Algorithm for DRM Signal Recognition in Time Domain and Hardware Realization

Lukáš Ručkay[1), 2)] and Jiří Nedvěd[1)]

[1)] ASICentrum
Novodvorská 994, 142 21 Prague, Czech Republic
lukas.ruckay@asicentrum.cz

[2)] Department of Circuit Theory
Faculty of Electrical Engineering CTU Prague
Technická 2, 166 27 Prague, Czech Republic

***Abstract*—This article presents a new algorithm for DRM (Digital Radio Mondiale) signal recognition. Such an algorithm is useful for fast scanning and classifying broadcast transmission, which is our goal. The algorithm which we have presented in [1] works in frequency domain and is based on a processing of spectrum properties. The recognition algorithm presented here works in time domain and takes advantage of signal time structure. The new algorithm is nearly three times faster and its HW demands are evidently smaller in the comparison with the previous one [1]. A basic introduction into relevant parts of the DRM standard is given and an efficient implementation of the introduced algorithm is described as well. We also provide results of live radio frequency scanning test where our algorithm shows recognition score higher than 90%.**

I. INTRODUCTION

DRM (Digital Radio Mondiale) [2] is a universal digital AM radio system with a near-FM sound quality available to listeners worldwide. The quality of DRM audio is excellent, and the improvement on analogue AM is noticeable. DRM can be used for a range of audio content, including multilingual speech and music. Besides providing near-FM quality audio, the DRM system has the capacity to integrate data and text.

Unlike digital systems requiring a new frequency allocation, DRM uses the existing AM broadcast frequency bands. A DRM signal is designed to fit the existing AM broadcast band plan, based on signals of 9 kHz or 10 kHz bandwidth. It has modes requiring as little as 4.5 kHz or 5 kHz bandwidth, plus the modes that can take advantage of wider bandwidths, such as 18 kHz or 20 kHz. Many existing AM transmitters can be easily modified to carry the DRM signals. Currently about 20 DRM programs can be listened to on short waves in Europe.

We are involved in the project 4S (Smart chipS for Smart Surroundings) [3], [4], whose goal is to develop a heterogeneous SoC (System on Chip) [5] with small processing tiles for various tasks interconnected by a NoC (Network on Chip). Control oriented processes with less computationally intensive demands are mapped on a flexible GPP (General Purpose Processor). Computationally intensive processes are mapped on an energy-efficient coarse-grain domain specific reconfigurable processor, the MONTIUM [5].

Our first goal in this project was to develop an algorithm for fast scanning and classifying a type of signal; to recognize if the signal is DRM or not. The second goal was to design an efficient implementation of the proposed algorithm as a separate block in the heterogeneous SoC.

Since DRM is a relatively new standard, there are only few receiver products and available papers. The open source project DREAM [6] develops a real-time software implementation of a DRM receiver on a PC platform; it uses a sound card as an input and output device. The open source implementation can be used for the evaluation of different algorithms.

Our previous work [1] describes the recognition of DRM signal in frequency domain using the spectrum properties. This algorithm provides a good recognition score (higher than 91%) but its most significant disadvantage is high hardware demands. It needs 8 192-points Fast Fourier Transform (FFT) and a recognition process takes 0.35 sec.

II. DRM SIGNAL DESCRIPTION

The DRM system uses a transmission type called COFDM (Coded Orthogonal Frequency Division Multiplex) [2]. This means that all data produced by a digitally encoded audio and associated data signals are spread out for transmission across a large number of closely spaced carriers. All of these carriers are contained within an allotted transmission channel. The DRM system is designed so that the number of carriers can vary depending on factors such as an allotted channel bandwidth and a degree of robustness required. The robustness of the DRM signal can be modified to match different propagation conditions.

A. *Transmission structure*

The transmitted signal is organized in transmission super frames. Each transmission super frame consists of three

4S project (IST-001908) supported by the Sixth Framework Programme of the European Community

transmission frames. Each transmission frame has duration T_f and consists of N_f OFDM (Orthogonal Frequency Division Multiplex) symbols. Each OFDM symbol is constituted by a set of K carriers and transmitted with duration T_s. The spacing between adjacent carriers is $1/T_u$. The symbol duration T_s is the sum of two parts:

- useful part with duration T_u,
- guard interval with duration T_g.

The guard interval consists of a cyclic continuation of the useful part, and is inserted before it. The OFDM symbols in a transmission frame are numbered from 0 to N_f-1. All symbols contain data and reference information.

