Design and Implementation of Power Efficient Micro Pipelined GALS Based 2-D FFT Processor Core

Y. J. M. Shirur and V. S. Chakravarthi
BNMIT/ECE, Bangalore, India
Email: {yashamallik, scveena}@bnmit.org

***Abstract*—Today's complex SOC solutions demand low power processors. Synchronous processors which consume more than 40% of power in clock circuitry are being conveniently replaced by low power delay-insensitive (DI) asynchronous logic. In this paper, a Micro-pipelined GALS based 2D Fast Fourier Transform [FFT] Processor is designed and implemented to perform power, area and timing analysis. The implemented design has given power advantage of 78.22% and timing advantage of 39.99% when compared with similar synchronous 2D FFT processor. The design is a synthesizable core which can be extended to be part of Complex DSP architecture and hence is a right choice for any MPSOC design.**

***Index Terms*—system on chip (SoC), delay insensitive (DI), 2D fast fourier transform, micro pipelined, globally asynchronous and locally synchronous (GALS), synthesizable core, MPSoC**

I. Introduction

In current scenario most of the System on Chip (SoC)s are synchronous and operates at a single clock frequency or its derivative. In synchronous processors more than 40% of the power is consumed by clock, it is used to synchronize the different modules in MP-SoC environment. The complexity of clock routing to various modules is increasing due to clock skew issues. Major challenges in such designs are clock distribution, timing closure, scaling up of clock frequencies, power consumption, and reusability. As the trend for smaller feature and higher speed continue, clock management and power dissipation increases affecting design performance. The power inefficiencies of clocked circuit's are emerging as the dominant factor hindering increased performance. The elimination of the clock results in reduced die size as well as reduced design complexity.

The delay-insensitive (DI) asynchronous modules/ cores have a number of advantages, especially in SoC, including reduced crosstalk between analog and digital circuits, ease of integrating multi-rate circuits, and ease of component reuse.

In asynchronous designs synchronization is performed through request and acknowledge lines, rather than through global clock. So that, modules can communicates using asynchronous handshake protocols. The number of handshake components determines the system complexity. In future, complex SoC designs will be predominantly asynchronous as predicted by ITRS 2011; the asynchronous part being driven by handshake Signals.

It is extremely important for designers to introduce asynchronous paradigms to make them more marketable and more prepared for the challenges faced by the digital design community in the years to come. Currently, companies such as ARM, Phillips, Intel, and others are incorporating asynchronous logic into some of their products using their own proprietary tools. ITRS Roadmap 2011 also reports that numbers of asynchronous blocks are going to be more compared to that of synchronous blocks in any complicated SoC. The advantages of pipelined architecture are small area, high data throughput and a simple control unit. The drawbacks associated with fully asynchronous design style can be overcome by adopting a globally asynchronous and locally synchronous design methodology.

In any Digital Signal Processing [DSP] application scenario, the design of optimized FFT processor is a scope of implementation. Fast Fourier Transform (FFT) is one of the most important algorithms in DSP [1]. For enhancing signal quality FFT algorithm is one of the basic block in DSP Processor and widely used in various areas like image and speech processing, medical electronics, telecommunication There are several other methods to calculate the Discrete Fourier transforms (DFT), but FFT is the most popular, efficient and fast method for calculating a DFT. Generally FFT is implemented as a synchronous design. It is proposed to design a FFT core as a reusable and ready to integrate block using GALS architecture [2], [3]. In any FFT synchronous design, delay is added between two blocks inside the radix-2 and also between each radix-2 butterfly stage. This increases the latency between the input and output frames. The problem associated with this method is extra hardware for clock distribution which in turn increases the die size and power consumption. This problem can be solved in three ways. The first way is to change the control unit. The second way is to design fully asynchronous structure. The third way is to create Globally Asynchronous and Locally Synchronous (GALS)

Manuscript received July 25, 2014; revised October 29, 2014.

doi: 10.12720/ijsps.3.2.166-171

FFT Architecture. The first option of keeping the architecture completely synchronous increases the design latency. The second option will have more complex control systems and results in a system that is harder to design and understand. The third option will have slightly different control system compared to synchronous structure but it improves the design latency and reduces the power consumption. These advantages of GALS structure inspired to design a GALS based 2D FFT Processor.

