A FLEXIBLE FFT PROCESSOR

By
[1]Fredrik Kristensen, [1]Peter Nilsson and [2]Anders Olsson

[1]Digital ASIC group, Department of electroscience, Lund University
Box 118, SE-221 00 Lund, Sweden

[2]Axis Communications AB, Emdalavägen 14, S-223 69 Lund,Sweden

Telephone: +46 46 222 9108, Fax: +46 46 12 99 48, E-mail: fredrik.kristensen@es.lth.se

ABSTRACT

In this paper a flexible FFT processor for OFDM systems is presented. In future wireless systems using OFDM each terminal should be able to handle a wide range of different applications, from pure voice communication to high-speed data transfers, with as little overhead in power consumption and protocol procedure as possible. As the terminals will be battery powered and should be cheap to manufacture an ASIC solution is required. To be able to adapt to different communication schemes and various communication channels the FFT processor must be configurable. To minimize power consumption no more precision than necessary should be used. A pipelined radix2^2 FFT processor, with the possibility of run time switching between 32-1024 points, have been designed in a standard CMOS 0.35 μm technology. Clock gating and low power memories have been used to reduce power consumption.

INTRODUCTION

Around the world new standards for wireless communication with Orthogonal Frequency Division Modulation (OFDM) have been created, e.g., Hiperlan/2, IEEE 802.11a, DAB (Digital Audio Broadcast), and DVB (Digital Video Broadcast). The advantage with OFDM is its ability to combat multipath propagation. The drawback is the sensitivity to Doppler spreading and frequency shifting but this is not that a big problem if it operates in a slow fading environment [3]. The key component in an OFDM system is the FFT (Fast Fourier Transform) and its inverse IFFT. However, the FFT algorithm is a demanding task and care must therefore be taken to design an efficient implementation. Since the FFT is common in many different digital signal-processing algorithms a portable device equipped with OFDM transmission could be able to use the FFT for both the communication and other signal transformations, if the FFT processor is made flexible and fast enough. Other tasks could be preprocessing of medical data from a sensor, in order to reduce the necessary amount of data that have to be transmitted.

This paper presents a pipelined FFT processor variable from 32 to 1024 points, where unused parts is turned of with clock gating in order to reduce the power consumption. Other features are easy switching between FFT/IFFT and a halt function to temporary freeze the processor.

ARCHITECTURE

The architecture of the FFT processor is based on Decimation In Frequency (DIF) with radix2^2 stages [1], as shown in Figure 1. The advantage with the radix2^2 FFT compared to a straight forward radix2 FFT is that every other complex multiplier is replaced with a trivial multiplier that only multiplies with $-j$, i.e., the real and imaginary part swap place and the new imaginary part change sign. Compared to other radix4 structures it has the advantage that the control is simple [1] and the need of memory is the smallest possible for a pipelined FFT processor.

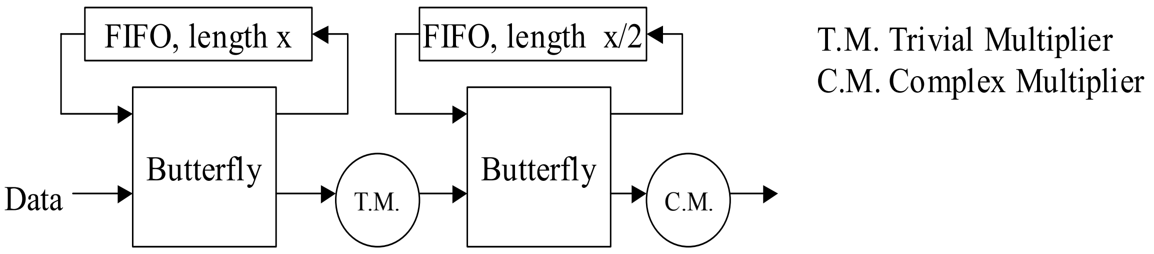

Figure 1: A radix2^2 stage. x = $2^{2*stage-1}$.

