On the Performance of SC-MMSE-FD Equalization for Fixed-Point Implementations

Michael Schwall
Karlsruhe Institute of Technology
Karlsruhe, Germany
Email: michael.schwall@kit.edu

Tamal Bose
University of Arizona
Tucson, USA
Email: tbose@arizona.edu

Friedrich K. Jondral
Karlsruhe Institute of Technology
Karlsruhe, Germany
Email: friedrich.jondral@kit.edu

***Abstract*—A fixed-point implementation of a minimum mean square error (MMSE) based frequency domain (FD) equalizer with soft interference cancellation (SC) is studied. The equalizer additionally processes *a priori* information about the transmitted symbols and is used for turbo equalization. In this paper, we analyze the quantization and the clipping for different fixed-point representations and modulation schemes. The analysis allows to derive efficient representations for all symbols within the equalizer. This procedure is demonstrated for a generic system configuration featuring a 16-QAM. Finally, a fixed-point implementation in an integrated design environment for FPGAs verifies the theoretical studies and shows the device utilizations for different FPGAs that are embedded in current software defined radios. The results show, that on average 10 bits per symbol are required for a near-optimum equalization performance utilizing less than 8% area of state of the art FPGAs.**

I. Introduction

Inter-symbol interference (ISI) is one of the major issues when mobile radios communicate in harsh transmission environments e.g. urban or mountainous areas. The interference even grows as the data rate of the communication link increases i.e. broadband links are established. In times of high definition video streaming and other high data rate applications this challenges the equalization task at the receiver's physical layer. Although multi-carrier techniques can bypass the ISI issue and enable simple equalizer structures, single-carrier is still preferred for power-limited and small radios.

In this paper, we consider a single-carrier transmission and an iterative equalization structure: a soft input soft output (SISO) equalizer incorporates *a priori* information about the transmitted symbols that is provided by a SISO decoder. The structure is also referred to as turbo equalizer [1]. We focus on the first SISO component and a linear minimum mean square error (MMSE) equalization [2] that is carried out in the frequency domain (FD) [3]. The *a priori* information is used in a soft interference cancellation (SC) process [4].

Back to the radios, fast digital signal processing units, like application specific integrated circuits (ASIC) or field programmable gate arrays (FPGA), are mainly restricted to fixed-point arithmetics. That means, that signals are processed with a limited dynamic range and quantization or clipping occurs [5]. Therefore, we will study the SC-MMSE-FD equalizer in terms of fixed-point related issues and provide an efficient implementation for a generic system configuration. Due to its integration into a turbo equalizer, the SC-MMSE-FD equalizer's contribution to the overall convergence behavior is analyzed in particular with the help of extrinsic information transfer (EXIT) charts. The results of the fixed-point study are also applicable for memory-reduced implementations or single instruction multiple data (SIMD) programming on general purpose processors (GPP). To the authors' best knowledge, there are no other publications that study the above mentioned fixed-point issues in detail. Nevertheless, implementations for a fixed-point DSP [6] or ASIC [7] have been presented before. These implementations either exploit the entire word length of the device or determine the fixed-point representation using numerical simulations for a specific system configuration.

This paper is organized as follows: section II introduces the system model and the SC-MMSE-FD equalizer. The fixed-point study in section III focuses on the occurring quantization and clipping. Based on these studies, an efficient fixed-point implementation for a generic system configuration is exemplarily derived in section IV. The implementation in an integrated design environment for FPGAs in section V verifies the theoretical results of the conducted studies. Furthermore, device utilizations are presented for different FPGAs that are embedded in current software defined radios. Section VI concludes and highlights the major outcomes of the paper.

