Computer Communications 209 (2023) 178–187

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

Improved TDD operation on Software-Defined Radio platforms towards future wireless standards[☆]

Thijs Havinga [*], Xianjun Jiao, Muhammad Aslam, Wei Liu, Ingrid Moerman

IDLab, Department of Information Technology at Ghent University - imec, Technologiepark-Zwijnaarde 126, B-9052 Ghent, Belgium

ARTICLE INFO

Keywords:
Software-Defined Radio
Self-interference
TDD
Turnaround time
System-on-chip

ABSTRACT

Software-Defined Radio (SDR) platforms are valuable for research and development activities or high-end systems that demand flexible wireless protocols. While low-latency digital baseband processing can be achieved using a dedicated processing unit, like an FPGA or hardware accelerator, its multi-purpose Radio Frequency (RF) front-end often poses a limitation. Zero Intermediate Frequency (ZIF) transceivers are favorable for SDR, however, even for Time Division Duplex (TDD) systems, these transceivers suffer from self-interference when the transmitting and receiving Local Oscillator (LO) is set to the same frequency. To achieve low self-interference, switching from receiving to transmitting mode is needed. However, the time this takes (turnaround time, TT) for configurable RF front-ends often violates the strict timing requirements of protocols like Wi-Fi and 5G, which require response times in the order of microseconds. In this work, we first evaluate the advantages and disadvantages of several methods to suppress self-interference of a ZIF transceiver. Next, a novel approach is proposed, which can reduce the TT to as low as 640 ns using the widely used AD9361 configurable ZIF RF front-end, while the noise floor is at the same level as achieved by the conventional way of switching between transmit and receive mode. We have realized and validated this approach using *openwifi* — an open-source Wi-Fi implementation on SDR. As a result, the receiver sensitivity is improved by up to 17 dB in the 2.4 GHz band and 9.5 dB in the 5 GHz band, for over-the-air transmissions.

1. Introduction

Software-Defined Radio (SDR) platforms enable researchers to experiment and prototype innovative wireless communication solutions more freely as compared to Commercial Off-The-Shelf (COTS) chips thanks to their flexibility and openness. Open-source projects that make use of SDR, such as OpenAirInterface [1], srsRAN [2] (previously srsLTE) and openwifi [3], have recently gained worldwide attention. Apart from providing an implementation of existing wireless standards, many innovations towards future wireless standards are realized on top of it. For example, [4] presents a new physical layer approach to optimize spectrum sharing based on srsLTE. On top of openwifi, [5] randomizes channel state information to preserve privacy, and [6] provides high-precision time synchronization. As these solutions require modifications to the physical layer implementation, they could not have been realized using COTS chips.

An SDR generally consists of two parts. The first part, a processing unit, performs the digital signal processing operations, such as modulating and demodulating. Next, an SDR has a configurable Radio Frequency (RF) front-end, which consists amongst others of a Digital-to-Analog Converter (DAC) and Analog-to-Digital Converter (ADC), an amplifier, mixer and several filters. The processing unit can be a separate host computer by using its Central Processing Unit (CPU) that runs the algorithms as programmed by the designer. Another option is to use a Field Programmable Gate Array (FPGA), in which the digital hardware itself is configured by the designer. The latter is especially interesting for experimenting with systems requiring a high data rate, e.g. for future 6G communications [7].

Currently, one of the downsides of SDR using a host computer is the latency induced by the data transfer between the radio front-end and the processing unit [8]. For example, according to [9], the latency between a host computer and a Universal Software Radio Peripheral (USRP) X310 using a PCIe link is 79 μs. For applications that require low latency, an SDR based on a System-on-Chip (SoC) is more suitable, as the connection between CPU and FPGA is faster. As shown in [9], the connection between CPU and FPGA on an SoC can be as low as 1.435 μs. Various works [6,10–12] require the low latency aspect.

For the implementation of Time Division Duplex (TDD) systems, a radio needs to switch from the receiving (Rx) to the transmitting (Tx) mode (and vice versa). The time this takes is called the turnaround time (TT). The efficiency of a Medium Access Control (MAC) protocol

[☆] This research was partially funded by the Flemish FWO SBO #S003921N VERI-END.com project.

[*] Corresponding author.

E-mail addresses: Thijs.Havinga@UGent.be (T. Havinga), Xianjun.Jiao@UGent.be (X. Jiao), Muhammad.Aslam@UGent.be (M. Aslam), Wei.Liu@UGent.be (W. Liu), Ingrid.Moerman@UGent.be (I. Moerman).

https://doi.org/10.1016/j.comcom.2023.06.026
Received 12 August 2022; Accepted 27 June 2023
Available online 3 July 2023

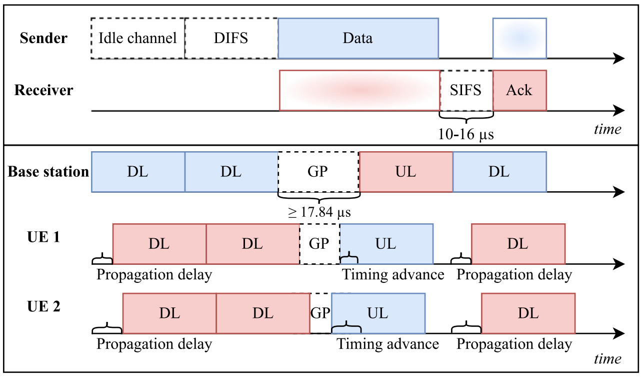


Fig. 1. TT should be within SIFS of Wi-Fi (top) and GP of 5G (bottom).

is limited by the TT, as during TT no effective data transfer or carrier sensing can take place. The mechanisms to handle TT in two representative wireless standards (i.e. Wi-Fi and 5G) are illustrated in Fig. 1. In the Wi-Fi standard, an acknowledgment frame as a response to a successfully received frame should be sent to the transmitter after a specific time, called Short InterFrame Space (SIFS). For the standards IEEE 802.11a/g/n, the SIFS is 10 µs when operating in the 2.4 GHz band and 16 µs in the 5 GHz bands [13]. The TT is only a part of the SIFS, as the PHY and MAC processing delay also contribute to the SIFS. Moreover, the slot time, which can be as low as 9 µs for IEEE 802.11n, is determined by the sum of TT, the MAC processing delay, the time for clear channel assessment and twice the air propagation time of the most distant stations (see Figure 10–21 in [13]). Thus, for the design of the MAC, the TT is assumed to be less than 9 µs. Furthermore, in any system using Carrier Sense Multiple Access (CSMA), it is important that the time between sensing an idle environment in Rx mode and actual transmitting in Tx mode is limited, to minimize collision probability.

In the 5G standard [14], a guard period (GP) is introduced to accommodate for a downlink (DL) to uplink (UL) delay, consisting of hardware switching, as well as propagation delays. Since the user equipment (UE) may be at different locations, they need to use a specific timing advance for their UL transmission in order to align the received signals at the base station, which takes up part of the guard period. For the smallest cell sizes, a GP of 2 OFDM symbols is recommended, which corresponds to 17.84 µs for a subcarrier spacing of 120 kHz (numerology 3). This thus defines the maximum value the hardware switching can take assuming a negligible propagation delay, but a non-zero propagation delay will limit the switching time even further.

The TT of chips designed for typical Wi-Fi MAC protocols is in the order of 1 µs [15]. It is evident that commercial Wi-Fi, cellular or other specialized chipsets can meet their individual protocols' TT requirement. However, an SDR platform does not have a chip optimized for one specific standard, since they are intended for prototyping vastly diverging wireless systems. Hence its front-end is more generic but less performing when it comes to TT, which to a large extent hinders the applicability of SDR in a real-life network.

In this paper, we consider Zero-Intermediate Frequency (ZIF) transceivers, more specifically the AD9361 front-end [16], which is widely used in the SDR research community, as well as in high-end low-volume products. ZIF transceivers use a local oscillator (LO) that produces a frequency equal to or close to the carrier frequency. In this way, less components are needed as compared to transceivers using an intermediate frequency, which makes it more suited for implementation on a chip. The lowest supported TT of the AD9361 is 18 µs according to the specification [17], which is too high for many existing MAC protocols and most certainly not qualified for the future standards.

To avoid the TT limitation, both Rx and Tx chain can remain active at all times, but this results in receiver sensitivity degradation due to Tx LO leakage when both chains are operating in the same frequency range. It is worth noticing that the LO leakage is present even if the system is not currently transmitting, which thus limits the performance of TDD (half-duplex) systems. Since the LO leakage exists in the same frequency range as the desired signal, it cannot be eliminated by conventional filtering. The AD9361 already provides RF DC offset and quadrature tracking calibration to mitigate phase and gain error. According to the datasheet, after these corrections, the carrier leakage of the AD9361 equals −50 dBc at 0 dB attenuation. At 2.4 GHz, the chip can achieve maximum 8 dBm output, meaning that the leakage can be as high as −42 dBm. Thus, if the leakage towards the Rx chain is not sufficiently suppressed, it can still drastically influence the receiver sensitivity. For example, the required receiver sensitivity of IEEE 802.11n for Modulation and Coding Scheme (MCS) 0 is −82 dBm and it is even as low as −101.8 dBm for 5G. For the AD9361, no information is given on Tx to Rx port isolation, since it is often dominated by the printed circuit board (PCB) layout, external structures to the chip or antenna isolation. As the AD9361 is a popular chip for modern SDR devices, the strong LO leakage has lead to a common barrier to achieve high performance TDD transceivers on SDR. Since the LO leakage is the main source of interference from the same device, it is hereafter also referred to as self-interference.

In this paper, we propose a method for self-interference-free operation of a configurable ZIF transceiver while maintaining a low turnaround time to improve TDD operation on SDR-based wireless systems. In the remainder of this paper, we first discuss the related work on this topic. Next, we explore different options to mitigate self-interference. We then show the realization of our novel method for an SDR-based Wi-Fi transceiver. Subsequently, its performance in terms of improved receiver sensitivity and transmission quality is evaluated using a professional wireless tester, after which we conclude the paper.

2. Related work

This paper builds upon early work we recently presented in [18]. Herein, we propose the general method to reduce TT and suppress LO leakage for TDD operation on the AD9361. The initial results are merely based on relative noise floor measurements. In this work, we thoroughly evaluate the performance and limitations of alternative methods to suppress the self-interference with acceptable TT. Moreover, we disclose specific operational details of the proposed solution. Furthermore, we perform receiver sensitivity and absolute noise floor tests in an anechoic chamber, and measurements related to the transmission quality.

In [19], the problem of self-interference in a ZIF transceiver for LTE-Advanced and 5G systems is addressed. Specifically, different sources of interference when using carrier aggregation are enumerated, since with this technique multiple LOs are active, causing sensitivity degradation. The self-interference mitigation techniques discussed include cross-talk prevention techniques and analog, digital or mixed-signal interference mitigation. Most of these solutions require different (analog) hardware designs. On the other hand, several digital self-interference mitigation techniques are presented in [20,21] and [22] in order to realize full-duplex systems on SDR platforms. However, this is a different problem, since in this case the Tx signal is known and can be used for cancellation, unlike the unpredictable noise from the Tx LO. [23] provides an improved In-phase/Quadrature (I/Q) imbalance algorithm to suppress the LO leakage. Although the performance is only validated in simulation, it could improve the image rejection ratio by 40 dB. As described in [19], in general the performance of digital techniques is still inferior to analog hardware solutions, especially when the required algorithmic complexity is limited.

The authors of [24] propose to use a digital intermediate frequency shift to avoid LO leakage. They shift the baseband signal by 5 MHz, such that it falls outside of their desired 5 MHz bandwidth. The in-band error between a transmitted and captured signal was measured

Table 1
An overview of the presented solutions to mitigate LO leakage on the AD9361 frontend.

Solution	Tx output power (dBm/17.5 MHz)		Advantage	Disadvantage
	2.4 GHz	5 GHz		
TDD/FDD Independent Mode	−82	−77	No self-interference.	TT is too high (≥18 μs) for Wi-Fi and 5G.
Standard FDD Mode	−62	−61	Zero TT.	Large amount of self-interference.
Offset tuning	−66	−69	Self-interference is suppressed with zero TT.	Needs additional filtering, which may lead to higher TT.
Hardware switch (USRP B210)	−82	−77	No self-interference with negligible TT.	Needs specific hardware.
RF port control	−75	−65	Self-interference is suppressed with extremely low TT.	Real-time control is needed.
LO control	−82	−77	No self-interference with low TT.	Real-time control is needed.

using the AD9361 RF front-end when transmitter and receiver were connected via a cable. In order to eliminate the I/Q imbalance image that is created by the shift, a baseband filter was needed. After this filter, the normalized mean square error was reduced by about 10 dB. Since this method cannot completely eliminate the self-interference and leads to additional latency and resources when filtering the spectral images, its performance gain is only limited.

In [25], a method to achieve zero TT is presented, which works by leaving both the Tx and Rx chain of the AD9361 RF front-end on. The switching between Tx and Rx is done in the digital baseband domain, which has separate processing units for Tx and Rx. Using a set-up with separate antennas placed exactly orthogonal, the receiver noise floor was measured to be 3.2 dB higher than using conventional TDD mode at a frequency of 2.41 GHz in a signal bandwidth of 2.6 MHz.

Implementation of Wi-Fi meeting the SIFS requirement using a ZIF transceiver has already been done in [12]. The authors use the AD9371, which has similar TT as the AD9361, to implement IEEE 802.11ac and full-duplex 802.11a/g. To meet the SIFS, their system consists of a separate Rx and Tx setup, which do not switch states, meaning that the Tx and Rx LO are always on, which will lead to severe sensitivity degradation in over-the-air tests as we show in this paper. Thus, such a setup does not represent a realistic system and will therefore not be suited for practical research or real-life deployment.

In [11], 2-3μs Rx-Tx switching delay is assumed for the calculation of a minimum interframe space for their custom MAC protocol. The authors show with their hardware platform using the AD9361 radio chip that an interframe space of 12 μs can be achieved. For this set-up, it is not clear whether both the Tx and Rx chain remain active all the time, nor do they mention the achieved receiver sensitivity.

In [26], multiple AD9361-based USRP B210 boards were used to create a 16-channel microwave measurement system. In order to limit the leakage within an individual B210, a custom shielding was designed. When transmitting a signal of 0 dBm at frequencies from 0.9 to 1.7 GHz, the leakage signal at the different RF ports could be degraded by up to 36 dB. Applying customized shielding is however not suited as a generic solution. In [27], a hardware modification is proposed to suppress leakage in duplex mode for the USRP N210, which uses a ZIF transceiver. It had mixed performance, which is why an adaptive digital filter was proposed as well, which makes use of the known transmitted signal. For best performance, the combination of the hardware modification and digital filter is needed, with which isolation of up to 35 dB between transmitter and receiver can be obtained.

In comparison with previous works, this paper proposes a solution that can fully eliminate self-interference without any hardware modification, at the cost of very limited additional digital hardware logic, targeting a TT well below 1 μs.

3. Self-interference suppression methods

This section discusses possible solutions to suppress self-interference using the widely used AD9361 RF front-end. The idea behind these solutions can in general be applied to any ZIF transceiver. For each method, we assess the accompanying TT and measure the potential Tx leakage. We provide a summary describing the advantages and disadvantages of each method in Table 1.

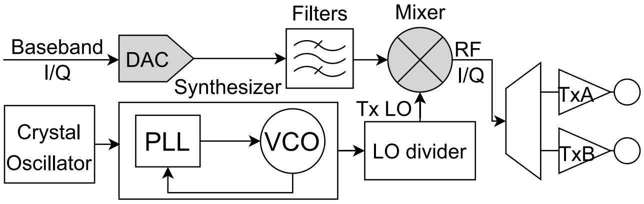


Fig. 2. A simplified diagram of the AD9361 Tx path.

3.1. *Tx-Rx switching*

A simplified diagram of the AD9361 Tx path is shown in Fig. 2. The AD9361 can operate in several modes, which affect the TT [17]. The different modes and accompanying minimum TT are described hereafter:

(1) *Standard Enable State Machine (ENSM) TDD Mode*: In standard TDD mode, Voltage-Controlled Oscillator (VCO) calibrations take place when the internal state machine changes from Rx to Tx state. The calibration time depends on the reference clock frequency, but takes at least 37 μs. After these calibrations, the Rx or Tx Phase-Locked Loop (PLL) needs to lock, which takes about 15 μs. Furthermore, when changing to Tx, the DAC needs to power up, which takes about 18 μs. Also, the Tx and Rx data paths need to be flushed, which takes 384 ADC clock cycles. E.g., for a typical ADC clock frequency of 160 MHz, this boils down to 2.4 μs. The latter two operations can be done in parallel with the PLL locking, so this method has a TT of at least 55 μs.

