WIP: Achieving Self-Interference-Free Operation on SDR Platform with Critical TDD Turnaround Time

Thijs Havinga, Xianjun Jiao, Muhammad Aslam, Wei Liu and Ingrid Moerman

IDLab, Ghent University - imec, Belgium

Email: {Thijs.Havinga, Xianjun.Jiao, Muhammad.Aslam, Wei.Liu, Ingrid.Moerman}@UGent.be

Abstract—**Software Defined Radio (SDR) platforms are valuable for research and development activities or high-end systems that demand real-time adaptable wireless protocols. While low latency can be achieved using the dedicated digital processing unit of a state-of-the-art SDR platform, its Radio Frequency (RF) front-end often poses a limitation in terms of turnaround time (TT), the time needed for switching from the receiving to the transmitting mode (or vice versa). Zero Intermediate Frequency (ZIF) transceivers are favorable for SDR, but suffer from self-interference even if the device is not currently transmitting. The strict MAC-layer requirements of Time Division Duplex (TDD) protocols like Wi-Fi cannot be achieved using configurable ZIF transceivers without having to compromise receiver sensitivity. Using a novel approach, we show that the TT using the AD9361 RF front-end can be as low as 640 ns, while the self-interference is at the same level as achieved by the conventional TDD mode, which has a TT of at least 55 μs. As compared to Frequency Division Duplex (FDD) mode, a decrease of receiver noise floor by about 13 dB in the 2.4 GHz band and by about 4.5 dB in the 5 GHz band is achieved.**

Index Terms—**software defined radio, self-interference, TDD, turnaround time, receiver sensitivity**

I. 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. In addition, the configurable nature of SDR makes it also a viable platform for high-end solutions in environments with dynamic characteristics or requirements.

An SDR generally consists of two parts. The first, 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). Another option is to use a Field Programmable Gate Array (FPGA), in which digital hardware is configured by how the designer programs it.

Currently, one of the downsides of SDR using a host computer is the latency induced by the data travelling between the radio front-end and the processing unit. For example, according to [1], the latency between a host computer and a Universal Software Radio Peripheral (USRP) version X310 using a PCIe link is 79 μs. For applications that require low latency, an SDR based on a System-on-Chip (SoC), where the connection between CPU, FPGA and RF front-end is fast, is more suitable. For example, the connection between CPU and FPGA can be as low as 1.435 μs [1].

For the implementation of Time Division Duplex (TDD) systems, the SDR 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 design of a Medium Access Control (MAC) protocol is limited by the TT. For instance in the Wi-Fi standard, an acknowledgement frame as a response to a successfully received frame should arrive at the transmitter within a minimum 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 [2]. In the 5G standard [3], a guard period of 17.84 μs is specified for a downlink to uplink pattern when using a subcarrier spacing of 120 kHz.

The TT of protocol-specific chips is in the order of 1 μs [4], whereas configurable RF front-ends have significantly higher TT. In this paper, we consider Zero-Intermediate Frequency (ZIF) transceivers, more specifically the AD9361 front-end [5], which is widely used in the SDR research community. 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 [6], which is too high for many MAC protocols. Operation in Frequency Division Duplex (FDD) mode, where both the Rx and Tx chain are active, avoids this TT, but suffers from receiver sensitivity degradation due to Tx LO leakage when operating at the same frequency. This leakage is present even if the system is not currently transmitting. It also cannot be eliminated by conventional filtering, since the noise exists in the same frequency range as the desired signal.

In this paper, we propose a method for self-interference-free operation of a ZIF transceiver while achieving a low turnaround time. For this, we consider the implementation of Wi-Fi using the *openwifi* platform [7], which is suited for SDRs based on a SoC. First, we discuss the related work on this topic. Next, we explore different options before presenting our proposed solution. Then, we show and evaluate the results of measurements on a real hardware platform, after which we conclude the paper.

II. RELATED WORK

The authors of [8] 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 5 MHz bandwidth. The in-band error between a transmitted and captured signal was measured 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 [9], a method to achieve zero TT by leaving both the Tx and Rx chain of the RF front-end on is presented. 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, there was still self-interference of 3.2 dB.

Implementation of Wi-Fi meeting the SIFS requirement using a ZIF transceiver has already been done in [10]. 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. 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 Rx-Tx switching is done, nor do they mention the receiver sensitivity.

In order to realize full-duplex systems on SDR platforms, the works [12], [13] and [14] present digital self-interference mitigation techniques. Such techniques, however, cannot eliminate the self-interference to the same extent as compared to switching between Rx and Tx due to the unpredictable noise coming from the LO. Furthermore, it comes at the cost of additional computation time or hardware resources.

III. PROPOSED SOLUTION

Here we discuss possible solutions to eliminate self-interference using the widely used AD9361 and a SoC, while still maintaining low latency. The idea behind these solutions can in general be applied to all ZIF transceivers.

A simplified diagram of the AD9361 Tx path is shown in Figure 1. The AD9361 can operate in Frequency Division Duplex (FDD) or Time Division Duplex (TDD) mode. In FDD mode, both Rx and Tx synthesizers remain on at all times, whereas in TDD operation these are powered up alternately. Furthermore, there are some options to tweak the operations, which are listed hereafter [6]:

1) *Standard Enable State Machine (ENSM) TDD Mode*: In this mode, Voltage-Controlled Oscillator (VCO) calibrations take place when the internal state machine changes
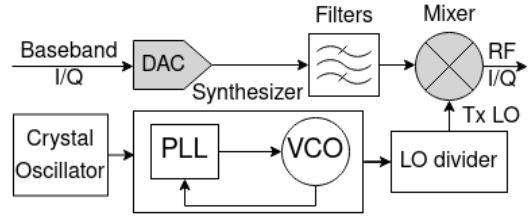

Fig. 1. Simplified diagram of the AD9361 Tx path.

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 to the PLL lock, so this method has a TT of at least 55 μs.

2) *Standard TDD Mode*: This mode is the same as the one mentioned above, except that 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 dominated by the DAC power up time of 18 μs.

3) *Standard TDD Mode/Dual Synth*: In this case, no calibrations take place and the PLLs do not have to be locked, since these are on at all times. However, the DAC still needs to power up and flushing needs to be done, resulting in at least 18 μs delay.

4) *FDD Independent Mode*: In this mode, both the Rx and Tx data paths are active, but can be independently controlled. No calibrations or flushing takes place, but the Tx DAC still powers up, resulting in 18 μs delay.

None of these methods can comply to strict MAC requirements like SIFS, thus we are committed to normal FDD mode. The AD9361 already provides RF DC offset and quadrature tracking calibration to mitigate phase and gain error, but these do not suffice to fully eliminate the self-interference, especially when antennas are attached to the Tx and Rx ports. Therefore, we propose the following method to further suppress self-interference in FDD mode. Instead of powering down the synthesizers in the AD9361, there is an option to power down only the LO divider, such that the signal does not reach the mixer [15]. Powering on the LO divider takes about 160 ns [16]. In FDD mode, both chains remain on completely, so there is no additional powering up or calibration needed.

Turning the LO on and off can only be done via an SPI write to the AD9361. A 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. Therefore, we implement SPI functionality in the FPGA. We use the maximum SPI frequency of the AD9361 (i.e. 50 MHz) and 24 bits are needed for the instruction to turn the Tx LO on or
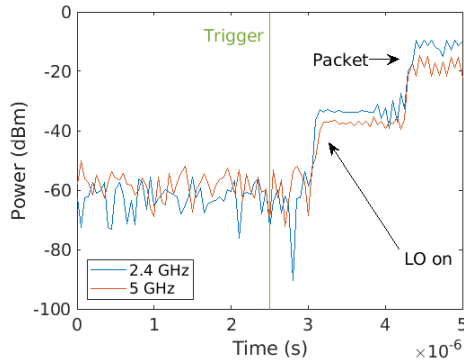
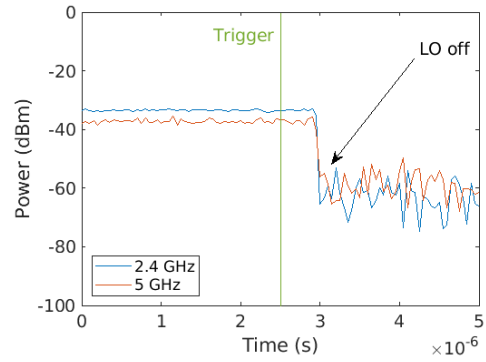Fig. 2. Tx power over time measured around Tx LO on command.

Fig. 3. Tx power over time measured around Tx LO off command.

off, resulting in a duration of 480 ns. Thus, the TT is reduced to 640 ns, which is well within the SIFS requirement of Wi-Fi.

IV. EXPERIMENTAL RESULTS AND EVALUATION

In this section, two aspects of the proposed solution are examined. First, TT is quantified by measuring the Tx power over time. Secondly, the impact on receiver sensitivity is assessed by performing receiver noise floor tests. The hardware used is the Xilinx Zynq UltraScale+ MPSoC ZCU102, with an AD-FMCOMMS2-EBZ front-end (containing the AD9361) connected via an FPGA Mezzanine Card (FMC) interface. The SPI module is implemented using Verilog and its outputs are connected to the SPI interface of the AD9361. The module uses only 33 lookup tables and 28 registers and can be found online [17] in *openwifi-hw/ip/xpu/src/spi.v*. The on/off trigger is given by the low MAC logic in FPGA.

