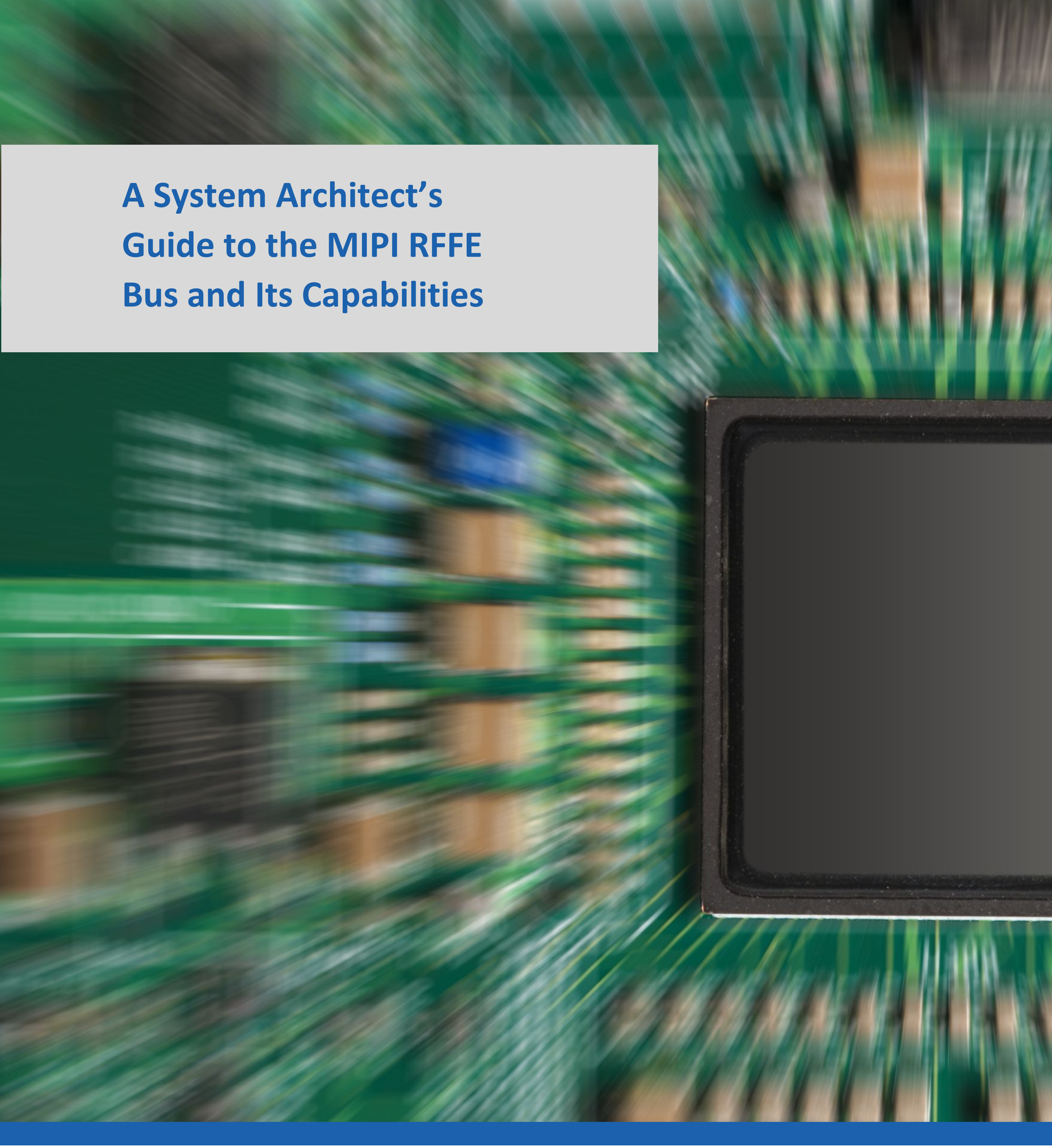
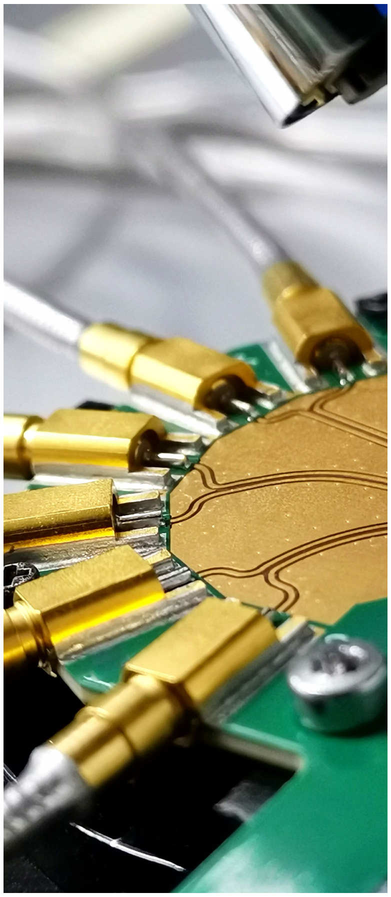
A System Architect's Guide to the MIPI RFFE Bus and Its Capabilities