B. *DRM signal time structure*

The OFDM parameters must be chosen to match propagation conditions and the coverage area that the operator wants to serve. Various sets of the OFDM parameters are therefore defined for different conditions of propagation, and their parameter values are listed in Tab. 1. The time-related OFDM symbol parameters (N_s, N_u, N_g) are expressed in multiples of the elementary time period T=83.33μs.

TABLE I. OFDM SYMBOL PARAMETERS [2].

Parameters list	Robustness mode			
	A	***B***	***C***	***D***
T (μs)	83.33	83.33	83.33	83.33
T_f (ms)	400	400	400	400
$T_s = T_u + T_g$ (ms)	26.67	26.67	20	16.67
T_u (ms)	24	21.33	14.67	9.33
T_g (ms)	2.67	5.33	5.33	7.33
T_g/T_u	1/9	1/4	4/11	11/14
N_f (-)	15	15	20	24
$N_s = N_u + N_g$ (-)	320	320	240	200
N_u (-)	288	256	176	112
N_g (-)	32	64	64	88

The most significant information about the time structure is that the guard interval contains a part of the useful part of an OFDM symbol. In other words, one part of the OFDM symbol is transmitted twice. The time structure of the OFDM symbol is shown in Fig. 1.

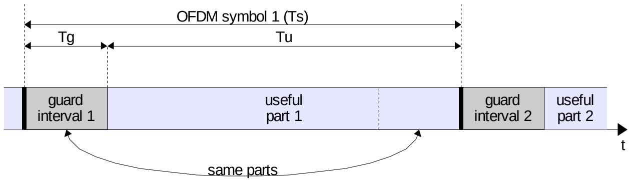

Figure 1. OFDM symbol time structure.

III. ALGORITHM FOR DRM SIGNAL RECOGNITION

The presented algorithm is based on a calculation of the autocorrelation function and uses the fact that the guard interval and the part of useful part inside the same OFDM symbol contain the same data, see Fig. 1.

An input signal representing a series of the OFDM symbols is a complex signal $x[n]$ represented by a real part $I[n]$ and an imaginary part $Q[n]$ and is delivered from a DDC (Digital Down Converter) block at frequency 24 kHz.

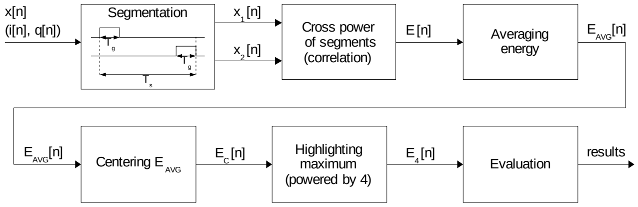

Figure 2. Block scheme of the recognition algorithm.

The autocorrelation function of signal $x[n]$ contains three typical peaks. A maximal peak is in accordance with the signal having a zero shift. Other two peaks are in accordance with the signal having a shift ±N_u (T_u) because of the guard interval position. In reality, the signal in the guard interval can be corrupted and thus the autocorrelation function need not have clear peaks. A block scheme of the recognition algorithm is shown in Fig. 2.

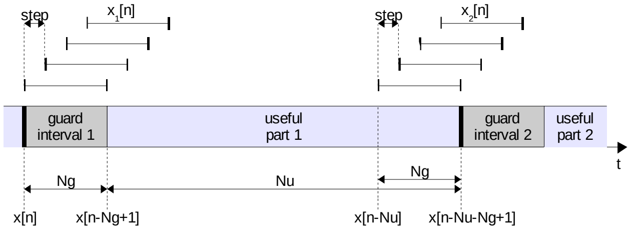

Figure 3. Segmentation and extraction signals $x_1[n]$ and $x_2[n]$.

