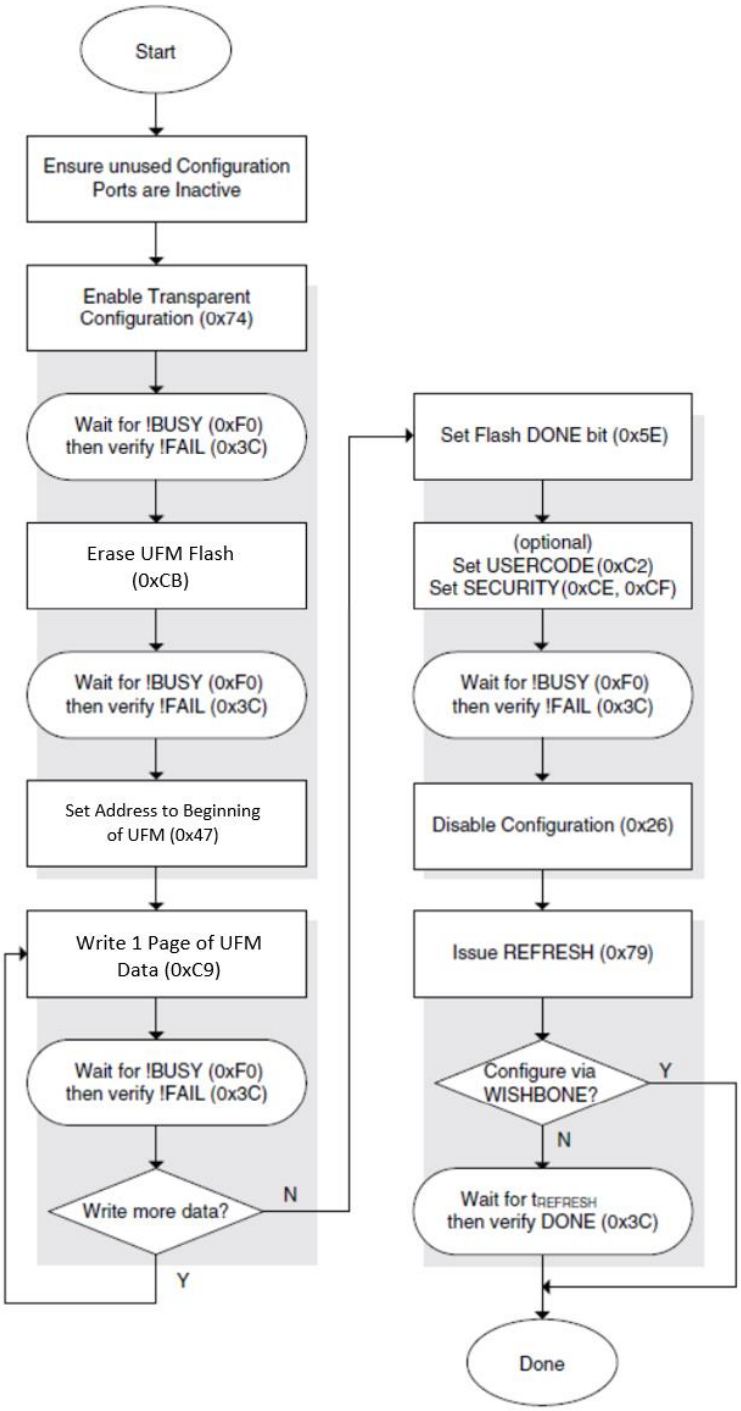
5.1.11. Refresh (0x79)

Issues the Refresh command that loads the data from the Configuration Flash into the FPGA making that the active image. The command has 2 operands and no data for a command length of 24 bits. The operands have fixed values. Once the command has been sent no commands may be set for a time of Trefresh or the FPGA will not boot (although power can then be cycled and it will boot into the new image).

Example Command: 0x79 00 00

5.2. FPGA Calibration Flash Update Procedure

The User Flash can be programmed over the SPI interface using a very similar algorithm to the SPI FPGA update. The flow chart below was taken from the SPI API document and modified to program the flash. The only modified sections are the Erase command (0xCB instead of 0x0E), the Set Address command (0x47 instead of 0x46), and the write data command (0xC9 instead of 0x70). Only the modified commands will be defined here. The MachXO3 6900 used on the AM9017 has 2046 pages of User Flash Memory (UFM). The rest of the commands are defined in section 5.1.



5.2.1. Erase UFM Flash (0xCB)

Erases the User Flash Memory. Must be done before writing new data to the UFM. The command has 3 operands (1 byte each) and no output data for an overall length of 32 bits.

Example command: 0xCB 00 00 00

5.2.2. Reset UFM Address (0x47)

Sets the Flash Address to the start of the UFM. The command has 3 parameters (1 byte each) and no output data for an overall length of 32 bits. The operands are fixed values.

