

# An Efficient FFT For OFDM Based Cognitive Radio On A Reconfigurable Architecture

Qiwei Zhang, Andre B.J. Kokkeler, Gerard J.M. Smit

Department of Electrical Engineering, Mathematics and Computer Science

University of Twente, Enschede, The Netherlands

Email: q.zhang@utwente.nl

**Abstract**— Cognitive Radio is a promising technology to utilize non-used parts of the spectrum that actually are assigned to licensed services. An adaptive OFDM based Cognitive Radio system has the capacity to nullify individual carriers to avoid interference to the licensed user. Therefore, there could be a considerably large number of zero-valued inputs/outputs for the IFFT/FFT in the OFDM transceiver. Due to the wasted operations on zero values, the standard FFT is no longer efficient. Based on this observation, we propose to use a computationally efficient IFFT/FFT as an option for OFDM based Cognitive Radio. Mapping this algorithm onto a reconfigurable architecture is discussed.

## I. INTRODUCTION

The increasing number of wireless multimedia applications leads to a spectrum scarcity. However, recent studies show that most of the assigned spectrum is underutilized. Cognitive Radio ([1], [2]) is proposed as a promising technology to solve the imbalance between spectrum scarcity and spectrum under-utilization. In Cognitive Radio, spectrum sensing locates the unused spectrum segments in a targeted spectrum pool and the aim is to use these segments optimally without harmful interference to the licensed user. This technology is also mentioned in [3] as *Spectrum Pooling*. Our research on Cognitive Radio is undertaken in the Adaptive Ad-hoc Freeband (AAF) project [4]. The goal of the project is to demonstrate an ad-hoc wireless communication network for emergency situations based on Cognitive Radio principles. Our work mainly focuses on mapping algorithms used in Cognitive Radio onto a reconfigurable platform.

In spectrum pooling, OFDM is proposed as the baseband transmission scheme. Those subcarriers which cause the interference to the licensed user should be nullified. Therefore, there are zero-valued inputs for the IFFT of the transmitter and zero-valued outputs for the FFT of the receiver. When zero-valued inputs/outputs outnumber non-zero inputs/outputs, the standard IFFT/FFT used for OFDM is no longer efficient. If there is a large number of zero inputs/outputs, we propose to use a computationally efficient IFFT/FFT based on [5]. We will discuss how this algorithm can be mapped onto a coarse grain reconfigurable architecture, called Montium, which is the key element on our proposed platform [6] for Cognitive Radio.

This paper is organized as follows. In section II our conceived OFDM system for Cognitive Radio is presented. We will introduce a computationally efficient FFT/IFFT in section

III. Then a discussion on how to implement the algorithm onto the proposed reconfigurable platform is followed in section IV.

## II. OFDM BASED COGNITIVE RADIO

Theoretically, an OFDM based Cognitive Radio system can optimally approach the Shannon capacity in the segmented spectrum by adaptive resource allocation on each subcarrier, which includes adaptive bit loading and adaptive power loading. For Cognitive Radio, OFDM also offers high data rates and robustness to multipath delay spread. Furthermore, OFDM is easy to integrate with spectrum sensing because of the hardware reuse of FFT cores. Therefore, OFDM is a good candidate for the Cognitive Radio baseband system.

In [6], we proposed an OFDM system with adaptive bit loading and power loading for Cognitive Radio. We could maximize the data rate of the system under a certain power constraint. It is formulated as follows:

$$\begin{aligned} \text{Max } R &= \sum_{k=1}^K \frac{F_k}{K} \log_2 \left( 1 + \frac{h_k^2 p_k}{N_0 \frac{B}{K}} \right) \\ \text{Subject to: } &\sum_{k=1}^K p_k \leq P_{total} \\ &F_k \in \{0, 1\} \text{ for all } k \\ &p_k = 0 \text{ for all } k \text{ which satisfies } F_k = 0 \end{aligned} \quad (1)$$

