10

Towards a Reconfigurable FFT : Application to Digital Communication Systems

Florent Camarda,Jean-Christophe Prevolet and Fabienne Nouvel
IETR/INSA Rennes
France

1. Introduction

Since the beginning of humanity, people have always wanted to communicate with each other. Today, this objective remains the same although a huge technological step has been taken since the first communication link. More recently, with the development of electronics systems, the number of interconnected devices has considerably increased and the quest of performances becomes more and more a reality.

In this context, digital communication systems have provided an efficient alternative to classical analog devices. For example, in the telecommunication domain, as analog television was a great technological progress less than a century ago, 3D HD television sets are currently being commercialized and will progressively replace the old devices.

The proliferation and variety of such digital devices have also led to the elaboration of multiple standards destined to ease the exchange of information between the different devices. Unfortunately, there are still too many standards to improve the portability and interconnections between different systems. The existence of such an amount of various standards is related to economical and technological constraints but also to political and strategical reasons which makes the interoperability between standards very difficult. This generally imposes the engineers to design as many circuits as standards which seems very inefficient and may be seen as a waste of time and money. This solution is even worse as a particular standard does not differ completely from another one. There are generally a lot of similarities between standards in terms of proposed algorithms and functions. It is sometimes sufficient to modify some parameters of a specific algorithm to change from one standard to another.

For example, in current communication systems, many applications referring to several standards make use of Orthogonal Frequency Division Multiplexing (OFDM) modulations (Weinstein & Ebert, 1971) such as Digital Terrestrial Television Broadcasting (DTTB), Digital Audio Broadcasting (DAB), Wireless Local or Personal Area Network (WLAN-WPAN), HomePlug 1.0 and HomePlug AV, HyperLAN, 802.11 standards, Digital Subscriber Line standards (xDSL), 3GPP, Long Term Evolution (LTE), etc.

The OFDM principle is mainly based on the use of the Fast Fourier Transform (FFT) and its Inverse (IFFT) (P. Duhamel & M. Vetterli, 1990). However, the sizes of FFT employed in all

these standards are various. DVB standards use 2048, 4096, 8192, 16384 or 32768 points FFT, Homeplug standards require 128 or 3072 points FFT, the chinese DTV standard employs a 3780 points FFT. Moreover, some of these communication systems need to implement several FFT sizes within a particular standard. It is the case for SC-FDMA or for all Time Domain Synchronous OFDM (TDS-OFDM) systems (J. Wanget al., 2003), in which a second FFT size is often needed for synchronization.

In order to develop a multi-standards receiver, the solution which is usually retained by manufacturers consists in implementing various Intellectual Properties (IPs) blocks, each associated to a unique function and a unique standard. This solution is often under-optimized in the sense that it does not allow the reuse of components within the chip and thus constitutes a waste of hardware resources.

Another approach is to design a generic and unique system capable of dealing with the specifications of different standards. This circuit has to be designed in such a way that it must be able to adapt itself to a particular standard or a dedicated application.

This concept, known as dynamic reconfigurability, provides not only a high flexibility but also permits to considerably reduce the time to market and the design effort.

Dynamic reconfigurability is a powerful concept since it enables a circuit to change its functionality on the fly according to the users' needs or the environmental conditions. The dynamic reconfiguration may be performed at different granularities depending on the technology. For example, some circuits allow to be reconfigured at a gate level by modifying the contents of a logic equation. This is typically the case of FPGAs circuits (Field Programmable Gate Array) which are widespread circuits that constitute a good alternative to ASICs (Application Specific Integrated Circuits). Other circuits may be reconfigured at a higher level of granularity, for example at a functional level. In this case, a specific operator (or function) is modified to change the behavior of an application.

In general, the level of performances depends on the granularity of reconfiguration. If the granularity is very fine, the flexibility is high but the timing performances are very low since a lot of resources have to be reconfigured. As a consequence, the time which is necessary to reconfigure the system constitutes a real issue and limits the use of such circuits in real time embeddded systems where timing constraints are generally tight.

Nevertheless, some circuits make intensive use of reconfiguration at a high level of granularity and permit to obtain not only high performances but also flexibility. This is generally performed by reconfiguring specific operators or group of operators in order to change the design functionality.

In this article, we present a new system capable of dealing with multi communication standards. The basis of this work has consisted in developing a unique FFT operator which is capable of being reconfigured to adapt to the different standards' specification. In our case, the FFT operator should be able to compute FFTs of size 2K, 4K, 8K and 3780 points.

In section 2, the nature of the FFT operator will be presented as well as the different possibilities for its implementation. Section 3 will describe the proposed reconfigurable architecture and obtained performances after implementation. An optimization is presented in section 4 with its performances results.

2. FFT Algorithms

The Fourier transform is a very useful operator for image or signal processing. Thus it has been extensively studied and the litterature about this subject is very rich. The Discrete Fourier Transform (DFT) is used for the digital signal processing and its expression is given in (1)

$$X(k) = \sum_{n=-\infty}^{\infty} x(n) e^{j2\pi \frac{nk}{N}} \tag{1}$$

It appears obvious that this expression can not be computed in a finite time due to the infinite bounds. From that, the usully computed expression is the N-points Fast Fourier Transform given in (2)

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{j2\pi \frac{nk}{N}} \tag{2}$$

The expression of the FFT is bounded and computable with a finite algorithmic complexity. This complexity is expressed as an order of multiplications and additions. Computing a N-points FFT without any simplification requires an algorithmic complexity of $\mathcal{O}(N^2)$ multiplications and $\mathcal{O}(N^2)$ where $\mathcal{O}$ denotes the "order of" multiplications and additions. Note that the real number of additions is $N(N-1)$ which is $\mathcal{O}(N^2)$. This reduction of complexity is however not sufficient for the large FFT sizes that are used in many digital communications standards.
A great reduction of this complexity can be achieved by using an efficient algorithm. It exists many FFT algorithms through the litterature like the Radix algorithm (J. W. Cooley & J. W. Tukey, 1965), the Prime Factor Algorithm (PFA) also called Good Thomas Algorithm (I. J. Good, 1958). We also can cite the algorithms of Rader-Brenner (C. M. Rader, 1968), Bruun's (G. Bruun, 1978) or the Winograd Fourier Transform Algorithm (WFTA) (S. Winograd, 1978). The most known and used of them is unmistakably the Radix algorithm. However other algorithms like the WFTA can be very convenient for specific FFT sizes. In this section, we focus on the development of the Radix algorithm. The Mixed-Radix algorithm will then be introduced. We also present two advantages of the WFTA and propose a way to combine the Radix and WFTA algorithms.

