A NOVEL 64-POINT FFT/IFFT PROCESSOR FOR IEEE 802.11(A) STANDARD

K. Maharatna, E. Grass and U. Jagdhold

IHP
Im Technologiepark 25, Frankfurt (Oder), Germany
Email: {maharatna, grass, jagdhold}@ihp-microelectronics.com

ABSTRACT

A novel 64-point FFT/IFFT processor is presented in this article, named TURBO64, developed primarily for the application for the IEEE 802.11(a) standard. The processor does not use any digital multiplier or RAM. It has been fabricated and tested successfully. Its core area is 6.8 mm^2 and the average power consumption is 41 mW at 1.8 V @ 20 MHz frequency. Compared to some other existing IP cores and ASIC chips TURBO64 needs a smaller number of clock cycles and consumes less power.

1. INTRODUCTION

Fourth generation wireless and mobile communication systems are currently in the focus of research and development. Consequently, much research effort is devoted to realize a system compliant with the IEEE 802.11(a) standard which essentially forms the baseline for fourth generation wireless communication [1]. Recent studies show that the most computation intensive parts of an IEEE 802.11(a) compliant modem are the 64-point FFT/IFFT and the Viterbi decoder [2]. In this paper, we concentrate on the optimum implementation of the 64-point FFT/IFFT module.

IEEE 802.11(a) standard specifies that the 64-point FFT/IFFT has to be computed in 3.2 μs. Thus, using the conventional Cooley-Tukey butterfly algorithm [3], which needs 192 complex multiplication operations, one complex multiplication has to be completed within 16.6 ns to satisfy this timing constraint. This corresponds to a frequency of 60 MHz. In our in-house 0.25 μm BiCMOS process one complex multiplier unit consumes 0.18 mm^2 silicon area and 21mW power at that frequency. This resulting excessive power dissipation is not applicable for wireless mobile applications. On the other hand, the data from the analog part of the modem arrives at a rate of 20 Msps. Thus, it appears desirable to operate the targeted processor at 20 MHz frequency. However, to satisfy the timing constraint at that frequency one needs to deploy several complex multiplier units in parallel, which results in increase of silicon area and power consumption.

In this paper, we present a new low power 64-point FFT/IFFT processor that consumes moderate silicon area and satisfies the timing constraint while operating at 20 MHz frequency. The rest of the paper is structured as follows: The theoretical background of the algorithm we have used is discussed in Section 2. Section 3 describes the architecture of the TURBO64 processor and the measurement results of the fabricated chip are described in Section 4. In Section 5, the new processor is compared with some other existing processors and conclusions are drawn in Section 6.

2. THEORETICAL BACKGROUND

The FFT A(r) of a complex data sequence B(k) of length N where r, k ∈ { 0, 1, …, N−1} can be described as,

$$A(r) = \sum_{k=0}^{N-1} B(k) W_N^{rk} \qquad (1)$$

where $W_N = e^{-2\pi j/N}$. One may formulate the radix-8 representation of the FFT in the following manner:
Let, N = 8T, r = s + Tt and k = l + 8m, where, s, l ∈ {0, 1, …, 7} and m, t ∈ {0, 1, …, T−1}. Applying these values in equation (1) and simplifying results in,

$$A(s+Tt) = \sum_{l=0}^{T-1} W_8^{lt} [W_{8T}^{sl} \sum_{m=0}^{T-1} B(l+8m) W_T^{sm}] \qquad (2)$$

For N = 64 and T = 8 the 64-point FFT can be expressed from equation (2) as

$$A(s+8t) = \sum_{l=0}^{7} [W_{64}^{sl} \sum_{m=0}^{7} B(l+8m) W_8^{sm}] W_8^{lt} \qquad (3)$$

Equation (3) shows that the 64-point FFT can be computed by first computing eight 8-point FFTs on the appropriate data slot (described in equation (3)) then multiplying them with 49 nontrivial complex interdimensional constants and then once again calculating eight 8-point FFT of the resultant data with appropriate data reordering.

The important point to be noted here is that for realization of an 8-point FFT using the Decimation In Time (DIT) butterfly algorithm, one needs not to use any explicit multiplication. The constants to be multiplied for the first two columns of the butterfly slices are 1 and j respectively. In the third slice, the multiplication by $1/\sqrt{2}$ can be realized using hard-wired shift-and-add operations.

Thus, from the implementation point of view, in the present approach 49 non-trivial complex multiplications are needed. This is 26 % less compared to that needed using radix-2 DIT FFT (66 non-trivial multiplications).

The IFFT can be performed by first swapping the real and imaginary parts of the incoming data and then performing the forward FFT on them and once again swapping the real and imaginary parts of the data at the output. This method allows to perform the IFFT without changing any internal coefficients and results in more efficient hardware implementation.

3. ARCHITECTURE OF THE TURBO64

The architecture for the targeted FFT/IFFT processor can be constructed by suitably mapping equation (3) into hardware. However, while constructing the architecture it is to be kept in mind that the FFT is an algorithm that involves global connections. For a process technology with a small number of metal layers this may lead to a routing overhead by 60%. This results in a performance degradation of the processor. Thus, it is necessary to reduce the number of global wires to a minimum. The schematic diagram of the FFT/IFFT processor is shown in Figure 1. It consists of an input unit, two fully parallel 8-point FFT units, a multiplier unit, an internal storage unit CB and the output unit. In the following subsections the operation of these modules is explained.

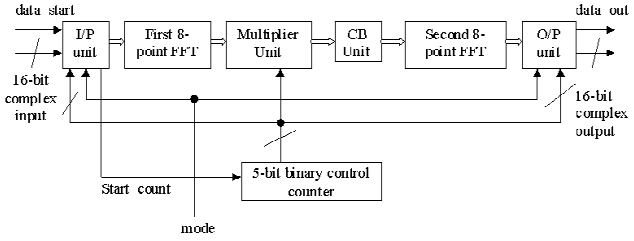

Figure 1. The architecture of the FFT/IFFT processor

Input unit: The input unit (I/P unit in Figure 1) consists of an input register bank having 16-bit wordlength that can store 57 complex samples. Three additional registers are also provided for temporary storage of data. The reason for keeping these three temporary registers will be clear during our discussion on the multiplier unit. It is equipped with two single bit signals "*mode*" and "*data_start*". While the first one enables the processor to understand which mode of operation is needed (whether FFT or IFFT), the second one indicates the presence of valid data at the input. After the assertion of the *data_start* signal, serial data are inputted at every clock cycle at the 56^{th} position of the input register bank and at every clock cycle the complex data having index i is shifted to the $(i-1)^{th}$ position where i $\in$ {1, .. 56}. The register bank is provided with eight complex 16-bit fixed wired outputs corresponding to the registers having index 8j, where j $\in$ {0, …, 7}. When the input buffer is filled, the appropriate data (see equation (3)) gets self aligned with these hard-wired outputs and is delivered parallely to the first 8-point FFT unit. This operation is continued in every cycle. However, because of the specific construction of the multiplier unit used here the downward shifting of data cannot take place at every cycle. This argument will be clear in the later part of our discussion. In such cases the input data are temporarily stored in the additional three registers before this downward data shifting operation resumes once again. A 6-bit counter controls the serial inputting of the data in the register bank which is enable with the assertion of the *data_start* signal. When the 56^{th} position of the input register bank is full a signal *Start_count* is asserted which enables the master control counter. The rest of the operations are controlled by the master counter. This method of downward shifting of data in conjunction with its self-alignment with the hard-wired outputs reduces the number of multiplexed signals by a factor of 64 and 8 at the input and output of the input unit respectively. This massive reduction of wiring allows a more efficient layout of the processor.

8-point FFT unit: It has been discussed in the Section 2, that for implementing the 8-point FFT one needs not to use conventional digital multipliers. This fact opens up the scope to implement a fully parallel 8-point FFT unit using combinatorial circuits only within a small silicon area. In our implementation, we followed this strategy. The multiplication by the factor $1/\sqrt{2}$ is realized using hard-wired shift-and-add operation. The basic 8-point FFT unit is used two times in our design thus, avoiding any feedback loop. This way, the employment of datapath pipelining becomes much easier.

Multiplier unit: The main performance bottleneck of the architecture is the multiplier unit. A fully parallel implementation of the 8-point FFT unit is the basis for a very fast implementation of the processor. However, to attain this, one has to complete 7 non-trivial complex multiplications at every cycle. At 20 MHz one complex multiplier in our library needs 0.18 mm^2 silicon area and 7mW power. Thus, employing seven complex multipliers in parallel may not be a good solution.

A close observation to the interdimensional constants to be multiplied with the 8-point FFT outputs reveals that only eight sets of them are unique. The 49 complex multiplications can be carried out using these eight sets of constants in addition to appropriate data rearrangement. Since, these eight sets of constants are already known *a priori*, one can realize each of them using hard-wired shift-and-add operation. Eight such units can be placed in parallel to complete one set of complex multiplications in one clock cycle. However, for the data coming from the first 8-point FFT unit at even time instants (t = 2, 4, 6), more than one clock cycle is needed since some of these

hard-wired constant units have to be used more than once to process the full set of data. Because of that reason, the downward shifting of the data in the input unit is suspended until the processing of the full set of 8-point FFT data coming at even time instant is finished. However, after the complex multiplication for the respective set of 8-point FFT data the downward shifting of the data in the input register bank resumes once again. In this approach one needs 12 clock cycles to complete all 49 complex multiplication operations. Compared to the approach with conventional complex multipliers, this approach results in a reduction of area and power. The synthesized cell area and power dissipation of a multiplier unit consisting of 7 complex multipliers in parallel is 1 mm^2 and 30 mW respectively at 20 MHz frequency. On the other hand, the synthesized cell area and power consumption of the multiplier unit in our solution is 0.6 mm^2 and 19 mW respectively at 20 MHz frequency. Another advantage of using the hard-wired constants is that, no separate coefficient storage is required. Thus, the proposed solution seems to be more efficient compared to the conventional multiplier-based approach.

CB unit: This is an internal register bank that can store 64 complex data. After the multiplication operation, the resulting data are stored in this temporary register bank from where they are delivered to the second 8-point FFT unit in a reshuffled manner. It has eight hard-wired outputs corresponding to the position 8j, where, $j \in \{0,\ldots, 7\}$. At every clock cycle the i^{th} data gets shifted to the $(i-1)^{th}$ position, where, $i \in \{1,\ldots, 63\}$. Like the operation in the input unit, this operation also automatically aligns the appropriate data with the input of the second 8-point FFT unit. This arrangement essentially reduces the wiring by a factor of 8 at the input and output of the CB unit.

Output unit: In the output unit (shown as O/P unit in Figure 1) we follow the same strategy as with the input unit. The output register bank is of length 57. In this case, the i^{th} data coming out from the second 8-point FFT unit is directly mapped to the $(8i)^{th}$ position of the output register bank (where, $i \in \{0, 1, \ldots, 7\}$) by hard-wired connection. The final output in serial form is taken out from the 0^{th} position of the output register as soon as the first data arrives. At every cycle as the new data arrives from the second 8-point FFT unit at the $(8i)^{th}$ positions of the output register bank, the old data corresponding to those positions are shifted downwards by one position and the data output mechanism proceeds in the same way. This unit shares the *mode* signal in common to the input unit and generates a single bit signal "*data_out*" that indicates the presence of valid data at the output for the next 64 cycles. Depending on the logic state of the *mode* signal the output unit either swaps the real and imaginary parts of the output vector (IFFT operation) or keeps the vector as it is (FFT operation).

4. MEASUREMENT RESULTS FOR TURBO64

The FFT/IFFT processor is fabricated using our in-house 0.25 μm 3 metal layer BiCMOS process. Our design flow consists of VHDL coding, synthesis using Synopsys Design Analyzer and floor planning & layout using Cadence Silicon Ensemble. The synthesis is done with a target frequency of 20 MHz. The synthesized cell area of the processor is 3.6 mm^2, which is equivalent to 62 k NAND gates in our library. The core area of the processor is 6.8 mm^2 after layout. It has 85 I/O pins and the complete area of the processor with pads is 13.5 mm^2.

The testing is carried out using an Agilent SoC 93000 chip tester. At 20 MHz frequency the latency of the processor is 3.85 μs (77 cycles) and the throughput is one full set of FFT result / 3.2 μs. However, the parallel to parallel FFT computation (i. e., when all the data are available at the input to the parallel output data availability) takes 1.15 μs i. e., 23 cycles. To our knowledge no 64-point FFT/IFFT processor reported so far has such a fast conversion time in terms of cycle counts.

The average power dissipation for the processor measured over 55 fabricated chips is 41 mW at 1.8 V power supply @ 20 MHz frequency and 84 mW at 2.5 V at the same frequency. The maximum frequency of operation at 1.8 V is 26 MHz and the same at 2.5 V is 38 MHz. These figures suggest that the designed processor is appropriate for low power high-speed applications like the wireless modem defined by the IEEE 802.11(a) standard. The die photograph of the chip is shown in Figure 2.

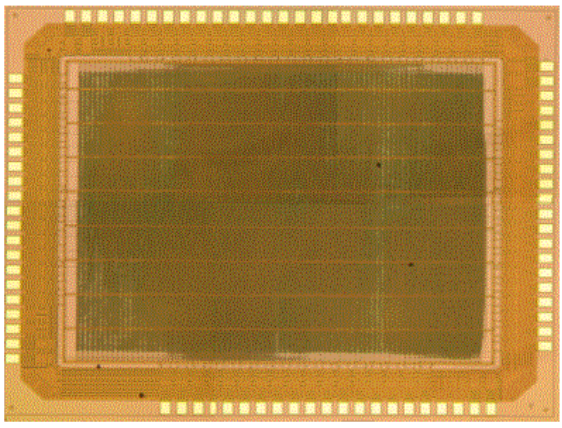

Figure 2. Die photograph of TURBO64 processor

5. PERFORMANCE COMPARISON

The performance of the processor is compared in Table 1 and Table 2 with some commercially available 64-point FFT/IFFT cores as well as fabricated ASIC chips reported in the literature respectively. From these tables the

proposed processor TURBO64 exhibits better performance in terms of the number of clock cycles, silicon area and power consumption.

6. CONCLUSIONS

In this article we presented a novel 64-point FFT/IFFT processor named TURBO64 which is compatible with the IEEE 802.11(a) standard. From the architectural point of view, the proposed design does not need conventional digital multipliers, which are normally area and power consuming components.

IP Core	Word length	Number of cycles
[4]	16	192
[5]	12	112
[6]	16	1536
[7]	16	96
TURBO64	16	23

Table 1. Comparison of TURBO64 with some commercial IP cores

Processor	Word length	Cyc-le	Techno-logy (μm)	Power (W)	Area (mm^2)
[8]	24	130	0.35	1.3	Core
[9]	16	208	2	-----	282
[10]	16	222	0.75	-----	156
TURBO64	16	23	0.25	0.041	6.8

Table 2. Comparison of TURBO64 with some fabricated FFT/IFFT processor

The processor requires a smaller number of clock cycles compared to some other commercially available IP cores. Compared to the other fabricated ASIC chips the present FFT processor exhibits better power performance and requires less silicon area. Though the processor is principally developed for application in a specific wireless modem, the performance of the proposed architecture suggests that it can be used for any application that requires fast operation as well as low power consumption. Currently, this processor is integrated with other components in a project that is aiming to implement a complete Hiperlan/2 and 802.11(a) compliant modem into a single chip.

6. REFERENCES

[1] *IEEE P802.11a/D7.0, "Draft supplement to standard [for] information technology-telecommunications and information exchange between systems - local and metropolitan area networks-specific requirements – Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: High speed physical layer in the 5 GHz band"*.

[2] E. Grass, K. Tittelbach, U. Jagdhold, A. Troya, G. Lippert, O. Krueger, J. Lehmann, K. Maharatna, N. Fiebig, K. Dombrowski, R. Kraemer, P. Maehoenen, "On the Single Chip Implementation of a Hiperlan/2 and IEEE802.11a Capable Modem", *IEEE Personal Communication*, vol. 8, no. 6, pp. 48 – 57, December 2001.

[3] J. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex Fourier series", *Math. Computation*, vol. 19, pp. 297 – 301, 1965.

[4] Xilinx Product Specification, "High-performance 64-point complex FFT/IFFT V1.0.5", *http://www.xilinx.com/ipcenter*.

[5] Altera Megafunction technical brief, "ISS High-performance 64-point FFT/IFFT", *http://www.spinnaker.co.jp/jp/datasheet/tb_fft_1_V002.pdf*.

[6] Icomm Technologies Inc. Product design data sheet, "FFT-1024 complex 1024-points FFT/IFFT processor".

[7] M3 Architecture Specification Rev. 1.3: Functional Specification, Technishe Universitaet, Dresden, 1999.

[8] J. V. McCanny, D. Trainor, Y. Hu and T. J. Ding, "Rapid design of complex DSP cores", *http://www.esscirc.org/papers-97/102.pdf*.

[9] T. Chen and L. Zhu, "An expandable column FFT architecture using circuit switching network", *Journal of VLSI Signal Processing*, Jan. 1994.

[10] T. Chen, G. Sunanda and J. Jin, "COBRA: A 100-MOPS Single-Chip Programmable and Expandable FFT", *IEEE Trans. VLSI*, vol. 7, no. 2, June 1999.