# i²S®

# Intelligent Interactive Synthesizer
# Pinout and Functions (packages 366 and 367)

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 generic document for the MFSH series of ($I^2S^®$) synthesizers in packages 366 and 367.   These synthesizers are 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 5V CMOS, inputs with pull-up resistors. Their input characteristic has also been set to accommodate 3V CMOS input signals at slightly reduced noise immunity.

- 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)
  +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) – available access on package 367 only. Voltage must be clean (low noise). Decoupling close to the Synthesizer package is recommended.
- LD
  Lock Detect, a 3.3V CMOS output, which turns high after the Synthesizer locks.
- RFout
  50$\Omega$ output of the Synthesizer.
- No Connection – leave these pins open (DO NOT CONNECT TO GROUND)

**Programming**

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

1. **Setting frequency allocation as a channel**. In this mode, an ASCII character code for the letter "C" in hexadecimal (43) indicates to the controller that mode frequency as a channel is selected. The information for the channel number follows in hexadecimal format having 32 bits with LSB sent first (total 40 bits sent). Based on the channe

selected, the controller then calculates the frequency of the channel $F_{CN}$ by the following equation:

$$F_{CN} = F_{C0} + (N*C_S), \text{ where:}$$

$F_{CN}$ - Frequency of the $N_{th}$ channel
$F_{C0}$ - Specified starting frequency of the synthesizer
$C_s$ - Channel spacing
$N$ - $N_{th}$ channel for synthesizer to tune

For example, channel 50 for the MFSH170340-50 would be sent as follows:
          Data (Hex)
      43 32 00 00 00

Similarly, the synthesizer can immediately be loaded for standby channel mode by sending ASCII character code for letter "c" in hexadecimal (63) and the data for the next wanted channel number. The programming format for this option is coded as "Little Endian" (MSB of the LSByte is sent first).

2.    **Setting frequency by known N (division ratio)**.  In this mode, an ASCII character code for the letter "D" in hexadecimal (44) indicates to the controller that mode frequency by known N is selected.  N is calculated by dividing the output frequency by the channel spacing (step size).  The information for the N division ratio follows in hexadecimal format having 32 bits with LSB sent first (total 40 bits sent).

For example, an N ratio of 2700 would be loaded as:
          Data (Hex)
      44   8C 0A 00 00

Similarly, the synthesizer can immediately be loaded for standby mode by sending ASCII character code for letter "d" in hexadecimal (64) and the data for the next wanted known N. The programming format for this option is coded as "Little Endian" (MSB of the LSByte is sent first).

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

Similarly, the synthesizer can immediately be loaded for standby mode by sending ASCII character code for letter "k" in hexadecimal (6B) and the next wanted frequency (KHz).

**Reference Frequency** –  The factory set reference frequency is usually 10 MHz, in some models can be field re-programmed by the user for any reference frequency, typically in the range of 10 to 150 MHz, as long as the reference frequency is an integer multiple of the step size.  Some models are factory set and cannot be field re-programmed, see specification sheet for specifics on field programmability.  For synthesizer models that are not field re-programmable, contact factory for your specific reference frequency requirement.

For those models which the reference frequency can be field reprogrammable, the factory start-up reference frequency can be changed in two ways:

1. A volatile "R" command change, where the reference register settings are loaded every time with each command to over ride the factory setting.  There is no limit to the number of register loads when using the "R" command setting.

2. A non-volatile command change where the factory setting is permanently changed in the EPROM by the user.  In this case, there is a maximum limit to the number of times that the EPROM can be written (up to 100 times).  This should be more than enough times to write to EPROM once the instruction is written by the user for the desired reference frequency there is usually no need to change again.

For the volatile change of the reference (setting sent with each frequency command), an ASCII character code for the letter "R" in hexadecimal (52) indicates to the controller that reference frequency is being temporarily changed.   The information for the division ratio follows in hexadecimal format having 32 bits with LSB sent first - 1 byte for the "R" command and 4 bytes for the division ratio (40 bits total).  The information is loaded in hexadecimal format.

For example, with a reference frequency of 10 MHz and a step size of 500 KHz, (division ratio of 20) the data would be sent as:

Data (Hex)
52  14  00  00  00

The programming format for this option is coded as "Little Endian" (MSB of the LSByte is sent first).

