

Adaptives Waterfilling und Verstärkungsregelung der OFDM
Datenübertragung einer Geothermie-Bohrlochsonde bei
unterschiedlichen Kabelarten

BACHELORARBEIT

für die Prüfung zum

Bachelor of Engineering

des Studienganges Elektrotechnik

an der

Dualen Hochschule Baden-Württemberg Karlsruhe

von

Fabian Michaelsen

Abgabedatum 21. August 2017

Bearbeitungszeitraum	12 Wochen
Matrikelnummer	213193607
Kurs	TEL14B2/NT
Ausbildungsfirma	Karlsruher Institut für Technologie (KIT)
Betreuer der Ausbildungsfirma	Dipl. Ing. Stefan Dietze
Gutachter der Studienakademie	Prof. Dr.-Ing. Gerald Oberschmidt

Erklärung

Ich versichere hiermit, dass ich meine Bachelorarbeit mit dem Thema: „Adaptives Waterfilling und Verstärkungsregelung der OFDM Datenübertragung einer Geothermie-Bohrlochsonde bei unterschiedlichen Kabelarten“ selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Ich versichere zudem, dass die eingereichte elektronische Fassung mit der gedruckten Fassung übereinstimmt.

Ort Datum

Unterschrift

Zusammenfassung

Im Rahmen dieser Bachelorarbeit soll ein Verfahren zur automatischen Kanaladaption der OFDM-Datenübertragung an unterschiedliche Kabelcharakteristika entwickelt werden. Dieses Adaptionverfahren soll bei der Teleinspektion verschiedener Bohrlochsonden verwendet werden. Es werden, nach dem Prinzip der pre-equalization, die Waterfillingkoeffizienten ermittelt und so das Leistungsspektrum notwendigerweise optimiert. Die Realisierung erfolgt mittels des Systemgenerator von Xilinx[™] unter Matlab-Simulink. Das Ziel ist die Implementierung in verschiedene FPGAs.

Abstract

Within the framework of this Bachelor thesis, a method for automatic channel adaptation of the OFDM data transmission to different cable characteristics is to be developed. This adaptation method is used in the tele-inspection of different borehole probes. The waterfilling coefficients are determined through the principle of pre-equalization, thus optimizing the power spectrum. The realization is carried out using the Xilinx[™]-Systemgenerator. The goal is implementation in various FPGAs.

Inhaltsverzeichnis

1	GeoKam - Projekt	1
1.1	Geothermie	2
1.2	Mechanik der GeoKam-Sonde	2
1.3	HiTES - Elektronik der GeoKam-Sonde	3
1.4	Ziel dieser Arbeit	4
2	Das Datenübertragungsverfahren (OFDM)	6
2.1	Kanalkodierung	7
2.2	Leitungskodierung	7
2.3	IQ-Mapper	8
2.4	Bandpasstransformation und Kanalmapping	8
2.5	(Inverse) Fast Fourier Transformation	9
2.6	ISI- und ICI-Freiheit bei OFDM	9
2.7	Framebuilder	10
2.8	Zusammenfassung	11
2.9	Besonderheiten des Empfängers	11
2.9.1	Framestart Detektion	11
2.9.2	Frequenzoffset Korrektur	12
2.9.3	Kanal Equalizer	12
3	Übertragungsstrecken	13
3.1	Kabelbeanspruchung	13
3.2	Pegelabschätzungen bei Kabel Wireline4k	13
3.3	Vergleich von Wireline4k und Testkabel500	15
4	Spektrale Vorverzerrung und Waterfilling	16
4.1	Waterfilling und Pre-equalization	16

Inhaltsverzeichnis

4.2	Überlegungen zur Kanalkapazität	18
5	Kanaladaptionsverfahren	19
5.1	AGC für den Steuerkanal ChDn - Phase 1	20
5.2	AGC für den Datenkanal ChUp - Phase 2	21
5.3	Verschiedene Methoden der Kanalvermessung - Phase 3	22
5.3.1	Sinc-Funktion oder Diracstoß	23
5.3.2	Kreuzkorrelation mit weißem Rauschen	24
5.3.3	Bandbegrenztes weißes Rauschen	24
5.3.4	Das Stepped Sub Carrier Sweep (SSCS)-Verfahren	25
6	Realisierung	26
6.1	AGC des Steuerkanal Empfängers	26
6.2	AGC des Datenkanal Empfängers	26
6.2.1	Fenster Diskriminator (Window Detektion)	27
6.2.2	Right Gain Detektion	29
6.2.3	Vorläufige Kommandogenerierung	30
6.2.4	Kommandogenerierung	30
6.2.5	Simulationsergebnis	31
6.2.6	Schnittstelle	32
6.3	Das SSCS-Verfahren	32
6.3.1	Timing	33
6.3.2	Anpassung an die FFT	33
6.3.3	Auslesen der FFT	34
6.3.4	Ablaufsteuerung	35
6.3.5	Statemachine	36
6.3.6	Koeffizientengenerierung	36
6.3.7	DDS im Sender	37
6.3.8	Simulation und Ergebnis	38
6.3.9	Implementierung in den vorhandenen Empfänger	39
6.4	Datenkanal Sender	39
6.4.1	Ablaufsteuerung	40
6.4.2	Waterfillingkoeffizienten laden	41
6.4.3	IFFT Skalierung	41

Inhaltsverzeichnis

6.4.4	Header generieren	43
6.4.5	Header auslesen	44
6.4.6	Simulationsergebnis	45
7	Implementierung	48
7.1	AGC des Steuerkanal Empfängers	48
7.2	Datenkanal Sender	48
7.3	Datenkanal Empfänger	49
7.4	SSCS-Koeffizientengenerierung	49
8	Fazit	51
	Literaturverzeichnis	52
9	Anhang	55

Abbildungsverzeichnis

2.1	Signalflussdiagramm des ChUp Tx	6
3.1	Vergleich der Strecken	15
4.1	Schematische Darstellung der Generierung von Waterfillingkoeffizienten	16
5.1	Kanaladaptionsverfahren	19
5.2	Zustandsfolgediagramm der AGC ChUp	21
6.1	Blockschaltbild der AGC ChUp in Simulink TM	27
6.2	Blockschaltbild des <i>WD</i> -Blocks in Simulink TM	28
6.3	Blockschaltbild des <i>RG</i> -Blocks in Simulink TM	29
6.4	Blockschaltbild des <i>get_pre_C</i> -Blocks in Simulink TM	30
6.5	Blockschaltbild des <i>get_C</i> -Blocks in Simulink TM	31
6.6	SSCS-Verfahren	32
6.7	Blockschaltbild des <i>get_value(i)</i> -Blocks in Simulink TM	34
6.8	Blockschaltbild der <i>Statemachine</i> -Blocks in Simulink TM	36
6.9	Blockschaltbild der Koeffizientengenerierung in Simulink TM	37
6.10	Blockschaltbild der DDS im Tx in Simulink TM	37
6.11	Vergleich der Referenz und SSCS generierten Waterfillingkoeffizienten bei SNR=25 dB	38
6.12	Blockschaltbild der Steuerung des Tx in Simulink TM	40
6.13	Blockschaltbild der IFFT-Skalierungsberechnung in Simulink TM	41
6.14	Blockschaltbild des ROM Speichers des Headers des Tx in Simulink TM	43
6.15	Blockschaltbild des RAM Speichers des Headers des Tx in Simulink TM	44
6.16	Blockschaltbild des <i>search_m</i> -Blocks in Simulink TM	44
6.17	Vergleich eines Tx Frames der HHI-Matlabskript Simulation und der SSCS Simulink Simulation	46

Abbildungsverzeichnis

6.18	Exemplarische FFT-Ausgabe im Rx der Präambel	47
7.1	AGC ChDn in Aktion (aufgenommen mit Chipscope™)	48
7.2	Blockschaltbild der veränderten DDS in Simulink™	49
7.3	Vergleich der Hardwareimplementierung und der Simulation	50
9.1	Blockschaltbild der AGC ChDn aus Simulink™	55
9.2	Präambel des ChUp in IQ-Ebene	56
9.3	Simulationsergebnis der AGC des ChUp	56
9.4	Blockschaltbild des SSCS-FFT-Blocks im Rx aus Simulink™	57
9.5	Blockschaltbild des FFT-Blocks aus Simulink™	58
9.6	Blockschaltbild des Control-Blocks aus Simulink™	59
9.7	Blockschaltbild der SSCS-Verfahrensgesamtsimulation aus Simulink™	60
9.8	Blockschaltbild des Senders aus Simulink™	61
9.9	Blockschaltbild des Waterfillingmoduls aus Simulink™	62
9.10	Blockschaltbild des gesamtem Rx aus Simulink™	63

Abkürzungsverzeichnis

ADC	Analog Digital Wandler	4
AGC	Automatic Gain Control	4
AWGN	Additives weißes Gaußsches Rauschen	
BER	Bitfehlerrate	14
ChDn	Steuerkanal Host-Sonde	4
ChUp	Datenkanal Sonde-Host	4
DAC	Digital Analog Wandler	4
DC	Gleichstrom	13
DDS	Direct Digital Synthesis	25
DFT	Diskrete Fourier Transformation	9
DOCSIS	Data Over Cable Service Interface Specification	17
FAmp	Filter- und Verstärkerplatine	4
FDM	Frequenzmultiplexverfahren	17
FFT	Fast Fourier Transformation	9
FIR	Filter mit endlicher Impulsantwort	
FPGA	Field Programmable Gate Array	3
FSD	Frame Start Detektion	22
HDR	Hot-Dry-Rock	2
HHI	Fraunhofer-Institut für Nachrichtentechnik, Heinrich-Hertz-Institut	1
HiTES	High Temperature Embedded Systems	1
IAI	Institut für Angewandte Informatik	1

Abbildungsverzeichnis

ICI	Inter Carrier Interferenz.....	10
IDFT	Inverse Diskrete Fourier Transformation.....	9
IFFT	Inverse Fast Fourier Transformation.....	8
ISI	Intersymbolinterferenz.....	10
KIT	Karlsruher Institut für Technologie.....	1
KKF	Kreuzkorrelationsfunktion.....	12
LED	Leuchtdiode.....	3
MAV	Moving-Average	
OFDM	Orthogonal Frequency-Division Multiplexing.....	4
OP	Operationsverstärker	
QAM	Quadraturamplitudenmodulation.....	8
RAM	Random-Access Memory.....	39
RMS	Quadratischer Mittelwert	
ROM	Read-Only Memory.....	24
RS	Reed Solomon.....	3
Rx	Empfänger.....	9
SC	Sub Carrier.....	9
SDR	Software Defined Radio.....	3
SNR	Signal-Rausch-Verhältnis.....	13
SSCS	Stepped Sub Carrier Sweep.....	28
TP	Tiefpass.....	27
Tx	Sender.....	4
VHDL	Very High Speed Integrated Circuit Hardware Description Language.....	3
WF	Waterfilling.....	32
WFC	Waterfilling Koeffizienten.....	4

1 GeoKam - Projekt

Aktuell soll sich die Art und Weise der Energieerzeugung ändern. Es wird versucht, von fossilen Energieträgern wie Braunkohle, Steinkohle, Erdgas und Mineralöl auf nachhaltige Energieträger, wie Wind- und Wasserkraft, Biomasse, Photovoltaik und Geothermie umzustellen. Auch Kernkraft erfreut sich nach den Havarien in Tschornobyl und Fukushima keiner großen Beliebtheit mehr und soll abgeschafft werden. Bis 2050 sollen 80% des deutschen Energiebedarfs mittels erneuerbarer Energien gedeckt werden. (Siehe: [5])

Die Photovoltaik ist nach dem Stand der aktuellen Technik in deutschen Breitengraden nicht oder eher weniger effizient, außerdem ist sie, wie Windenergie nicht grundlastfähig. Das Potential der Wasserkraft und Biomasse ist in Deutschland beinahe ausgeschöpft, was sich laut der Studie [23] auch im Vergleich der Jahre 2013 und 2014 niederschlägt. So ist der Anteil der Wasserkraft an den erneuerbaren Energien um 10,9% zurück gegangen, der Biomasseanteil stieg bloß um 4,5%. Der Anteil der geothermischen Energie verdoppelte sich hingegen und das Potential ist noch kaum ausgeschöpft. Außerdem ist Geothermie im Gegensatz zu Photovoltaik und Windkraft absolut grundlastfähig. Für die Inspektion von Geothermiebohrungen werden Sonden benötigt, welche ein Livevideo zur Inspizierung der Bohrlochinnenwand und zugehörige Messdaten zu einer Hoststation überträgt. Eine Sonde dieser Art wird vom Institut für Angewandte Informatik (IAI), einem Institut des Karlsruher Institut für Technologie (KIT), in Zusammenarbeit mit dem Fraunhofer-Institut für Nachrichtentechnik, Heinrich-Hertz-Institut (HHI) und der Brunnen-Regenerierungs und Brunnenprüfdienst GmbH (BRG) mit Hilfe der Förderung durch das *Bundesministerium für Wirtschaft und Energie* entwickelt und trägt den Namen *GeoKam*. Die Sonde ist ein Baustein des *ZWERG*-Projektes, welches ein Zusammenschluss bei der Entwicklung von Inspektionswerkzeugen für Geothermiebohrungen bildet. (Siehe: [7]) Vor allem der Kommunikationsbaustein soll in vielen derartigen Projekten Verwendung finden. Die verwendete Wireline, also das Kabel, an welchem die Sonde befestigt ist, ist bei der Firma BRG in Verwendung und kommerziell erhältlich. Die hochtemperaturtaugliche Elektronik der Sonde heißt High Temperature Embedded Systems (HiTES) und wird in Sektion 1.3

beschrieben.

1.1 Geothermie

Geothermie ist die in der Erde gespeicherte Wärmeenergie, also Erdwärme. Nur 0,1% des Erdvolumens sind kälter als 100 °C. Es geht also darum, die für Dampfkraftwerke verwendbaren 99,9% des Erdvolumens zu nutzen. Die durchschnittliche Oberflächentemperatur der Erde beträgt 14 °C und steigt durchschnittlich alle 100 m Tiefe um 3 °C an. (Siehe: [1])

Ab einer Tiefe von 400 m spricht man von „Tiefen Geothermie“. Zur kommerziellen Energieerzeugung wird je nach Beschaffenheit das so genannte Hot-Dry-Rock (HDR)-Verfahren bei Tiefen von 3 bis 6 km eingesetzt.

Es wird laut [4] über eine oder mehrere so genannte *Injektions-* bzw. *Verpressbohrungen* Wasser unter hohem Druck in das wenig permeable Gestein eingepresst, um Fließwege aufzubrechen oder vorhandene aufzuweiten und so den Weg zu der so genannten *Produktions-* oder *Förderbohrung* frei zu machen. Existiert eine permeable Gesteinsschicht in entsprechender Tiefe, so kann auf ein Aufbrechen des Fließweges verzichtet werden. Über den Wärmetauscher im Gestein aus vielen kleinen Fließwegen des Wassers kann die Erdwärme abtransportiert und mittels eines Dampfkraftwerks genutzt werden. Für die extremen Umgebungsbedingungen solcher Bohrungen wurde die *GeoKam* entwickelt.

Das Verfahren der Fließwegbildung nennt man *hydraulische Stimulation* oder *Fracking*, welches medial kontrovers diskutiert wird.

1.2 Mechanik der GeoKam-Sonde

Die Sonde mit einem Durchmesser von 95 mm und einer Länge von 2,4 m (Siehe: [20]) ist an der 4,2 km langen Wireline (Siehe: [19]) für eine Operationstiefe von etwa 4 km, abhängig von der Höhe der Wassersäule und dem Temperaturgradienten des Bohrlochs, ausgelegt. Dort wird von einer maximalen Umgebungstemperatur von 165 °C und einem Druck von 48 MPa, was 480 bar entspricht, ausgegangen. Diese Eckdaten erlauben eine Inspektion von 80% der deutschen tiefen Geothermiebohrungen und Thermalbrunnen.

Im hinteren Teil der Sonde befindet sich der *Cablehead*, also die Befestigung der Wireline. In der Standardausrüstung folgt dann ein Kühlvorrat aus Eis, dieser ermöglicht laut [20] einen

1 GeoKam - Projekt

Betrieb der gekühlten Elektronik von zwei Stunden bei maximaler Umgebungstemperatur plus insgesamt weiterer zwei Stunden Ablass- und Aufholzeit. Nach dem gekühlten Elektronikbereich, welcher mittels eines Stahl-Dewars isoliert ist folgt das Kameramodul. Im Elektronikbereich kann eine Maximaltemperatur von 90 °C eingehalten werden. Das Kameramodul ist mit einem Glas-Dewar isoliert und mit drei Kameras ausgestattet. Der gesamte Kamerakopf ist drehbar gelagert, um für die zwei Seitenkameras einen optimalen Blickwinkel durch die sechs Glasöffnungen des Gehäuses einzustellen. Weiter gibt es noch eine Frontkamera. Die Beleuchtung ist mittels Hochleistungs-Leuchtdioden (LEDs) realisiert, welche an der Außenhülle der Sonde als Verschleißteile und innerhalb und außerhalb des Dewars angebracht sind. Verschlusszeit, Fokus und Blende der Kameras können mittels Servos verstellt werden.

1.3 HiTES - Elektronik der GeoKam-Sonde

Als Hauptplatine kommt das so genannte *HiTES-Brd02* zum Einsatz. Als Hauptrechner wird eine Field Programmable Gate Array (FPGA) des Typs *Spartan-6 XA6SLX75* mit erweitertem Temperaturbereich in der *Automotiv*-Version der Firma XilinxTM verwendet. Dies bietet maximale Flexibilität durch die Einbindung erstellbarer Hardware-Blöcke, so genannten *IP-Cores*. Diese werden mittels der Programmiersprache Very High Speed Integrated Circuit Hardware Description Language (VHDL) programmiert.

Der Field Programmable Gate Array (FPGA) eignet sich besonders gut für die Parallelisierung von Prozessen, weswegen er häufig bei De/Kodierungen, De/Chiffrierungen, De/Komprimierungen und Software Defined Radio (SDR) eingesetzt wird. Die Datenübertragung der GeoKam wird mittels SDR realisiert, es kommt eine Reed Solomon (RS)-Kanalkodierung zum Einsatz und das Livebild der Kamera muss für die Übertragung JPEG komprimiert werden. In Verbindung mit der hohen maximalen Betriebstemperatur von 125 °C und der hohen Flexibilität durch das Einbinden verschiedener *IP-Cores* stellt der FPGA das ideale Bauteil für das *HiTES-Brd02* dar. (Siehe: [24])

Eine *Leon-CPU* (Sparc-Derivat) mit dem Open-Source-Echtzeitbetriebssystem *eCos* wird als *IP-Core* eingebunden. Das Echtzeitbetriebssystem verwaltet die Kommunikation zwischen Sonde und Host mittels des Ethernet Protokolls *IEEE 802.3u* (Siehe: [7]). Außerdem vereinfacht es die Anbindung verschiedener Komponenten via CAN, I²C und LVDS durch Open-Source-Softwaremodule und es kann beliebige C-Software ausgeführt werden.

Zwei baugleiche *HiTES-Brd02* fungieren mit unterschiedlichen IP-Cores als Sonden- bzw.

Host-Platine. So wird ein JPEG-Kompressor, ein RS-Kodierer, der **Datenkanal Sonde-Host (ChUp)** Modulator und der **Steuerkanal Host-Sonde (ChDn)** Demodulator als IP-Core in den FPGA der Sonde gebrannt. (Siehe: [7])

Sowohl der für SDR benötigte Analog Digital Wandler (ADC) als auch der Digital Analog Wandler (DAC) befinden sich auf der Hauptplatine, so wird die analoge Signalverarbeitung mittels einer Steckplatine vorgenommen. Das Eingangs-, wie auch das Ausgangssignal ist differenziell, um Gleichtaktstörungen durch die verbauten Servos und Gleichspannungsnetzteile zu unterdrücken.

Von dem Projektpartner HHI wurde die in Kapitel 2 beschriebene OFDM-Datenübertragung inklusive der analogen Steckplatinen namens „Filter- und Verstärkerplatine (FAmp)“ für Sonden- und Hostmodul entwickelt. Mittels dieser Platinen wird im Frequenz Duplexbetrieb die Datenübertragung in zwei Frequenzbereichen für ChUp und ChDn auf die Powerline-Leistungsversorgung aufgekoppelt. Die hochfrequenten Störungen der Hochspannungs DC/DC-Netzteile müssen durch AC-Blockglieder entfernt werden. Mittels eines *PIC*-Mikrocontrollers auf der Analogplatine kann über I²C die Verstärkung des Empfangs-Verstärkers angepasst werden. (Siehe: [19])

1.4 Ziel dieser Arbeit

Im Rahmen dieser Bachelorarbeit soll für Powerline versorgte Sonden ein Verfahren zur automatischen Kanaladaption der Orthogonal Frequency-Division Multiplexing (OFDM)-Datenübertragung an unterschiedliche Kabellängen und -charakteristika entwickelt werden. Hierzu werden

- eine **Automatic Gain Control (AGC)** für den Steuerkanal Host-Sonde (ChDn),
- eine AGC für den Breitband-Datenkanal Sonde-Host (ChUp),
- ein Verfahren für die Vermessung der Wireline,
- eine Berechnung der **Waterfilling Koeffizienten (WFC)** und
- ein mittels WFC **adaptiver Sender (Tx)**

benötigt.

1 GeoKam - Projekt

Bei der Realisierung der Ziele erfolgen alle Entwicklungen und Simulationen mittels Matlab-Simulink™. Die benötigten AGCs werden im ersten Schritt entworfen, es folgt der Entwurf eines automatisierbaren Vermessungsverfahrens mit anschließender WFC-Berechnung. Abschließend erfolgt die Optimierung am Tx um variable WFC verarbeiten zu können. Ziel dieser Arbeit ist die vollständige Integration der benötigten zusätzlichen Bausteine in die vorhandenen Modulator- und Demodulatorsimulationen. Nach der Integration soll ein simulierter Funktionstest des gesamten Verfahrens mit abschließender Modulation und Demodulation eines Frames über verschiedene Kabel folgen. Nach der vollständigen Simulation können diese Matlab-Simulink™-Projekte mittels Xilinx-Systemgenerators™ in VHDL synthetisiert werden, um sie in VHDL-Projekte zu implementieren. Die Funktionalität wird abschließend durch das FPGA-Debug-Programm Xilinx-ChipScope™ überprüft.

2 Das Datenübertragungsverfahren (OFDM)

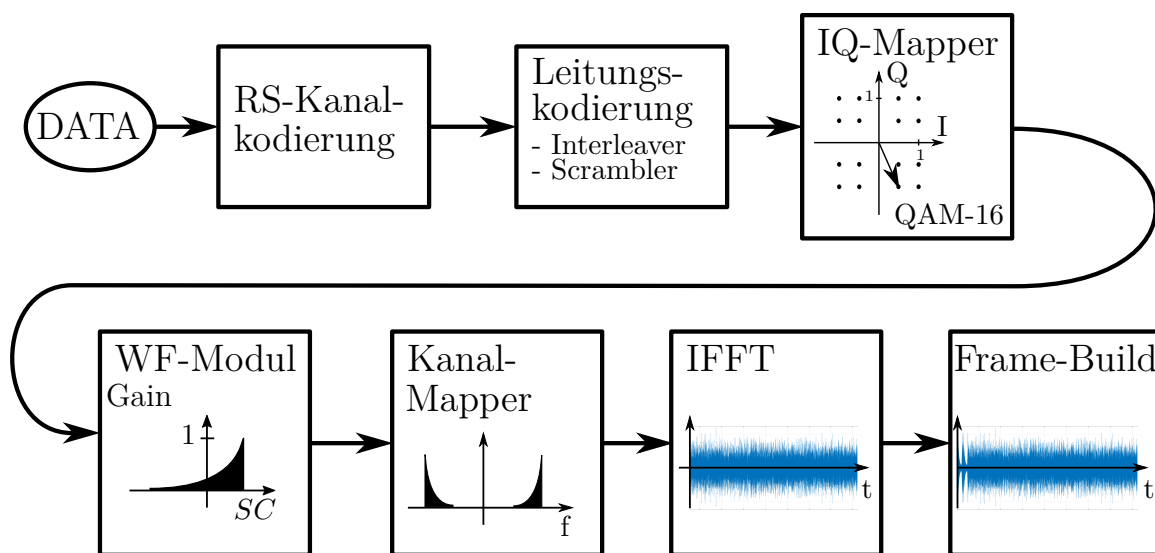


Abbildung 2.1: Signalflussdiagramm des ChUp Tx

Das Datenübertragungsverfahren wurde vom Fraunhofer-Institut für Nachrichtentechnik, Heinrich-Hertz-Institut kurz HHI entwickelt und liegt als mittels Xilinx-Systemgenerators[™] generierbare Matlab-Simulink[™]-Simulation und Matlab Skript vor. Es kann also vollständig in das Verfahren eingegriffen werden.

Im folgenden wird auf das verwendete Verfahren in Theorie und Implementierungspraxis kurz und Schritt für Schritt eingegangen. Verwiesen wird hierbei auf den Abschlussbericht des HHI [19] und auf [9], [17] und [8].

Es wird ein Frequenzduplexverfahren nach Abbildung 2.1 verwendet, bei dem sich der zuverlässige und schmalbandige Steuerkanal Host-Sonde (ChDn) im niedrigen Frequenzbereich und der breitbandige Datenkanal Sonde-Host (ChUp) im oberen zur Verfügung stehenden Frequenzbereich befindet. Als Modulation wird sowohl beim ChDn als auch

2 Das Datenübertragungsverfahren (OFDM)

beim ChUp Orthogonal Frequency-Division Multiplexing (OFDM) eingesetzt.

2.1 Kanalkodierung

Bei der *Kanalkodierung* wird gezielt Redundanz eingefügt, um die Fehlersicherheit zu erhöhen. Zur Fehlersicherung wird sowohl bei ChDn und bei ChUp eine RS-Kanalkodierung eingesetzt.

Es werden Gruppen von Bits, üblicherweise acht Bit, zu einer Zahl in einem endlichen Körper mit Erweiterungskörper $GF(p^m)$ zu einem so genannten Symbol, zusammengefasst. Wobei für p prinzipiell eine beliebige Primzahl möglich wäre, in der Praxis wird allerdings auf Grund des dualen Zahlensystems die kleinste Primzahl $p = 2$ genommen. Für m kann prinzipiell jede ganze Zahl verwendet werden, häufig wird allerdings $m = 8$ genommen, da so jedes Byte für ein Symbol steht.

Die Datenbytes werden als Koeffizienten eines Polynoms aufgefasst. Durch die Modulo-Division des Datenpolynoms mit dem so genannten Generatorpolynom wird definiert Redundanz erzeugt und an das Datenpaket angehängt.

Im Empfänger wird durch Division mit dem Generatorpolynom das so genannte Syndrom $s(x)$ erzeugt. Ist dieses null, so ist bei der Übertragung kein Fehler aufgetreten und die Daten können nach Abschneiden der Redundanz direkt verwendet werden.

Ist $s(x) \neq 0$, so ist ein Fehler aufgetreten. Es wird mittels des Berlekamp-Massey-Algorithmus die Fehlerstelle gefunden, anschließend kann analytisch der Fehler behoben werden, indem das Syndrom null gesetzt wird. (Siehe: [15])

Die RS-Kanalkodierung wird mittels der Parameter Gesamtsymbolanzahl n und Datensymbole k beschrieben. Die Hammingdistanz d wird aus n und k mittels $d = n - k + 1$ berechnet. In Kurzform also $RS[n, k, d]$. Es können somit $f = d - 1$ Fehler erkannt und $f/2$ Fehler korrigiert werden. (Siehe: [2])

Im ChUp wird ein $RS[156, 116, 41]$ verwendet. Es können also 20 Symbolfehler korrigiert werden und die Nettodatenrate durch die Kodierung sinkt um $d/n = 41/156 \approx 0,26 \hat{=} 26\%$.

2.2 Leitungskodierung

Die *Leitungskodierung* passt den Sendedatenstrom an den Übertragungskanal und die verwendete Modulation an. Es wird keine Redundanz entfernt oder hinzugefügt.

2 Das Datenübertragungsverfahren (OFDM)

Bei OFDM-Modulationen müssen konstante Symbolsequenzen verhindert werden, da diese zu einem übermäßig hohen *Crest*-Faktor (Siehe [9]) führen. Der Crest- oder Scheitelfaktor beschreibt das Verhältnis von Effektivwert zu Scheitelwert. Durch die Fouriertransformation der aus konstanten Symbolen resultierenden *rect*-Funktion, kommt es im Zeitbereich zu *sinc*-Funktionen mit hohem Scheitelwert aber geringem Effektivwert.

Da die Ausgangsamplitude durch den DAC und die Analogbauteile begrenzt ist, gilt es bei möglichst geringer Amplitude möglichst viel Leistung zu senden. Um dies zu gewährleisten, müssen die Daten „verrauscht“ werden. Hierfür wird ein so genannter *Scrambler* eingesetzt, welcher mittels eines rückgekoppelten Schieberegisters mit *XOR*-Verknüpfungen die Daten „verwürfelt“.

Bei der GeoKam wird ein Scrambler mit 8 Stellen und dem Polynom

$$S(x) = x^7 + x^4 + 1 \quad (2.1)$$

verwendet. Ein Interleaver wird nicht benötigt, da nicht mit Burstfehlern zu rechnen ist.

2.3 IQ-Mapper

Im *IQ-Mapper*, zu sehen in Abbildung 2.1 oben rechts, wird der serielle Datenstrom bei einer Quadraturamplitudenmodulation (QAM) (Siehe [9]) auf Symbole in der IQ-Ebene aufgeteilt. Es werden N Bit zu einem Symbol zusammengefasst, also müssen $M = 2^N$ mögliche Symbole zur Verfügung stehen.

Es wird eine QAM-16-Modulation, also 16 Symbole im IQ-Diagramm und somit $N = \log_2(M) = 4$ Bit pro Symbol, verwendet.

Anschließend werden die Symbole mit den Waterfilling-Koeffizienten skaliert. Dazu mehr in Kapitel 4.

2.4 Bandpasstransformation und Kanalmapping

In der *Bandpasstransformation*, in der Literatur auch oft Kanalmapping genannt, werden die vom IQ-Mapper vorgegebenen komplexen Datensymbole im Basisband auf das Spektrum der Übertragung gelegt und so in die Bandpass-Domain überführt.

Um ein rein reelles Zeitsignal am Ende der Inverse Fast Fourier Transformation (IFFT) der Größe N zu gewährleisten, muss der Realteil symmetrisch zur Ordinate bei $n = N/2$ und

2 Das Datenübertragungsverfahren (OFDM)

der Imaginärteil punktsymmetrisch zum Ursprung bei $n = N/2$ sein. Durch die Festlegung auf ein reelles Zeitsignal kann also nur das halbe Spektrum frei belegt werden.

Invers wird im Empfänger (Rx) mittels der *Basisbandtransformation* das Bandpasssignal ins Basisband transformiert.

Es wird eine $N = 2048$ er (I)FFT beim ChUp verwendet. Bei einer Abtastrate von $f_s = 16$ MHz entspricht das einem Spektrum von -8 MHz bis 8 MHz und einem Abstand der Sub Carrier (SC) von

$$\Delta f = f_s = \frac{f_a}{N} = \frac{16 \text{ MHz}}{2048} = 7,8125 \text{ kHz.} \quad (2.2)$$

Verwendet werden 312 der möglichen 1024 SC.

2.5 (Inverse) Fast Fourier Transformation

Die Fast Fourier Transformation (FFT) bzw. IFFT ist eine ressourcensparende Implementierungen der Diskrete Fourier Transformation (DFT) bzw. Inverse Diskrete Fourier Transformation (IDFT). Der geringere Ressourcenbedarf der (I)FFT gegenüber der (I)DFT wird unter anderem erreicht, in dem eine Größe von 2^n gewählt wird, was bei Dualrechnern zu einer idealen Implementierung führt.

Mittels der IFFT wird das in den vorangegangenen Stufen erzeugte Bandpasssignal in die Zeitebene transformiert. Invers wird das Zeitsignal für die Demodulation im Rx mittels der FFT in den Frequenzbereich transformiert.

2.6 ISI- und ICI-Freiheit bei OFDM

Die Impulsantwort des Sende- und Empfangsfilters wird als Rechteck mit der Zeitkonstanten $1/T_s$ festgelegt und die Abtastrate f_a auf $f_a = N/T_s$. Im Frequenzbereich wird der Sende- und Empfangsfiler durch Multiplikation mit dem Sendesignal angewendet. Daraus folgt bei Festlegung auf eine feste Abtastrate und (I)FFT-Größe eine Zeitkonstante des Sende- und Empfangsfilters von

$$T_s = \frac{N}{f_a}. \quad (2.3)$$

Bei der Transformation eines N -Samples langen Spektrums entsteht ein N -Samples langes

2 Das Datenübertragungsverfahren (OFDM)

Zeitsignal. Der Sende- und Empfangsfilter kann also allein durch die (I)FFT-Länge angewendet werden.

$$T_a = \frac{1}{f_a} \iff T_s = N \cdot T_a \quad (2.4)$$

Wird die Rechteckfunktion $\text{rect}(t/T_s) = \square(t/T_s)$ fouriertransformiert so entsteht eine sinc Funktion

$$\mathcal{F}\{\square(t/T_s)\} = \frac{T_s}{\sqrt{2\pi}} \cdot \text{sinc}(\pi f \cdot T_s) \quad (2.5)$$

mit Nullstellen bei

$$f_0 = \frac{i}{T_s} \quad ; \quad i \in \mathbb{Z}, i \neq 0 \quad (2.6)$$

wobei $f_0 = f_s$. Die Nullstellen des Sende- und Empfangsfilters liegen also spektral betrachtet auf den benachbarten SC-Trägerfrequenzen. Eine **Inter Carrier Interferenz (ICI)** kann damit ausgeschlossen werden. Durch den Grenzwert

$$\lim_{f \rightarrow 0} \text{sinc}(\pi f \cdot T_s) \stackrel{!}{=} \lim_{f \rightarrow 0} \frac{\cos(\pi f \cdot T_s)}{1} = 1 \quad (2.7)$$

entsteht bei dem SC keine weitere Skalierung, wenn die IFFT-Skalierung richtig gewählt ist (Siehe [9]).

Um eine **Intersymbolinterferenz (ISI)** zu vermeiden muss, zwischen den IFFT-Intervallen ein Schutzintervall geschaltet werden, um dem Übertragungskanal Zeit zu geben, sich einzuschwingen. Dieses Schutzintervall wird auch *Guardintervall* oder *Cyclic Prefix* genannt und ist mindestens so lang wie die maximale Gruppenlaufzeit des Kanals. Hierbei wird das Ende der IFFT-Periode im Guardintervall wiederholt. So ist eine OFDM-Datenübertragung ICI und ISI frei.

Bei der 2048er (I)FFT des ChUp wird ein Guardintervall der Länge 512 eingesetzt. Eine IFFT Periode ist somit $2048 + 512 = 2560$ Samples lang.

2.7 Framebuilder

Die soeben beschriebene ISI- und ICI-Freiheit kann nur bei Samplesynchronität von Tx und Rx gewährleistet werden. Hierfür wird eine Header-Sequenz bestehend aus Pilotton

2 Das Datenübertragungsverfahren (OFDM)

und Präambel zu Beginn jedes Frames abgespielt. Die Synchronisation erfolgt mittels der Kreuzkorrelation des Pilottons. Über die Präambel wird der Kanal Equalizer trainiert, um die Phase und Amplitude fein einzustellen. Beim ChUp kommen zwei identische Präambeln zum Einsatz, um den Phasenoffset und Phasendrift im Laufe eines Frames zu schätzen und zu korrigieren, mehr dazu in Sektion 2.9.2.

2.8 Zusammenfassung

Ein ChUp Frame dauert 5,14 ms und besteht aus einem Header mit 4 Samples Pilotton plus $2 \cdot 2560 = 5120$ Samples Präambel, 30 Datenpaketen à 312 SC und Datensymbolen à 4 bit also 156 Byte. Zwischen den Frames wird eine Pause von 163 μ s zur Trennung eingelegt. Insgesamt werden also pro Frame $156 \text{ B} \cdot 30 = 4,68 \text{ kB}$ gesendet. Daraus folgt die Bruttodatenrate von

$$\frac{4,68 \text{ kB}}{5,14 \text{ ms} + 163 \mu\text{s}} = 882,52 \text{ kB/s} = 7,06 \text{ Mbit/s} \quad (2.8)$$

Zusammenfassend wird beim ChDn und beim ChUp eine OFDM mit 4 respektive 312 Sub Carrier (SC) verwendet. Der ChDn nutzt eine spektrale Bandbreite von 31,25 kHz mit einer Mittenfrequenz von 101 kHz, während der ChUp eine Mittenfrequenz von 2,5 MHz bei einer Bandbreite von 2,4 MHz verwendet.

Der ChDn hat eine Bruttodatenrate von 256 kbit/s.

2.9 Besonderheiten des Empfängers

Alle vorher beschriebenen Module kommen im Empfänger in inverser Reihenfolge zum Einsatz, es gibt allerdings ein paar Besonderheiten des Rx.

2.9.1 Framestart Detektion

Der Rx des ChUp ist mittels eines Verfahrens aus Hochpassfilterung, Quadrierung, Mittelung und anschließender Division mit einem verzögerten Mittel synchronisiert. Es wird versucht, den starken Leistungsanstieg von Temperaturrauschen und Störungen in der Framepause und dem Senderahmen zu detektieren.

Im Rx des ChDn wird mittels eines Filters mit endlicher Impulsantwort (FIR) eine Kreuz-

2 Das Datenübertragungsverfahren (OFDM)

korrelation durchgeführt, indem der Pilotton gespiegelt als FIR-Koeffizienten geladen wird. Mittels der Kreuzkorrelationsfunktion (KKF)-Peaks kann ein Senderahmen detektiert werden.

2.9.2 Frequenzoffset Korrektur

Da die Quarzoszillatoren der Sonden- und Hostplatine nicht synchronisiert sind, kommt es zu einer leichten Frequenzverschiebung der Modems. Auf Grund der verstimmten Oszillatoren kann die Phase innerhalb eines Senderahmens driften und so die Übertragung verschlechtern.

Beim ChUp wird deswegen ein Verfahren verwendet, bei dem der Phasendrift zwischen zwei Präambeln ermittelt wird. Die ermittelte Phasenverschiebung kann nun, bei Annahme einer konstanten Frequenzverschiebung, also einer linearen Phasenänderung, korrigiert werden. Dabei wird mit einem komplexen Korrekturterm multipliziert.

2.9.3 Kanal Equalizer

Lineare Verzerrungen der Wireline werden im Kanal Equalizer mittels der Präambel korrigiert. Die Präambel liefert Referenzwerte für jeden verwendeten SC. Durch komplexe Division von Soll- und Istwert wird der Kanal abgebildet. Der Reziprokwert der Kanalabbildung stellt den Korrekturterm der Datensymbole dar.

Die Kanalverzerrung erfolgt zu Beginn jedes Senderahmens, in dessen Dauer der Kanal als quasistationär angenommen werden kann. In Abbildung 9.2 sind alle 312 Symbole einer Präambel des ChUp in der IQ-Ebene aufgetragen, sie sind äquidistant vom Ursprung und gleichmäßig über den Winkel verteilt.

3 Übertragungstrecken

Die *Wireline4k* ist die eingangs erwähnte kommerziell genutzte Wireline. Das Testkabel500 mit einer Länge von 500 m wurde für Testzwecke beschafft.

3.1 Kabelbeanspruchung

Die Wireline ist eines der kritischsten Elemente der Geoinspektionssonde, da hier hohe mechanische, thermische und elektrische Anforderungen aufeinandertreffen. Bei den eingangs beschriebenen Umgebungsbedingungen muss das verwendete Kabel sein Eigengewicht und das der Sonde tragen.

Auch die Powerline-Datenübertragung, also die Versorgung und Datenübertragung über ein Kabel, stellt erhebliche Anforderungen an die Hardware. Es wird ein konstanter Wellenwiderstand benötigt, um Reflektionen zu verhindern. Außerdem sollte der Gleichstrom (DC)-Widerstand möglichst gering sein um möglichst viel Leistung in der Sonde zur Verfügung zu haben und die benötigte DC-Spannung zur Leistungsübertragung zu minimieren. (Siehe [19] und [14])

3.2 Pegelabschätzungen bei Kabel Wireline4k

Es ist, wie eingangs erwähnt, mit vielen verschiedenen additiven Störungen und Rauschen auf dem Übertragungsweg zu rechnen. Im folgenden soll das zu erwartende ideale Signal-Rausch-Verhältnis (SNR) errechnet werden. Die Parameter sind dem Bericht [19] und der Matlab[™] - Simulink[™] Simulation entnommen.

Zu Grunde liegt eine Kabeltemperatur $\vartheta = 100\text{ °C}$, ein Wellenwiderstand von $R = 50\Omega$ und eine Bandbreite $\Delta f = 3\text{ MHz}$. Es ergibt sich also mittels

$$U_{R,eff} = \sqrt{4k_B T R \Delta f} \quad (3.1)$$

3 Übertragungsstrecken

$$U_{50\Omega,eff,Noise} = \sqrt{4k_B \cdot (100^\circ\text{C} + 273,15)\text{K} \cdot 50\Omega \cdot 3\text{MHz}} = 1,76\mu\text{V} \quad (3.2)$$

effektive Rauschspannung am Rx.

Gesendet wird mit $U_{pp} = 8\text{V}$. Somit ist $\hat{U} = U_{pp}/2 = 4\text{V}$, also bei einem durchschnittlichen Scheitel -oder Crestfaktor von $k_s \approx -12\text{dB}$ ergibt sich

$$U_{eff,Tx} = \hat{U} \cdot k_s = 4\text{V} \cdot 10^{\frac{-12}{20}} \approx 1\text{V}. \quad (3.3)$$

Die maximale Dämpfung im verwendeten Frequenzband ist Abbildung 3.2 in [19] mit $a_{dB} = -85\text{dB}$ entnommen, dadurch ergibt sich ein effektiver Spannungswert von

$$U_{eff,Rx} = U_{eff,Tx} \cdot a = 1\text{V} \cdot 10^{-85/20} = 56\mu\text{V}. \quad (3.4)$$

Es ergibt sich also ein SNR von

$$SNR_{dB,Rx} = 20 \cdot \log\left(\frac{U_{eff,Rx}}{U_{50\Omega,eff}}\right) = 20 \cdot \log\left(\frac{56\mu\text{V}}{1,76\mu\text{V}}\right) \approx 30\text{dB}. \quad (3.5)$$

Hier muss allerdings das Rauschen des Eingangs-Operationsverstärkers (OP) beachtet werden, welches sich aus der Rauschspannung $v = 1,2\text{nV}/\sqrt{\text{Hz}}$ (Siehe: [3]) und der Bandbreite $B = 3\text{MHz}$ folgendermaßen ergibt:

$$U_{eff,Amp} = v \cdot \sqrt{B} = 1,2\text{nV}/\sqrt{\text{Hz}} \cdot \sqrt{3\text{MHz}} = 2,08\mu\text{V} \quad (3.6)$$

Es ergibt sich also ein $SNR_{dB,ADC}$ von

$$SNR_{dB,ADC} = 20 \cdot \log\left(\frac{U_{eff,Rx}}{U_{50\Omega,eff} + U_{eff,Amp}}\right) = 20 \cdot \log\left(\frac{56\mu\text{V}}{1,76\mu\text{V} + 2,08\mu\text{V}}\right) \approx 23\text{dB}. \quad (3.7)$$

Wird eine größere Sondierungstiefe benötigt, so muss entweder die Sendeleistung erhöht werden, oder es müssen einzelne SC der OFDM abgeschaltet werden, um spektral betrachtet mehr Leistung in die günstigeren SC einbringen zu können. (Siehe Kapitel 4.)

Das errechnete SNR würde auch höhere Modulationsformen ermöglichen, so könnte laut [15] z.B. bei einer festgelegten Bitfehlerrate (BER) von $BER = 5 \cdot 10^{-5} 1/\text{bit}$ die Modulation von einer QAM-16 auf eine QAM-64 erhöht und die eineinhalb fache Datenrate erzielt werden. Allerdings ist aufgrund der Powerline-Versorgung der Sonde mit weiterem

3 Übertragungsstrecken

Rauschen zu rechnen, weswegen eine höhere Sicherheit eingebaut wird.

3.3 Vergleich von Wireline4k und Testkabel500

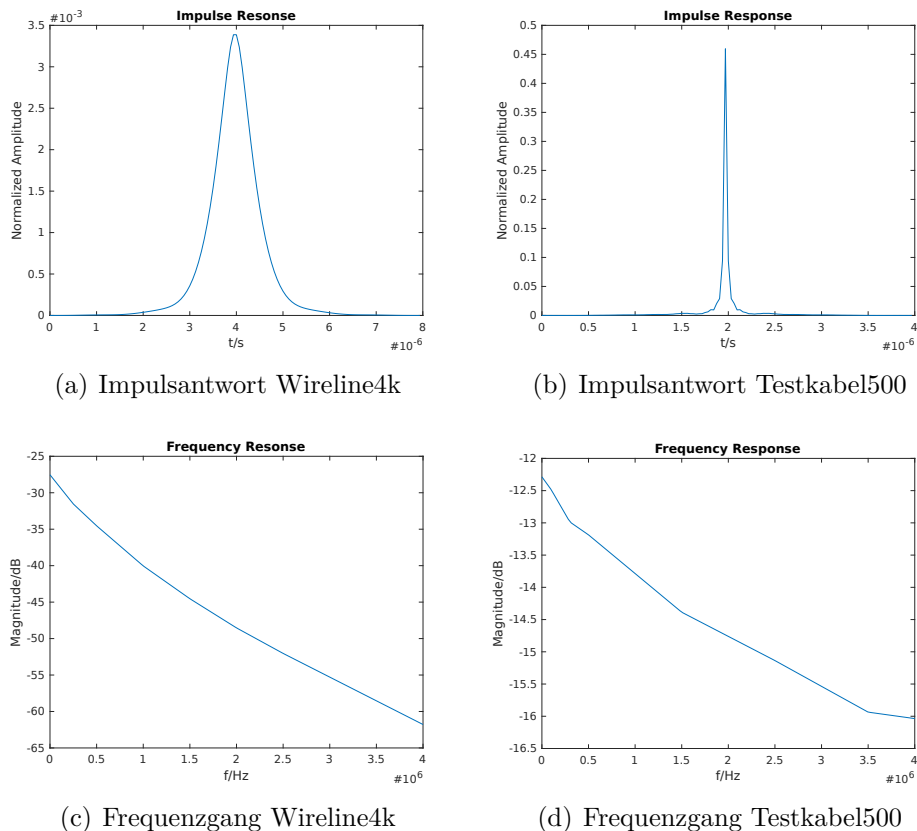


Abbildung 3.1: Vergleich der Strecken

Abbildung 6.11 zeigt in (a) und (b) jeweils die Impulsantwort der Wireline4k mit einer Länge von 4,2 km und des Testkabels500 mit einer Länge von 500 m. In (c) und (d) ist analog der Frequenzgang der beiden Strecken zu sehen. Zu beachten sind die unterschiedlich skalierten Achsen. Im Vergleich fällt die viel schmalere Impulsantwort des Testkabels auf, welches dadurch für eine Datenübertragung deutlich günstigere Voraussetzungen bietet. Wie zu erwarten ist die Fouriertransformation einer schmalen Impulsantwort breitbandig und der Dämpfungsgradient über die Frequenz gering.

4 Spektrale Vorverzerrung und Waterfilling

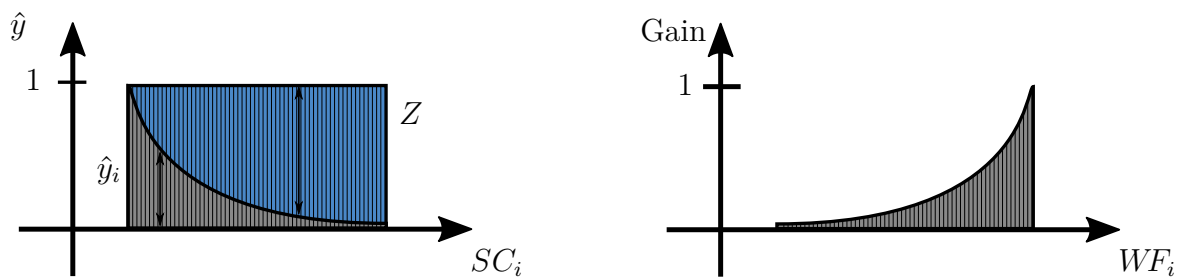


Abbildung 4.1: Schematische Darstellung der Generierung von Waterfillingkoeffizienten

4.1 Waterfilling und Pre-equalization

Waterfilling wird bei Mehrträgerdatenübertragungen verwendet. Es beschreibt die Vorverzerrung des Leistungsspektrums im Sender, um ein definiertes Leistungsspektrum beim Empfänger zu generieren. (Siehe: [17])

In [9] wird ein Waterfilling Algorithmus beschrieben, bei dem, bildlich gesprochen, die Rauschleistung jedes SC über den SC-Index aufgetragen wird. Eine beschränkte Menge Wasser, welche die gesamte maximale Ausgangsleistung repräsentiert, wird über dem vorhandenen Rauschen auf den SC verteilt. Es bildet sich eine konstante Oberfläche, bei der Träger mit wenig Rauschen mit viel Leistung belegt werden. Stark verrauschte Träger bekommen entweder sehr wenig Leistung oder werden abgeschaltet.

Um diese Art des Waterfillings sinnvoll nutzen zu können, wird unter anderem variable Modulationsarten der SC vorausgesetzt, damit das verbesserte SNR genutzt werden kann. Da bei dem in dieser Arbeit beschriebenen Verfahren die Rauschleistung pro SC auf Grund des Koaxialkabels als konstant angenommen werden kann, werden alle SC laut [9] mit

4 Spektrale Vorverzerrung und Waterfilling

gleicher Leistung belegt. Um am Rx ein konstantes SNR aller SC zu generieren, muss die Leistung am Tx invers zur Übertragungskennlinie der Wireline verteilt werden. Ein solches Verfahren der Vorverzerrung im Tx wird in [22] beschrieben und „*Pre-equalization*“ genannt. Hier geht es um einen Datenübertragungsstandard namens Data Over Cable Service Interface Specification (DOCSIS) auf Koaxialkabeln für Kabelfernsehen. Der Rx erkennt eine Ungleichverteilung des Empfangsleistungsspektrums und schickt Korrektorkoeffizienten zum Tx.

Pre-equalization wird heutzutage auch bei glasfasergestützten Kommunikationssystemen eingesetzt, um den „Frequenzgang“ der Glasfaser zu kompensieren und so ein konstantes optisches SNR am Rx zu generieren. (Siehe: [12])

Im folgenden stellt der Waterfillingkoeffizientenvektor also eine Kombination aus klassischem Waterfilling nach [9] bei konstanter Leistungsverteilung multipliziert mit dem inversen Frequenzgang der Wireline dar. Bei einem Frequenzmultiplexverfahren (FDM) oder bei OFDM kann der Waterfillingkoeffizientenvektor eingepflegt werden, indem man vor der IFFT jeden SC mit dem zugehörigen Waterfillingkoeffizienten multipliziert.

Würde man alle SC mit konstanter Amplitude belegen, würde beim Empfänger das abgetastete Spektrum \hat{y}_i des Übertragungskanals über den SC Index i abgebildet. Dies ist schematisch in Abbildung 6.3.6 zu sehen. Der Waterfillingkoeffizient WF_i des jeweiligen SC ergibt sich aus dem Reziprokwert des Eingangsspektrums, also

$$WF_i = \frac{1}{\hat{y}_i}. \quad (4.1)$$

Der Terminus „*Waterfilling*“ kann auch bei „*Pre-equalization*“ verwendet werden, wenn man nicht die Rauschleistung, sondern die Empfangsleistung der SC aufträgt und „Wasser“, in Abbildung 6.3.6 blau dargestellt, einfüllt, bis sich ein konstantes Leistungsspektrum bildet. Die Füllhöhe des Wasser Z ist somit bei einer Normierung der Maximalamplitude auf 1

$$Z = 1 - \hat{y}_i = 1 - \frac{1}{WF_i} \iff WF_i = \frac{1}{1 - Z}. \quad (4.2)$$

In Abbildung 6.3.6 sind rechts die resultierenden Waterfillingkoeffizienten zu sehen. Im letzten Schritt muss die Skalierung der IFFT α_{IFFT} angepasst werden.

Die Ausgangsleistung der N -Tap IFFT wird auf 1 normiert.

4.2 Überlegungen zur Kanalkapazität

Durch das *Waterfilling* soll die Kanalkapazität möglichst optimal ausgenutzt werden. Dieser Abschnitt dient also der Bewertung des aktuellen Verfahrens.

Die Kanalkapazität C eines AWGN Kanals ergibt sich laut [17] aus der Spektralen Bandbreite B , der Signalleistung P_{Rx} und der Rauschzahl N_0 mittels der Gleichung

$$C = B \cdot \log_2 \left(1 + \frac{P_{Rx}}{B \cdot N_0} \right). \quad (4.3)$$

$\frac{P_{Rx}}{B \cdot N_0}$ kann aus dem SNR bestimmt werden, somit ergibt sich:

$$C = B \cdot \log_2 \left(1 + 10^{SNR_{dB}/10} \right) \quad (4.4)$$

Bei einer Bandbreite von $B = 3$ MHz und einem SNR von $SNR_{dB,Rx} = 30$ dB beträgt die maximale Kanalkapazität $C = 30$ Mbit/s. Durch das Verstärkerrauschen verringert sich das SNR auf $SNR_{dB,ADC} = 23$ dB daraus folgt eine Kanalkapazität von $C = 23$ Mbit/s. Das verwendete OFDM-Verfahren kommt auf 6,4 Mbit/s. Für diese Anwendung ist die Datenbandbreite ausreichend und das Verfahren ist robust ausgelegt.

5 Kanaladaptionsverfahren

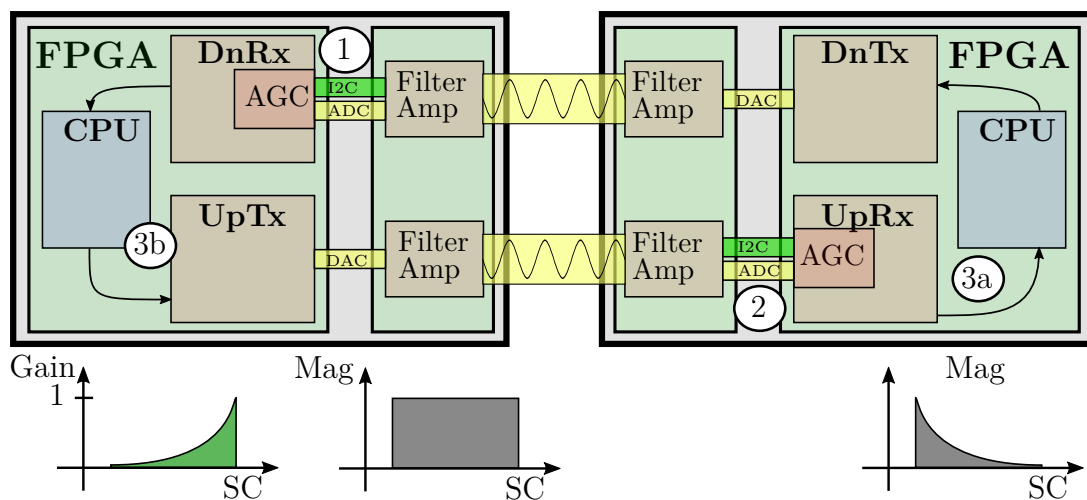


Abbildung 5.1: Kanaladaptionsverfahren

Die verwendeten Matlab-Simulink™-Simulationen wurden unter der Zuhilfenahme von [6], [10] und [8] entwickelt und optimiert.

Die Kanaladaptation kann grundsätzlich in drei Phasen aufgeteilt werden:

1. Einpegeln der AGC des ChDn

Um Kommandos an die Sonde schicken zu können, muss im ersten Schritt der Eingangspegel des ChDn adaptiv gestaltet werden.

2. Einpegeln der AGC des ChUp

Vor dem Einmessen der Wireline muss für die ideale Aussteuerung des ADCs der Host-Platine die passende Analogverstärkung eingestellt werden.

3. Kanaladaptionsverfahren

- Mittels eines Messverfahrens (schematisch als Rechteckfunktion über dem Sub Karrierindex (SC) in Grafik 5.1 dargestellt) muss der Betrag des Frequenzgangs

des Übertragungskanals (rechts unten in Abbildung 5.1 zu sehen) ermittelt werden. Der Reziprokwert jeder SC Magnitude repräsentiert normiert auf den SC mit der größten Verstärkung die Waterfillingkoeffizienten (grün in Abbildung 5.1 zu sehen) für die verwendete Wireline.

- b) Die Waterfillingkoeffizienten müssen mittels des ChDn an die Sonde übertragen werden und in den ChUp Tx-Modulator eingepflegt werden.

5.1 AGC für den Steuerkanal ChDn - Phase 1

Die AGC des Steuerkanals ChDn kann aufgrund der geringen Dämpfungen im unteren Frequenzbereich und der hohen ADC-Auflösung rein digital realisiert werden.

Der ADC bietet eine Auflösung von 14 Bit. Bei Vollaussteuerung kann also mit einem SNR durch Quantisierungsrauschen von

$$SNR_{\text{dB}} \approx 14\text{Bit} \cdot 6 \text{ dB/Bit} = 96 \text{ dB} \quad (5.1)$$

gerechnet werden. Zur Demodulation der QAM-16 wird laut [15] bei einem BER von 10^{-6} ein SNR von $\approx 15\text{dB}$ benötigt.

Wie in Abbildung 6.11 (c) zu sehen, ist die Dämpfung der Wireline4k im Frequenzbereich des ChDn $\approx 35\text{dB}$. Ist der ADC bei direkter Verbindung zwischen Sonde und Host voll ausgesteuert und wird die Wireline4k verwendet, ergibt sich ein SNR von $SNR_{\text{dB}} \approx 96 \text{ dB} - 35 \text{ dB} = 61 \text{ dB}$. Die Pegelanpassung kann also ohne Veränderung der Analogverstärkung erfolgen.

Für die Synchronisation wird eine Kreuzkorrelation mit einer „Zadoff-Chu“-Sequenz verwendet (Siehe [16]). Implementiert wird diese in der „Framestart-Detektion“ mittels eines FIR. Es wird die gespiegelte Synchronisations- oder Pilotsequenz als Koeffizientenvektor in den FIR eingetragen. Durch die Quadrierung des Filterausgangssignals wird das Vorzeichen entfernt und eine Spreizung vorgenommen. Anschließend kann mittels eines Schwellwertes, genannt *KKF-Threshold*, der so genannte *Framestart* detektiert werden. KKF steht für Kreuzkorrelationsfunktion. Siehe auch Sektion 2.9.1.

Da der KKF-Peak linear vom Eingangspegel abhängt, muss für einen konstanten Pegel gesorgt werden, um den KKF-Schwellwert richtig einzustellen.

Die Pilotsequenz steuert den DAC des Tx voll aus, auch nach dem Übertragungskanal stellt sie also eine gute Möglichkeit dar, die digitale Vorverstärkung mittels Normierung

der Pilotsequenz auf eine definierte Höhe einzustellen.

5.2 AGC für den Datenkanal ChUp - Phase 2

Im Frequenzbereich des ChUp muss bei der Verwendung verschiedener Wireline mit einer Dynamik von etwa 90 dB gerechnet werden, da mittels des Waterfillings die Sendeleistung auf den am stärksten gedämpften SC abgestimmt wird. Die Anpassung der Eingangsverstärkung erfolgt analog.

Die AGC wird im FPGA realisiert, jedoch müssen Steuerbefehle an die FAmP-Steckplatine geschickt werden um die Analogverstärkung zu verstellen. Aufgrund der daraus resultierenden Latenz und dem bislang unbekanntem Kanaleinfluss muss die Eingangsverstärkung mittels des quadratischen Mittelwerts (RMS) des ADC-Signals eingestellt werden. Hierfür muss der Crest- oder Scheitelfaktor des Empfangssignals bekannt sein, denn es wird ein Amplitudenfenster festgelegt.

Während der Laufzeit eines erkannten Frames darf keine Adaption der Verstärkung vorgenommen werden. In diesem Fall würden alle Daten ab diesem Zeitpunkt verloren gehen, da der Channelequalizer nur zu Beginn mit Hilfe der Präambel adaptiert wird.

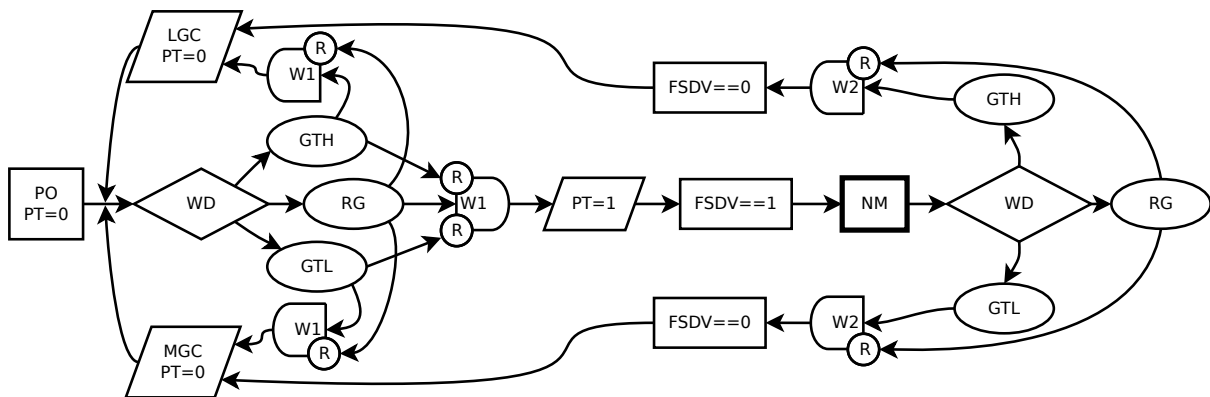


Abbildung 5.2: Zustandsfolgediagramm der AGC ChUp

In Abbildung 5.2 ist das Zustandsfolgediagramm der AGC des ChUp zu sehen. Ziel ist der **NM** (Normal Mode). Es wird mit State **PO** (Power On) begonnen, bei dem **PT** (Pass Through) gleich Null ist. Mittels der **WD** (Window Detection) wird ermittelt, ob der ADC-Signalpegel

- (a) zu klein ist, dann ist die Analogverstärkung zu niedrig, also **GTL** (Gain To Low),

5 Kanaladaptionsverfahren

- (b) stimmt, somit stimmt auch die Verstärkung, also *RG* (Right Gain) oder
- (c) zu groß ist, dann ist die Analogverstärkung zu hoch, also *GTH* (Gain To High).

Beim Messen des Signalpegels kommt es zu Schwankungen. Um erst bei einer sicheren Über- oder Unterschreitung des gewählten Fensters eine Korrektur vorzunehmen, wird eine Zeit *W1* definiert und gewartet, bis ein Kommando zur Änderung der Analogverstärkung abgeschickt wird. Die Wartezeit wird zurückgesetzt, wenn der Signalpegel wieder im angestrebten Bereich ist. Analog wird bei einem Verlassen des Fensters nach *RG* die Wartezeit auch zurück gesetzt.

Die Wartezeit *W1* setzt sich aus der Verzögerung der Kommandos zur Famp-Platine und der Größe des gleitenden Mittelwerts der Signalamplitude zusammen. Anschließend wird entweder ein *LGC* (Lower Gain Command) bei zu hohem Pegel oder ein *MGC* (More Gain Command) bei zu niedrigem Pegel abgesetzt. Hierbei wird $PT = 0$ gesetzt. Mittels der Regelschleife aus *WD*, *GTH/GTL* und *LGC/MGC* wird iterativ die Verstärkung geregelt. Das Spannungsfenster kann abhängig von den Verstärkungsabstufungen gewählt werden (Siehe Sektion 6.2).

Nach *RG* und *W1* ist die richtige Verstärkung gefunden und *PT* wird gesetzt. Wird von der Frame Start Detektion (FSD) ein Senderahmen erkannt, kann in den *NM* übergegangen werden. Das Ausgangssignal der FSD ist *FSDV* (FSD Valid). Verlässt die Amplitude das Fenster für *W2*, also die Größe des gleitenden Mittelwerts, wird gewartet, bis der aktuelle Senderahmen beendet ist. Anschließend wird mittels *LGC/MGC* korrigiert. Dieses Signal kann als *Frame Lost* aufgefasst werden.

5.3 Verschiedene Methoden der Kanalvermessung - Phase 3

Zur Generierung der Pre-Equalization- und Waterfillingkoeffizienten muss der Übertragungskanal vermessen werden. Hierzu können verschiedene Methoden eingesetzt werden, welche hier diskutiert werden. Die WFC ergeben sich aus den normierten und invertierten Kanalparametern.

5.3.1 Sinc-Funktion oder Diracstoß

Eine Möglichkeit, die Wireline zu vermessen, besteht darin, eine Diracstoßfolge zu senden. Durch die Faltung des Systems mit dem gesendeten Diracimpuls $s(t) = \delta(t)$ würde die Impulsantwort am Rx folgendermaßen empfangen:

$$y(t) = s(t) * h(t) = \delta(t) * h(t) = h(t) \quad (5.2)$$

Problematisch ist hier allerdings, dass die Ausgangsamplitude durch den DAC und die analoge Signalverarbeitung begrenzt ist, wodurch nur mit geringer Leistung gesendet werden könnte. Außerdem ist ein Diracimpuls spektral breitbandiger als die verwendete Datenübertragung. Es würde also in spektral uninteressanten Bereichen Sendeleistung umgesetzt. Dadurch ist diese Methode eher weniger geeignet.

Dennoch könnte zur Optimierung dieses Verfahrens die Impulsantwort des Bandpassfilters der Datenübertragung gesendet werden. Hierdurch würde nur in dem spektral verwendeten Bereich Leistung umgesetzt. Im Zeitbereich entsteht so eine modulierte sinc-Funktion. Auch hier ist der Crestfaktor sehr hoch, somit lässt sich vergleichsweise mit wenig Leistung senden. Das Spektrum der Tiefpassdomain ist also:

$$X_{tp}(f) = \begin{cases} 1 & \text{wenn } |f| > B/2 \\ 1/2 & \text{wenn } |f| = B/2 \\ 0 & \text{wenn } |f| < B/2 \end{cases} \quad (5.3)$$

Durch die Faltung mit einer cos-Funktion, also zwei an der Ordinate gespiegelten Diracimpulsen, entsteht das Bandpassspektrum:

$$X_{bp}(f) = X_{tp} * \left(\frac{1}{2} \delta(f - f_0) + \frac{1}{2} \delta(f + f_0) \right) \quad (5.4)$$

Durch die Fouriertransformation der einzelnen Komponenten kann die Zeitbereichsfunktion hergeleitet werden. Dies kann auch mittels der vorhandenen IFFT umgesetzt werden: (Siehe: [11])

$$x_{bp}(t) = x_{tp} \cdot \cos(2\pi f_0 t) = \text{sinc}\left(\frac{2\pi f}{B}\right) \cdot \cos(2\pi f_0 t) \quad (5.5)$$

5.3.2 Kreuzkorrelation mit weißem Rauschen

Zur Verbesserung des Crestfaktors und der damit verbundenen verbesserten Sendeleistung könnten die verwendeten SC auch, wie z.B. in [18] beschrieben, mittels Kreuzkorrelation von Rauschsequenzen (pseudo-random) vermessen werden. Es würde ein bekanntes Rauschsignal gesendet und dieses kreuzkorreliert, daraus folgt die Impulsantwort der Wireline. Nun müsste eine hochauflösende FFT vorgenommen werden.

Für eine hoch genug aufgelöste Kreuzkorrelation müsste entweder offline die interne CPU verwendet werden oder es müsste eine weitere Hardwareeinheit bereitgestellt werden. Es ist also von einem erheblich höherem Ressourcenbedarf auszugehen.

5.3.3 Bandbegrenzttes weißes Rauschen

Man könnte auch alle verwendeten SC mit Phasenrauschen konstanter Amplitude belegen, welches mit Hilfe der IFFT synthetisiert wird. Bei wenig konstruktiver Interferenz der SC sinkt der Crestfaktor. Oder es ließen sich vorgenerierte Rauschsamples vom Read-Only Memory (ROM) abspielen.

Alle bis dato genannten FFT-basierten Verfahren haben zwei Nachteile:

1. Beim Empfang ist eine **Synchronisation** oder eine **Fensterung** der FFT erforderlich. Werden nicht exakt die selben Datensampels der IFFT in der FFT umgesetzt, kommt es zu Verzerrungen und Störungen des Spektrums. Dies ist auch als Leck-Effekt (Leakage effect) bekannt und tritt auf, wenn kein ganzzahliges Periodenviel-faches durch die FFT-Werte abgebildet wird.
Um diese Störungen zu mindern, muss eine Fensterung der FFT vorgenommen werden, oder es muss mittels eines Pilottons synchronisiert werden.
2. Durch die konstante **Leistungsverteilung** am Tx gehen bei längeren Wireline die hochfrequenten SC im Rauschen unter.

Auf Grund dieser Nachteile sind breitbandige Kanalmessungen in diesem Fall wenig vorteilhaft. Es müsste iterativ gearbeitet werden, um auch die Dämpfung der hochfrequenten SC hinreichend genau zu messen. Außerdem müssten mehrere FFT-Perioden gemittelt werden, um genau genug messen zu können.

5.3.4 Das Stepped Sub Carrier Sweep (SSCS)-Verfahren

Es wurde also ein Verfahren gewählt, bei dem jeder SC nacheinander einzeln gemessen wird. Hierbei kann mit maximaler Ausgangsamplitude gesendet werden, wodurch das SNR am Rx optimal ist. Dieses Verfahren wurde hier Stepped Sub Carrier Sweep (SSCS) genannt. Das benötigte sin- oder cos-Signal kann mit der IFFT oder einer Direct Digital Synthesis (DDS) synthetisiert werden. Im Rx kann mittels der FFT die Empfangsamplitude aufgenommen werden. Da bekannt ist, welcher SC angefordert wurde und somit zu erwarten ist, ist bekannt, welcher FFT Wert zu speichern ist. Durch die FFT wird schmalbandig gemessen, wodurch kaum breitbandige Rauschleistung die Messung verfälscht. Es folgt die in Sektion 6.3 beschriebene Generierung, Übertragung und Einpflegung der Waterfillingkoeffizienten.

6 Realisierung

Durch die Matlab-Simulink[™]-Simulationsumgebung können und werden die hier entwickelten Verfahren mit Hilfe des Xilinx-Systemgenerators[™] in VHDL-Code umgewandelt.

6.1 AGC des Steuerkanal Empfängers

Die in Sektion 5.1 beschriebene AGC ist als Blockschaltbild im Anhang in Abbildung 9.1 zu sehen. In der Schleife aus *S1* und *R2* wird das absolute Maximum der ADC-Daten gesucht. Das aktuelle Sample wird mit dem bis dato maximalen Wert verglichen. Ist es größer, wird es gespeichert. Mit *C4* wird die Minimalamplitude am Eingang und damit die Maximalverstärkung festgelegt. Es wird der Reziprokwert gebildet, der mittels des *CMults* die Maximalamplitude einstellt. In *D1* werden die ADC-Daten in dieser Zeit verzögert. Während eines detektierten Frames darf der Faktor der AGC nicht verändert werden, weswegen die Schleife mittels *I2* unterbrochen ist, während die *Eieruhr_v0.4* abläuft.

6.2 AGC des Datenkanal Empfängers

Die Block-Übersicht der Realisierung ist in Abbildung 6.1 zu sehen. Das Verfahren soll sowohl während der OFDM Übertragung als auch bei der SSCS-Kanalmessung funktionieren. Mit dem Blocks *WD* (Window Detect) wird erkannt, ob das ADC-Signal im richtigen Spannungsfenster ist. Mittels *RG* wird erkannt, ob das Amplitudenfenster stimmt und anschließend werden verschiedene Zähler zurück gesetzt. *get_C* bzw. *ger_pre_C* erzeugt schlussendlich Kommandos, welche durch Firmware entgegengenommen werden und an die FAMP-Platine weiter geleitet werden müssen.

6 Realisierung

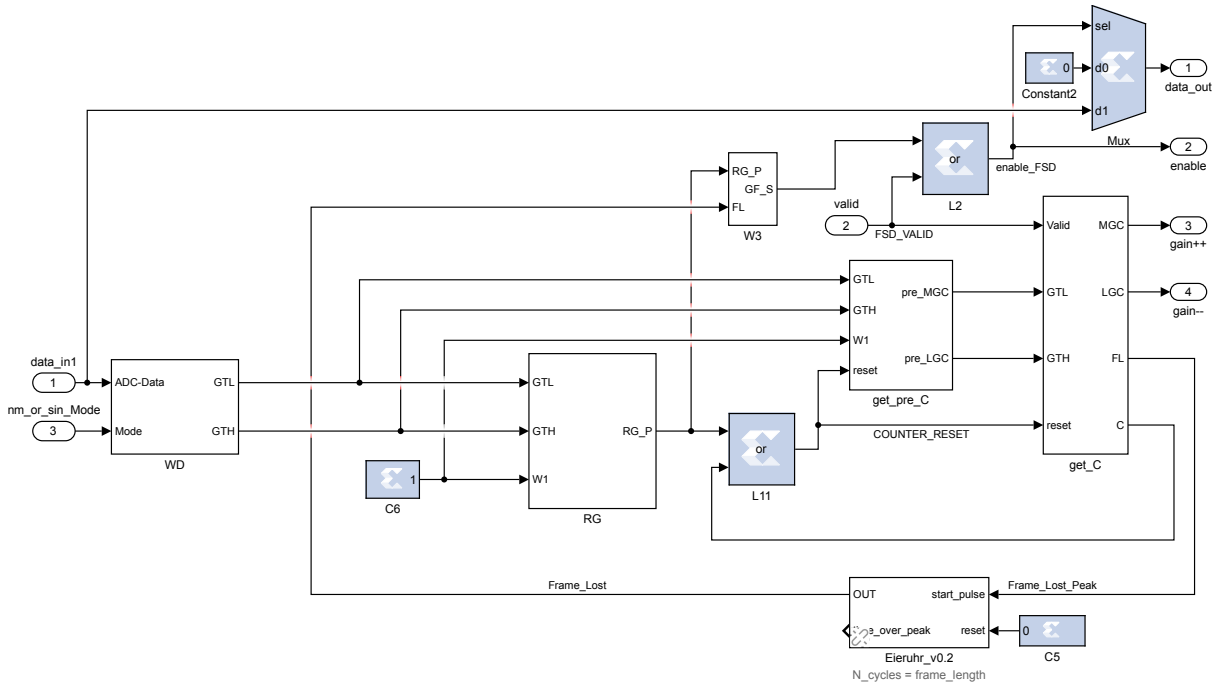


Abbildung 6.1: Blockschaltbild der AGC ChUp in SimulinkTM

6.2.1 Fenster Diskriminator (Window Detektion)

Um eine Übersteuerung des ADCs zu verhindern, muss die Amplitude des Eingangssignals ermittelt werden. Hierfür wird durch den Blocks *ADC2RMS* der quadratische Mittelwert (RMS) des ADC-Signals errechnet. Es wird die Quadratwurzel aus dem Quadrat des Eingangssignals gezogen, welches als Summe vieler cos- und sin-Funktionen entwickelt werden kann. Es gilt, wenn von einem reellen Signal ausgegangen wird:

$$\sqrt{\cos^2(x)} = \frac{1}{\sqrt{2}} + \sqrt{\frac{\cos(2x)}{2}} \quad (6.1)$$

Um den Gleichanteil zu messen, muss mittels eines FIR-Tiefpass (TP) die Oberwelle entfernt werden, siehe Formel 6.2. Um bei konstruktiver Interferenz im Guardintervall des Empfangssignals die Verfälschung des Messergebnisses zu minimieren, wird ein gleitender Mittelwert-Filter (MAV) verwendet. Die Störimpulse sind in Abbildung 9.3 links oben zu sehen. Sie sind Resultat des Einschwingvorgangs des Kanals und führen zu keinem Fehler. Zur ressourcenschonenden Implementierung dieses Filters wird die Samplerate vermindert: Mittels des Blocks *Down Sample* wird alle 256 Samples ein Wert ausgeschnitten, das

6 Realisierung

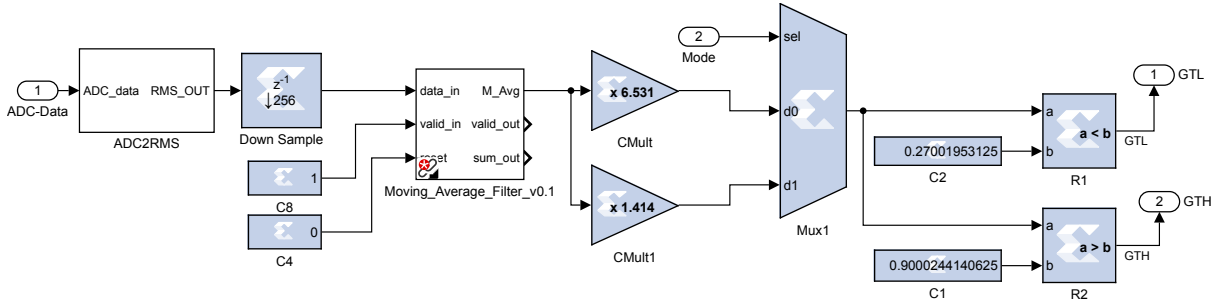


Abbildung 6.2: Blockschaltbild des *WD*-Blocks in SimulinkTM

Abtasttheorem von Shannon und Nyquist wird somit gezielt verletzt und Aliasing in geringem Maße in Kauf genommen. Denn einerseits wird schon im Block *ADC2RMS* eine Tiefpassfilterung vorgenommen, andererseits ist die Wahrscheinlichkeit, während eines Störimpulses abzutasten gering. Mittels des Blocks *Moving_Average_Filter_v0.1* wird die MAV-Filterung über acht Samples vorgenommen. Die Amplitude des Eingangssignals lässt sich mittels des Crest-Faktors errechnen. Nach idealer Tiefpassfilterung fallen hochfrequente Signalanteile weg:

$$\frac{1}{\sqrt{2}} + \sqrt{\frac{\cos(2x)}{2}} \xrightarrow[\text{TP}]{\text{FIR}} \frac{1}{\sqrt{2}} \quad (6.2)$$

Um die ursprüngliche Amplitude des Empfangssignals zu errechnen, wird mit dem Crest-Faktor k_s multipliziert. Bei einer harmonischen sin- oder cos-Schwingung gilt $k_s = \sqrt{2}$ und damit

$$\frac{1}{\sqrt{2}} \cdot k_s = 1. \quad (6.3)$$

Bei einem OFDM-Signal ist der Crest-Faktor von den Sendedaten abhängig. Bei stochastischen Sendedaten kann bei dem verwendeten Verfahren ein Crest-Faktor von

$$k_{s,OFDM} = 16 \text{ dB} = 10^{16/20} = 6,5 \quad (6.4)$$

angenommen werden (Siehe [19]). Kommt es zu Wiederholungen oder Mustern in den Daten, wird der Crest-Faktor schnell deutlich größer. Aus diesem Grund wird ein Scrambler eingesetzt. Bei dem schon in Teilen beschriebenen neuen Stepped Sub Carrier Sweep (SSCS)-Verfahren werden harmonische cos-Signale gesendet. Zur Umschaltung zwischen normalem Empfangsbetrieb und dem Messbetrieb wird der Block *Mux1* verwendet.

6 Realisierung

Die Festlegung des Spannungsfensters ist auch abhängig von der Schrittgröße der Verstärkungsanpassung. Um ein Hin- und Herspringen zwischen zwei Verstärkungsstufen zu vermeiden, muss für jeden Eingangspegel eine feste Verstärkung eingestellt werden können. Ist der Eingangspegel nahe des Schwellwertes, kann die Verstärkung geändert werden. Der resultierende Pegel soll einen Sicherheitsabstand zu dem Schwellwert einhalten.

In der Simulation wird von $g_{\pm} = \pm 6$ dB, also Faktor 2 pro Stufe ausgegangen. Daraus resultieren die Schwellwerte, welche über die Konstanten $C1$ und $C2$ eingestellt werden und mittels $R1$ und $R2$ überprüft werden. Die untere Schwelle ist als $s_{min} = 0,27$ der Maximalaussteuerung gewählt, da

$$y^* = s_{min} \cdot g = 0,54. \quad (6.5)$$

Bei einer oberen Schwelle von $s_{max} = 0,9$ resultiert ein Sicherheitsabstand von $\Delta y = s_{max} - y^* = 0,36$. Analog dazu resultiert aus

$$y^* = s_{max} \cdot g^{-1} = 0,46, \quad (6.6)$$

also einem Sicherheitsabstand von $\Delta y = s_{min} - y^* = 0,190$. Bei einer Schrittgröße von $g_{\pm} = \pm 6$ dB kann zwischen Wireline4k und Testkabel500 von einer Dynamik in Höhe von 58 dB $- 16$ dB $= 42$ dB, also sieben Verstärkungsstufen ausgegangen werden.

6.2.2 Right Gain Detektion

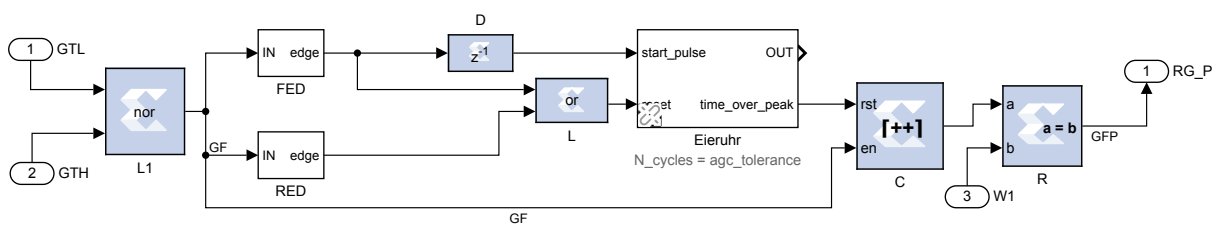


Abbildung 6.3: Blockschaltbild des RG -Blocks in SimulinkTM

Abbildung 6.3 zeigt das Blockschaltbild der Right Gain Detektion. Dieser Block wird benötigt, um bei einem Pegel nahe der Schaltschwelle sicher eine stabile und richtige Verstärkung zu detektieren.

Bei der richtigen Verstärkung ist weder GTL (Gain To Low) noch GTH (Gain To High)

6 Realisierung

gesetzt. Diese Logik wird mittels $L1$ umgesetzt. Ist GF (Gain Found) gesetzt, zählt C hoch. Nach der extern angelegten Wartezeit $W1$ wird der Impuls RG_P (Right Gain Peak) ausgegeben.

Um bei einem Pegel nahe der Schaltschwelle die Funktion der AGC zu gewährleisten wird, mittels der *Eieruhr* eine Toleranz gegenüber kurzzeitigem Verlassen des Fensters eingebaut. Problematisch bei solchen Pegeln ist, dass im Worst-Case-Szenario kein Kommando abgeschickt wird aber auch kein Signal durchgeschaltet wird, da dann alle Warte-Zähler durch Impulse unregelmäßig zurückgesetzt werden.

6.2.3 Vorläufige Kommandogenerierung

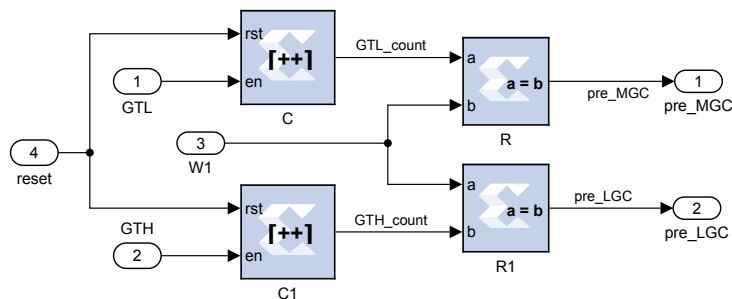


Abbildung 6.4: Blockschaltbild des get_pre_C -Blocks in Simulink™

Für die Generierung der Kommando zur Verstärkungsanpassung wird der Block get_pre_C benötigt, dessen Blockschaltbild in Abbildung 6.5 dargestellt ist. Der Reset der Warte-Zähler erfolgt unter anderem durch den soeben beschriebenen RG_P -Signals. Nach der extern angelegten Wartezeit $W1$ wird entweder der Impuls pre_MGC (More Gain Command) oder pre_LGC (Less Gain Command) ausgegeben.

6.2.4 Kommandogenerierung

Die vorläufige Kommandogenerierung funktioniert unabhängig vom $valid$ -Signal der FSD. Problematisch ist hierbei, dass kein Kommando zur Verstärkungsanpassung während der Laufzeit eines detektierten Senderrahmens umgesetzt werden darf. Der Kanal Equalizer stellt sich mittels der Präambel zu Beginn des Senderrahmens ein, bei einer Pegeländerung während eines Senderrahmens kommt es aber zu Datenverlust.

Aus diesem Grund werden nur während der Adaption der Verstärkung ohne PT direkt

6 Realisierung

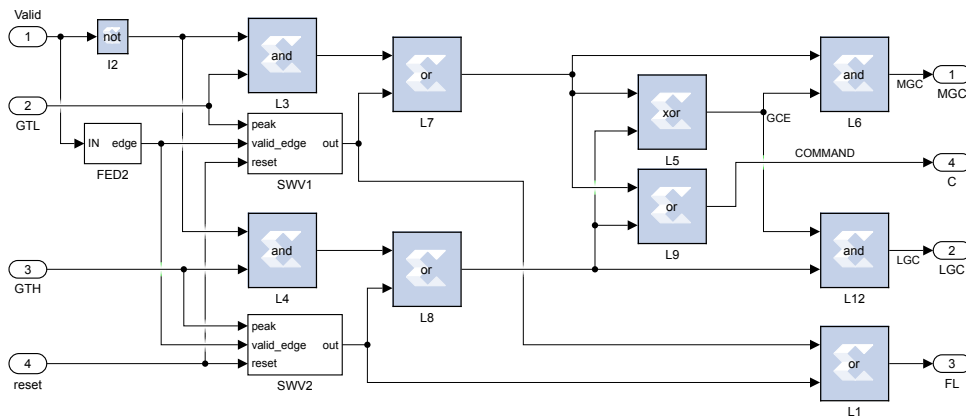


Abbildung 6.5: Blockschaltbild des *get_C*-Blocks in SimulinkTM

die Kommandos der *get_pre_C* ausgegeben. Muss im normalen Empfangsbetrieb die Verstärkung angepasst werden, ist mit hoher Wahrscheinlichkeit ein Fehler aufgetreten. Es wird trotzdem gewartet, bis der aktuelle Senderahmen abgelaufen ist und erst dann ein Kommando ausgegeben. Die normale Datenübertragung ist unterbrochen und der *FL* (Frame Lost)-Puls wird ausgegeben. So kann extern einen Timer gestartet werden, der den Ausgang während der Kommandolaufzeit deaktiviert.

Die Kommando De/Aktivierung erfolgt mittels einfacher boolescher Logik. Der Block *SWV* (Store While Valid) speichert ein Kommando der vorangestellten Stufe, wenn das *valid*-Signal gleich eins ist und gibt es bei einer fallenden *valid*-Flanke aus.

6.2.5 Simulationsergebnis

Abbildung 9.3 zeigt das Simulationsergebnis der AGC des ChUp. Die externe Vorverstärkung (r.m.) wird testweise verstellt. Die Verstärkung, welche mittels der AGC eingestellt wird (r.o.), sollte invers zur externen Vorverstärkung sein. Die Summe der Testverstärkung und der AGC-Verstärkung sollte also gleich Null sein.

Bevor ein Senderahmen erkannt wird, soll die Verstärkung direkt nachgeführt werden. Das Ausgangssignal der AGC ist in Abbildung 9.3 l.o. zu sehen. Bei einer Veränderung der Testverstärkung, die über den Fensterrahmen hinausgeht, wird der Ausgang abgeschaltet und die Verstärkung nachgeregelt. Die Steuerbefehle sind l.m. und l.u. zu sehen. Nachdem ein Senderahmen von der FSD erkannt wurde, werden keine Kommandos gesendet.

Erst nach der fallenden *valid*-Flanke (r.u.) wird das Kommando gegeben und der Ausgang

bis zur Umsetzung gesperrt.

6.2.6 Schnittstelle

Für die spätere Implementierung wird eine Schnittstelle benötigt, welche die Kommandos zur Anpassung der Verstärkung aus der Hardware in I²C-Anweisung umsetzt und diese so an den PIC auf der FAmP-Platine sendet.

6.3 Das SSCS-Verfahren

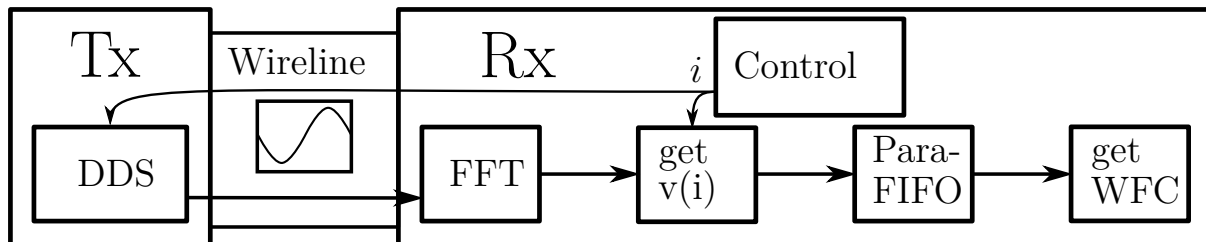


Abbildung 6.6: SSCS-Verfahren

Das SSCS-Verfahren kann in vier Schritte unterteilt werden, wobei Schritt vier in Sektion 6.4 beschrieben wird. Diese Aufzählung ist eine feinere Aufteilung des in Kapitel 5 beschriebenen Punktes 3a).

1. In einer Schleife werden alle SC mittels DDS und FFT Vermessen und im FIFO gespeichert.
2. Berechnung und Normierung der WFC.
3. Übertragung der WFC an den Tx.
4. Generierung des Headers nach Waterfilling (WF)-Spektrum im Tx.

Die Funktionalität der einzelnen Blöcke ist in Abbildung 6.6 schematisch dargestellt. Die DDS im Tx synthetisiert das angeforderte Signal. Die Messung erfolgt durch die FFT und den *get v(i)*-Block. Mit dem *Para-FIFO* werden die Kanalparameter zwischengespeichert und in *get WFC* in Waterfillingkoeffizienten umgewandelt. In Abbildung 9.4 ist das Simulations-Blockschaltbild des SSCS-Blocks zu sehen. Die AGC, sowie der Tx sind extern angeschlossen und die Gesamtsimulation ist in Abbildung 9.7 zu sehen.

6.3.1 Timing

Bis eine Messung gestartet werden kann, muss mindestens die Jitterzeit T_j der Datenübertragung plus der zu erwartenden Latenz T_L gewartet werden. Als zu erwartende Kommandoverzögerung muss also $CD = T_j + T_L$ gewählt werden. Ein Frame des ChDn dauert 2,4 ms. Ein Jitter von zwei Frames kann angenommen werden. Daraus ergibt sich eine Kommandoverzögerung von $CD^* \approx 10$ ms. Eine FFT-Periode mit 2048 Samples dauert

$$FFT_P = \frac{2048}{16 \text{ MHz}} = 128 \text{ } \mu\text{s}. \quad (6.7)$$

Somit gilt $CD^* \gg FFT_P$. Wird der nächste SC via ChDn angefordert, kann die vorherige Messung durchgeführt werden. Somit ergibt sich bei $N = 312$ SC eine Messdauer T_m von

$$T_m = N \cdot CD^* = 3,12 \text{ s}. \quad (6.8)$$

Anschließend werden die ermittelten Waterfilling-Koeffizienten versendet und im ChUp-Modulator eingepflegt. Das Blockschaltbild des Hauptverfahrens ist in Abbildung 9.7 zu sehen.

6.3.2 Anpassung an die FFT

Abbildung 9.5 zeigt den FFT-Block des Verfahrens. Die FFT wird sowohl von der Demodulation als auch vom SSCS-Verfahren verwendet. Aus diesem Grund wird durch den FFT_IN_MUX zwischen normalen Empfangsmodus (mode=0) und SSCS-Modus (mode=1) umgeschaltet.

Im SSCS-Modus wird direkt das ADC-Signal an den Eingang der FFT gelegt. Eine Messung wird gestartet, wenn ein Kommando verschickt wird. Über D ist CD^* als Verzögerung eingestellt und mit E_FM wird das *valid*-Signal der FFT generiert, welches 2048 Samples lang gesetzt ist.

Die Kanalmessung benötigt aufgrund der höheren Dynamik des Spektrums und der daraus resultierenden höheren Auflösung einen Bypass der FFT-Skalierung. Durch den Block Abs_Value wird das quadratische Mittel des FFT-Ausgangs gebildet.

6 Realisierung

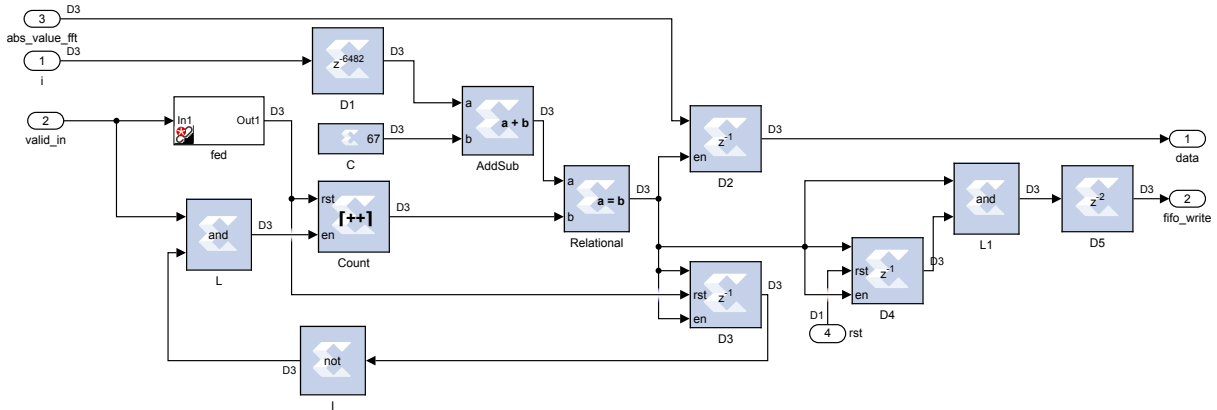


Abbildung 6.7: Blockschaltbild des $get_value(i)$ -Blocks in Simulink™

6.3.3 Auslesen der FFT

Der Block in Abbildung 6.7 zeigt das Auslesen der FFT an der i 'ten Stelle. Bei der FFT-Ausgabe ist die positive und negative Frequenzhälfte vertauscht. Dadurch beginnt das ausgegebene Spektrum mit der niedrigsten positiven Frequenz.

Das OFDM Verfahren des ChUp verwendet $N_{SC} = 312$ SC, beginnend bei $N_{SO} = 64$, also von SC $n_{SC} = N_{SO} = 64$ bis $n_{SC} = N_{SO} + N_{SC} = 376$. Der durch die Konstante C eingestellte Offset ergibt sich aus dem Sampleoffset N_{SO} plus einem Offset von drei Samples, wegen des valid-Signals der FFT.

Die Kommandoverzögerung CD_{nS} ergibt sich aus der Anzahl der äquivalenten Samples, also

$$CD_{nS} = CD \cdot f_a. \quad (6.9)$$

Bei der in Section 6.3 evaluierten Kommandoverzögerung ergibt sich damit

$$CD_{nS}^* = CD^* \cdot f_a = 10 \text{ ms} \cdot 16 \text{ MHz} = 16000. \quad (6.10)$$

Eine Kommandoverzögerung von 16000 Samples ist in der Simulation nicht praktikabel, da sich so die Simulationsdauer stark erhöht. Es wird also von einer Kommandoverzögerung $CD_{nS} = 3000$ ausgegangen.

Vor Beginn einer FFT-valid-Periode muss der richtige SC Index i gewählt sein. Die FFT verursacht eine Verzögerung von $N_{D,FFT} = 4232$ Samples. Die Verzögerung des Blocks $D1$ ergibt sich aus CD_{nS} und der FFT-Verzögerung. Somit wurde mittels $D1$ eine Verzögerung

6 Realisierung

von $N_{D1} = CD_{nS} + N_{D,FFT} - 750 = 6482$ realisiert.

Die Anzahl der validen FFT-Samples wird mittels des Zählers *Count* gezählt. So kann beim Erreichen des Referenzwertes der Betrag des empfangenen Sinus an der *i*.ten Stelle mit einem externen FIFO gespeichert werden. In *D2* wird der Wert zwischengespeichert und liegt an Port 1 *data* an. Danach wird an Port 2 *fifo_write* ein Puls zum Speichern angelegt. Wegen der Schleife aus *D3* und *I* bleibt der Zähler *Count* bei $n = i + 67 + 1$ stehen. Der Block *falling edge detection (fed)* generiert bei einer fallenden *valid*-Flanke einen Impuls, um alle Steuerspeicher und den Zähler in Reset zu versetzen.

6.3.4 Ablaufsteuerung

Mittels des Blocks *SSCS_Control*, dessen Blockschaltbild in Abbildung 9.6 zu sehen ist, wird der Ablauf der Kanalvermessung und Koeffizientengenerierung gesteuert. Hauptbestandteil dieser Steuerung ist die in Sektion 6.3.5 beschriebene *Statemachine*. Das Verfahren wird über den *Startpuls* gestartet. Anschließend ist es in vier *States* aufgeteilt:

1. *wait AGC*: Warten, bis die AGC des ChUp die richtige Verstärkung eingestellt hat.
2. *Measure Noise(MN)*: Der Rauschpegel wird gemessen.
3. *Normal Measurement(NM)*: Die Wireline wird vermessen.
4. *End*: Die Messung ist fertig.

Mittels das Logikblocks *L* wird der *SendC*-Puls generiert. Bei einem *SendC*-Puls soll extern der ausgegebene Subträgerindex *i* mittels des ChDn an den Tx des ChUp gesendet werden. Nach dem Senden eines Kommandos beginnt immer ein neuer Zyklus und/oder eine Messung. Die *Command Delay (CD)* wird abgewartet und die *Statemachine* mittels des *WAIT_DONE*-Pulse getriggert.

Der *WAIT_DONE*-Pulse startet außerdem den Timer *E_FM2*. Anschließend wird eine MAV-Periode der AGC abgewartet. Daraufhin wird gewartet, dass der Eingangspegel stimmt, was durch *AGC_EN* signalisiert wird. Die Erkennung dieser steigenden Flanke erfolgt mittels des *red*-Blocks und wird in *Store_Peak1* gespeichert. Das resultierende *AGC-valid*-Signal deaktiviert die AGC während einer Messung.

Zum Einpegeln der AGC wird der niederfrequenteste SC verwendet, da hierbei von der geringsten Dämpfung, also dem höchsten Pegel ausgegangen werden kann. Mit den

6 Realisierung

Logikblöcken $L1$, $L2$, $L7$ und $L8$ wird der Zähler $Count$ angesteuert. Die ermittelten Kanalparameter werden im $FIFO$ gespeichert. Nach Beendigung einer Messung erfolgt das Auslesen des $FIFO$ s.

Für eine günstigere Berechnung der Waterfillingkoeffizienten, beschrieben in Sektion 6.3.6, beginnt die Messung mit dem hochfrequentesten SC, also bei $i = 312$. SC 313 ist als „DDS aus“ definiert, um die Rauschleistung zu messen. Der SC Index wird also im Laufe einer Messung von $i = 313$ bis $i = 1$ dekrementiert.

6.3.5 State machine

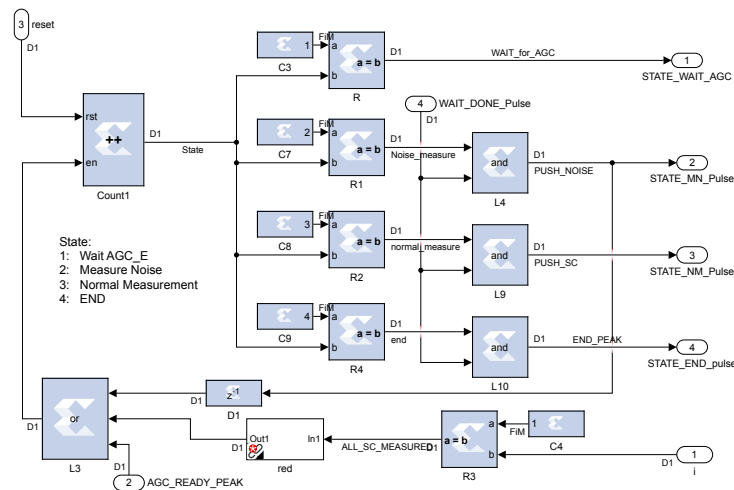


Abbildung 6.8: Blockschaltbild der *State machine*-Blocks in Simulink™

In Abbildung 6.8 ist der Aufbau des *State machine*-Blocks der Steuerung des SSCS-Verfahrens zu sehen. Die dreimalige Inkrementierung des States erfolgt durch den Zähler $Count1$ über den Logikblock $L3$. Der Übergang zu State zwei wird mittels des AGC_READY -Peak ausgelöst. Dann erfolgt nur eine Rauschpegelmessung und es wird direkt von State zwei zu State drei übergegangen. Das Erreichen des letzten SC bei $i = 1$, wird mittels $R3$ registriert und es wird zu State vier gewechselt.

6.3.6 Koeffizientengenerierung

Die Waterfillingkoeffizienten ergeben sich aus dem Kehrwert der Kanalparameter und werden auf auf den Höchstwert normiert. Die dafür nötige Signalflussdiagramm ist in

6 Realisierung

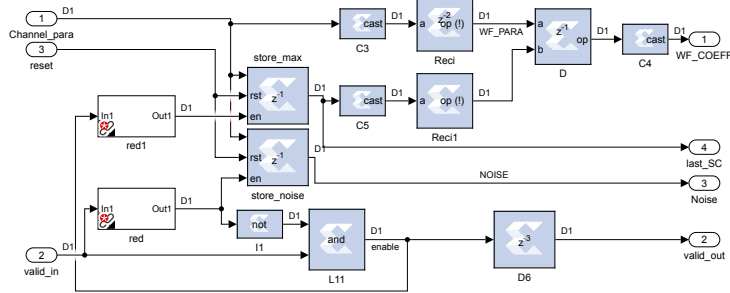


Abbildung 6.9: Blockschaltbild der Koeffizientengenerierung in Simulink™

Abbildung 6.9 zu sehen. Der erste Wert im *FIFO* ist die Rauschmessung, welche mittels *store_noise* gespeichert wird. Zu erwarten ist, dass der hochfrequenteste SC den kleinsten Pegel hat, also normiert man auf ihn. Außerdem kann aus dem Verhältnis vom hochfrequentesten SC zum Rauschen stichprobenartig das SNR ermittelt werden. Um die Rauschmessung am Koeffizientenausgang zu ignorieren, wird mittels *I1* und *L11* das erste Sample des *valid*-Signals entfernt. So kann mittels *red1* der Pegel des hochfrequentesten SC gespeichert werden.

6.3.7 DDS im Sender

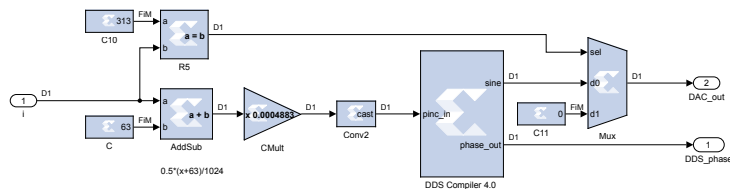


Abbildung 6.10: Blockschaltbild der DDS im Tx in Simulink™

Die Generierung der benötigten sin- oder cos-Funktion wird mit einer DDS realisiert. Die Beschaltung ist in Abbildung 6.10 zu sehen. Das Phaseninkrement P_{ink} lässt sich aus der FFT Sampleanzahl N_{FFT} und der Abtastrate f_a mittels Dreisatz herleiten. Jeder SC kann als ein FFT Sample aufgefasst werden. Bei der Festlegung auf ein reelles Ausgangssignal können

$$N_{FFT}^* = \frac{N_{FFT}}{2} \quad (6.11)$$

SC verwendet werden. Ein Phaseninkrement von $P_{ink} = 1$ würde zu einer DDS Ausgangs-

6 Realisierung

frequenz von $f_{DDS} = f_a$ führen, was zu Fehlern führt. Aus einem Phaseninkrement von $P_{ink} = 0,5$ folgen zwei Samples pro Periode also einer Ausgangsfrequenz von $f_{DDS} = f_a/2$. Der SC mit $n = N_{FFT}^*$ entspricht also einem Phaseninkrement von $P_{ink} = 0,5$. Es folgt also

$$2 \cdot P_{ink} = \frac{n}{N_{FFT}^*}. \quad (6.12)$$

Wobei n durch den SC Index i ausgedrückt werden kann:

$$n = i + 63 \quad (6.13)$$

Daraus folgt, wenn $N_{FFT} = 2048$:

$$\frac{P_{ink}}{0,5} = \frac{i + 63}{N_{FFT}^*} \iff P_{ink} = \frac{i + 63}{2 \cdot N_{FFT}^*} = \frac{i + 63}{N_{FFT}} = \frac{i + 63}{2048} \quad (6.14)$$

Mittels *AddSub* und der Konstanten C wird die benötigte Summe berechnet. Der Faktor $2048^{-1} = 4,88 \cdot 10^{-4}$ kann mittels *CMult* realisiert werden. Um das Abschalten der DDS bei $i = 313$ zu gewährleisten, kann mit dem *Mux* das DAC-Signal zu konstant null umgeschaltet werden.

6.3.8 Simulation und Ergebnis

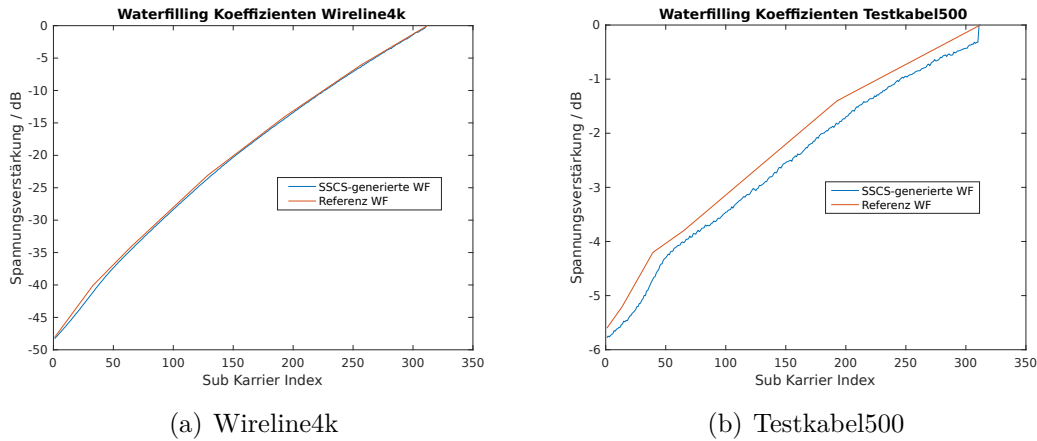


Abbildung 6.11: Vergleich der Referenz und SCS generierten Waterfillingkoeffizienten bei SNR=25 dB

Um eine praxisnahe Simulation der Messung durchzuführen, wird ein AWGN-Kanal mit

6 Realisierung

25 dB SNR angenommen. In Abbildung 6.11 sind die mittels Matlabskript generierten Waterfillingkoeffizienten, die als Referenz verwendet werden können (orange), und die mit dem SSCS-Verfahrens generierten Waterfillingkoeffizienten (blau) zu sehen. Als Ergebnis kann man feststellen, dass sowohl die *Wireline4k* als auch das *Testkabel500* durch SSCS adäquat abgebildet werden können. Zu beachten ist die unterschiedliche Achsenformatierung, weshalb der geringe Offset in (a) nicht auffällt.

Der Hauptblock der SSCS-Simulation ist in Abbildung 9.7 zu sehen. Im unteren Bereich befindet sich der Tx, auf dessen Realisierung in Sektion 6.4 weiter eingegangen wird. Für eine unkomplizierte Simulation wird für die Verfahrensentwicklung nicht der komplette Modulator des ChUp eingesetzt, sondern nur die DDS. Es wird auch kein kompletter Demodulator eingesetzt, und somit müssen die unbelegten Eingänge des für das SSCS-Verfahren abgewandelte *SSCS_FFT*-Blocks mit Konstanten belegt werden.

Die in Sektion 6.2 beschriebene AGC ist ein essentieller Bestandteil des SSCS-Verfahrens und wird daher auch in die Verfahrenssimulation eingebunden.

6.3.9 Implementierung in den vorhandenen Empfänger

In Abbildung 9.10 ist die Implementierung des SSCS-Verfahrens in den Rx zu sehen. Alle aus Abbildung 2.1 bekannten Blöcke sind invers auch hier zu finden, zusätzlich ist hier die in Sektion 6.2 realisierte AGC, die *Framestart_detection* und das *Cyclic_Prefix_Remove* zu sehen. Mittels des *Cyclic_Prefix_Remove* wird das Guardinterval entfernt (vgl. Sektion 2.6). Bei einem Puls an *SSCS_start_pulse* beginnt die SSCS Kanalmessung mit anschließender Koeffizientengenerierung, ansonsten wird normal empfangen.

6.4 Datenkanal Sender

Nach der erfolgreichen Messung, Koeffizientengenerierung und Übertragung muss der Waterfillingkoeffizientenvektor in den Tx des ChUp eingepflegt werden. Im bisherigen Verfahren mit festen WFC für die Daten ist die Umstellung auf variable WFC trivial, statt ROM wird ein Random-Access Memory (RAM) verwendet. Problematisch ist die Anwendung der WFC auf den Frame-Header, da dieser bisher aus einem ROM im Zeitbereich abgespielt wird. Um dessen Spektrum anzupassen, muss der Header via IFFT neu generiert und im RAM abgelegt werden. Hierzu wurde in den Tx des ChUp ein Verfahren integriert (Siehe Abbildung 9.8). Für ein besseres Verständnis der beschriebenen Blöcke

6 Realisierung

kann auch Abbildung 2.1 zur Hilfe genommen werden.

6.4.1 Ablaufsteuerung

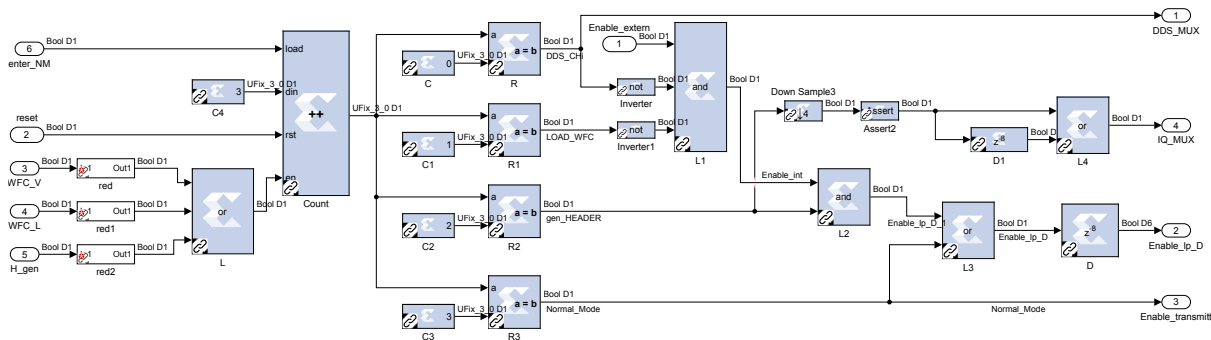


Abbildung 6.12: Blockschaltbild der Steuerung des Tx in SimulinkTM

Es werden vier States des in Abbildung 6.12 dargestellten *Control*-Blocks festgelegt:

1. *DDS_CHi*, hier wird während der Kanalmessung die DDS an den DAC durchgeschaltet.
2. *LOAD_WFC*, die Waterfillingkoeffizienten werden geladen.
3. *gen_HEADER*, der Header wird generiert und im RAM des *Frame Build*-Blocks gespeichert.
4. *Normal Mode*, also der normale Datensendebetrieb.

Der *control*-Block ist in Abbildung 9.8 links unten zu sehen. Nach dem Einschalten wird von State eins ausgegangen, die DDS ist an den DAC geschaltet und erwartet Befehle des Rx. Ist die Messung vorüber, werden die Waterfillingkoeffizienten mittels ChDn verschickt und anschließend aus dem Empfangs-FIFO ausgelesen. Bei einer steigenden *WFC_V* (Waterfilling Coefficients Valid)-Flanke erfolgt der Übergang in State zwei und die Waterfillingkoeffizienten werden geladen.

Durch die steigenden *WFC_L* (Waterfilling Coefficients Loaded)-Flanke kann ermittelt werden, wann alle WF-Koeffizienten geladen sind. Es folgt somit State drei, *gen_HEADER*. Hier wird aus der Präambel und dem Pilot im Frequenzbereich, mittels der vorhandenen Bausteine (IFFT, etc.), der Header im Zeitbereich generiert, und im RAM des *Framebuild*

6 Realisierung

abgelegt. Die steigende H_gen (Header generated)-Flanke signalisiert, dass der Tx bereit für die Datenübertragung ist und es folgt State vier *Normal Mode*.

Bei einer bekannten Übertragungstrecke soll die Möglichkeit bestehen, direkt in State vier zu wechseln. Hierfür wird ein Zähler $Count$ mit $load$ und din -Eingang verwendet, um bei einem Signal auf $enter_NM$ (enter Normal Mode) direkt den Ausgang zu belegen.

6.4.2 Waterfillingkoeffizienten laden

In Abbildung 9.9 ist der Aufbau der *Waterfillingmoduls* zu sehen. Mittels des B -Port des *Dual Port RAM* werden die Waterfillingkoeffizienten geladen, indem bei WF_COEFF_VALID gleich eins die RAM Adresse mittels $Count2$ dekrementiert. Initialisiert wird $Count2$ mit $n = 312$. Durch die Dekrementierung beim Einlesen und die Inkrementierung beim Auslesen wird der WF-Koeffizientenvektor gespiegelt. Das ist nötig, da wie in Sektion 6.3.6 beschrieben, die Messung mit dem hochfrequentesten SC, die Anwendung der WFC aber beim niederfrequentesten SC beginnt.

Das vollständige Laden der Koeffizienten kann mittels das Unterlaufs von $Count2$ detektiert werden. So wird mittels C und R das WFC_LOADED -Signal für die Steuerung generiert.

Mittels des A -Ports werden die Koeffizienten ausgelesen. Die IQ-Symbole werden in (I)FFT-Perioden verarbeitet, also 312 Symbole für 312 SC. Die Multiplikatoren $Mult$ und $Mult1$ verwenden die WF-Koeffizienten aus dem *Dual Port RAM*.

6.4.3 IFFT Skalierung

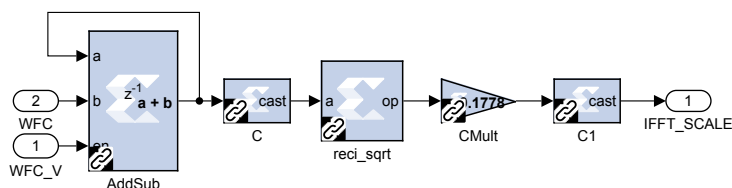


Abbildung 6.13: Blockschaltbild der IFFT-Skalierungsberechnung in SimulinkTM

Nach [13] wird die IFFT mittels

$$\alpha_{IFFT,1} = \frac{1}{N} \quad (6.15)$$

6 Realisierung

normiert. Bei konstruktiver Interferenz ergibt sich aus der normierten IDFT Summe des spektralen Datenvektors a_j der Zeitbereichsdatenvektor a_k

$$a_k = \alpha_{IFFT,1} \sum_{i=0}^{N-1} \exp\left(-2\pi j \cdot \frac{ik}{N}\right) \cdot a_j \quad (6.16)$$

maximal

$$a_{k,\max} = \alpha_{IFFT,1} \sum_{i=0}^{N-1} 1 = \frac{1}{N} \cdot N = 1. \quad (6.17)$$

Durch die verwendeten WFC lässt sich die Normierung präzisieren, da hier die Maximalamplitude durch den WFC an der Stelle i vorgeben ist, so ergibt sich:

$$\alpha_{IFFT,1}^* = \left(\sum_{i=0}^{N-1} WFC_i \right)^{-1} \quad (6.18)$$

Der maximale Crest-Faktor $k_{s,\max}$ des Zeitsignals lässt sich mittels des IFFT Normierungsfaktor α_{IFFT} berechnen. Es wird von einem unskalierten IFFT-Signal ausgegangen.

$$k_{s,\max} = \frac{|X|_{\max}}{|X|_{\text{eff}}} = \frac{1}{\frac{\alpha_{IFFT}}{N \cdot T_a}} = N \quad (6.19)$$

Ist der Crest-Faktor eines Spektrums bekannt, so kann mit diesem Faktor multipliziert werden.

Es gilt unskaliert

$$IFFT(FFT(x)) = x \cdot N. \quad (6.20)$$

Wird der benötigte Normierungsfaktor $1/N$ auf IFFT und FFT aufgeteilt und auf Grund der Summenstruktur als Skalar verschoben ergibt sich

$$\frac{IFFT\left(\frac{FFT(x)}{\sqrt{N}}\right)}{\sqrt{N}} = x. \quad (6.21)$$

Die Normierung erfolgt also nach [21] auf $1/\sqrt{N}$. Durch die Verteilung auf das Spektrum mittels der Basisbandtransformation und der Verwendung der WFC ergibt sich:

$$\alpha_{IFFT,2} = \left(2 \cdot \sum_{i=0}^{N-1} WFC_i \right)^{-\frac{1}{2}} \quad (6.22)$$

6 Realisierung

Experimentell wurde ein weiterer Skalierungsfaktor zur optimalen Aussteuerung des DAC ermittelt und so die in Abbildung 6.13 zu sehende Formel mit folgender Arithmetik entwickelt.

$$\alpha_{IFFT,impl} = 10^{-\frac{15}{20}} \cdot \left(\sum_{i=1}^{312} WFC(i) \right)^{-\frac{1}{2}} \quad (6.23)$$

Somit kann aus den WFC die implementierte IFFT-Skalierung errechnet werden.

6.4.4 Header generieren

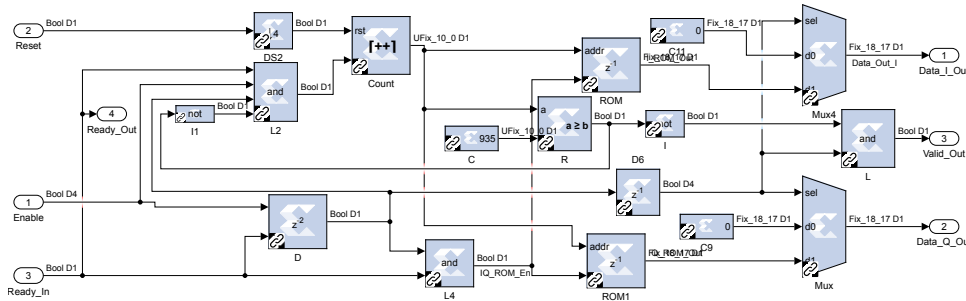


Abbildung 6.14: Blockschaltbild des ROM Speichers des Headers des Tx in Simulink™

Die verwendete Präambel und der Pilotton liegen als Basisbandsignal für alle 312 SC vor. Mittels der IFFT kann also sowohl die Präambel als auch der Pilotton im Zeitbereich generiert werden. Die vorhandene Bausteine können verwendet werden, es muss allerdings umgeschaltet werden, dies wird durch die Mux-Blöcke in Abbildung 9.8 ermöglicht. In Abbildung 6.14 ist die Simulink-Schaltung des ROM-Speichers des Headers zu sehen. Es müssen drei Basisband (I)FFT-Perioden bearbeitet und transformiert werden. Daraus ergibt sich eine ROM-Größe von

$$N_{ROM} = 3 \cdot 312 = 936. \quad (6.24)$$

Mittels des Zählers *Count* wird über die *Ready*-Rückkopplung und das *Enable*-Signal die ROM-Adresse inkrementiert, bis bei $n = N_{ROM} - 1$ alle Werte abgefragt wurden. So werden die Pilot- und Präambel-Symbole an die Basisbandtransformation übergeben.

6 Realisierung

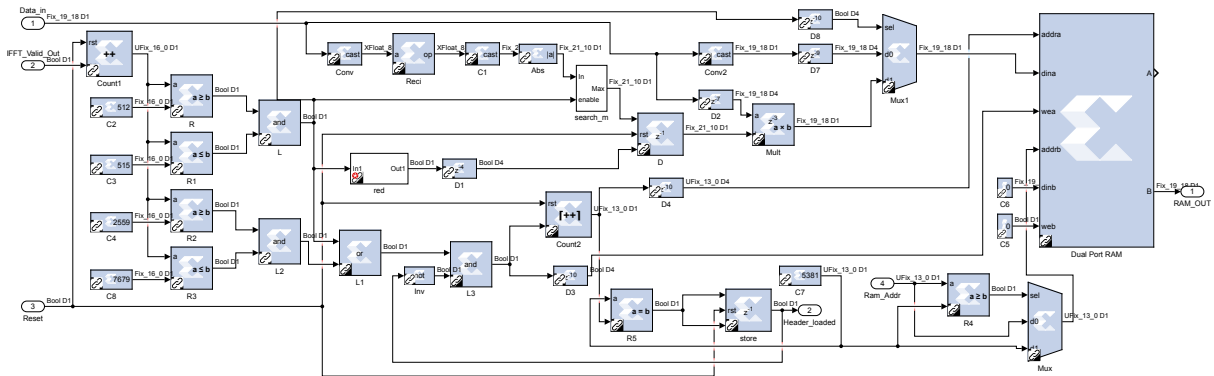


Abbildung 6.15: Blockschaltbild des RAM Speichers des Headers des Tx in Simulink™

6.4.5 Header auslesen

Nach dem Block $tdV2tD$, also nach der Anwendung der WFC, der Basisbandtransformation und der IFFT müssen die richtigen Zeitbereich-Samples gespeichert werden, um sie später im Normalbetrieb als Frameheader abzuspielen.

Der Zähler *Count1* zählt die Samples für, die *IFFT_Valid_Out* gleich eins gilt. Eine IFFT-Periode wird durch das Guardintervall, siehe Sektion 2.6, um 512 Samples verlängert indem die letzten Samples einer Periode an den Anfang gestellt werden. Nach der IFFT ist der Pilotton Sample eins bis vier im Zeitbereich. Bei einem Guardintervall der Länge 512 muss der Pilotton nach der Transformation Sample 512 bis 515 sein. Mittels *C2*, *C3*, *R* und *R1* wird ein Valid-Signal für den Pilotton generiert. Die vier Samples des Pilottons werden ähnlich wie in Sektion 6.1 beschrieben auf eins normiert. Es wird allerdings erst der Reziprokwert gebildet und anschließend mittels *search_m* das Minimum gesucht.

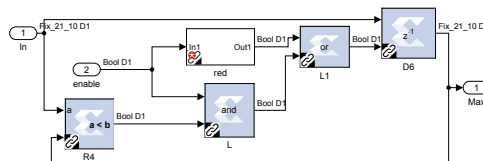


Abbildung 6.16: Blockschaltbild des *search_m*-Blocks in Simulink™

Der interne Aufbau dieses Blocks ist in Abbildung 6.16 zu sehen. Um das Minimum zu suchen, muss der erste valide Wert gespeichert werden, da ansonsten der Defaultwert null der kleinste ist. Durch *red* wird dies verwirklicht. Anschließend wird der Speicher *D6* nur enabled, wenn $In < Max$ und somit den aktuellen *In*-Wert speichert.

6 Realisierung

Da während eines Frames auch kleinere Reziprokwerte anliegen können, wird mittels *red* und *D1* der Speicherimpuls für *D* erzeugt. So kann der Normierungsfaktor des Pilottons gespeichert werden. Normiert wird mittels *Mult*. Die Umschaltung von Pilotton zu normaler Präambel erfolgt durch *Mux1*. Die Verzögerung der Minimalwertsuche *D2* und der Multiplikation *Mult* wird von *D7*, *D8*, *D4* und *D3* ausgeglichen. Die Generierung des *wea*(write enable)-Signal des *Dual Port RAM* erfolgt durch die Logik aus *L1* und *L3*. Durch *L3* wird verhindert, dass bei einem geladenen Header der *Dual Port RAM* überschrieben wird.

Die RAM-Adresse wird von *Count2* inkrementiert. Die Framepause ist auf 256 Samples festgesetzt, daraus resultiert der Startwert von *Count2* mit 256. Nach

$$C4 = 256 + 4 + 2 \cdot (2048 + 515) + 1 = 5381 \quad (6.25)$$

Samples ist der Header geladen. Dies wird von *R5* erkannt und mit *store* gespeichert. Um beim Auslesen des RAMs den Datenbereich nicht zu überschreiten, wird durch *R4* die Adresse mit der Maximaladresse verglichen und mit *Mux* ausgewählt.

Nach 2560-Pilot-IFFT-Samples, also nach Sample $C4 = 2559$, beginnen die Präambeln. Wegen der Frequenz Offset Korrektur, siehe Sektion 2.9.2, werden zwei Präambeln mit Guardintervall verwendet. Die nun folgenden

$$2 \cdot (2048 + 515) = 5120 \quad (6.26)$$

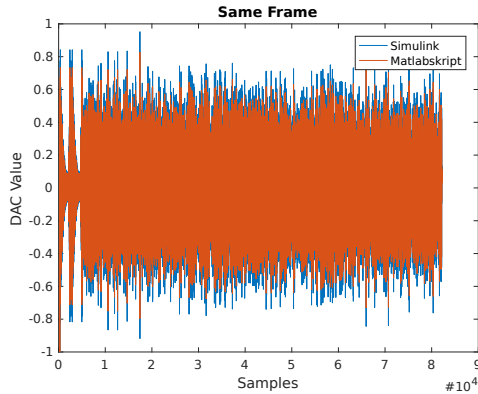
Samples müssen also direkt ins RAM geschrieben werden. Für die Konstante *C8* gilt also:

$$C8 = 5120 + C4 = 7679 \quad (6.27)$$

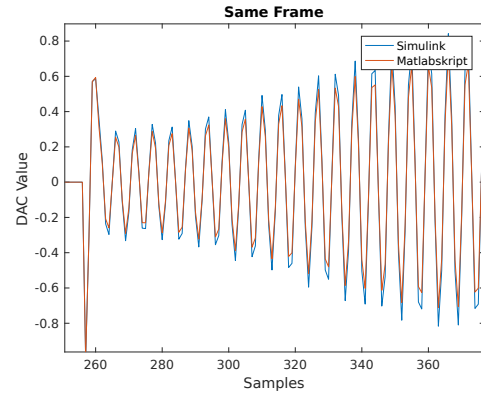
6.4.6 Simulationsergebnis

Zur Verifikation der Simulink™-Simulation wird der entstandene Frame mit einem Script generierten Frame gleicher Daten verglichen. Dieser kann als Referenz verwendet werden. In Abbildung 6.17 (a) und (c) ist der gesamte Referenzframe (orange) und generiert (blau), einmal für die Wireline4k und für das Testkabel500 zu sehen, in (b) und (e) ist ein Ausschnitt des Referenzframes zu sehen. Auffallend ist eine leicht unterschiedliche IFFT-Skalierung. Der Pilotton liegt allerdings exakt übereinander, was auf die Normierung zurückzuführen ist.

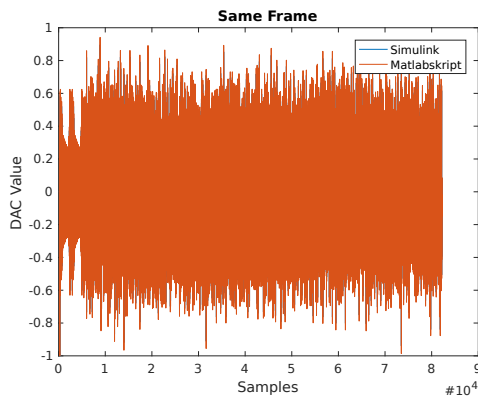
6 Realisierung



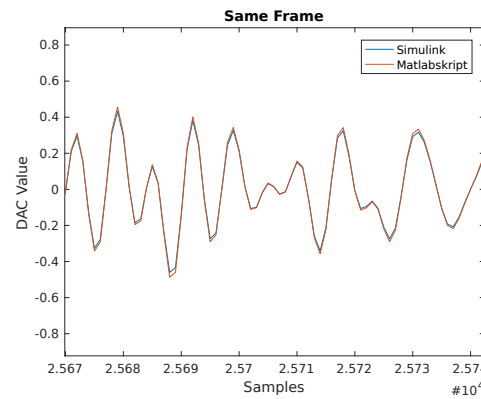
(a) Waterfilling Wireline4k gesamtes Frame



(b) Waterfilling Wireline4k Ausschnitt



(c) Waterfilling Testkabel500 gesamtes Frame



(d) Waterfilling Testkabel500 Ausschnitt

Abbildung 6.17: Vergleich eines Tx Frames der HHI-Matlabskript Simulation und der SCS Simulink Simulation

Zur Verifikation des generierten Frames wird ein Empfang simuliert. Durch einen FIR lässt sich die Impulsantwort der Wireline auf des Sendesignal anwenden und anschließend mittels des Rx in Simulink empfangen. Der Empfang und die Synchronisation funktionieren einwandfrei. In Abbildung 6.18 ist der Realteil des Ausgangssignals der Präambel nach der FFT im Rx zu sehen, durch das WF ist die Empfangsleistung konstant verteilt.

6 Realisierung

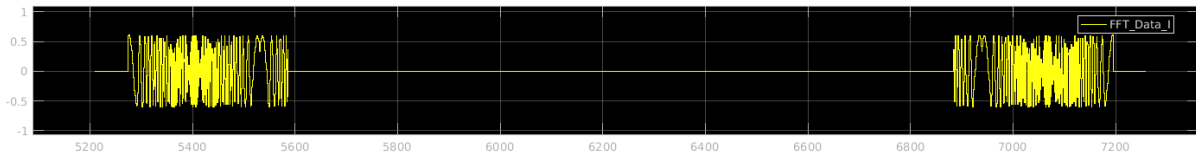


Abbildung 6.18: Exemplarische FFT-Ausgabe im Rx der Präambel

Zusammenfassend wurde

1. die Wireline4k und das Tsstkabel500 mittels SSCS vermessen,
2. die WFC passend zu der verwendeten Wireline ermittelt,
3. die ermittelten WFC genutzt, um einen Frame-Header im Tx zu generieren,
4. ein Tx Frame mit dem eingestellten Leistungsspektrum generiert und
5. dieses Frame jeweils mittels des modifizierten Rx empfangen.

In der Simulation funktioniert das vorgeschlagene Verfahren also.

7 Implementierung

7.1 AGC des Steuerkanal Empfängers

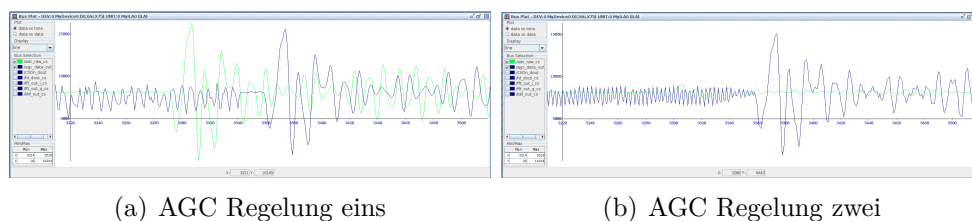


Abbildung 7.1: AGC ChDn in Aktion (aufgenommen mit Chipscope™)

Die AGC des ChDn wird mittels des Xilinx-Systemgenerators™ in VHDL generiert. Zur Erprobung des generierten IP-Cores kann ein vorhandenes VHDL-Projekt verwendet werden, in welches der ChDn Rx mit digitaler AGC eingebunden wird. Mittels Chipscope™ können Binärsignale während des Betriebs des FPGAs aufgezeichnet werden. So wurde in Abbildung 7.1 zwischen zwei Verstärkerstufen der FAMP Platine umgeschaltet. In (a) ist die Verstärkung hoch eingestellt, die Amplitude bleibt weitgehend unverändert. In (b) wurde die Verstärkung der FAMP-Platine um 20 dB abgesenkt. Das Ausgangssignal (blau) bleibt unverändert in der Amplitude.

7.2 Datenkanal Sender

Alle in Sektion 6.4 beschriebenen Erweiterungen führen zu vier zusätzlich benötigten DSP-Slices. Der Begriff DSP-Slice beschreibt einen 19x19 Bit Multiplizier-Hardware-Block im FPGA. DSP-Slices werden bei häufigem Bedarf an Multiplikationen eingesetzt. Der verwendete Spartan-6 FPGA bietet 134 DSP-Slices. Hierbei werden 96 DSP-Slices verwendet.

7.3 Datenkanal Empfänger

Da im Rx deutlich mehr Berechnungen (z.B. WFC-Generierung, AGC und Normierung der WFC) werden hier 113 DSP-Slices verwendet. Andere FPGA Ressourcen sind deutlich weniger beansprucht, weswegen sie hier nicht weiter beachtet werden. Dies sind aber noch zu viele um den ChUp Rx mit dem ChDn Tx zu kombinieren.

7.4 SSCS-Koeffizientengenerierung

Es soll eine SSCS-Basierte Koeffizientengenerierung mit Hardware durchgeführt werden. Da noch keine I²C Interface zwischen SSCS - Rx und AGC des ChUp besteht und außerdem das ChDn-Modem noch nicht zusammen mit dem ChUp-Modem in einen FPGA passt wird eine vereinfachte Messung der WFC durchgeführt. Der in Abbildung 9.4 zu sehende *SSCS-FFT*-Block kann ohne den ChUp Demodulator verwendet werden und wird in ein einfaches VHDL-Projekt implementiert. Der *start_m_pulse* kann extern angelegt werden und startet die Messung.

Ein einfacher Sender kann mit dem Block-Diagramm in Abbildung 9.8 generiert werden. Da keine WFC geladen werden und kein *enter_nm*-Signal angelegt wird ist die DDS immer an den DAC geschaltet. Ohne ChDn muss die Synchronisation über I/O-Ports der HiTES-Platine erfolgen. Sowohl Sweep als auch Messung werden mittels einer I/O-Leitung gestartet, welche gegen logisch eins geschaltet wird.

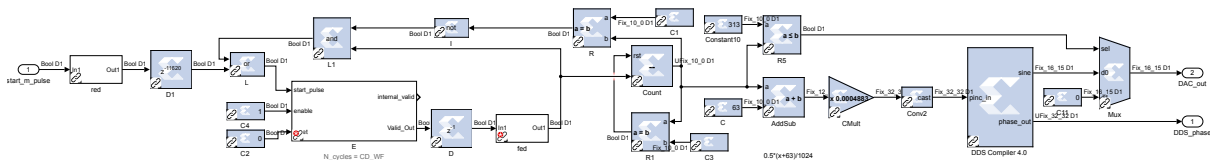


Abbildung 7.2: Blockschaltbild der veränderten DDS in SimulinkTM

Zur Generierung des Sweeps muss die eigentlich durch den ChDn gesteuerte DDS automatisiert werden (Siehe Abbildung 7.2). Mittels *Count* wird der SC-Index i dekrementiert bis $i = 1$. Die Schleife aus E , D , fed , $L1$ und L steuert den Zähler. Damit ein Puls und kein dauerhafter high-Pegel am *start_m_puls* sicher gestellt werden kann wird der *red*-Block eingesetzt. $D1$ verzögert den Startpuls um das Einpegeln der AGC zu simulieren und die Rauschpegelmessung zu überspringen. Die Implementierung erfolgt durch Einbindung in ein VHDL-Projekt.

7 Implementierung

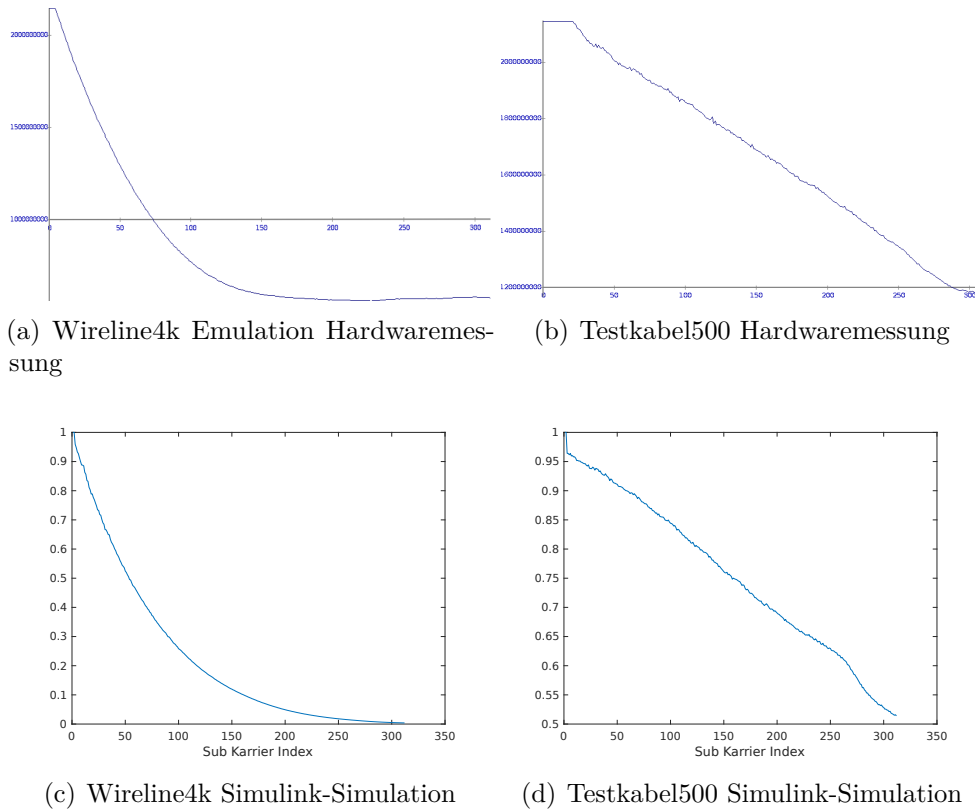


Abbildung 7.3: Vergleich der Hardwareimplementierung und der Simulation

Das Messergebnis ist in Abbildung 7.3 zu sehen. Ohne AGC ist der Eingangspegel schlecht zu regeln, deshalb ist in (a), bei den WFC des Testkabel500, klipping festzustellen. Um die Messung zu ermöglichen wird der Pegel mittels Dämpfungsgliedern eingestellt. Im Vergleich von (a) zu (c) und von (b) zu (d) fällt eine eindeutige Ähnlichkeit auf. Durch die direkte Ausgabe der 32 Bit Zahl in (a) und (b) als Unsigned stimmt die Skalierung der Achsen nicht. Es müsste gelten $y^* = y/2^{32}$.

Es wurde gezeigt das das SSCS-Verfahren auch in in der Realität mit Kabel-Übertragungsstrecke funktioniert. Es bedarf allerdings noch einiger Optimierung.

8 Fazit

Ziel dieser Arbeit ist die automatische Kabeladaption einer OFDM-Datenübertragung an verschiedene Kabel einer Powerline-versorgten Bohrlochsonde. Es wird eine Pegelanpassung und eine Optimierung des Sendeleistungsspektrums benötigt. Dafür wurde ein Adaptionverfahren mit Matlab-Simulink[™] entwickelt:

Zuerst muss sich die AGC des Steuerkanal Host-Sonde (ChDn) automatisch einpegeln. Aufgrund der Schmalbandigkeit des ChDn sind hier keine weiteren Maßnahmen nötig.

Dann folgt die Analogverstärkungsanpassung via AGC des Datenkanals Sonde-Host (ChUp). Nach einer Phase der Kanalmessung werden Waterfilling Koeffizienten (WFC) ermittelt und via ChDn an den Sender (Tx) des ChUp übermittelt. Dort erfolgt die Anwendung der WFC auf den Header und es kann normal gesendet werden.

Das Adaptionverfahren, einschließlich AGCs, Kanalmessung, WFC- und Header-Generierung, wurde vollständig neu entwickelt, simuliert und in die bestehenden Modems integriert. Bei der Suche nach einem passenden Kanal-Messverfahren wurden verschiedene Ansätze diskutiert. Daraus ergab sich das hier so genannte Stepped Sub Carrier Sweep (SSCS)-Messverfahren, bei dem schmalbandig alle Sub Carrier (SC) der Breitband-OFDM-Datenübertragung des ChUp vermessen werden.

Die AGC des ChDn und die SSCS-Kanalmessung mit WFC-Generierung wurden in VHDL implementiert, in VHDL-Projekte eingebunden und als IP-Cores auf dem FPGA realisiert. Außerdem wurde die SSCS-WFC-Messung jeweils in Realität mit der 4,2 km-Kabelemulation und einem 500 m Kabel getestet und mit dem Simulationsmodell verglichen.

Zur vollständigen Inbetriebnahme des Verfahrens muss eine I²C-Schnittstelle zwischen AGC des ChUp und der Filter- und Verstärkerplatine (FAmp) entwickelt werden. Der Ressourcenbedarf im FPGA ist in Teilen zu hoch und es besteht Optimierungsbedarf. Man kann z.B. Floating- auf Fixed-Point-Berechnungen umstellen.

Zusammenfassend kann gesagt werden, dass das Adaptionverfahren realisierbar ist und in Teilen bereits in der Praxis funktioniert.

Literatur

- [1] Mathias BAUER. *Handbuch Tiefe Geothermie : Prospektion, Exploration, Realisierung, Nutzung*. Hrsg. von Willi [Hrsg.] FREEDEN, Hans [Hrsg.] JACOBI und Thomas [Hrsg.] NEU. Berlin, Heidelberg, 2014. URL: <http://swbplus.bsz-bw.de/bsz40957533xcov.htm><http://dx.doi.org/10.1007/978-3-642-54511-5> [siehe S. 2].
- [2] Martin BOSSERT. *Kanalcodierung*. 3., überarb. Aufl. München: Oldenbourg, 2013. ISBN: 978-3-486-72128-7. URL: http://deposit.d-nb.de/cgi-bin/dokserv?id=4211533&prov=M&dok_var=1&dok_ext=htm [siehe S. 7].
- [3] Analog DEVICES. *Datasheet ADA4930*. Techn. Ber. 2010 [siehe S. 14].
- [4] Bundesverband Geothermie E.V. *Stand des Europäischen Hot Dry Rock-Forschungsprojektes Soultz-Sous-Forêts*. Hrsg. von Erwin KNAPEK. 5. Juli 2017. URL: <http://www.geothermie.de/wissenswelt/archiv/deutsch/stand-des-europaeischen-hot-dry-rock-forschungsprojektes-soultz-sous-forets.html> [siehe S. 2].
- [5] Hans-Martin HENNING und Andreas PALZER. „A comprehensive model for the German electricity and heat sector in a future energy system with a dominant contribution from renewable energy technologies“. In: *Renewable and Sustainable Energy Reviews*. 23. Juli 2013 [siehe S. 1].
- [6] Josef HOFFMANN. *MATLAB und SIMULINK in Signalverarbeitung und Kommunikationstechnik : beispielorientierte Einführung; Simulation praxisnaher Prozesse und Systeme*. Scientific computing. : DM 69.90, sfr 63.00, S 510.00. München [u.a.]: Addison-Wesley-Longman, 1999. ISBN: 3-8273-1454-2. URL: <http://swbplus.bsz-bw.de/bsz080092195cov.htm>; <http://swbplus.bsz-bw.de/bsz080092195v1g.htm> [siehe S. 19].

Literatur

- [7] Jörg ISELE u. a. „Kamerainspektionssystem für tiefe Geothermie (GeoKam)“. Abschlussbericht. Institut für Angewandte Informatik - Karlsruher Institut für Technologie, 6. Juni 2016 [siehe S. 1, 3, 4].
- [8] Friedrich JONDRALE. *Nachrichtensysteme : Grundlagen, Verfahren, Anwendungen; mit ... 10 Tabellen*. 4., überarb. und erweiterte Aufl. Erscheint: März 2011. Wilburgstetten: Schlembach, 2011. ISBN: 978-3-935340-68-7. URL: <http://www.schlembach-verlag.de/buecher.php?bnr=68;http://d-nb.info/101057227X/04> [siehe S. 6, 19].
- [9] Karl-Dirk KAMMEYER. *Nachrichtenübertragung : mit 38 Tabellen*. Hrsg. von Martin [Hrsg.] BOSSERT. 5., durchges. und erg. Aufl. Studium. Literaturverz. S. [815] - 836. - Mit 475 Abb. und 38 Tab. Wiesbaden: Vieweg + Teubner, 2011. ISBN: 978-3-8348-0896-7. URL: <http://ws.gbv.de/covers/isbn/9/97/9783834808967.jpg;http://swbplus.bsz-bw.de/bsz347355870inh.htm> [siehe S. 6, 8, 10, 16, 17].
- [10] Karl-Dirk KAMMEYER und Kristian KROSCHEL. *Digitale Signalverarbeitung : Filterung und Spektralanalyse; mit MATLAB-Übungen; mit 33 Tabellen*. 7., erw. u. korr. Aufl. Studium. Wiesbaden: Vieweg + Teubner, 2009. ISBN: 978-3-8348-0610-9. URL: http://deposit.d-nb.de/cgi-bin/dokserv?id=3250168&prov=M&dok_var=1&dok_ext=htm;http://swbplus.bsz-bw.de/bsz306173670inh.htm [siehe S. 19].
- [11] Michael KELLER. *Digitale Signalverarbeitung -Kurzschrift*. 19. Apr. 2017 [siehe S. 23].
- [12] Fan LI, Xinying LI und Jianjun YU. „Performance Comparison of DFT-Spread and Pre-Equalization for 8×244.2 -Gb/s PDM-16QAM-OFDM“. In: 1. Jan. 2015. URL: <http://ieeexplore.ieee.org/document/6923980/> [siehe S. 17].
- [13] Alfred MERTINS. *Signaltheorie : Grundlagen der Signalbeschreibung, Filterbänke, Wavelets, Zeit-Frequenz-Analyse, Parameter- und Signalschätzung*. Wiesbaden, 2013. URL: <http://swbplus.bsz-bw.de/bsz377586676cov.htmhttp://dx.doi.org/10.1007/978-3-8348-8109-0> [siehe S. 41].
- [14] Fabian MICHAELSEN. „Qualitätsmessungen und Störquellenbetrachtungen der OFDM-Übertragung einer Bohrlochsonde und Optimierung der Powerline-Versorgung“. Praxisbericht drittes und viertes Semester. 5. Sep. 2016 [siehe S. 13].
- [15] Gerald OBERSCHMIDT. *Grundlagen der Übertragungstechnik Vorlesung an der Dualen Hochschule Baden-Württemberg, Karlsruhe*. 2017 [siehe S. 7, 14, 20].

Literatur

- [16] Branislav M. POPOVID. „Generalized Chirp-Like Polyphase Sequences with Optimum Correlation Properties“. In: *IEEE TRANSACTIONS ON INFORMATION THEORY, VOL. 38*. 4. Juni 1992 [siehe S. 20].
- [17] John G. PROAKIS. *Digital communications*. 4. ed. McGraw-Hill series in electrical and computer engineering. Boston [u.a.]: McGraw-Hill, 2001. ISBN: 0-07-232111-3; 0-07-118183-0. URL: <http://www.gbv.de/dms/ilmenau/toc/553516566.PDF> [siehe S. 6, 16, 18].
- [18] Dietmar RUDOLPH. *Skript: Funk-Kanal*. 2017 [siehe S. 24].
- [19] Benjamin SCHUBERT u. a. *Geokam Abschlussbericht*. Techn. Ber. Heinrich-Hertz-Institut, 30. Aug. 2015 [siehe S. 2, 4, 6, 13, 14, 28].
- [20] Luigi SPATAFORA, Jörg ISELE und Benedict HOLBEIN. „The GeoKam – A Tool for Video Inspections in Hot Deep Geothermal Boreholes“. In: 23. Feb. 2016 [siehe S. 2].
- [21] Sumit A. TALWALKAR und S. Lawrence MARPLE. „TIME-FREQUENCY SCALING PROPERTY OF DISCRETE FOURIER TRANSFORM (DFT)“. In: *ICASSP 2010*. Georgia Institute of Technology. 2010. ISBN: 978-1-4244-4296-6 [siehe S. 42].
- [22] Brady VOLPE. „DOCSIS Pre-Equalization: Vastly Powerful, Often Undervalued“. In: URL: <http://www.zcorum.com/wp-content/uploads/Pre-Equalization-Final.pdf> [siehe S. 17].
- [23] BDEW Bundesverband der Energie- und WASSERWIRTSCHAFT E.V. *Energiemix in Deutschland*. Hrsg. von Mathias BUCKSTEEG. 5. Juli 2017. URL: <https://www.bdew.de/internet.nsf/id/energiemix-de> [siehe S. 1].
- [24] Friedrich WIEGEL. „Entwurf und Implementierung eines Phasenmodulationsverfahrens mit einem FPGA zur Realisierung der drahtgebundenen Kommunikation einer Bohrlochsonde“. Bachelorthesis. Dualen Hochschule Baden-Württemberg Karlsruhe, 2013 [siehe S. 3].

9 Anhang

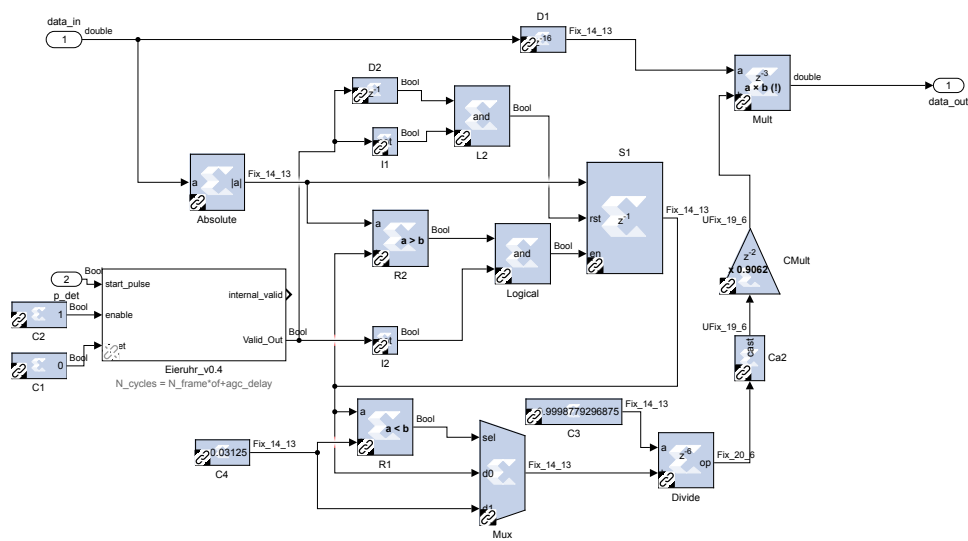


Abbildung 9.1: Blockschaltbild der AGC ChDn aus Simulink™

9 Anhang

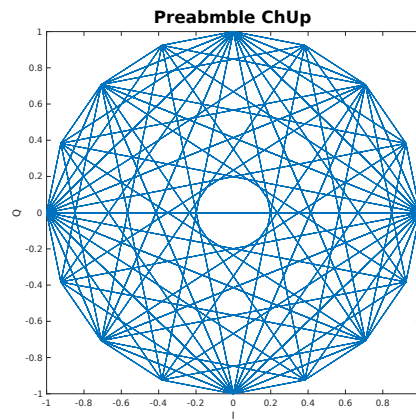


Abbildung 9.2: Präambel des ChUp in IQ-Ebene

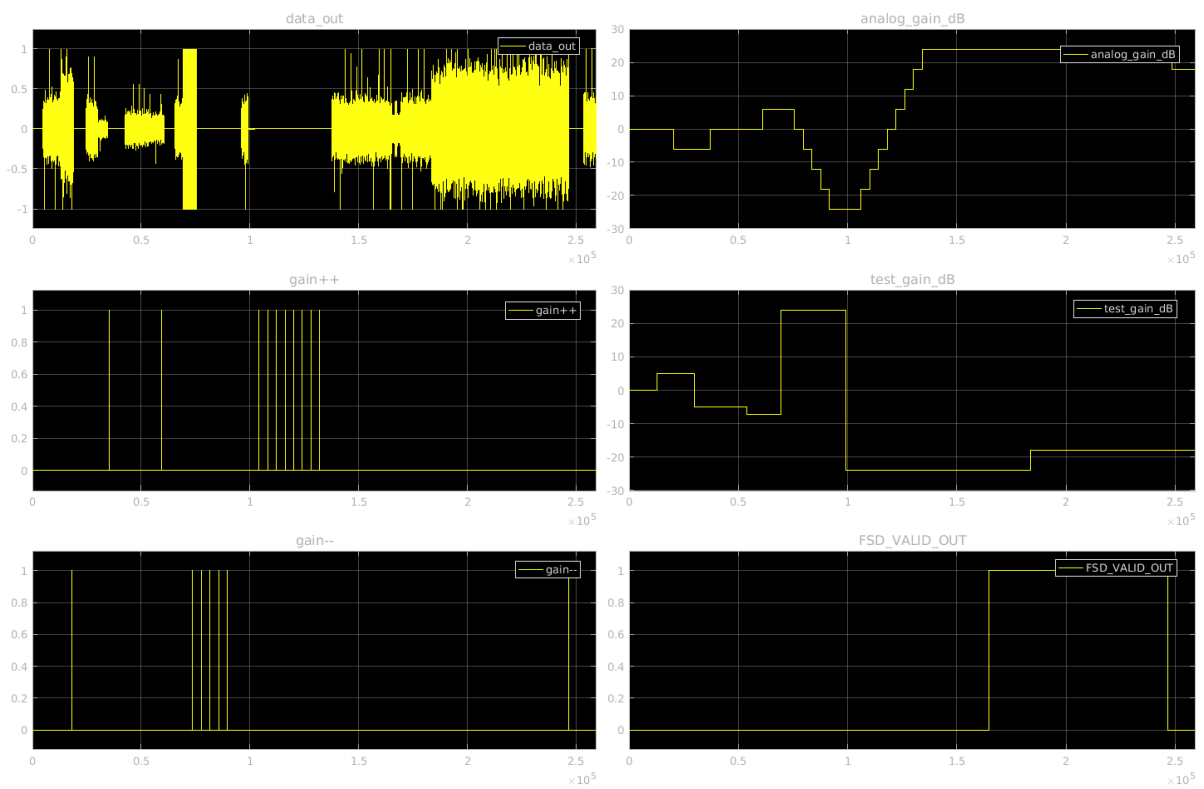


Abbildung 9.3: Simulationsergebnis der AGC des ChUp

9 Anhang

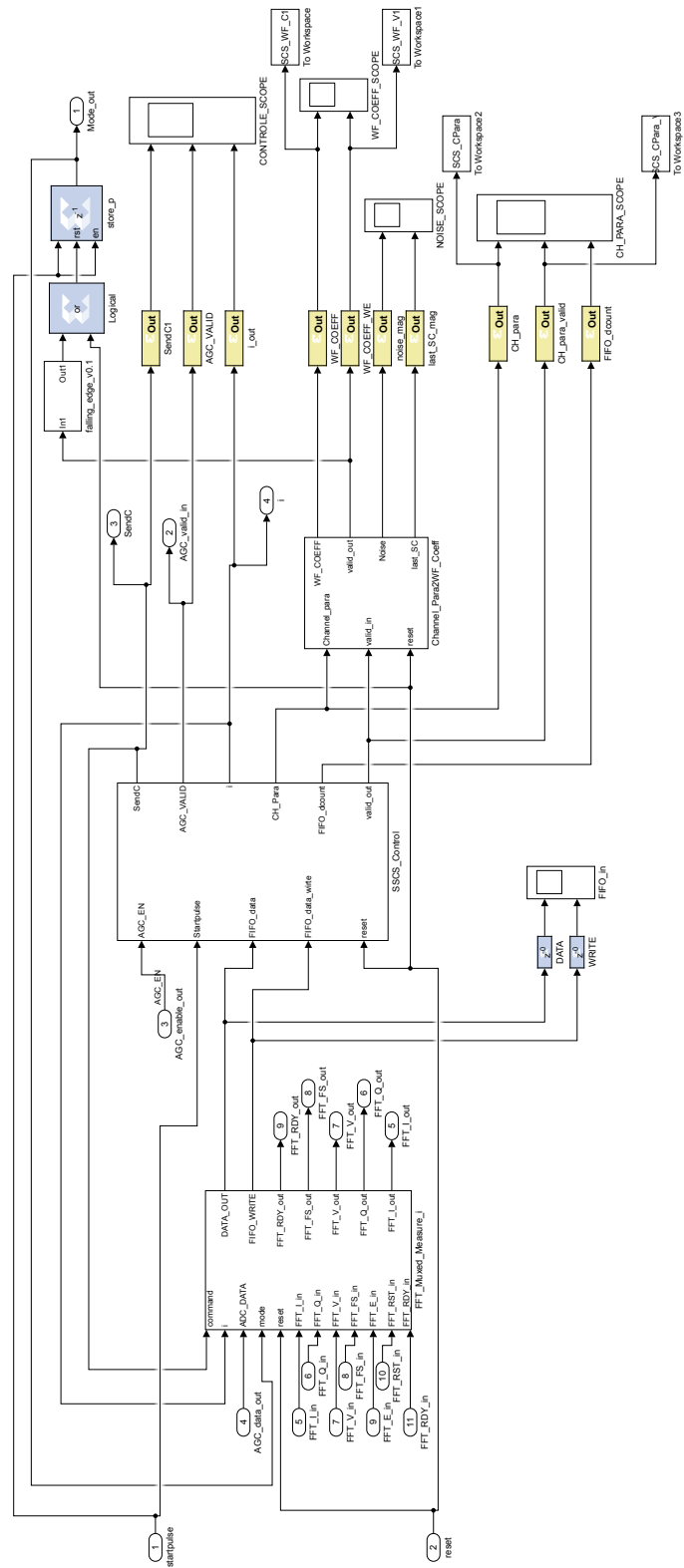


Abbildung 9.4: Blockschaltbild des SSCS-FFT-Blocks im Rx aus Simulink™

9 Anhang

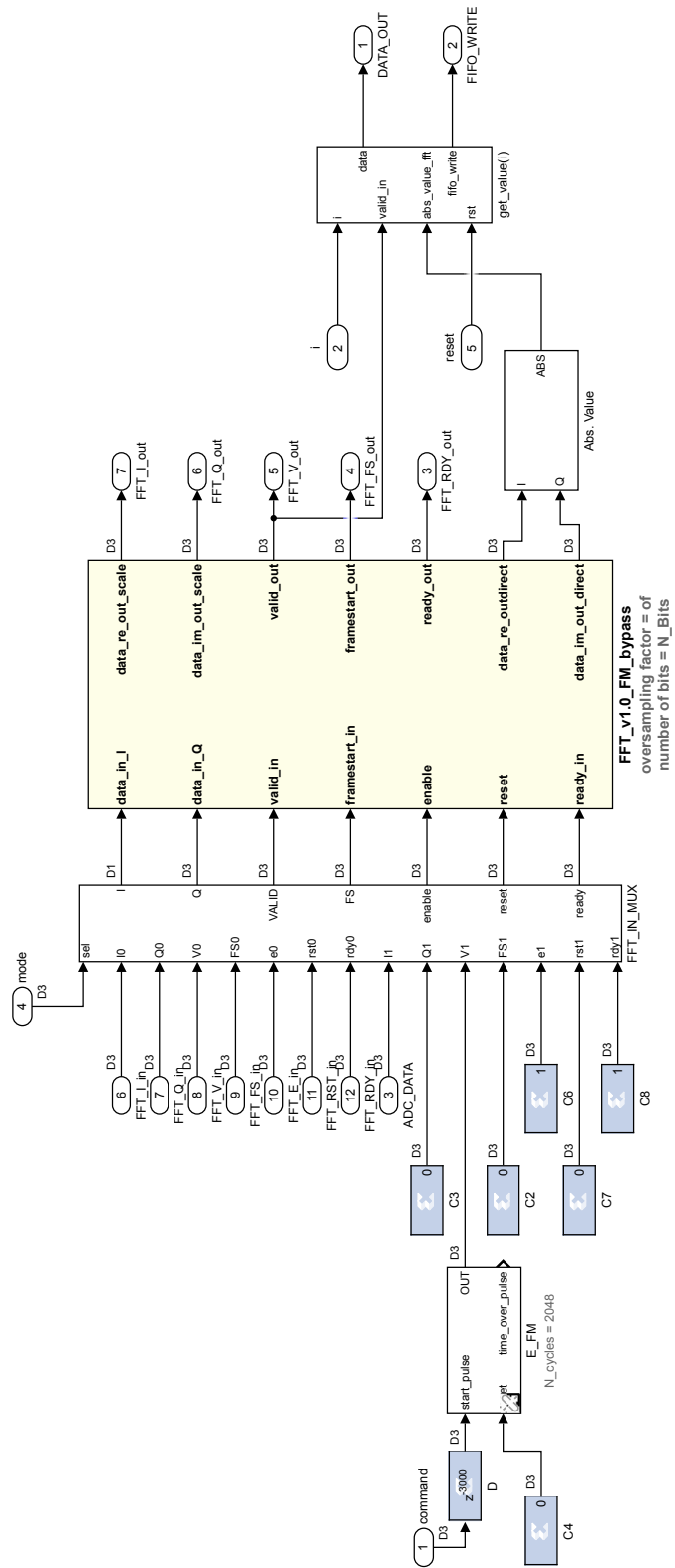


Abbildung 9.5: Blockschaltbild des FFT-Blocks aus Simulink™

9 Anhang

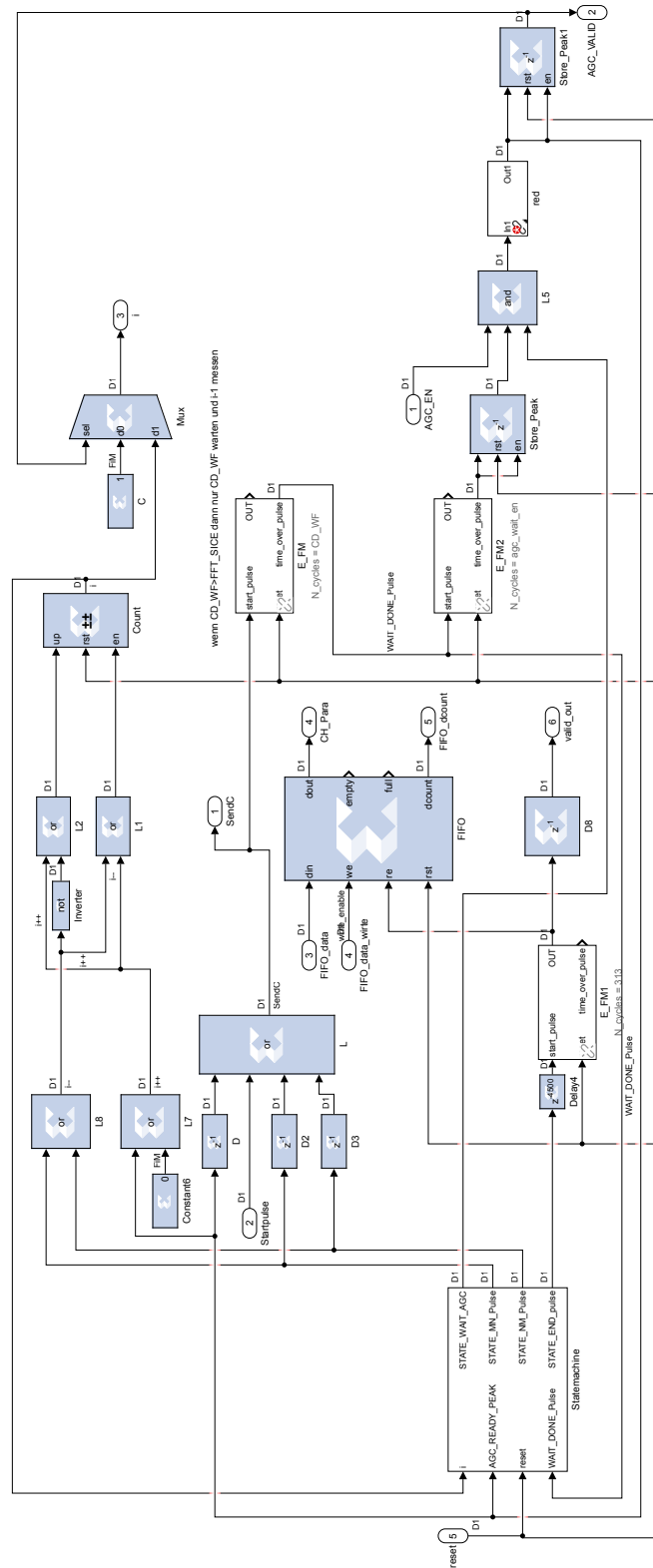


Abbildung 9.6: Blockschaltbild des Control-Blocks aus Simulink™

9 Anhang

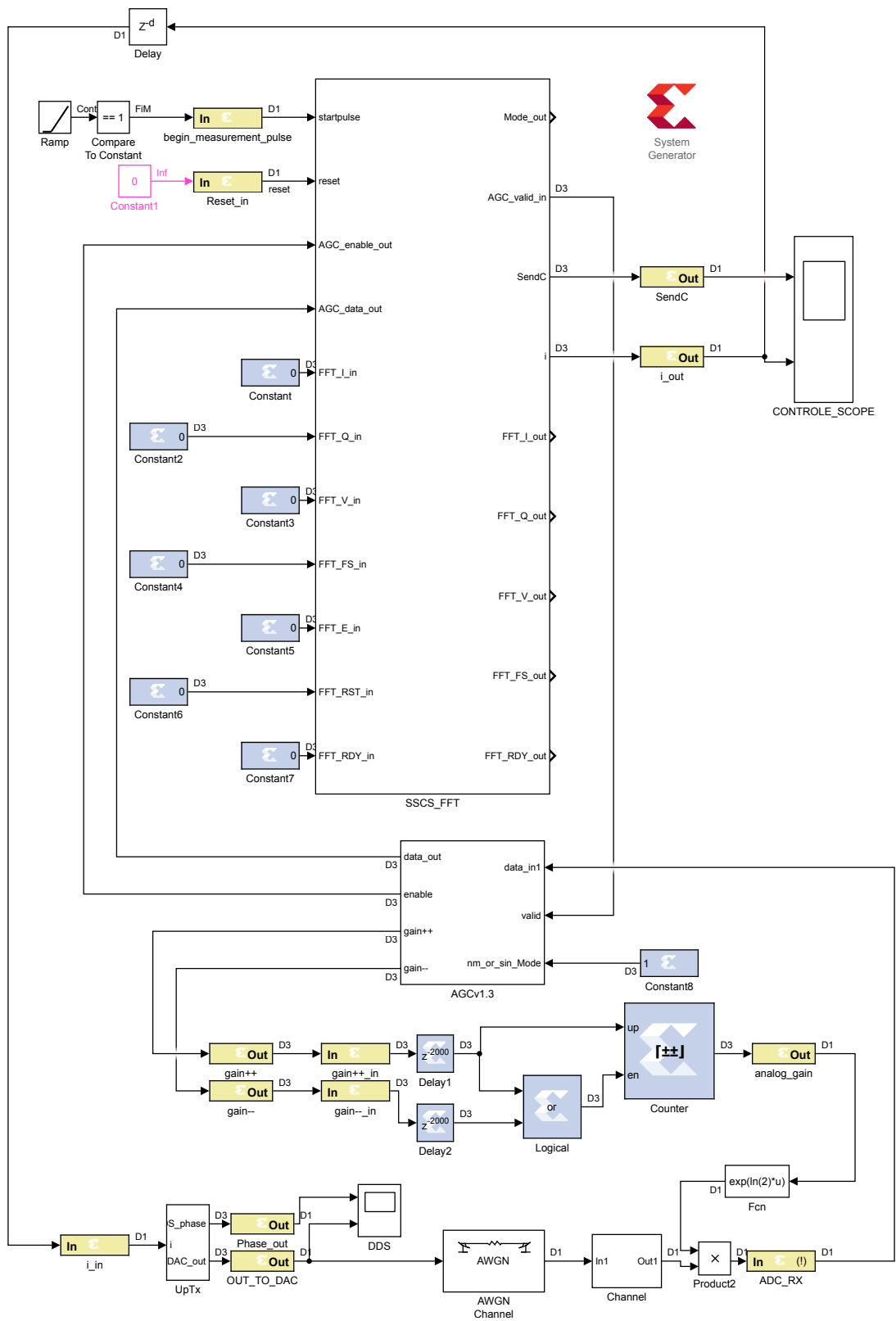


Abbildung 9.7: Blockschaltbild der SSCS-Verfahrensgesamtsimulation aus Simulink™

9 Anhang

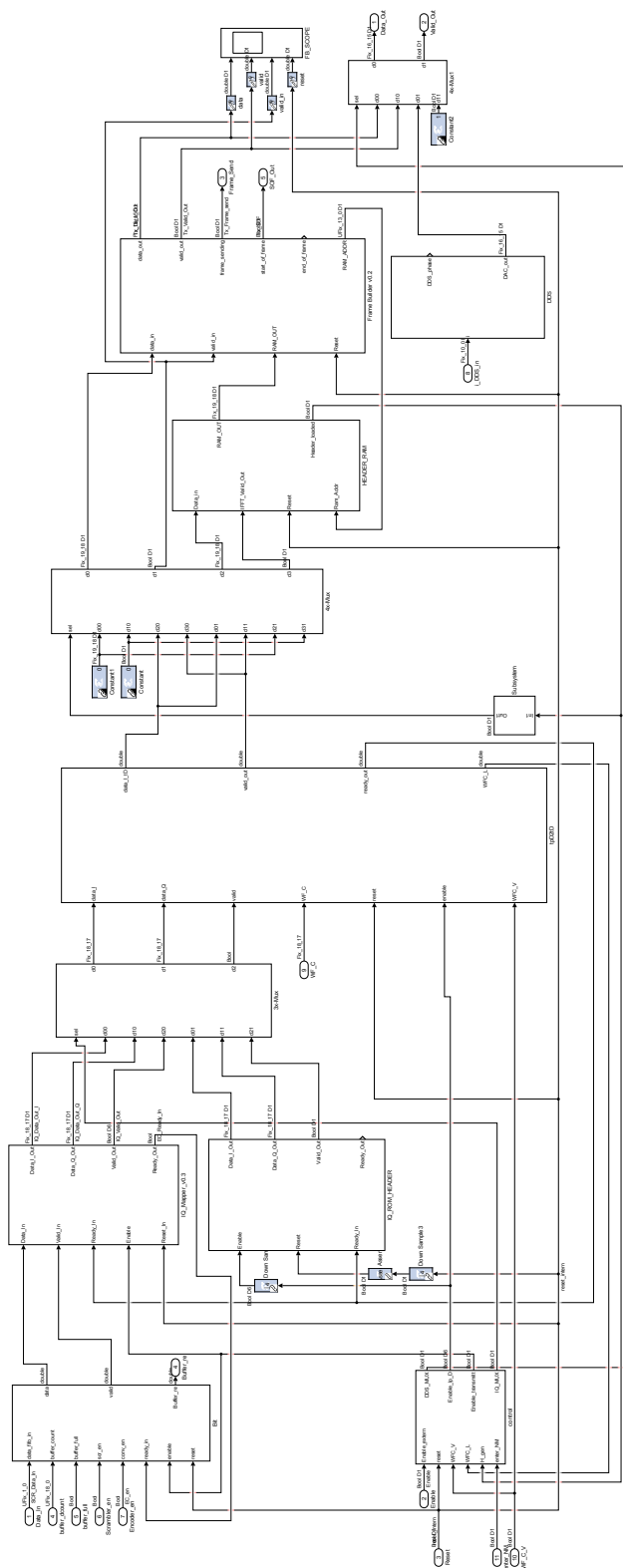


Abbildung 9.8: Blockschaltbild des Senders aus Simulink™

9 Anhang

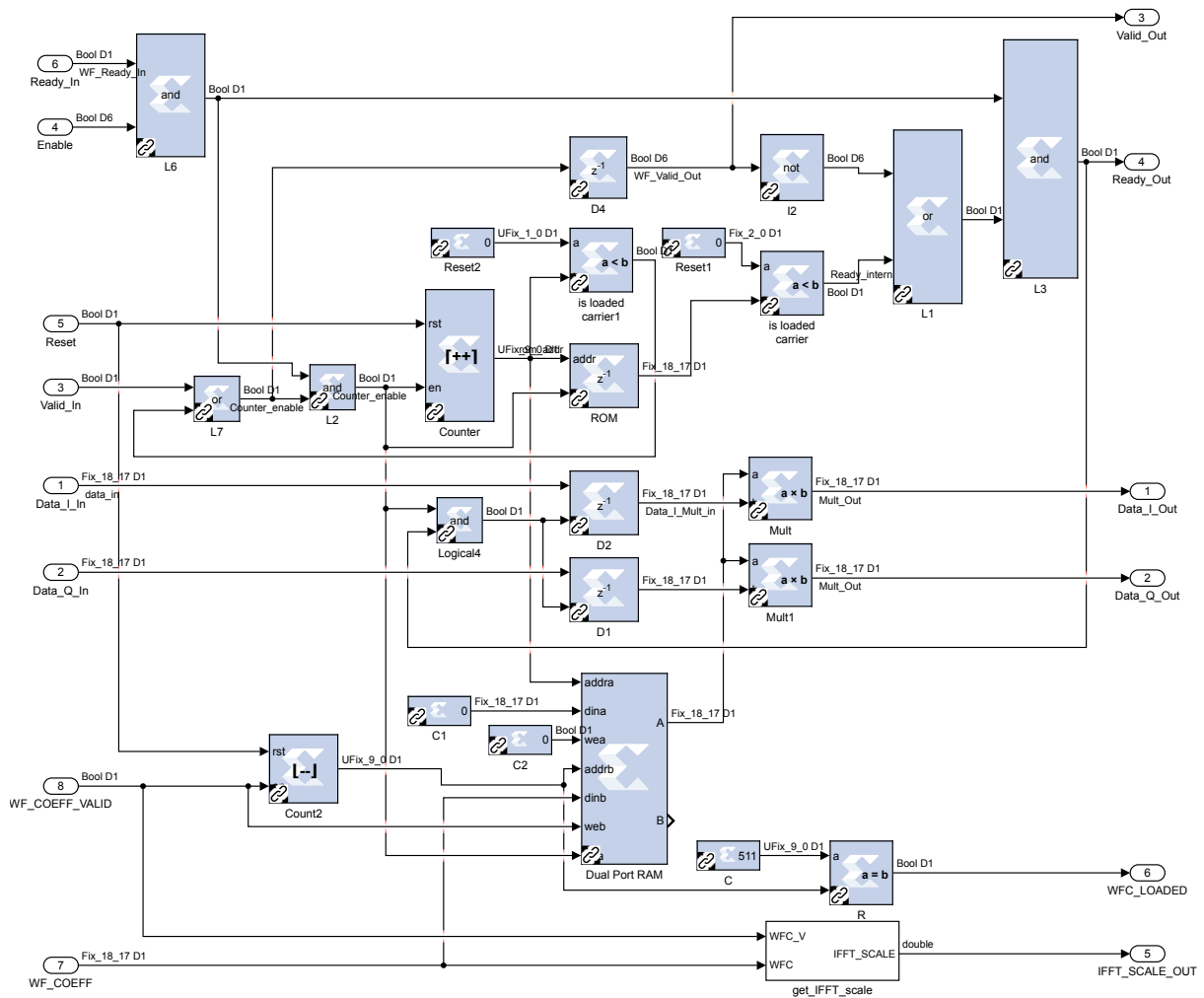


Abbildung 9.9: Blockschaltbild des Waterfillingmoduls aus Simulink™

9 Anhang

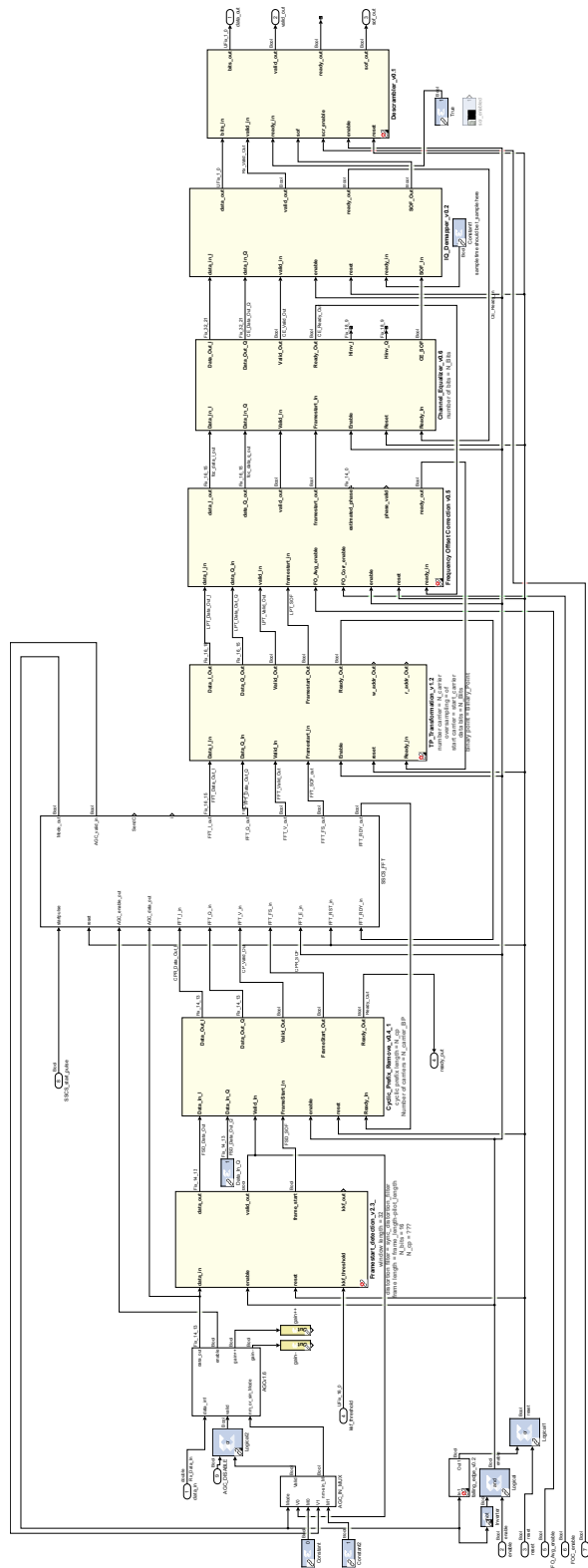


Abbildung 9.10: Blockschaltbild des gesamten Rx aus Simulink™