A. Turnaround time

For quantifying TT, the Tx power has been measured using a signal analyzer (Anritsu MS2690A [18]) with a 10 MHz span and by setting the center frequency equal to the Tx LO frequency. A General Purpose Input/Output (GPIO) pin on the SoC is connected to the signal that initiates the SPI write. This pin is used as trigger for the signal analyzer, which then measures the Tx power 2.5 μs before and after the time at which the SPI write was initiated. Measurement points are taken every 0.05 μs. We have measured the Tx power for two LO frequency configurations: Wi-Fi channel 1 (2.4 GHz band) and channel 44 (5 GHz band). The output power when turning the Tx LO on and off is shown in Figure 2 and Figure 3 respectively. In Figure 2, it can be seen that after the trigger (green vertical line), the power shoots up and quickly stabilizes, indicating that the LO is turned on. The second jump indicates the start of a packet transmission. The time between the trigger and when the Tx LO is powered up is about 0.65 μs, while the time between the trigger and when the Tx LO is powered down is only about 0.5 μs as shown in Figure 3.

Thus, the calculated TT in Section III from Rx to Tx falls within the measurement resolution, while the TT from Tx to Rx down is even lower, leading to TT values comparable to

what can be achieved using protocol-specific chips. TT seems further to be independent on the frequency configuration. However, the difference between the average power with LO on or off is about 30 dB in the 2.4 GHz band, whereas it is about 22 dB in the 5 GHz band.

B. Self-interference

We assess the influence of self-interference by recording the noise floor at the Rx antenna. The expected decrease in noise floor is smaller than the difference in Tx power as measured above, since the noise will not leak to the Rx completely.

We investigate three scenarios, namely FDD mode, Standard ENSM TDD Mode (referred to as TDD) and our LO control. Using the FDD mode, the SIFS requirement is met, but no effort is done on mitigating self-interference. The Standard ENSM TDD Mode is also selected since it is the recommended way to implement Rx-Tx switching achieving no self-interference, despite the large TT.

We utilize the trigger-based I/Q capture feature of *openwifi* to obtain the noise floor (see application note [17]). During the measurement, the application is set to trigger always. Then the I/Q samples from the Rx antenna are constantly captured via a side channel in the FPGA. This module transfers the samples via DMA to the Linux driver. Using a user space application, these samples can be exported and analyzed offline.

We utilized two VERT2450 antennas from Ettus Research [19], connected to ports Tx1 and Rx1 of the AD9361, oriented orthogonal to each other. In each of the measurements, the LO leakage and quadrature calibration algorithms of the AD9361 are applied. The Automatic Gain Control (AGC) was set to a manual level of 62 dB for all measurements. Unintentionally captured packets from the environment were filtered from the results. The noise floor for the different modes on channel 1 is shown in Figure 4, while Figure 5 shows the noise on channel 44. The average relative power of the noise floor in the full 20 MHz bandwidth for each mode is calculated and shown in Table I. It can be seen that for both frequency bands, the power of the noise floor for TDD is slightly higher than our LO control. When using FDD mode, the noise floor power is about 13 dB higher in the 2.4 GHz band and more than 4 dB higher in the 5 GHz band. The difference between the bands can be
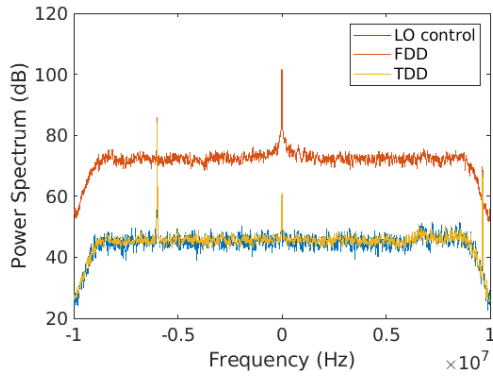
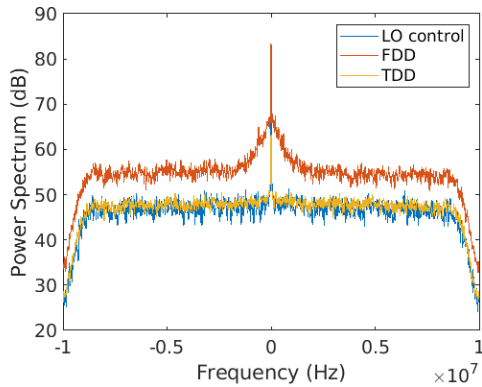Fig. 4. Receiver noise floor between different modes in the 2.4 GHz band.