To compute the 1024 points FFT five radix2^2 stages are connected as shown in Figure 2, resulting in a memory size of 1023 words and four complex multipliers (there is no need for a complex multiplier in stage 1). This design makes it possible to do FFTs of size 64, 256, and 1024. To realize the sizes 32, 128 and 512 a radix2 stage that handle all three sizes is added to stage 5. Realizing that the radix2 stage and the radix2^2 stage 5 are never used simultaneously, parts of the hardware are reused. The last FIFO in stage 5 and the complex multiplier are shared between the radix2 and radix2^2 stages. All internal control is managed with a counter that ripples through the stages in time with the data. The counter is placed internally on the chip.

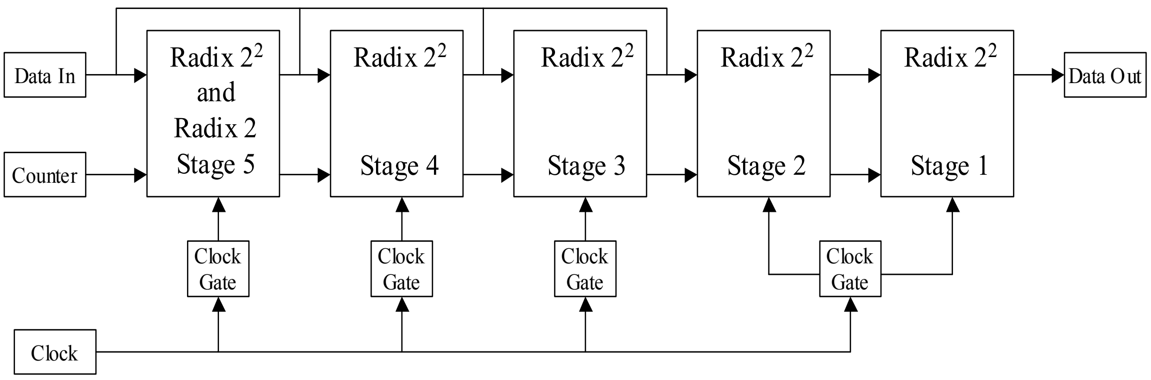

Figure 2: Architecture of the FFT processor.

The pipelined structure makes the processor fast [2]. Apart from the latency of size-1 clock cycles, the processor produces one output for each input value. The structure also supports easy reconfiguring of the size, through bypassing some of the stages. For example, in a 32 points FFT the data pass through stage 5 (the radix2 part), bypass stage 4 and 3, and finally pass through stage 2 and 1. To reduce the power consumption, the clock for the bypassed stages is gated with a clock gate [4], shown in Figure 3a. The timing is not critical as shown in Figure 3b. To turn off the gated clock for one clock cycle, the Enable signal is lowered at any time in the clock cycle and then raised again in the following clock cycle.

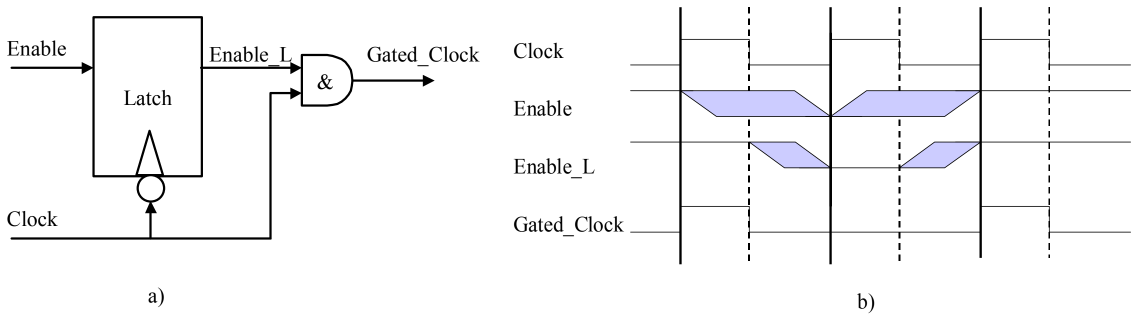

Figure 3: a) Clock gate and b) Timing diagram

BIT WIDTH AND MEMORIES

One of the more important decisions designing a pipelined FFT processor is the bit width in the architecture, since it will affect the precision, number of gates, and power consumption. The most straightforward implementation is to have the same bit width throughout the whole FFT processor. However, this will give a poor performance since the data has to be shifted down before each butterfly, to avoid overflow. In a 1024 points FFT the ten least significant bits is lost if a fixed data path is used.