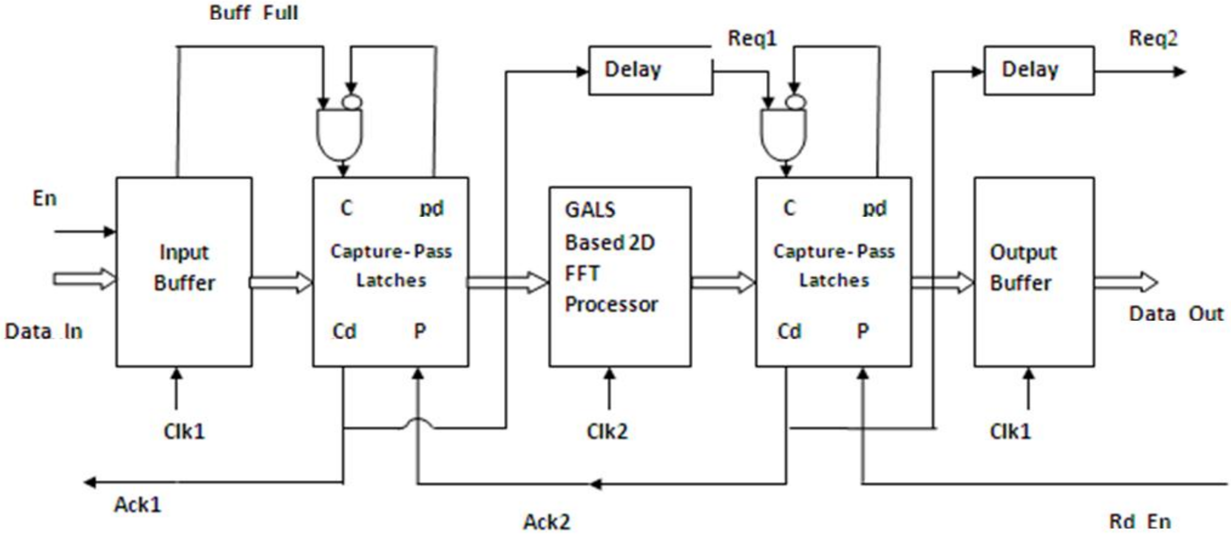


Figure 1. Micro pipelined GALS based 2D FFT processor architecture

II. 2D FFT Processor Architecture Coredesign Specification and Implementation

The aim is to design GALS based 2D FFT processor. The complete internal structure of Micro pipelined GALS based 2D FFT Architecture is as shown in Fig. 1. The data is stored in input buffer which is input for the 2D FFT processor. The data conversion of Time domain representation of image pixel values into Fourier or frequency domain is done by the proposed micro pipelined GALS 2-D FFT Processor.

The input and output buffer blocks works at clk1 frequency and FFT computation block works at clk2 frequency. The communication between two different clock domains is performed using asynchronous handshaking protocol. The asynchronous handshaking protocol implemented in this paper is Sutherland's 2-phase Micro-pipeline. This protocol uses Capture pass element and a Muller c element.

The design implementations of each block are given below:

Input buffer : The data is stored in this FIFO which is later fed to FFT processor for computation. Since it is an 8-point FFT processor, it has sixteen inputs (real and imaginary part of each pixel value). It has been used 24-bit to represent each real and imaginary part of the value. If all the 8-points input values are provided simultaneously, the processor must need 8x24x2=384 pins to load all inputs. So, Serial to Parallel FIFO is used, which takes 24-bit data at every clock cycle and loads the data to the processor in 16 clock cycles. The FIFO contains 24-bit data input lines [Data-In], Clock [Clk1], Write Enable [En], buffer full [Buff_Full] and 2x8x24 output lines [Data_out]. The FIFO stores the new data when_En signal is high. The Buff_Full signal becomes high when all the real and imaginary parts of the inputs are stored. At the same time the stored inputs are fed to FFT processor. The input buffer block diagram is depicted in Fig. 2.