(2) *TDD Mode - No VCO cal*: This mode is the same as the one mentioned above, except that VCO calibrations are disabled. When the LO frequency does not change from frame to frame, it is not necessary to re-calibrate every time. Thus, the TT using this method is usually dominated by the DAC power up time of 18 μs.

(3) *TDD Mode - Dual Synth*: Like in the previous mode, no calibrations take place. *Dual Synth* means that both synthesizers in Tx and Rx path are activated at all times, such that the PLLs do not have to be locked when switching. However, the DAC still needs to power up and flushing needs to be done, resulting again in at least 18 μs delay.

(4) *FDD Independent Mode*: Similar to the mode above, both the Rx and Tx synthesizers are active in this mode. However, the data paths can be independently controlled to use arbitrary timing. This allows for enabling the Tx path already at the end of the Rx state, but this would lead to self-interference during this time. Thus, the Tx path should only be activated once the Rx path is deactivated. Like before, no VCO calibrations or DAC data path flushing takes place, but the DAC still needs time to power up, resulting in 18 μs delay.

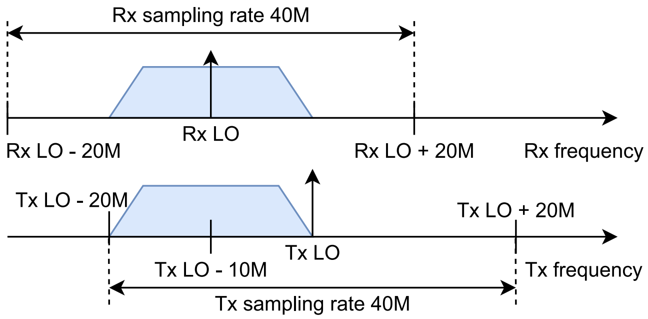


Fig. 3. Tx and Rx RF frequency with different offset to mitigate self-interference.

(5) *Standard FDD Mode*: In this mode, both the Rx and Tx synthesizer and both data paths (including DAC) remain on at all times. This means that the TT is zero.

In order to examine the potential leakage to the Rx port, the Tx output power for each of the above methods was measured using the Anritsu MS2690 A [28] spectrum analyzer. Its center frequency was set equal to the Tx LO frequency and it measures with a channel width of 17.5 MHz, which is the minimum bandwidth required to cover all the occupied sub-carriers of Wi-Fi a/g/n in 20 MHz mode. This bandwidth is chosen since noise in this bandwidth can influence the receiver sensitivity, and secondly the bandwidth should be minimum to maximize the visibility of the self-interference. The pre-amplifier was set on and the reference level was set to −60 dBm. We use the AD-FMCOMMS3-EBZ [29] front-end that contains the AD9361. It is connected via an FPGA Mezzanine Card (FMC) interface to the Xilinx Zynq UltraScale+ MPSoC ZCU102 [30].

In all of the aforementioned TDD modes and FDD independent mode, the internal state was set to Rx and no I/Q samples were streamed to the Tx path. Then in each of these modes, the power is measured to be −82 dBm/17.5 MHz and −77 dBm/17.5 MHz in the 2.4 GHz and 5 GHz band, respectively (first row in Table 1). However, the smallest delay possible using any of these options is 18 μs. Thus, none of these methods can meet the strict MAC requirements like SIFS or the guard period for 5G in higher numerologies. Therefore, operation in standard FDD mode is needed. In this mode, when setting the Tx and Rx LO on the same frequency, TDD-like operation can be achieved by selecting the respective processing modules in the digital domain, as is done in [25]. Without any further optimization, the Tx power caused by LO leakage is then measured to be −62 dBm/17.5 MHz in the 2.4 GHz and −61 dBm/17.5 MHz 5 GHz band (second row in Table 1).

3.2. Offset tuning

In FDD mode, the Tx and Rx LO can be tuned at different frequencies, similar to the digital intermediate frequency shift in [24]. The idea is to shift the Tx LO out of band to mitigate its leakage. The offset is limited by the sampling rate and analog bandwidth of the chip. For a bandwidth B, the Tx LO should be tuned at least $\frac{B}{2}$ higher or lower than the Rx LO in order for it to fall outside of the receiving bandwidth.

For example, for a Wi-Fi bandwidth of 20 MHz, this would be at least 10 MHz. The AD9361 has a maximum sampling rate of 61.44 MHz, but in order to limit the hardware resources needed to match this rate to the baseband sampling rate, the former is limited to 40 MSps. This implies that the Tx LO offset of 10 MHz is also its maximum, as shown in Fig. 3.

The Tx output power in FDD mode when utilizing this offset is measured to be −66 dBm/17.5 MHz and −69 dBm/17.5 MHz in the 2.4 GHz and 5 GHz band, respectively (third row in Table 1). This is respectively 4 and 8 dB lower than without the offset, but it is not equal to the level achieved using TDD mode, as such it still influences the receiver sensitivity.

Moreover, the frequency shift should be pre-compensated by the baseband processing unit (the FPGA in case of a SoC). However, transceivers usually have to interpolate the signal to match the digital baseband rate to the actual DAC rate due to the oversampling process required before converting it to an analog signal. This procedure generates spectral images, which results in out-of-band emission when not filtered correctly. The AD9361 has a programmable Finite Impulse Response (FIR) filter and three fixed-coefficient half-band (HB) filters between the baseband signal and DAC input. Since the HB filters are not configured to filter baseband signals with a frequency offset, the spectral images will not be suppressed sufficiently. Although the programmable FIR filter or pre-filtering in the FPGA can be applied, a large amount of taps is needed to achieve a sufficiently steep frequency response. This will introduce additional latency, which directly influences the TT. Therefore, this method seems unsuitable to eliminate the self-interference with low TT.

3.3. Hardware switching

Many widely adopted SDR boards include the AD9361 front-end, such as the popular host-controlled USRP B2xx and E3xx series [31,32]. These platforms have additional hardware switches before the RF ports, providing between 15 to 39 dB isolation. They can usually switch between Tx and Rx states in the order of nanoseconds, up to a maximum of 2.2 μs [33,34]. These switches are controlled by the on-board FPGA, based on transmit and receive tags that are added to the I/Q samples, which are streamed from the host computer. Using these tags allows for operating the AD9361 in standard FDD mode, such that the turnaround time is dominated by the inherent latency for transferring the data to and from the host computer. As discussed, this latency is much higher than required by Wi-Fi and 5G.

In order to examine the potential leakage using these hardware switches, the USRP B210 and B200mini were utilized. When operating in receiving mode, the output power of the Tx/Rx port was measured with the spectrum analyzer, whose center frequency was set equal to the Tx and Rx LO of the USRPs. We measure −82 dBm/17.5 MHz and −81 dBm/17.5 MHz in the 2.4 GHz for the B210 and B200mini, respectively. In the 5 GHz band, the Tx power equals −77 dBm/17.5 MHz for both boards (fourth row in Table 1). This is at the same level of what can be achieved using the AD-FMCOMMS3-EBZ front-end in TDD mode.

Thus, a hardware switch can indeed eliminate the self-interference. But, in order to keep the front-end suitable for multiple purposes, the switch should be digitally controllable. This is necessary e.g. when using the SDR to receive and transmit concurrently, which is the desired behavior for a full-duplex or radar system. As explained earlier, in addition to the hardware switch, better isolation might be achieved by the use of a different printed circuit board (PCB) layout. However, since modifying non-programmable hardware is not always feasible for practical reasons, it is highly desirable to have a different solution that does not require adding additional hardware switches or PCB modification.

3.4. Tx path control

Since powering the synthesizers and DAC of a transceiver requires a significant amount of settling time, these components should not be powered off when a critical TT is needed. Thus, the AD9361 should then be configured to standard FDD mode, as is done in [25]. Hereafter, we discuss two novel methods to suppress the Tx LO leakage when AD9361 is operating in this mode.

As can be seen in Fig. 2, in case of the AD9361, the final RF I/Q signal is demultiplexed into either Tx port A or B. Since only one port can be used at a time, the other is often grounded on the PCB. This is also the case for the AD-FMCOMMS3-EBZ. When operating the AD9361 in FDD mode and not actively transmitting, we can select the grounded port B to limit the Tx LO leakage. Once the control command is given, this will take effect within negligible time. We will further refer to this method as *RF port control*. In this case, the Tx output power equals −75 dBm/17.5 MHz and −65 dBm/17.5 MHz in the 2.4 GHz and 5 GHz band, respectively (fifth row in Table 1). Although the RF port control can suppress the Tx LO leakage significantly, namely by 13 dB and 4 dB in the 2.4 GHz and 5 GHz band respectively, it is not reduced up to the same level as measured in AD9361's TDD mode. This is why we look for another solution.

Following the Tx path in Fig. 2 from the antenna ports in reverse order, we first find the mixer. However, no information in the datasheet of AD9361 is provided on the time it takes to power the mixer up or down, but since it has the baseband signal (i.e. output of various filters) as input, it will need to be flushed, which is not ideal for TT optimization. Further on the path, we find the LO divider. This component makes sure that the synthesizer frequency is matched with the wanted LO frequency. For the AD9361, powering the LO divider on takes about 160 ns [35], which is well below the strict MAC requirements. Nonetheless, a way to control this component in real-time is needed. We will further refer to this method as *LO control*. Measurements using the spectrum analyzer show that the same Tx power level is achieved as in TDD mode when the LO divider of the AD9361 is powered off in FDD mode (see the sixth row in Table 1).

3.5. Summary

In total we discussed six different solutions to use a ZIF transceiver for SDR-based TDD systems, each of which has a different level of self-interference and TT. Furthermore, they come with various advantages and disadvantages. A comparison between the different solutions discussed is summarized in Table 1. Considering all these aspects, the option of LO control seems the most suited for protocols demanding a critical TT without modifying the hardware.

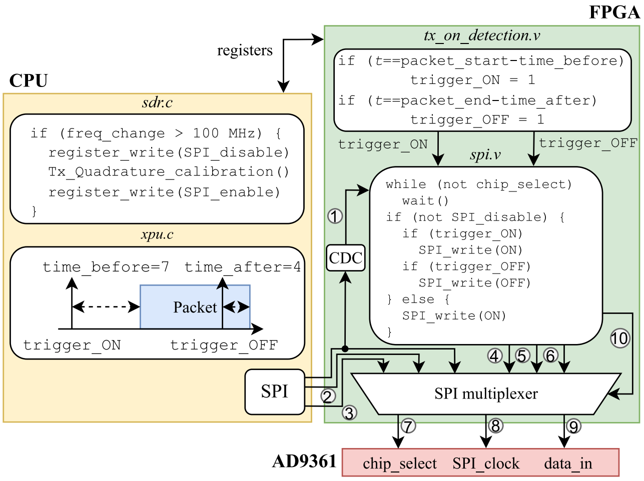
~~4. Realization~~

To show the realization of the method LO control and to assess its influence on the receiver sensitivity of a typical radio transceiver, we have implemented and integrated the proposed solution into openwifi [3]. This is an open source implementation of the Wi-Fi protocol compatible with the Linux mac80211 subsystem, suited for several SDRs based on the AD9361 chip and SoC architecture. It consists of a driver part written in the C language, which is executed on the embedded CPU, and digital hardware logic, which is implemented on the FPGA. The CPU also runs a driver that is specific for the AD9361.

4.1. Motivation and challenges

There are several challenges that need to be solved in order to be able to control the components on the RF front-end's Tx path (either the RF port or LO divider) in real-time. Firstly, the AD9361 only provides an option for controlling the components via an SPI write operation [36]. The CPU connected to the AD9361 can issue such an SPI write, but this does not ensure real-time operation due to its unpredictable response time in software. Even using a real-time operating system (RTOS), the time it takes to switch from a lower to a higher priority task with preemption can be more than 20 μs, depending on the type of CPU and RTOS [37]. This would already violate the SIFS of Wi-Fi and GP of 5G. Therefore, we propose to provide SPI functionality in the FPGA, which can be directly connected to the SPI interface of the AD9361.

Low level details regarding timing should be considered in order to optimally control the components in the Tx path at a per-packet basis. If before a transmission the control is too late, such that a sufficient part of the Wi-Fi preamble is not correctly transmitted, then a receiver cannot detect the packet anymore. Next to this, since the Tx LO phase is sensitive to noise on its power supply [16], switching the LO divider too close to the packet starting moment might influence the transmitted signal's quality. If, on the other hand, the control after the packet is too late, the receiver sensitivity would be degraded for some time, possibly leading to failed reception. Since the SPI write and LO divider powering up take some time, this should be taken into account.



```
if (freq_change > 100 MHz) {
  register_write(SPI_disable)
  Tx_Quadrature_calibration()
  register_write(SPI_enable)
}
```

```
if (t==packet_start-time_before)
      trigger_ON = 1
if (t==packet_end-time_after)
      trigger_OFF = 1
```

```
while (not chip_select)
  wait()
if (not SPI_disable) {
  if (trigger_ON)
    SPI_write(ON)
  if (trigger_OFF)
    SPI_write(OFF)
} else {
  SPI_write(ON)
}
```

Fig. 4. Hardware/software design for the proposed solution. Lines 1, 2 and 3 are the chip select, SPI clock and data lines coming from the CPU, respectively. Lines 4–6 are the corresponding SPI bus signals generated by the FPGA. Lines 7–9 are the selected lines going to the AD9361, which is controlled by line 10.

Furthermore, by adding additional control of the components in Tx path independently from the original AD9361 driver, incompatibility may arise for certain functionalities. This is the case for the Tx Quadrature calibration, which is issued by the driver when the LO frequency has changed more than 100 MHz. This would happen, for example, when the Wi-Fi transceiver scans the available channels. During Tx Quadrature calibration the Tx LO needs to be powered on, which is normally checked by the driver. Also, the calibration should be done for the RF port that is used to transmit, not for the grounded port. For either method, the incompatibility issue should thus be solved. Moreover, the AD9361 driver also uses the SPI interface via CPU to control the chip, mostly for initial configuration. Moving all of this control to the SPI module in the FPGA would require a significant amount of hardware logic. Therefore, keeping the functionality in the CPU seems more feasible. Then, care should be taken that transactions from the CPU and FPGA do not conflict with each other. In the next sections, we present our hardware/software design to address these challenges while realizing the precise control of components in the AD9361 Tx path for optimal TT, receiver sensitivity and transmission quality.

4.2. Hardware/software design

The complete hardware/software design including some pseudocode is shown in Fig. 4. The basis of our solution consists of the SPI module that can control either the RF port or the LO divider of the AD9361. It is written in Verilog and uses only 33 lookup tables and 28 registers. The source code of this module can be found online [38] in *openwifi-hw/ip/xpu/src/spi.v*.

Timing control: The trigger for the SPI transaction is given by the lower MAC logic in the FPGA, specifically in *openwifi-hw/ip/xpu/src/tx_on_detection.v*. This module has a trigger that indicates when the I/Q samples are starting to be streamed to the RF front-end. In order to fine-tune the timing control before and after a packet transmission, the moment the SPI call is initiated can be adapted via registers defined in *openwifi/driver/xpu/xpu.c*. The timing values can be set with a

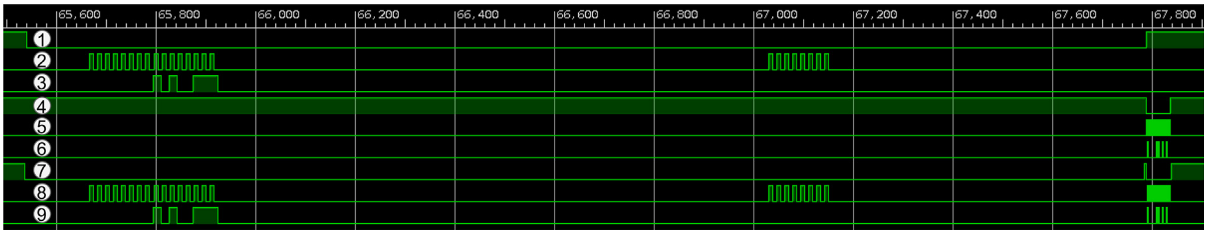


Fig. 5. ILA traces showing that the FPGA postpones its SPI write until the CPU finishes its transaction. The numbers 1–9 correspond to the lines in Fig. 4.

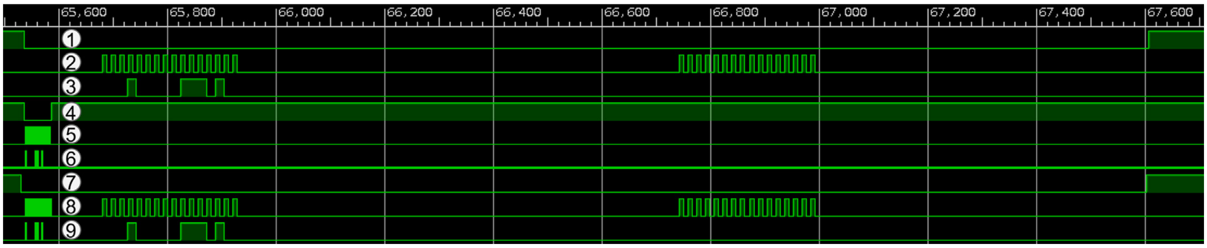


Fig. 6. ILA traces showing that the FPGA can finish its SPI write before the CPU starts its SPI clock. The numbers 1–9 correspond to the lines in Fig. 4.

granularity of 0.1 μs and are called `time_before` and `time_after`, respectively, in the pseudocode of Fig. 4. We use the maximum SPI frequency of the AD9361 (i.e. 50 MHz) and 24 bits are needed for the instruction, resulting in a duration of 480 ns for one SPI write. Since RF port control takes effect immediately, the SPI transaction is the only delay for this method. In case of LO control, the additional 160 ns power up time of the LO divider needs to be added as well.

Tx Quadrature calibration: As explained, in case RF port control is used, the right RF port should be selected during Tx Quadrature calibration. For LO control, care should be taken that the LO divider is turned on. The most robust solution to this is to set the Tx Quadrature calibration of the AD9361 driver in manual mode. In case of an LO frequency change of more than 100 MHz, the calibration procedure should be called from the openwifi driver. Before that, the SPI module should be triggered to either select the right RF port or turn the LO divider on. During calibration, the SPI module should be disabled such that it cannot control the components on the Tx path. After the calibration, it is enabled again. We provide a way to realize this procedure in the openwifi driver (*openwifi/driver/sdr.c*) and the relevant pseudocode is given in Fig. 4.

Resolving SPI conflicts: To allow both the CPU and FPGA to communicate with the AD9361 via SPI, we propose a solution that does not disturb the existing SPI functionality. To this end, the input to the chip select, SPI clock and data lines should be selected carefully. The module in the FPGA should detect whether the CPU is currently writing, which is realized by monitoring its chip select line, as shown in Fig. 4. Since the CPU and FPGA are driven by a different clock source, a clock domain crossing (CDC) circuit is needed to stabilize the signal coming into the FPGA. A typical CDC consists of two or more flip-flops coupled back-to-back. A CDC core provided by Xilinx is used in our design. Depending on the duration of the SPI transaction and the requirements of the protocol, the FPGA should postpone its writing or preempt the ongoing transaction. In the proposed SPI module, the FPGA postpones its writing until the CPU finishes its transaction (write and read). We verify the correct functionality by using an Integrated Logic Analyzer (ILA), which shows the internal logic inside the FPGA during operation. The ILA trace of when the FPGA wants to write during a CPU transaction is shown in Fig. 5. Only when the FPGA is allowed to write, its chip select line, SPI clock and data line are selected by the SPI multiplexer, as shown in Fig. 4. Otherwise, the lines from the CPU are connected to the AD9361. The worst case scenario is when the FPGA wants to write when the CPU has just started an SPI transaction. In this case, the FPGA write would be postponed for the duration of the CPU transaction, which can take a couple of microseconds. Depending on whether the FPGA write was to switch to Tx or Rx mode, either the first few microseconds of the start of the packet will not be transmitted correctly, or the receiver sensitivity is degraded for a few microseconds after the end of a packet transmission. However, once the initial configuration of the AD9361 by the CPU is done, during normal operation hardly any additional CPU activity on the SPI bus is observed, making the chance of FPGA needing to wait for the CPU's SPI transaction very low.

On the other hand, the CPU should not initiate an SPI transaction when the FPGA is currently using the lines. The Zynq UltraScale+ MPSoC, for example, has a multi-master mode, which can postpone a write when it detects that another master is currently busy. However, since the CPU activates the chip select already around 1 μs before it starts the transaction, the FPGA write will be finished before the CPU transaction and postponing is not needed for short transactions. In order to verify this behavior, we faked a collision by letting the FPGA write directly after the CPU sets the chip select line low (which is normally not possible by our design). In Fig. 6, it can be seen that the FPGA then still finishes well before the CPU starts the actual transaction. Thus, when the CPU initiates an SPI transaction just after the FPGA started a write, the FPGA would be able to finish it without conflict.

An additional benefit of the proposed design is that the SPI module can be used to interact with the AD9361 in real-time for other purposes as well. An example would be to recall a Fast Lock Profile, to allow for fast frequency changes.

5. Experimental evaluation

In this section, we evaluate the performance of the proposed LO control solution. First of all, measurements via coaxial cable as well as over-the-air are performed in order to quantify the influence of Tx LO control on the receiver sensitivity and noise floor. Next, we examine the influence of our solution on the transmission quality by measuring the error vector magnitude (EVM). Since the solution is integrated with openwifi, both receiver sensitivity and EVM are measured using Wi-Fi transmissions.

5.1. Receiver sensitivity and noise floor

The IEEE 802.11 standard defines minimum receiver sensitivity levels for each MCS at the point where the Packet Error Rate (PER)

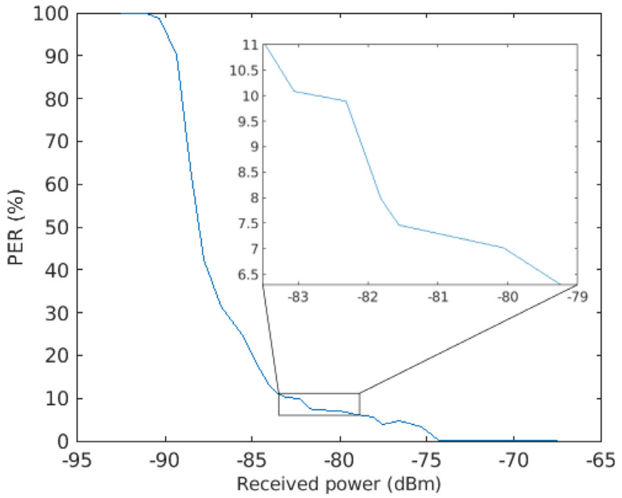


Fig. 7. PER measurement as a function of received power, including a zoomed region around the 10% PER area.

is 10% or less. In order to examine the PER as a function of received power for our setup, first a measurement is done using a coaxial cable. Furthermore, we calibrate RSSI measurements to an absolute received power level using a conducted test with a professional wireless tester.

5.1.1. Conducted tests

We utilize the ZCU102 with FMCOMMS3 as receiver in the 2.4 GHz band. It is connected with a coaxial cable with known loss L (1.3 dB) to the R&S CMW270 wireless connectivity tester [39]. The tester is set to generate a packet created via the MATLAB WLAN Toolbox [40], appended with 1 ms of noise with the same amplitude as the packet. The generated packet is compliant with the high throughput (HT) format, containing 512 PSDU bytes. In total, 10,000 packets using MCS 0 were transmitted by the CMW270 with a Tx power level P_T, varying from −93 to −66 dBm. The received power P_R in dBm is then calculated as: $P_R = P_T - L$. The ZCU102 runs openwifi in monitor mode, meaning that it passively decodes any packets without having to associate with an access point. Using statistics collected in the FPGA and read out via the driver, the number of correctly received packets (meaning the Frame Check Sequence (FCS) is correct) is determined after each measurement.

The resulting PER as a function of received power is shown in Fig. 7. It can be seen that around 10% PER, the curve is relatively flat, meaning that a change in received power only slightly affects the PER. For this reason, achieving a PER that is consistently just below 10% in an over-the-air test is challenging. Therefore, for each over-the-air sensitivity measurement, we give the PER and its standard deviation as well.

In order to be able to record the absolute received power in over-the-air tests, where attenuation is no longer predetermined, we utilize the RSSI measurement feature of the AD9361, which measures the power level in dB with 0.25 dB accuracy and compensates for the receive path gain. Both AD-FMCOMMS2-EBZ and AD-FMCOMMS3-EBZ [29] RF front-ends are involved in the receiver sensitivity measurements. The former is optimized for the frequency range 2.4-2.5 GHz, whereas the latter performs better over the complete RF frequency of 70 MHz to 6 GHz, due to the use of different baluns. Using the known cable loss and transmit power of the CMW270, a calibration was performed by calculating the offset for the linear relation between RSSI and received power. This is done at a frequency of 2.437 GHz and 5.220 GHz (Wi-Fi channel 6 and 44) for both front-ends.