A modification of the computation of the autocorrelation function of signal $x[n]$ is a computation the of cross-correlation function of segments $x_1[n]$ and $x_2[n]$, see Fig. 3. Both segments having length N_g can be described as follows:

$$\begin{aligned} x_1[n] &= x[n], \\ x_2[n] &= x[n-N_u]. \end{aligned} \qquad (1)$$

Since we do not need the whole cross-correlation function $R_{12}[n]$ of signals $x_1[n]$ and $x_2[n]$ but only the zero cross-correlation coefficient $R_{12}[0]$, we compute cross power $E[n]$ of segments $x_1[n]$ and $x_2[n]$ as follows:

$$\begin{aligned} E[n] &= \sum_{k=0}^{N_g-1} x[n-k] \cdot x[n-N_u-k] \\ &= \sum_{k=0}^{N_g-1} x_1[n-k] \cdot x_2[n-k]. \end{aligned} \qquad (2)$$

The cross power $E[n]$ is maximal if it is computed between the guard interval and a corresponding part of the useful part. For achieving a higher algorithm credibility, the cross power is computed from N_{sym} OFDM symbols (N_{sym}=8 by default).

An example of the cross power $E[n]$ is shown in Fig. 4. Each robustness mode (modes A, B, C and mode D) has a different duration of the guard interval (N_g) and a duration of the OFDM symbol (N_s), therefore the cross power has to be computed separately for each robustness mode. Fig. 4 shows the cross power for each of the robustness modes. The waveform of the cross power for mode B (Fig. 4) contains typical peaks, which have a distance corresponding to the duration of the useful part of the OFDM symbol.

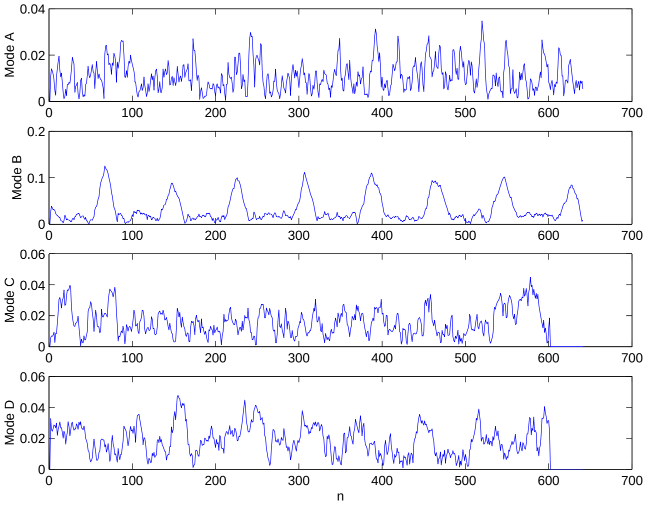

Figure 4. Illustration of cross power $E[n]$.

The cross power is averaged in the next step to suppress random components included in the DRM signal. The averaged cross power $E_{avg}[n]$ is computed from N_{sym} OFDM symbols (from signal having length N_{sym} OFDM symbols - robustness mode B) as follows:

$$E_{avg}[n] = \frac{1}{N_{sym}} \sum_{k=0}^{N_{sym}-1} E[n + k \cdot N_s]. \quad (3)$$

After that the averaged cross power is rotated so that its maximum is centred to the middle of waveform. The raising to the power of 4 highlights the centred cross power.

$$E_4[n] = \left(E_{avg}[n]\right)^4 \text{ and centering.} \quad (4)$$

Finally, an evaluation of $E_4[n]$ is performed to determine whether the examined signal is DRM or not. There are two thresholds used for this purpose. The first threshold (*tr1*) is equal to 1/8 of the maximum of $E_4[n]$. The second threshold (*tr2*) is equal to 7/8 of the maximum of $E_4[n]$.

The signal is marked as DRM if $E_4[n]$ fulfils the following requirements:

- The crossing through the first and second threshold is only and only twice.
- In the area where the first threshold is passed over and the second one not the $E_4[n]$ has to be monotonically increasing or decreasing.
- The direction of $E_4[n]$ (increasing/decreasing) depends on index n as follows:

$$\begin{aligned} &tr1 < E_4[n] < tr2 \text{ and } n \le N_g/2 \Rightarrow \text{increasing} \\ &tr1 < E_4[n] < tr2 \text{ and } n \ge N_g/2 \Rightarrow \text{decreasing} \end{aligned} \quad (5)$$

These requirements result in an approximation of a shape of the cross power to a triangular shape.

IV. Algorithm Optimization

The algorithm described above was realized and verified in the MATLAB system and therefore it works in a 64 bits floating point format. For an implementation, we need some optimizations to achieve minimal HW demands.

The calculation of the cross power $E[n]$ is very computationally demanding and therefore we used recurrent calculation of $E[n]$ as follows:

$$\begin{aligned} E[n] &= E[n-1] + x_1[n] \cdot x_2[n] \\ &\quad - x_1[n - N_g] \cdot x_2[n - N_g] \\ &= E[n-1] + x[n] \cdot x[n - N_u] \\ &\quad - x[n - N_g] \cdot x[n - N_u - N_g]. \end{aligned} \quad (6)$$

Since $x_1[n]$ and $x_2[n]$ in (6) are the complex signals, we have to express the cross power separately for the real and imaginary parts. For simplicity, we use operator $\mathrm{E}\{x_1, x_2\}$ for calculating the cross power according to (6). For achieving a higher recognizing score, the cross power calculation was slightly modified as follows:

$$\begin{aligned} E[n] = &\left|\mathrm{E}\{\mathrm{Re}(x_1), \mathrm{Re}(x_2)\} + \mathrm{E}\{\mathrm{Im}(x_1), \mathrm{Im}(x_2)\}\right| \\ &+ \left|\mathrm{E}\{\mathrm{Re}(x_1), \mathrm{Im}(x_2)\} - \mathrm{E}\{\mathrm{Im}(x_1), \mathrm{Re}(x_2)\}\right|. \end{aligned} \quad (7)$$

Another HW complexity reduction is reached by considering only each *step*th sample in $E[n]$, where *step*=8. By this we obtained a sampled cross power $E_{sampled}[n]$:

$$E_{sampled}[n] = E[n \cdot step]. \quad (8)$$

The next step is an averaging which is simplified only to an accumulation:

$$E_{avg}[n] = \sum_{k=0}^{N_{sym}-1} E_{sampled}[n + k \cdot \frac{N_s}{step}]. \quad (9)$$

V. IMPLEMENTATION

In this section, an implementation is discussed. The system was described in VHDL and designed to the ASIC 0.13μm technology. An FPGA implementation was used for evaluation and testing purposes as well. The whole block realizing the algorithm for recognition DRM signal is called "Signal recognizer" (SR) and its block scheme is shown in Fig. 5. A sampling frequency of the input data is f_s=24 kHz and system frequency can be 8 MHz or 24 MHz. Since the parameters N_g, N_u, N_s differ for all modes, the algorithm is implemented in four versions with sharing all blocks.

A. DATAMAP block

The function of this block is to adjust the input level of real and imaginary parts for SR. Both real and imaginary input data are multiplied by value 2^{exp}, where *exp* is in the range from 0 to 6, and cut off to 10 MSB bits. Multiplying of the input data is realized as a binary shift. If input signals overflow this 10 bits range, an error flag is provided.

B. FIFO block

At least one OFDM symbol is needed to store for the calculation $E[n]$. The longest OFDM symbol contains 320 elementary time periods *T*. Since period *T* is twice longer than the sampling period, the depth of FIFO must be 641 cells. One FIFO 641×20 for both real and imaginary parts is used. The FIFO is realized as RAM 640×20, 20 bits register for the last cell and control logic.

Because of the recurrent computation of $E[n]$, the first and last addresses are needed and so are 9 additional addresses inside the address range. These outputs lead to block IQ_MUX which prepares (multiplexes) data for IQ_ALU block.

C. IQ_ALU block

This block serves to the computation of the cross power according to (7) with slight modification where accumulators *ACC1* and *ACC2* are used for results saving:

$$\begin{aligned} ACC1_{mode} &= \mathrm{E}\left\{\mathrm{Re}(x_1), \mathrm{Re}(x_2)\right\} + \mathrm{E}\left\{\mathrm{Im}(x_1), \mathrm{Im}(x_2)\right\}, \\ ACC2_{mode} &= \mathrm{E}\left\{\mathrm{Re}(x_1), \mathrm{Im}(x_2)\right\} - \mathrm{E}\left\{\mathrm{Im}(x_1), \mathrm{Re}(x_2)\right\}. \end{aligned} \tag{10}$$

The block contains two parallel 10 bits multipliers for the computation of the cross power. After multiplication results are truncating to 16 bits to decrease bus widths. These truncated results are added or subtracted in dependency on control signals. 17 bits wide result is enlarged to 25 bits and added to or subtracted from an old value stored in accumulators. Since this block is used for all modes and each mode requires two accumulators (*ACC1* for real and *ACC2* for imaginary part), 8 accumulators are there.

D. AVERAGER block

This block is used for the averaging the cross power and saving the results. Its function can be described with the help of (10) as follows:

$$E_{mode}[n] = \left|ACC1_{mode}\right| + \left|ACC2_{mode}\right|, \tag{11}$$

$$E_{sampled}[n] = E_{mode}[n \cdot step]. \tag{12}$$

In this block a decimation of the cross power by factor *step*=8 is performed as well and therefore RAM for saving the results is organized into 270×29 bits. A memory space is shared as follows: 80 addresses for mode A, 80 addresses for mode B, 60 addresses for mode C, and 50 addresses for mode D. The averaging is done from N_{sym}=8 OFDM symbols of mode A. Since the OFDM symbols of mode C and D are shorter than in modes A and B it is possible to perform the averaging from more OFDM symbols for modes C and D in the same time. Actually, the averaging is done from 8 symbols for modes A and B, from 10 symbols for the mode C, and from 12 symbols for mode D. During the accumulating cross power of the last symbol, a maximum of the averaged cross power is searched.

E. EVALUATOR block

This block evaluates $E_{sampled}[n]$ and the output signalizes whether an input stream is DRM or not. Moreover, if the input stream is DRM, then the mode of DRM is signalized.

According to the maximum of $E_{sampled}[n]$, the input signal is scaled (a barrel shifter is used) to achieve better dynamics. The signal is truncated from 29 bits to 16 bits after scaling. The truncated signal is twice raised to the power of 2 and

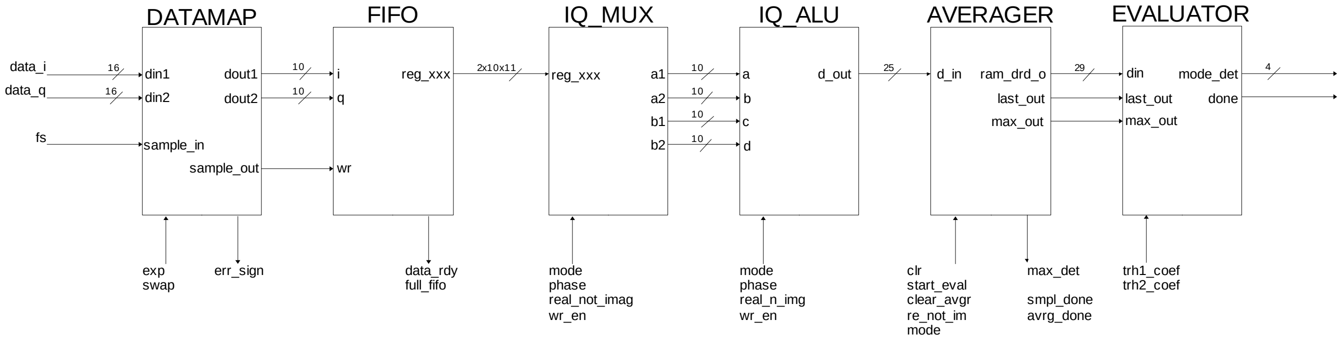

Figure 5. Block scheme of algorithm implementation (only data path, control logic is not included).

truncated back to 16 bits. After that the threshold levels are computed from an adjusted signal and they are compared with register values trh1_coef and trh2_coef. The difference between the adjusted and a temporary delayed signal is used for direction detection with the help of sign operator. A pattern evaluator evaluates relationships among comparison results.

F. Hardware demands

A synthesis into ASIC TSMC 0.18µm and into the Xilinx FPGA was performed to evaluate the hardware (HW) demands of the proposed algorithm. The whole system is designed as fully synchronous with one clock domain, therefore the synthesis is very simple.

The post ASIC TSMC 0.18µm synthesis results are summarized in Tab. 2. Migration of the design into the 0.13µm 4S project target technology will reduce the size to about 60% of the 0.18µm die area [7].

TABLE II. ASIC SYNTHESIS RESULTS.

Area of combinational cells	120 582	μm^2
Area of non-combinational cells	73 563	μm^2
Total cell area	194 145	μm^2
Area of cell NAND2X1	9.979 μm^2	
Number of Equivalent Gates (NAND2X1)	19 455	-

The second synthesis (for evaluating and testing purpose) was done into Xilinx device Virtex2 2v3000ff1152-4 with tool ISE 6.3i. Place and Route was done as well. Post place and route results are summarized in Tab. 3. In both Tab. 3 and Tab. 4, the RAM blocks are not included.

The design was successfully constrained to achieve system clock frequency 30MHz.

TABLE III. FPGA SYNTHESIS RESULTS.

Parameter	Used	Available
# of Slice Flip Flops	1 049	28 672
# of 4 input LUTs	3 031	28 672
# of occupied Slices	1 917	14 336
# of bonded IOBs	311	720
# of global CLKs	1	16
# of Equivalent Gates	35 462	

VI. TESTING AND RESULTS

The successfulness of recognition depends on the parameters as the first and second thresholds, the number of the OFDM symbols used for averaging (N_{sym}), and other parameters of the recognizing algorithm.

The first test results come from MATLAB model that also covers all quantization effects. Several records of real data from various frequency bands with length of several minutes are used as input data. In this test, we evaluated a recognition score and acquisition time in dependency on parameter N_{sym}. The results are summarized in Tab. 4, where # of samples represents required number of signal samples to perform recognition, and # of recognitions represents how many times the recognition was done for the DRM signal and non-DRM signal.

From the achieved recognition score and acquisition time point of the view, N_{sym}=8 seems to be the best choice. With this value, the recognition score reaches a value higher than 88% and the latency is ≈0.22 sec (4 recognitions per second). The algorithm presented in [1] uses FFT of size 8 192 which demands excessive HW for realization. An advantage of FFT algorithm is the recognition score higher than 91% for the acquisition time of 0.35 sec and 94% for the acquisition time of 0.7 sec.

TABLE IV. RECOGNITION SCORE.

Parameters				
Thresholds *tr1* and *tr2* [-]		1/8 and 7/8		
N_{sym} [-]		6	8	12
# of samples [-]		3 840	5 120	7 680
Samples acquisition time [sec]		0.16	0.2133	0.32
DRM signal	# of recognitions [-]	8 491	6 811	4 539
	score [%]	83.77	88.27	89.95
Non DRM signal	# of recognitions [-]	6 497	4 870	3 246
	score [%]	99.49	99.36	99.63

Another test was performed with a FPGA version. In this test data from a digital receiver was processed by our algorithm (FPGA version of SR) and by DREAM software running on a PC. Each carrier frequency position from Tab. 5 was listened to for 5 minutes. The status of the DREAM FAC (Fast Access Channel) indicator and the status of the FPGA recognizer were logged every second. Based on a number of DREAM and SR hits, the percentual recognition score was calculated. The results of both systems are summarized in Tab. 5.

TABLE V. LIVE RADIO TEST RECOGNITION RESULTS[1].

Carrier frequency [kHz]	Trans-mitter power [kW]	Location / Radio programme	DREAM score [%]	SR score [%]	Signal strength
5 990	50	Julingster / RTL DRM 2	46	90	varying
6 095	50	Julingster / RTL Radio	100	96	strong
7 240	40	Flevo / RNW	99	100	strong
7 265	200	Wertachtal / DW	99	100	strong
7 320	33	Rampisham / BBCWS	97	99	strong
9 470	50	Kvitsoy / BBCWS	100	99	strong
13 620	120	Sulaibiyah / RadioKuwait	0	73	weak
15 435	90	Sines / DW	5	40	weak
15 440	90	Sines / DW	94	100	strong
15 780	35	Taldom / VoR	97	99	strong

Tab. 5 shows that the recognition scores achieved by DREAM and SR are nearly the same and in most cases higher than 96%. However, there are several cases in which SR reaches a better score.

[1] Live reception in Prague, Lat:50.028, Lon:14.430 (WGS-84) on 31 May 2006 11:00 AM (GMT+2)

In the case of carrier frequency 5 990 kHz the varying signal strength caused that DREAM system achieved significantly lower recognition score (46%), whereas SR exploited a possibility to adjust the level of the input signal in the block DATAMAP and amplified the input signal up to four times (*exp* bit value up to 2), and therefore reached 90% score. The SNR of the input signal, the statuses of both systems, as well as the value of *exp* (labeled as "Recognizer scaling") are shown in Fig. 6.

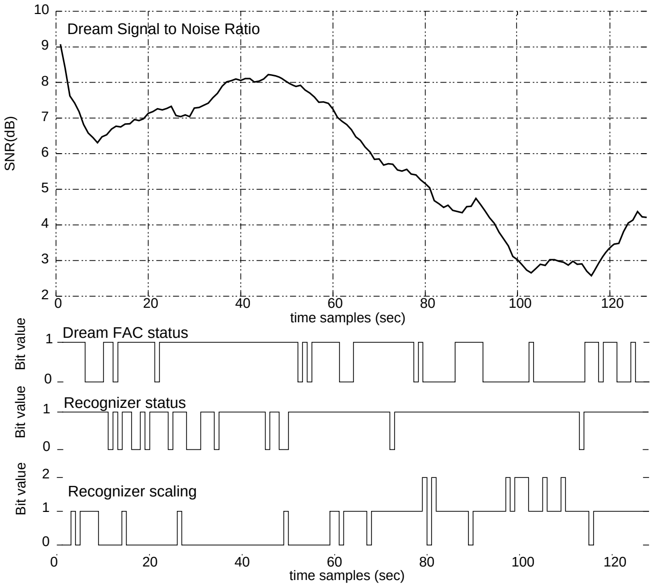

Figure 6. A 2-minutes portion of the waveform showing SNR and the recognition results of DREAM and SR.

At carriers 13 620 kHz and 15 435 kHz the DRM signal was very weak (relative signal strength is included into Tab. 5) and DREAM could hardly process and recognize such a signal. On the other hand, SR achieved the recognition score of 73% (or 40%) which was done mainly due to the possibility to amplify the input signals in the DATAMAP block.

VII. CONCLUSION

We introduced a new algorithm for recognition DRM signal. The designed algorithm works in time domain and exploits the fact that the same part of the OFDM symbol is transmitted twice. It uses modified autocorrelation function for this exploitation. The proposed algorithm is very robust and even insensitive to frequency tuning inaccuracy as high as several kHz.

High recognition score (more than 88%) is achieved for 8 OFDM symbols and the thresholds equal to 1/8 and 7/8. The processing of 8 OFDM symbols requires 5 120 samples of signal sampled by 24 kHz. The samples acquisition time for that given number of samples is only 0.22 sec. The algorithm presented here is faster and HW demands are significantly smaller in the comparison to the algorithm presented in [1].

Live radio test results are also provided and they confirm high achieved recognition score. Moreover, SR recognized DRM signal in more cases than DREAM system did. In these tests, SR achieved the recognition score higher than 96% for a strong signal.

An efficient implementation of the designed algorithm is described as well. Each block in the data path is discussed in detail. Finally, results of the two syntheses are provided. The chip area is approx. 0.2mm^2 in TSMC 0.18µm technology. Implementation into FPGA Xilinx Virtex2 device (2v3000ff1152-4) requires approx. 13% of logic.

VIII. ACKNOWLEDGEMENT

This research is conducted within the Smart Chips for Smart Surroundings (4S) project (IST-001908) supported by the Sixth Framework Programme of the European Community.

This work has been also supported by the research program Transdisciplinary Research in Biomedical Engineering No. MSM6840770012 of the Czech University in Prague, and the Grant GACR 102/03/H085: Biological and Speech Signal Modelling.

IX. REFERENCES

[1] L. Ručkay, "Recognition of DRM Signal in Frequency Domain and Hardware Demands", in Proceedings of the 2006 IEEE Workshop on Design and Diagnostics of Electronic Circuits and Systems. Prague, CTU Publishing House, 2006, vol. 1, pp. 73-74.

[2] European Telecommunication Standard Institute (ETSI), "Digital Radio Mondiale (DRM); System Specification. ETSI ES 201 980 edition", Sophia Antipolis, France, April 2003.

[3] G.J.M. Smit, E. Schüler, J. Becker, J. Quevremont and W. Brugger, "Overview of the 4S project", International Symposium on System-on-Chip (SoC 2005), pp. 70-73, November 2005.

[4] 4S project web: http://www.smart-chips.net

[5] P.M. Heysters, G.J.M. Smit and E. Molenkamp, "A flexible and energy-efficient coarse-grained reconfigurable architecture for mobile systems", The Journal of Supercomputing, vol. 26, pp. 283–308, 2003.

[6] A. Kurpiers and V. Fischer, "Open-source implementation of a digital radio mondiale (DRM) receiver", in 9th International IEE Conference on HF Radio Systems and Techniques, pp. 86-90, June 2003.

[7] TSMC web: http://www.tsmc.com/ english/b_technology

[8] T. Wiangtong, P.Y.K. Cheung and W. Luk, "Hardware/software codesign", Signal processing magazine, vol. 22, No. 3, 2005, pp. 14-22.