SIGNAL CRAFT TECHNOLOGIES



Introduction

With the proliferation of smartphones, smartwatches, and tablets in our society, instant and unhindered access to communication, information, and entertainment has become an expectation in our daily lives. Fast wireless network technologies such as LTE, WIFI, and Bluetooth have made it possible for numerous products and applications to maintain on-demand connections to the Internet virtually anywhere the consumer wants.

Emerging radio technologies such as 5G cellular and 802.11ax will push download speeds into the Gigabit-per-second range and latencies into single-digit milliseconds, but supporting these – as well as the full gamut of existing radio standards – ultimately results in highly complex radio architectures. For system engineers, overcoming these challenges is particularly important as their products will target consumers who demand the latest in wireless technology but are unwilling to on sacrifice product size, performance, or cost.

To address some of these market pressures, the MIPI RF Front-End Control Working group was formed to develop a highly efficient but flexible control interface capable of addressing current and future market needs. Working group membership consists of representatives from multitude of leading wireless industry companies.

MIPI RFFE Scope and Purpose

The primary focus of the MIPI Radio Frequency Front-End (RFFE) protocol is to provide optimal RF front-end connectivity and control. While developing the standard, the MIPI RFFE working group placed an emphasis on increasing data throughput, all while achieving a reduction in the power consumption, PCB space, and component cost necessary to support a highly sophisticated RF front-end. As an industry supported standard, RFFE seeks to minimize technology fragmentation of RF front-end interfaces and promotes interoperability of all RF front-end components across the industry.

As an industry standard, RFFE seeks to minimize technology fragmentation of RF frond-end interfaces and promotes interoperability of all components across the industry.

The MIPI RFFE bus supports virtually all types of front-end devices, including low-noise amplifiers (LNAs), RF switches, power amplifiers (PAs), filters, antenna tuners, power management modules, and sensors.

Key Features of RFFE

This section presents a high-level overview of the key features of an RFFE compatible bus. Further details can be obtained in the MIPI RFFE specifications.

Enhanced Data Throughput

Bi-directional serial data throughput similar to SPI speeds can be achieved with the RFFE bus. The latest specifications support data rates up to 52 Mb/s for write and 26 Mb/s for read. The lowest supported clock rate is 32kb/s.

A side benefit of increased data throughput is that Master and Slave devices can interact with each other much quicker and then enter into a low power state sooner, leading to improved battery life in portable wireless products.

Simple 2-Wire Interface

MIPI RFFE utilizes a physically simple interface consisting of just two wires (plus V_{IO}), a bi-directional SDATA and SCLK, which is sourced by the bus Master. Both SDATA and SCLK are single-ended, ground referenced, rail-to-rail, voltage mode signals. Data is transferred across the bus starting with the Most Significant Bit (MSB) and ending with the Least Significant Bit (LSB)

To reduce active power consumption, both SCLK and SDATA are operated with unterminated end-points. On complex RF front-ends with numerous devices, a common two-wire interface eliminates the need for multitude of additional inputs and outputs (IOs) that would be required as SPI chip selects, interrupts, or dedicated control IOs, thus easing PCB wiring complexity and minimizing device pin counts.