II. System Model

A. Received Symbols

We focus on a digital single-carrier block transmission over a frequency selective channel. The coherence time of the channel is assumed to be larger than the block duration so that blocks can be treated independently. The transmitter uses a bit-interleaved coded modulation with a linear M-ary modulation scheme. Due to the frequency domain equalization in the receiver, each block of transmitted, complex-valued symbols $\mathbf{s} = [s_0, \ldots, s_{N_s-1}]^T$ of length N_s is provided with a cyclic prefix (CP). The length of the CP is chosen to avoid inter-block interference. The transmitted symbol power is normalized so that $\mathbb{E}\{|s_k|^2\} = 1$. The impulse response of the channel is given by $\mathbf{h} = [h_0, \ldots, h_{N_h-1}]^T$ and known at the receiver. Assuming perfect synchronization, the received symbols after removing the CP are given by

$$\mathbf{r} = \mathbf{H}\mathbf{s} + \mathbf{n}, \tag{1}$$

where $\mathbf{H} \in \mathbb{C}^{N_s \times N_s}$ is the circulant channel matrix [8] and $\mathbf{n} \in \mathbb{C}^{N_s}$ is the additive noise whose elements n_k are $\mathcal{CN}(0, \sigma_0^2)$ distributed. The noise variance σ_0^2 is known at the receiver and the signal-to-noise ratio (SNR) is defined as $\mathrm{SNR} = -10\log(\sigma_0^2\ \log_2(M))$ dB.

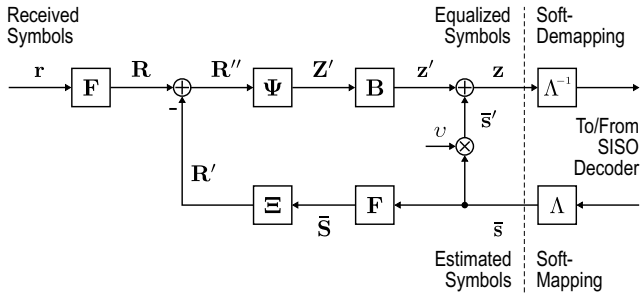

Fig. 1. Block diagram of the SC-MMSE-FD equalizer

B. SC-MMSE-FD Equalizer

The concept of soft interference cancellation in context of MMSE estimation was initially introduced in [4] to mitigate the multiple access interference in CDMA systems. It was later extended for turbo equalization by Tüchler *et al.* [2]. Kansanen and Matsumoto further applied frequency domain equalization to reduce the overall complexity [3]. The main concept of the SC-MMSE-FD equalizer is to remove the remaining ISI after an MMSE based equalization with the help of *a priori* known symbol estimates. The estimates are provided by a SISO decoder in terms of log-likelihood ratios and (softly) mapped to symbols (Λ).

The block diagram of the SC-MMSE-FD equalizer is depicted in figure 1. Given the received symbols (1), the estimated symbols $\bar{\mathbf{s}}$ and full channel state information, an equalized symbol block $\mathbf{z} = [z_0, \ldots, z_{N_s-1}]^T$ is calculated by

$$\mathbf{z} = \upsilon\bar{\mathbf{s}} + \mathbf{B}\boldsymbol{\Psi}(\mathbf{F}\mathbf{r} - \boldsymbol{\Xi}\mathbf{F}\bar{\mathbf{s}}), \tag{2}$$

where the following substitutions have been made

$$\begin{aligned} \mathbf{F}(l,j) &= e^{-i\frac{2\pi}{N_s}lj}, \quad 0 \le l,j \le N_s-1,\ i=\sqrt{-1} \\ \mathbf{B} &= N_s^{-1}\,\mathbf{F}^H \\ \boldsymbol{\Xi} &= \mathbf{F}\mathbf{H}\mathbf{B} \\ \varphi &= N_s^{-1}\,\bar{\mathbf{s}}^H\bar{\mathbf{s}} \\ \boldsymbol{\Psi} &= \boldsymbol{\Xi}\left((1-\varphi)\,\boldsymbol{\Xi}\boldsymbol{\Xi}^H + \sigma_0^2\mathbf{I}\right)^{-1} \\ \upsilon &= N_s^{-1}\,\mathrm{trace}(\boldsymbol{\Psi}\boldsymbol{\Xi}). \end{aligned} \tag{3}$$

Since the MMSE equalization matrix $\boldsymbol{\Psi}$ and the channel transfer matrix $\boldsymbol{\Xi}$ are diagonal, the SC-MMSE-FD equalizer can be efficiently implemented using fast Fourier transforms (FFTs) and element-wise mathematical operations.

The equalized symbols z_k can be approximated using an equivalent AWGN channel model [3]

$$z_k \approx \upsilon s_k + w_k. \tag{4}$$

The distribution of the noise w_k is $\mathcal{CN}(0, \upsilon^2(\varphi-1)+\upsilon)$. The model and its characteristics are used to perform the (softly) demapping (Λ^{-1}) from symbols to log-likelihood ratios, which are forwarded to the SISO decoder again. Moreover, the model enables a complexity-reduced analysis of the equalization performance since it is independent of the received signal. The model will thus be used for the fixed-point study in section III to analyze the SC-MMSE-FD equalizer's contribution to the overall convergence behavior.

TABLE I. FIXED-POINT REPRESENTATION: NOTATION, VALUE RANGE AND RESOLUTION

Type	Notation	Max. value	Min. value	Resolution
unsigned	`uFix_N_n`	$2^{N-n} - 2^{-n}$	0	2^{-n}
signed	`Fix_N_n`	$2^{N-n-1} - 2^{-n}$	-2^{N-n-1}	2^{-n}

III. FIXED-POINT STUDY

This section describes the preliminary considerations for a fixed-point implementation of an SC-MMSE-FD equalizer. The implementation requires a fixed-point representation for each processed *symbol* within the equalizer. Hence, symbol values become quantized and magnitude-limited. Furthermore, we focus on the symbols that are resource demanding i.e. have a high contribution to the overall logic area or memory consumption. Parameters or scalars will be represented by generic fixed-point representation, e.g. `Fix_32_16`. To study the effects of the implementation, we assume, that the algorithm for the (inverse) Fourier transformation is performed with negligible fixed-point related issues. That means, that the dynamic range of the representation is adapted after each butterfly operation. These assumptions allow a hardware and intellectual property (IP) core independent analysis. The consequences of scaling methods are discussed in section V.

We consider a fixed-point representation with a word length of N bits. A word is separated into an integer and a fractional part by a fixed *virtual* comma. The number of fractional bits is n, with $n \le N$. For the sake of simplicity, we define the number of integer bits (without the sign bit) as m. The word length is hence given by

$$N = m + n + 1. \tag{5}$$

The prevalent notation is `Fix_N_n` for a signed representation and `uFix_N_n` for an unsigned. The characteristics are summarized in table I. The dynamic range of a representation is defined as $\mathrm{DR} = 20\log\left(2^N\right)$ dB. All operations are performed using Two's complement [5]. ⊞ *Clarification:* The word length is the only parameter of interest that needs to be studied. The virtual comma merely represents a scaling to match the amplitude of the symbol values to the amplitude of the fixed-point representation. For example, all symbol values could be pre-scaled to integer values, so that the need for a fixed-point becomes obsolete. However, it is more intuitive (according to the authors' opinion) to study the dynamic range of the symbol values first and then define a fixed-point representation using the virtual comma ⊞. Therefore, we will start to study the effects of the quantization in section III-A to determine the required number of fractional bits n. Afterwards, investigations on clipping will allow to estimate the number of integer bits m in section III-B.

A. Quantization

To study the quantization effects caused by a fixed-point implementation, an analytic quantization model is set up first: a *real* number x is quantized to the value

$$[x]_q = x + q, \tag{6}$$

where the error q is an uniformly distributed random variable with zero-mean and a variance $\sigma^2 = \Delta^2/12$. We assume *rounding* since no other method of quantization yields lower

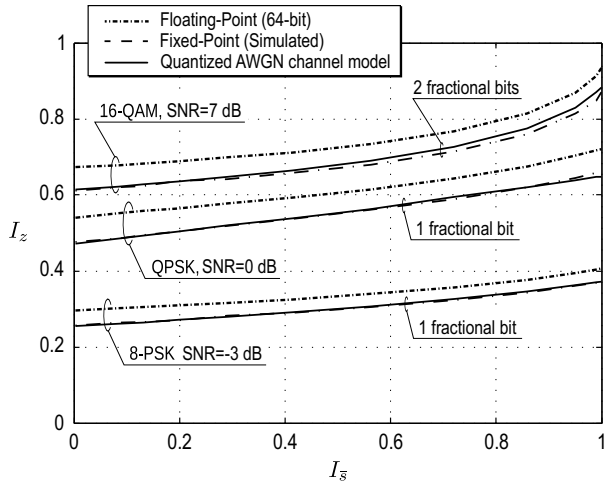

Fig. 2. EXIT chart for different system configurations and numbers of fractional bits ($N_s = 1024$; no clipping)

variance [5] i.e. σ^2 only depends on the resolution $\Delta = 2^{-n}$ (*cf.* table I). The quantization model is applied to all processed symbols within the SC-MMSE-FD equalizer. We will use a *quantized version* of the equivalent AWGN channel model in (4) to easier evaluate the effects of the quantization. In this case, the variance of the quantized symbol $[z_k]_q$ is given by

$$\begin{aligned}\mathrm{Var}\{[z_k]_q\} &= \mathrm{Var}\{z_k\} \\ &\quad + (\alpha+\upsilon^2)\sigma^2_{\bar{s}} + \alpha\sigma^2_{\bar{S}} + \sigma^2_{\bar{s}'} \\ &\quad + \beta(\sigma^2_r + \sigma^2_R + \sigma^2_{R'} + \sigma^2_{R''}) \\ &\quad + \sigma^2_{Z'} + \sigma^2_{z'} + \sigma^2_z,\end{aligned} \tag{7}$$

where $\alpha = N_s^{-1}\mathrm{trace}(\mathbf{\Psi\Xi\Xi}^H\mathbf{\Psi}^H)$, $\beta = N_s^{-1}\mathrm{trace}(\mathbf{\Psi\Psi}^H)$ and $\sigma^2_\square$ is the variance of the introduced quantization error of the symbol $\square$.

The accuracy of the quantized AWGN channel model is proven with an extrinsic information transfer (EXIT) chart [9] that illustrates the performance of the SC-MMSE-FD equalizer. Therefore, the mutual information I_z of the equalized symbol is calculated with respect to the mutual information $I_{\bar{s}}$ of the estimated symbols which are generated by a test setup. The EXIT chart in figure 2 exemplarily shows the progress for a simulated SC-MMSE-FD equalizer implemented with 64-bit floating point precision, a simulated fixed-point implementation with no clipping effects and the quantized AWGN channel model. It is obvious, that due to the fixed-point representation, the performance of the SC-MMSE-FD declines i.e. as a consequence, the number of iterations to reach a certain convergence behavior in a turbo equalizer is increased. But it can also be seen, that the quantized AWGN channel model allows a precise estimate of the simulated fixed-point progress, independent of the modulation scheme or the noise variance. Therefore, it will be used to determine efficient fixed-point representations in section IV. The variance in (7) can also be used for Kansanen's analytical method for EXIT chart computation where the SNR of the equivalent channel is now given by $\mathcal{L} = \frac{\upsilon}{\mathrm{Var}\{[z_k]_q\}}$ (*cf.* [3], eq. 24).

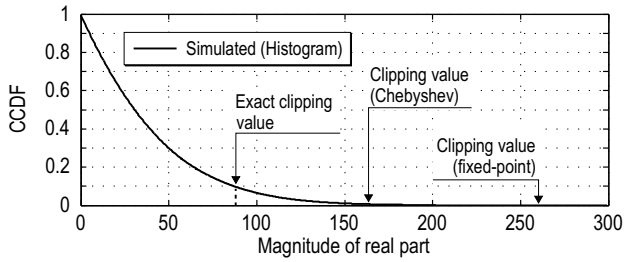

Fig. 3. CCDF of $|\mathrm{Re}\{Z'_k\}|$ for different clipping values ($P_{cl} = 0.1$; $N_s = 1024$; SNR=10 dB; $\varphi = 0.9$)

B. Clipping

Clipping occurs if the number of integer bits is not sufficient to represent all amplitudes of a real random variable x. Instead, amplitudes that exceed the range of the fixed-point representation are mapped to the maximum or minimum values (*cf.* table I). If x is continuously distributed, a probability for clipping $P_{cl} = P\{|x| \geq c\}$ needs to be additionally specified. Given the distribution characteristics of x, we use Chebyshev's inequality [10]

$$c^2 \leq \mathrm{E}\left\{x^2\right\}\left(P\{|x| \geq c\}\right)^{-1} \tag{8}$$

to determine the clipping value c, that will be exceeded with a probability equal or smaller than P_{cl}. Therefore, c is treated as an *upper* bound and used to calculate the required number of integer bits

$$m = \lceil \log_2(c) \rceil . \tag{9}$$

Compared to the exact clipping value, which would require full knowledge about the distribution of x, c overestimates the exact value and hence causes an approximation error. Additionally, c is set to 2^m, since the fixed-point representation leads to clipping values to be a power of two. Although other inequalities, like Chernoff's bound [10], are tighter and allow more precise estimates of c, the calculation of the expectation in (8) is quite convenient and therefore provides a complexity trade-off.

To apply the considerations from above to the symbols processed within the SC-MMSE-FD equalizer, we assume circular symmetric distributions so that only the real parts need to be analyzed in the following. It is also assumed, that the channel impulse response $\mathbf{h}$ is normalized, so that $\mathbb{E}\{|h_l|^2\} = 1$ and the distribution of each coefficient is $\mathcal{CN}(0,1)$. Using the system model, the expectations for each symbol in (8) are approximated by

$$\begin{aligned}
\mathrm{E}\left\{(\mathrm{Re}\{\bar{S}_k\})^2\right\} &= N_s\varphi/2 \\
\mathrm{E}\left\{(\mathrm{Re}\{\bar{s}'_k\})^2\right\} &= \upsilon^2\varphi/2 \\
\mathrm{E}\left\{(\mathrm{Re}\{r_k\})^2\right\} &= \left(1+\sigma_0^2\right)/2 \\
\mathrm{E}\left\{(\mathrm{Re}\{R_k\})^2\right\} &= N_s\left(1+\sigma_0^2\right)/2 \\
\mathrm{E}\left\{(\mathrm{Re}\{R'_k\})^2\right\} &= N_s\varphi/2 \\
\mathrm{E}\left\{(\mathrm{Re}\{R''_k\})^2\right\} &= N_s\left(1+\sigma_0^2-\varphi\right)/2 \\
\mathrm{E}\left\{(\mathrm{Re}\{Z'_k\})^2\right\} &= N_s\upsilon/2 \\
\mathrm{E}\left\{(\mathrm{Re}\{z'_k\})^2\right\} &= \upsilon/2 \\
\mathrm{E}\left\{(\mathrm{Re}\{z_k\})^2\right\} &= \left(\upsilon^2\varphi+\upsilon\right)/2.
\end{aligned} \tag{10}$$

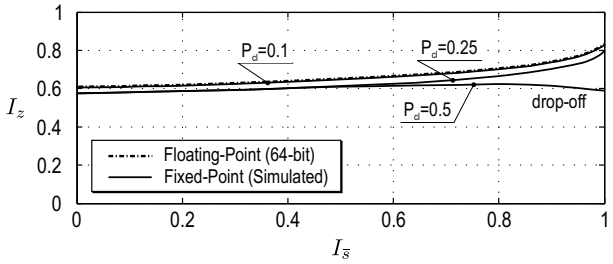

Fig. 4. EXIT chart for different clipping probabilities ($N_s = 1024$; SNR = 5 dB, no quantization)

These equations hold true for large block lengths N_s, since the summation within the FFT approximately satisfies the central limit theorem [10]. From (10) it can be seen, that the expectations depend on the noise variance σ_0^2 or the average power φ of the estimated symbols. Furthermore, the FFT increases the dynamic range of a symbol by $20\log(N_s)$ dB. A special case is the number of integer bits for the estimated symbols: the soft mapping (*cf.* figure 1) itself limits the magnitude of the real and imaginary parts for $\bar{\mathbf{s}}$ to one [8], hence $m = 1$. Figure 3 exemplarily depicts the simulated complementary cumulative distribution function (CCDF) [10] for the real part's magnitude of symbol Z'_k and different clipping values. Remark: the exact clipping (based on the simulated CCDF) for $P_{cl} = 0.1$ occurs at a magnitude of 80.9. Using (10), the upper bound is calculated to 169.1 yielding $m = 8$. Hence, the clipping due to a fixed-point representation with 8 integer bits occurs at 256 which in turn corresponds to an actual clipping probability of <0.01. Additionally, the EXIT chart in figure 4 shows the effect of the clipping probability. It can be seen, that a clipping probability of $P_{cl} = 0.1$ already allows a near-optimum progress. Increasing the probability lowers the mutual information I_z and can cause a drop-off for high *a priori* information reliabilities.

IV. Fixed-Point Implementation

Based on the preliminary considerations in section III, an *efficient* fixed-point implementation of the SC-MMSE-FD equalizer can be derived. Efficient means, that a near-optimum equalization performance is achieved with a minimum number of (fractional and integer) bits. We will demonstrate this procedure for the following generic system configuration:

- Modulation scheme: 16-QAM
- SNR range: -2 dB to 12 dB
- Block length: 1024 symbols
- Channel: $h_l \sim \mathcal{CN}(0,1)$, $\mathbb{E}\{|h_l|^2\} = 1$, $N_h = 10$

At first, the minimum number of fractional bits n is determined using the quantized AWGN channel model from section III-A. Since different quantizations of the added/subtracted symbols (*cf.* figure 1) conflict with the discrete power levels of the modulation scheme and cause additional quantization noise, the number of fractional bits is chosen uniformly throughout the entire equalizer. As an example, if the 16-QAM symbol $0.9487 + 0.3162i$ is quantized using $n = 2$ $(1 + 0.25i)$ and $n = 3$ $(1 + 0.375i)$, the result of the subtraction is $0.125i$ and not zero. The EXIT charts in figure 5 depict the progress

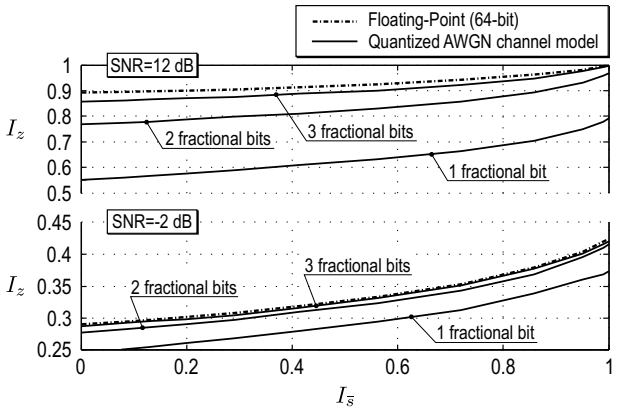

Fig. 5. EXIT charts for the given system configuration and different numbers of fractional bits ($N_s = 1024$; 16-QAM; no clipping)

TABLE II. Number of Integer Bits

Symbols	SNR=-2 dB		SNR=12 dB		**Maximum**
	φ low	φ high	φ low	φ high	m
$\mathbf{r}$	2	2	2	2	**2**
$\mathbf{R}$	7	7	7	7	**7**
$\bar{\mathbf{S}}$	5	7	5	7	**7**
$\mathbf{R}'$	5	7	5	7	**7**
$\mathbf{R}''$	7	6	7	4	**7**
$\mathbf{Z}'$	6	7	7	10	**10**
$\mathbf{z}'$	1	2	2	5	**5**
$\bar{\mathbf{s}}'$	0	3	0	8	**8**
$\mathbf{z}$	1	3	2	8	**8**
low: $0 < \varphi < 0.1$, high: $\varphi > 0.95$					

of the mutual information for the given SNR and different numbers of fractional bits. It can be seen, that for the given system configuration already 3 bits are sufficient to reach a near-optimum equalization performance. Hence, the number of fractional bits is set to $n = 3$ for all representations.

The number of integer bits m is determined using (9) and (10). Table II shows the results for the extreme cases of the given system configuration and a clipping probability of $P_{cl} = 0.1$. To achieve a consistent equalization performance, an efficient fixed-point representation has to cover all parameter values. The number of integer bits for each symbol is therefore set to the maximum number. Lowering the clipping probability from 0.1 to 0.01 would increase the number of integer bits for each symbol by one bit.

The fixed-point representations for all symbols are finally given by the considerations and evaluations that were made above and summarized in table III. They will now be evaluated in an integrated design environment for FPGAs.

V. Evaluation in an FPGA IDE

The results of the fixed-point study and the representations that were obtained for the given system configuration are now evaluated in an integrated design environment (IDE) for FPGAs. We use Xilinx's System Generator for DSP (ISE 13.4) [11] to model the SC-MMSE-FD equalizer in Simulink (Matlab R2011b) [12]. The FFT is implemented with Xilinx's LogiCORE IP Fast Fourier Transform v7.1 [13] in a pipelined, streaming I/O architecture. All other operations are performed with basic signal processing blocks, like multiplier and adder. At a first step, all scaling operations within the FFT are

TABLE III. FINAL FIXED-POINT REPRESENTATION

Symbols	n	m	N	DR [dB]	**Notation**
$\mathbf{r}$	3	2	6	36.12	`Fix_6_3`
$\mathbf{R}$	3	7	11	66.23	`Fix_11_3`
$\bar{\mathbf{s}}$	3	1	5	30.10	`Fix_5_3`
$\bar{\mathbf{S}}$	3	7	11	66.23	`Fix_11_3`
$\mathbf{R}'$	3	7	11	66.23	`Fix_11_3`
$\mathbf{R}''$	3	7	11	66.23	`Fix_11_3`
$\mathbf{Z}'$	3	10	14	84.29	`Fix_14_3`
$\mathbf{z}'$	3	5	9	54.19	`Fix_9_3`
$\bar{\mathbf{s}}'$	3	8	12	72.25	`Fix_12_3`
$\mathbf{z}$	3	8	12	72.25	`Fix_12_3`

deactivated. The fixed-point representations for all symbols are chosen according to table III.

Due to time consuming simulations in Simulink, the SC-MMSE-FD equalizer implementation is only evaluated for specific system parameter values. Nevertheless, the results in figure 6 show, that the expected equalization performance is achieved. After the synthesis and the place&route, the device utilization summary was analyzed and is shown in table IV. As target FPGAs from Xilinx [14] we used a Kintex-7 (XC7K410T), a Spartan-6 (XC6SLX150) and a Virtex-6 (XC6VLX550T). The FPGAs are embedded in software defined radios from Ettus Research (USRP X310, USRP B210) [15] or Nutaq (μSDR420) [16]. The utilization does not account for the FPGA area that is needed for the host communication or the signal preprocessing. In case of the Spartan series, memory optimization of the FFT blocks had to be performed. The fixed-point representation was not affected by this. Activating the scaling mechanisms inside the FFT operations reduced the overall utilization by only 3%. In this case, the fixed-point representations in table III have to be scaled. Nevertheless, the required dynamic ranges remain the same.

VI. CONCLUSION

This paper provides a study on the performance of SC-MMSE equalization for fixed-point implementations. Simple models that emulate the effects of quantization and clipping have been presented. The accuracy of the models was verified using EXIT charts and simulated fixed-point implementations. Given a generic system configuration featuring a 16-QAM, we demonstrated the derivation of an efficient fixed-point representation. The results showed, that on average around 10 bit per symbol are required, to achieve a near optimum equalization performance. The theoretically obtained results for the representations were finally evaluated in an integrated design environment for FPGAs. The simulated equalization performance of the implementation was according to the expected theoretical results. After the synthesis and the place&route, the device utilization summary was analyzed for different FPGAs that are embedded in current software defined radios. For state of the art FPGAs, like the Xilinx Kintex-7, the overall utilization is less than 8%. The results of this paper enable area and memory efficient implementations on FPGAs or GPPs or can be used for SIMD programming.

ACKNOWLEDGMENT

The first author acknowledges that this work was partially supported by the Karlsruhe House of Young Scientists.

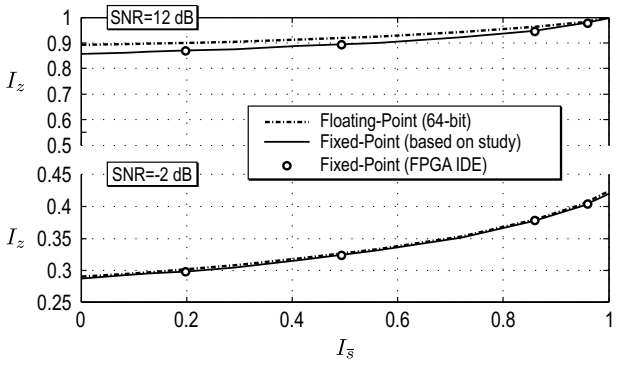

Fig. 6. EXIT chart for the given system configuration ($N_s = 1024$; 16-QAM)

TABLE IV. DEVICE UTILIZATION SUMMARY

Number of	**Kintex-7**	**Spartan-6**	**Virtex-6**
Slice FFs	23299 (4%)	23922 (12%)	23762 (3%)
Slice LUTs	21459 (8%)	24834 (26%)	24930 (7%)
DSP48s	52 (3%)	61 (33%)	52 (6%)

Flip-Flop (FF); Look-up-table (LUT)

REFERENCES

[1] C. Douillard, M. Jezequel, C. Berrou, P. Didier, and A. Picart, "Iterative correction of intersymbol interference: Turbo-equalization," *European Trans. Telecommun.*, vol. 6, no. 5, pp. 507–512, Sept. 1995.

[2] M. Tüchler, R. Koetter, and A. Singer, "Turbo equalization: Principles and new results," *IEEE Trans. Commun.*, vol. 50, no. 5, pp. 754 –767, May 2002.

[3] K. Kansanen and T. Matsumoto, "An analytical method for MMSE MIMO turbo equalizer EXIT chart computation," *IEEE Trans. Wireless Commun.*, vol. 6, no. 1, pp. 59–63, Jan. 2007.

[4] X. Wang and H. Poor, "Iterative (turbo) soft interference cancellation and decoding for coded CDMA," *IEEE Trans. Commun.*, vol. 47, no. 7, pp. 1046 –1061, July 1999.

[5] L. R. Rabiner and B. Gold, *Theory and Application of Digital Signal Processing*. Prentice Hall, June 1975.

[6] R. L. Bidan, C. Loat, and D. Leroux, "Real-time MMSE turbo-equalization on the TMS320C5509 fixed-point DSP," *Proc. IEEE Inter. Conf. on Acoustics, Speech, and Signal Processing*, pp. V325–328, May 2004.

[7] C. Studer, S. Fateh, and D. Seethaler, "ASIC implementation of soft-input soft-output MIMO detection using MMSE parallel interference cancellation," *IEEE Journal of Solid-State Circuits*, vol. 46, no. 7, pp. 1754–1765, July 2011.

[8] M. Tüchler and A. Singer, "Turbo equalization: An overview," in *IEEE Trans. Inf. Theory*, vol. 57, no. 2, Feb. 2011, pp. 920–952.

[9] S. ten Brink, "Convergence of iterative decoding," *Electronics Letters*, vol. 35, no. 10, pp. 806–808, May 1999.

[10] A. Gut, *Probability: A Graduate Course (Springer Texts in Statistics)*. Springer, Feb. 2010.

[11] (2014, June) Xilinx - System Generator for DSP. [Online]. Available: http://www.xilinx.com/tools/sysgen.htm

[12] (2014, June) Simulink - Simulation and Model-based Design. [Online]. Available: http://www.mathworks.de/products/simulink/

[13] (2014, June) Xilinx - LogiCORE IP Fast Fourier Transform v7.1. [Online]. Available: http://www.xilinx.com/products/intellectual-property/FFT.htm

[14] (2014, June) Xilinx - All Programmable FPGAs. [Online]. Available: http://www.xilinx.com/products/silicon-devices/fpga/

[15] (2014, June) Ettus Research. [Online]. Available: http://www.ettus.com/

[16] (2014, June) Nutaq. [Online]. Available: http://www.nutaq.com/