The fact that the input stages in a DIF FFT contain the largest FIFOs can be used to find a better solution. If the bit width is reduced at the input and then allowed to increase towards the output the required memory as well as power consumption and area will decrease significantly. That the data path will be wide close to the output will not effect the size of the memory that much, since the last FIFOs are short, in fact the last one is only one word long. Allowing the data path to increase with one bit in each butterfly, i.e., no precision is lost in the butterfly, compensates for the loss in Signal to Noise Ratio (SNR) due to the reduced input bit width.

Figure 4a and 4b shows SNR versus memory size and number of input bits for a 1024 points FFT with a fixed and an increasing data path. The values are obtained from a C-model of a fixed and an increasing data path FFT compared to a 64 bits floating point FFT. The input data has an upper and lower limit of $\pm 1/\sqrt{2}$, to ensure that no overflow occur in the complex multiplier. The twiddle factor used in this model is 12 bits, which puts the upper limit of the SNR to about 60 dB as seen in the figure.

To get as high SNR as possible from the FFT processor the input data is normalized to the highest valid input value, in this case $1/\sqrt{2}$. To avoid a division the normalization could be obtained in an OFDM system by setting the maximum output from the A/D converter in the receiver to be $1/\sqrt{2}$.

The designed FFT has 8 input bits (real and imaginary part is 8 bits each) and the width of the data path is increasing, resulting in an 18 bits wide output. The processor has a SNR of 49 dB and use 20 Kbits of memory. The bit width is chosen to cope with signal constellations up to 16 points, e.g., 16-QAM. The input word consists of 2 bits to represent the constellation, 3 bits for soft decoding, 1 bit to ensure that the data is between $\pm 1/\sqrt{2}$, and 2 bits noise and interference margin [3].

