

## Research Article

# Implementing a DVB-T/H Receiver on a Software-Defined Radio Platform

**Yong Jiang, Wen Xu, and Cyprian Grassmann**

*Infineon Technologies AG, 85579 Neubiberg, Germany*

Correspondence should be addressed to Wen Xu, wen.xu@ieee.org

Received 1 September 2008; Accepted 16 February 2009

Recommended by Daniel Iancu

Digital multimedia broadcasting is available in more and more countries with various forms. One of the most successful forms is Digital Video Broadcasting for Terrestrial (DVB-T), which has been deployed in most countries of the world for years. In order to bring the digital multimedia broadcasting services to battery-powered handheld receivers in a mobile environment, Digital Video Broadcasting for Handheld (DVB-H) has been formally adopted by ETSI. More advanced and complex digital multimedia broadcasting systems are under development, for example, the next generation of DVB-T, a.k.a. DVB-T2. Current commercial DVB-T/H receivers are usually built upon dedicated application-specific integrated circuits (ASICs). However, ASICs are not flexible for incoming evolved standards and less overall-area efficient since they cannot be efficiently reused and shared among different radio standards, when we integrate a DVB-T/H receiver into a mobile phone. This paper presents an example implementation of a DVB-T/H receiver on the prototype of Infineon Technologies' Software-Defined Radio (SDR) platform called MuSIC (Multiple SIMD Cores), which is a DSP-centered and accelerator-assisted architecture and aims at battery-powered mass-market handheld terminals.

Copyright © 2009 Yong Jiang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

## 1. Introduction

The DVB-T system was developed and agreed in 1997 by the DVB Project [1]. In the last ten years, DVB-T systems have been successfully deployed not only in Europe but also in the rest of the world. Now the next generation of DVB-T called DVB-T2 is under development. In order to bring the digital video broadcasting service to battery-powered mobile receivers, for example, mobile phone handheld devices, the DVB-T standard is extended to Digital Video Broadcasting-Handheld (DVB-H) with additional features such as 4K mode, in-depth interleaving, time-slicing, and additional forward error correction [2].

Recently, more radio systems, such as GSM, WCDMA, HSPA, GPS, FM radio, Bluetooth, WiFi, and DVB-H have been integrated into mobile handset terminals because these "all in one" mobile terminals provide an "anytime, anywhere" access to information in an easy way for the end user. In fact, the market already requires this kind of terminals and some manufacturers have promptly reacted, for example, Nokia with its N95 and Apple with its iPhone3G. This

creates new challenges to the semiconductor manufacturers, which should keep the stringent requirements for power consumption, silicon area, and time-to-market together with the increasing requirements for throughput and number of coexisting radio standards in a mobile phone terminal. Infineon Technologies provides its innovative SDR platform MuSIC (Multiple SIMD Cores) to meet the requirements. MuSIC is a DSP-centered and accelerator-assisted architecture [3]. The key attraction of the SDR concept is the ability to support multiple standards on the same chip by changing the software only and the feasibility to share the processing resources among several standards in case of a nonsimultaneous execution. In this work, we present an example implementation of a DVB-T/H receiver on the prototype of Infineon Technologies' Software-Defined Radio platform MuSIC which aims at mass-market handset terminals.

The paper is organized as follows. Section 2 provides a system overview of a DVB-T/H receiver showing the main algorithmic functions comprising the baseband processing chain. The architecture of MuSIC and its programming

