SYNERGY® MICROWAVE CORPORATION

201 Mclean Blvd, Paterson, NJ 07504 AN7600 Rev.:-- 01/28/13

Intelligent Interactive Synthesizer Pin-out and Functions (FSWX200800-100)

This document describes the operating features and pin-out of the new generation of Synergy Microwave's "Intelligent Interactive Synthesizers" ($I^2S^®$). It is a specific document for the FSWX200800-100 synthesizer. This synthesizer is truly intelligent incorporating an internal controller that allows for optimal factory set performance and minimal software development by the user. Programming is easily done through four simple options that the user can adopt for fast in-circuit implementation.

What is the need for synthesizers having an integrated controller?

There are several good reasons to do so!

- The operational settings of the PLL chip have to be calculated for each frequency setting to be within the operational specifications. For example in wideband VCO's, the computation of the divider values has to guarantee that no chip internal frequency limits are exceeded.
- The settings of the charge pump current and other parameters, which finally define the phase noise and switching speed can be optimized easily with these intelligent features.
- Improves the speed of the system processor and makes the software easier to re-use.
- Operation can be made independent of the PLL IC used. The use of a controller allows change of the $I^2S^®$ IC without any impact on the system software.
- Integrated error detection and signaling is possible without customer programming.
- Reduces design time for the user. A real plug and play solution!

Other benefits of the $I^2S^®$ are:

- Intensive internal filtering of the supply voltages.
- Monitoring of internal voltages.
- Optimized layout to reduce impact of external ground loops.
- Standard programming interface for ALL models.
- No hardware programming for a specific PLL IC required.

Software Interface Pins

This section describes the software interface pins that are used to address the synthesizers. Some of the pins are not connected and can be used at a later time for software implementation of additional features. All I/O’s are 3.3V CMOS, inputs with pull-up resistors.

- Latch Enable
  LE for SPI port. (SS)
- Data In
  Data for SPI port. (MOSI)
- Clock In
  Clock for SPI port. (SCK)

Hardware Interface Pins

- Fref
  Reference frequency input.
- Vcc(Digital)
  +3.3V DC supply for the digital section of the Synthesizer. A 47uF tantalum bypass capacitor is highly recommended if the supply isn’t clean to achieve the specified phase noise performance!
- Vcc(VCO)
  Supply voltage for the VCO according to the datasheet. This has to be a clean, low noise supply. Decoupling close to the Synthesizer package is recommended.
- Vcc(Tune)
  Supply voltage for the VCO tuning according to the datasheet or from the internal voltage converter. This has to be a clean, low noise supply. Decoupling close to the Synthesizer package is recommended.
- LD
  Lock Detect, a 3.3V CMOS output, which turns high after the Synthesizer locks.
- RFout
  50Ω output of the Synthesizer.
- No Connection – leave these pins open (DO NOT CONNECT TO GROUND)

  Note: External noise from the power supplies must be limited to no more than 30uV/Sqrt_Hz to achieve the specified phase noise performance.

Programming

The user can address the I^{2}S$^{®}$ synthesizers in one of four different options to program the output frequency:

1. **Setting the frequency in MHz by a 16bits Hexadecimal number**. In this mode, the output frequency is represented in MHz as a 16 bits Hexadecimal number. The programming format for this option is coded as "Little Endian" (MSB of the LSByte is sent first). This allows representing a frequency between 1 to 65535MHz.

   For example, FSWX200800-100 would be sent as follows:

Frequency [MHz]	Programming (code sent to device's SPI)
2000	0xD007
2001	0xD107
4000	0xA00F
8000	0x401F

   Note: Most microcontrollers use this format in representing integer numbers in internal registers. If that kind of microcontroller is used, the device can be programmed by simply sending the frequency register data to the SPI interface controlling the device.

2. **Setting the frequency in MHz by a 24bits BCD number** In this mode, the output frequency is represented in MHz as a 24 bits BCD number. This allows representing a frequency between 1 to 65535MHz

   For example, FSWX200800-100 would be sent as follows:

Frequency [MHz]	Programming (code sent to device's SPI)
2000	0x002000
2001	0x002001
4000	0x004000
8000	0x008000

3. **Setting frequency directly (kHz)**. In this mode, an ASCII character code for the letter "K" in hexadecimal (4B) indicates to the controller that mode frequency directly (kHz) is selected. The information is loaded in hexadecimal format (1 byte for the "K" command and 8 bytes for the frequency command (72 bits total). No real values allowed, the instruction must be in integer format (KHz).

For example, the frequency of 4,000,000 KHz (4 GHz) would be sent as:

Data ASCII(Hex)		Hex
4B 30 30 33 44 30 39 30 30	or	K003D0900

4. **Setting frequency directly (MHz)**. In this mode, an ASCII character code for the letter “M” in hexadecimal (4D) indicates to the controller that mode frequency directly (MHz) is selected. The information is loaded in hexadecimal format (1 byte for the “M” command and 8 bytes for the frequency command (72 bits total). No real values allowed, the instruction must be in integer format (MHz).

   For example, the frequency of 4,000 MHz (4 GHz) would be sent as:

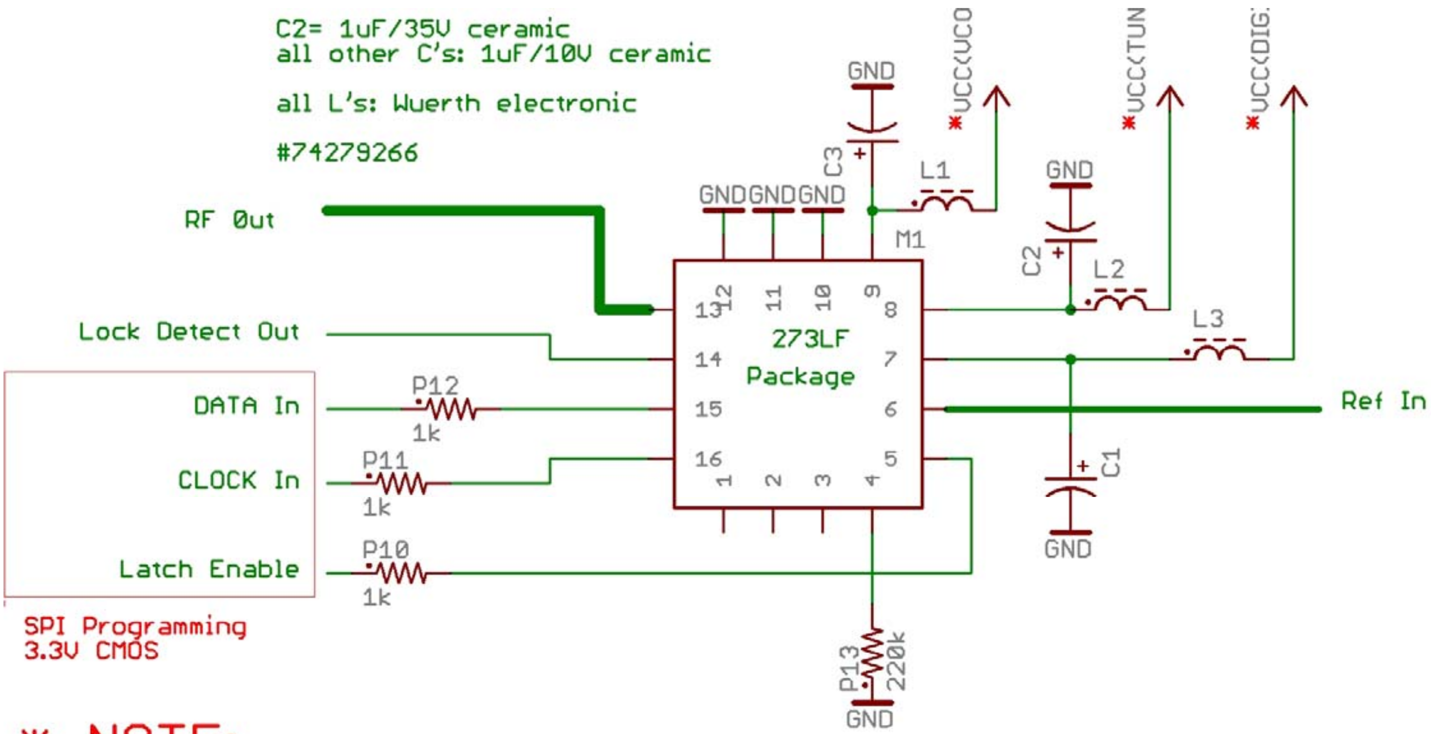
Data ASCII(Hex)		Hex
4D 30 30 30 30 30 46 41 30	or	M00000FA0

Reference Frequency – The factory set reference frequency is 100 MHz, which gives the maximum phase noise and spurious performance. Contact factory if other reference frequency is desired

Maximum SPI programming speed is recommended not to exceed 200kb/sec in present designs.

Suggested Connection Diagrams:

FSW Series



*** NOTE:**

Common Voltage Supplies can be Connected together and Filtered as indicated in the diagram.

Please Check the Specific Product Specification Sheet for the Supply Voltages.

Table I

Interface Commands

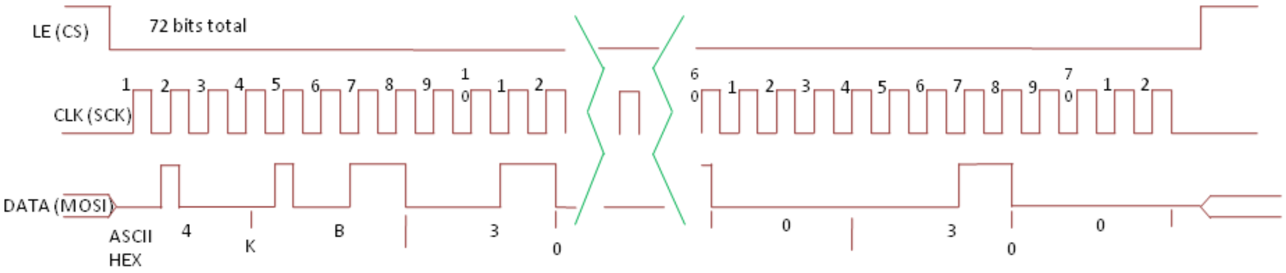
Command Character	Command Description	Parameter	Byte Position								
			1	2	3	4	5	6	7	8	9
-	Setting the frequency in MHz by a 16bits Hexadecimal number	Unsigned 16bit Integer, LSByte first	LSB	MSB							
-	Setting the frequency in MHz by a 24bits BCD number	Unsigned 24bit Integer, LSByte first	BCD	BCD	BCD						
K	Set Active Frequency in kHz	Hexadecimal characters	0x4B	H	H	H	H	H	H	H	H
M	Set Active Frequency in MHz	Hexadecimal characters	0x4D	H	H	H	H	H	H	H	H

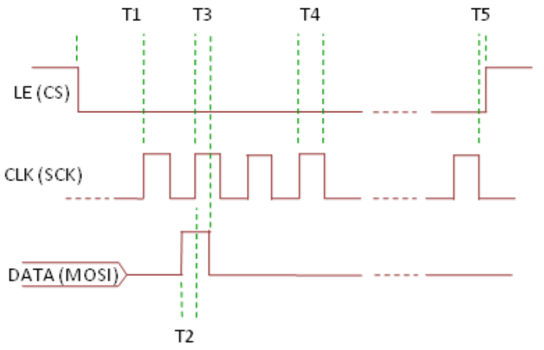
All SPI communication uses most significant bit first
“B” – Binary value with least significant byte first.
“H” – ASCII value of a hexadecimal character (uppercase) most significant character first.

Examples:

“16bit” Command:	Set Active N using Frequency of 4000MHz →	0xA00F
“24bit” Command:	Set Active N using Frequency of 4000MHz →	0x004000
“K” Command:	Set Active N using Frequency of 4000000kHz →	4B 30 30 33 44 30 39 30 30 or “K003D0900”
“M” Command:	Set Active N using Frequency of 4000MHz →	4D 30 30 30 30 30 46 41 30 or “M00000FA0”

TIMING DIAGRAM





T1: >5uSec
T2: >50 nSec
T3: >1 uSec
T4: >2.5 uSec
T5: >20 nSec