

Implementation of a Combined OFDM-Demodulation and WCDMA-Equalization Module

David van Kampen, Klaas L. Hofstra, Jordy Potman and Sabih H. Gerez
University of Twente, Faculty of EEMCS,
Signal and System Group,
P.O. box 217 - 7500 AE Enschede - The Netherlands
Phone: +31 53 489 2773 Fax: +31 53 489 1060
E-mail: s.h.gerez@utwente.nl

Abstract—For a dual-mode baseband receiver for the OFDM Wireless LAN and WCDMA standards, integration of the demodulation and equalization tasks on a dedicated hardware module has been investigated.

For OFDM demodulation, an FFT algorithm based on cascaded twiddle factor decomposition has been selected. This type of algorithm combines high spatial and temporal regularity in the FFT data-flow graphs with a minimal number of computations. A frequency-domain algorithm based on a circulant channel approximation has been selected for WCDMA equalization. It has good performance, low hardware complexity and a low number of computations. Its main advantage is the reuse of the FFT kernel, which contributes to the integration of both tasks.

The demodulation and equalization module has been described at the register transfer level with the in-house developed Arx language. The core of the module is a pipelined radix-2³ butterfly combined with a complex multiplier and complex divider. The module has an area of 0.447 mm² in 0.18 μ m technology and a power consumption of 10.6 mW. The proposed module compares favorably with solutions reported in literature.

Keywords—OFDM demodulation, WCDMA, frequency-domain equalization, architecture design.

I. INTRODUCTION

Support of multiple radio standards will be a key feature for next-generation mobile receivers. Multi-standard receivers allow to maximally utilize the available radio environment. The research described in this paper has been performed in the context of the design of a dual-mode receiver that supports OFDM-WLAN and WCDMA standards. These standards give complementary service as both high data rates and full mobility can be obtained. It is challenging to integrate these computationally complex standards on a signal processing architecture that is both area and power efficient.

The WCDMA and OFDM-WLAN receiver systems can be studied at different levels when searching for possibilities for efficient integration. Consider first

the standards at the system level. Figure 1 shows the required system-level tasks of a general radio receiver. In the analog front-end, the radio signal is mixed to baseband. The standards operate at different frequency bands, e.g. WCDMA downlink at 2 GHz and OFDM-WLAN standard IEEE 802.11a at 5 GHz. The first signal processing tasks in the digital domain tasks include filtering, e.g. for pulse shaping used in WCDMA [1] and synchronization of the the signal in time and frequency. The subsequent demodulation and equalization tasks reverse the distortion of the multipath channel propagation and result in equalized symbol sequences, e.g. *quadrature phase shift keying* (QPSK) symbols.

The modulation techniques used in WCDMA and OFDM-WLANs differ significantly. With *orthogonal frequency division multiplexing* (OFDM), multiple symbols are transmitted over parallel carriers. The carriers can be efficiently demodulated with a *fast Fourier transform* (FFT) operation to obtain the original symbols. The symbols then have to be equalized separately to reduce amplitude and phase distortion due to multipath propagation.

In a *code division multiple access* (CDMA) communication system, multiple symbol sequences are code-multiplexed to enable simultaneous transmission. Each symbol sequence is encoded with a unique code with a higher rate. The WCDMA standard uses *orthogonal variable spreading factor* (OVSF) codes. All encoded signals are summed and this combined CDMA signal is then scrambled and modulated on a single carrier. Multipath propagation destroys the code-orthogonality of the original CDMA signal. Equalization is then required to correct the distorted signal. The equalized samples are subsequently descrambled and despread into separate symbol sequences by correlating with the proper codes.

After the demodulation and equalization tasks, the

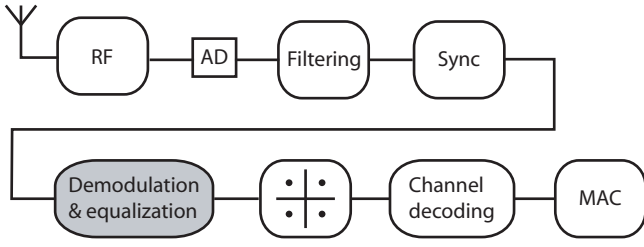


Fig. 1. Tasks in a general radio receiver

equalized symbols are demapped to bits. The bits are then processed by the channel decoding block for error correction and detection. The result is finally forwarded to the upper protocol layer, i.e. the *medium access control* (MAC) layer.

Efficient integration of the demodulation and equalization tasks on a dedicated hardware module has been investigated more extensively. This paper presents the results and is organized as follows. First the algorithm exploration for the demodulation and equalization tasks are presented. Algorithms have been evaluated using the following criteria: performance of the algorithm, computational efficiency and related hardware complexity. Furthermore, the reusability of the corresponding hardware kernels between the set of algorithms is important, because it determines the efficiency of integration. In Section III, the implementation and synthesis of the hardware module that processes these algorithms is discussed. In Section IV a comparison is made with equalization and demodulation hardware implementations reported in literature. Finally, the conclusions are presented.

II. ALGORITHM EXPLORATION

A. OFDM demodulation

For OFDM demodulation, an FFT algorithm based on cascaded twiddle factor decomposition [2] has been selected. This type of algorithm combines high spatial and temporal regularity in the FFT data-flow graphs with a minimal number of computations. Concretely, the radix-2³ FFT algorithm is available for FFTs whose length N is a power of eight. It has been chosen because the number of carriers in OFDM-WLANs is 64. The decomposition starts with a four-dimensional index map of the discrete frequency and time index of the *discrete Fourier transform* (DFT)

equation:

$$n = \left\langle \frac{N}{2}n_1 + \frac{N}{4}n_2 + \frac{N}{8}n_3 + n_4 \right\rangle \quad (1)$$

$$k = \left\langle k_1 + 2k_2 + 4k_3 + 8k_4 \right\rangle \quad (2)$$

The DFT equation then becomes:

$$X(k_1 + 2k_2 + 4k_3 + 8k_4) =$$

$$\sum_{n_4=0}^{\frac{N}{8}-1} \sum_{n_3=0}^1 \sum_{n_2=0}^1 \sum_{n_1=0}^1 x\left(\frac{N}{2}n_1 + \frac{N}{4}n_2 + \frac{N}{8}n_3 + n_4\right) W_N^{nk} \quad (3)$$

The twiddle factor W_N^{nk} , which is short for $e^{-j\frac{2\pi}{N}nk}$, is then decomposed as follows:

$$\begin{aligned} W_N^{nk} &= W_N^{\left(\frac{N}{2}n_1 + \frac{N}{4}n_2 + \frac{N}{8}n_3 + n_4\right)(k_1 + 2k_2 + 4k_3 + 8k_4)} \\ &= W_N^{\frac{N}{2}n_1 k_1} W_N^{\frac{N}{4}n_2(k_1 + 2k_2)} W_N^{\frac{N}{8}n_3(k_1 + 2k_2 + 4k_3)} \times \\ &\quad W_N^{n_4(k_1 + 2k_2 + 4k_3 + 8k_4)} \\ &= (-1)^{n_1 k_1} (-j)^{n_2(k_1 + 2k_2)} W_8^{n_3(k_1 + 2k_2 + 4k_3)} \times \\ &\quad W_N^{n_4(k_1 + 2k_2 + 4k_3)} W_N^{8n_4 k_4} \end{aligned} \quad (4)$$

If (4) is substituted in (3), and the summation is expanded for indices n_1 , n_2 and n_3 one finds a set of 8 DFTs of length $\frac{N}{8}$:

$$\begin{aligned} X(k_1 + 2k_2 + 4k_3 + 8k_4) &= \\ \sum_{n_4=0}^{\frac{N}{8}-1} \left[T_{\frac{N}{8}}(n_4, k_1, k_2, k_3) W_N^{n_4(k_1 + 2k_2 + 4k_3)} \right] W_N^{n_4 k_4} \end{aligned} \quad (5)$$

where the following equations represent three cascaded radix-2 butterfly stages, hence the name radix-2³ butterfly (shown in Figure 2):

$$\begin{aligned} T_{\frac{N}{8}}(n', k_1, k_2, k_3) &= \\ H_{\frac{N}{4}}(n', k_1, k_2) + W_8^{(k_1 + 2k_2 + 4k_3)} H_{\frac{N}{4}}\left(n' + \frac{N}{8}, k_1, k_2\right) \end{aligned} \quad (6)$$

$$\begin{aligned} H_{\frac{N}{4}}(n'', k_1, k_2) &= \\ B_{\frac{N}{2}}(n'', k_1) + (-j)^{(k_1 + 2k_2)} B_{\frac{N}{2}}\left(n'' + \frac{N}{4}, k_1\right) \end{aligned} \quad (7)$$

and

$$B_{\frac{N}{2}}(n''', k_1) = x(n''') + (-1)^{k_1} x\left(n''' + \frac{N}{2}\right) \quad (8)$$

By recursively applying the decomposition of the remaining set of $\frac{N}{8}$ DFTs (i.e. by applying another

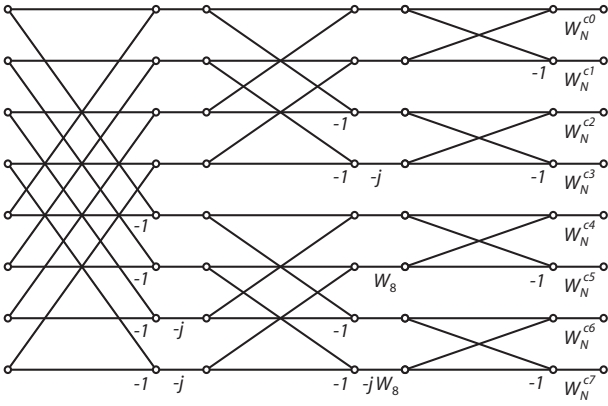


Fig. 2. Signal flow graph of the radix- 2^3 butterfly

factorization step to index n_4 and discrete frequency k_4 in the summation of (5)), the complete FFT algorithm with radix- 2^3 butterflies is obtained. The signal flow graph of the complete FFT using the radix- 2^3 algorithm has a structure similar to the radix-2 algorithm with high spatial regularity. The non-constant twiddle-factor multiplications are now conveniently located at the output nodes of the radix- 2^3 butterfly. Inside the radix- 2^3 butterfly, there are several constant multiplications. In hardware, the multiplication with twiddle factor $-j$ can be implemented without a multiplier. Further, the twiddle factor in (6) can be decomposed into:

