

## **Implementation of DAB Channel and Source Decoders on ARM Cortex-A8 Platform**

Swetha R<sup>1</sup>, Sri P. Nagaraju<sup>2</sup>, Mr. Rahul Ramachandran<sup>3</sup>

<sup>1</sup>PG student, Digital Communications, Department of Telecommunications, R V College of Engineering, Bangalore - 59. swetsraj@gmail.com

<sup>2</sup>Associate Professor, Department of Telecommunications, R V College of Engineering, Bangalore - 59. nagarajup@rvce.edu.in

<sup>3</sup>Technical Specialist, RBEI/ECF4, Bangalore, Rahul.Ramachandran@in.bosch.com.

---

**Abstract**— The Digital Audio Broadcasting (DAB) was developed by the European Eureka-147 project. DAB system is used especially for mobile reception. Until now, most radios are supporting Hardware IP's for receiving DAB radio standards. Recently, there has been effort in realizing the receiver functionality in software. The present document defines the DAB receiver design which includes channel and source decoding.

The implementation of all the modules of DAB channel decoder is done in software, by taking MATLAB simulation as reference. DAB Baseband decoder is implemented in C-code. Porting of C code on ARM Cortex A8 core is done using ARM DS-5 IDE. Profiling is done for all the modules of channel and source decoder connected to the target platform. Cycle intensive modules are thus optimized using NEON coprocessor intrinsic in ARM cortex-A8 platform. The output of source decoder should not have any audible artifacts. The output of optimized baseband decoder is compared for bit exact with the generic implementation. Baseband filter Module achieved optimization with overall ~57% reduction, FFT module of ~20% reduction achieved and Viterbi optimization resulting in ~69% reduction. These optimizations have brought sufficient speed up improvement by reducing the number of cycles of execution.

---

**Keywords**- ARM cortex-A8, Baseband Filter, FFT, DAB Channel Decoding and demodulation, Error correction, MPEG layer II, NEON Intrinsic, profiling, Viterbi decoding.

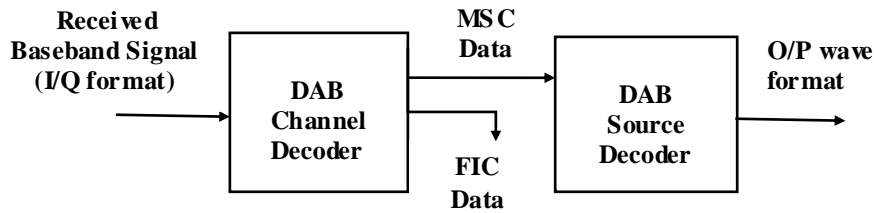
### **I. INTRODUCTION**

Digital audio broadcasting (DAB) is the digital radio technology for broadcasting radio stations used in several countries, particularly in Europe. Over a specific spectrum it offers many radio programmes than a FM radio. Compared to FM radio, DAB is more robust with regard to noise and multipath fading for mobile reception. A detailed description of the DAB system can be found in ETSI EN 300 401 [1].

Modern radio receivers for car and home entertainment contain more and more software. This enables radio maker to support different radio standards on the same hardware platform. Also updates or extensions become possible even in a very last stage of the development process. With state of the art signal processors, it is also possible to perform the baseband processing of the DAB signal on a software platform, which gave rise to SBR (Software Based Receiver). This also makes the Radio standards portable to multiple platforms.

The DAB Baseband Decoder modules have to be developed in C-Code and validate the functionality using reference DAB Baseband signals. After this phase, the C-Code will be ported to the standard high performance embedded core ARM cortex A-8 processor with NEON extension. The next step is to optimize the C-code for a ARM cortex A-8 processor. ARM cortex A-8 has neon extension which has vector processing and has multi parallel processing unit.

Figure 1 shows the block diagram of DAB baseband Decoder which consists of channel decoder and source decoder. The input to the channel decoder is the DAB received baseband signal which is in I/Q format. The channel decoder outputs two types of data i.e., FIC and MSC data.



*Figure 1. DAB Baseband decoder*

The general transport mechanisms used in the DAB system for transmission of Digital audio programs and data services are FIC and the MSC. The Fast Information Blocks (FIB) constitutes to form FIC, carries the necessary control information to interpret the MSC configuration. Multiplex Configuration Information (MCI) is the essential part of this control information, which contain multiplex structure information and its reconfiguration when necessary. Other types of information that can be included in FIC are Service Information (SI), the conditional access (CA) management information and Fast Information Data Channel (FIDC). In order to allow a rapid and safe response to the MCI, the FIC is transmitted without time interleaving and protection level is high against transmission errors.

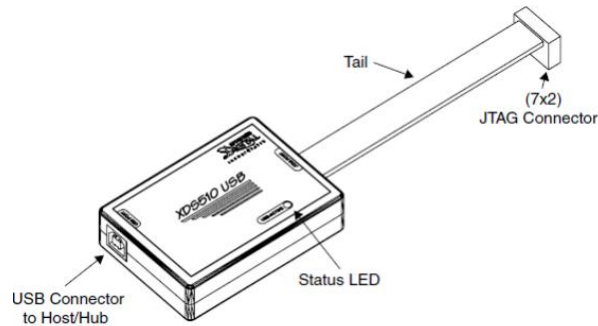
The MSC is made up of a sequence of common interleaved frames (CIF). The MSC consists of multiplex of sub-channels. In the MSC, two transport modes are defined for service components. They are stream mode and the packet mode. In a given sub channel, at a fixed bit rate, the stream mode provides a transparent transmission from source to destination. The Conveying of several data service components into a single sub-channel is done in packet mode. Each sub-channel may carry one or more service components.

The source decoder used here is MPEG layer II also called as MP2. The output of source decoder will be in wave format. We have chosen a high-end ARM Cortex-A8 processor as target implementation platform for the implementation of the DAB Baseband Decoder. The reasons are it has the ability to scale in speed from 600MHz to greater than 1GHz, provides high performance, has superscalar micro architecture. The processor supports NEON technology for multi-media and SIMD processing and binary compatibility with ARM926, ARM1136, and ARM1176 Processors.

## **II. HARDWARE AND SOFTWARE REQUIREMENTS**

The hardware components required are:

- A. **Target SoC board:** The target board consists of ARM Cortex-A8 and digital signal processor along with the memory unit, external memory, peripherals and tuners.
- B. **Spectrum Digital XDS510 JTAG USB Emulator:** The XDS510 USB JTAG Emulator is designed to be used with digital signal processors (DSPs) and microprocessors which operate with +3.3 or +5 volt levels on the JTAG interface. This emulator is powered from USB line which means from the target system, no power is drawn.



**Figure 2. XDS510 JTAG Emulator**

Figure 2 shows the XDS510 USB. The key items identified are Status LEDs, JTAG connector, Tail and USB connector to the host PC or hub. It is Compatible with Texas Instruments Code Composer Studio.

The software components required are as follows:

A. Microsoft Visual C++ 2008 Express Edition:

Microsoft Visual C++ is implementation of the C and C++ compiler by Microsoft and associated languages-services and specific tools for integration with the Visual Studio IDE. It can compile either in C mode or C++ mode. It also supports the use of intrinsic functions, which are functions recognized by the compiler itself and not implemented as a library. Intrinsic functions are used to expose the SSE instruction set of modern CPUs.

B. Eclipse for ARM DS5:

It is an Integrated Development Environment (IDE) that combines the Eclipse IDE with the compilation and debug technology of the ARM tools. Eclipse for DS5 provides :Project Manager which enables to perform project tasks such as adding or removing files and importing, exporting, or creating projects and managing build options. Read, write or modify C/ C++ or ARM assembly language source files are enabled in Editors. DS-5 uses the C/C++ and DS-5 Debug perspectives.

DS-5 supports the NEON development. It is advanced SIMD (Single Instruction, Multiple Data) processing unit. This means that it can apply a single type of instruction to many pieces of data at one time in parallel. This is extremely helpful when it comes to media processing such as audio/video filters and codecs.

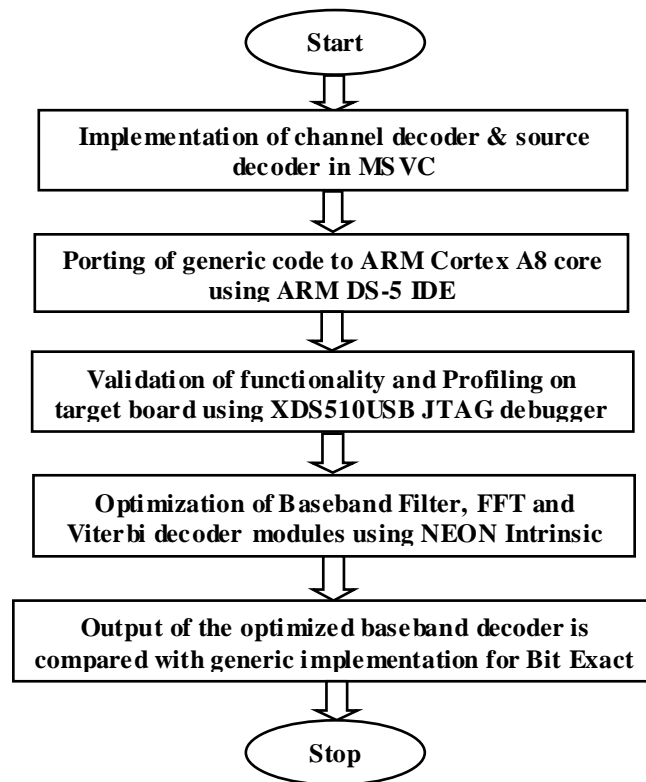
### **III. IMPLEMENTATION**

The Implementation of DAB Channel Decoder mainly consists of designing of three modules, which are baseband filter, Fast Fourier transform and Viterbi decoder module. The following sections give the complete implementation details.

#### **3.1 Implementation flowchart**

The Implementation flow is as follows: The first step is to study MSVC coding and ARM cortex A-8 with NEON extension. Next step is implementation of DAB channel decoder in software, by taking MATLAB simulation as reference. Next step is to implement the DAB baseband decoder in

generic platform-MSVC and Porting of generic code to ARM Cortex A8 core using ARM DS-5 IDE.



*Figure 3. Flowchart for Implementation*

The functionality is validated and Profiling is done on target board using XDS510USB JTAG debugger. The cycle intensive modules are determined and performance optimization of cycle intensive modules is carried out using NEON Intrinsic. Once the optimization of baseband filter, FFT and viterbi module is done, the final step is to validate the outputs of MSC, FIC and source decoder using reference DAB Baseband signals i.e., output of the optimized baseband decoder is compared with generic implementation for bit exact.

### **3.2 Implementation Of DAB Channel Decoder**

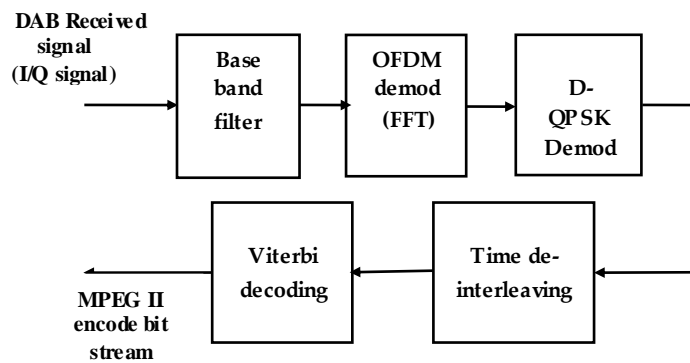
The DAB channel decoder implementation consists of implementing of main modules like Baseband Filter, D-QPSK demodulation, OFDM demodulation (FFT), time deinterleaving and viterbi decoding.

The DAB received baseband signal is fed to the channel decoder. The outputs of channel decoder are FIC and MSC. The MSC is fed to the source decoder and FIC is given to the user interface for service control. The channel decoder shall be built of signal processing modules as depicted in the figure 4.

The Input is complex DAB baseband signal @ 2.048 MHz, 16 bit resolution and the Outputs are FIC and MSC sub-channels data streams (MSC contains the encoded MP2 Frame). A typical portion of DAB baseband processing is one OFDM symbol. In DAB transmission mode I, this comprises 1.246 ms with 2552 complex baseband samples @ 2.048 MHz sampling frequency.

A possible next higher framing in DAB signal processing is the DAB transmission frame with a magnitude two times higher than an OFDM symbol: In DAB transmission mode I, this comprises 96 ms with 196608 complex baseband samples @ 2.048 MHz.

In the three other DAB transmission modes II, III and IV, the transmission frame length is 24 ms, 24 ms and 48 ms, respectively. For a constant calling scheme within the runtime environment, the 96 ms period can be applied for these modes, too. As a consequence, more than one transmission frame (two or four) is contained in one baseband signal buffer.



*Figure 4. DAB Channel Decoder*

**Baseband Filter:** The baseband filter performs additional channel filtering to the signal (esp. adjacent channel filtering). FIR filtering based on 33 filter coefficients is used as the building block for baseband filter. The filter is 0 dB filter. The filtering computation is implemented using the circular buffer access of the delay line. This is meant for optimized implementation in any DSP platform. The input for this module is baseband I/Q signals. The in phase and quadrature samples are computed separately and FIR filter is called separately.

The BBF module is implemented in C using MSVC and in ARM DS5 as well using Neon intrinsic for optimization. Both the outputs are compared.

**OFDM Demodulation :** At the receiver after symbol timing synchronization has been accomplished and after detection of the null symbol and the phase reference symbol guard interval samples, the output array of the synchronization block contain a data signal which comprises of OFDM symbol samples. The OFDM symbols from synchronized data array is demodulated using the OFDM demodulator block. It is the main block in the receiver side.

In DAB spectrum, the complex amplitudes of the carriers is calculated by applying FFTs to perform the demodulation of the OFDM symbols. The information of the modulated data by means of a DQPSK modulation is obtained in these amplitudes. The FFT length's varies from 256, 512, 1024 and 2048 according to the various DAB transmission modes I-IV. A Radix-2 FFT with decimation in time is implemented so that the specified DAB transmission modes can be realized by a simple control of the FFT memory addressing. An Automatic frequency Control (AFC) is necessary to cope with the frequency drift of the baseband signal for which a new approach has been chosen. In this module cyclic prefix and zero padding removal is done. To transform the OFDM symbol block back to frequency domain, the FFT block performs the FFT operation on every OFDM symbol i.e., on I/Q interleaved baseband outputs.

The N-point DFT can be regarded as a combination of N/2-point DFT of the even numbered time-domain samples  $x[2m]$  and N/2-point DFT of the odd-numbered time domain sample  $x[2m+$

1] multiplied by  $W_N^k$  as shown in Figure 5(a). In addition, the basic arithmetic module called the 'butterfly' operation, with one addition and one subtraction, is indicated. Similarly, one (N/2)-point DFT can be further decomposed into two (N/4)-point DFTs and so on, until 2-point DFTs. If  $N = 2^v$ , the decomposition procedure can be applied v times. Figure 5(b) shows the signal flow graph of an 8-point radix-2 decimation-in-time FFT algorithm. The time-domain samples are not arranged in normal order, but in a bit-reversed-addressing order.

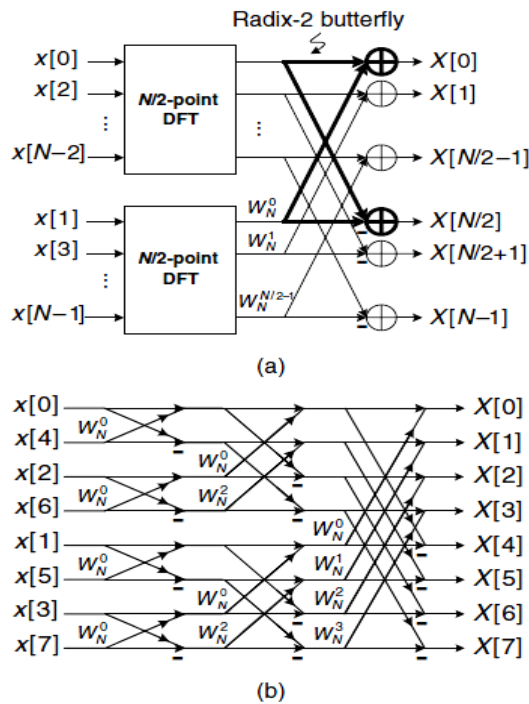


Figure 5. (a) Signal flow graph of radix-2 decimation-in-time decomposition  
 (b) Its application in 8-point DFT computation.

The FFT module is implemented in C using MSVC and in ARM DS5 as well using Neon intrinsic for optimization. Here for the transmission mode I, the N point FFT used is 2048. Both the outputs are compared for validation.

**Demodulation:** The D-QPSK demodulator unit performs the differential carrier demodulation and QPSK de-mapping and hence produces bit metrics.

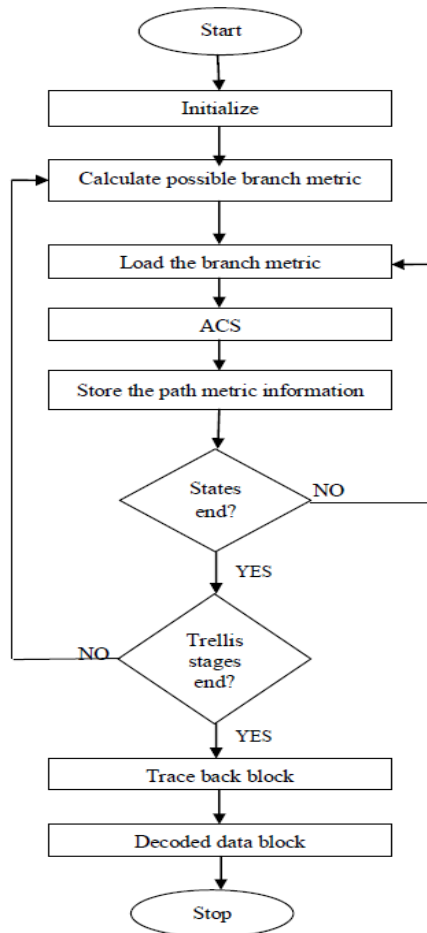
**Time deinterleaving:** MSC sub-channels are de-interleaved in time domain using the TDI.

**Viterbi Decoding:** The Viterbi decoder performs the decoding of the convolutional-coded data streams, including energy descrambling and depuncturing. (This is the inner FEC in the system)

As shown in Figure 6, the Branch Metric Unit (BMU) first receives the noisy symbols and computes the corresponding branch metrics. Next, the Add-Compare-Select (ACS) unit adds the branch metrics to old state metrics, and updates the metrics for all states by making proper comparison. The function of the Trace-Back Unit (TBU) is to search the best survivor path based on the accumulated decision results generated by ACS unit (ACSU).

Trace back and decode unit consists of following steps:  
 Step1: find the survivor paths for N + D input pairs of Bits.

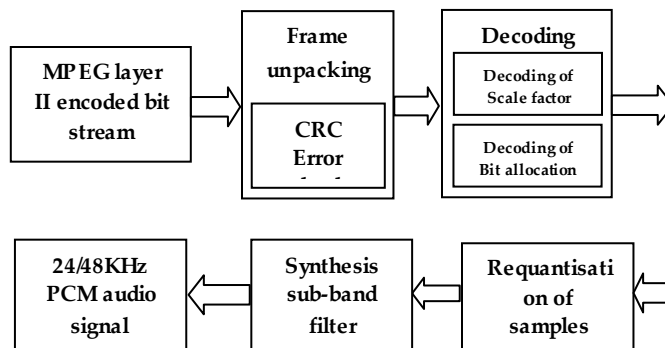
- Step2: Traceback from end of any survivor paths to the beginning.
  - Step3: Send N bits to the output
  - Step4: find the survivor paths for another N pairs of input bits
  - Step5: go to step2.
- Where N is bits sending to output after each trace back  
 D is block of metrics (data)



*Figure 6. Flow Chart of Viterbi decoder*

### 3.3 Implementation of DAB Source decoder

Figure 7 shows the block diagram of the standard MPEG audio decoder. The decoder parses this bit-stream and restores the quantized sub band values, which are used to reconstruct the PCM samples using a synthesis filter bank.



*Figure 7. Block diagram of DAB source decoder*

The input to the source decoder is the Encoded MP2 Stream from DAB Channel Decoder Output (DAB Audio Frame) and the Output is 24/48 KHz PCM Audio. It comprises all signal processing blocks for decoding a MPEG1 Layer II and MPEG2 Layer II compressed data stream into 24/48 KHz PCM Audio Signal. MPEG Audio utilizes psychoacoustics to significantly reduce the data rate required by an audio stream. It reduces or completely discards certain parts of the audio that the human ear can't hear, either because they are in frequencies where the ear has limited sensitivity, or are masked by other (typically louder) sounds

MPEG Audio is divided into 3 layers. Each higher layer is more computationally complex, and generally more efficient at lower bitrates than the previous. The layers are backward compatible as higher layers reuse technologies implemented by the lower layers. The DAB Receiver standard uses Layer-II Decoders such as MPEG-1 Audio Layer II or MPEG-2 Audio Layer II (MP2, sometimes called Musicam or MUSICAM).

Description of the blocks:

**Frame Unpacking:** Unpacks the data of the frame to recover the various elements of information. This also does a CRC error check of the bit stream.

**Decoding:** Decodes the Scale factors and Bit allocation information.

**Reconstruction of Samples:** Reconstructs the quantized sub-band samples.

**Synthesis Sub-band Filter:** Transforms the sub-band samples back to produce digital PCM audio signals.

**Optimization:** When profiling is done, we got to know that few modules like baseband filter, FFT and viterbi modules consumes more number of cycles for execution. So therefore these modules has been complemented by high level optimization which are mainly concentrated to further reduce execution time of cycles spent on arithmetic computations. In addition some memory oriented transformations like loop unrolling is done which make use of single instruction multiple data concept (SIMD). The C code of baseband filter and FFT is replaced by loop unrolling and Neon intrinsic code which uses parallel pipeline resulted in the optimization with overall of ~57% and ~20% reduction. And in the Viterbi C-code, the add compare select unit is replaced by the Neon assembly code which resulted in the optimization resulting in ~69% reduction.

#### IV. IMPLEMENTATION RESULTS

During the development of the DAB Channel Decoder, the decoder modules are verified against an existing reference implementation. This implementation provides the reference input and output signals for verifying each signal processing block. During the development of the DAB Source Decoder, the decoder modules are verified against an existing reference implementation or using ISO reference. In case of non bit exactness, MATLAB test scripts can also be made use of. In the initial phase, all the testing/debugging will be carried out in simulator environment.



The baseband filter, FFT and Viterbi decoder module is first implemented in C using MSVC and in ARM DS5 as well using Neon intrinsic for optimization as these modules consumes more number of cycles. Both the outputs are compared and found bit exact. For the verification purpose we write the output of individual blocks into a file and compare it. The output of FIC is dumped in .dat file and MSC in .mpg file. The source decoder output is a .wav audio file. The audio file is heard without any artifacts. These optimizations have brought sufficient speedup improvement by reducing the number of cycles to be executed, so as to allow real time execution of the decoder.

## V CONCLUSION

The DAB baseband decoders are successfully implemented in Software Based Radio (SBR). The audio output of source decoder is bit exact and heard without any artifacts. And the number of bit errors are zero. After the optimization of C code by NEON intrinsic on ARM cortex-A8, the MIPS have reduced greatly. The output of C code and optimized code is found bit exact. Performance optimizations of cycle intensive modules are achieved.

## REFERENCES

- [1] ETS 300 401 V1.4.1, "Radio broadcasting systems; DAB to mobile, portable and fixed receivers", ETSI, 2006-01.
- [2] Ming-Der Shieh, Chien-Ming Wu, Hsiao-Hsing Chou, "Design and implementation of a DAB channel decoder", IEEE Transactions on Consumer Electronics, vol. 45, pp. 553-562, August 2000.
- [3] EN 300 798 V1.1.1, "DAB Distribution interfaces, Digital baseband In-phase and Quadrature (DIQ) interface", ETSI, 2003.
- [4] K.Yamauchi, S.Kakiuchi, A.Takebe, M.Sugitomo, "Digital Audio Broadcasting Receiver Development", Pioneer Electronic Corporation, Japan 1995.
- [5] R. Olde Dubbelink, Philips, " DAB, General-purpose and application-specific design of a DAB channel decoder", Eureka 147 – DAB consortium, EBU Technical Review Winter 1993
- [6] Andreas Muller, DAB-Software Receiver Implementation, WILEY, 2008.
- [7] Wolfgang Hoeg, Thomas Lauterbach, Digital Audio Broadcasting, Principles and Applications of Digital Radio, 2nd Edition, Wiley, 2003.
- [8] Guoyu Wang, Hongsheng Zhang, Mingying Lu, Chao Zhang, Tao Jiang, and Guangyu Guo, "Low-Cost Low-Power ASIC Solution for Both DAB+ and DAB Audio Decoding", IEEE transactions on very large scale integration (VLSI) systems, 1063-8210 © 2013
- [9] Moohong Lee, Byungjik Keum, Jeong Han Jeong, Young Serk Shim, and Hwang Soo Lee, Member, IEEE, "A Software-based Receiver Running on a Single DSP for Terrestrial Digital Multimedia Broadcasting" IEEE transactions, September 9, 2008
- [10] Matthias Kamuf, Viktor Öwall, John B. Anderson, "Optimization and Implementation of a Viterbi Decoder Under Flexibility Constraints", IEEE transactions on circuits and systems—I: Regular Papers, VOL. 55, NO. 8, September 2008
- [11] Jinjin He, Huaping Liu, Zhongfeng Wang, Xinming Huang, and Kai Zhang, " High-Speed Low-Power Viterbi Decoder Design for TCM Decoders", IEEE Transactions on very large scale integration systems, Vol.20, No.4, April 2012.
- [12] [www.worlddab.org](http://www.worlddab.org)