For the non-volatile change of the factory setting, the R command is sent as indicated previously and followed by the letter "r" in hexadecimal (72), indicating to the controller that the EPROM will be written to with a new non-volatile user setting.  For example, a reference frequency of 10 MHz and step size of 500 kHz (division ratio of 20) the data would be sent as:

4

                              Data (Hex)
         1)            52 14 00 00 00
         2)            72

The programming format for this option is coded as "Little Endian" (MSB of the LSByte is sent first).

**Start-up (wake-up) Frequency** – Upon power up, the synthesizer is factory programmed to automatically self start at the specified starting frequency.  The factory set start-up frequency can be field re-programmable to start at any other frequency within the specified bandwidth. This feature adds flexibility for adaptation in fixed frequency applications where on-demand programming is unavailable.

The reprogramming process involves sending the new frequency command either by channel, by known N division ratio or frequency directly in (kHz) and then followed by the ASCII letter "f" command in Hex (66).

For example, the sequence is as follows to change the wake-up frequency to 4,000,000 kHz (4 GHz):

                              Data (Hex)
         1)            4B 30 30 33 44 30 39 30 30
         2)            66

**Swap Active/Standby frequencies –** When an ASCII character code for the letter "S" in hexadecimal (53) is sent, it indicates to the controller that swap frequency command requests swapping the active and the standby frequency register settings.  The information follows in hexadecimal format having 8 bits with LSB sent first - 1 byte for the "S" command.

                    Data (Hex)
                       53

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

By default all I$^2$S® are set to a reference frequency of 10 MHz unless otherwise noted in the datasheet.  If the reference frequency in an application is different the first data string sent to the Synthesizer has to be the R command! On turn on, the Synthesizer defaults to the lowest specified frequency (channel 0).  Standby frequency defaults to maximum frequency unless otherwise specified in the datasheet.
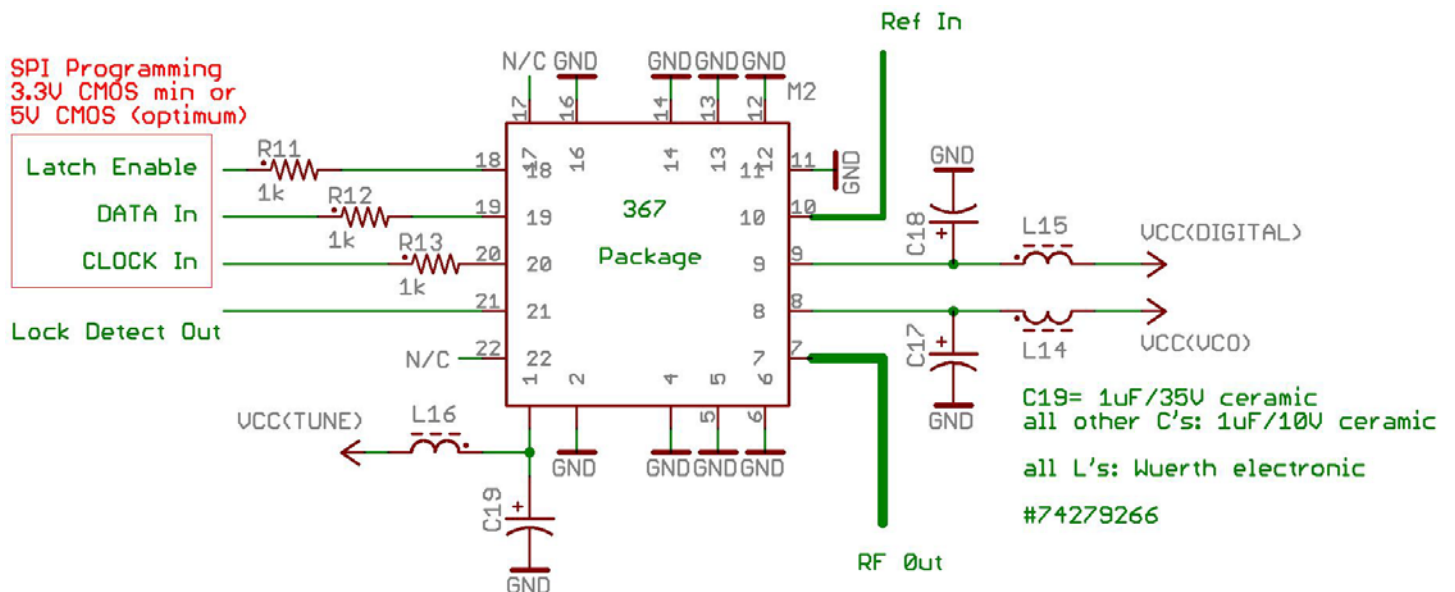
Table I shows some examples of the different programming options and figure 1 shows a typical timing diagram. The number of bits in the diagram below applies to the example for MHz or kHz commands, but the timing applies for all test commands described in the above text.
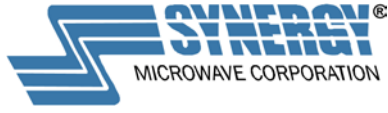
5