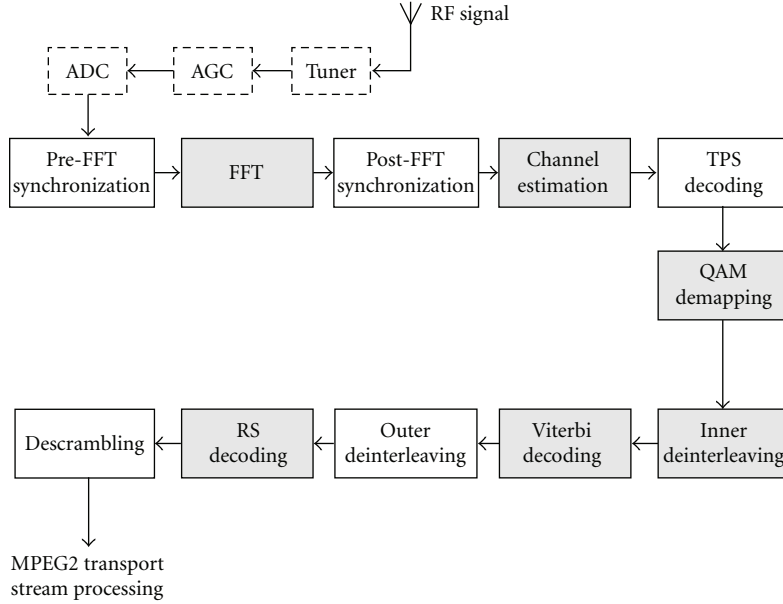


FIGURE 1: Functional block diagram of a DVB-T/H receiver.

model are introduced in Section 3. Section 4 investigates the computational requirements of DVB-T/H and its potential for parallelization on MuSIC. In Section 5, some conclusions and hints for the future work are given.

## 2. DVB-T/H Receiver Algorithms

In this work, we only consider the physical layer processing of the DVB-T/H receiver and omit all analog components, higher layer protocols, and application processing. The functional block diagram in Figure 1 shows a conventional DVB-T/H receiver structure. The RF signal is received by the receiver antenna, downconverted by the tuner circuitry, scaled by an Automatic Gain Control (AGC) circuitry, and then digitized by the Analog-to-Digital Converter (ADC). The baseband processor receives the digitized signal as complex samples from the ADC and delivers the descrambled MPEG transport stream to a higher layer protocol and application processor.

**2.1. Synchronization and Fast Fourier Transformation.** In a typical receiver, a pre-FFT acquisition stage is required to obtain the OFDM symbol timing, the OFDM symbol length, that is, size of the Fast Fourier Transformation (FFT), and the cyclic prefix length, where the latter two parameters are adjustable by the transmitter. The principle of the pre-FFT synchronization is based on the availability of the cyclic prefix in the OFDM symbols [4]. In addition, the carrier frequency offset (CFO) existing between the transmitter and the receiver is also partly estimated in this functional block, that is, only the fractional part of it, and compensated in time domain. Hence, only CFOs which are an integer value of the subcarrier spacing will remain in the OFDM signal. The recovered OFDM symbols are transformed by means

of the FFT, which acts as a matched filter for the OFDM signal. The post-FFT synchronization obtains estimations for the integer part of the CFO and the initial sampling clock frequency offset (SCFO) in frequency domain, which are then compensated in time-domain OFDM signal in order to reduce Inter-Carrier Interference (ICI).

After the acquisition phase, the pre-FFT synchronization is inactive, and the post-FFT synchronization is turned to tracking mode. Due to the instability and drift of the oscillator at receiver the CFO and SCFO will vary during the data receiving phase. Therefore, it is necessary to track the small residual SCFO and CFO to ensure the orthogonality of the OFDM subcarriers and accurate timing after the acquisition phase. The detectors of the residual CFO and SCFO are based on the temporal correlation of OFDM signal in frequency domain. For the investigated DVB-T/H receiver, the continual pilot signals are used to estimate the residual SCFO and CFO. The principle of the tracking methods is described in [5]. Figure 2 shows an example behavior of the implemented SCFO tracking algorithm under additive white Gaussian noise (AWGN) channel with 4 dB channel SNR and an SCFO of 30 ppm. Our simulations show that in the tracking mode, the residual SCFO and CFO, depending on the reception channel conditions, remain very small after a number of OFDM symbols.

A more detailed analysis of the synchronization strategies for an OFDM-based receiver is provided in [6]. It should be noticed that most of the synchronization tasks are only active during the acquisition phase. Because of the use of time-slicing in DVB-H it is more necessary to minimize the whole synchronization time.

**2.2. Channel Estimation, Equalization, and TPS Decoding.** After FFT the impairment of the transmission channel, especially the Doppler frequency drift by high-speed mobility of

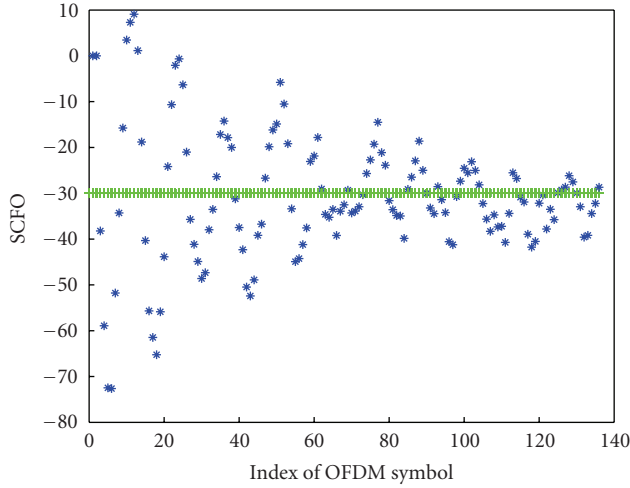


FIGURE 2: Typical behavior of the implemented SCFO tracking algorithm for SCFO = 30 ppm. (“+” = actual SCFO, “\*” = SCFO estimated by the tracking algorithm).

the handheld terminal, will be mitigated through the channel equalization which is characterized as

$$X_{l,k} = \frac{Y_{l,k}}{H_{l,k}}, \quad (1)$$

where  $k$  is the subcarrier index in the OFDM symbols, and  $l$  is the OFDM symbol index in the OFDM frame.  $X$  and  $Y$  are the equalized and received OFDM symbols, respectively.  $H_{l,k}$  are the coefficients of the channel transfer function (CTF). In order to estimate the coefficients  $H_{l,k}$ , two cascaded interpolation filters are used as proposed in [7].

The principle of the channel estimation is depicted in Figure 3. The value of  $X_{l,k}$  and  $Y_{l,k}$  corresponding to the scattered pilots is known beforehand at the receiver. Therefore, at a first stage the CTF coefficients  $H_{l,k}$  for the scattered pilots can be readily estimated using (1). For subcarriers sharing the same index  $k_0$ , a 6-tap time-direction interpolation filter is used in which the six closest scattered pilots with the same index  $k_0$  are used as supporting points, thereby obtaining the coefficients  $H'_{l,k_0}$ . Afterwards, both  $H$  and  $H'$  are used as the supporting points in a 12-tap frequency-direction filter to calculate the CTF for the remaining subcarriers. Thus, those positions for which a coefficient  $H'$  is computed can be seen as “pseudo” pilots, which are marked as  $\otimes$  in Figure 3. The interpolation filters use the scattered pilots in each OFDM symbol as a priori knowledge. Therefore, the scattered pilots have to be determined before the channel estimation can be started.

In a regular DVB-T receiver the exact positions of the scattered pilots are determined after decoding the transmission parameter signaling (TPS) bits. The TPS bits also provide information about other system parameters required to demodulate and decode the received signal. They are organized into blocks of 68 bits, and each bit is transmitted on several subcarriers within one OFDM symbol. Hence, 68 symbols, that is, one OFDM frame in DVB-T/H, are required to transmit a whole TPS block. The TPS bits are modulated

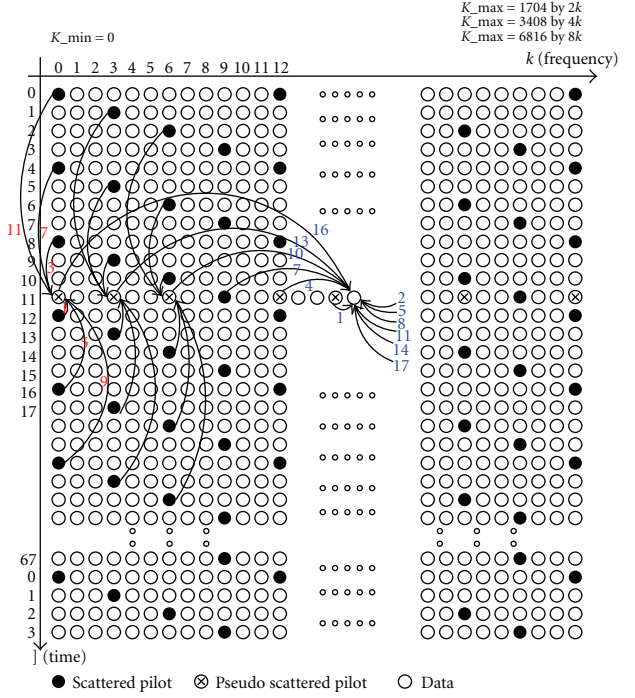


FIGURE 3: DVB-T/H frame structure and illustration of the channel estimation principle.

using Differential Binary Phase Shift Keying (DBPSK) in the time direction. This allows recovery of the bits without channel equalization.

The OFDM frame synchronization using TPS bits implies a synchronization time ranging from 17 up to 84 OFDM symbols depending on the very first received OFDM symbol in the frame. However, such a long synchronization time is not acceptable in a DVB-H receiver due to time slicing, where the receiver is allowed to power off when it is inactive leading to significant power savings. To solve this problem a fast scattered pilot synchronization scheme has been proposed in [8], which needs only one OFDM symbol to identify the scattered pilot and is based on the temporally repetitive structure of the scattered pilots. This method also makes it possible to execute the channel estimation and equalization prior to TPS bit decoding, thereby improving the robustness of the TPS decoding itself under extremely bad transmission conditions [9]. After DBPSK decoding of the TPS bits, the frame synchronization is required to determine the position of the respective TPS bit in one TPS block, as the semantics of each TPS bit defined in [10] depends on its position in one TPS block.

**2.3. Demapping.** With the decoded TPS signal the constellation parameters used by the transmitter for the QAM Gray-mapping are made available to the receiver. The QAM demapper can then recover data bits from the data subcarriers, that is, one QAM symbol demapped onto one  $v$ -bit binary word according to Gray-mapping and the constellation parameters, where  $v$  equals 2 for QPSK, 4 for 16-QAM, and 6 for 64-QAM. Soft decision must be

available to the Viterbi decoder. It improves the error-correction capability and makes the correct interpretation of depunctured bits possible [11]. Every bit of the  $v$ -bit binary word is represented as one  $n$ -bit binary word to represent the signed distance between the constellation point and the decision border, called softbits. A higher resolution of the soft decision increases not only the gain of the Viterbi decoder but also the implementation complexity. A tradeoff between performance and implementation complexity has to be found by means of simulation. Our simulation results show that a 5-bit resolution is a proper choice.

**2.4. Inner Deinterleaving and Viterbi Decoding.** In order to improve the long burst error correction capability, several interleaving stages are specified in the DVB-T/H standard. Because of the bitwise sequent address generation algorithm and the large range of the symbol deinterleaving (e.g., the interleaving over 6048 QAM symbols by 8k mode), a software implementation for the deinterleaving in SIMD core is not efficient. To avoid the sequent address generation a lookup table can be used, but it will be very large for on-chip memory. The inner deinterleaving can be done better with a scalar processor or in hardware accelerator. Following the deinterleaver, a Viterbi decoder with soft decision is used to perform convolutional decoding.

**2.5. Outer Deinterleaving, Reed-Solomon Decoding, and Descrambling.** The bit stream obtained after the convolutional decoding is reorganized as a byte stream and further processed by the outer deinterleaver, thereby improving the long burst correction capabilities of the receiver. Afterwards, the byte stream is processed by Reed-Solomon decoder and descrambled at byte level.

The functions marked in gray in Figure 1 are core functions demanding most of the computing power and storage capacity and are therefore the critical functions for a software implementation on an SDR platform. We have analyzed their computational requirements and investigated the possibilities for a parallel realization on the MuSIC platform, as it will be discussed in Section 4. To achieve best performance under various network conditions and coverage scenarios, the DVB-T/H standard provides the network operators numerous system parameters which are listed in Table 1. The combinations of these parameters derive a net bit rate from 3.11 Mbit/s at 5 MHz channel bandwidth, QPSK, 1/2 code rate, 1/4 guard interval, up to 31.67 Mbit/s at 8 MHz channel bandwidth, 64-QAM, 7/8 code rate, 1/32 guard interval. The receiver must comply with all the possible combinations of parameters the transmitter may use. At this point it is reasonable to analyze only the worst case scenario, that is, a transmission at 31.67 Mbit/s.

The whole DVB-T/H receiver described has a good system performance. Our simulation results show that for AWGN channel with a channel SNR of more than 4 dB, or for typical urban channel with vehicular speed of 6 km/h (TU6) and a channel SNR of more than 10 dB, user data can be decoded with a bit error rate smaller than  $10^{-3}$ .

TABLE 1: DVB-T/H system parameters (see [10]).

Elementary period $T$	7/64, 1/8, 7/48, 7/40 [ $\mu$ s]
Channel bandwidth $B$	8, 7, 6, 5 [MHz]
Number of carrier $N$	8k, 4k, 2k
Number of guard $N_g$	1/32, 1/16, 1/8, 1/4 [in $N$ ]
Number of used carrier $N_u$	6817, 3408, 1705
Number of data carrier $N_d$	6048, 3024, 1512
Code rate $r$	7/8, 5/6, 3/4, 2/3, 1/2
Symbol duration $T_u$	$T_u = T \cdot N$
Carrier spacing $\Delta f$	$1/T_u$
QAM constellation	64, 16, 4

### 3. Architecture and Programming Model

The intensive work carried out at Infineon Technologies resulted in a versatile processor architecture which is able to cope with the performance, power, and area requirements of a multistandard SDR approach [12, 13]. This processor mainly consists of a cluster of four single-instruction multiple-data (SIMD) DSP cores (see Figure 4). Each SIMD core contains four processing elements (PEs) and operates with a clock frequency of 300 MHz. To relax the timing requirements for the memory and to resolve pipeline hazards, each core runs four threads which are switched by a fixed time multiplexing mechanism. This is equivalent to running 16 threads at 75 MHz each.

Long instruction words (LIWs) of the PE array show memory, arithmetic, and communication components. The SIMD core controller is in fact a 32-bit general purpose processor (GP). The GP communicates with the other units via instruction and data FIFOs.

The cluster of the SIMD cores is accompanied by dedicated configurable hardware accelerators for coding/decoding and for filtering operations. In addition, there is an ARM processor for the execution of the protocol stacks. For a more detailed discussion about the MuSIC processor, we refer to [14–16].

The programming model for this architecture is the multithreaded programming in C. Wrapped into functions called by threads, the purely data parallel parts (associated with SIMD cores) are programmed in a data parallel language extension of C.

To support multithreaded programming, the Infineon Lightweight Operating System (ILTOS) has been developed. It provides the means to create and synchronize threads to asynchronously send and query messages between them and to allocate and free shared memory.

Functions to be executed on an SIMD core are written in Data Parallel C Extension (DPCE) language, a superset of the C language [17]. DPCE offers parallel data types and operations on them. A compiler which takes a DPCE source and produces synchronized C code for GP core and DMA transfers (to be translated further by a C compiler for the GP) as well as PE assembly was developed at Infineon. This compiler is not yet optimized, though. To achieve best performance, inline assembly is used for the PE array and



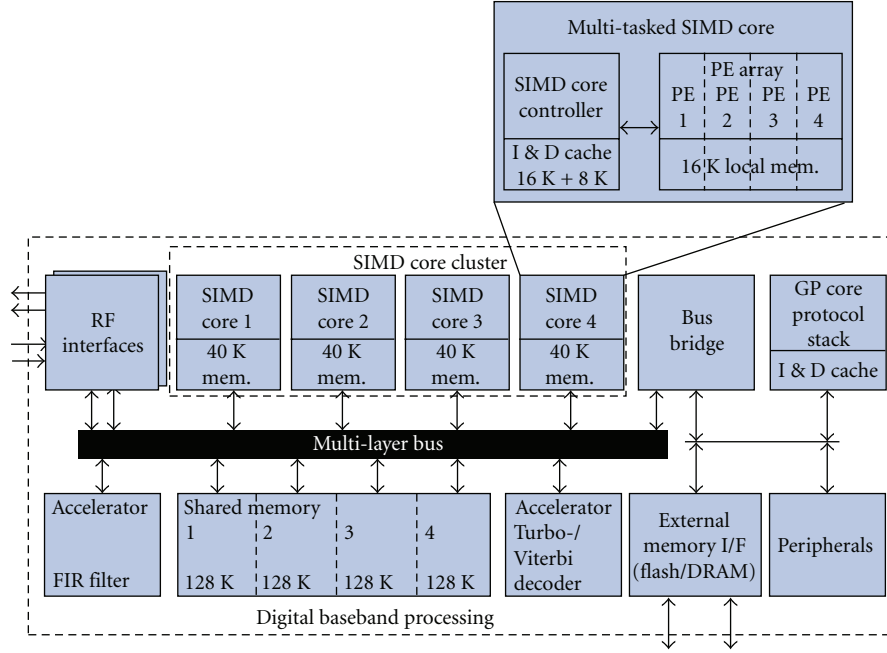


FIGURE 4: The SDR baseband platform MuSIC.

explicit DMA configuration. What remains is mainly the C language with some intrinsic functions for PE and DMA control plus an assembly source code library for the PE. Implementations can be done completely without PE and DMA by writing pure C programs. These will then run on the GP core alone. This feature is of importance for testing assembly implementations.

Moreover, a virtual prototype of the entire MuSIC platform based on SystemC has been developed at Infineon. The virtual prototype is a cycle- and bit-accurate software-based simulator. It contains models of all processors, accelerators, busses, memories, and peripherals which will be available in the real hardware. The same software can be run on both the virtual prototype and the real hardware.

#### 4. Computational Requirements for an Implementation on MuSIC

**4.1. Fast Fourier Transformation (FFT).** FFT is the core function of an OFDM receiver. The most commonly used algorithm for FFT calculation is the well-known butterfly algorithm. The computational requirement of FFT is very high, especially for the large FFT block which in DVB-T/H can be 8192 complex samples. The theoretical computational complexity of FFT with  $N$  complex samples based on radix-2 algorithms is given as follows:

$$\begin{aligned} \frac{N}{2} \cdot \log_2 N, & \text{ complex multiplications,} \\ N \cdot \log_2 N, & \text{ complex additions.} \end{aligned} \quad (2)$$

In the case  $N$  equals 8192, 4 instructions are needed for 1 complex multiplication and 2 instructions are needed for

1 complex addition, it leads to 425984 instructions for 8k mode based on radix-2 without data level parallelism. In the ideal parallel case of 4 data paths, that is, 4 PEs in an SIMD core, the cycle count can be reduced from 425984 to 106496. An example implementation of 8k FFT on MuSIC was measured out. It shows that about 20 percent overhead in cycles is needed to overcome all data transfer and temporary storage. This shows the MuSIC architecture fully exploits the data level parallelism of the FFT algorithms.

**4.2. Channel Estimation and Equalization.** Section 2.2 describes the channel estimation in our DVB-T/H receiver. Its computational complexity is analyzed here. The first step, that is, the interpolation in time direction, is carried out for those subcarriers  $k_0 = 0, 3, 6, \dots$  for which a scattered pilot is available (see Figure 3). It can be described as follows:

$$\begin{aligned} H'_{l,0} &= w_1(1)H_{l-11,0} + w_1(2)H_{l-7,0} + \dots + w_1(6)H_{l+9,0}, \\ H'_{l,3} &= w_2(1)H_{l-10,3} + w_2(2)H_{l-6,3} + \dots + w_2(6)H_{l+10,3}, \\ H'_{l,6} &= w_3(1)H_{l-9,6} + w_3(2)H_{l-5,6} + \dots + w_3(6)H_{l+11,6}. \end{aligned} \quad (3)$$

The second step is based on the scattered pilots  $H$  and the estimations  $H'$  computed in time direction within one OFDM symbol, namely

$$\begin{aligned} H''_k &= v_1(1)H_{k-16} + v_1(2)H_{k-13} + \dots + v_1(12)H_{k+17}, \\ H''_{k+1} &= v_2(1)H_{k-17} + v_2(2)H_{k-14} + \dots + v_2(12)H_{k+16}, \end{aligned} \quad (4)$$

where  $w_n(m)$  and  $v_n(m)$  are the filter coefficients and depend on the distance between the supporting point and the

TABLE 2: Computational requirements.

Functional unit	MOPS
FFT	461
Channel estimation and correction	155
Soft demapping and quantization	170
Viterbi decoding	4868*
RS decoding	24*
Total	5678 (786**)

The worst case is considered here, that is, 8 MHz channel bandwidth, 8k mode, 1/32 guard interval, 64 QAM, 7/8 code rate, and 31.67 Mbit/s.

\*Based on [18] and scaled according to the system parameters.

\*\*Without Viterbi and RS decoding.

estimated point, which is also shown in Figure 3. Therefore, it needs  $1 \times 2 = 2$  real multiplications (RMs) to calculate  $H$ . There are 568 scattered pilots in one 8k OFDM symbol. For each pseudopilot  $H'$ , 12 real multiplication-accumulations (rMACs) are required. There are 1705 pseudopilots in one 8k OFDM symbol. 24 rMACs are required for each of the remaining sub-carriers to calculate its CTF coefficient  $H''$ . There are 4544 such subcarriers in one 8k OFDM symbol. The channel correction requires 2 RMs for every payload subcarrier. Because split of the TPS and continual pilots from the payload subcarriers by means of software is even more expensive than the equalization for the TPS and continual pilots, we equalize all the subcarriers excluding the scattered pilots, that is, 6249 subcarriers need to be corrected.

Together it needs 13634 RMs and 129516 rMACs per OFDM symbol in 8k mode for the channel estimation and correction, which equals 143150 instructions without parallel processing. Because the time direction interpolation is not causal, FIFO buffering is required to delay the input OFDM symbols. In the case of the 6-tap time direction filter, all subcarriers  $Y$  within 12 OFDM symbols and the CTFs  $H$  at all scattered pilot positions within 23 OFDM symbols need to be stored, which leads to the most memory demand of the DVB-T/H receiver.

**4.3. Demapping.** The Gray mapping is used in DVB-T/H. The demodulation of one subcarrier, in the case of 64-QAM, needs 8 operations. The quantization of one soft decision needs 3 operations and in the case of 64-QAM, each payload subcarrier implies 6 soft-decisions. So  $6048 \times (8 + 6 \times 3) = 157248$  operations are needed for the demapping and quantization of one OFDM symbol.

The computational complexity of each function is listed in Table 2 as required million operations per second (MOPS), where one OFDM symbol duration is 924 microseconds, which determines the real-time processing requirement. The Viterbi and Reed-Solomon decoders demand enormous computing power and are therefore implemented as hardware accelerators. In this manner the computational requirements can be reduced from formerly 5678 MOPS down to 786 MOPS. For 4 data paths in data-level parallelism the real-time processing requirement is reduced to 197 million cycles per second. With use of thread level parallelism, this computing requirement can be

affordably met with 3 of 16 threads at 75 MHz on MuSIC in the case of about 15 percent implementation overhead and can be sufficiently met with 4 threads, that is, one SIMD core, in the case of about 50 percent implementation overhead.

## 5. Conclusions

In this paper, we first analyzed the example algorithms of a DVB-T/H receiver in detail, and then gave a brief introduction on the architecture and programming model of the prototype of Infineon SDR platform MuSIC. Based on the algorithms and the hardware architecture, and based on our implementation on MuSIC, we estimated and partly measured the computational requirements of the relevant functions for a DVB-T/H receiver on MuSIC. The results show that it is feasible to implement a DVB-T/H receiver on MuSIC with one of the four SIMD cores. It should be noted that the control functions for the respective functional blocks have not been considered till now. This will be studied in the future work.

## Acknowledgments

The authors gratefully acknowledge fruitful discussions with Mirko Sauermann, Mathias Richter, Dominik Langen, Reinhard Rueckriem, and Professor Ulrich Ramacher. This work has been supported by the German BMBF (Bundesministerium für Bildung und Forschung) project MxMobile.

## References

- [1] "History of the DVB Project," <http://www.dvb.org/>.
- [2] ETSI, EN 302 304 v1.1.1, "Digital Video Broadcasting (DVB); Transmission System for Handheld Terminals (DVB-H)," November 2004.
- [3] U. Ramacher, "Software-defined radio prospects for multi-standard mobile phones," *Computer*, vol. 40, no. 10, pp. 62–69, 2007.
- [4] J.-J. van de Beek, M. Sandell, and P. O. Börjesson, "ML estimation of time and frequency offset in OFDM systems," *IEEE Transactions on Signal Processing*, vol. 45, no. 7, pp. 1800–1805, 1997.
- [5] M. Speth, S. Fechtel, G. Fock, and H. Meyr, "Optimum receiver design for OFDM-based broadband transmission—part II: a case study," *IEEE Transactions on Communications*, vol. 49, no. 4, pp. 571–578, 2001.
- [6] M. Speth, S. A. Fechtel, G. Fock, and H. Meyr, "Optimum receiver design for wireless broad-band systems using OFDM—part I," *IEEE Transactions on Communications*, vol. 47, no. 11, pp. 1668–1677, 1999.
- [7] P. Hoher, S. Kaiser, and P. Robertson, "Two-dimensional pilot-symbol-aided channel estimation by Wiener filtering," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '97)*, vol. 3, pp. 1845–1848, Munich, Germany, April 1997.
- [8] L. Schwoerer, "Fast pilot synchronization schemes for DVB-H," in *Proceedings of the 4th IASTED International Multi-Conference on Wireless and Optical Communications*, pp. 420–424, Banff, Canada, July 2004.

- [9] H. Ye, "TPS decoder in an orthogonal frequency division multiplexing receiver," US patent no. 7123669, 2006.
- [10] ETSI, EN 300744 V1.5.1, "Digital Video Broadcasting (DVB); Framing Structure, Channel Coding and Modulation for Digital Terrestrial Television," November 2004.
- [11] U. Reimers, *DVB the Family of International Standards for Digital Video Broadcasting*, Springer, Berlin, Germany, 2005.
- [12] C. Grassmann, M. Richter, and M. Sauermann, "Mapping the physical layer of radio standards to multiprocessor architectures," in *Proceedings of the Conference on Design, Automation and Test in Europe (DATE '07)*, pp. 1412–1417, Nice, France, April 2007.
- [13] H.-M. Blüthgen, C. Sauer, M. Gries, et al., "Finding the optimum partitioning for multi-standard radio systems," in *Proceedings of the Software Defined Radio Technical Conference (SDR '05)*, pp. 1–6, Orange County, Calif, USA, November 2005.
- [14] W. Raab, H.-M. Blüthgen, and U. Ramacher, "A low-power memory hierarchy for a fully programmable baseband processor," in *Proceedings of the 3rd Workshop on Memory Performance Issues (WMPI '04)*, pp. 102–106, Munich, Germany, June 2004.
- [15] H.-M. Blüthgen, C. Grassmann, W. Raab, U. Ramacher, and J. Hausner, "A programmable baseband platform for software-defined radio," in *Proceedings of the Software Defined Radio Technical Conference (SDR '04)*, Phoenix, Ariz, USA, November 2004.
- [16] H.-M. Blüthgen, C. Grassmann, and U. Ramacher, "A software programmable multiple-standard radio platform," in *Proceedings of the 14th IST Mobile & Wireless Communications Summit*, pp. 1–5, Dresden, Germany, June 2005.
- [17] "Data Parallel C Extensions (DPCE)," <http://www.crescentbaysoftware.com/dpce/index.html>.
- [18] M. Hosemann, G. Cichon, P. Robelly, et al., "Implementing a receiver for terrestrial digital video broadcasting in software on an application-specific DSP," in *Proceedings of IEEE Workshop on Signal Processing Systems (SIPS '04)*, pp. 53–58, Austin, Tex, USA, October 2004.