Fig. 5. Receiver noise floor between different modes in the 5 GHz band.

explained by the difference in Tx power as measured before. It is expected that the receiver sensitivity will be improved to the same extent.

V. CONCLUSION

SDR platforms using ZIF transceivers suffer from self-interference in FDD mode using the same frequency for Rx and Tx. By turning the LO divider of the AD9361 on and off via an SPI writing operation in real-time, we have created a way to eliminate self-interference. This method decreases the receiver noise floor by about 13 dB in the 2.4 GHz band and more than 4 dB in the 5 GHz band as compared to FDD mode. This is a slightly better performance as compared to what can be achieved in conventional TDD mode. However, as apposed to this mode, which has a TT of at least 55 µs, our approach only takes 0.64 µs for Rx to Tx switching and just 0.5 µs for Tx to Rx switching. Our solution thus provides a way to implement TDD-based protocols on SDRs even with very

strict latency requirements, while not having to compromise the receiver noise floor. For future work, we suggest to verify the impact on receiver sensitivity in over-the-air tests.

TABLE I
AVERAGE RELATIVE POWER OF THE NOISE FLOOR.

Method	Average power (dB)	
	2.4 GHz	5 GHz
FDD	66.4	58.0
TDD	53.3	53.7
LO control	53.0	53.4

REFERENCES

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

[2] "IEEE Standard for Information technology - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," *IEEE Std 802.11-2016 (Revision of IEEE Std 802.11-2012)*, pp. 1–3534, 2016.

[3] "5G NR User Equipment (UE) radio transmission and reception," *ETSI TS 38 101-4*, 2019, v15.0.0.

[4] *MAX2828/MAX2829 Single-/Dual-Band 802.11a/b/g World-Band Transceiver ICs*, Maxim Integrated, 4 2004, Rev. 0.

[5] *AD9361 Reference Manual*, Analog Devices, 2014, Rev. A.

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

[7] X. Jiao, W. Liu, M. Mehari, M. Aslam, and 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.

[8] G. C. Tripathi and 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.

[9] M. Aslam, X. Jiao, W. Liu, and I. Moerman, "An Approach to Achieve Zero Turnaround Time in TDD Operation on SDR Front-End," *IEEE Access*, vol. 6, pp. 75 461–75 470, 2018.

[10] H. Wu, S. Lu, T. Wang, Z. Yuan, C. Peng, Z. Li, Z. Tan, B. Ding, Y. Li, and J. Liu, "The Tick Programmable Low-Latency SDR System," *GetMobile: Mobile Computing and Communications*, vol. 21, pp. 26–30, 05 2018.

[11] O. Seijo, J. A. López-Fernández, and I. Val, "w-SHARP: Implementation of a High-Performance Wireless Time-Sensitive Network for Low Latency and Ultra-low Cycle Time Industrial Applications," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 5, pp. 3651–3662, 2021.

[12] H. Wu, T. Wang, J. Chen, S. Liu, S. Tian, S. Lu, M. Ma, L. Song, and B. Jiao, "GRT-duplex: A Novel SDR Platform for Full-Duplex WiFi," *Mobile Networks and Applications*, vol. 21, 12 2016.

[13] L. Anttila, V. Lampu, S. A. Hassani, P. P. Campo, D. Korpi, M. Turunen, S. Pollin, and M. Valkama, "Full-Duplexing With SDR Devices: Algorithms, FPGA Implementation, and Real-Time Results," *IEEE Transactions on Wireless Communications*, vol. 20, no. 4, pp. 2205–2220, 2021.

[14] M. S. Amjad and F. Dressler, "Software-Based Real-Time Full-Duplex Relaying: An Experimental Study," *IEEE Transactions on Green Communications and Networking*, vol. 4, no. 3, pp. 647–656, 2020.

[15] *AD9361 Register Map Reference Manual*, Analog Devices, 2014, Rev. 0.

[16] *AD9361 for WiFi (TDD) application*, ADI Engineer Zone, 2014, Accessed on 06-01-2022. [Online]. Available: https://ez.analog.com/wide-band-rf-transceivers/design-support/f/q-a/79230/ad9361-for-wifi-tdd-application

[17] X. Jiao, "open-sdr," https://github.com/open-sdr, Accessed on 05-04-2022.

[18] *MS2690A/MS2691A/MS2692A Signal Analyzer Operation Manual*, Anritsu Corporation, Accessed 10-01-2022. [Online]. Available: https://dl.cdn-anritsu.com/en-au/test-measurement/files/Manuals/Operation-Manual/MS269xA/MS269xA_SignalAnalyzer_Operation_Manual_e_33_0.pdf

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