# Suggested Connection Diagrams:

## MFSH Series (package outline 366)



"Ref In" and "RF Out" are internally AC coupled ports

## MFSH Series (package outline 367)



6

"Ref In" and "RF Out" are internally AC coupled ports

**SYNERGY**®
MICROWAVE CORPORATION
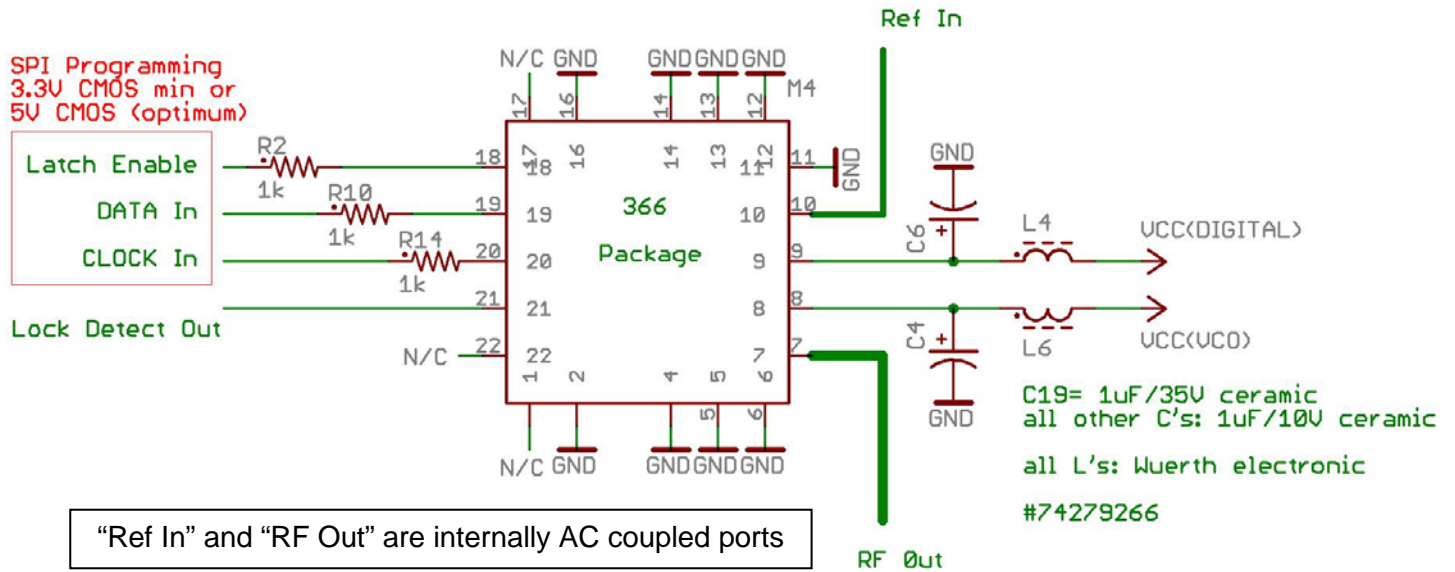
**Table I**

Interface Commands

| Command Character | Command Description | Parameter | Byte Position | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| C | Set Active Frequency as Channel | Unsigned Long Integer, LSByte first | 0x43 | B | B | B | B | | | | |
| c | Set Standby Frequency as Channel | Unsigned Long Integer, LSByte first | 0x63 | B | B | B | B | | | | |
| D | Set Active Division Ratio | Unsigned Long Integer, LSByte first | 0x44 | B | B | B | B | | | | |
| d | Set Standby Division Ratio | Unsigned Long Integer, LSByte first | 0x64 | B | B | B | B | | | | |
| K | Set Active Frequency in kHz | Hexadecimal characters | 0x4B | H | H | H | H | H | H | H | H |
| k | Set Standby Frequency in kHz | Hexadecimal characters | 0x6B | H | H | H | H | H | H | H | H |
| f | Reset for factory default start-up (wake-up) frequency to non-volatile setting | No parameter | 0X66 | | | | | | | | |
| R | Reference Frequency In Multiples Of Channel Spacing | Unsigned Long Integer, LSByte first | 0x52 | B | B | B | B | | | | |