2.1 The radix algorithm

In 1965 Cooley and Tukey proposed a new way to compute the FFT for sizes that are power of 2, this is the birth of the Radix algorithms. The basic principle is based on the decomposition of a FFT of size N in two FFTs of size $\frac{N}{2}$.

2.1.1 Decimation in time / Decimation in frequency

This decomposition can be achieved by two means which are called Decimation In Time (DIT) and Decimation In Frequency (DIF) depending on the regrouped terms. Equation (3) present the decomposition for the DIT while equations (4) and (5) are for the DIF. Thanks to this the algorithmic complexity becomes reduced to $\mathcal{O}(\frac{N}{2}^2)$ since $N = 2$.

$$\begin{aligned} X(k) &= \sum_{n=0}^{N-1} x(n) e^{j2\pi \frac{nk}{N}} \\ &= DFT_{\frac{N}{2}}\left[x(2n')\right] + e^{j2\pi \frac{k}{N}} DFT_{N/2}\left[x(2n'+1)\right] \quad \text{with } n' \in \left\{0, ..., \frac{N}{2}-1\right\} \end{aligned} \tag{3}$$

This decomposition is called DIT because the N time samples are reordered in two groups of N/2 samples. One group contains the even indexed samples, while the second one contains the odd indexed samples. FFT of size N/2 are then computed for each group. The frequency samples are computed in order.

$$\begin{aligned} X(2k') &= \sum_{n=0}^{N-1} x(n) e^{j2\pi \frac{n2k'}{N}} \\ &= DFT_{\frac{N}{2}}\left[x(n) + x(n+\frac{N}{2})\right] \quad \text{with } k' \in \left\{0, ..., \frac{N}{2}-1\right\} \end{aligned} \tag{4}$$

$$\begin{aligned} X(2k'+1) &= \sum_{n=0}^{N-1} x(n) e^{j2\pi \frac{n(2k'+1)}{N}} \\ &= DFT_{\frac{N}{2}}\left[\left(x(n) - x(n+\frac{N}{2})\right) e^{j2\pi \frac{n}{N}}\right] \quad \text{with } k' \in \left\{0, ..., \frac{N}{2}-1\right\} \end{aligned}$$

This decomposition is called DIF because the even indexed frequency samples are computed by the first N/2-points FFT while the odd indexed samples are computed by the second one. The time samples are not reordered before computation. In order to simplify these equations all terms in $e^{j2\pi \frac{x}{N}}$ are usually expressed as W_N^x.

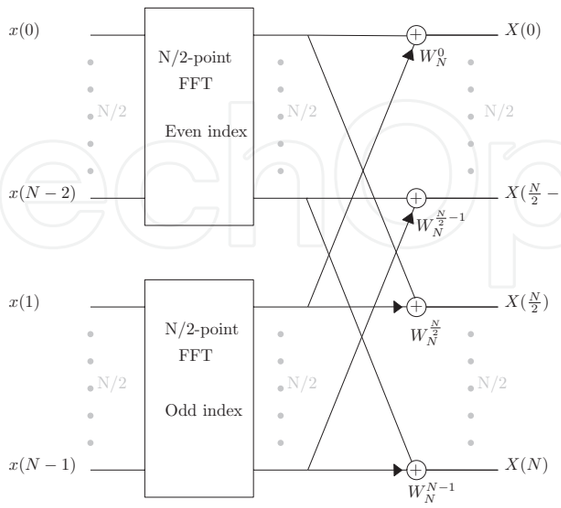


Fig. 1. Radix-2 Decimation In Time

Figure (1) represents the computation of the previous DIT. With this structure it is still necessary to perform N multiplications by the twiddle factors W_N^k to compute the N FFT outputs. However it is possible to use a property of the twiddle factors expressed in (5), to remove half of these multiplications, by replacing some additions by substractions.

$$W_N^{x+\frac{N}{2}} = -W_N^x \tag{5}$$

Then the opimized DIT decomposition will be computed as presented in figure (2)

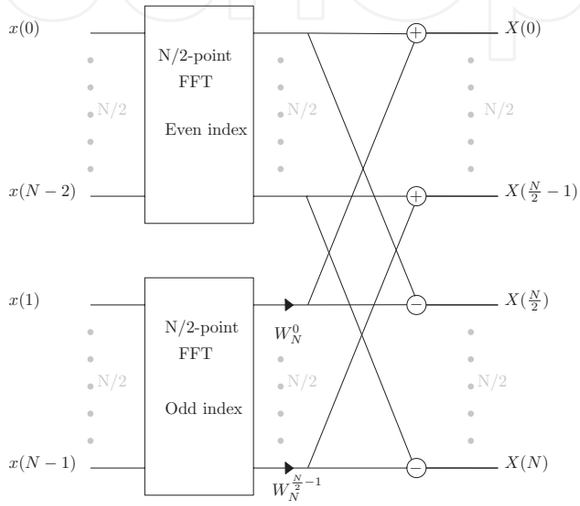


Fig. 2. Simplified Radix-2 Decimation In Time

Figure (3) presents the computation scheme for the DIF decomposition. In this last one, it appears that the N time samples are precomputed in order while the frequency samples are reordered in an even indexed group and an odd indexed group. This decomposition also requires $\frac{N}{2}$ multiplications by the twiddle factors and use the same Radix-2 butterfly in pre-computation that was used for the post computation of the DIT computation.
For these last two schemes, the algorithmic complexity is again reduced of $\frac{N}{2}$ multiplications but is still $\mathcal{O}(\frac{N}{2}^2)$. However, if the FFT size is power of 2, this decomposition can be performed until requiring $\mathcal{O}(\frac{N}{2}log_2(N))$ 2-points FFT butterflies. Because the computation of a 2-points FFT, represented figure (4), does not involve any multiplication, such decomposition requires only multiplications by the twiddle factors. This leads to an algorithmic complexity of $\mathcal{O}(\frac{N}{2}log_2(N))$.
Figure 5 shows the result of such a decomposition for a 8-points FFT with both decimations. The first thing that we can observe on this picture is that many of $\frac{N}{2}log_2(N)$ twiddle factors are $W_N^0 = 1$ which allows a reduction of the algorithmic complexity. A second important point is the high regularity in the data flow mapping. The data reordering is also very regular and can be determined by the "bit reversed" algorithm.
While all this study has been led for the Radix-2 algorithm, it is possible to generalize it to other Radix, and to the Radix-4 in particular.

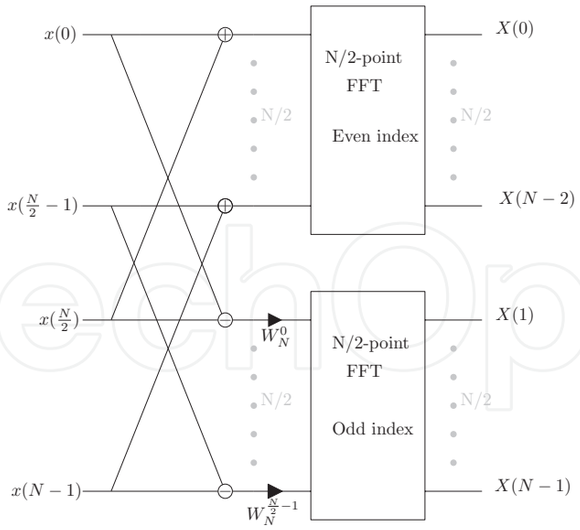


Fig. 3. Radix-2 Decimation In Frequency

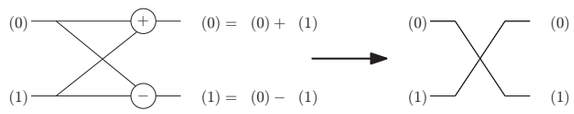
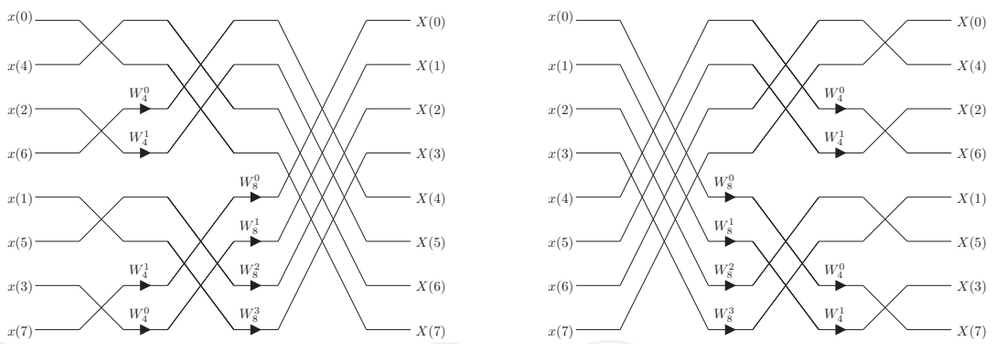


Fig. 4. Radix-2 Butterfly



(a) DIT (b) DIF

Fig. 5. 8-points FFT by using Radix-2 algorithm with DIT or DIF

2.1.2 Radix-R algorithm

As the Radix-2 butterfly computes a FFT of size 2, the Radix-4 butterfly does the same for FFTs of size 4. The four outputs of this butterfly only need trivial multiplications by $\{-1, 1, -j, j\}$ as presented in the equation (6).

$$\begin{aligned} X(0) &= x(0) + x(1) + x(2) + x(3) \\ X(1) &= x(0) + jx(1) - x(2) - jx(3) \\ X(2) &= x(0) - x(1) + x(2) - x(3) \\ X(3) &= x(0) - jx(1) - x(2) + jx(3) \end{aligned} \tag{6}$$

This butterfly however presents a gain regarding the algorithmic complexity when it is used instead of the Radix-2 by reducing the necessary number of stages in the decomposition. The major drawback of using this algorithm is that it is suitable for FFT sizes that are power of 4, which reduces considerably the field of applications.
As with the Radix-2 or the Radix-4 algorithm, it is possible to use the Radix-R algorithm for other R values. Nevertheless, the Radix-R butterfly will require no trivial multiplications into the butterfly and thus increase its algorithmic complexity. The equation 7 shows that Radix-3 butterfly computation requires 4 complex multiplications by coefficients around the unit circle. Pay attention not to confuse these coefficients with the twiddle factors needed between the stages.

$$\begin{aligned} X(0) &= x(0) + x(1) + x(2) \\ X(1) &= x(0) + W_3^1.x(1) + W_3^2.x(2) \\ X(2) &= x(0) + W_3^2.x(1) + W_3^1.x(2) \end{aligned} \tag{7}$$

Except for R multiple of 2, where it may be decomposed by using $R = 2$ or $R = 4$, a Radix-R butterfly requires $(R-1)^2$ muliplications and $R(R-1)$ additions. Thus the complexity becomes $\mathcal{O}(R^2 log_R(N))$. This is the reason why sizes which are power of two are promoted in most of digital communication standards.

2.2 Mixed-radix algorithm

The Radix-2 algorithm proposed by Cooley and Tukey can be derived to provide the mixed-Radix algorithm (R. C. Singleton, 1969) (G. L. DeMuth, 1989). This last one employs the combination of different Radix-R algorithms allowing the computation of more FFT sizes. For example, to compute a 12-points FFT, the size N can be decomposed in $N = 2 \times 2 \times 3$. In this configuration, the FFT will be computed by a first and a second stage of six Radix-2 butterflies and a third stage of four Radix-3 butterflies. Nevertheless this FFT can also be computed by other decompositions such as $N = 2 \times 3 \times 2$, $N = 3 \times 2 \times 2$, $N = 3 \times 4$ or $N = 4 \times 3$. Due to the reduced number of computation stages and to the optimal complexity of the Radix-4 butterfly, the last two decompositions will require the lower complexity.
To summarize, the complexity of a Radix or Mixed-Radix algorithms will depend on the complexity of the Radix-R butterflies that are used and the number of twiddle factors necessary between the stages.

2.3 Advantages of the Winograd fourier transform algorithm

In 1978, S. Winograd presented his works on the computation of the DFT by using cyclic convolution and introduced the Winograd Fourier Transform Algorithm which reduces the algorithmic complexity of the FFT. This reduction is achieved thanks to two points.
The first gain comes from the computation of butterfly of size R, which may be computed with a complexity of $\mathcal{O}(R)$ instead of $\mathcal{O}(R^2)$. The algorithms for $R \in \{2, 3, 4, 5, 7, 8, 9\}$ are available in (S. Winograd, 1978).
The second optimization is performed thanks to an ingenious mapping of the data flow. This mapping does not implement any twiddle factors between stages. However, this mapping can only be realized for FFT sizes which can be decomposed in several terms R_i that must be mutually primes. This is the major constraint of this type of optimization.

For a FFT of size N that can be decomposed in $N = R_1 \times R_2$ which are mutually primes, the mapping will be determined by the following rules.
First the input data vector $[x(0)...x(N-1)]$ have to be reordered in a two dimensionnal matrix of size (R_1, R_2). The position (r_n, c_n) of the sample $x(n)$ in this matrix is given by the relation (8).

$$\begin{aligned} r_n &= n \mod R_1 \\ c_n &= n \mod R_2 \end{aligned} \tag{8}$$

Then the computation of the Winograd butterfly R_1 will be computed over the R_2 groups of data which will be taken column-wide and similarly for the R_2 butterfly.
If the FFT size has to be decomposed in $I > 2$ terms (still mutually primes), this mapping can be applied with a two dimentionnal matrix of size $(R_1, \prod_{i=2}^{I} R_i)$. Thus the computation of the $\prod_{i=2}^{I} R_i = \frac{N}{R_1}$ will be performed using the same algorithm with the dimensions $(R_2, \prod_{i=3}^{I} R_i)$ for the R_1 groups generated by the first grouping.

2.4 Combined radix and WFTA algorithms

As the decomposition of the FFT in small-N FFTs was explained previously, it is possible to regroup and compute these small-N FFT as desired. So it is possible to compute a part of the FFT using the WFTA algorithm and the other part using the Mixed-Radix algorithm. However the decomposition has to be properly performed in order to take advantage of the properties of these two algorithms.
As the WFTA has better performances than the Radix algorithm, it is preferable to first regroup the mutally primes factors and compute a small FFT with the WFTA. Then the other terms will require a Mixed-Radix mapping. However the first advantage of the WFTA which reduce the complexity of the butterflies have to be use for Radix-R butterflies when $R \notin \{2, 4\}$[1].
Another optimization can be achieved by an efficient organization of the Radix/WFTA FFT. The number of *different* twiddle factors Nb_W that will have to be stored in memory for the FFT computation can be reduced if the different stages are well organized. This number of *different* coefficients is $\mathcal{O}(\prod_{i=1}^{i_0}(R_i)$ for the stage i_0, $i_0 \neq 1$ with the DIT computation, or for the $I - i_0$, $i_0 \neq 1$ for the DIF. For $i_0 = 1$ all the coefficients are $W_N^0 = 0$.

3. Proposed architecture

3.1 Presentation

In order to design the most flexible architecture capable of implementing the FFT, we decomposed the algorithm in several WFTA or Radix blocks depending on the considered N size. For example, a 64-points FFT may be decomposed in $N = 4 \times 4 \times 4$ computation stages, each stage being configured using the *Radix-4* butterfly. In the case of a 3780-points FFT, it may be decomposed in $N = 3 \times 3 \times 3 \times 4 \times 5 \times 7$ which leads to implement the *WFTA-3*, *Radix-4*, *WFTA-5* and *WFTA-7* butterflies (Z.-X. et al., 2002).
Such decompositions make it possible to consider a reconfigurable WFTA/Radix pipelined architecture that allows to compute several sizes of FFT by reconfiguring each stage's butterfly.

[1] in fact the WFTA butterfly complexity is the same as the Radix butterfly for $R = 2$ or $R = 4$

In this section, we focus on the presentation of an architecure dedicated to a multiple Digital Terrestrial Television Broadcasting (DTTB) demodulation. The concerned sandards are DVB-T (ETSI, 2004), ISDB-T (ARIB, 2001) and DTMB (Chinese National Standard, 2006) for which the possible FFT sizes are 2048 (2K), 4096 (4K), 8192 (8K) and 3780 points. Nevertheless, it is possible to adapt the architecture to other FFT sizes. Table (1) presents a way to decompose these sizes by implenting the different Radix or WFTA butterflies. The presence of a Radix-8 butterfly is necessary to compute the 8192-points FFT in six stages, but another stage of computation can be added in order to split this butterfly in two by using Radix-4 and Radix-2 butterflies.
As shown in section 2.4, the number of different twiddle factors coefficients that will have to be computed depends on the placement of the different butterflies. To provide the output samples in order, a DIT computation is considered. With such a decomposition, the quantity of required twiddle factors will be minimized by placing the higher radix butterflies at the last stages of computation. Moreover, as the second advantage of the WFTA can be employed in the 3780 decomposition for the $3 \times 4 \times 5 \times 7 = 420$ part, all these 420-points FFT computations have to be processed in the last stages.

	Butterflies Configuration					
	Stage 1	Stage 2	Stage 3	Stage 4	Stage 5	Stage 6
3780	WFTA-3	WFTA-3	WFTA-3	Radix-4	WFTA-5	WFTA-7
2048	Radix-2	Radix-4	Radix-4	Radix-4	Radix-4	Radix-4
4096	Radix-4	Radix-4	Radix-4	Radix-4	Radix-4	Radix-4
8192	Radix-4	Radix-4	Radix-4	Radix-4	Radix-4	Radix-8

Table 1. A possible decomposition of the concerned FFT sizes

Thus, such a decomposition in six stages of computation may be performed by the 6-stages reconfigurable architecture presented in figure (6).
Each stage is built on a reconfigurable WFTA/Radix module implementing the corresponding butterfly. The module may be reconfigured depending on the chosen FFT size. The butterfly input and output data are stored in the symbol memory. The address decoders select the memory address to read the inputs and write the results. A twiddle block is also implemented at each stage of the computation (except for the last one). This block is located between the butterfly module and the symbol memory and performs the multiplications by the appropriate twiddle factors. The input and output buffers are used to respectively store the time domain and frequency domain OFDM samples. Finally, the "memory data and address" multiplexers switch the memories between stages of computation.

3.2 Operating description

The N complex samples corresponding to the S_n OFDM time domain symbol are stored in the input buffer. During this storage, the WFTA/Radix module of the first stage processes its computation according to the required configuration (Radix-2, WFTA3 or Radix-4). This process is performed on the N previously stored samples that constitute the S_{n-1} OFDM symbol.
Each subsequent module performs the same operation in parallel until the WFTA/Radix module in the stage 6, that operates on the S_{n-6} OFDM symbol. While these computations are performed, the frequency domain samples corresponding to the FFT of the S_{n-7} are available at the output buffer.

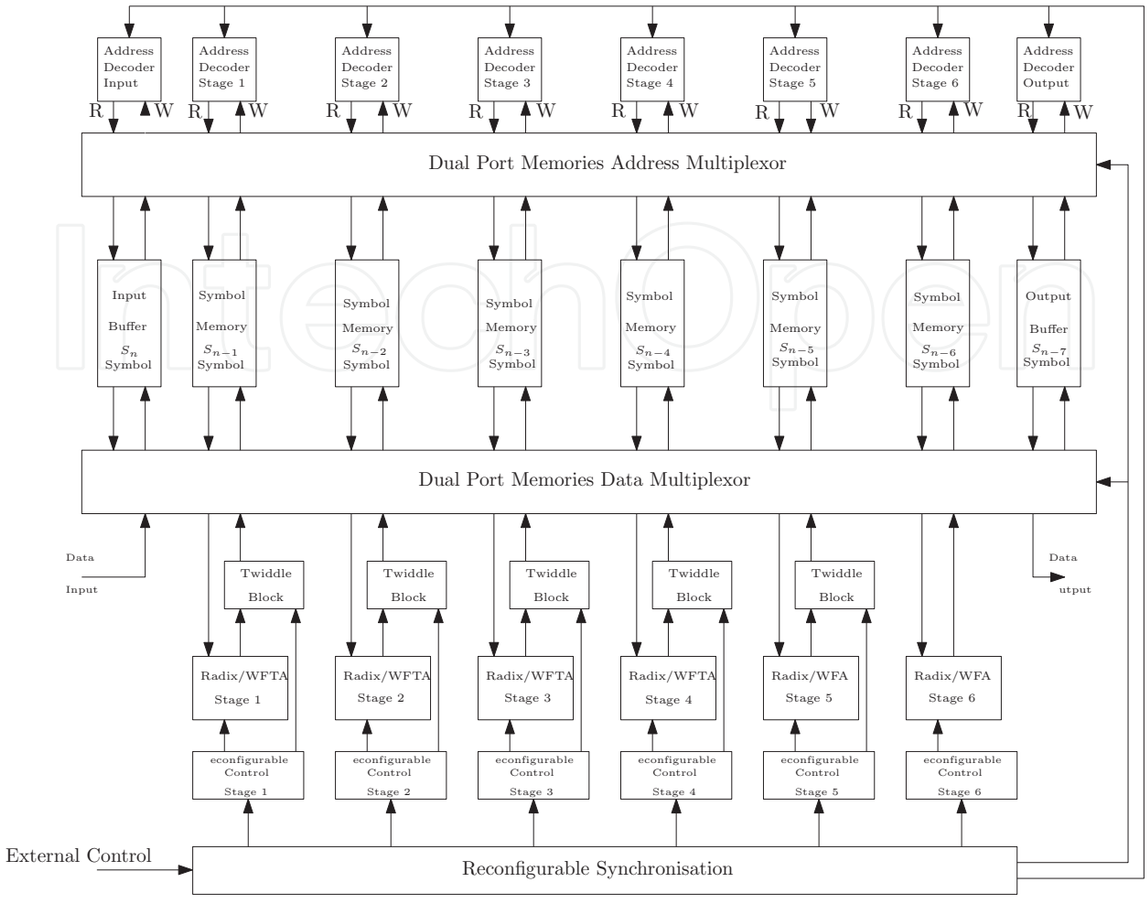


Fig. 6. 6 stages reconfigurable architecture for FFT

As soon as the S_n symbol is completely stored, the different computation stages must be completed. Memories are then circularly switched i.e. the previously input buffer becomes the symbol memory 1, the symbol memory i becomes the symbol memory $i+1$ and the output buffer becomes the new input buffer. This permutation is performed by the two memory multiplexers, one for the data bus, the other for the address bus.
When the algorithm requires the twiddle factors multiplications, these operations are computed by the twiddle block at the output of the butterfly, before being stored in the symbol memory.

3.3 Presentation of the Processing Blocks

3.3.1 WFTA/radix module

Each WFTA/Radix module is composed of three stages. A first stage aims to compute additions, a second performs the multiplications by coefficients and the last adder stage generates the output results. The figure (7) presents the architecture of such a module.
First, the set of butterfly inputs are stored in the butterfly input register. As soon as possible, the adder/substractor starts to process two values. These values are either the time domain samples stored in the register or a previous result which comes from a reconfigurable delay block. The second stage performs a multiplication of the results coming from the first stage by the coefficients which are stored in the coefficient memory. As the coefficients are either real or

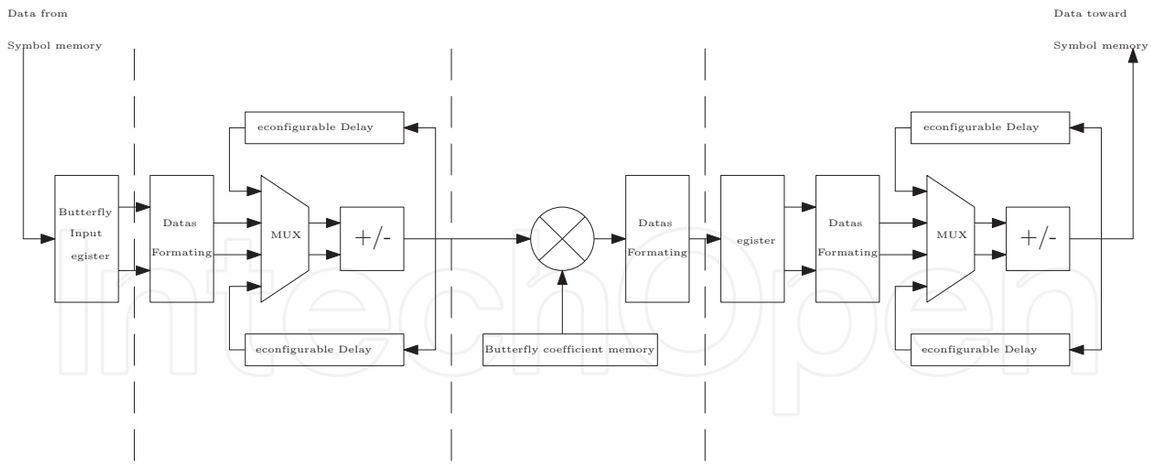


Fig. 7. Architecture of a WFTA/Radix module

pure imaginary, only two real multipliers are necessary to implement the complex multiplier. Since there are only few coefficients, their resource utilization is negligible compared to the resources required by symbol memories. Finally, the output adder/substractor part operates similarly to the first one and provides the butterfly outputs.

3.3.2 Twiddle blocks

As no twiddle factors are needed at the output of the last stage, the twiddle block does not appear on the figure 6. The architecture of the twiddle blocks is depicted in figure (8). This block just performs the complex multiplications by the twiddle factors stored in the associated memory. When multiplications are not necessary, a multiplication by $W_N^0 = 1$ is computed to maintain a constant propagation delay through the architecture.
Such a multiplication can be implemented by either four real multipliers and four real adders or three real multipliers and five real adders.

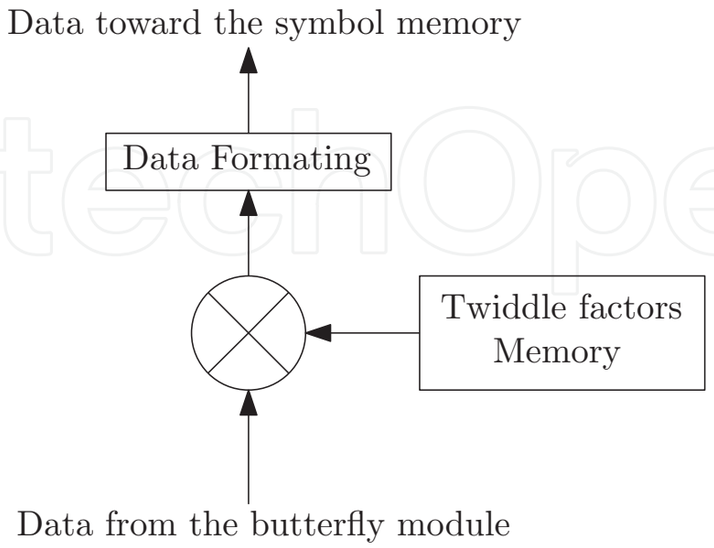


Fig. 8. Twiddle block

3.3.3 Symbol memories, address decoders

The symbol memories have to be connected to the right stage depending on the computation step. Then, during the computation stage, the memory will send data to the butterfly while recording data from the twiddle block. The time domain samples are overwritten by the frequency domain samples. Nevertheless, the read and write addresses are always different at the same time, due to the butterfly latency. For that purpose, a Simple Dual Port Random Acces Memory (SDP-RAM) have been implemented. The size of this Random Access Memory (RAM) has been defined according to the maximum number of points to be computed in the design (8192 samples). Each complex sample is coded into 32 bits (16 bits for the real part and 16 bits for the imaginary part). Thus, the size of the symbol memory has been set to $2 \times 16 \times 8192$ bits which corresponds to $32kB$.
The address decoders compute the right sequence of memory address that have to be read and written. Thus an address decoder block is dedicated to a stage of computation. As the size of the FFT can be reconfigured, these blocks also have to be reconfigured.

3.4 Performances of the 6-Stages architecture

The complete architecture has been simulated and implemented on an Altera stratix EP3SE50F484C2. This FPGA has been envisaged as a prototyping circuit for future implementation in an Application Specific Integrated Circuit (ASIC). The considered FPGA exhibits a lot of logical resources as well as dedicated blocks that are optimized for MAC (Multiplications/Accumulations) operations and memory. Thus, it is perfectly suited for the considered types of applications.

3.4.1 Resources usage

Implementation results for the 6-stages architecture are provided in Table (2). In this table, the implemented architecture is compared to commercial IPs provided by Altera (Altera, 2009) in terms of resource usage.

	Logic Cells	Registers	Memory bits	DSP blocks
Available on Stratix III	38,000	38,000	5,455,872	384
6-stages architecture	4,077	2,544	3,769,080	44
Altera 2K IP	4,138	6,943	208,329	40
Altera 4K IP	4,557	7,530	425,563	40
Altera 8K IP	5,270	8,670	884,785	48

Table 2. Resources Utilization for the 1-Stage and the 6-Stages Architecture After Implementation on Stratix III

It seems important to notice that the proposed architecture requires less logic cells, registers and DSP block than its counterparts. This is especially true if we consider that this architecture is also able to compute different sizes of FFT (2K, 3780, 4K, 8K).
Another interesting point is that the memory resource is the most consuming in this architecture. This is mainly due to the parallelization of several stages in the architecture. Thus the equivalent of six OFDM symbols must be stored in memory.
Globally, implementation results demonstrate the feasibility of implementing such architectures on reconfigurable circuits without consuming too many resources. This makes

it possible to envisage the implementation of this architecture onto ASICs which generally present higher performances and exhibit more logical resources.

3.4.2 Time performances

The time that is required to store a complete symbol into the input buffer RAM is the OFDM symbol duration T_{OFDM}. This duration depends on the sampling frequency, the size of the IFFT and the guard interval duration, used for the transmission. The minimun T_{OFDM} and associated maximum OFDM throughput for each standard and FFT sizes are provided in Table (3).

		DVB	ISDB	TDMB
3780	Duration (μs)	-	-	555
	Throughput ($symb/s$)	-	-	1802
2048	Duration (μs)	231	259	-
	Throughput ($symb/s$)	4329	3861	-
4096	Duration (μs)	462	519	-
	Throughput ($symb/s$)	2165	1927	-
8192	Duration (μs)	924	1039	-
	Throughput ($symb/s$)	1082	962	-

Table 3. Minimum OFDM symbol duration with Guard Interval and corresponding throughput

The architecture must, at least, respects these performances to be suitable in the OFDM receiver. Table (4) describes the number of clock cycles required to compute a complete stage of the FFT. According to this table, it may be seen that the WFTA-7 stage demands the most important computation time per sample.

FFT size	3780	2K	4K	8K
WFTA7	11351	-	-	-
WFTA5	7570	-	-	-
WFTA3	3787	-	-	-
Radix-8	-	-	-	16398
Radix-4	3787	2055	4103	8199
Radix-2	-	2055	-	-

Table 4. Module Execution Time in Number of Clock Cycles.

In order to provide a comparison between the proposed architecture and existing solutions, several implementation tests have been led. The obtained results are presented in Table (5) which summarizes the symbol rate as well as the computation latencies for the proposed architecture according to different configurations. The comparison has been performed with different IPs of Altera implementing the corresponding sizes of DFT. The maximum rate that may be achieved is conditioned by the WFTA/Radix Module 6 which exhibits the most important complexity.
Regarding Table (5), for each FFT size, the 6-stages architecture delivers a similar symbol rate than the commercial IPs. Note that a significant difference remains for the 8192 points FFT. This is due to the internal structure of our proposed architecture that only consists of 6 stages

		Proposed 6-stages Architecture	Altera IP
Symbol rate @ 100MHz (OFDM symbols / s)	2K	48,661	48,828
	4K	24,372	24,414
	8K	6,098	12,207
	3780	8,809	NC
Latency (clock cycles)	2K	10,282	6,144
	4K	20,522	12,288
	8K	49,201	24,576
	3780	30,289	NC

Table 5. Symbol Rate and Latency According to Different Configurations

of computations. In order to compute a 8K FFT, it is then necessary to use a Radix-8 Module which requires much more computing cycles compared to Radix-4 or Radix-2 Modules.
Concerning the latency, an increase has to be deplored in the proposed architecture. This is mainly due to the time that is necessary to store a complete symbol into an intermediate buffer between two consecutive stages of computation. Nevertheless, the drawback of latency may be compensated by the high symbol rate which is available at the output of the design.

4. Optimization for the studied applications

The previous architecture is optimized for a high symbol rate. However, it is possible to modify it by reducing the number of stages from six to one as described in figure (9).

4.1 Operating description

According to the figure (9), during the storage of a current symbol in the input buffer, the unique WFTA/Radix Module will be reconfigured to compute the different stages of the whole FFT. The connected symbol memory is employed for the computation of all stages. At the end of the computation, the input buffer becomes the new symbol memory, the symbol memory becomes the output buffer with the FFT transform results and the previous output buffer becomes the new input buffer which will store the new incoming OFDM symbol.
This operating mode allows a flexible number of computation stages. Thus the 8192-points FFT may be computed by using only *Radix-4* and *Radix-2* algorithms in seven stages of computation ($8192 = 4 \times 4 \times 4 \times 4 \times 4 \times 4 \times 2$).
The used module can be implemented in 6 possible configurations: *Radix-8*, *WFTA7*, *WFTA5*, *Radix-4*, *WFTA3* and *Radix-2*. All combinations of these possibilities can be envisaged to compute several FFT sizes. The number of stages is related to the number of terms in the FFT size decomposition. Nevertheless, an increase of this number implies a reduction of the symbol rate.
Furthermore, for particular sizes of FFT which are neither power-of-2 nor adapted to the WFTA mapping algorithm, new Twiddle factors have to be calculated and stored in the twiddle block memory. In order to take into account all possible configurations, it is possible to implement an external memory which contains all these coefficients and load them in the Twiddle memory block according to the chosen FFT size.

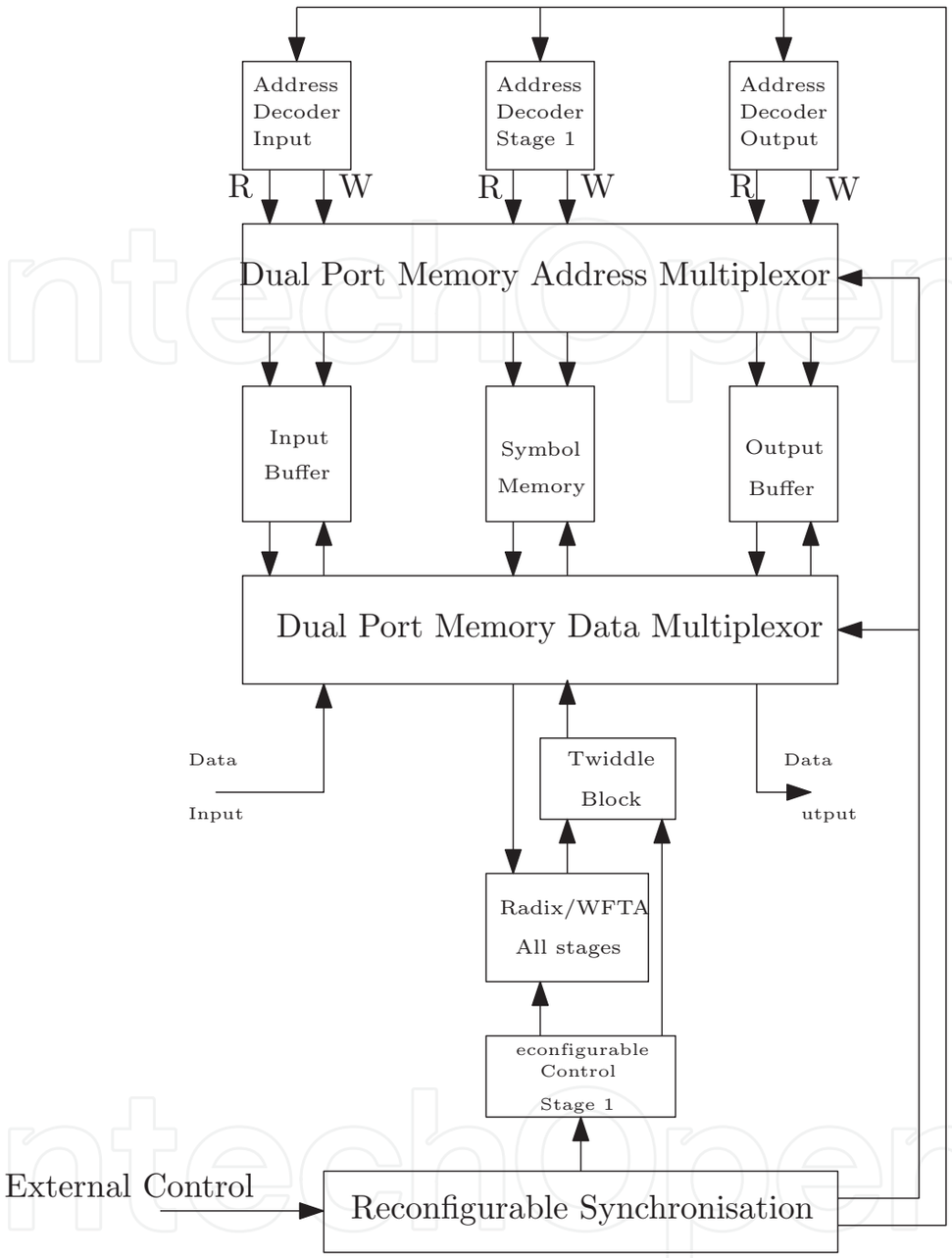


Fig. 9. 1-stage reconfigurable architecture for FFT

4.2 Reconfiguration of the architecture

As the reconfiguration of the WFTA/Radix Modules for the 6-stages architecture occurs only when the FFT size is modified, the reconfiguration is completely static. On the other hand, the Butterfly module of the 1-stage architecture must be reconfigured during runtime. This reconfiguration is possible almost instantaneously. In fact, only the control of the resources has to be reconfigured. Thus it is possible to reconfigure it by using a multiplexed control.

4.3 Resources Usage for the 1-Stage architecture

The 1-stage architecture has been implemented into the same FPGA target. The corresponding resources usage is summarized in Table (6). In this configuration, the obtained results are more competitive than Altera IPs, especially for memory usage. This may easily be seen when comparing the number of memory bits in the 8K mode. Moreover, it is clear that logical resources are reduced significantly.

	Logic Cells	Registers	Memory bits	DSP blocks
Available on Stratix III	38,000	38,000	5,455,872	384
1-stage architecture	811	548	622,896	8
Altera 2K IP	4,138	6,943	208,329	40
Altera 4K IP	4,557	7,530	425,563	40
Altera 8K IP	5,270	8,670	884,785	48

Table 6. Resources Utilization for the 1-Stage and the 6-Stages Architecture After Implementation on Stratix III

4.4 Timing considerations for the 1-Stage architecture

The execution times and latency that are depicted in Tables (4) and (5) are given for the 6-stages architecture but are still available for the 1-stage architecture. On the other hand, a comparison between the symbol rate performances achieved by the 1-stage architecture and the required symbol rate imposed by the DTTB standards are provided in Table (7).

FFT Size	1-stage architecture @ 100MHz	DVB-T/H	ISDB-T	TDMB	FLO
2048	8110	4329	3861	-	-
4096	4062	2164	1926	-	1200
8192	1742	1082	962	-	-
3780	2935	-	-	1801	-

Table 7. OFDM Symbol Rate Comparison Between Studied Standards and Proposed Architecture

According to these results, we may conclude that the optimized structure is still suitable for the FFT computation of the considered standards. However a compomize is always possible by combining the two methods seen in sections 3 and 4

5. Conclusion

In this chapter, we have presented two algorithms that may be combined to compute a FFT. Depending on the size of this transform, some advantages can be exploited by taking into account a meticulous organization in the algorithms combination. Thus, an optimal reduction of the algorithmic complexity will imply an optimal use of the hardware resources. Moreover, an architecture has been proposed for the computation of these algorithms. This architecture is able to deal with a large amount of FFT sizes, decomposable in product terms that are 2,3,4,5,7 or 8. A growth either of the largest FFT size or of the number of reconfigured sizes imply the use of more memory resources which is the most delicate point of the architecture.

A major advantage of this architecture is its possibility to be dynamically reconfigured from one to another FFT size. This allows to reuse the same circuit to compute several FFT sizes within the same modulation/demodulation standard, as it is necessary with the DTMB standard or the SC-FDMA modulation. As explained in section 4, this architecture provides a high flexibility that permits to achieve the best compromize between hardware resources and computation throughput. For prototyping purposes, this architecture has been successfully simulated and implemented in a FPGA. Nevertheless, the architecture targets an implementation in ASIC circuits whose technology exhibits higher performances than FPGA. Furthermore, the field of application is wide for this architecture. As mainly expressed, the convergence of multiple standards in a same device may be performed without implementing several modulation or demodulation circuits. Another example may be to envisage an implementation of OFDM/MIMO (Multiple Input Multiple Output) systems since the number of FFTs directly varies with the number of antennas in this type of applications. Finally, these concepts may also be used in a software radio context since it deals with reconfiguration of resources according to the channel or to environmental changes.

6. References

Weinstein, S. B. & Ebert, P. M. (1971). Data transmission by frequency-division multiplexing using the discrete fourier transform, In : *IEEE Transactions on Communication Technology, vol 19, pp. 628-634*

J. Wang, Z. Yang, C. Pan, M. Han & L. Yang (2003). A combined code acquisition and symbol timing recovery method for TDS-OFDM , In : *Broadcasting , IEEE Transactions on, vol. 49, pp. 304-308*

I. J. Good (1958). The interaction algorithm and practical Fourier analysis , In : *J. R. Statist. Soc., ser. B, vol.20, pp. 361-372*

P. Duhamel & M. Vetterli (1990). Fast Fourier transforms: a tutorial review and a state of the art, In : *Signal Processing, vol. 19, pp. 259-299*

J. W. Cooley & J. W. Tukey (1965). An algorithm for the machine calculation of complex fourier series , In : *Proc. Math. Comp., pp. 297-301*

S. Winograd (1978). On computing the Discrete Fourier Transform, In : *Proc. Math. Comp., pp. 175-199*

C. M. Rader (1968). Discrete Fourier transforms when the number of data samples is prime, In : *Proc. IEEE, vol. 56, pp. 1107-1108*

G. Bruun (1978). z-Transform DFT filters and FFTs, In : *IEEE Trans. on Acoustics, Speech and Signal Processing (ASSP), vol. 26, pp. 56-63*

R. C. Singleton (1969). An algorithm for computing the mixed-radix fast fourier transform, In : *IEEE Transactions on audio and electroacoustics, vol 17, NO.2, pp. 93-103*

G. L. Demuth (1989). Algorithms for defining mixed radix FFT flow graphs, In : *IEEE Transactions on acoustics, speech and signal processing, vol 37, NO.9, pp. 1349-1358*

ETSI (2004). Digital Video Broadcasting-Terrestrial (DVB-T); Framing structure, channel coding and modulation for DTV, In : *ETSI standard, Nov. 2004*

ARIB (2001). Integrated Services Digital Broadcasting-Terrestrial (ISDB-T); specification on channel coding, framing structure and modulation, In : *ARIB standard, May 2001*

Chinese National Standard (2006). Framing Structure, Channel Coding and Modulation for Digital Television Terrestrial Broadcasting System, In : *Chinese National Standard GB 20600-2006*

Z.-X. Yang, Y.-P. Hu, C.-Y. Pan & L. Yang (2002). Design of a 3780-point IFFT processor for TDS-OFDM, In : *Proc. IEEE, vol. 48, pp. 57-61*

Altera (2009). In : *FFT mega core function user guide*

Fourier Transforms - New Analytical Approaches and FTIR Strategies

Edited by Prof. Goran Nikolic

ISBN 978-953-307-232-6
Hard cover, 520 pages
Publisher InTech
Published online 01, April, 2011
Published in print edition April, 2011

New analytical strategies and techniques are necessary to meet requirements of modern technologies and new materials. In this sense, this book provides a thorough review of current analytical approaches, industrial practices, and strategies in Fourier transform application.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Florent Camarda, Jean-Christophe Prevolet and Fabienne Nouvel (2011). Towards a Reconfigurable FFT: Application to Digital Communication Systems, Fourier Transforms - New Analytical Approaches and FTIR Strategies, Prof. Goran Nikolic (Ed.), ISBN: 978-953-307-232-6, InTech, Available from: http://www.intechopen.com/books/fourier-transforms-new-analytical-approaches-and-ftir-strategies/towards-a-reconfigurable-fft-application-to-digital-communication-systems

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen