A look at the MIPI RFFE standard

By Victor Wilkerson
Director - Digital Design
WiSpry Inc.

and

Gernot Hueber
Head of RF Innovations
DICE GmbH & Co KG / Infineon

MIPI RF Front-End Control Interface (RFFE) is the specification of a bus interface specifically tailored for the needs of current and future mobile wireless systems to control the slave devices in an RF front-end. This article gives an overview of MIPI RFFE Version 1.0 and discusses the desires and rationale behind the development of RFFE, including a survey of existing alternatives.

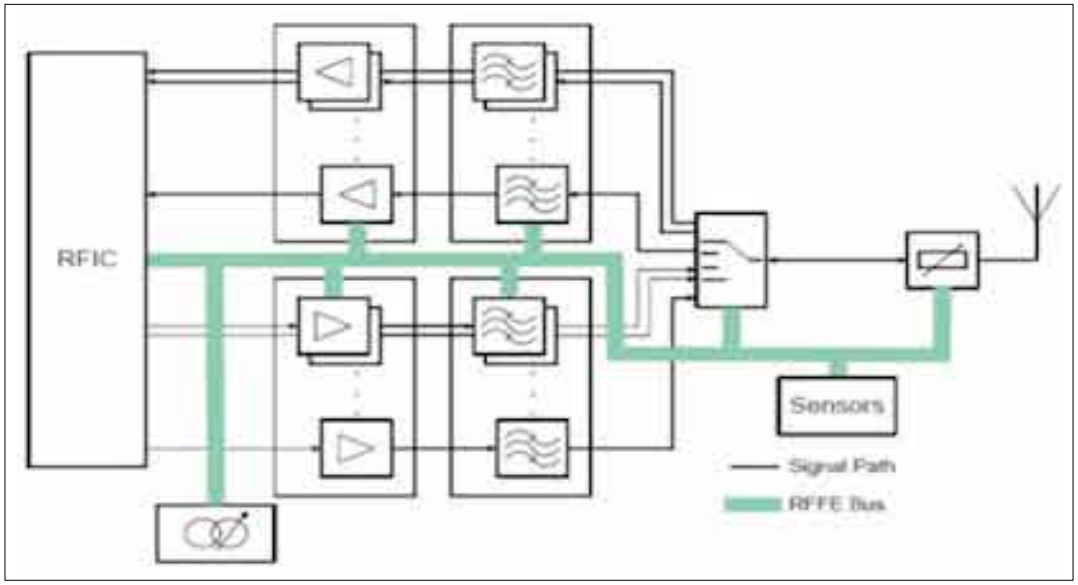

Figure 1: Example architecture for an RFFE bus-controlled front-end.

Mobile radio platforms

During the last decade, the number of mobile phones has grown rapidly, with 1.4 billiondevices shipped in 2010, and with no saturation of demand in sight. Over that same time, the complexity has evolved from voice-only 2G devices, then to 3G and, most recently, 4G multifunctional smartphones. In addition, cellular communication complexity has increased with connectivity for Wi-Fi, Bluetooth, GPS, FM radio, and other capabilities.

The addition of these wireless standards creates a need for multiradio solutions which cover ten or more frequency bands. As a result, the number of devices required for RF front-ends has also increased. For phone manufacturers, the complexities of control for all these devices becomes an ever-mounting challenge.

The MIPI Alliance Specification for RFFE seeks to address this challenge by providing a bus interface connecting the transceiver, or radio, to the myriad of RF front-end devices such as LNAs, PAs, antenna switches, antenna tuners, DC/DC converters, filters, sensors, and the like. These devices may be controlled by an RFIC, or, alternately, some other source. This commentary provides a brief overview of the features of the RFFE spec which makes it uniquely suitable for the tasks presented by this burgeoning environment.

RF front-end design

Multi-band and multiradio support is a necessity for nearly all cellular mobiles. New spectrum allocations for global wireless communications are being defined, or redefined, by the relevant authorities. Historically, spectrum allocations have been country-specific, and thus globally inconsistent, since they are typically controlled by local governments. Unfortunately, spectrum alignment has been driven by industry economies of scale, rather than foresight and global cooperation among governing bodies. Without spectrum alignment, device standardization is extremely difficult to achieve. With the wireless communications industry a relative newcomer to spectrum allocation, the disparate frequency bands will continue to follow localized needs rather than broader, global opportunities for optimization.

As a result, carriers require support for their selected frequency bands, and some selected roaming bands, while the terminal suppliers prefer to offer a minimum number of different terminal designs in order to maximize volumes and efficiencies. The results in the need to increase the number of bands supported. An additional outcome of this is that the band usage is not necessary Radio Access Technology (RAT) dependent, but it does require that radios need to be able to identify the various RATs used, and to act accordingly.

Carriers also expect that radio performance will constantly be improved, despite these ever-increasing demands. Apart from higher throughput there are demands for improvements in other areas, such as better sensitivity to increase cell coverage and improved interference rejection for better connection quality. Combining multiRAT and multi–band with ever-increased performance is a serious design challenge. But because it is extremely difficult to provide improvements in all of these areas with existing solutions, radio suppliers have typically resorted to less integrated solutions. Thus this inherent tendency results in less integration, rather than more.

The mobile device industry also is addressing the need for increased performance and functionality by introducing new types of components for use in RF front-ends. Many of these new components represent new technologies, and which may increase the component count in the RF front-end, even though these components support numerous functions. However, these components still require dedicated parallel signals paths, with a commensurate need to control all of these paths separately. For example a power amplifier (PA) module may incorporate multiple PAs and signal chains, and thus may require increased control capabilities.

Another factor driving performance improvements is the need to correct for the imperfections caused by analog components. Previous approaches pre-distorted the signal, but there is a trend toward more complex solutions, incorporating closed loops and real-time tuning. Examples are closed loop polar transmitters and antenna tuners, which introduce completely new components and functions into the RF front-end.

It is evident that the increasing complexity to control the RF front-end will require an expanded intelligence level in order to manage numerous signal chains and increased complexities. The RFFE control bus has been developed to meet these challenges (**figure 1**).

The MIPI Alliance RFFE Specification is designed to be the channel for time-critical information within the RF front end while meeting requirements typical to mobile device applications. One of the major objectives of MIPI RFFE specification is to provide device compatibility ensuring device interoperability devices from any vendor. With the current myriad of arrangements for RF front-end control, it is extremely difficult for systems designers to select the appropriate devices and to utilize a common control scheme. And for device providers, the ability to support all the desired control schemes implies producing either a variety of devices, or a complex multiinterface product.

Another objective of the RFFE specification is to satisfy a demand for low pin count, both at the master (RFIC), and in particular for the slave devices, thus saving costly I/Os at the package level and area for traces on the PCB by using a bus system to which all devices are connected. Real time programming and high-speed, low latency command transfer are an imperative for timing accurate control and the provisioning of all devices in any high dynamic use case. It is also important to provide functions for the cases where acquisition of status from front-end devices is required.

Because RF front-end devices may vary both with regards to their indicated needs, as well as their ability to support control complexity, it was important to provide a range of optional features within RFFE. These features allow for compact silicon implementations ranging from simple devices supporting only a minimum of commands and functionality to more complex devices with efficient command sequences. The specification also supports scalability and power-down modes as an enabler for achieving minimum current consumption in systems. This was another major objective in the development of the RFFE Specification.

The MIPI RFFE specification is designed to replace existing control solutions, such as SPI interfaces (a three-wire bus with proprietary vendor -specific protocol layer), which suffer from a lack of standardization due to the variety of telegram structures despite the commonality afforded by similar signal structure. Dedicated analog or digital control lines also lack common definition, and consume a large number of pins and PCB area for the traces since they typically do not provide for multidevice support, and so must be replicated for each new device. Other potential candidates, such as I2C, have never been an acceptable solution in sensitive wireless applications, since they lack some of the fundamental features desired.

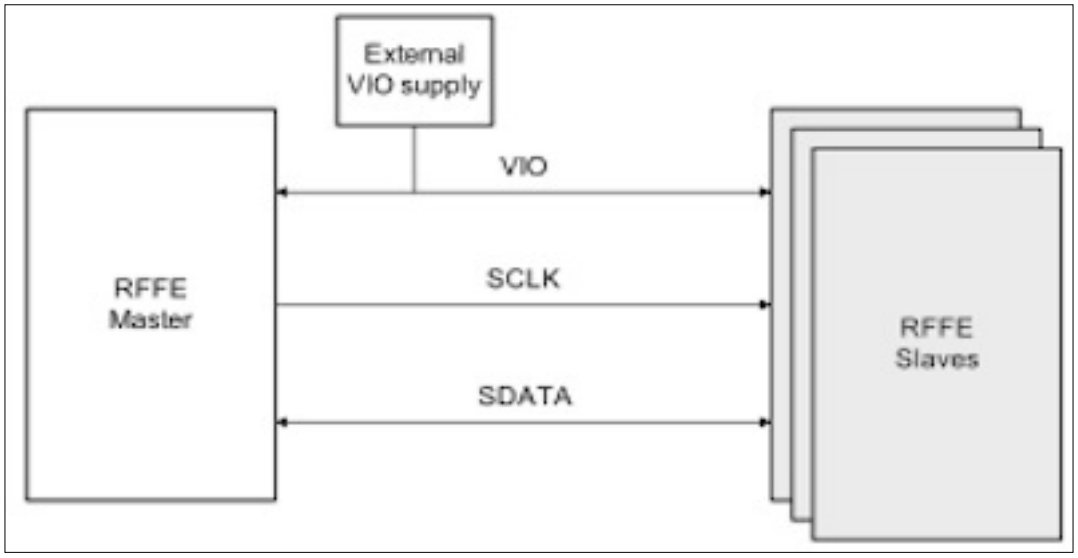

Figure 2: High level depiction of an RFFE system.

RFFE specification

The MIPI RFFE Specification defines an interface between RFFE-capable devices, with one master device and up to 15 slaves on a single RFFE bus. It uses two signal lines, a clock signal (SCLK) controlled by the master, a unidirectional/bidirectional data signal (SDATA), and an I/O supply/reference voltage (VIO). The choice of SDATA attribute is based on whether a slave device is write-only, or whether it supports read/write capability. RFFE bus components are connected in parallel to the SCLK and SDATA lines of the bus. Line drivers always exist for both SCLK and SDATA in the master, whereas only slaves supporting readback functionality need a line driver for SDATA. Each physical slave must have one SCLK input pin, one SDATA input or bidirectional pin, and a VIO pin to ensure signal compatibility between devices. **Figure 2** shows a representative arrangement of devices in an RFFE system. Note that VIO can be supplied externally, as shown, or it may be sourced from the master device.

The master drives the RFFE clock signal. The maximum operating frequency for SCLK is 26 MHz, although lower rates down to 32 kHz may be used in accordance with the RFFE Specification. In idle or inactive periods, SCLK does not toggle and is held at a logic zero.

In order to achieve efficient interface implementations, several measures have been realized in the RFFE specification. A low gate count for the interface implementation of the RFFE Slaves is achieved by optional command sequences on the slave giving the implementer the choice to use a minimum set of sequences fitting to the need and capability of the front-end device. Moreover, readback is an optional feature with buffer strengths. Thus area and current consumption are significantly relaxed by allowing readback transmissions to occur at half the normal interface clock rate.

RFFE supports up to fifteen logical slaves on a single bus. A master may host more than one RFFE bus, so the limit on RF front-end slaves is effectively unlimited. Users may elect to engage multiple RFFE buses, even though each one may have less than 15 slaves. This may be done for performance reasons, or for some other consideration; examples might be for interference mitigation, or to provide physical separation.

The protocol messages, which in RFFE are called command sequences, consist of three parts: a sequence start condition (SSC), the command frame followed by some number of data frames, and finally a bus park cycle, which closes the command sequence.

RFFE defines a variety of command sequences to accomplish read and write accesses to slaves on the bus, with the primary differences being the amount of addressable space available, and the size of payload data which may be transferred within a single command sequence. Supported payload sizes range from 7 bits to 16 bytes within a single command sequence. The amount of memory which may be addressed by the different command types ranges from 1 byte up to 64 kbytes, and a slave may possess anywhere from 0 bytes to 64 kbytes. **Figure 3** shows the memory ranges for various command types in RFFE.

As an example of the command sequences provided by RFFE, **figure 4** shows a "Register Write" command sequence, which may be used for writing 8 bits of data to a register identified by bits A4-A0. The slave device which

will respond to a particular command sequence is identified by a slave address, which is located in bits SA3-SA0 of the command sequence. Data fields in RFFE are assumed to be multiples of bytes, so data frames consist of 8 bits of data plus a parity bit. Address frames are also 9 bits in length, with 8 bits of address information plus a parity bit. RFFE Command frames always consist of 12 bits, and a parity bit. Note that each Frame in a command sequence carries a parity bit with odd parity calculated over that frame. Depending on the command sequence type, the number of address and data frames within a message may vary.

SCLK and SDATA are CMOS-like signals, i.e. single-ended, ground referenced, rail-to-rail, voltage mode signals. The electrical reference for both signals is relative to the I/O reference voltage, VIO. The SCLK and SDATA terminals use the same signaling levels. RFFE components must support either 1.2 V or 1.8 V for operation but they may support both voltage levels.

In order to keep the RF EMI aggressor contribution of RFFE devices to a minimum, all signals must conform to a range of signal slew rates which are defined in the specification. The RFFE master may also suspend all RFFE traffic during critical periods by holding off SCLK and lowering VIO to shut off slave interfaces. The EMI susceptibility of RFFE devices is minimized by stipulating a minimum hysteresis for all device inputs.

RFFE slaves support several operating modes, including startup, active state, shutdown, and optionally, a low-power mode. The startup mode defines a specific set of actions used to initialise any RFFE device before it becomes active. The active state is the normal operating mode for an RFFE system. The low-lower mode allows a Slave to put defined device-specific registers in a low-power state for the duration of this mode. In addition to these modes implementers are allowed to define other operating modes for their devices, provided

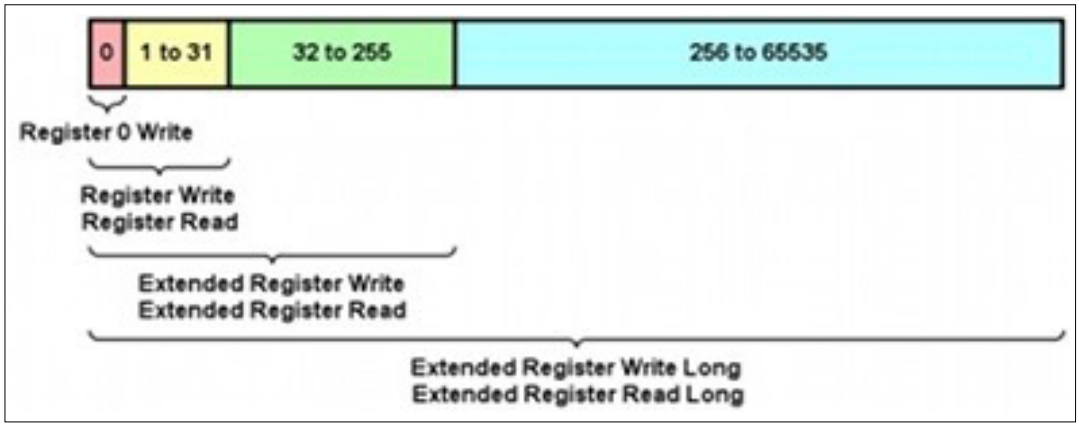

Figure 3: RFFE slave address space.

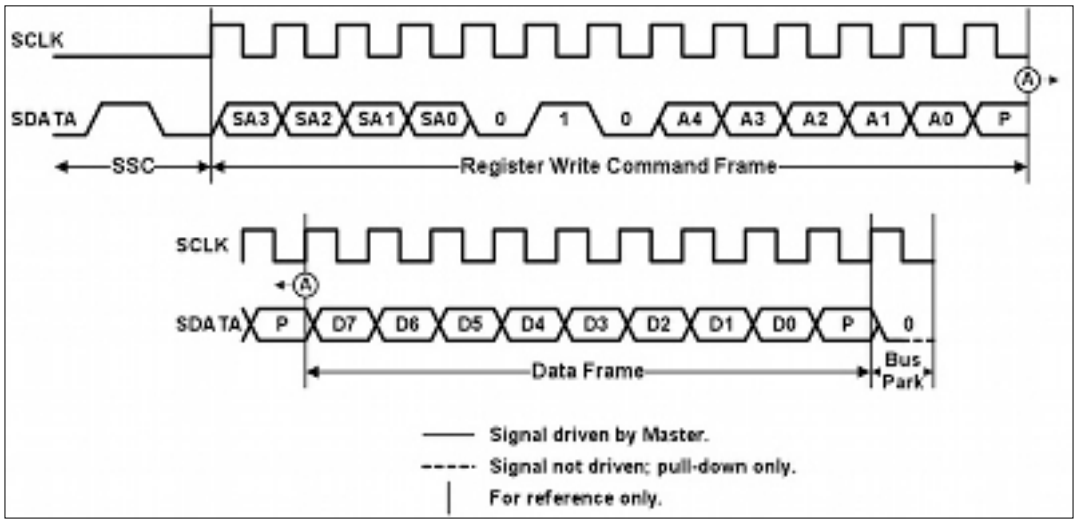

Figure 4: Register write command sequence.

that these proprietary modes are defined in such a manner that the RFFE-defined modes, and transitions, are retained.

Group trigger functionality is an essential feature in RFFE to allow one or more devices to meet precise timing requirements, especially if there is a risk of bus congestion using ordinary messages. Triggers provide a tool to help solve instantaneous bandwidth limitations, and to achieve simultaneous precise timing at multiple destinations. RFFE triggers may be used if multiple registers in the same device, or in multiple devices, are required to be loaded at exactly the same time or if the timing of the system requires multiple registers to be loaded, or specific actions to take place, within a timing-critical window of time.

The RFFE Specification was the joint effort of cellular handset suppliers, front-end device providers, and RFIC vendors. The following companies have contributed to the RFFE Specification: Agilent Technologies Inc., Broadcom, Fujitsu Limited, Infineon Technologies AG, LG Electronics Inc., LnK, Motorola, National Semiconductor, Nokia Corporation, Panasonic Corporation, Peregrine Semiconductor, Qualcomm Incorporated, Research In Motion Ltd, Renesas Technology Corp., RF Micro Devices, Rohde & Schwarz Gmbh & Co KG, Skyworks Solutions, Inc., ST-Ericsson, Synopsys, Inc.,Texas Instruments, Inc., and WiSpry, Inc.

The release of the RFFE Specification, and the accompanying Application Note, provides access to these documents for all MIPI Alliance member companies. There has been significant interest generated by the release of these documents from a large number of companies. It is anticipated that the availability of RFFE-compliant devices on the market will occur in the very near future.

Next steps

In July 2010, the RFFE Specification achieved final approval for the first release, v1.00. At the same time the RFFE Working Group released v1.00 of an RFFE Application Note, v1.00. The Application Note also includes an FAQ section with responses for the most commonly received questions regarding the RFFE interface.

The MIPI RFFE Working Group is currently nearing completion of a Protocol Implementation Conformance Statement (PICS) for RFFE, which is intended to aid in the ability for implementers to ensure device conformance to requirements of the RFFE Specification. At the same time the Working Group has begun the process of identifying and defining future extensions to RFFE. These may include better support for highly complex RF front-end topologies like MIMO and dual-RFIC solutions, and to provide for Slave initiated sequences on the bus, such as interrupts.