From February 2010 *High Frequency Electronics*

Control Interfaces for RF and Microwave Frequency Synthesizers

By Alexander Chenakin
Phase Matrix, Inc.

Here is a thorough overview of the many options for the external control of a frequency synthesizer, or any other programmable RF/microwave device.

Frequency synthesizers come in a variety of forms ranging from tiny PLL chips and moderate-size modules to bench-top signal generators [1-5]. Single-chip synthesizers are available in a die form or as surface-mount integrated circuits. They include key elements (such as RF and reference dividers, phase detector, lock indicator, etc.) required to build a simple single-loop PLL synthesizer. More complex ICs include a built-in VCO, multiple PLLs, DDS, and other valuable components integrated on a single chip. Such ICs are installed on a printed-circuit board (PCB) with additional circuitry (e.g., loop filter components). The PCB-based modules range from small, surface-mount, "oscillator-like" designs to more complex connectorized assemblies. The level of complexity varies from simple single-loop PLLs to sophisticated multiloop and DDS-based designs. Such PCB assemblies can be packaged into a metal housing and are presented as stand-alone, complete synthesizer modules.

Connectorized synthesizer modules (often called "bricks") can be used to build larger bench-top and rack-mountable signal generators for test-and-measurement applications. They come with high-end technical characteristics, precise calibration, and extended functionality including frequency and power sweep, various modulation modes, built-in modulation sources, and many other functions. The synthesizers come with various control interfaces that are almost as diverse as the number of synthesizer designs. The most popular interfaces are reviewed next; more details can be found in [6].

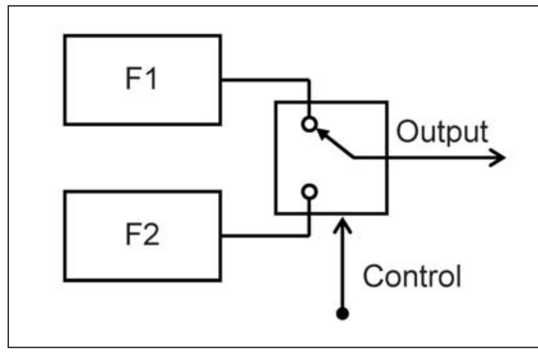

Figure 1 · Simplest dual-frequency synthesizer example.

Parallel Interface

The control interface is an electrical link that provides connection and data exchange between two (or more) devices such as a frequency synthesizer and host controller. The parallel interface assumes transmitting and receiving control signals over multiple wires at one time. The number of wires heavily depends on the number of functions to be controlled. To illustrate this, let's consider a hypothetically simplest dual-frequency synthesizer example shown in Figure 1. Two different fixed frequencies are generated in blocks F1 and F2 respectively and the frequency change is performed with an electronic switch. Only a single control line is required to switch between two frequencies. If we need more output frequencies, more switches and more control lines must be added. Using a binary code, we can control as many as 2^n frequencies, where n is the number of control lines (in

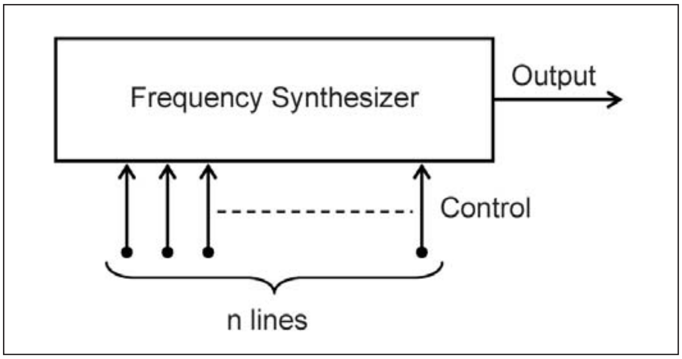

Figure 2 · Parallel interface.

addition to a ground connection), as illustrated in Figure 2. Alternatively, we can use a binary-coded decimal (BCD) control, which may be more convenient to set a frequency with decimal digits but requires a higher number of control lines. Besides setting the frequency, we may also need to control output power and other synthesizer functions that require even more control lines. Furthermore, the interface can also include some extra auxiliary signals such as a lock indicator, trigger, etc. Hence, the number of control wires grows with the design complexity.

The main advantage of the parallel interface is high communication speed since all control signals are sent simultaneously. While data transmission in parallel is very fast, it usually requires many control lines, bulky connectors on both sides, and a complex, multi-wire connecting cable.

SPI

Serial peripheral interface (SPI) is a synchronous serial data link introduced by Motorola, Inc. that offers full duplex communication, relatively high throughput, and flexibility. The idea behind the SPI is to send controlling bits via a single line; one-by-one rather than all together. Another line is added to receive some information from the device under control. In order to synchronize the data streams, an auxiliary synchronization signal (such as clock pulses) is needed. And finally, we may want to control not one but several devices via the same wires. This is accomplished using an additional, auxiliary line that allows the selection of a particular device. Thus, a multi-device, full-duplex interface can be physically constructed with four signal lines as depicted in Figure 3. The controlling device is called master, and the device under control is called slave. The control lines are asserted to carry the following functions:

SCLK—Serial CLocK—is used for synchronization of data streams
MOSI—Master Output, Slave Input—is used to stream data from the master device to the slave

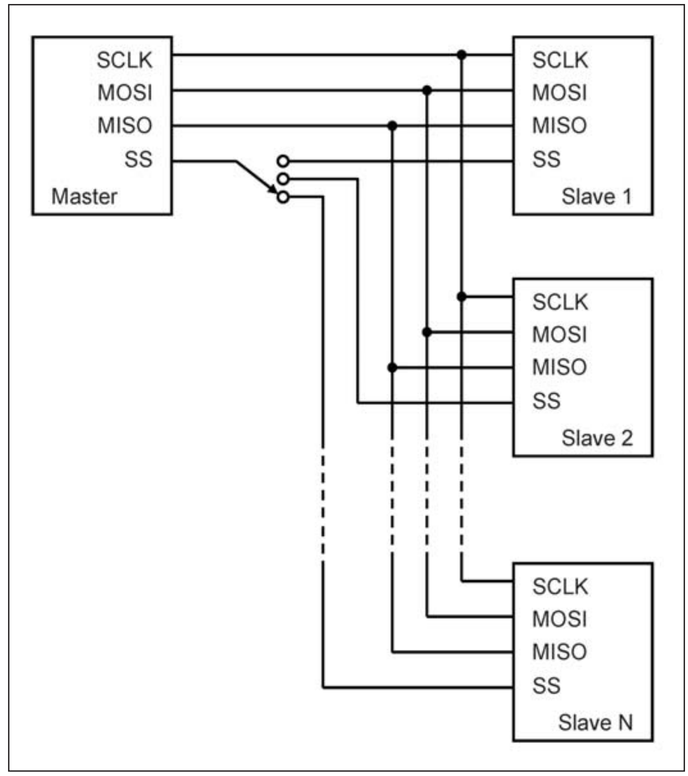

Figure 3 · The SPI interface utilizes four signal lines.

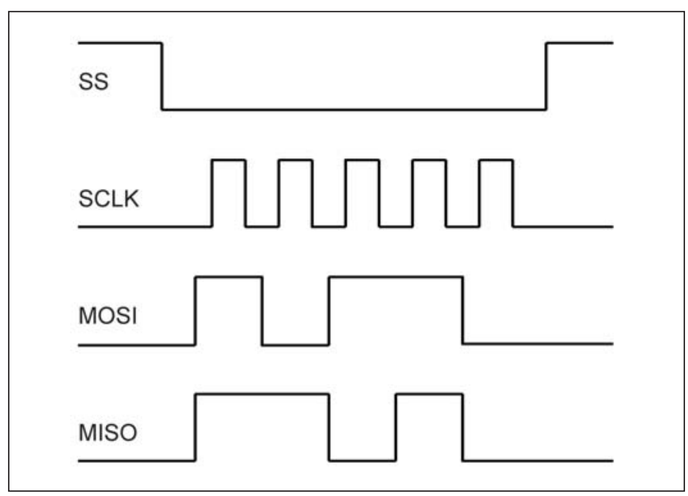

Figure 4 · SPI communication timing diagram.

MISO—Master Input, Slave Output—is used to stream data from the slave device to the master
SS—Slave Select—is used to select a particular slave device

The communication is initiated by the master that sets the SS signal low for a desired slave device as shown in Figure 4. If only a single slave device is used in the system, the SS signal is not necessarily required and in many cases may be set to ground. With multiple slave devices, however, an independent SS signal is required from the master for each slave device; thus, only one slave may be chosen at a time. After selecting a slave, the master starts streaming the data through the MOSI line, simultaneously providing clock pulses on the SCLK line. The SCLK is aligned with MOSI in such a way that the slave device processes the data bit-by-bit with every clock pulse. Other slave devices that have not been chosen disregard the SCLK and MOSI signals. Besides, they must not drive the common MISO line. Most slave devices have an internal switch that disconnects or puts into a high-impedance state their MISO output when the device is not selected, thus, allowing multiple devices to share the same line.

To better understand the communication process, let's consider the slave device input as a shift register, which is essentially a cascade of flip-flops, sharing the same clock as shown in Figure 5. A signal on the register's data input line is transferred to the first flip-flop output on the rising (or falling) edge of the clock signal. With the second clock pulse, this signal is further transferred to the output of the second flip-flop, etc. Thus, the series of data bits is shifted down and appears on the corresponding flip-flop outputs. In other words, the register converts an input data stream from serial to parallel format. Similarly, a parallel controlling word on the transmitter side can be converted to a serial format and delivered to the receiving device with a minimal number of physical connections between the devices.

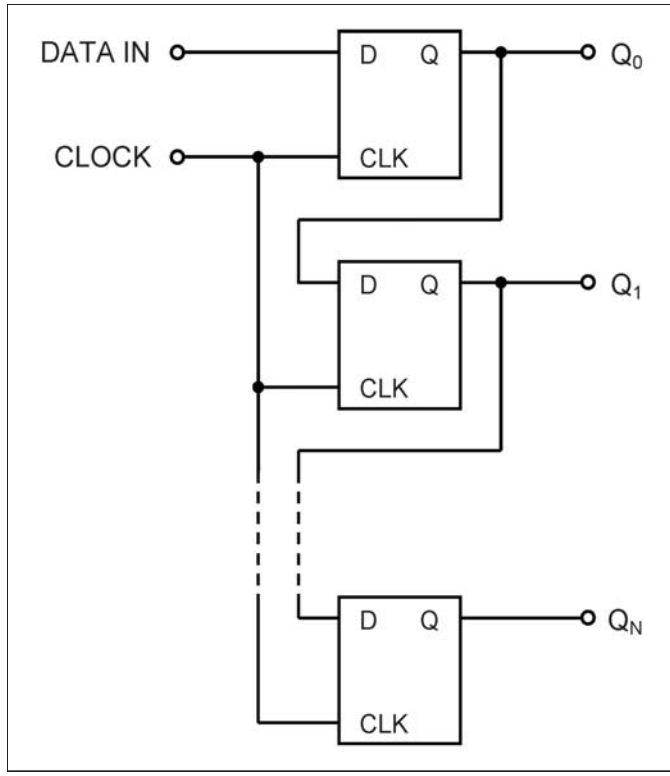

Figure 5 · A shift register allows conversion of a serial data stream into a parallel format required for device control.

Although the concept seems straightforward, a number of SPI modifications exist because of the lack of a strict standard. Each device is described by its own specifications including maximum clock rate, timing characteristics, number of bits and their definitions, polarity of control signals, etc. Moreover, the MOSI and MISO signals are sometimes combined together into a common data line. The MISO signal is often omitted entirely, which allows programming the slave device but not reading information from it. This SPI modification is called "3-Wire" in contrast to the normal four-wire arrangement and is widely used in PLL synthesizer chips. Overall, the SPI interface is extensively used in both IC and module-level synthesizer designs to allow small packages and highly integrated functionality.

I^2C

The I^2C interface was introduced by Philips Semiconductors in the early 1980s. The name I^2C translates into "Inter IC" since the idea was to allow easy communication between components residing on the same circuit board. Currently I^2C is used not only within a single board, but also may be used to connect separate devices using a cable.

Each device connected to the bus is software addressable by a unique address. I^2C is a multi-master bus, meaning that multiple masters can initiate data transfer over the shared bus. The main advantage of the I^2C interface is its simplicity. Only two bidirectional lines (serial data and serial clock) are required for communication. Disadvantages include relatively low communication speeds and the lack of automatic bus configuration.

RS-232

RS-232 is another serial interface that, for a long time, was the primary standard for computer serial ports. The standard was introduced by the Electrical Industries Association and evolved from the need to connect electromechanical teletypewriters to modern electronic devices and personal computers. While the standard recommends a 25-pin connector, 9-pin connectors are common, and a three-wire arrangement is often used when the full capabilities of RS-232 are not required. In the latter case, communication is established via the transmit-data, receive-data, and ground pins. RS-232 can be a good choice if the synthesizer needs to be controlled from a personal computer. Its main disadvantage is relatively low speed. As a result, it is being replaced by much faster USB and Ethernet connections.

Figure 6 · A USB interface allows instant deployment and evaluation of frequency synthesizer modules.

Figure 7 · A VXI signal generator covers the 0.01 to 20 GHz frequency range.

USB

Universal serial bus (USB) is today's most popular way of connecting various devices to a personal computer. Compared to RS-232, USB is faster, smaller, and simpler to use. The current USB version 2.0 provides up to 480 Mbit/second of data transfer and will be replaced with an even faster USB 3.0 rated to 5 Gbit/second. USB also supports plug-and-play connectivity, meaning that devices are detected by the computer's operating system and configured automatically as soon as they are attached. USB cables can be up to 30 meters long and can also be used to bias relatively low-power devices. These features make USB a very desirable option in the design of frequency synthesizer modules since it allows instant deployment or simply evaluation of a synthesizer using a personal computer as illustrated in Figure 6.

Ethernet

Ethernet is another well-known interface that enables communication through local area networks (LANs). It was developed at Xerox PARC in the 1970s and is currently standardized by the Institute of Electrical and Electronic Engineers (IEEE) under IEEE 802.3. The interface assumes the sharing of a common connection among several devices. Communication is carried out by sending data packets (i.e., blocks of data) between the devices connected to a network, with each block going to a specific destination device. The Ethernet interface is also utilized within LXI platform, which will be reviewed later.

GPIB

General Purpose Interface Bus (GPIB) is a special interface for test-and-measurement applications. It was originally introduced by Hewlett Packard (now Agilent Technologies, Inc.) as the HPIB bus to control measurement instruments. In 1975, the interface was standardized by the IEEE under the IEEE-488 standard. The GPIB bus has 24 lines including eight signal lines used for data transfer, three for handshake, five for bus management, and eight ground returns. It allows connection of multiple off-the-shelf instruments into a complex automated test system.

VXI

Based on the 1970s-era VME bus developed for computer control systems, VXI stands for VME eXtensions for Instrumentation. VXI was introduced in the mid-1980s as an open system platform for synthetic instrumentation. One of the principles behind synthetic instrumentation in general, and VXI in particular, is to offer a cost-efficient modular approach for building complex test-and-measurement equipment. It enables the emulation of various traditional bench-top instruments employed in automatic test systems using a reconfigurable combination of core hardware modules.

A VXI instrument includes a chassis (also called mainframe) that contains several spaces (slots) where individual VXI modules can be installed. The mainframe also contains all necessary DC power supplies and provides communication between individual components and a host controller, which is usually an external computer—although controllers in VXI modules are also used. A module (such as a signal generator shown in Figure 7) fits into one or more slots in the chassis and connects through the VXI bus that delivers all necessary power, communications and bias lines.

The VXI specifications are governed by the VXI Bus Consortium, which was founded in 1987 by a group of interested companies to define mechanical, electrical, and software features of VXI instrumentation.

PXI

PXI stands for PCI eXtensions for Instrumentation and is a further enhancement of the synthetic instrumentation concept (PCI stands for Peripheral Component Interconnect). The PXI standard was introduced by National Instruments Corporation in 1997 and is currently governed by the PXI Systems Alliance (PXISA). The alliance includes more than 50 companies chartered to promote the standard, ensure interoperability, and maintain PXI specifications.

Similar to VXI, a typical PXI instrument is built using a PXI chassis and a number of individual modules that fit into PXI slots (such as a

Figure 8 · A 3 to 9 GHz synthesizer module available in PXI form.

synthesizer module shown in Figure 8). However, the size of the chassis and the modules is significantly smaller; a typical PXI module measures approximately 4 by 6 inches in dimensions. Moreover, the host computer can be built as a PXI component and plugged into the chassis. Therefore, a whole instrument or even an ATE system can be completed within a single PXI frame. Another distinct advantage is higher communication speed compared to the VXI environment. It should be noted that the PXI chassis backplane uses essentially the same PCI bus used in personal computers. Thus, the development and operation of PXI systems is not much different from that of standard Windows-based applications. A newer PXI Express standard (released in 2005) further increases the available PXI bandwidth by taking advantage of PCI Express technology. Users benefit from significantly increased bandwidth, ensured backward compatibility, and additional timing and synchronization features.

LXI

LXI stands for LAN eXtensions for Instrumentation and is another interface for test-and-measurement applications. It was introduced in 2004 by Agilent Technologies, Inc. and is currently maintained by the LXI Consortium. The LXI concept offers integration advantages of modular instruments without the constraints of card-cage architectures. It is based on a well-established Ethernet protocol that allows connecting individual instruments into a network. LXI can be used at any level of network complexity ranging from a single component and a controlling computer to complex multi-instrument systems operated remotely through the Internet.

The LXI standard defines three classes of instruments. The base class C incorporates a Web browser via an Ethernet port as well as IVI (Interchangeable Virtual Instrument) driver. Class B brings synchronization capability via the IEEE 1588 precision time protocol and also supports peer-to-peer messaging. The IEEE 1588 protocol synchronizes clocks in multiple devices to ensure proper event time stamping and execution of synchronized events. Finally, Class A adds a fast hardware trigger bus, which offers lower-latency synchronization compared to the Class B.

AXIe

AXIe (Advanced TCA Extensions for Instrumentation and test) is a recent addition to the synthetic instrumentation interfaces that supports both PXI and LXI standards (TCA stands for Telecommunications Computing Architecture). It is governed by the AXIe Consortium that was formed in 2009 by Agilent

Technologies, Inc., Aeroflex Corporation, and Test Evolution Corporation. AXIe addresses a wide range of ATE systems, rack-and-stack modular, bench-top, and module plug-ins. It offers higher performance per rack inch, greater scalability, more flexibility, and easy integration with various platforms.

References

1. V. Kroupa, *Frequency Synthesis: Theory, Design and Applications,* New York: John Wiley & Sons, 1973.

2. V. Manassewitsch, *Frequency Synthesizers: Theory and Design,* 3rd ed., New York: John Wiley & Sons, 2005.

3. V. Reinhardt, et al., "A Short Survey of Frequency Synthesizer Techniques," *Proc. 40th Annual Symposium on Frequency Control,* May 1986, pp. 355-365.

4. Z. Galani and R. Campbell, "An Overview of Frequency Synthesizers for Radars," *IEEE Transactions on Microwave Theory and Techniques,* Vol. 39, No. 5, May 1991, pp. 782-790.

5. A. Chenakin, "Frequency Synthesis: Current Solutions and New Trends," *Microwave Journal,* May 2007, pp. 256-266.

6. A. Chenakin, *Frequency Synthesizers: Concept to Product,* Norwood, MA: Artech House, 2010.

Author Information

Dr. Alexander Chenakin is the vice president of the Signal Sources Group at Phase Matrix, Inc. He has led the development of advanced products for Celeritek, Nextek, Micro Lambda Wireless, General Electronic Devices, and other companies. In 2005 Dr. Chenakin joined Phase Matrix, Inc. where he oversees the development of advanced frequency synthesizer products for test and measurement applications. His professional achievements have been widely presented in trade publications and international conferences. Dr. Chenakin is a senior IEEE member and was invited speaker for several IEEE-sponsored events. He can be reached by phone at 408-954-6409 or by e-mail at achenakin@phasematrix.com. The company web site is: www.phasematrix.com

Reminder

All past technical articles are available for download in the Archives section of our web site:

www.highfrequencyelectronics.com