where  $R$  is the data rate;  $K$  is the number of the subcarriers.  $N_0$  is the noise power density,  $B$  is the band of interest for Cognitive Radio,  $h_k$  is the subcarrier gain and  $p_k$  is the power allocated to the corresponding subcarrier.  $F_k$  is the factor indicating the availability of subcarrier  $k$  to Cognitive Radio, where  $F_k = 1$  means the  $k$ th carrier can be used by Cognitive Radio. The system power minimization can also be applied under the constraint of a constant data rate. We formulate it as follows:

$$\begin{aligned} \text{Min } &\sum_{k=1}^K p_k = P_{total} \\ \text{Subject to: } &R = \sum_{k=1}^K \frac{F_k}{K} \log_2 \left( 1 + \frac{h_k^2 p_k}{N_0 \frac{B}{K}} \right) \\ &F_k \in \{0, 1\} \text{ for all } k \\ &p_k = 0 \text{ for all } k \text{ which satisfies } F_k = 0 \end{aligned} \quad (2)$$

A simplified system is illustrated in Figure 1. A bit allocation vector indicates how many bits are loaded on each subcarrier. The number of bits corresponds to the different modulation types used for each subcarrier. The bit allocation vector is determined by the spectrum occupancy information from spectrum sensing and the SNR of subchannels [7]. The

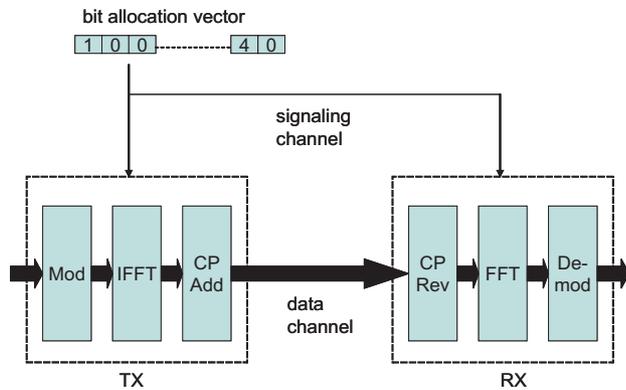


Fig. 1. OFDM for Cognitive Radio

basic idea is to load more bits on good subcarriers and load zeros to carriers which cause interference to the licensed user or lead to poor transmissions. The bit allocation vector is disseminated via a signaling channel so that the transmitter and the receiver have the same information. The inputs to the IFFT on the transmitter are complex samples with arbitrarily distributed zeros. A cyclic prefix is added for each OFDM symbol before transmission. On the receiver side, the OFDM symbol is obtained after removing the cyclic prefix. The FFT is performed and outputs the original complex samples together with zeros.

In our conceived OFDM system, there could be a large number of zero inputs/outputs for IFFT/FFT when a large part of the spectrum is not available to Cognitive Radio or there are many bad channels. If the number of zero values are considerably large, the standard FFT is no longer efficient due to the wasted operations on zero values. In [8] and [5], this fact was observed and several algorithms with low complexity have been proposed. Inspired by their work, we propose to use a low complexity algorithm as an option for OFDM based Cognitive Radio.

### III. AN EFFICIENT FFT ALGORITHM

From a system point of view, the FFT and IFFT are the critical parts of OFDM transceivers. They are also the most computational intensive blocks in OFDM [7]. Therefore, an inefficient IFFT/FFT can considerably waste computational power of the overall system. The inefficiency of the standard FFT if there are many zero valued inputs is illustrated by an example in Figure 2. Because the IFFT and the FFT are the same in terms of the computational structure we confine all the following discussions on the FFT. In Figure 2, we show the computational structure of the 8-point Decimation-In-Frequency Radix-2 FFT where only two inputs are non-zero. All the operations indicated by the dashed line are

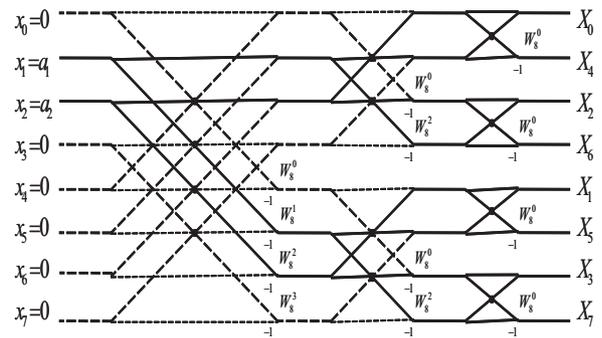


Fig. 2. 8-point DIF Radix-2 FFT

actually wasted on the zero-valued inputs. In the first stage, the first butterfly and the last butterfly are completely unnecessary and the other two butterflies are partial butterflies, which are less complex and appear in the second stage as well. By pruning all those unnecessary operations, the computational power saving is considerable. This idea first appeared in [8] known as *FFT pruning*.

#### A. FFT Pruning

In [8], a DIF (Decimation-In-Frequency) FFT pruning algorithm was suggested. Later the FFT pruning algorithm was extended to DIT (Decimation-In-Time) FFT [9] and to both input and output pruning [10]. In fact the basic idea of all these pruning algorithms is to determine the index of the butterflies to be chosen for calculations. The index is generated at runtime by conditional statements. However, the effort to generate the index can be considerable due to the execution of conditional statements. The index shows irregularity because of the irregular position of zero inputs/outputs. This irregularity makes the hardware implementation of FFT pruning difficult. The idea of FFT pruning was applied to a multichannel OFDM system in [11], where a significant reduction of complexity was suggested. However, the application on OFDM in [11] assumes zero values with a considerable regularity. This is not true for Cognitive Radio where subcarriers are switched off at random positions based on the spectrum occupancy information and the subchannel condition.

Therefore, the OFDM based Cognitive Radio will not be based on FFT pruning but requires an efficient FFT algorithm which applies efficiently to zero inputs/outputs with arbitrary distributions. Moreover, this algorithm should have regularity which facilitates an efficient hardware implementation.

#### B. Transform Decomposition

In [5], Sorensen et al. proposed an efficient algorithm called *transform decomposition*. It is shown that transform decomposition is more efficient and flexible than FFT pruning. Transform decomposition can be seen as a modified Cooley-Tukey FFT where the DFT is decomposed into two smaller DFTs. We give a short introduction on transform decomposi-

tion. The DFT is defined as:

$$X(k) = \sum_{n=0}^{N-1} x(n)W_N^{nk} \quad k = 0, 1, \dots, N-1 \quad (3)$$

,where  $W_N^{nk} = e^{-j\frac{2\pi nk}{N}}$ . We consider the case where  $L$  outputs are nonzero. Let  $N$  be factorized as two integers  $N_1$  and  $N_2$ , so  $N = N_1N_2$ . The index  $n$  can be written as:

$$n = N_2n_1 + n_2 \quad (4)$$

$$n_1 = 0, 1, \dots, N_1 - 1 \quad n_2 = 0, 1, \dots, N_2 - 1$$

Substitute  $n$  in (3) with (4) and then the DFT can be rewritten as:

$$X(k) = \sum_{n_2=0}^{N_2-1} \sum_{n_1=0}^{N_1-1} x(N_2n_1 + n_2)W_N^{(N_2n_1+n_2)k}$$

$$= \sum_{n_2=0}^{N_2-1} \left[ \sum_{n_1=0}^{N_1-1} x(N_2n_1 + n_2)W_N^{N_2n_1k} \right] W_N^{n_2k} \quad (5)$$

We define:

$$X_{n_2}(\langle k \rangle_{N_1}) = \sum_{n_1=0}^{N_1-1} x(N_2n_1 + n_2)W_{N_1}^{n_1k}$$

$$= \sum_{n_1=0}^{N_1-1} x_{n_2}(n_1)W_{N_1}^{n_1k} \quad (6)$$

where  $\langle \cdot \rangle_{N_1}$  denotes modulo  $N_1$ . So (5) can be written as:

$$X(k) = \sum_{n_2=0}^{N_2-1} X_{n_2}(\langle k \rangle_{N_1})W_N^{n_2k} \quad (7)$$

Therefore the original  $N$ -point DFT with  $L$  nonzero outputs is decomposed into two major parts: the  $N_2 N_1$ -point DFTs in (6) which can be implemented as  $N_2 N_1$ -point FFTs and the multiplications with twiddle factors and recombinations of the multiplications in (7). Because the index  $k$  only consists of  $L$  nonzero values, only  $L$  twiddle factors are multiplied with each  $X_{n_2}(\langle k \rangle_{N_1})$  for  $n_2 = 1, 2, \dots, N_2$ . This multiplication part results in a reduction of the computation. The mathematical derivation for the transform decomposition algorithm with zero inputs is rather similar, details can be found in [5]. Unlike FFT pruning, transform decomposition does not need conditional statements to choose the butterfly for calculations.

### C. Transform Decomposition for Hardware Implementation

Transform decomposition shows considerable regularity which facilitates its hardware implementation. Based on the discussion in the previous section, we will show the computational structure of transform decomposition followed by a complexity analysis for our targeted Cognitive Radio system.

Although the algorithm in [5] applies to both the power-of-two FFT and the prime factor algorithm, we will only consider the power-of-two case because only power-of-two FFTs are used in the proposed OFDM system. When only  $L$  subcarriers are activated, there are  $L$  nonzero inputs/outputs for the IFFT/FFT. We choose  $N_1$  as the nearest power-of-two

integer larger than  $L$  and as a factor of  $N$ . This choice of  $N_1$  helps to exploit more regularities. We show the computational structure in Figure 3. Basically the computation can be divided

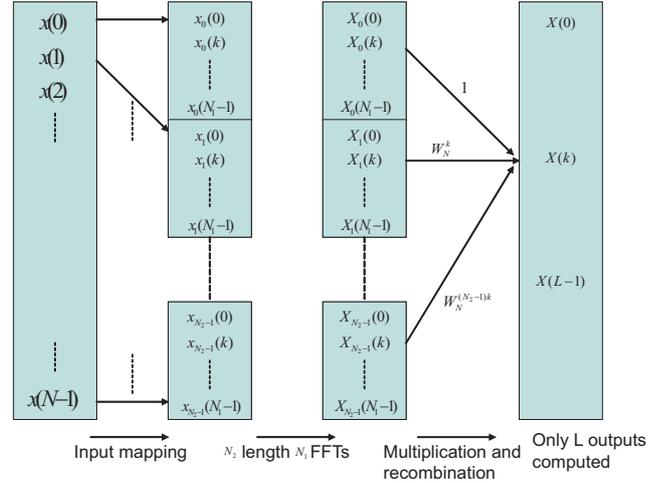


Fig. 3. Computational structure of transform decomposition

into two stages: FFTs and multiplications with recombination. Before the FFT computation, the input samples are mapped to  $N_2$  memory blocks according to (6). Then the  $N_1$ -point radix-2 FFT is performed on each of the  $N_2$  memory blocks. The results in the  $k \bmod N_1$  memory positions in each memory block are multiplied by twiddle factors and recombined to produce the output  $X_k$ . We can find the regularity in the computational structure: the memory addressing is constantly hopping from the same position in one block to another. Because  $N$ ,  $N_1$  and  $N_2$  are all power-of-two integers, we can use the most significant bits to indicate the block address. The block based addressing is done by changing the most

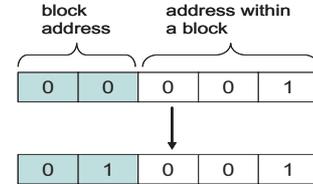


Fig. 4. A block address example

significant bits. For example in Figure 4, suppose 32 memory positions are divided into 4 blocks with 8 memory positions in each block. The first two bits are used as block address. If the address change from position 1 in the first block to position 1 in the second block, we increase the most significant bits by 1 from 00 001 to 01 001.

According to the computational structure in Figure 3, we make quantitative analysis on the computational complexity by counting the number of complex multiplications. The number of multiplications for transform decomposition equals:

$$Mul_{td} = (N_2 - 1) * L + \frac{N}{2} \log_2 N_1 \quad (8)$$

where  $N_1$  is the nearest power-of-two integer larger than  $L$ . The number of multiplications for  $N$ -point radix-2 FFT is:

$$Mul_{radix-2} = \frac{N}{2} \log_2 N \quad (9)$$

Figure 5 shows the computational complexity comparison between the radix-2 FFT and transform decomposition. Suppose the total number of subcarriers for OFDM is 1024. When less than half of the subcarriers can be used by Cognitive Radio, transform decomposition can reduce the computational complexity. It is shown that the saving can be considerable when

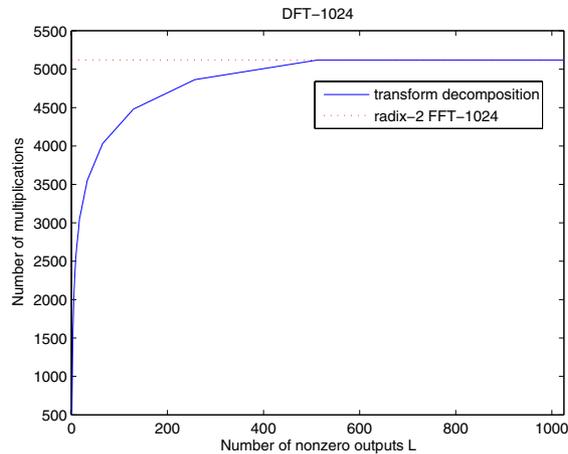


Fig. 5. Comparison of computation for radix-2 FFT and transform decomposition  $N=1024$

the number of nonzero values is small. For example, when only 32 out of 1024 subcarriers are available for Cognitive Radio, transform decomposition offers 30% saving of computations. Because FFT and IFFT are the most computational intensive parts in an OFDM transceiver, the savings can significantly reduce the computational complexity of the overall system. Therefore, transform decomposition can be an efficient option for an OFDM based Cognitive Radio system when only a small number of subcarriers are available for Cognitive Radio. To support this option, a reconfigurable architecture has to be reconfigured from radix-2 FFT to transform decomposition and vice versa.

#### IV. MAPPING ONTO A RECONFIGURABLE ARCHITECTURE

##### A. A Reconfigurable Platform for Cognitive Radio

As already foreseen by Mitola [1], Cognitive Radio is the final point of software-defined radio platform evolution: a fully reconfigurable radio that changes its communication functions depending on network and/or user demands. In [6], we proposed a heterogeneous reconfigurable System-on-Chip platform to support the physical layer reconfigurability of Cognitive Radio shown in Figure 6. The SoC is a heterogeneous tiled architecture, where tiles can be various processing elements including General Purpose Processors (GPPs), Field Programmable Gate Arrays (FPGAs), Application Specific

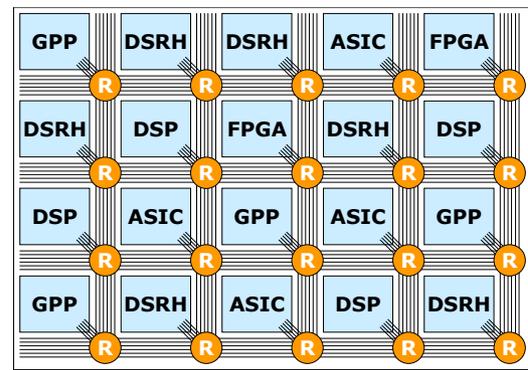


Fig. 6. Heterogeneous Multiprocessor tiled SoC

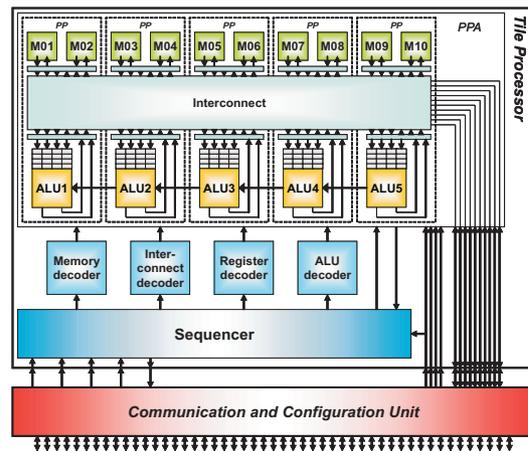


Fig. 7. Montium tiled processor

Integrated Circuits (ASICs) and Domain Specific Reconfigurable Hardware (DSRH) modules. The tiles in the SoC are interconnected by a Network-on-Chip (NoC). Both the SoC and NoC are dynamically reconfigurable, which means that the programs (running on the reconfigurable processing elements) as well as the communication links between the processing elements are configured at run-time. The Montium [12] tile processor (see Figure 7) developed at the University of Twente is an example of DSRH. It targets the digital signal processing (DSP) algorithm domain, which is the heart of the wireless baseband processing. Our previous work [13] shows that the Montium architecture is flexible enough to adapt to different algorithms with good energy-efficiency. Therefore, the Montium tiled processor is the key element in our proposed reconfigurable platform for Cognitive Radio.

##### B. Montium Tiled Processor

The Montium is an example of DSRH which targets the 16-bit digital signal processing (DSP) algorithm domain. At first glance the Montium architecture bears a resemblance to a Very Long Instruction Word (VLIW) processor. However, the control structure of the Montium is very different. For (energy-) efficiency it is imperative to minimize the control overhead. This can be accomplished by statically scheduling

instructions and using instruction decoders. The lower part of Figure 7 shows the Communication and Configuration Unit (CCU) and the upper part shows the reconfigurable Tile Processor (TP). The CCU implements the interface for offtile communication. The TP is the computing part that can be configured to implement a particular algorithm. By statically scheduling instructions at compile time, the overhead of both communication and control is reduced. Therefore, the instruction decoding does not result in excessive switching of control signals which consumes considerable energy. The ALUs can do basic DSP operations like multiplications and additions and they can also perform basic logic functions. The five identical ALUs (ALU1...ALU5) in a tile can exploit spatial concurrency to enhance performance. This parallelism demands a very high memory bandwidth, which is obtained by having 10 local memories (M01...M10) in parallel. The small local memories are motivated by the locality of reference principle. Each memory has a reconfigurable Address Generation Unit (AGU).

### C. Algorithm Mapping

In this section, we discuss how transform decomposition is mapped to the Montium based reconfigurable platform. The computational structure of transform decomposition in Figure 3 includes two parts: computation and memory addressing. The computations consist of: butterfly operations for the FFT in the first stage and multiplications with recombination in the second stage. The ALU in the Montium can efficiently perform those operations [12]. The memory addressing of transform decomposition needs block based address. Thanks to the AGU of the Montium, block based addressing is supported by changing the base register which represents the most significant bits of an address. The same approach has been used in [14] to implement a Prime Factor Algorithm (PFA) onto the Montium.

The transform decomposition can be done on the Montium in following steps:

- **step 1:** According to the number of zeros in bit allocation vector, a general purpose processor in the platform will choose from two options for DFT: radix-2 FFT or transform decomposition. When the number of zeros exceeds a certain threshold, transform decomposition will be chosen to reduce the computational complexity. If transform decomposition is chosen, the configuration code and twiddle factors generated by the general purpose processor will be sent to the Montium via Network-on-Chip.
- **step 2:** After the mapping the input samples into memory blocks, the Montium will perform a radix-2 FFT for each of the memory blocks.
- **step 3:** The intermediate results are multiplied by twiddle factors and recombined to produce the outputs to be sent to the Network-on-Chip.

The Montium needs  $\frac{N}{2} \log_2 N_1$  clock cycles for  $N_2$  length  $N_1$  FFT in step 2 and  $(N_2 - 1) * L$  clock cycles for the complex multiplications with recombination in step 3. No clock cycles are wasted for the data re-ordering. A radix-2

1024 point FFT takes 5120 clock cycles on the Montium. In a case where only 32 out of 1024 values are nonzero, transform decomposition on the Montium only takes 3522 clock cycles. If the Montium runs at 100MHz, transform decomposition needs 35.2  $\mu$ s instead of 51.2  $\mu$ s for a radix-2 FFT. In [12], the power consumption of the Montium, in 0.13  $\mu$ m technology, is estimated at 0.577 mW/MHz. The energy consumption of the transform decomposition on the Montium for the given case costs only 2.1  $\mu$ J.

## V. CONCLUSION

In this paper, we present a computationally efficient FFT/IFFT algorithm, namely transform decomposition, as an option for OFDM based Cognitive Radio in case a large number of subcarriers are nullified. A reconfigurable platform is used to support this option. Mapping transform decomposition onto a coarse-grain reconfigurable processor, the Montium, has been discussed. The Montium architecture matches the computational structure of the algorithm very well. The estimation shows that this efficient algorithm on the Montium offers a faster computation and a significant energy saving.

## ACKNOWLEDGMENT

The work is sponsored by the Dutch Ministry of Economic affairs Freeband AAF project.

## REFERENCES

- [1] J. Mitola III. *Cognitive Radio: An Integrated Agent Architecture for Software Defined Radio*, PhD Thesis, Royal Institute of Technology, Sweden, May. 2000.
- [2] S. Haykin *Cognitive radio: Brain-empowered wireless communication*, IEEE J. Select. Areas Commun., vol. 23, no.2:pp 201-220, Feb. 2005.
- [3] T.A. Weiss and F.K. Jondral *Spectrum pooling: An innovative strategy for the enhancement of spectrum efficiency*, IEEE Commun. Mag., Mar. 2004
- [4] www.freeband.nl
- [5] Henrik V. Sorensen and Sidney Burrus *Efficient Computation of the DFT with Only a Subset of Input or Output Points*, IEEE Trans. on Signal Processing, Mar. 1993
- [6] Qiwei Zhang, Andre B.J. Kokkeler and Gerard J.M. Smit *A Reconfigurable Radio Architecture for Cognitive Radio in Emergency Networks*, European Conference on Wireless Technology, September 2006, Manchester, UK
- [7] Qiwei Zhang, Fokke W. Hoeksema, Andre B.J. Kokkeler and Gerard J.M. Smit *Towards Cognitive Radio for emergency networks*, Book Chapter in Mobile Multimedia: Communication Engineering Perspective, Nova Publishers, 2006
- [8] John D. Markel *FFT Pruning*, IEEE Trans. on Audio and Electroacoustics, Dec. 1971
- [9] David P. Skinner *Pruning the Decimation In-Time FFT Algorithm*, IEEE Trans. on Acoustics, Speech, and Signal Processing, Apr. 1976
- [10] T.V. Sreenivas and P.V.S. Rao *FFT Algorithm for Both Input and Output Pruning*, IEEE Trans. on Acoustics, Speech, and Signal Processing, Jun. 1979
- [11] Shousheng He and Mats Torkelson *Computing Partial DFT for Comb Spectrum Evaluation*, IEEE Signal Processing Letters, Jun. 1996
- [12] Paul Heysters *Coarse-Grained Reconfigurable Processors; Flexibility meets Efficiency*, PhD Thesis, University of Twente, Sep. 2004
- [13] Paul M. Heysters and Gerard J.M. Smit *Mapping of DSP algorithms on the montium architecture*, In Proceedings of Reconfigurable Architecture Workshop, 2003
- [14] Arnaud Rivaton, Jerme Quevremont, Qiwei Zhang, Pascal T. Wolkotte, and Gerard J.M. Smit. *Implementing Non Power-of-two FFTs on Coarse Grain Reconfigurable Architectures*, International Symposium on System-on-Chip, 2005. pp. 82-85