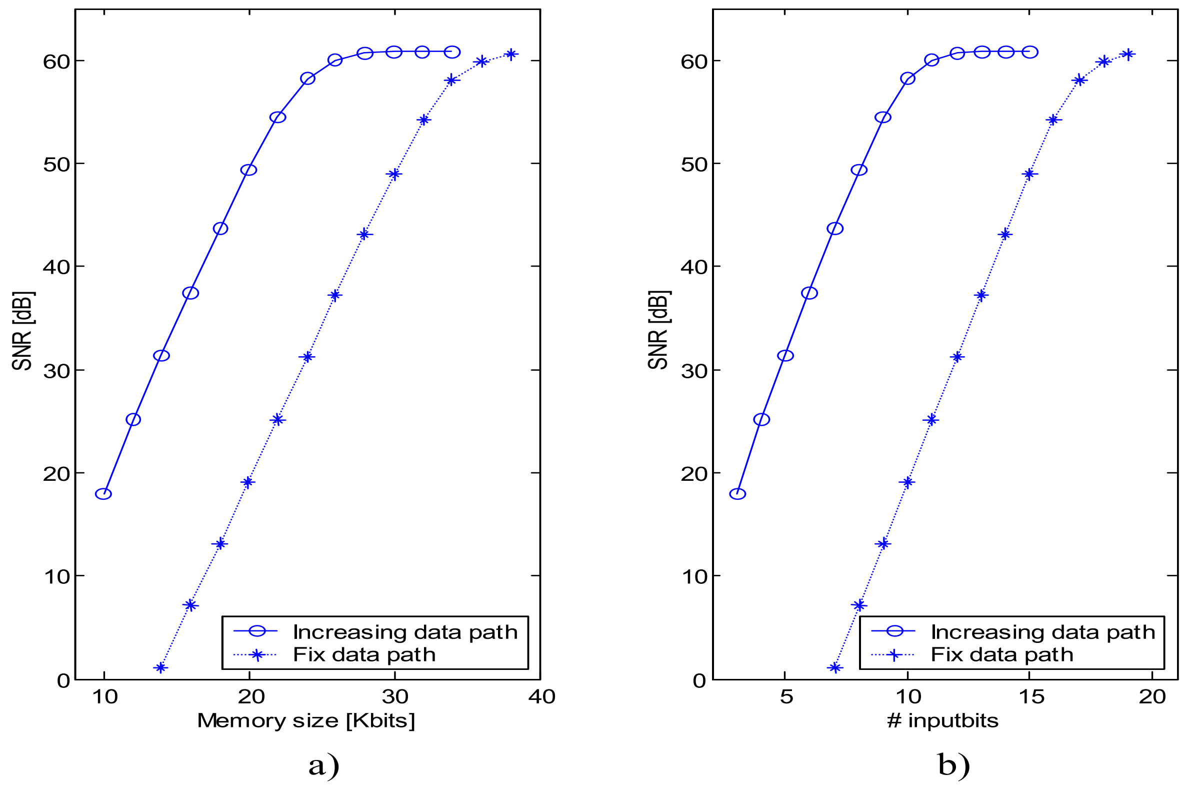

Figure 4: a) SNR versus memory size and b) SNR versus number of input bits, for a 1024 points FFT with increasing and fixed data path.

The next important decision is how to implement the FIFOs. As there is one read and one write operation in each clock cycle, the FIFO could be implemented in at least four different ways:

1. Flip-flops in series, gives an easy solution with no need of control logic. Flip-flops have a fast access time but occupy a large area, even for small FIFOs.
2. One dual-port memory. A dual-port memory is smaller than flip-flops but is still more than two times larger than two single port memories.
3. Two single-port memories of half length, with alternating read/write. A solution that is smaller than a dual-port memory, but with twice as many memories to place and route.
4. One single-port memory of half length and double width, which reads and writes every other clock cycle. The smallest solution for FIFOs larger than 400 bits, see Figure 5.

Figure 5 shows estimated area for different implementations of FIFOs in the used 0.35 μm technology. There are no interconnections included in the flip-flop area. All memory FIFOs includes control logic and a 50 μm power ring on three sides of the block. In the designed FFT processor the FIFOs were chosen according to Figure 5. All FIFOs longer then 8 words were implemented as single-port memories of double width and the remaining FIFOs were implemented with flip-flops.

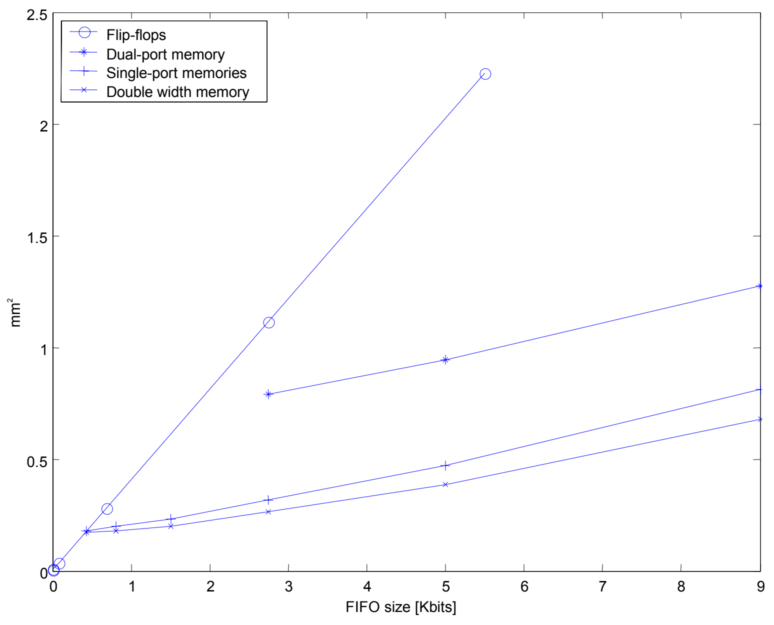

Figure 5: Area versus FIFO size for different implementations.

RESULTS

A pipelined 32-1024 points FFT processor are implemented in a standard CMOS 0.35 μm technology with five metal layers. The processor is estimated to compute a 1024 points FFT in less than 13 μs, with a clock frequency of 83 MHz. The estimated time for a 32 and 256 points FFT is, respectively, 0.4 and 3.1 μs at the same frequency. The FFT processor is resizable between 32-1024 points and unused blocks are deactivated with clock gating. The designed processor reaches a SNR of 49 dB with 8 input bits for a 1024 points FFT. Figure 6 shows the chip layout of the FFT processor. The chip has 84 pins, three twiddle factor ROMs, and 6 RAMs. The core area is 4.94 mm^2.

There are no reliable figures on the power consumption to report, since the chip was not yet returned from fabrication while this report was written, but care has been taken during the design to keep it low, e.g., low power memories are used in the FIFOs, unused stages in the FFT is turned of, and the bit width is kept narrow where it is significant to do so.

Future improvements could be to replace the current complex multiplier with a distributed complex multiplier [5], which reduce the hardware size from three to two real multipliers. A scaling and clipping block should be added to the output, in order to reform the output to desired form and size.

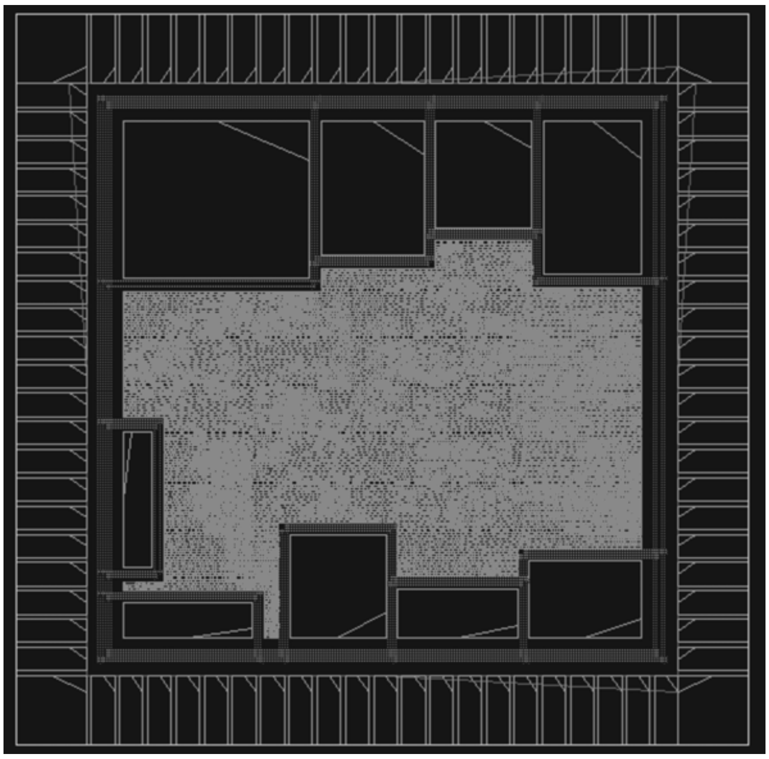

Figure 6: Chip plot of the FFT processor.

References

1. S. He, Concurrent VLSI Architecture for DFT Computing and Algorithms for Multi-output Logic Decomposition, Diss No. 133, Lund University, Sweden, 1995.
2. Weidong Li and Lars Wanhammar, "A PIPELINED FFT PROCESSOR", 1999 IEEE Workshop on Signal Processing Systems, pp. 654 –662, 1999.
3. Heiko Schmidt and Karl-dirk Kammeyer, "Quantization and its Effects on OFDM Concepts for Wireless Indoor Applications", University of Bremen, Germany. http://www.comm.uni-bremen.de.
4. Koorosh Nazifi and Gal Hasson, Industry's First RTL Power Optimization Feature Significantly Improves Power Compiler's Quality of Results, 1998, *http://www.synopsys.com/news/pubs/rsvp/spr98/rsvp_spr98_6.html*.
5. Anders Berkeman, Viktor Öwall, and Mats Torkelson, "A Low Logic Depth Complex Multiplier Using Distributed Arithmetics", IEEE journal of solid-state circuits, vol.35, NO. 4, April 2000.