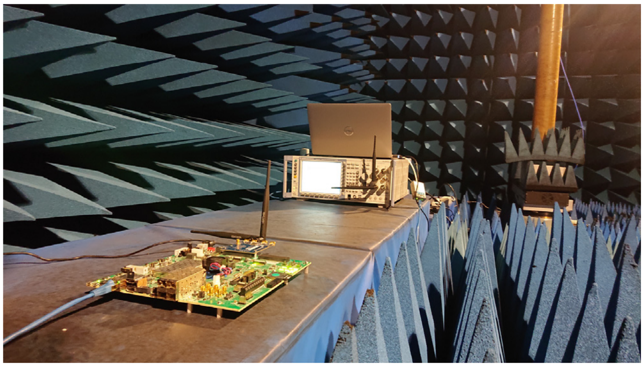


Fig. 8. Measurement setup in the anechoic chamber.

5.1.2. Over-the-air tests

In order to eliminate any interference from unwanted signals, the over-the-air measurements are performed in a full anechoic chamber, provided by the research infrastructure of IDLab [41]. Fig. 8 shows a picture of the experimental setup.

We utilize two VERT2450 antennas from Ettus Research [42], connected to ports Tx1 and Rx1 of the used RF front-end. These are dipole antennas with toroidal polarization, providing 2.35 dBi peak gain and 1.96 dBi average gain at 2.45 GHz and 0.27 dBi peak gain and −2.23 dBi average gain at 5.25 GHz in the H-plane. In order to have the largest antenna isolation, they are oriented 90 degrees apart. For reference, we also measure the performance when the antennas are oriented at 0 degrees (meaning parallel to each other).

Again, we let the CMW270 transmit 10k packets for each receiver sensitivity measurement. The center frequency is set to either 2.437 GHz or 5.220 GHz. The ZCU102 runs openwifi in monitor mode (configured either on Wi-Fi channel 6 or 44) with the added LO control functionality in FPGA, which turns the LO on before packet transmission and off afterwards. In each measurement, the receiver's Automatic Gain Control (AGC) is set to manual mode with the maximum gain. For every measurement configuration, the transmit power of the CMW270 is first adapted so that the PER at the receiver is just below 10%. Then the measurement is repeated five times with the previously determined transmit power. While the CMW270 is transmitting, the relative RSSI value is recorded, which is converted to absolute received power in dBm using the calibration result.

The obtained receiver sensitivity for MCS 0 with and without the LO control mechanism for different settings (i.e., antenna placement, RF boards and frequency bands) is shown in Table 2. In order to assess the receiver sensitivity for higher modulation types (requiring higher signal quality), measurements using packets with MCS 7 are done as well. Due to limited availability of the anechoic chamber, this was done using FMCOMMS2 only. The results are given in Table 3. For FMCOMMS2, we observe that the receiver sensitivity can be improved by 16.50 dB with an antenna configuration of 0 degrees in the 2.4 GHz band. In the 5 GHz band, the improvement is limited to 9.50 dB. For packets using MCS 7, although much higher signal power is required, similar improvement in receiver sensitivity is observed. For FMCOMMS3, the improvement in the 2.4 GHz band is about 3 dB smaller as compared to FMCOMMS2, whereas it is similar in the 5 GHz band. At 90 degrees, the improvement is about 3 dB and 1 dB smaller in the 2.4 GHz and 5 GHz band, respectively, for both front-ends, which is expected due to better antenna isolation.

In order to prove that the receiver sensitivity improvements are actually thanks to the effective removal of LO leakage, we measure the received power level via RSSI, under the same configurations as the previous measurements, except that CMW270 is not transmitting any packet. Since the measured power contains no desired signal, it is hereafter referred to as noise floor. The results of the noise floor difference with and without LO control are shown in Fig. 9. We also

Table 2
Receiver sensitivity for different configurations with and without LO control for MCS 0.

Antenna placement	LO control	2.4 GHz (FMCOMMS2)		5 GHz (FMCOMMS2)		2.4 GHz (FMCOMMS3)		5 GHz (FMCOMMS3)	
		Sensitivity (dBm)	PER $\pm\sigma$ (%)	Sensitivity (dBm)	PER $\pm\sigma$ (%)	Sensitivity (dBm)	PER $\pm\sigma$ (%)	Sensitivity (dBm)	PER $\pm\sigma$ (%)
0 degrees	On	−84.75	7.6 ± 1.0	−83.00	9.7 ± 0.5	−85.00	10.3 ± 1.0	−84.50	8.7 ± 0.3
	Off	−68.25	7.9 ± 0.6	−73.50	8.1 ± 0.6	−71.50	9.3 ± 1.3	−75.00	7.4 ± 0.8
90 degrees	On	−83.50	9.4 ± 0.3	−83.50	7.8 ± 1.0	−82.00	8.8 ± 0.5	−84.50	8.9 ± 0.7
	Off	−69.75	9.1 ± 0.3	−75.50	6.3 ± 1.3	−72.00	6.4 ± 0.6	−76.50	6.8 ± 1.2

Table 3
Receiver sensitivity using FMCOMMS2 with and without LO control for MCS 7.

Antenna placement	LO control	2.4 GHz		5 GHz	
		Sensitivity (dBm)	PER $\pm\sigma$ (%)	Sensitivity (dBm)	PER $\pm\sigma$ (%)
0 degrees	On	−69.50	9.6 ± 0.4	−67.25	8.8 ± 0.3
	Off	−52.50	8.9 ± 0.4	−57.75	8.8 ± 0.5
90 degrees	On	−67.00	8.9 ± 0.3	−61.00	5.5 ± 1.4
	Off	−52.50	6.6 ± 0.2	−52.50	9.4 ± 0.5

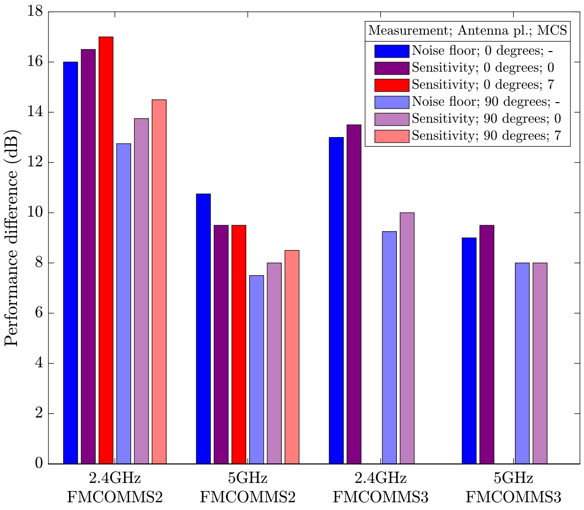


Fig. 9. Difference in noise floor and sensitivity with and without LO control for the different frequency bands, RF boards, antenna placement and MCS.

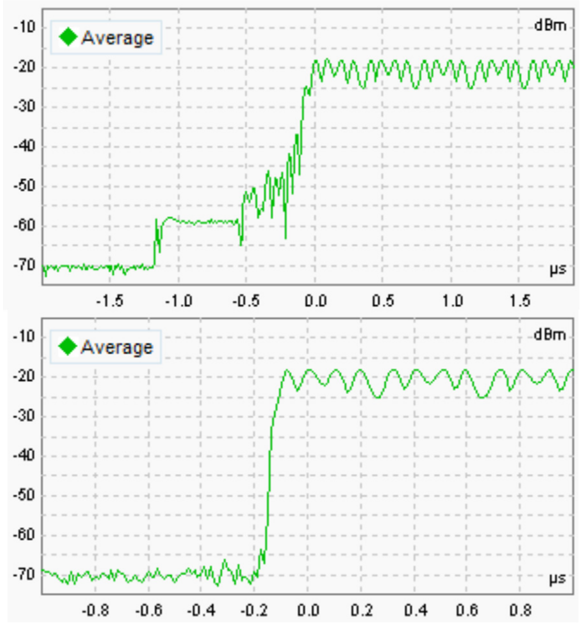


Fig. 10. Tx output power over time when the LO divider is powered on 1.2 µs (top) or 0.1 µs (bottom) before the start of a packet.

show the receiver sensitivity differences reported here, which are directly derived from the results in Table 2 and 3 under the corresponding setting, without compensating for the varying PER.

It can be seen that for FMCOMMS2 with an antenna configuration of 0 degrees, the noise floor difference in the 2.4 GHz band is 16 dB, which results in similar receiver sensitivity improvement for both MCS 0 and MCS 7. In general, it can be seen that the difference in noise floor directly relates to the observed receiver sensitivity, regardless of the used MCS or antenna placement. Furthermore, we observe that LO control in the 5 GHz band has a lower impact, namely up to 10.75 dB noise floor difference, as can be explained by the frequency selectivity of the baluns. The noise floor and sensitivity difference of FMCOMMS3 for the 5 GHz band is similar to that of the FMCOMMS2. In the 2.4 GHz band, the differences using FMCOMMS3 are less substantial than for FMCOMMS2, meaning that the wide-band balun also limits the LO leakage.

5.2. Transmission quality

In order to examine the influence of controlling the Tx path components on the transmission quality, the EVM of all carriers of a transmitted Wi-Fi packet by an SDR is examined by the CMW270. The measurements are carried out using the ZCU102 with FMCOMMS2, which is configured to transmit 1000 IEEE 802.11n packets with a payload of 1400 bytes using the packet injection mode (see *openwifi/doc/app_notes/inject_80211.md* in [38]). The board is connected via a coaxial cable to the CMW270, which reports the EVM of all carriers.

Firstly, we measured no difference in EVM with and without RF port control, regardless of the switching time before the packet. However, the method LO control has a slight influence on the EVM of the transmitted signal. This is observed even when the trigger for the SPI transaction is given sufficiently in advance, such that the LO divider is powered up just before the start of the packet. As explained, this may be due to that the phase of the LO is affected by its power supply. Therefore, we also examined the EVM in case the LO divider is powered up well before the start of a packet, in order to let the power supply stabilize. In all cases, the SPI command to turn the LO divider off is triggered 0.4 µs before the end of the packet, such that it is powered down around 0.1 µs after the packet. It is found experimentally that delaying powering the LO divider off even more does not influence the EVM.

We measure three different configurations, namely when the LO divider is switched on at all times (no control), when it is switched on around 1.2 µs or 0.1 µs before the start of a packet. For the latter two cases, the Tx power over time (averaged over 100 packets) measured by the CMW270 is displayed in Fig. 10. The packet transmission

Table 4
The Error Vector Magnitude (dB) of the transmitted constellation for MCS 0 and MCS 7 with different LO control settings.

LO switch	2.4 GHz			5 GHz		
	Always on	1.2 μs before	0.1 μs before	Always on	1.2 μs before	0.1 μs before
EVM MCS 0	−38.2 ± 0.67	−38.0 ± 0.67	−37.8 ± 0.65	−34.9 ± 0.98	−33.8 ± 0.88	−33.0 ± 0.79
EVM MCS 7	−38.8 ± 0.73	−38.6 ± 0.71	−38.4 ± 0.71	−35.2 ± 0.98	−33.9 ± 0.91	−33.0 ± 0.83

corresponds to an output power of around −20 dBm. Turning the LO divider on leads to a jump in the output power as compared to the noise floor of around −70 dBm. The resulting EVM and its standard deviation for both MCS 0 and 7, in the 2.4 GHz band (channel 6) and 5 GHz band (channel 44) are shown in Table 4. It can be seen that in the worst case, where the LO divider is switched on 0.1 μs before the start of the packet, the EVM is 0.4 dB worse than without switching in the 2.4 GHz band for both MCSs. This can be limited to 0.2 dB when switching it 1.2 μs before the packet starts. In the 5 GHz band, the EVM is degraded by around 2.0 dB for both MCSs. It can be limited to almost 1 dB when switching it 1.2 μs before. The larger effect in the 5 GHz band can be explained by the fact that at higher LO frequencies, its phase noise performance degrades more. It is important to note that there is practically no variation in EVM per symbol in all scenarios, which confirms that the control operation is done correctly. Thus, in case of switching the LO divider a shorter time in advance, the EVM of the whole packet is slightly degraded, rather than only the first few symbols being affected.

Since the transmit EVM requirement of IEEE 802.11n is only −5 dB for MCS 0 and −27 dB for MCS 7, either configuration does not significantly impact the performance of the system. Therefore, we achieve a TT of around 0.7 μs. In case the EVM specifications are more strict, the time between the LO divider being powered up and the start of the packet transmission should be delayed up until the maximum allowed TT. Another option is to use RF port control, which comes at the cost of more Tx LO leakage.

6. Conclusion

SDR platforms generally use a multi-purpose RF front-end to enable prototyping of various wireless communication solutions. A ZIF transceiver, which is nowadays the preferred SDR front-end architecture, suffers from self-interference due to LO leakage. This leads to performance degradation even for TDD (half-duplex) systems. Several known methods to suppress the self-interference are discussed, which either cannot achieve a TT suitable for common wireless protocols, e.g. Wi-Fi and 5G systems, have limited performance, or are practically infeasible due to the required hardware modifications. A novel solution is proposed that operates the RF chip in FDD mode at the same Rx and Tx frequency to keep TT short, while the LO divider is controlled in real-time to completely eliminate self-interference. A comprehensive hardware/software design is presented, that includes calibration and control target conflict resolving, for the realization of this method on a SoC consisting of a CPU and FPGA. Using the AD9361 ZIF RF chip, our approach only takes 640 ns for Rx to Tx switching and 480 ns for Tx to Rx switching, while the potential LO leakage is measured to be as low as in the conventional TDD mode. Over-the-air measurements show that an improvement of up to 17 dB and 9.5 dB in receiver sensitivity, respectively in the 2.4 GHz and 5 GHz band, can be observed. The improvement is directly related to measured noise floor differences. A very minor increase in the transmitted signal's EVM is observed, which can be reduced by adapting the time at which the LO divider is switched on before a packet is sent.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

References to the open-source code accompanying this work have been given in the paper.

References

[1] N. Nikaein, M.K. Marina, S. Manickam, A. Dawson, R. Knopp, C. Bonnet, OpenAirInterface: A flexible platform for 5G research, SIGCOMM Comput. Commun. Rev. 44 (5) (2014) 33–38, http://dx.doi.org/10.1145/2677046.2677053.

[2] I. Gomez-Miguelez, A. Garcia-Saavedra, P. Sutton, P. Serrano, C. Cano, D. Leith, srsLTE: an open-source platform for LTE evolution and experimentation, ISBN: 978-1-4503-4252-0, 2016, pp. 25–32, http://dx.doi.org/10.1145/2980159.2980163.

[3] X. Jiao, W. Liu, M. Mehari, M. Aslam, I. Moerman, openwifi: a free and open-source IEEE802.11 SDR implementation on SoC, in: 2020 IEEE 91st Vehicular Technology Conference, VTC2020-Spring, IEEE, 2020, pp. 1–2.

[4] F. Pereira de Figueiredo, D. Stojadinovic, P. Maddala, R. Mennes, I. Jabandzic, X. Jiao, I. Moerman, SCATTER PHY: An open source physical layer for the DARPA spectrum collaboration challenge, Electronics 8 (2019) 1343, http://dx.doi.org/10.3390/electronics8111343.

[5] M. Cominelli, F. Kosterhon, F. Gringoli, R. Lo Cigno, A. Asadi, IEEE 802.11 CSI randomization to preserve location privacy: An empirical evaluation in different scenarios, Comput. Netw. 191 (2021) 107970, http://dx.doi.org/10.1016/j.comnet.2021.107970, URL https://www.sciencedirect.com/science/article/pii/S138912862100102X.

[6] J. Haxhibeqiri, X. Jiao, M. Aslam, I. Moerman, J. Hoebeke, Enabling TSN over IEEE 802.11: Low-overhead time synchronization for wi-fi clients, in: 2021 22nd IEEE International Conference on Industrial Technology, ICIT, 1, IEEE, 2021, pp. 1068–1073, http://dx.doi.org/10.1109/ICIT46573.2021.9453686.

[7] C.M. Karle, M. Kreutzer, J. Pfau, J. Becker, A hardware/software co-design approach to prototype 6G mobile applications inside the GNU radio SDR ecosystem using FPGA hardware accelerators, in: International Symposium on Highly-Efficient Accelerators and Reconfigurable Technologies, in: HEART2022, Association for Computing Machinery, New York, NY, USA, 2022, pp. 33–41, http://dx.doi.org/10.1145/3535044.3535049, URL https://doi.org/10.1145/3535044.3535049.

[8] D.M. Molla, H. Badis, L. George, M. Berbineau, Software defined radio platforms for wireless technologies, IEEE Access 10 (2022) 26203–26229, http://dx.doi.org/10.1109/ACCESS.2022.3154364.

[9] X. Jiao, I. Moerman, W. Liu, F.A.P. de Figueiredo, Radio hardware virtualization for coping with dynamic heterogeneous wireless environments, in: Cognitive Radio Oriented Wireless Networks, Springer International Publishing, Cham, 2018, pp. 287–297.

[10] L. Baldesi, F. Restuccia, T. Melodia, ChARM: Nextg spectrum sharing through data-driven real-time O-RAN dynamic control, in: IEEE INFOCOM 2022 - IEEE Conference on Computer Communications, 2022, http://hdl.handle.net/2047/D20423481.

[11] O. Seijo, J.A. López-Fernández, I. Val, W-SHARP: Implementation of a high-performance wireless time-sensitive network for low latency and ultra-low cycle time industrial applications, IEEE Trans. Ind. Inform. 17 (5) (2021) 3651–3662, http://dx.doi.org/10.1109/TII.2020.3007323.

[12] H. Wu, S. Lu, T. Wang, Z. Yuan, C. Peng, Z. Li, Z. Tan, B. Ding, Y. Li, J. Liu, The tick programmable low-latency SDR system, GetMobile: Mob. Comput. Commun. 21 (2018) 26–30, http://dx.doi.org/10.1145/3229316.3229326.

[13] IEEE standard for information technology - part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications, 2021, http://dx.doi.org/10.1109/IEEESTD.2021.9363693, IEEE Std 802.11-2020 (Revision of IEEE Std 802.11-2016), 1–4379.

[14] 5G NR user equipment (UE) radio transmission and reception, 2019, ETSI TS 38 101-4, V15.0.0.

[15] MAX2828/MAX2829 single-/dual-band 802.11a/b/g world-band transceiver ICs, 19–3455, 2004, Maxim Integrated, Rev. 0.

[16] AD9361 reference manual, UG-570, 2014, Analog Devices, Rev. A.

[17] TDD Mode Switching Time, Design Support AD9361/AD9363/AD9364, TDD Mode Switching Time - Documents - Design Support AD9361/AD9363/AD9364 - EngineerZone, URL https://ez.analog.com/wide-band-rf-transceivers/design-support/w/documents/10069/tdd-mode-switching-time.

[18] T. Havinga, X. Jiao, M. Aslam, W. Liu, I. Moerman, WIP: Achieving self-interference-free operation on SDR platform with critical TDD turnaround time, in: 2022 IEEE 23rd International Symposium on a World of Wireless, Mobile and Multimedia Networks, WoWMoM, 2022, pp. 173–176, http://dx.doi.org/10.1109/WoWMoM54355.2022.00051.

[19] S. Sadjina, C. Motz, T. Paireder, M. Huemer, H. Pretl, A survey of self-interference in LTE-advanced and 5G new radio wireless transceivers, IEEE Trans. Microw. Theory Tech. 68 (3) (2020) 1118–1131, http://dx.doi.org/10.1109/TMTT.2019.2951166.

[20] H. Wu, T. Wang, J. Chen, S. Liu, S. Tian, S. Lu, M. Ma, L. Song, B. Jiao, GRT-duplex: A novel SDR platform for full-duplex WiFi, Mob. Netw. Appl. 21 (2016) http://dx.doi.org/10.1007/s11036-016-0710-z.

[21] L. Anttila, V. Lampu, S.A. Hassani, P.P. Campo, D. Korpi, M. Turunen, S. Pollin, M. Valkama, Full-duplexing with SDR devices: Algorithms, FPGA implementation, and real-time results, IEEE Trans. Wireless Commun. 20 (4) (2021) 2205–2220, http://dx.doi.org/10.1109/TWC.2020.3040226.

[22] M.S. Amjad, F. Dressler, Software-based real-time full-duplex relaying: An experimental study, IEEE Trans. Green Commun. Netw. 4 (3) (2020) 647–656, http://dx.doi.org/10.1109/TGCN.2020.2966140.

[23] X. Peng, Z. Wang, J. Mo, C. Wang, J. Liu, F. Yu, A blind calibration model for I/Q imbalances of wideband zero-IF receivers, Electronics 9 (11) (2020) http://dx.doi.org/10.3390/electronics9111868, URL https://www.mdpi.com/2079-9292/9/11/1868.

[24] G.C. Tripathi, M. Rawat, SDR solution for enhanced quality wider bandwidth communication, in: 2019 IEEE MTT-S International Microwave and RF Conference, IMARC, 2019, pp. 1–5, http://dx.doi.org/10.1109/IMaRC45935.2019.9118657.

[25] M. Aslam, X. Jiao, W. Liu, I. Moerman, An approach to achieve zero turnaround time in TDD operation on sdr front-end, IEEE Access 6 (2018) 75461–75470, http://dx.doi.org/10.1109/ACCESS.2018.2883253.

[26] P. Meaney, A. Hartov, T. Raynolds, C. Davis, S. Richter, F. Schoenberger, S. Geimer, K. Paulsen, Low cost, high performance, 16-channel microwave measurement system for tomographic applications, Sensors 20 (18) (2020) http://dx.doi.org/10.3390/s20185436, URL https://www.mdpi.com/1424-8220/20/18/5436.

[27] A.M.M. Chandran, M. Zawodniok, Transmitter leakage analysis when operating USRP (N210) in duplex mode, in: 2015 IEEE International Instrumentation and Measurement Technology Conference (I2MTC) Proceedings, 2015, pp. 340–345, http://dx.doi.org/10.1109/I2MTC.2015.7151291.

[28] MS2690A/MS2691A/MS2692A signal analyzer operation manual, m-w2852ae-33.0, Anritsu corporation, 2022, URL https://dl.cdn-anritsu.com/en-au/test-measurement/files/Manuals/Operation-Manual/MS269xA/MS269xA_SignalAnalyzer_Operation_Manual_e_33_0.pdf. (Accessed 10 January 2022).

[29] AD9361 Wideband Software Defined Radio Board, Analog Devices, URL https://www.analog.com/en/design-center/evaluation-hardware-and-software/evaluation-boards-kits/eval-ad-fmcomms3-ebz.html.

[30] ZCU102 evaluation board user guide, ug1182, xilinx, 2019, Version 1.6.

[31] B200/B210/B200mini/B205mini, Ettus Research LLC, 2021, URL https://kb.ettus.com/B200/B210/B200mini/B205mini.

[32] Ettus USRP E300 Embedded Family Hardware Resources, Ettus Research LLC, 2020, URL https://kb.ettus.com/Ettus_USRP_E300_Embedded_Family_Hardware_Resources.

[33] SKY13335-381LF: 0.1 to 6.0 GHz GaAs SPDT Switch, Skyworks Solutions, Inc, 2014, URL https://www.skyworksinc.com/-/media/SkyWorks/Documents/Products/301-400/SKY13335_381LF_201093D.pdf.

[34] SKY13416-485LF: 0.1 to 6.0 GHz SP6T Antenna Switch, Skyworks Solutions, Inc, 2019, URL https://www.skyworksinc.com/-/media/SkyWorks/Documents/Products/701-800/SKY13416_485LF_201679I.pdf.

[35] AD9361 for WiFi (TDD) application, ADI Engineer Zone, 2014, URL https://ez.analog.com/wide-band-rf-transceivers/design-support/f/q-a/79230/ad9361-for-wifi-tdd-application. (Accessed 06 January 2022).

[36] AD9361 register map reference manual, UG-671, analog devices, 2014, Rev. 0.

[37] I. Ungurean, Timing comparison of the real-time operating systems for small microcontrollers, Symmetry 12 (4) (2020) http://dx.doi.org/10.3390/sym12040592, URL https://www.mdpi.com/2073-8994/12/4/592.

[38] X. Jiao, Open-sdr, GitHub Repository, 2022, https://github.com/open-sdr. (Accessed 5 April 2022).

[39] R&S CMW Wideband Radio Communication Tester, Rohde & Schwarz, 2019, Version 05.00.

[40] WLAN Toolbox, MATLAB, URL https://nl.mathworks.com/help/wlan/index.html.

[41] IDLab, Electromagnetism, URL https://www.ugent.be/ea/idlab/en/research/research-infrastructure/electromagnetism.htm.

[42] WSS007 Dual Band Antenna With SMA(M)(Black), Ettus Research LLC, 2011, URL https://kb.ettus.com/images/9/9e/ettus_research_vert2450_datasheet.pdf.