Large Address Space

The MIPI RFFE specification supports a large 16-bit address space, providing ample capacity to control even the most complex of radio front-end architectures. Slave registers are byte-addressable by the Master. Register Write/Read Command Sequences support five address bits or a total of thirty-two addressable registers. The Extended Register Write/Read Command Sequences support eight address bits to access up to 256 addressable registers. The optional Extended Register Write/Read Long Command Sequences have 16 address bits to access up to 65536 register locations.

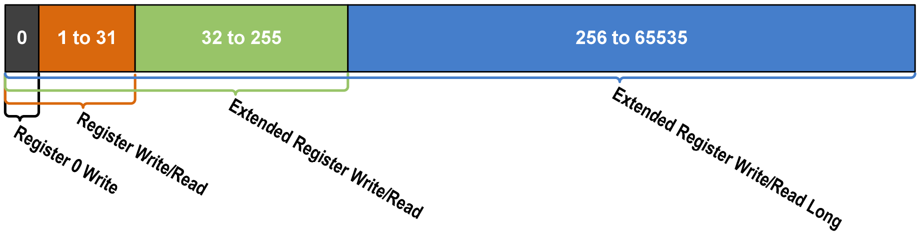


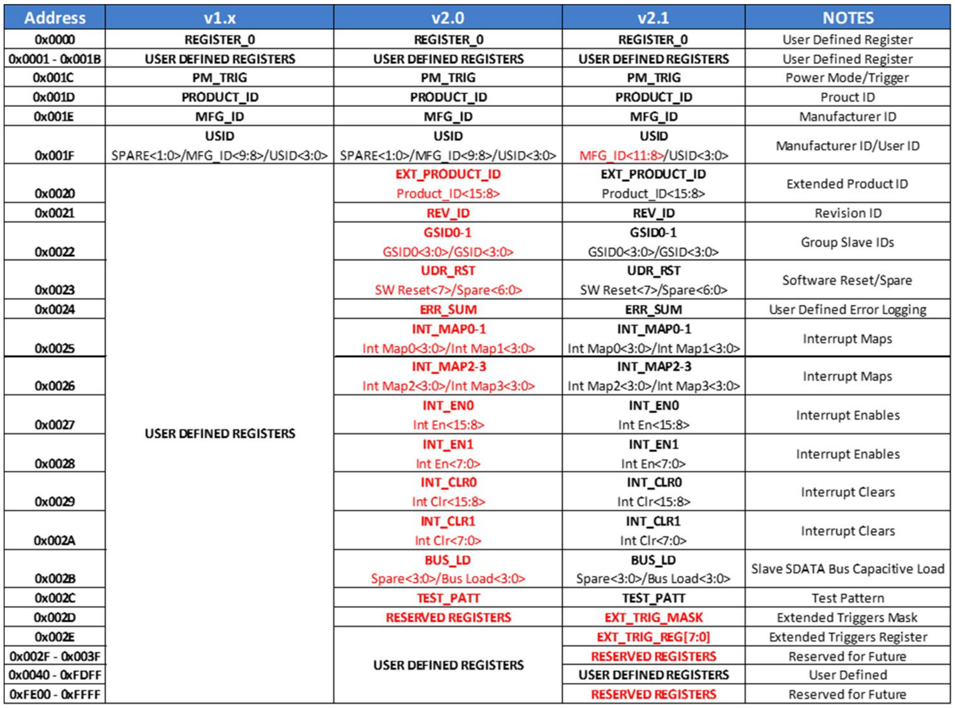
Figure 1. Addressing (source: MIPI RFFE Specification 2.0)

Slave Register Types

Extended Register Write/Read Command Sequences have 16 address bits to access up to 65536 register locations.

RFFE's working group has carefully thought out and defined a standardized register space that all RFFE compliant Slaves must adhere to. Doing so ensures interoperability of all RFFE Masters and Slaves, both in terms of the hardware and the software programming used to control devices within the RF front-end.

Far from being stagnant, the RFFE specification has and will continue to evolve to support the ever-increasing sophistication of wireless product technologies. The following table documents the historical evolution of RFFE Slave register definitions.



Address	v1.x	v2.0	v2.1	NOTES
0x0000	REGISTER_0	REGISTER_0	REGISTER_0	User Defined Register
0x0001 - 0x001B	USER DEFINED REGISTERS	USER DEFINED REGISTERS	USER DEFINED REGISTERS	User Defined Register
0x001C	PM_TRIG	PM_TRIG	PM_TRIG	Power Mode/Trigger
0x001D	PRODUCT_ID	PRODUCT_ID	PRODUCT_ID	Prouct ID
0x001E	MFG_ID	MFG_ID	MFG_ID	Manufacturer ID
0x001F	USID SPARE<1:0>/MFG_ID<9:8>/USID<3:0>	USID SPARE<1:0>/MFG_ID<9:8>/USID<3:0>	USID MFG_ID<11:8>/USID<3:0>	Manufacturer ID/User ID
0x0020	USER DEFINED REGISTERS	EXT_PRODUCT_ID Product_ID<15:8>	EXT_PRODUCT_ID Product_ID<15:8>	Extended Product ID
0x0021		REV_ID	REV_ID	Revision ID
0x0022		GSID0-1 GSID0<3:0>/GSID<3:0>	GSID0-1 GSID0<3:0>/GSID<3:0>	Group Slave IDs
0x0023		UDR_RST SW Reset<7>/Spare<6:0>	UDR_RST SW Reset<7>/Spare<6:0>	Software Reset/Spare
0x0024		ERR_SUM	ERR_SUM	User Defined Error Logging
0x0025		INT_MAP0-1 Int Map0<3:0>/Int Map1<3:0>	INT_MAP0-1 Int Map0<3:0>/Int Map1<3:0>	Interrupt Maps
0x0026		INT_MAP2-3 Int Map2<3:0>/Int Map3<3:0>	INT_MAP2-3 Int Map2<3:0>/Int Map3<3:0>	Interrupt Maps
0x0027		INT_EN0 Int En<15:8>	INT_EN0 Int En<15:8>	Interrupt Enables
0x0028		INT_EN1 Int En<7:0>	INT_EN1 Int En<7:0>	Interrupt Enables
0x0029		INT_CLR0 Int Clr<15:8>	INT_CLR0 Int Clr<15:8>	Interrupt Clears
0x002A		INT_CLR1 Int Clr<7:0>	INT_CLR1 Int Clr<7:0>	Interrupt Clears
0x002B		BUS_LD Spare<3:0>/Bus Load<3:0>	BUS_LD Spare<3:0>/Bus Load<3:0>	Slave SDATA Bus Capacitive Load
0x002C		TEST_PATT	TEST_PATT	Test Pattern
0x002D		RESERVED REGISTERS	EXT_TRIG_MASK	Extended Triggers Mask
0x002E		USER DEFINED REGISTERS	EXT_TRIG_REG[7:0]	Extended Triggers Register
0x002F - 0x003F			RESERVED REGISTERS	Reserved for Future
0x0040 - 0xFDFF			USER DEFINED REGISTERS	User Defined
0xFE00 - 0xFFFF			RESERVED REGISTERS	Reserved for Future

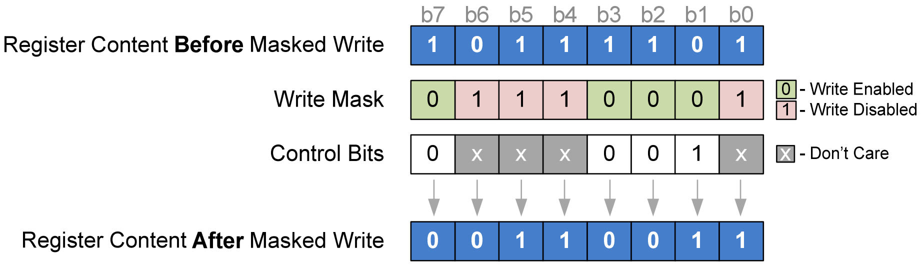
Figure 2. Evolution of RFFE Slave register definitions
(source: MIPI RFFE Standard v1.x, v2.0, v2.1)

Masked Write

Masked Write is a new feature that was defined in v2.0 of the specification. Masked Write allows for specific bits within a register to be individually updated. This capability enhances control of registers where bits are shared amongst more than one control function. What previously required a multi-command (read/write) sequence can now be implemented with a single Masked Write command.

Only User-Defined Registers respond to a Masked Write Command Sequence and use of Masked Writes should be restricted to multi-function registers.

Masked Write allows for specific bits within a register to be individually updated.



	b7	b6	b5	b4	b3	b2	b1	b0
Register Content **Before** Masked Write	1	0	1	1	1	1	0	1
Write Mask	0	1	1	1	0	0	0	1
Control Bits	0	x	x	x	0	0	1	x
Register Content **After** Masked Write	0	0	1	1	0	0	1	1

Figure 3. Masked Register Write Functionality
(Source: MIPI RFFE Specification v2.1)

As an industry standard, RFFE seeks to minimize technology fragmentation of RF frond-end interfaces and promotes interoperatbility of all components across the industry.

Flexibility and Extensibility

The RFFE bus provides a great deal of system control flexibility through its wide range of optional features. Each RFFE bus instance allows for up to 15 Slave devices due to bus load capacitance limitations. However, extensibility to greater than 15 Slaves in a system is possible with additional bus instances.

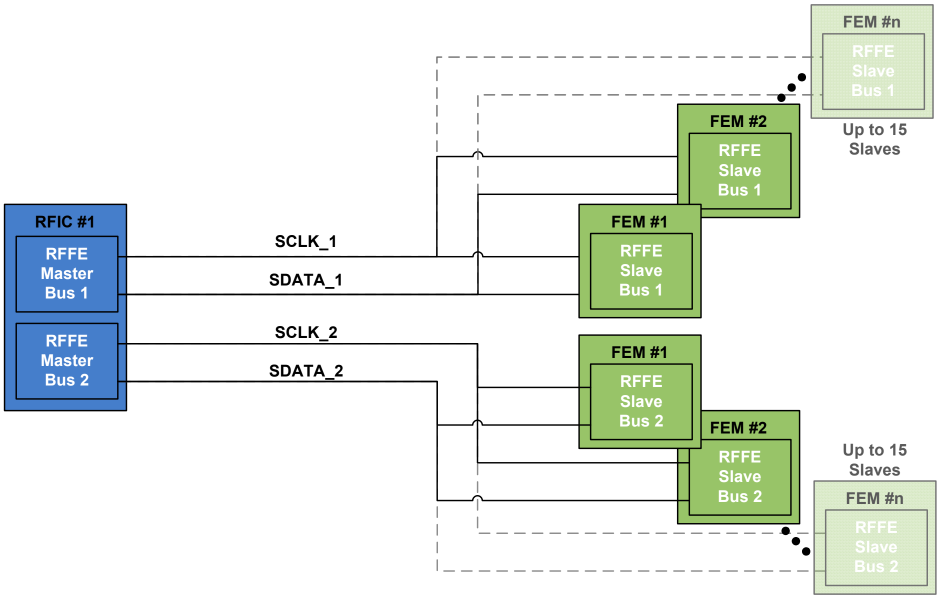


Figure 4. RFFE Extensibility Through Multiple Buses
(Source: MIPI RFFE Specification v2.1)

Error Detection and Recovery

To ensure data integrity, an MIPI RFFE bus utilizes parity bits to detect data error events. When an error is detected, the Master and Slave must react according to well-defined set of procedures in section 6.3 of the MIPI RFFE specification.

In-Band Slave Interrupts eliminate the need for dedicated Slave pins while maintaining near real-time response.

Interrupt Capable Slaves

Support for Interrupt-Capable Slaves (ICS) was introduced in v2.0 of the MIPI RFFE specification. In-band Interrupts eliminate the need for dedicated Slave pins to signal interrupts while maintaining near “real-time” service response. The physical simplicity of the bus is a result of the Slave’s ability to autonomously indicate service requests as messages within the 2-wire bus. The specification allocates a unique time period as an interrupt slot for a Slave. A total of 16 interrupts slots are available on the bus which are dynamically assigned to Slaves during bus configuration. A maximum of four interrupt slots can be assigned to any given Slave.

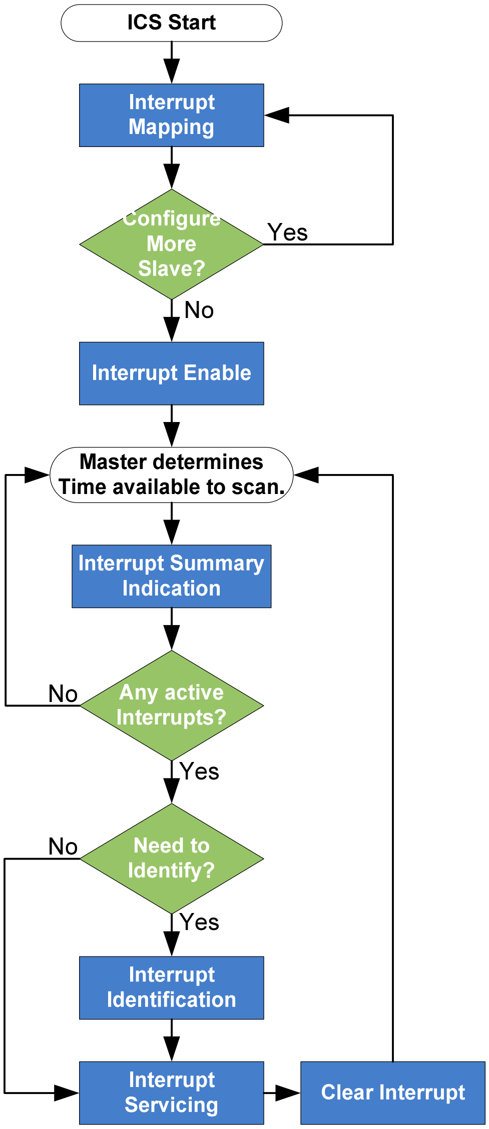


Figure 5. Interrupt Handling Flow Chart (Source: MIPI RFFE Specification v2.1)

With triggers, the Master can queue up numerous values in Slave shadow registers when convenient and then latch those values into the Slave's operational registers precisely when needed by the system.

Triggers

Triggers provide a means for programming values to be simultaneously loaded into several different register locations. These registers may coexist within a single Slave, but may also span multiple Slaves on the same bus. Trigger functionality is essential to providing deterministic control RF front-end devices across a system.

Triggers can be an effective tool in overcoming bandwidth limitations. The Master can queue up numerous programming values in Slave shadow registers when convenient and then latch those values into the Slave's operational registers precisely when needed by the system.

MIPI RFFE v2.1 introduced the concept of extended triggers, increasing the total number of assignable triggers from three to eleven. This capability significantly raised the bar on the level of sophistication that can be applied in controlling complex RF front-ends.

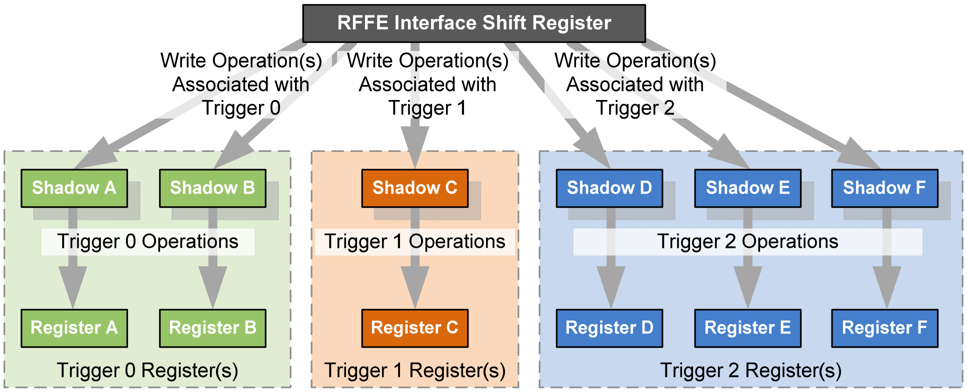


Figure 6. Illustration of RFFE Trigger Functionality
(Source: MIPI RFFE Specification v2.1)

Broadcast Messages

Broadcast messages are a feature of RFFE that enables concurrent writing of command sequences to multiple Slaves. This feature leverages either a specific Group Slave ID (GSID) or the dedicated Broadcast Slave ID (BSID), Address 0b0000. The BSID targets all Slaves on the bus. GSIDs are 4-bit identifiers that the Master can assign to any number of Slaves on the bus during bus configuration.

Common Interface Voltage

Use of the RFFE dictates that all RF Slave devices utilize a common interface reference voltage that is sourced by the bus Master. This restriction reduces pin and net wiring requirements, simplifies system power requirements, and leads to a lower product cost.

The RFFE specification defines a multitude of different operating voltages that can be used and won’t restrict applications where it may be advantageous to use a certain bus reference voltage.

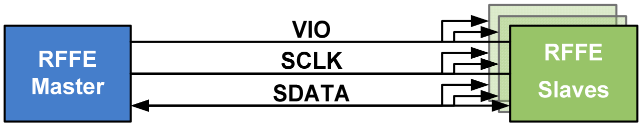


Figure 7. RFFE Common IO Voltage Source
(Source: MIPI RFFE Specification v2.1)

Electromagnetic Compatibility Optimization

The RFFE specification has taken into consideration the need to meet various regulatory limits for electromagnetic interference (EMI). To help mitigate emissions, the bus was designed with optimized slew rates.

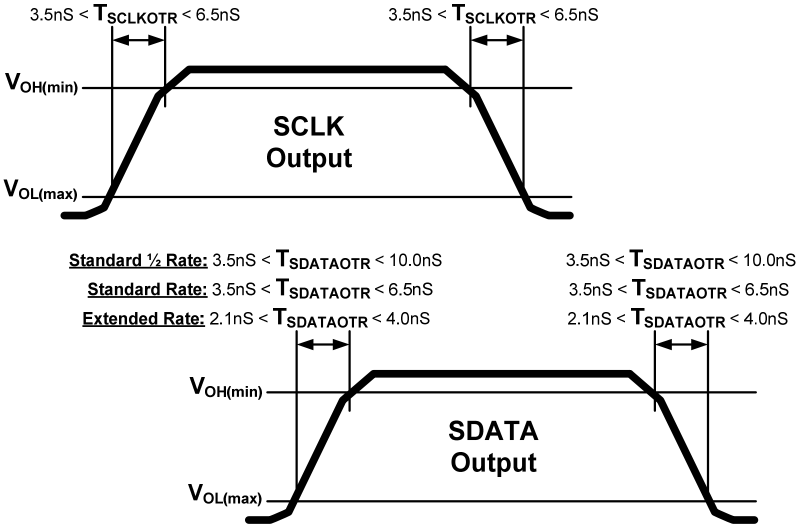

3.5nS < $T_{SCLKOTR}$ < 6.5nS
3.5nS < $T_{SCLKOTR}$ < 6.5nS
$V_{OH(min)}$
SCLK
Output
$V_{OL(max)}$
Standard ½ Rate: 3.5nS < $T_{SDATAOTR}$ < 10.0nS
Standard Rate: 3.5nS < $T_{SDATAOTR}$ < 6.5nS
Extended Rate: 2.1nS < $T_{SDATAOTR}$ < 4.0nS
3.5nS < $T_{SDATAOTR}$ < 10.0nS
3.5nS < $T_{SDATAOTR}$ < 6.5nS
2.1nS < $T_{SDATAOTR}$ < 4.0nS
$V_{OH(min)}$
SDATA
Output
$V_{OL(max)}$

Figure 8. EMI Friendly Controlled Data Slew Rates
(Source: MIPI RFFE Specification v2.1)

MIPI RFFE also takes EMI susceptibility into consideration. External emissions sources can potentially couple onto the bus nets and lead to signal integrity issues. RFFE buses utilize CMOS gates that incorporate hysteresis to increase immunity against unwanted sources of interference.

Multi-master support allows for up to four different masters to arbitrate control of the bus.

Multi-Master Support

Version 2.0 MIPI RFFE introduced provisions for multi-master (MM) support on the bus. This new capability allows for up to four different Masters to arbitrate control of the bus and share utilization of some or all of the RF front-end devices in the system.

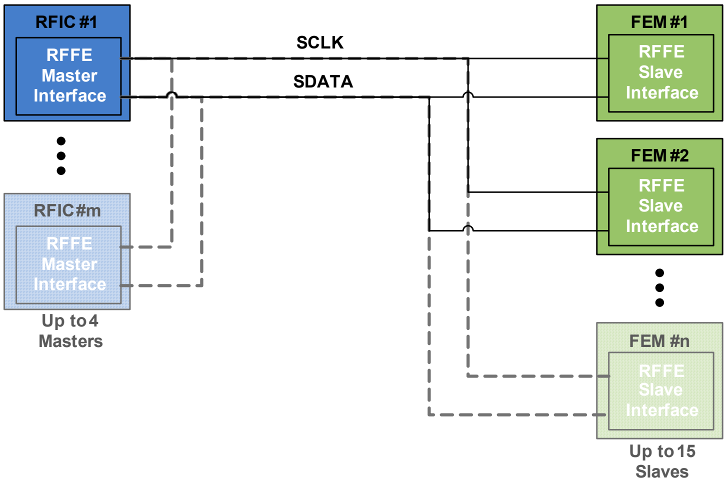


Figure 9. Illustration of Multi-Master Architectures
(Source: MIPI RFFE Specification v2.1)

Evolution of the RFFE Standard

v1.0 – Initial Release (July 2010)

v1.1 – Release Update (Nov 2011)

- Enhanced definitions of several requirements from v1.0 specification

v2.0 – Major Update (Dec 2014)

- Multi-master (supports Carrier Aggregation)
- Extension of bus operating frequencies
- Synchronous read
- Slave interrupt support
- Additional registers for unified control (new reserved registers and functions)
- Common function register locations for ease of HW and SW development
- V1.x backward compatibility

v2.1 – Current Version (Apr 2018)

- Support for longer trace lengths up to 45 cm as opposed to the previous 15 cm – allows bus support for products that are physically larger, such as laptops
- Master context transfer (MCT) RFFE command sequence
- Extended triggers – additional assignable triggers increased from 3 to 11
- Masked write command sequence – the master can control individual aspects of slave's programmable content, which simplifies what previously could only be achieved with a multi-command sequence down to a single command
- RFFE over M.2 connector

v3.0 – Future Release

- Improved throughput and reduced latency
- Enhanced timing capability to support needs of 5G devices

Far from being static, RFFE must address current industry requirements, while continuously evolving to meet future market needs.

Interfacing to MIPI RFFE Devices

Figure 10. SCT SC4415 RFFE Controller

The MIPI RFFE standard offers numerous performance enhancements over legacy methods of controlling RF front-end devices, but these benefits do come at a cost – a substantially more sophisticated electrical interface and protocol. Implementation of this interface could prove daunting; however, these complexities need not deter engineers seeking to leverage the benefits of incorporating RFFE-enabled devices in their system. Evaluation of RFFE prototype designs or production test systems can easily be streamlined with the SCOUT Serial Bus Controller from SCT. SCOUT is the highest performing, lowest cost USB-to-RFFE serial bus adapter on the market.

SCOUT automates the complex and timing critical tasks.

It was design with determinism, low latency, and fast execution in mind.

SCOUT automates the complex and timing critical protocol tasks. Its physical interfaces have been designed with simplicity and flexibility in mind. Software-selectable internal supplies can support logic levels for +1.2V, +1.8V, +2.5V, and +3.3V bus operation. Applications requiring custom interface voltages between +0.8V to +3.6V can also be supported with an external voltage source.

SCOUT was designed with determinism, low latency, and fast execution in mind. Timing critical performance is obtained using an FPGA architecture and user-configurable command cues. Predefined command sequences and trigger modes enable multiple executions of each sequences using either software or hardware triggers.

Common applications include 3rd party component evaluation, system integration, troubleshooting, and manufacturing test automation. A SCOUT SC4415 will give true RFFE compliant control your Slave devices faster than any other solution on the market.

SignalCraft Technologies is working closely with customers and MIPI to provide protocol upgrades and support as the RFFE standard evolves and new RFFE-compliant RF front-end devices emerge on the market.

Learn More About SCOUT

https://www.signalcraft.com/products/test-measurement/serial-bus-controllers/

About SignalCraft Technologies

We build brilliantly designed, high frequency digital and RF products, 100% in-house from the ground up to your specs and schedule. From leading global test brands to industrial communications startups, technical leaders trust SignalCraft as their wireless product development partner.

signalcraft.com