| r | Changes factory set reference frequency to new non-volatile setting | No parameter | 0x72 | | | | | | | | |
| S | Swap Active / Standby Frequencies | No parameter | 0x53 | | | | | | | | |

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.
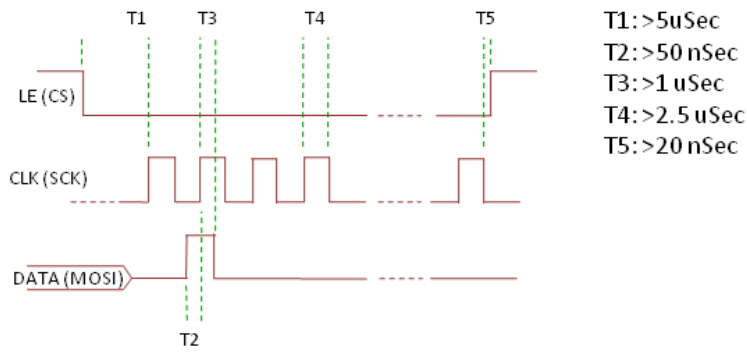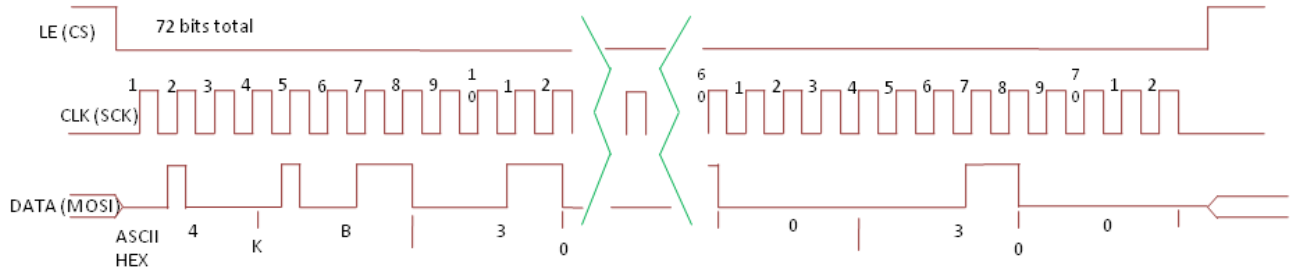"C" – ASCII value of option parameter.

Examples:
"D" Command:     Set Active N using Division Ratio of 2700 →          44 8C 0A 00 00     (2700 = 0x00000A8C)
"d" Command:     Set Standby N using Division ratio of 2700 →          64 8C 0A 00 00     (2700 = 0x00000A8C)
"K" Command:     Set Active N using Frequency of 4000000kHz →          4B 30 30 33 44 30 39 30 30 or "K003D0900"
"k" Command:     Set Standby N using Frequency of 1100000kHz →          6B 30 30 31 30 43 38 45 30 or "k0010C8E0"
"C" Command:     Set Active N using Channel (Channel 0 at 1000 MHz) →          43 E8 03 00 00 or "C000003E8"
"c" Command:     Set Standby N using Channel (Channel 1 at 1000 MHz) →          63 E7 03 00 00 or "c000003E8"
"r" Command[1]:          →          72
"f" Command[2]:          →          66

1.  "r" command must be preceded by "R" command to permanently reset factory setting for 10 MHz reference.
2.  "f" must be preceded by either "K", "D" or "C" Commands to change the factory pre-set start frequency.

7

**TIMING DIAGRAM**



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

Revisions: