



OFDM System

RTL Implementierung des Rx Pfades

Quantisieren der Matlab Kette

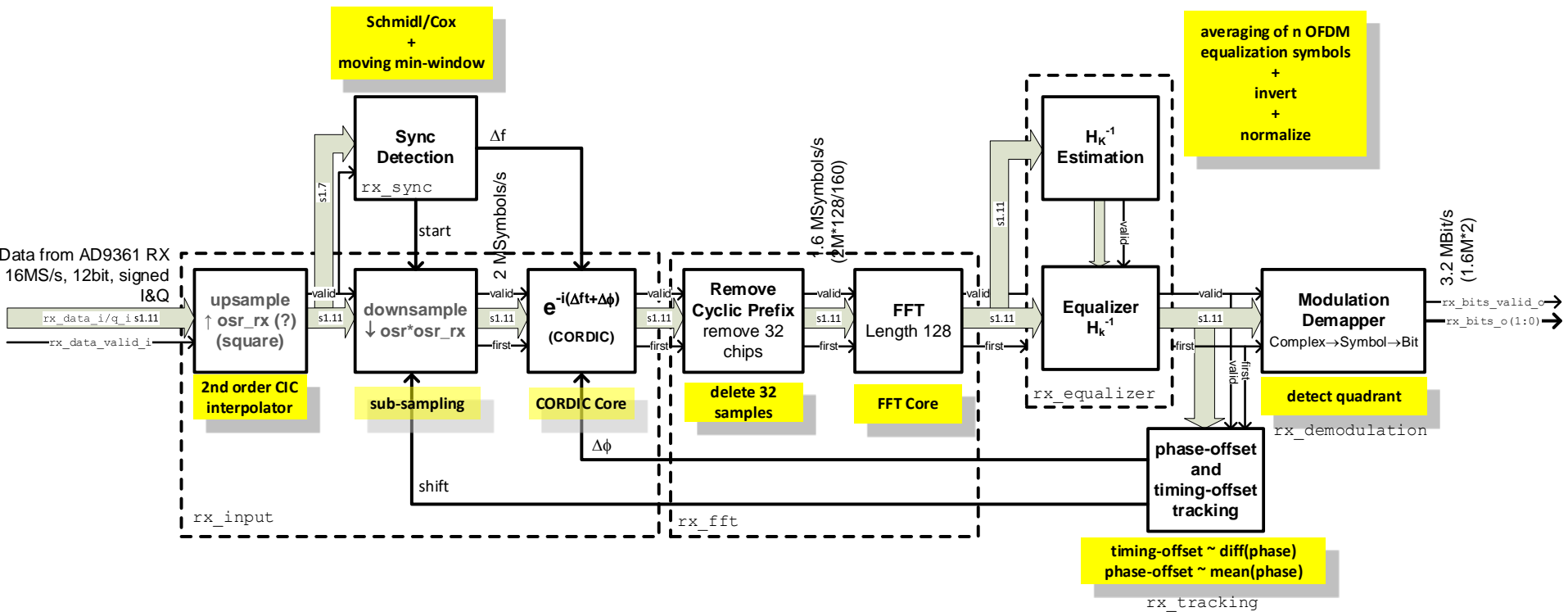


- Eingangsdaten: 12bit signed
 - interpretiert als s1.11 (Wertebereich: $(2^{11}-1 \dots -2^{11})/2^{11} = +1-2^{-11} \dots -1$)
 - Verwendung von quantisierten floating point Zahlen
 - z.B.: s1.11 Quantisierung: $x_{\text{quant}} = [\text{round} | \text{floor} | \text{ceil}] (x * 2^{11}) / 2^{11}$
- Interne Quantisierung soweit reduzieren bis sich Auswirkung im EVM zeigt
 - Quantisierung zwischen den Blöcken
 - Interne Quantisierung
 - Interne Quantisierung der FFT ist ohne einem quantisierten Modell der FFT schwer nachzubilden
 - Reduktion der Bitbreite ist besonders bei rechenintensiven Blöcken wichtig
 - Auswahl von Algorithmen geringer Komplexität
 - z.B.: CIC für Interpolation der RX-Filterdaten
 - Interpolationsrate minimieren (welche Überabtastung wird für die zeitliche Synchronisation benötigt)
 - Modellierung dieser Algorithmen in Matlab
 - Komplexe Algorithmen auf der niedrigstmöglichen Abtastrate implementieren (FFT, CORDIC)

Aufgabenteilung der RX Kette

- Top-Level Integration
 - Testbench: File I/O und Instantiierung des Top-Level der RX Kette
 - Top-Level: Struktur und Verdrahtung
- Synchronisation
 - Timing- und Frequenzsynchronisation nach Schmidl/Cox
- RX Upsampling, Downsampling, Frequenz- und Phasensynchronisation mit CORDIC
 - Upsampling mit CIC zweiter Ordnung
 - Downsampling: Start gesteuert durch Synchronisationsblock und Timingtracking
 - Anzeigen des ersten Samples jedes OFDM Symbols
 - CORDIC korrigiert Frequenzoffset und Phasenoffset
- Remove of Cyclic Prefix und FFT
 - Auswahl einer FFT
 - Struktur um Datendurchsatz sicherzustellen
- Equalizer
 - Summation der Equalisierungssymbole, berechnen von H_K^{-1}
 - Korrektur der Empfangen Daten nach der Ermittlung von H_K^{-1}
- Phase und Timing Tracking + Modulation Demapper
 - Messen des Phasenverlaufes von Datensymbolen
 - Ermitteln von Zeitversatz und Phasenversatz
 - Rückmeldung an Frequenzkorrektur-CORDIC und Downsampler

Blockdiagramm Rx Kette



Top-Level Testbench

- Name: `rx_top_tb-e/a/c.vhd`
- Systemtakt 80MHz
- Reset active-low
- Einlesen der RX Daten (2x12bit @ 16MS/s) aus Datei
- Ausgabe der RX-Bits in Datei
- Instantiierung der Top-Level Hierarchie
- Dateiname der Eingangsdaten: `rx_data.txt`
 - ein IQ Werte paar pro Zeile (integer, +2047...-2048, s1.11)
- Dateiname der Ausgangsdaten: `rx_bits.txt`
 - zwei Bits pro Zeile in der Reihenfolge der Subträger

RX Top-Level

- **Name:** rx_top-e/a/c.vhd
- **Instantiiert die Blöcke der RX Kette**
 - rx_sync
 - rx_input
 - rx_fft
 - rx_equalizer
 - rx_tracking
 - rx_demodulation

| Portname | Mode | Breite | Beschreibung |
|-----------------|------|--------|--|
| sys_clk_i | In | 1 | Systemtakt, 80MHz |
| sys_rstn_i | In | 1 | Systemreset, low-aktiv |
| rx_data_i/q_i | in | 12 | Rx Daten von AD9361 (I/Q Paar) |
| rx_data_valid_i | in | 1 | Valid Signal der Eingangsdaten, aktiv für einen Taktzyklus pro Datum |
| rx_bits_o | out | 2 | Rx Bits eines Subträgers |
| rx_bits_valid_o | out | 1 | Valid Signal der Ausgangsdaten, aktiv für einen Taktzyklus pro Datum |

VHDL Verhaltensmodell



- Schnittstellen und Datentypen wie im RTL Modell
- Struktur wie auf RTL
- Subblöcke auf Verhaltensebene modellieren
 - zeitliches Verhalten mit Clock modelliert
 - z.B.: $F_{clk}=80\text{MHz}$ → neues Eingangssample jeden fünften Takt
 - Unterschied zu Matlabmodell: sequentielle Verarbeitung der Daten
 - rekursive Implementierung der Algorithmen

Ressourcen

- Intel (Altera) IP Cores:
 - <https://www.intel.com/content/www/us/en/products/details/fpga/intellectual-property/dsp.html>
 - FFT:
 - <https://www.intel.com/content/www/us/en/products/details/fpga/intellectual-property/dsp/fft.html>
 - CIC:
 - <https://www.intel.com/content/www/us/en/products/details/fpga/intellectual-property/dsp/cic-compiler.html>
 - https://elearning.fh-ooe.at/pluginfile.php/489300/mod_resource/content/2/Hogenauer1981-AnEconomicalClassofDigitalFiltersforDecimationandInterpolation.pdf
 - CORDIC
 - https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/ug/ug_cordic.pdf

Aufgabenstellung – Teil 1



- Quantisieren Sie den von Ihnen zu bearbeitenden Teil der RX Kette
- Modellieren Sie diesen Teil der RX Kette in Matlab so, dass er die HW Implementierung abbildet
- Geben Sie ein Blockschaltbild der HW-Implementierung an
 - Schätzen Sie den HW Aufwand für den Block ab
- Erstellen Sie ein VHDL Verhaltensmodell, dass den Datenfluss auf RTL Ebene nachbildet (cycle-true)