$$\begin{aligned} W_8^{(k_1+2k_2+4k_3)} &= (-1)^{k_3} (-j)^{k_2} W_8^{k_1} \\ &= (-1)^{k_3} (-j)^{k_2} \left(\frac{\sqrt{2}}{2} (1-j) \right)^{k_1} \quad (9) \end{aligned}$$

This twiddle factor requires only two real multiplications. It can be implemented in hardware using a fixed-coefficient multiplier, which is cheaper than a general-purpose multiplier. When a fixed-coefficient multiplier is used, there are only $\log_8 N$ stages with N non-trivial multiplications per stage. The computational complexity of the N -point FFT is therefore only $N \log_8 N = \frac{N}{3} \log_2 N$, which is the same as for split-radix FFT algorithm [3] that is known for its computational efficiency. Split-radix FFT however, has a more complicated structure than the radix- 2^3 butterfly.

B. Equalization in WCDMA

The traditional way to detect the symbols in a distorted CDMA signal is a correlation-based Rake receiver. The Rake receiver algorithm has both low computational and hardware complexity. For heavily

loaded cells, i.e. when many channelization codes are used, the performance of the Rake receiver decreases significantly. Especially for the high-data rate HSDPA mode, algorithms that give better performance are desired because heavily loaded cells occur frequently. In those cases, *multi-user detection* (MUD) algorithms give good performance although at relatively high complexity [4]. MUD algorithms take the information of all users jointly into account to estimate the signal of each individual CDMA user. These type of algorithms are best used in the base station receiver, where all user signals have to be processed. Equalization algorithms on the other hand, are more suitable for use in the mobile receiver because the composite CDMA signal experiences a single channel from base station to the mobile receiver.

The WCDMA equalizer has to reverse the effect of the noisy multipath fading channel on the transmitted composite CDMA signal $t(n)$. The channel can be modeled as a linear filtering operation:

$$r(n) = \sum_{l=0}^{L-1} h(l,n)t(n-l) + g(n) \quad (10)$$

where coefficient $h(l,n)$ represents the l^{th} complex component of the discretized channel impulse response at time n . These channel coefficients also include the contribution of the transmitter and receiver pulse-shaping filters [4]. The excess delay, or the maximum relative delay of the multipath components, is L samples. The noise component is modeled with variable $g(n)$.

Using matrix equations the channel model becomes:

$$\mathbf{r} = \mathbf{H}\mathbf{t} + \mathbf{g} \quad (11)$$

where received vector \mathbf{r} is of length N . The channel coefficient matrix \mathbf{H} has dimension $N \times (N + L - 1)$ and the transmitted vector \mathbf{t} is of length $N + L - 1$, because of the multipath delay effect. The noise contribution is contained in vector \mathbf{g} , which has a length of N samples.

For block-based equalization, the estimate of the transmitted vector, $\hat{\mathbf{t}}$, is obtained by multiplying the received vector \mathbf{r} with an equalization coefficient matrix \mathbf{W} , with dimension $N \times N$, as in:

$$\hat{\mathbf{t}} = \mathbf{W}\mathbf{r} = \mathbf{W}(\mathbf{H}\mathbf{t} + \mathbf{g}) \quad (12)$$

When equalization matrix \mathbf{W} is calculated for the channel transfer function of (12) based on the *minimum mean squared error* (MMSE) criterion, one finds:

$$\mathbf{W} = \left(\mathbf{H}^H \mathbf{H} + \frac{\sigma_g^2}{\sigma_t^2} \mathbf{I} \right)^{-1} \mathbf{H}^H \quad (13)$$

where σ_g^2 is the noise variance and σ_t^2 is the variance of the transmitted CDMA signal. This ratio therefore corresponds to the inverse of the signal-to-noise ratio.

This linear time-domain MMSE-equalization algorithm gives good performance at considerable complexity. For example, calculating the coefficient matrix requires a matrix inversion operation, which is costly to do in hardware. Multiplication of vector \mathbf{r} with matrix \mathbf{W} can be approximated with a linear filtering operation. The filter coefficients are then taken from a column of matrix \mathbf{W} . Filtering in the frequency domain instead of the time domain can reduce the number of operations. For example, filtering of long sequences in the frequency domain with the overlap-save technique requires less multiplications than linear time-domain filtering, due to the efficient FFT algorithm [3].

