Technische Hochschule Ingolstadt

Dokumentation zum Projekt

Thema:
Implementierung des 802.11p-Standards auf einer Wireless Open-Access Research Platform

Authoren	:	Dominik Bayerl und Philipp Sebastian Schmieder
Studiengang	:	Master Informatik
Betreuende Professoren:	:	Prof. Dr.-Ing. Ernst-Heinrich Göldner und Prof. Dr. Stefan Hahndel
Im Fach	:	IM_Projekt
Semester	:	Wintersemester 2017/2018

Inhaltsverzeichnis

1 Projektbeschreibung und Anmerkungen

Author: Philipp Sebastian Schmieder

In diesem Projekt soll der IEEE 802.11p-Standard auf einer Wireless Open-Access Research Platform implementiert werden. 802.11p ist der festgelegte Wireless Kommunikationsstandard für die Car2x Kommunikation. Zum Ende des Projekts soll es möglich sein die Plattform mit kommerziellen 802.11p Geräten kommunizieren zu lassen. Die folgenden Kapitel beschreiben die Grundlagen die zur Umsetzung einer Wireless Kommunikation auf der Wireless Open-Access Research Platform wichtig sind, dabei wird ein Überblick über die Hardware und über die vorhandene Software gegeben. Um dem Leser später den 802.11p-Standard näher zu bringen, wird der als Software für die Plattform bereits umgesetzte 802.11a-Standard mit dem 802.11p-Standard verglichen und die Unterschiede in der physikalischen Schicht verdeutlicht. Die weiteren Kapitel befassen sich mit der Umsetzung des Standards auf der Plattform und der Validierung der Ergebnisse. Als letztes wird ein Ausblick gegeben welche Optimierungsmöglichkeiten in Zukunft noch vorgenommen werden können und wie die Plattform noch erweitert werden könnte.

2 Die Wireless Open-Access Research Platform

Author: Philipp Sebastian Schmieder

Die Wireless Open-Access Research Platform, kurz WARP, ist eine ständig weiterentwickelte frei verfügbare Plattform zur Entwicklung von Wireless Applikationen und Netzwerken. Die Plattform kombiniert eine hochperformante Hardware mit diversen Referenz Software Designs die für eigene Anwendungen angepasst oder weiterentwickelt werden können.

2.1 Das WARP v3 Kit

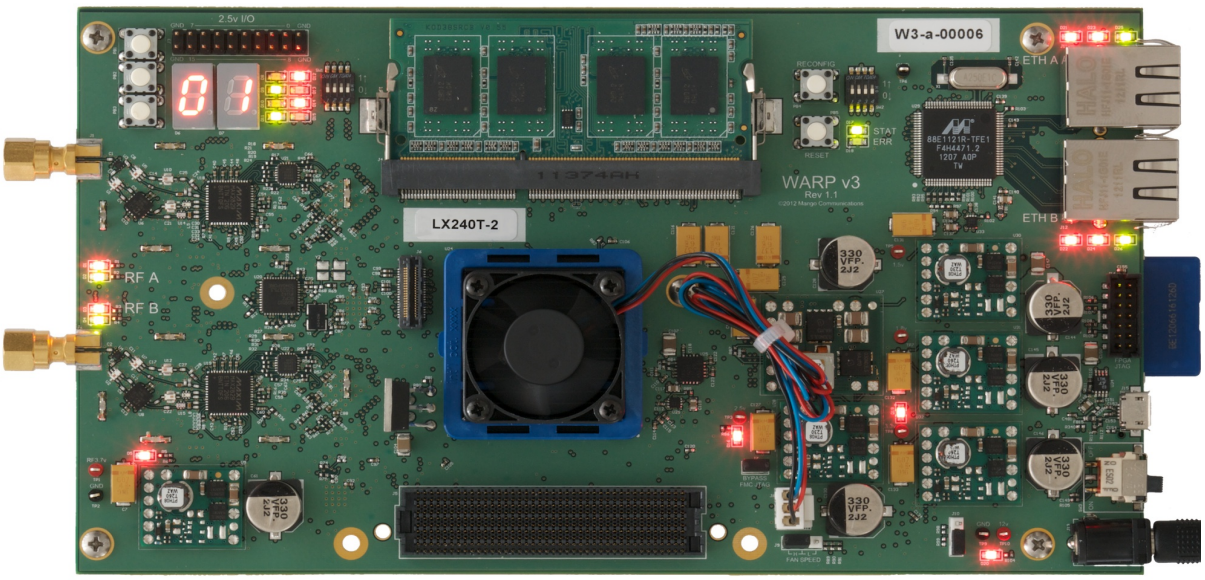

Abbildung 1: Die Warp Hardware von Mango Communications [MANGO v3, 2017]

Das Mango v3 Kit ist die bisher neueste Hardware Version für die Plattform (Stand: Dezember 2017), das Kit verfügt über einen hochperformanten Virtex-6 FPGA von Xilinx. Zur Konfiguration des FPGA verfügt das Kit über diverse Schnittstellen wie JTAG und einen SD-Karten Stecker. Das FPGA ist volatile, das heißt es muss nach jedem Ausschalten beim Neustart neu konfiguriert werden. Des weiteren verfügt das v3 Kit noch über 2 Gigabit Ethernet Schnittstellen, eine U-Art Schnittstelle,diverse Speicher und Spannungsbauteile, zwei Oszillatoren und 2 baugleiche Radio Frequency (RF) Interfaces.

2.2 Konfiguration des RF Interfaces

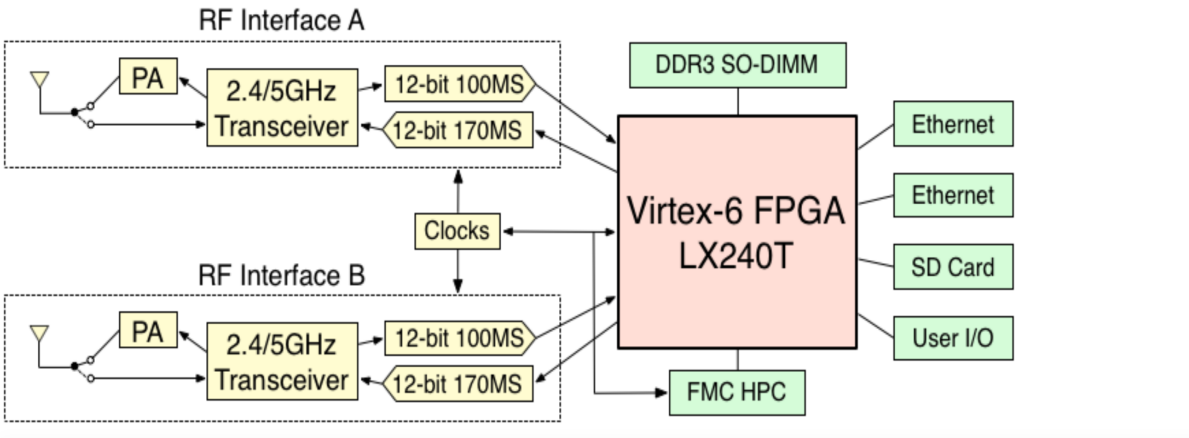

Abbildung 2: Design des Warp v3 Kits [v3 Design, 2017]

Die RF Interfaces verfügen jeweils über einen 12 Bit Digital Analog Wandler mit maximal 100 Mega Samples pro Sekunde, einen 12 Bit Analog Digital Wandler mit maximal 170 Mega Samples pro Sekunde, einen Transceiver, der sowohl 2,4 GHz als auch 5GHz implementiert, einen Dual Band Verstärker für den Sendepfad sowie einen Switch, um zwischen Sende- und Empfangspfad zu wechseln. Den Abschluss jedes RF Interfaces bildet ein Koaxialer Stecker für Hochfrequenzanwendungen (SMA).

Taktfrequenz

Das Mango v3 Kit verfügt über 2 unabhängige Oszillatoren. Einen mit einer Taktfrequenz von 200 MHz für das FPGA und einen eigenen für die RF Interfaces, die beide auf den selben Oszillator zugreifen, mit einer Taktfrequenz von 80 MHz, was dem Basisband entspricht. Es können auch eigene Taktfrequenzen über den FMC Slot eingefügt werden. Das Einstellen der Taktfrequenz im FPGA und den RF Interfaces mit den Oszillatorfrequenzen ist nicht ganz einfach und erfordert einiges Wissen über die Bauteile und macht das Einstellen der Taktfrequenz so hinreichend komplex. Die beiden Puffer Bausteine in Abbildung 3 geben die je-

weilige Taktfrequenz an das RF Interface weiter, wobei der Sampling Clock Buffer noch über einen Teiler verfügt.

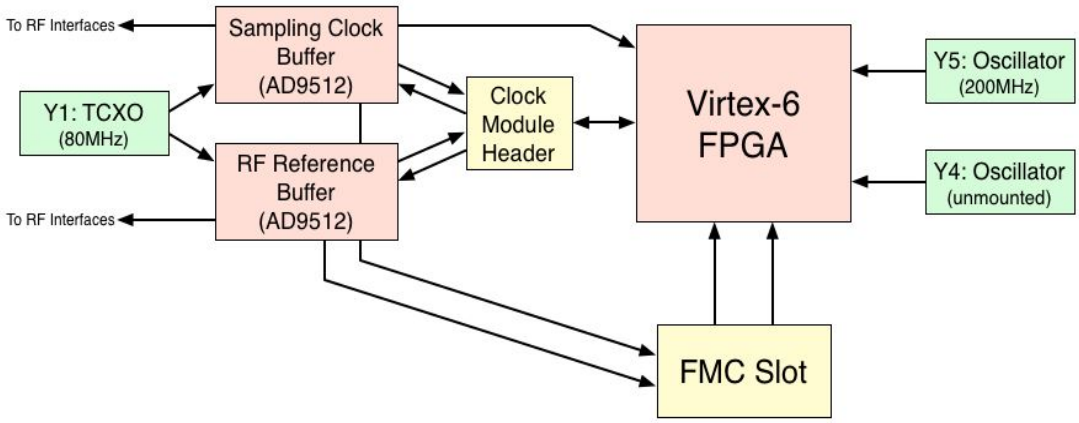

Abbildung 3: Übersicht über die Taktung des Mango v3 Kits [v3 Kit Clocking, 2017]

Einstellen des Analog/Digital- und Digital/Analogwandlers

Der erste Baustein jedes RF Interfaces ist das 10/12-Bit Schwachstrom Breitband Mixed Signal Front End(MxFE) AD9963, dieser Baustein beinhaltet sowohl den Analog/Digitalwandler (ADC) als auch den Digital/Analogwandler (DAC). Der DAC verfügt über einen 3 stufigen Interpolationsfilter, der eine 1x-,2x-,4x- oder 8x-fache Abtastratenerhöhung ermöglicht. Die Interpolationsstufen können zur Laufzeit über ein SPI-Interface eingestellt werden in dem folgende Registerflags aus Abbildung 4 gesetzt werden.

Interpolationsstufe	DAC Registerflags
1x	-
2x	INT0
4x	INT0, INT1
8x	INT0, INT1

Abbildung 4: Zu setzende Register für die Auswahl des Interpolationsfilters

Der ADC verfügt über einen 2x-fachen Dezimator der durch das Registerflag "DECßur Laufzeit über SPI ein- bzw. ausgeschaltet werden kann. Um die richtige, gewünschte Abtastfrequenz für den DAC und ADC einzustellen muss sowohl der

Takt der TXCLK vom FPGA, die Taktausgabe des Refernce Buffers, der Divider des Sampling Clock Buffers als auch der Devider bzw. Interpolationsfilter richtig eingestellt werden.

Transceiver, Verstärker und Switch

Hinter dem AD9963 sitzt der MAX2829 Transceiver, dieser generiert das Trägersignal und kann sowohl 2,4GHz als auch den 5GHz Bereich implementieren. Nach dem Transceiver folgt der Verstärker oder Power Amplifier (PA) und der Switch der zwischen Sende- und Empfangspfad wählt. Alle drei Bausteine können über SPI eingestellt werden.

3 Das Warp Reference Design 802.11

Author: Philipp Sebastian Schmieder

Das Einstellen des RF Interfaces stellt alleine durch das abstimmen der Taktfrequenzen eine relativ komplexe Aufgabe dar, dabei hat man sich noch nicht mit dem Design für das FPGA beschäftigt, welches wahrscheinlich noch einmal mehrere Monate Zeit kostet. Um ein grundlegendes Design für das FPGA und das RF Interface zu haben, gibt es von der Warp Website ein Referenz Design für diverse IEEE 802.11 Standards. Das Referenz Design implementiert jeweilige OFDM PHYs und DCF MACs und enthält fertige Designs für 802.11a, 802.11g und 802.11n. Es macht Sinn für die Grundeinstellung auf das Referenz Design zuzugreifen da man so eine Echtzeit FPGA Implementierung bekommt und einen ersten Ansatzpunkt hat um den PHY und MAC des IEEE 802.11p-Standard auf dem Mango v3 Kit zu implementieren.

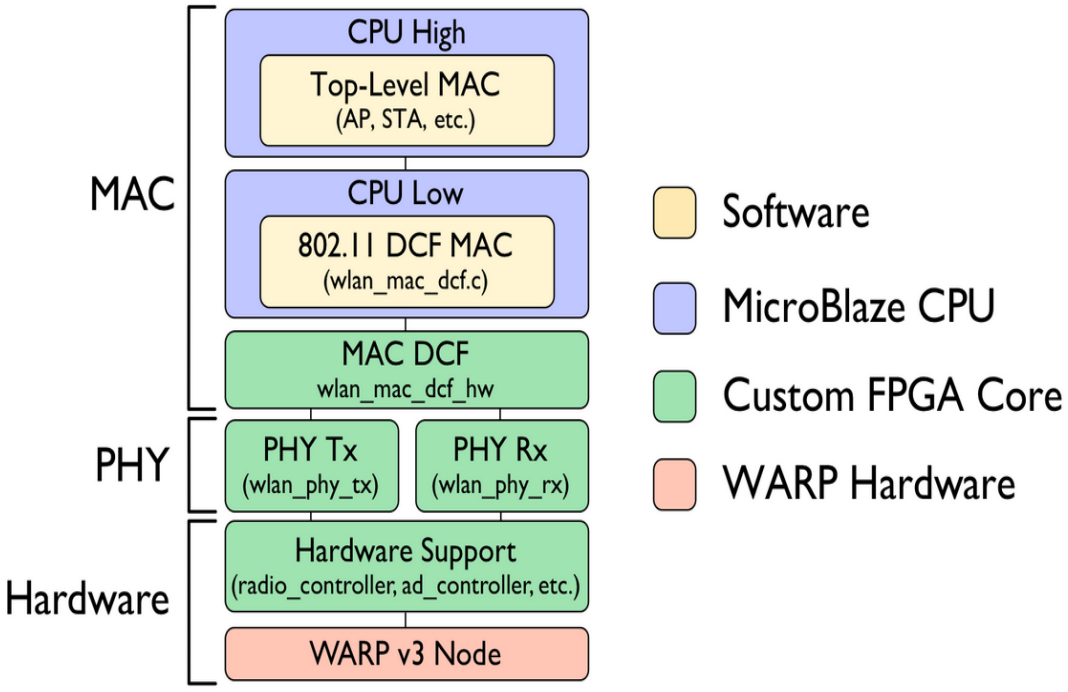

Abbildung 5: Aufbau des Warp 802.11 Reference Designs [Ref. Architecture, 2017]

Abbildung 5 gibt einen Überblick über die Kerne des Referenz Design und zeigt wo diese hinterlegt sind. Die beiden CPU Kerne, CPU High und CPU Low, sind auf 2 Micro Blaze CPUs implementiert. Die Micro Blaze CPU ist ein von Xilinx entworfenes Design um CPUs auf dem FPGA zu konfigurieren, dem entsprechend laufen die beiden CPUs in einem Design im FPGA des Mango v3 Kit. CPU High übernimmt unter anderem das Senden aller Pakete die nicht zum Kontrollfluss gehören, er kontrolliert alle Handshakes mit anderen Knoten und falls Daten über die Ethernetschnittstelle weitergeleitet oder von dieser über das RF Interface gesendet werden sollen, übernimmt dieser Kern das ein- bzw. auspacken. Die CPU Low übernimmt alle Interaktionen zwischen dem MAC und dem PHY, sowie das senden von ACKs, planen von Backoffs, einstellen des Contention Windows und initiiert erneute Übertragungen falls nötig. Der MAC DCF ist direkt auf dem FPGA implementiert und ist die Schnittstelle der beiden PHY Kerne Tx und Rx zur MAC Schicht. In diesem Kern sind unter anderem die Carrier Sense Mechanismen und

die Taktzyklen realisiert. In den PHY Tx und PHY Rx Kernen befindet sich die Implementierungen der physikalischen OFDM Schicht, hier werden Daten umgewandelt und in Signale verpackt bzw. Signale ausgepackt und in Daten gewandelt. In dem letzten Kern, dem Hardware Support, liegen alle Treiber für das Kit. Um genau zu verstehen wie das Referenz Design einen Standard implementiert, lohnt es sich die Umsetzung des 802.11a Standard genauer anzusehen und zu versuchen 2 verschiedenen Warp v3 Kits miteinander kommunizieren zu lassen. Ist der Schritt gelungen kann 802.11a so angepasst werden, dass der PHY und MAC zum 802.11p passen.

4 IEE Standard 802.11p

Author: Philipp Sebastian Schmieder

Um zu verstehen was man beim 802.11a-Standard ändern muss um den 802.11p-Standard nachzubilden, müssen zunächst die Unterschiede zwischen diesen beiden Standards geklärt werden. Hier soll nun der 802.11p-Standard und dessen Aufbau im PHY und MAC erklärt werden.

4.1 Der PHY Layer

Zunächst ist es wichtig zu verstehen wie der PHY bei 802.11p aufgebaut ist, also wie die Kanäle, das Frequenzband und die Modulation aussehen. Das Frequenzband und die Kanäle von 802.11p liegen im 5GHz Bereich. Abbildung 6 gibt einen Überblick über alle Frequenzen und Kanäle, der Kanaltyp ist als CCH oder SCH angegeben. CCH ist Kontrollkanal für Signalisierungs- und Steuerungsinformationen und SCH sind Synchronisationskanäle und dienen z.B. zur Erkennung der Kanalstruktur.

Kanaltyp	Mittenfrequenz	Kanalnummer
G5-CCH	5,9 GHz	180
G5-SCH1	5,880 GHz	176
G5-SCH2	5,890 GHz	178
G5-SCH3	5,870 GHz	174
G5-SCH4	5,860 GHz	172
G5-SCH5	5,910 GHz	182
G5-SCH6	5,920 GHz	184

Abbildung 6: Frequenzen und Kanäle für IEE 802.11p nach [Car2x, 2017]

Im Gegensatz zu 802.11a liegt bei 802.11p die **Bandbreite bei 10MHz**, das Modulationsverfahren ist **OFDM mit 52 Subcarriern** (48 Data Carriers und 4 Pilot Carriers). Die Modulationsarten sind equivalent zu 802.11a. [Car2x, 2017]

4.2 Der MAC Layer

Der Kanalzugriff im MAC Layer erfolgt Ad-hoc, das heißt es gibt keinen einzelnen Access Point und die Teilnehmer konkurrieren um die Kanäle. Die Kollisionskontrolle erfolgt durch Carrier Sense Mutiple Access / Collision Avoidence(CSMA/CA). Der Quality of Service wird durch das prioritätsbasierte Verfahren Enhanced Distributed Channel Access realisiert (EDCA).

5 Umsetzung

Author: Dominik Bayerl

Im Rahmen des Projekts wurden die zwei grundsätzlichen Funktionen umgesetzt, die zur Nutzung der Hardware für weitere Experimente erforderlich sind. Dies ist einerseits die Erweiterung des 802.11-Reference-Designs auf den 802.11p-Standard und andererseits die Implementierung einer Ethernet-Schnittstelle zum Empfang und Senden von Daten über einen angeschlossenen Computer. Auf beide Funktionen soll im Folgenden kurz eingegangen werden.

5.1 WARP Reference Design für 802.11p

Auf die grundsätzliche Funktionsweise von *802.11p* wurde bereits in **Abschnitt 4** eingegangen. Dabei wurde deutlich, dass der Standard sehr ähnlich zum bereits implementierten *802.11a* ist (insbesondere die OFDM-Waveform) und sich vor allem in zwei wesentlichen Merkmalen, den Channel-Frequenzen und der Channel-Bandbreite unterscheidet. Es bietet sich daher an, die Implementierung auf Basis des vorhandenen Frameworks vorzunehmen.

5.1.1 Channel-Frequenzen

Für das Teilnehmer-Multiplexing in WLAN-Funksystemen werden üblicherweise Channels verwendet, d.h. es können mehrere getrennte Funknetze dadurch unabhängig voneinander existieren, indem sie verschiedene Channels und dadurch verschiedene Frequenzen für die Kommunikation nutzen. Im 802.11p Standard sind acht verschiedene Channel-Typen spezifiziert, die für unterschiedliche Aufgaben reserviert sind. Table 1 gibt einen Überblick über die spezifizierten Kanäle (*Draft ETSI EN 302 663 V1.2.0* 2012).

Tabelle 1: 802.11p Channels.

Channel Type	Center frequency	IEEE 802.11 channel number	Channel spacing	Default data rate
G5-CCH	5900 MHz	180	10 MHz	6 Mbps
G5-SCH2	5890 MHz	178	10 MHz	12 Mbps
G5-SCH1	5880 MHz	176	10 MHz	6 Mbps
G5-SCH3	5870 MHz	174	10 MHz	6 Mbps
G5-SCH4	5860 MHz	172	10 MHz	6 Mbps
G5-SCH5	5850 MHz	182	10 MHz	6 Mbps
G5-SCH6	5910 MHz	184	10 MHz	6 Mbps
G5-SCH7	nach IEEE 802.11, 5470 MHz*to*5725 MHz	94 bis 145	verschiedene	abhängig von der Bandbreite

Es wird deutlich, dass eine Erweiterung des verfügbaren Frequenzbandes von 802.11a (5180 MHz*to*5825 MHz) auf 802.11p (5850 MHz*to*5925 MHz) notwendig ist.Die RF-Frequenz wird auf dem Mango WARPv3 Board durch den RF-Transceiver MAX2829 (siehe **Unterabschnitt 2.2**) erzeugt. Dieser kann via SPI durch die Low-CPU des FPGAs konfiguriert werden („max2828/max2829 single-/dual-band 802.11a/b/g world-band transceiver“ 2004). Zur Einstellung der Center-Frequenz des Transceivers sind dabei insbesondere die Register *Band Select and*

PLL, *Integer-Divider Ratio* und *Fractional-Divider Ratio* wichtig. Über das *Band Select* Register wird das Frequenzband (5 GHz) ausgewählt und durch den Vorteiler (engl. Divider) wird die Grundfrequenz des Oszillators durch einen rationalen Teiler (Ganzzahl und Fraktion) auf den gewünschten Wert abgeleitet. Zur Anpassung der verfügbaren Frequenzen ist daher eine Änderung der möglichen Register-Werte des RF-Transceivers notwendig. Im WARP Reference Design erfolgt die Konfiguration des Transceivers durch den radio_controller IP Core. Änderungen der Konfiguration erfolgen also am elegantesten im Treiber des Peripherals. Konkret bedeutet dies, dass in der Datei `edk/pcores/radio_controller.c` Änderungen für drei Lookup-Tables `rc_tuningParams_5GHz_freqs`, `rc_tuningParams_5GHz_reg3` und `rc_tuningParams_5GHz_reg4` notwendig sind, nämlich müssen die Register-Werte für die hinzugefügten Channels hinterlegt werden. Die Berechnung der Werte kann händisch nach („max2828/max2829 single-/dual-band 802.11a/b/g world-band transceiver“ 2004) oder durch das beiligende Python-Skript erfolgen. Anschließend müssen die zusätzlichen Kanäle zur Verwendung “freigeschalten” werden. Dies erfolgt in der Software der Low-CPU.

5.1.2 Channel Bandbreite

Die verwendete Bandbreite des Kanals hängt direkt von der gewählten Sampling-Rate des verwendeten ADC/DAC-Wandlers AD9963 ab. Dieser ist ebenfalls über die SPI-Schnittstelle durch die Low-CPU konfigurierbar, die Implementierung erfolgt über den w3_ad_controller Core. Für 802.11p werden vorrangig Kanäle der Bandbreite 10 MHz verwendet. Diese Bandbreite ist bereits im 802.11 Reference Design implementiert, muss jedoch in der Software der Low-CPU durch einen Aufruf der Funktion `set_phy_samp_rate()` aktiviert werden. Der 10 MHz-Modus wird dabei durch die Konstante `PHY_10M` ausgewählt. Im 802.11p Reference Design wird ist dies im *wlan_mac_low_11p*-Projekt implementiert.

5.2 Validierung der Umsetzung am Spektrometer

Eine Möglichkeit die Implementierung auf der Plattform zur Validierung war die Auswertung des Ausgangssignals an einem Spektrometer. Zunächst wurde das Signal bei Einstellen des 802.11a-Standards gemessen.

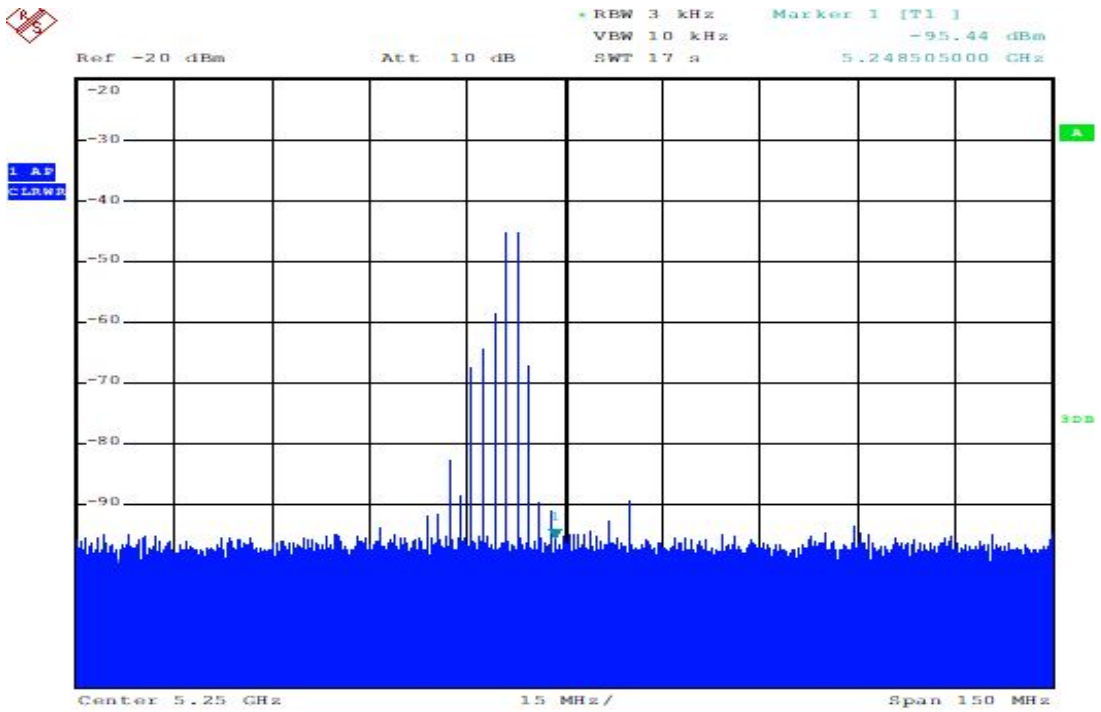

Abbildung 7: Signal des Warp V3 Boards mit 802.11a

In der Auswertung ist gut zu erkennen, dass das Kit die eingestellte Centerfrequenz bei 5,2GHz genau trifft und auch die Bandbreite mit 20MHz passt. Als nächstes wurde das Mango v3 Kit auf den 802.11p-Standard eingestellt und das Ausgangssignal vermessen. Auch hier wird die eingestellte Centerfrequenz bei 5,9 GHz getroffen und auch die Bandbreite stimmt mit 10MHz. Somit konnte verifiziert werden, dass die Plattform den 802.11p PHY implementiert und das Mango v3 Kit standardkonform sendet.

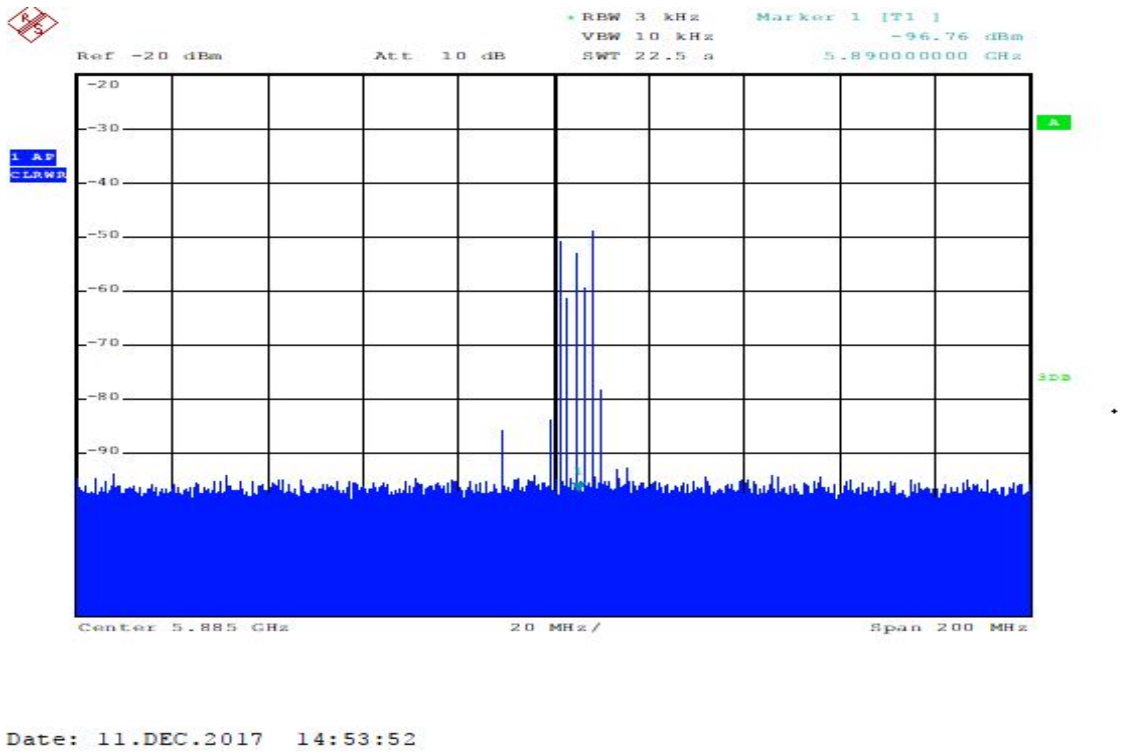

Abbildung 8: Signal des Warp v3 Boards mit 802.11p

5.3 Ethernet-Schnittstelle

Für das Projekt wurde beschlossen, dass die Umsetzung von Funktionalität möglichst auf einem normalen Computer erfolgen soll. Die Gründe dafür sind, dass dort bereits eine Vielzahl von spezialisierten Tools (u.a. Wireshark und PCAP) vorhanden sind, deren Implementierung den Rahmen des Projekts bei weitem sprengen würde. Zusätzlich ermöglicht wird es dadurch einfacher ermöglicht, Fehler in der Software zu debuggen und 3rd-party Komponenten (wie beispielsweise MATLAB) anzubinden. Die Realisierung dieser Design-Ziele erfolgt durch die Instrumentierung des WARP v3 Board über eome Ethernet-Schnittstelle. Darüber können sowohl Daten des WLAN-Kanals empfangen und an einen Computer weitergeleitet, als auch Daten von einem normalen Rechner als 802.11p Frames gesendet werden.

5.3.1 Hardware

Das WARPv3 Board besitzt zwei 1 Gbps-Ethernet-Interfaces. Dabei ist Interface **B** durch das Reference-Design reserviert, um darüber Experimente steuern zu können(Mango Communications 2017). Die Schnittstelle zur Datenübertragung wurde deshalb auf Interface **A** realisiert. Diese kann über ein normales RJ45 Ethernet-Kabel mit einem beliebigen Rechner verbunden werden.

5.3.2 RFtap

Die Übertragung der WLAN-Frames über eine Ethernet-Schnittstelle ist nur möglich, wenn diese vorher in ein entsprechendes Transport-Protokoll verpackt werden. Dies ist dem Umstand geschuldet, dass 802.11-Frames keine gültigen Ethernet(-II)-Frames sind und umgekehrt. Würde das WARP-Board die empfangenen 802.11 Frames vollständig identisch auf die Ethernet-Schnittstelle übertragen, werden die Frames von der Netzwerkkarte des angeschlossenen Rechners verworfen und erreichen dessen Betriebssystem bzw. Anwendungen erst gar nicht. Es wurden zwei verschiedene Protokolle zur Übertragung von 802.11 Frames über die Ethernet-Schnittstelle evaluiert: - radiotap, das spezialisiert ist auf "[...] 802.11 frame injection and reception"(„radiotap", o. J.) - RFtap, ein Protokoll "[...] designed to provide Radio Frequency (RF) metadata about packets"(„rftap", o. J.) Für beide Protokolle existiert eine gute Unterstützung in bestehenden Netzwerk-Analyse Tools wie Wireshark. Im Rahmen des Projekts wurde das *RFtap* Protokoll in die High-CPU Software implementiert. Die Gründe dafür sind die einfachere Implementierung gegenüber radiotap und die erweiterte Funktionalität. Da ein RFtap-Frame verschiedenste Payload-Pakete verpacken kann, ist es unter anderem auch möglich, damit radiotap-Frames zu übertragen. RFtap kann folglich als Obermenge von radiotap angesehen werden. Zusätzlich ist es mit RFtap Möglich, die empfangenen Pakete um weitere Informationen (insbesondere physikalische Parameter) zu annotieren. fig. 9 zeigt schematisch den Aufbau eines RFtap-Frames. Für die direkte Verbindung zwischen dem WARPv3 Board und einem Computer sind die verwendeten MAC- und IP-Adressen unkritisch, da auf den Interfaces in die-

sem Fall keine Filterung stattfindet. In der derzeitigen Implementierung sind die Felder daher leer (Wert 0). Für künftige Projekte wäre es denkbar, diese Informationen sinnvoll zu befüllen. Dies ermöglicht beispielsweise ein Routing von RFtap-Frames über gewöhnliche, kommerzielle Netzwerk-Hardware. In Wireshark sind RFtap-Dissectors für UDP-Frames auf Destination-Port **52001** implementiert. Erkannt werden die Frames durch die Magic Numbers `0x52 0x46 0x74 0x61` (ascii: RFta) zu Beginn des Frames. Es folgt die Länge (unsigned integer, 2 Byte) des RFtap-Headers (ohne Datenteil!) in 32-bit Words und ein Flags-Bitfield (2 Byte), das die nachfolgenden Header-Flags spezifiziert(„specifications“, o. J.). Dabei können (aufgrund des Längenfelds) beliebige zusätzliche Felder an das Ende des Headers angefügt werden, die dann jedoch nicht durch einen Dissector abgedeckt werden können. Der RFtap-Frame endet mit dem RF-Payload. Besonders interessant ist die Angabe des DLT-Felds, da dadurch der Typ der Nutzdaten spezifiziert wird. Bei korrekter Angabe nutzt Wireshark dann automatisch den richtigen Dissector um die Payload zu analysieren (beispielsweise LINKTYPE_IEEE802_11, IEEE 802.11 wireless LAN Frame („link-layer header types | tcpdump/libpcap public repository“ 2017)). Das Senden von Frames von einem Rechner erfolgt analog, in umgekehrter Reihenfolge.
[FPGA]: Field Programmable Gate Array [SPI]: Serial Peripheral Interface *[RF]: Radio Frequency* [ADC]: Analog Digital Converter *[DAC]: Digital Analog Converter* [DLT]: Data Link Type

6 Optimierungsmöglichkeiten und weitere Ideen

Author: Dominik Bayerl

Zum derzeitigen Zeitpunkt bestehen noch offene Punkte zur weiteren Optimierung der Software, die aufgrund des beschränkten zeitlichen Rahmes des Projektes nicht mehr umgesetzt werden konnten. Dabei handelt es sich größtenteils um Unschönheiten und Performance-Maßnahmen in der Software der High-CPU (Sniffer-Applikation), die jedoch nicht die grundsätzliche Funktion einschränken.

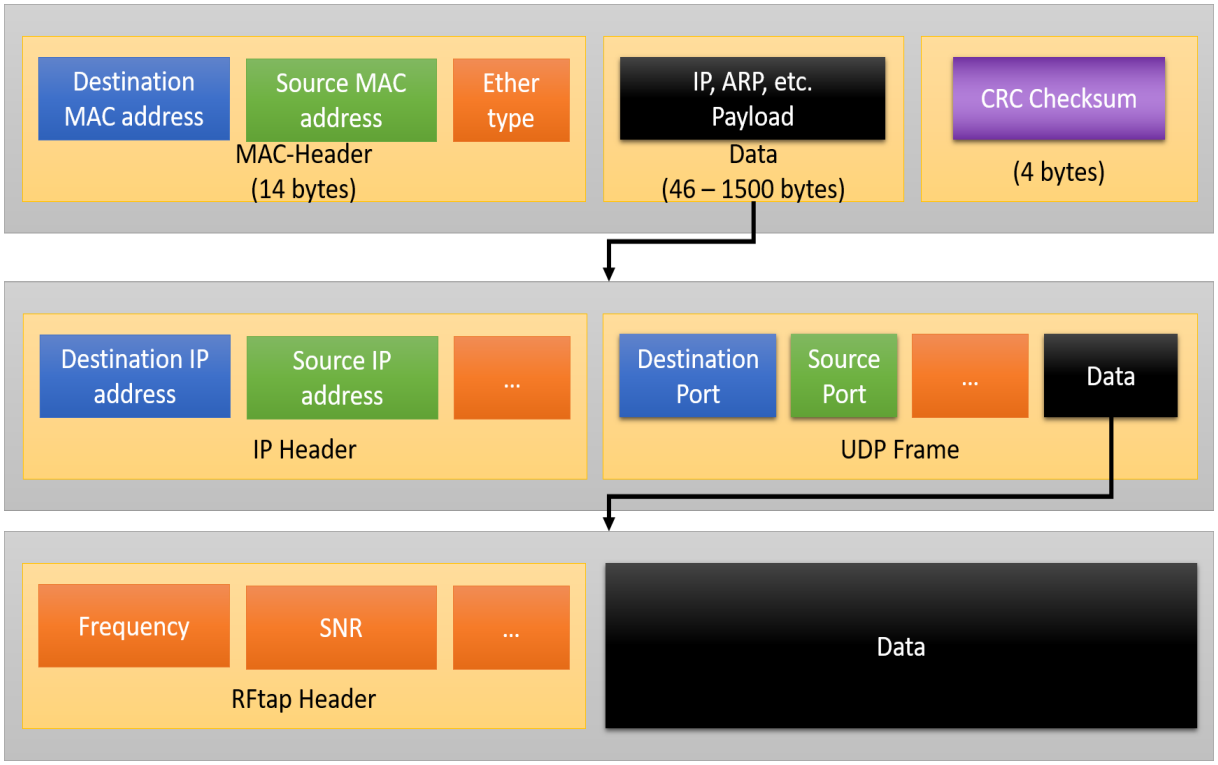

Abbildung 9: RFtap Frame Aufbau.

Einige Verbesserungen sollen im Folgenden knapp skizziert werden.

6.1 Optimierung des IP-Stacks

Der IP-Stack wird immer dann benötigt, wenn ein Paket vom Wireless auf das LAN-Interface übertragen wird und umgekehrt. Beispielsweise ist es zur Verpackung der WLAN-Frames notwendig, diese in das RFtap-Format zu bringen, wobei dieses aus einem UDP-Frame besteht.
Derzeit ist dies in der Datei `wlan_mac_high_sniffer/rftap.c` als Chainable-Funktionen implementiert. Dies bedeutet, dass die einzelnen Bestandteile des Ethernet-Frames stückweise konstruiert und dem Buffer hinzugefügt werden. Dadurch, dass der Buffer front-alloziert ist (d.h. es ist lediglich die Start-Adresse und die Länge des Buffers bekannt) ist es nicht möglich, die Header der Frames direkt dem Beginn des Buffers hinzuzufügen, da andernfalls der Datenbereich des Frames überschrieben werden würde.
Um dies zu umgehen werden alle Header in einem separaten Buffer abgelegt und

anschließend der Datenteil an das Ende der Header kopiert (Funktion `mpdu_rx_process()` in `wlan_mac_high_sniffer/wlan_mac_sniffer.c`). Der Kopiervorgang ist dabei potentiell ein Performance-Flaschenhals. Die Notwendigkeit eines einzelnen Buffers der den kompletten Ethernet-Frame enthält, ergibt sich durch die derzeitige Verwendung des Ethernet-Interfaces des FPGAs im einfachen DMA-Modus. Die tatsächliche Übertragung der Daten auf die Ethernet-Schnittstelle erfolgt anschließend ohne weitere Beteiligung der CPU durch das Ethernet-Peripheral. Im erweiterten DMA-Modus bietet der Ethernet-IP-Core die Möglichkeit der Datenübertragung zur Ethernet-Schnittstelle aus verschiedenen Speicherbereichen. Dieses Konzept wird bei Xilinx als "Scatter-Gather-DMA" (*grobe Übersetzung*: Verstreutes-Sammeln-DMA) bezeichnet („logicore ip axi ethernet (v3.01a)" 2012). Die Funktionsweise besteht darin, dass der DMA-Schnittstelle nicht mehr die Buffer-Adresse und deren Länge übergeben wird, sondern die Adresse eines sogenannten "Buffer Descriptors". Diese Datenstruktur besteht unter anderem aus der Buffer-Adresse und einem Längenfeld (siehe fig. 10). Sobald das DMA-Peripheral alle Daten aus dem im Buffer Descriptor referenzierten Buffers übertragen wird, wird ein Interrupt an die CPU ausgelöst.
Der Vorteil dieses Verfahrens besteht darin, dass eine zweite Indirektionsschicht eingeführt wird; dadurch ist es nicht mehr notwendig, dass sämtliche DMA-Daten sequentiell im Speicher liegen. Üblicherweise implementiert man dazu eine Single-Linked-List (FIFO Queue) aus Buffer-Descriptors, die nach jedem DMA-Interrupt weitergeschalten wird. Für den konkreten Fall der Ethernet-Frames ermöglicht dieses Verfahren, die einzelnen Header-Bestandteile unabhängig im Speicher ablegen zu können. Dadurch ist ein echter Zero-Copy Modus - also ohne Daten kopieren zu müssen - möglich.

6.2 Nutzung der CFO-Estimates

Im Empfangspfad des WARPv3 existiert bereits ein Block zur Korrektur eventuell vorhandener Frequenzabweichungen der Sender bzw. des Empfängers (Abschnitt 3). Dabei wird mittels des LTS-Feldes über mehrere Frames die Frequenz

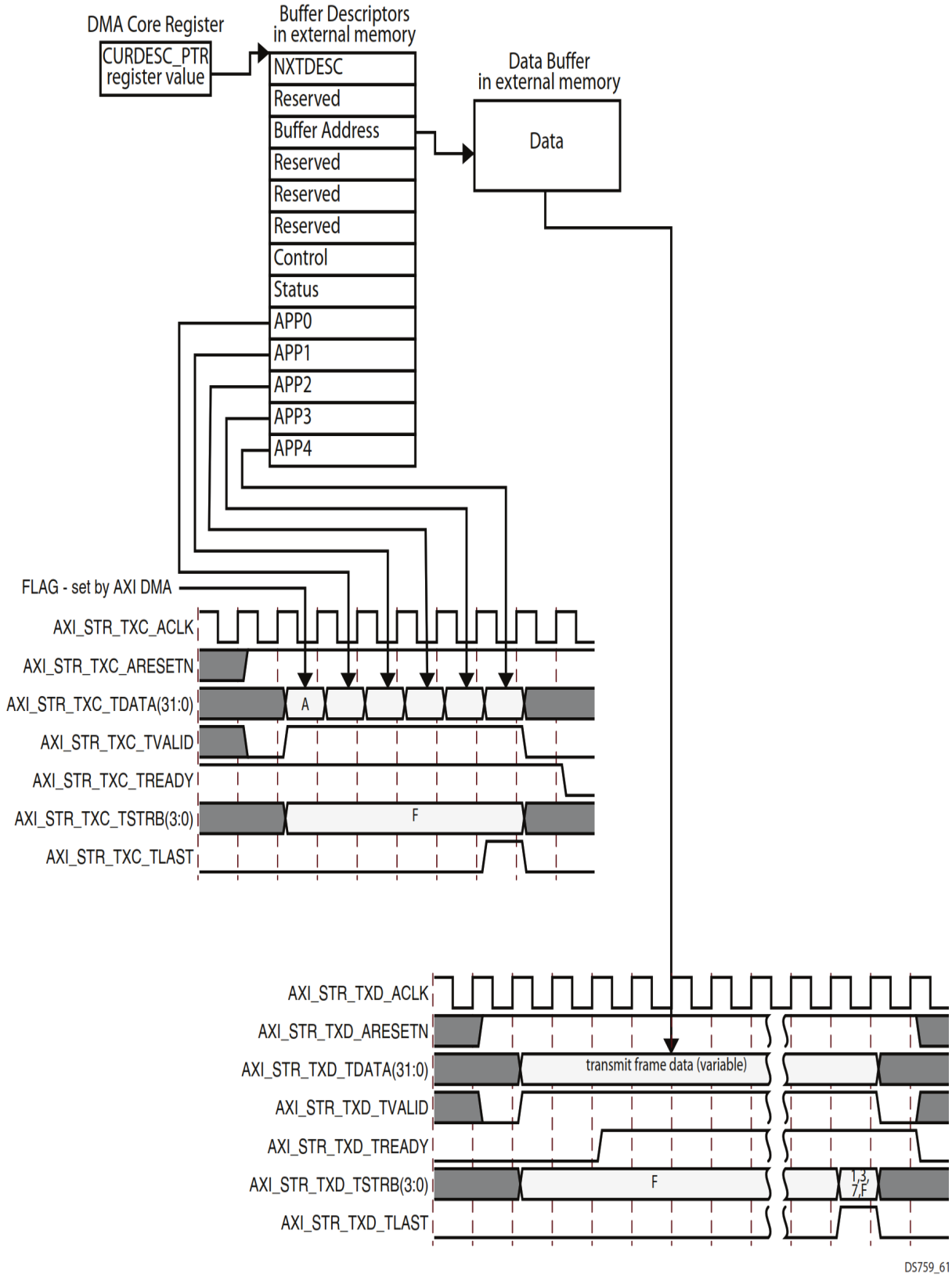

Abbildung 10: Xilinx DMA Buffer Descriptors.

des empfangenen Signals gegenüber der Center-Frequenz des gewählten Channels bestimmt und anschließend zur Korrektur der Fourier-Transformation der einzelnen Carriers genutzt. In den meisten kommerziellen WiFi-Transceivern wird die Informationen anschließend verworfen, da sie für die darüberliegende Schicht (Layer 2, MAC Layer) nicht benötigt wird. Nicht so im WARP Reference Design: die geschätzte CFO wird durch die Low-CPU an die High-CPU in der Methode `mpdu_rx_process()` in der Datei `wlan_mac_high_sniffer/wlan_mac_sniffer.c` als Feld `cfo_est` der Struktur `rx_frame_info_t` übergeben. Gleichermaßen wird die Information bereits durch die Sniffer-Applikation in den RFtap-Frames an die Ethernet-Schnittstelle übertragen (*Flag 3, Frequency offset field is present*). Dies ermöglicht die Auswertung der CFO-Estimates beispielsweise im Wireshark eines angeschlossenen Computers. Die Information ist besonders deswegen interessant, da sie (unter anderem) für zwei Anwendungsfälle genutzt werden kann, die im Folgenden dagelegt werden sollen:

6.2.1 Doppler-Effekt

Wie jedes elektromagnetische Signal unterliegen auch die 802.11p-WLAN Signale dem Doppler-Effekt. Dieser besagt, dass ein Signal der Frequenz f_S das von einem Sender S an einen Empfänger B derart übertragen wird, dass Sender und Empfänger eine Relativgeschwindigkeit $v_S - v_B = v \neq 0$ besitzen, beim Beobachter eine Frequenzabweichung $f_B = \frac{f_S}{\gamma} = f_S\sqrt{1-\frac{v^2}{c^2}} \approx f_S\left(1-\frac{v^2}{2c^2}\right)$ erfährt. Diese Frequenzabweichung muss durch den Empfänger detektiert und kompensiert werden. Für die Anwendung WLAN wird dies durch die Korrektor der CFO übernommen. Hat ein Empfänger nun Kenntnis über den statischen CFO (bedingt durch ungenaue Oszillatoren) eines Senders, kann er durch die Messung des aktuellen CFOs eine dynamische Frequenzabweichung bestimmen, bei der der Doppler-Effekt eine nicht unerhebliche Rolle spielt. Für 802.11p bedeutet dies, dass zwei Fahrzeuge, die miteinander im Funkkontakt stehen, ihre gegenseitige Relativgeschwindigkeit ohne Zusatzhardware über die WLAN-Schnittstelle bestimmen könnten. Dies ermöglicht eine Reihe weiterer Funktionen, wie Notbremsassistenten, Adaptive

Tempomaten und ähnliches.

6.2.2 PHY-Fingerprinting

In einer separaten Teilgruppe des Projektes wurde ein Verfahren zur Manipulation von Funknetzen, sog. MAC-Spoofing und Evil-Twin-APs evaluiert. Die Verfahren basieren darauf, dass die Merkmale die zur Identifikation der Funkteilnehmer verwendet werden, sehr leicht manipulierbar sind (MAC-Adresse bzw. SSID/BSSID). Für nicht weiter kryptographisch gesicherte Netzwerke (Offene WLANs) stellen diese Attacken ein erhebliches Sicherheitsrisiko dar, da dadurch eine Reihe weiterer Angriffe (Man-in-the-middle, Phishing, ARP-Spoofing, ...) ermöglicht werden. Das Mango-Board bietet gegenüber kommerzieller WLAN-Hardware die Möglichkeit, sämtliche Parameter der Funkübertragung zu erfassen, insbesondere auch die des physikalischen Layers, beispielsweise in Form der Center-Frequency-Offsets. Diese Parameter sind unabhängig von den gesendeten Daten, sondern werden ausschließlich durch die RF-Charakteristik der Hardware des Senders beeinflusst und eignen sich dadurch als Merkmal zur Identifikation eines einzelnen Sende-Moduls. Durch ein geeignetes Fingerprinting-Verfahren über mehrere verschiedene Merkmale (CFO-Estimates, Signal-Power, Noise-Power) kann dadurch eine Zuordnung anderer Merkmale (MAC-Adresse, SSID) zu einem physikalischen Sender geschaffen werden. Dadurch wird es ermöglicht, oben genannte Angriffe erkennen zu können, da im Falle eines vorhandenen Angreifers zwei verschiedene Sendemodule (mit unterschiedlichen Fingerprints) die selben High-Level Merkmale (MAC-Adresse, SSID) nutzen würden.(„using rftap to detect mac spoofing“ 2016)

6.2.3 Linux Kernel

Im Verlauf des Projekts zeigte sich, dass der Ansatz des 802.11 Reference Designs als Bare-Metal Software (d.h. ohne Betriebssystem) mehrere Schwächen besitzt: eine Iteration der entwickelten Software bedingt stets eine komplette Neuprogrammierung des FPGA-Designs. Desweiteren ist es nicht ohne weiteres möglich,

Konfigurationsparameter (Channel, Baseband, ...) während des Betriebs anpassen zu können. Diese Funktion wurde zwar rudimentär über eine UART-Konsole eingebaut, hat jedoch Schwächen in der Bedienbarkeit und Robustheit. Weitere fehlende bzw. nur im Ansatz vorhandene Funktionen sind eine Debugging-Schnittstelle (`xil_printf()` über die UART-Konsole), ein Scheduler (Scheduling auf der CPU-High implementiert, non-preemptive round-robin mit Auflösung im Millisekunden-Berich) und die Möglichkeit zur Nutzung der von Xilinx bereitgestellten Peripheral-Treiber (insbesondere die Ethernet-Schnittstelle). Diese offenen Punkte können durch den Einsatz eines Betriebssystems gelöst werden. Xilinx bietet bereits einen an die MicroBlaze-Architektur angepassten Linux Kernel („xilinx wiki - microblaze“, o. J.) an. Zusätzlich sind für die meisten IP-Cores Linux-Treiber vorhanden, die einfach integriert werden können („xilinx wiki - linux drivers“, o. J.).
Ungelöst ist dabei die Problematik der Treiber für benutzerdefinierte Peripherals - insbesondere für die *radio_controller*, die zentraler Bestandteil des WARP Reference Designs sind. Hier ist eine Anpassung der standalone-Treiber an die Schnittstelle des Linux-Kernels notwendig.
[CFO]: Center Frequency Offset [DMA]: Direct Memory Access *[FPGA]: Field Programmable Gate Array* [FIFO]: First in First out *[IP-Core]: Intellectual Property Core* [LTS]: Long training sequence *[PLL]: Phase-locked Loop* [RF]: Radio Frequency

Abbildungsverzeichnis

Tabellenverzeichnis

Literatur

[ABOUT WARP, 2017] http://warpproject.org/trac/wiki/about

[MANGO v3, 2017] http://mangocomm.com/static/img/w3/w3_ kit_ med.jpg , Herausgeber des Wissens-Portals: DATACOM Buchverlag GmbH, 2017 (letzter Aufruf 08.07.2017)

[v3 Design, 2017] https://warpproject.org/trac/wiki/HardwareUsersGuides/WARPv3, Herausgeber des Wissens-Portals: DATACOM Buchverlag GmbH, 2013 (letzter Aufruf 08.07.2017)

[v3 Kit Clocking, 2017] https://warpproject.org/trac/wiki/HardwareUsersGuides/WARPv3/C © 2017 RUS-CERT, Universität Stuttgart, 2017 (letzter Aufruf 08.07.2017)

[Ref. Architecture, 2017] http://warpproject.org/trac/attachment/wiki/802.11/files/wlan_ ref_ des_ arch.png, © 2017 RUS-CERT, Universität Stuttgart, 2017 (letzter Aufruf 08.07.2017)

[Car2x, 2017] https://warpproject.org/trac/wiki/HardwareUsersGuides/WARPv3/C © 2017 RUS-CERT, Universität Stuttgart, 2017 (letzter Aufruf 08.07.2017)

Draft ETSI EN 302 663 V1.2.0. 2012. ETSI.

„link-layer header types | tcpdump/libpcap public repository“. 2017. *tcpdump.org*. http://www.tcpdump.org/linktypes.html.

Mango Communications, Inc. 2017. „Mango 802.11 Reference Design Experiments Framework Documentation“. https://warpproject.org/docs/mango-wlan-exp/.

„max2828/max2829 single-/dual-band 802.11a/b/g world-band transceiver“. 2004. *datas-*

heets.maximintegrated.com. https://datasheets.maximintegrated.com/en/ds/MAX2828-MAX2829.pdf.

„radiotap“. o. J. *radiotap.org*. http://www.radiotap.org/.

„rftap“. o. J. *rftap.github.io*. https://rftap.github.io/.

„specifications“. o. J. *rftap.github.io*. https://rftap.github.io/specifications/.

„logicore ip axi ethernet (v3.01a)“. 2012. *Xilinx.com*. https://www.xilinx.com/support/documentation/ip_documentation/axi_ethernet/v3_01_a/ds759_axi_ethernet.pdf.

„using rftap to detect mac spoofing“. 2016. *Rftap.github.io*. https://rftap.github.io/blog/2016/09/01/rftap-wifi.html.

„xilinx wiki - linux drivers“. o. J. *Wiki.xilinx.com*. http://www.wiki.xilinx.com/Linux+Drivers.

„xilinx wiki - microblaze“. o. J. *Wiki.xilinx.com*. http://www.wiki.xilinx.com/MicroBlaze#x-MicroBlaze%20Linux.