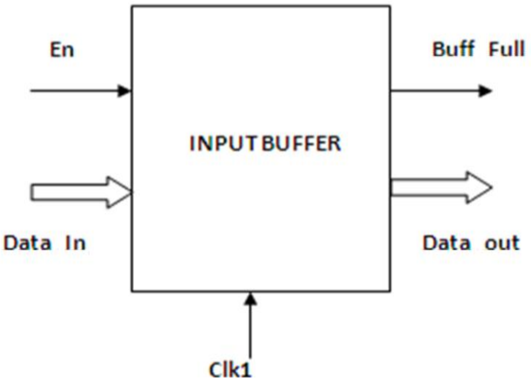


Figure 2. Input buffer to store the image data

GALS based 2D FFT processor: FFT computational block is the heart of FFT processor. This block receives input data from the input buffer and sends the computed data to the output buffer. It has (8x2x24=384 bits) input lines and (8x2x24=384 bits) output lines. It contains 3 radix stages. Each radix stage is consisting of four butterflies which are computed simultaneously. The equations used for radix-2 butterfly computation are:

$$Out1 = A + B\,\omega \tag{1}$$

$$Out2 = A - B\,\omega \tag{2}$$

where, *A* and *B* are the complex numbers, ω is the twiddle factor. The radix-2 butterfly structure of Decimation in Time (DIT) FFT computation is illustrated in Fig. 3.

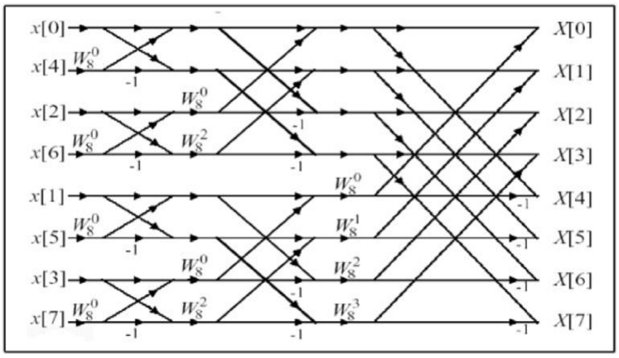


Figure 3. 3 stage radix-2 butterfly structure of FFT computation block

Equation (1) and (2) can be represented into real and imaginary components as shown:

$$A = a + Bj \tag{3}$$

$$B = c + d \tag{4}$$

$$\omega = x + yj \tag{5}$$

$$B\omega = (cx - dy) + (cy\ dx)j \tag{6}$$

$$Out1 = \big(a + (cx - dy)\big) + \big(b + (cy + dx)\big)j \tag{7}$$

$$Out2 = \big(a - (cx - dy)\big) + \big(b - (cy + dx)\big) \tag{8}$$

Equation (7) and (8) can be implemented by signed complex multiplier and adders.

Implementation of 1-D FFT computational block: Architecture of radix-2 butterfly is shown in Fig. 4. It uses a 49 bit complex multiplier which performs equation 6 for which it uses 48 bit booth multiplier to compute the multiplication of 24 bit input and 24 bit twiddle factor. Carry look ahead adder1 performs addition and the carry look ahead adder 2 performs the subtraction.

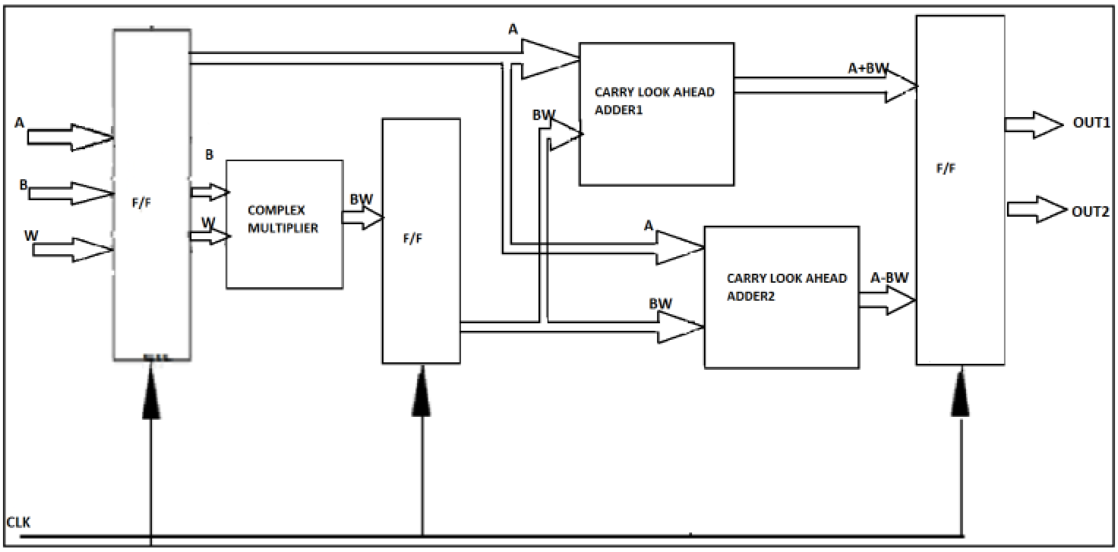


Figure 4. Architecture of radix-2 butterfly

Capture and pass latches (asynchronous handshaking protocol): The asynchronous protocol used in this processor is 2-phase handshaking. In 2-phase handshaking it requires 2 transitions to complete one transition. As seen in Fig. 5 the request signal is from transmitter to receiver, and the acknowledge signal is from receiver to the sender.

In 2-phase protocol initially both request and acknowledge signals are assumed to be in reset condition, a transition on the request signal indicates that the data is ready and the same transition on the acknowledge signal indicates that the data is captured. 2-phase handshaking is also referred as Micro Pipeline. This pipeline is chosen because of its simplicity. To design its control and data latches, it has few complex signals [1]. The control consists of only a two input AND gate and Muller-c elements as shown in Fig. 6.

Initially all the requests and acknowledges are in the reset state. A transition on the req_0 produces a same transition on the ack_1 after one gate delay. The req_0 is delayed and applied to the second stage as req1; the delay depends on the worst case combinational delay of that stage. One of the inputs of the Muller C-elements is inverted, so a logical level transition from zero to one on the input request line will fire the first Muller C-element.

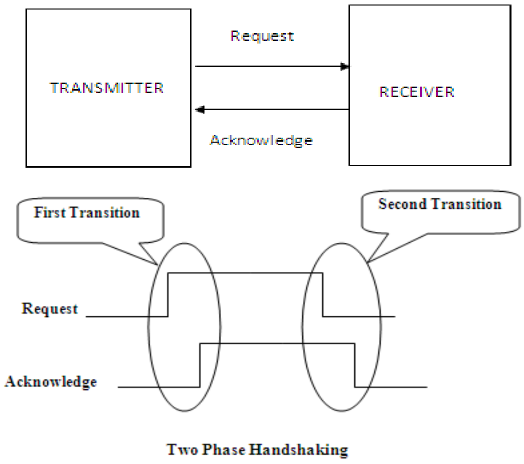


Figure 5. Basic asynchronous communication protocols [4], [5].

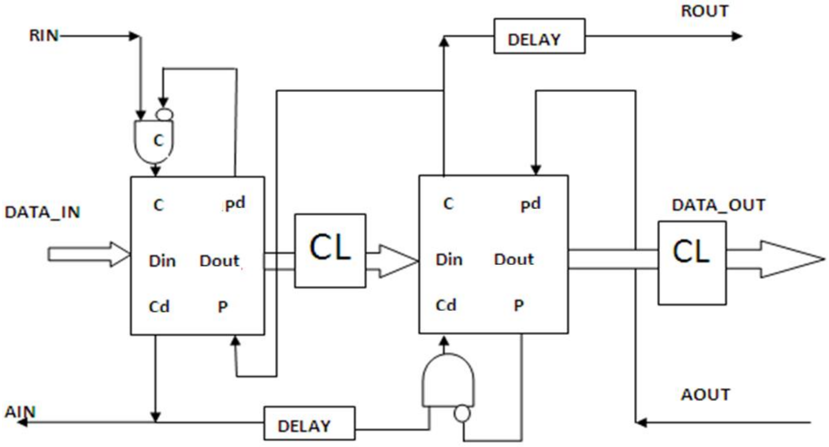


Figure 6. Sutherland's micro pipeline [6]

Then the first latch becomes opaque and captures a new data value. A delayed event will then appear at the Capture done (Cd) pin of the latch which will send acknowledgment to original request and also fire the next C-element. The second latch responds this action by moving into an opaque state which followed by a delayed event on its Capture done (Cd) line. A delayed event on "Cd" of the second latch will cause event on Pass (P) of the first latch, which turns it in to transparent mode and also at the same time it fires a third Muller C-element.

The request control signal continues to propagate down to the Micro pipeline structure with the data until an event occur on output request line at the opposite end which results in the fourth latch being opaque. Since the first output request was never acknowledged, additional requests on the input request line will eventually fill the FIFO.

Output buffer: It is a parallel input and serial data(24 bits) output buffer, which takes all the input bits (384 bits) at same time and generates one 24 bits output at every clock cycle. This block contains (2x8=16, 24 bits) input lines [Data-In], Clock [Clk1], Read Enable [Rd_En], and 24 output lines [Data_out].

The output from the FFT computational block is stored in the output buffer. When Rd_En is high, output buffer sends out each 24 bit data stored at every clock cycle. For the 8-point FFT, output buffer takes 16-clock cycles to send out all the data stored. Fig. 7 depicts the block diagram of the output buffer block.

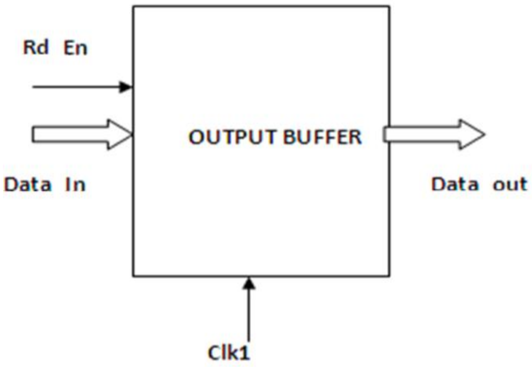


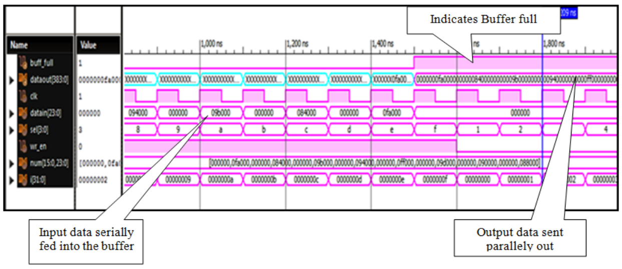
Figure 7. Output buffer to store the computed data

III. Results and Dicussion

Both the Asynchronous and Synchronous 2D FFT Processor are coded in Verilog [7] and simulated to check the functionality using NCSIM simulator fro Cadence. The synthesis reports for Power and area are taken from Cadence Encounter Tool and RTL Complier to do performance.

A. Simulation Results of Input Buffer

The Fig. 8 shows the simulation of Input Buffer. It is seen that the inputs are sequentially coming into it and after all the inputs are stored, the buff_full signal goes high and all the inputs are sent on the 384 output lines at once.



Indicates Buffer full

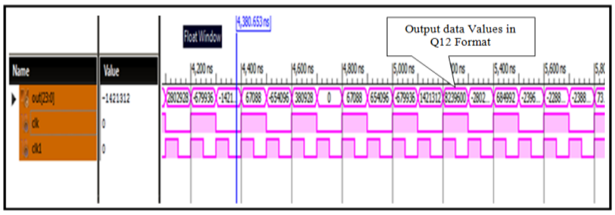
Input data serially fed into the buffer

Output data sent parallely out

Figure 8. Simulation results of input buffer

B. Simulation Result of 2-D FFT Processor

Fig. 9 shows the Simulation results of FFT Computation. The output pixel values are in Q12 format. The values obtained are also verified with theoretical values.



Output data Values in Q12 Format

Figure 9. Simulation results of 2D FFT processor

C. Simulation of Handshake Block

Fig. 10 shows the simulation of the handshake block. It is seen that only when the rin and pass signals are high the input values are passed to the output.

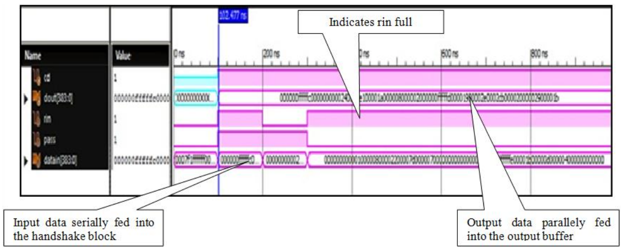

Indicates rin full
Name
Value
cd
dout[383:0]
rin
pass
datain[383:0]
102,477 ns
0 ns
200 ns
600 ns
800 ns
Input data serially fed into the handshake block
Output data parallely fed into the output buffer

Figure 10. Simulation results of handshake block

D. Simulation of Output Buffer 2-D

Fig. 11 shows the simulation of the output buffer. Here the input on 3072 input lines is stored in one cycle and the read_en signal becomes high. The next cycle onwards the stored values are sent out sequentially.

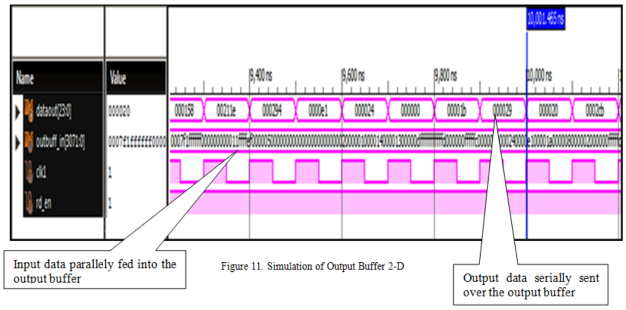

10,001.465 ns
Name
Value
dataout[23:0]
outbuff_in[3071:0]
clk1
rd_en
9,400 ns
9,600 ns
9,800 ns
10,000 ns
Input data parallely fed into the output buffer
Figure 11. Simulation of Output Buffer 2-D
Output data serially sent over the output buffer

Figure 11. Simulation of output buffer 2-D

IV. Design Summary

According to the synthesis reports obtained from RTL Complier from Cadence EDA Tool for 2D Micro pipelined GALS based asynchronous and Synchronous architectures of FFT Processor the timing, power and area are tabulated and compared.

A. Timing

Timing of the Micro pipelined GALS architecture has been reduced by 39.99% when compared to the Synchronous architecture. Table I shows the comparison.

B. Power

Power of the Micro pipelined GALS architecture has been reduced by 78.22% when compared to the Synchronous architecture which is depicted in Table II.

C. Area

Area of the Micro pipelined GALS architecture has been increased by 31.87% when compared to the Synchronous architecture. This trade is affordable if speed and power optimization is the main criteria.

TABLE I. Timing Comparison

Architecture	**Arrival Time in Pico Seconds**
Synchronous FFT Processor	18480
Micro pipelined GALS FFT Processor	11089
Percentage Reduction	**39.99%**

TABLE II. Power Comparison

Architecture	Leakage Power in Nano watts	Dynamic Power in Nano watts	Total Power in Nano watts
Synchronous FFT Processor	331071.361	8700920.839	9031992.208
Micropipelined GALS FFT Processor	392086.126	1574726.477	1966812.603
Percentage Reduction	**78.22%**		

V. Conclusion

As the need for faster and low power digital signal processing product growing, the need of the faster FFT processor increases. A lot of research has been done to design low latency FFT processor, while keeping area reasonable. The synchronous FFT processor has many difficult challenges to overcome in terms of latency, clock skew and power. These challenges have made an asynchronous FFT processor design an increasingly practical alternative. But fully asynchronous design is difficult to design, test and also require more area for designing control circuit. GALS FFT Processor is faster than existing FFT processor. In this paper it has been shown that competitive results are achieved. The delay of proposed 2-phase handshaking protocol depends upon operating speed of the combinational logic.

It is seen that the timing of our proposed design is reduced by 39.99% when compared to Synchronous design. Similarly power also reduces by 78.22%. But the area of the GALS architecture is increased by 31.87%.

Since the inputs are taken in a sequential manner and outputs are given out in a sequential manner the number of input and output pins required is reduced.

Acknowledgement

The authors wish to acknowledge BNMIT Management for supporting this work.

This work is carried out as a part of research work on DFT Techniques for Asynchronous Dominant Soc.

References

[1] T. D. Patel, "Micropipeline a data path of FFT," Ph.D dissertation, San Diego State University, USA, 2010.

[2] D. M. Chapiro, "Globally asynchronous locally synchronous systems," PhD Thesis, Stanford University, USA, Oct. 1984.

[3] European Union's Seventh Framework Program, "GALS interface for complex digital system integration," EU Project GALAXY, 2009.

[4] Overview of asynchronous logic. [Online]. Available: http://comp.uark.edu/~smithsco/async_intro.pdf

[5] S. B. Fuber and P. Day. (Jun. 1996). Four phase micropipeline latch control circuits. *IEEE Trans. on VLSI Syst.* [Online]. 4(2). pp. 247-253. Available: http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=502196&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxpls%2Fabs_all.jsp%3Farnumber%3D502196

[6] P. Day and J. V. Wood. (Jun. 1995). Investigation into micropipeline latch design style. *IEEE Trans. on Very Large Scale (VLSI) Syst.* [Online]. 3(2). pp. 264-272. Available: http://www-classes.usc.edu/engr/ee-s/552/coursematerials/ee552-S1.pdf

[7] N. M Botros, *HDL Programming VHDL and Verilog*, 4th ed., Biztantra, Dreamtech Press, 2009.

Mrs. Yasha Jyothi M Shirur is Associate Professor in the department of Electronics & Communication Engineering, BNM Institute of Technologies, BSK II stage, Bangalore. She holds Post Graduate degree in Electronics from BMSCE, Bangalore. Currently she is pursuing Ph. D in VLSI testing in asynchronous SOC.

She is having more than 15 years of experience in teaching. She has served many Institutions in different capacity.

Mrs. Yasha Jyothi M Shirur is IEEE Member and Life Member ISTE. She has published around thirteen papers in Journals, International and National Conferences.

Dr. Veena S Chakravarthi has over two decades of semiconductor industry experience in managing design and development of large multiprocessor SoCs. She has served for ITI Ltd., Mindtree Consulting Pvt. Ltd, Centillium India Ltd., and Transwitch India Ltd., in various capacities and is now Professor in the department of Electronics and Communications, BNM Institute of technology, BSK II stage, Bangalore. She has to her credit around 6 multiprocessor multimillion gate tapeouts and over 20 silicon proven IP cores designs. She has received the PhD degree from Bangalore University for her research work in "Generalized power optimization methodology for ASICs" and has published around twenty papers in National and International conferences. She has been recognized as one of the six "Smart Techies" from Silicon India group for her contribution to semiconductor industry in the year 2008. Her current interests being Low power SOCs, MPSOCs, and Design methodologies. She has served as a PON consultant to Ikanos communications for their PON product developments and was consultant to Periera Ventures, US. She is the IEEE Senior Member. At present she is the architect Consultant to Asarva Chips and Technologies for their WLAN SOC development.