Frequency-domain equalization (FDE) is computationally more efficient than *time-domain equalization* (TDE) [5]. An algorithm based on a circulant channel approximation [6] results in even lower complexity and has therefore been selected for implementation. The algorithm starts with approximating the linear convolution of the transmitted signal with the channel impulse response in (10) with a circular convolution of these two signals. For block-based processing the channel transfer function is now approximated as:

$$\mathbf{r} = \mathbf{H}_C \mathbf{t} + \mathbf{g} \quad (14)$$

where the received block \mathbf{r} of length N is assumed to depend only on the samples of the current transmitted block \mathbf{t} , which is of length N as well. In reality the last $L - 1$ samples of the previous transmitted block affect the received samples of the current block, but this contribution is now neglected. The channel matrix \mathbf{H}_C has size $N \times N$ and is circulant. The channel impulse response is assumed constant during the entire block as well. An example of a circulant channel matrix with a channel impulse response $h(n)$ of length four and four padded zeroes (so $N = 8$) is given as:

$$\mathbf{H}_C = \begin{bmatrix} h_0 & 0 & 0 & 0 & 0 & h_3 & h_2 & h_1 \\ h_1 & h_0 & 0 & 0 & 0 & 0 & h_3 & h_2 \\ h_2 & h_1 & h_0 & 0 & 0 & 0 & 0 & h_3 \\ h_3 & h_2 & h_1 & h_0 & 0 & 0 & 0 & 0 \\ 0 & h_3 & h_2 & h_1 & h_0 & 0 & 0 & 0 \\ 0 & 0 & h_3 & h_2 & h_1 & h_0 & 0 & 0 \\ 0 & 0 & 0 & h_3 & h_2 & h_1 & h_0 & 0 \\ 0 & 0 & 0 & 0 & h_3 & h_2 & h_1 & h_0 \end{bmatrix} \quad (15)$$

A special property of the circulant matrix \mathbf{H}_C , which is fully determined by vector \mathbf{h} , is that it can be decomposed as:

$$\mathbf{H}_C = \mathbf{F}^{-1} \text{diag}(\mathbf{F}\mathbf{h})\mathbf{F} = \mathbf{F}^{-1} \Lambda_{H_C} \mathbf{F} \quad (16)$$

where \mathbf{F} is the Fourier matrix corresponding to the DFT operation and \mathbf{F}^{-1} its inverse. The *diag* operation places the values of a vector on the diagonal elements of a matrix. Matrix Λ_{H_C} is the diagonalized eigen value matrix of \mathbf{H}_C , where the eigen values of \mathbf{H}_C are equal to the frequency transform of vector \mathbf{h} .

When calculating the equalization matrix in (13) and expanding the circulant channel matrix one finds:

$$\mathbf{W}_C = \mathbf{F}^{-1} \left[\Lambda_{H_C}^H \Lambda_{H_C} + \frac{\sigma_g^2}{\sigma_t^2} \mathbf{I} \right]^{-1} \Lambda_{H_C}^H \mathbf{F} \quad (17)$$

The estimate of the transmitted vector $\hat{\mathbf{t}}_C$ is now found as:

$$\hat{\mathbf{t}}_C = \mathbf{W}_C \mathbf{r} = \mathbf{F}^{-1} \left[\Lambda_{H_C}^H \Lambda_{H_C} + \frac{\sigma_g^2}{\sigma_t^2} \mathbf{I} \right]^{-1} \Lambda_{H_C}^H \mathbf{F} \mathbf{r} \quad (18)$$

This corresponds to equalization of the received vector \mathbf{r} in the frequency domain. $\mathbf{F}\mathbf{r}$ first gives the frequency transform vector \mathbf{R} . This vector is then equalized with frequency-domain equalization coefficient matrix \mathbf{M}_C :

$$\mathbf{M}_C = \left[\Lambda_{H_C}^H \Lambda_{H_C} + \frac{\sigma_g^2}{\sigma_t^2} \mathbf{I} \right]^{-1} \Lambda_{H_C}^H \quad (19)$$

Note that matrix \mathbf{M}_C is diagonal and its k^{th} diagonal element is equal to:

$$M_{C,k} = \frac{H_k^*}{|H_k|^2 + \frac{\sigma_g^2}{\sigma_t^2}} \quad (20)$$

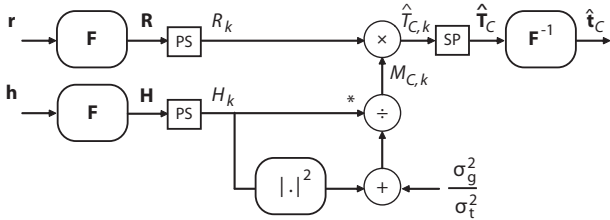


Fig. 3. Data-flow graph for MMSE-equalization in the frequency-domain using a circulant channel approximation

where H_k is the k^{th} discrete frequency component of the frequency transform of the (estimated) channel impulse response.

By transforming frequency transform $\hat{\mathbf{T}}_C$ to the time domain by multiplying with \mathbf{F}^{-1} , the estimated chip vector $\hat{\mathbf{t}}_C$ is finally found.

This frequency-domain equalization algorithm based on a circulant approximation has relatively low complexity. Firstly, the matrix inversion needed for calculating the MMSE time-domain equalization coefficients in (13) has been reduced to the simpler division operation in (20). Secondly, the matrix multiplication of MMSE-TDE reduces to vector operations. Matrix \mathbf{M}_C is diagonal, which implies that the matrix multiplication comes down to multiplying its diagonal values element-wise with vector \mathbf{R} . In addition, the (inverse) frequency transforms are vector operations as well. When the length of \mathbf{r} is chosen a power of two, an efficient FFT algorithm can be used.

Figure 3 shows the data-flow graph of frequency-domain equalization based on a circulant channel approximation. In the figure, PS means the transition from parallel (blocks) to serial (samples), and SP is the transition from serial to parallel.

The question is whether the linear filtering of the multipath channel can be approximated with a circulant channel model. The circulant approximation is allowed when zero-padding or a cyclic prefix is employed [6]. The WCDMA standard does not use these mechanisms, so another approach has to be taken to allow this approximation. It has been shown in [6] and [5], that equalization based on a circulant channel approximation results in large differences for samples at the boundaries of the equalized block. The overlap-cut technique [6] can be used, because samples at the center of the block are good estimates of the transmitted signal. Overlap-cut only takes the properly equalized chips of the center part of the block and reconstructs the entire transmitted signal in an overlapping fashion.

Simulations of a WCDMA receiver system with the MMSE-FDE algorithm using overlap-cut gives performance close to linear MMSE-equalization in the time domain [6]. Using a block size of $N = 256$ and an overlap of 32 samples at both edges, the algorithm gives sufficient performance at moderate computational complexity.

MMSE-FDE has low hardware and computational complexity compared to MMSE-TDE. However, its main advantage is that it enables the integration of OFDM demodulation and WCDMA equalization on a single architecture, since the FFT computational kernel is now used by algorithms of both systems.

C. Despreading in HSDPA mode

The despreading operation converts the equalized CDMA signal to multiple sequences of user symbols. In the WCDMA standard, several channelization codes with different spreading factors can be used. In the HSDPA mode, however, the spreading factor of the data channels is fixed to 16. The channelization codes are orthogonal and correspond to a row in a Hadamard matrix of size 16×16 . This can be exploited to perform the despreading operation for multiple codes in the HSDPA mode more efficiently with the Hadamard transform. An N -point Hadamard transform corresponds to multiplying the Hadamard matrix of size $N \times N$ with a vector of length N . The two-point Hadamard matrix \mathbf{H}_2 is defined as:

$$\mathbf{H}_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad (21)$$

The N -point Hadamard matrix, with N being a power of 2, is defined as:

$$\mathbf{H}_N = \mathbf{H}_{\frac{N}{2}} \otimes \mathbf{H}_2 \quad (22)$$

where \otimes means a tensor product.

Similar to the DFT, the Hadamard transform can be done in $N \log_2 N$ rather than N^2 operations, by recursively decomposing the matrix. In analogy with the FFT, the resulting algorithm is called the *fast Hadamard transform* (FHT). For $N = 8$, for example, one finds the Hadamard matrix \mathbf{H}_8 given as:

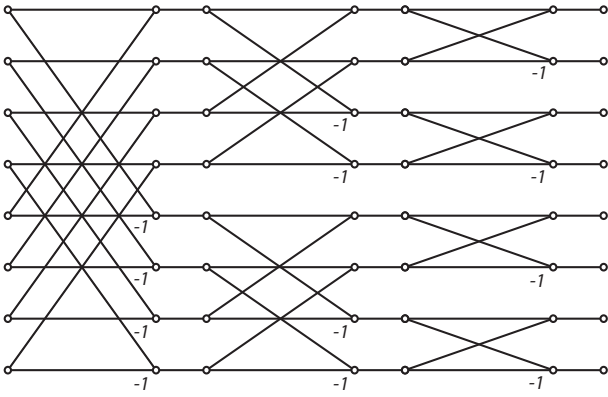


Fig. 4. Signal flow graph of a 8-point fast Hadamard transform

$$\begin{aligned}
 \mathbf{H}_8 &= \begin{bmatrix} H_4 & H_4 \\ H_4 & -H_4 \end{bmatrix} = \begin{bmatrix} H_2 & H_2 & H_2 & H_2 \\ H_2 & -H_2 & H_2 & -H_2 \\ H_2 & H_2 & -H_2 & -H_2 \\ H_2 & -H_2 & -H_2 & H_2 \end{bmatrix} \\
 &= \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix} \quad (23)
 \end{aligned}$$

The signal flow graph of a 8-point fast Hadamard transform is given in Figure 4. The SFG of this FHT has the same structure as the SFG of the radix- 2^3 butterfly (Figure 2), the only difference being that it does not require multiplications with twiddle factors.

With the FHT, despreading of the worst case HS-DPA scenario of 15 active codes can be efficiently done by performing an FHT of length 16. Only 64 additions/subtractions are needed to calculate all symbols of the 16 possible channelization codes. This is a reduction in number of add/subtract operations of almost a factor four. Additionally, the similar structure of the signal flow graphs of the FHT and the cascade-decomposition butterflies allows elegant integration on a single butterfly implementation in hardware.

III. HARDWARE IMPLEMENTATION

A. Module design

An architecture has been designed of a module that is dedicated to process the demodulation and equal-

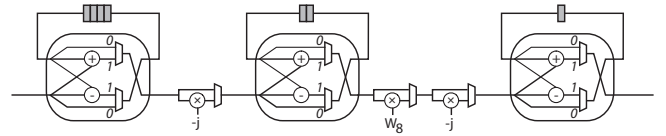


Fig. 5. Data path of the pipelined radix- 2^3 butterfly unit

ization algorithms. The core of this module is a pipelined radix- 2^3 butterfly shown in Figure 5. It consists of three radix-2 butterfly components, with FIFO buffers and IO-multiplexers. The multiplexers either pass the input node or the radix-2 butterfly output to the FIFO buffer. For example, in the first four cycles, the top four nodes of the radix- 2^3 signal flow graph are loaded in the first FIFO buffer. In the following four cycles, the lower nodes are fed and the radix-2 butterflies of the first stage of the radix- 2^3 SFG can then be calculated. The complete radix- 2^3 SFG can thus be performed with this pipelined unit in a few cycles. Multiplication with internal twiddle factors $-j$ and W_8 is optionally selected depending on the clock cycle. The pipelined butterfly unit can be reconfigured as well to perform a radix- 2^2 butterfly [2] needed to calculate a 256-point FFT.

In Figure 6 the data path of the demodulation and equalization module is presented. The input of the butterfly is connected to the data RAM. The output of the pipelined butterfly unit is connected to a complex multiplier. It is used to apply the twiddle factors (stored in the twiddle ROM), for descrambling and for equalization (stored in the coefficient RAM). The output of the multiplier is subsequently stored in the data RAM. To calculate the complex MMSE equalization coefficients, a divider module is instantiated. The dividend input is the complex conjugate of the butterfly output. The divisor input is the absolute squared of the butterfly output plus the noise-to-signal ratio. The absolute squared value is calculated on the multiplier as well.

B. Module implementation

B.1 Datapath implementation

The architecture of the demodulation and equalization module has been implemented using the *Arx* language [7, 8] developed in-house. *Arx* descriptions are always at the *register-transfer level* (RTL), i.e. the user needs to indicate the computations performed in each clock cycle. A “behavioral” style of description amounts to lumping large numbers of computations into a single clock cycle. The *Arx* toolset gen-

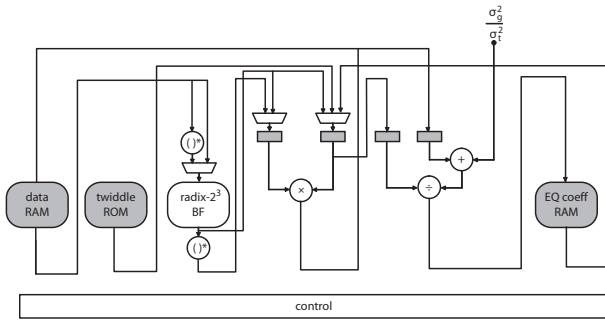


Fig. 6. Data path of the demodulation and equalization module

erates either C++ code for fast simulation or VHDL for synthesis. Although fixed-point data types (bit-true descriptions) are required for synthesis, floating-point data types are supported for design evaluation through simulation.

Stepwise refinement has been used to arrive from a behavioral floating-point description to a fixed-point cycle-true design for the arithmetic blocks and interconnect of the module. For the divider, a structural VHDL description of a parameterizable array-divider has been obtained from [9]. Its designers claim smaller area, faster performance and less power consumption than the array divider available from the Synopsys Designware library.

For OFDM demodulation/equalization control has been implemented to run a 64-point FFT. For the WCDMA system, control has been implemented for the FDE algorithm with a block size of 256 and for calculation of the MMSE-equalization coefficients from a channel impulse response vector. The hardware and control for generation of descrambling codes and for the despreading operation have not been implemented yet.

B.2 Finite word length optimization

For the WCDMA-FDE mode, a finite word length optimization has been done for all nodes in the data path of the module. The goal was to meet a performance target with minimal area, i.e. minimal data path width. The target was an uncoded bit-error rate smaller than 10^{-2} at 20 dB SNR for 15 users with spreading factor 16 in the Vehicular A channel [1]. The resulting word length at the input and output ports of the functional units is 10 bits.

B.3 Performance measurement

The uncoded BER performance of WCDMA equalization running on the optimized module is shown in

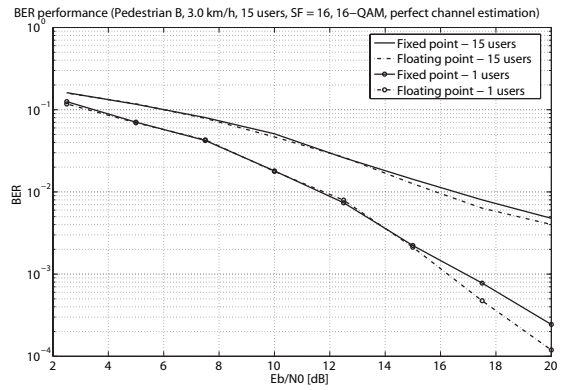


Fig. 7. BER performance plot of the demodulation and equalization module for the Pedestrian B (3 km/h) channel model

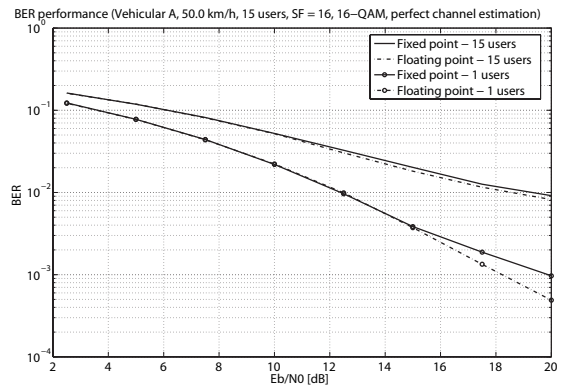


Fig. 8. BER performance plot of the demodulation and equalization module for the Vehicular A (50 km/h) channel model

Figures 7 and 8. The ITU Pedestrian B and ITU Vehicular A channel models [1] with added white Gaussian noise have been used in the simulations. The single-user case is compared to the case where 15 channelization codes with spreading factor 16 are used. The performance of the floating-point algorithm is shown as well and is closely approached by that of the finite-word length hardware model. The performance plot is still an upper bound, because ideal channel estimation was used.

B.4 Synthesis results

The module has been synthesized with the Synopsys Design Compiler using a $0.18 \mu\text{m}$ UMC library. The total area of the arithmetic functional units and control is 0.109 mm^2 . The area of the required memories for storage of samples, equalization coefficients and twiddle factors is relatively high: 0.338 mm^2 . The total area of the module is therefore 0.447 mm^2 .

The power consumption has been estimated using Synopsys Power Compiler in combination with switching activity collected from Modelsim VHDL simulation. The RTL description of the circuit has been simulated with signals typical for the WCDMA-FDE mode. A power consumption figure of 10.6 mW has been found for the circuit and memories running at 50 MHz. The normalized power consumption is therefore 0.21 mW/MHz.

IV. COMPARISON

Several solutions for combining OFDM-WLAN and WCDMA standards have been found in literature. With the obtained information only a rough comparison can be made with the proposed module because the designs do not all have the same functionality and hardware synthesis results were not always provided.

A. Performance

Nilsson et al. [10] and Harju et al. [11] combine an FFT with a Rake receiver for their multi-standard systems also capable of processing WCDMA and OFDM-WLANs. The performance of the proposed frequency-domain equalization for WCDMA will be better than with the Rake receiver used in these receiver systems.

The performance of Heyne et al. [12] is similar to the MMSE-FDE equalizer. Their accelerator is based on a CORDIC processor element that can be used for both equalization in a WCDMA receiver and performing an FFT operation. The system equations for WCDMA equalization are solved in a least-squares sense by a QR-decomposition. The required Givens-transformations are then performed with the CORDIC processor elements.

Several tile-based reconfigurable architectures have been studied as well, for example the architecture of Kapoor et al. [13], of Matúš et al. [14] and the Montium tile processor [15]. These DSP architectures can be configured to run the same FDE algorithm, thus giving similar performance.

B. Area

Table I summarizes the area figures of the architectures translated to the area in 0.18 μm technology. Note that these architectures do not all have the same functionality. Furthermore, the area of the required memory is not always included. Nilsson et al. have implemented two separate accelerators for FFT and Rake receiver. The total area of these accelerators is larger than the area of the proposed design. They have implemented despreading and scram-

TABLE I
AREA FIGURES OF DIFFERENT ARCHITECTURES

Implementation	Area [mm^2]
Proposed architecture	0.447
Nilsson	0.470
Kapoor	0.600
Matúš	1.46
Montium	3.51

ble code generation, but the used memory area is not included in this area figure. Either way, there is no gain in reusing arithmetic blocks between the demodulation and accelerators. In the architecture of Harju et al. multipliers are shared between the correlator engine for the Rake receiver and six radix-2 FFT butterflies. They would likely gain area by sharing the multipliers but synthesis results are not provided. Heyne et al. cleverly integrate equalization and demodulation with algorithms that reuse a CORDIC element. The hardware has not been implemented. The tile-based reconfigurable architectures (Kapoor, Matúš, Montium) all pay a large price in area for their programmability. This level of flexibility is not required.

C. Power consumption

The power consumption can be compared with the Montium tile processor. The normalized power consumption of the Montium for FFTs was 0.550 mW/MHz in 0.13 μm technology. Their power figure to calculate a 64-point FFT or run a Rake receiver is about two to four times higher than the power consumption figure for the WCDMA equalization mode in 0.18 μm technology [16]. The average dynamic power consumption of the Montium is smaller than for several other reconfigurable tile processors, FPGAs and microprocessors [15]. ASIC solutions, however, are more energy efficient than the Montium tile processor.

V. CONCLUSIONS

We have presented a solution for efficient integration of the demodulation and equalization tasks of the OFDM-WLAN and WCDMA standards for a dual-mode baseband receiver architecture.

The selected cascade decomposition FFT algorithms for OFDM demodulation are computationally efficient and their signal flow graphs have high spatial and temporal regularity which is advantageous for hardware implementation.

The selected MMSE frequency-domain equalization algorithm for WCDMA equalization is favorable for efficient integration of both standards on a dual-mode receiver since it reuses the FFT kernel. The algorithm gives good performance for the high data rate modes of WCDMA and has a low computational and hardware complexity.

The fast Hadamard transform is a computationally efficient algorithm for despreading the HSDPA data channels in WCDMA. The signal flow graph of fast Hadamard transforms has the same structure as that of the selected radix-2³ butterfly which allows elegant integration in hardware.

An RTL model of the demodulation and equalization module has been efficiently implemented in Arx. The finite-word-length optimized demodulation and equalization module gives performance close to the floating-point models. The total area of the module is 0.447 mm² and it has a power consumption of 10.6 mW for the WCDMA-FDE mode. The proposed circuit compares favorably in terms of area, power and performance with results found in literature.

REFERENCES

- [1] 3GPP. *3GPP TS 25.101: UE Radio Transmission and Reception (FDD)*, v6.0.0 edition, March 2003.
- [2] S. He and M. Torkelson. Designing pipeline FFT processor for OFDM (de)modulation. *Proceedings of the URSI International Symposium on Signals, Systems, and Electronics*, pages 257–262, October 1998.
- [3] J.G. Proakis, C.M. Rader, F. Ling, C.L. Nikias, M. Moonen, and I.K. Proudler. *Algorithms for statistical signal processing*. Prentice-Hall Inc., Upper Saddle River, New Jersey, U.S.A., 2002.
- [4] R. Tanner and J. Woodard. *WCDMA - requirements and practical design*. John Wiley & Sons Ltd, The Atrium, Southern Gate, Chichester, England, 2004.
- [5] D. Lo Iacono, E. Messina, C. Volpe, and A. Spalvieri. Serial block processing for multi-code WCDMA frequency domain equalization. *Proceedings of the IEEE Wireless Communications and Networking Conference*, 1:164–170, March 2005.
- [6] I. Martoyo, T. Weiss, F. Capar, and F.K. Jondral. Low complexity CDMA downlink receiver based on frequency domain equalization. *Proceedings of IEEE 58th Vehicular Technology Conference*, 2:987–991, October 2003.
- [7] K.L. Hofstra, S.H. Gerez, and D. van Kampen. A language and toolset for the synthesis and efficient simulation of clock-cycle-true signal-processing algorithms. *Proceedings of the 16th Annual Workshop on Circuits, Systems and Signal Processing, ProRISC 2005, Veldhoven, The Netherlands*, November 2005.
- [8] K.L. Hofstra and S.H. Gerez. Arx: A toolset for the efficient simulation and direct synthesis of high-performance signal processing algorithms. In *International Conference on High Performance Embedded Architectures and Compilers*, Ghent, Belgium, January 2007.
- [9] Kirchhof-Institut für Physik. ALICE TRD - ALU project. <http://www.kip.uni-heidelberg.de/ti/TRD/projects>, 2001.
- [10] A. Nilsson, E. Tell, and D. Liu. An accelerator architecture for programmable multi-standard baseband processors. *Proceedings of the Wireless Networks and Emerging Technologies conference, 2004, Banff, Canada.*, July 2004.
- [11] L. Harju and J. Nurmi. A programmable baseband receiver platform for WCDMA/OFDM mobile terminals. *Proceedings of the IEEE Wireless Communications and Networking Conference*, 1:33–38, March 2005.
- [12] B. Heyne and J. Götze. A Cordic based reconfigurable CDMA equalizer for long codes and its comparison to the Rake receiver. *Proceedings of the Software Defined Radio Technical Conference, Phoenix, USA*, November 2004.
- [13] A. Kapoor, S.H. Gerez, F.W. Hoeksema, and R. Schiphorst. A reconfigurable tile-based architecture to compute FFT and FIR functions in the context of software-defined radio. *Proceedings of the 16th Annual Workshop on Circuits, Systems and Signal Processing, ProRISC 2005, Veldhoven, The Netherlands*, November 2005.
- [14] E. Matúš, O. Prátor, A. Zoch, G. Cichon, and G.P. Fettweis. Software reconfigurable baseband ASSP for dual mode UMTS/WLAN 802.11b receiver. *Proceedings of the 13th IST Mobile and Wireless Communications Summit, Lyon, France*, June 2004.
- [15] P.M. Heysters. *Coarse-grained reconfigurable processors*. PhD thesis, University of Twente, September 2004.
- [16] D. van Kampen. Implementation of an OFDM-demodulation and WCDMA-equalization module for a dual-mode receiver. Master's thesis, University of Twente, SAS11.06, 2006.