Example command: 0x47 00 00 00

5.2.3. Write 1 Page UFM Data (0xC9)

Writes the 128 bit data section of the command into the current Flash address. The Flash address is incremented by 1 after the completion of this command. The command has 3 operands and 128 bits of data to be written to the FPGA for a command length of 160 bits. The operands have fixed values. The Busy Flag must be polled after this command is sent.

Example Command: 0xC9 00 00 01 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F

The data in the above example is 0x00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F

6. Tuner Synchronization (multiple tuners)

The AM9017 has the capability to have the tuning LO synchronized with other AM9017s. This is accomplished by sending a SYNC pulse on pin 30 of the multipin connector. Approximately 265us after the SYNC pulse is applied the tuning LOs of all AM9017s that received the pulse will have a defined phase relationship with each other. That defined relationship is consistent for a given center frequency. Tuning to a new frequency, tuning back, and applying the sync pulse will result in the same relationship between the two channels until power is cycled.

6.1. SYNC Electrical Characteristics

The SYNC pulse is applied to pin 30 on the AM9017.

DC Electrical Characteristics

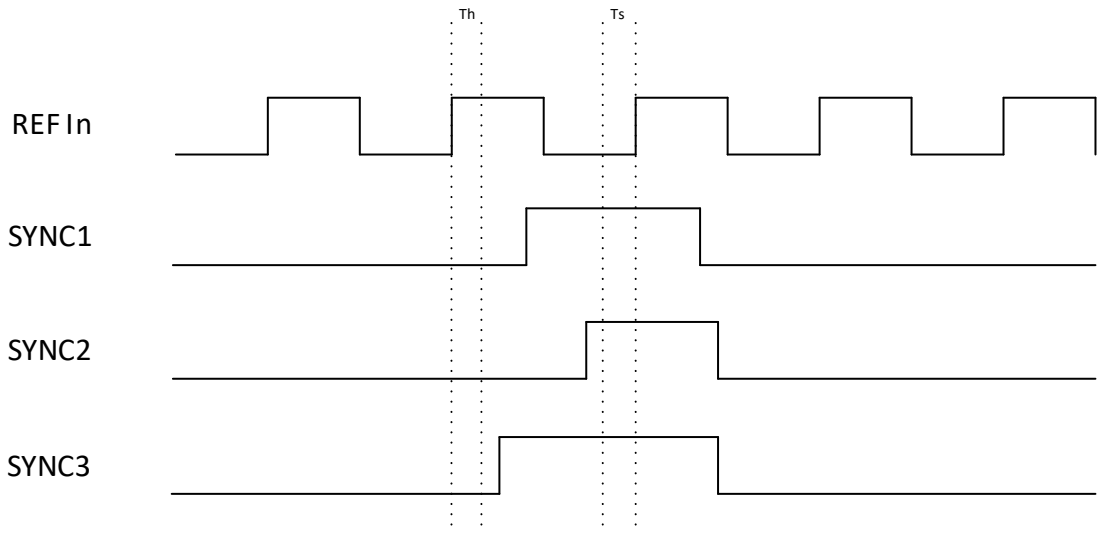
(T = 25 °C unless otherwise specified)

Parameter	Testing Conditions	Minimum	Typical	Maximum
Logic Level Low		+0V	--	0.4V
Logic Level High		+1.4V	--	Vcc*

*Vcc is the voltage applied to the multipin +3.3V pins.

6.2. SYNC Timing

For the phase relationship between the multiple tuners' tuning LOs to be consistent over time the SYNC pulses need to arrive at those tuners within the same reference clock pulse while maintaining setup time (Ts) to the next clock pulse and hold time to the previous clock pulse (Th). The pulses must also start long enough after the previous rising clock edge to guarantee they are all read on the same rising edge. The SYNC pin is edge triggered so only the rising edge of the pulse has to meet the timing. The diagram below shows valid SYNC timing for 3 tuners. All 3 rising SYNC edges are in the same REF clock period and within the hold/setup time window. It is assumed that the REF In and all SYNC lines are distributed with equal lengths to each tuner. The REF In clock is 100 MHz so the window for SYNC arrival is 5.5nS.



(T = 25 °C unless otherwise specified)

Parameter	Testing Conditions	Minimum	Typical	Maximum
Th		2 nS	--	--
Ts		2.5 nS	--	--

6.3. SYNC Behavior

This section will discuss how the SYNC operates in different scenarios.

6.3.1. Integer N tuning (100 MHz step size, starting at 400 MHz Fc)

The SYNC pulse only has to be sent once if the center frequency is an integer multiple of the reference. The SYNC pulse will have to be resent if a center frequency that is not an integer multiple of the reference is used.

6.3.2. Fractional Tuning (Fc not a multiple of the reference)

Fractional tuning requires the SYNC pulse to be sent after every change in frequency.