A HYPER-PIPELINING APPROACH FOR HIGH-SPEED SINGLE-MICRO-PIPELINE DESIGN FOR HYBRID MULTIPLIER

Uday Arun, A. K. Sinha, K.S. Yadav
Bharath Institute of Science & Technology, Chennai, Tamil Nadu, India
ABES Engg. College, Ghaziabad, Uttar Pradesh, India
Northern India Engg. College, New Delhi, India

Abstract

Power-awareness indicates the scalability of the system energy with changing conditions and quality requirements. Multipliers are essential elements used in DSP applications and computer architectures. Although Boolean multipliers have natural power awareness to the changing of input precision, deeply pipelined designs do not have this benefit. A Single-Edge-Triggering (SET) & Double-Edge-Triggering (DET) one-dimensional pipeline gating scheme is proposed to improve the speed, the power awareness and area in designs. In one-dimensional pipeline, clock the registers (D-FF) at both edges i.e. positive as well negative (data flow in pipeline) and horizontal direction (within each pipeline stage). This technique only needs very little additional area and the overhead is hardly noticeable. A set of Hybrid Multiplier were designed and tested. Results show that the new 32-bit Hybrid Multiplier using this technique has low power and high speed at lower clock frequency as compare to other designs.With increasing clock frequencies and silicon integration, power aware computing has become a critical concern in the design of embedded processors and systems-on-chip. One of the more effective and widely used methods for power aware computing is Dynamic Voltage Scaling (DVS). In order to obtain the maximum power savings from DVS, it is essential to scale the supply voltage as low as possible while ensuring correct operation of the processor. The critical voltage is chosen such that under a worst case scenario of process and environmental variations, the processor always operates correctly. However, this approach leads to a very conservative supply voltage since such a worst case combination of different variability will be very rare. A SET & DET D-flip-flop pipeline multiplier pipeline Hybrid Multiplier design will work at the clock frequency in the range of 50MHz to 250MHz and SET based pipeline Hybrid Multiplier design will work at the clock frequency in the range of 50MHz to 500MHz. We have used hyper-pipelining approach in pipeline design by changing register-retiming approach i.e. combined memory at DET to further improve the speed as well as area & power consumption. The various design & analysis presented in result section.

1. Introduction

The hybrid multiplication mode is a combination of the semi-parallel and sum of multiplication modes where bit sections from two unique input streams are multiplied with two different coefficient values. This mode is useful in applications that require complex multiplication like FFTs where each signal generally has a real and imaginary component that could be multiplied by two unique coefficient values. The partial products obtained from each bit section within the components are shift accumulated to obtain the final result.

Real-Time signal processing hardware requires efficient multiplier units. However, each application needs different sample rates; from speech to image or radar a wide frequency range is covered. Digital Signal processing (DSP) is used in a wide range of applications such as telephone, radio, video, radar, sonar, satellite communication, etc. The sample rate requirements vary from application to application and can range anywhere from 10 KHZ to 400MHZ. Real-Time implementation of these systems require hardware architectures which can process input signal samples as they are received, as opposed to storing them in registers & processing them in batch mode.

In several cases; it has an important cost in area and runs faster than the throughput required by the application. Most of the DSP computations involve the use of multiply accumulate operations & therefore the design of fast and efficient multipliers is imperative. Moreover, the demand for portable applications of DSP architectures has dictated the need for low power design.

Multipliers play an important role in today's digital signal processing and various other applications. With advances in technology, many researchers have tried and are trying to design multipliers which offer either of following – high speed, low power consumption, regularity of layout and hence less area or even combination of them in one multiplier. Thus, making them suitable for various high speed, low-power, and compact VLSI implementation.

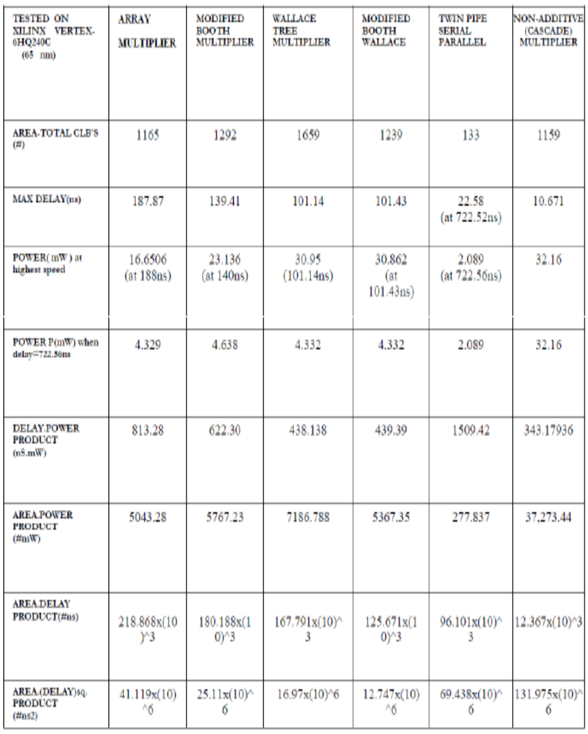
The common multiplication method is add and shift algorithm. In parallel multipliers number of partial products to be added is the main parameter that determines the performance of the multiplier. To reduce the number of partial products to be added, Modified Booth algorithm is one of the most popular algorithms. To achieve speed improvements Wallace- Tree algorithm can be used to reduce the number of sequential adding stages. Further by combining both Modified-Booth algorithm and Wallace-Tree technique we can see advantage of both algorithms in one multiplier.

However with increasing parallelism, the amount of shifts between the partial products and intermediate sums to be added will increase which may result in reduced speed, increase in silicon area due to irregularity of structure and also increased power consumption due to increase in interconnect resulting from complex routing.

On the other hand serial parallel multipliers compromise speed to achieve better performance for area and power consumption. So selection of a parallel or serial multiplier actually depends on the nature of application.

In this paper, we compare four parallel, one serial-parallel and one cascade multiplier for area, speed, power consumption and different product of them as various performance measures and summarize the results in a tabular form.

As main idea here is that of comparison of different multipliers, we need a common base for comparison of all multipliers. For this reason synthesis was done using same design constraints and keeping the target device same for all multipliers. As a result of this each multiplier in its own may not have optimal performance measures and can be optimized individually.



TESTED ON XILINX VERTEX-6HQ240C (65 nm)	ARRAY MULTIPLIER	MODIFIED BOOTH MULTIPLIER	WALLACE TREE MULTIPLIER	MODIFIED BOOTH WALLACE	TWIN PIPE SERIAL PARALLEL	NON-ADDITIVE (CASCADE) MULTIPLIER
AREA-TOTAL CLB'S (#)	1165	1292	1659	1239	133	1159
MAX DELAY(ns)	187.87	139.41	101.14	101.43	22.58 (at 722.52ns)	10.671
POWER(mW) at highest speed	16.6506 (at 188ns)	23.136 (at 140ns)	30.95 (101.14ns)	30.862 (at 101.43ns)	2.089 (at 722.56ns)	32.16
POWER P(mW) when delay=722.56ns	4.329	4.638	4.332	4.332	2.089	32.16
DELAY.POWER PRODUCT (nS.mW)	813.28	622.30	438.138	439.39	1509.42	343.17936
AREA.POWER PRODUCT (#mW)	5043.28	5767.23	7186.788	5367.35	277.837	37,273.44
AREA.DELAY PRODUCT(#ns)	218.868x(10)^3	180.188x(10)^3	167.791x(10)^3	125.671x(10)^3	96.101x(10)^3	12.367x(10)^3
AREA.(DELAY)sq. PRODUCT (#ns2)	41.119x(10)^6	25.11x(10)^6	16.97x(10)^6	12.747x(10)^6	69.438x(10)^6	131.975x(10)^6

Figure 1. IEEE Journal Research Paper (System & Circuit Design

From fig.1 we can see that delay of Non-Additive (Cascade) multiplier is the least. Hence it is fastest among six multipliers. DP product is also least for above multiplier and is a good choice for this performance measure. Serial Parallel multiplier is a best choice when speed is not important but reduced area and power consumption is of more interest and also for AP and AD product Serial Parallel multiplier is a good choice. However, one of the most important in applications where area and power consumption is strictly restricted and speed is not much important it is not wise to use parallel multipliers as they occupy more silicon area and consume more power compared to serial parallel multipliers.

We can conclude that each multiplier has its own advantage and disadvantage for different performance measures and choice of a specific multiplier completely depends upon the nature of application and constraints on area, delay, power or combination of them for that particular application. One can select a multiplier from seeing this table depending on which performance criteria is most critical for a particular application. Multipliers compared may be optimized individually by either changing VHDL coding style, adding certain constraints specific to an architecture during synthesis, changing target device part number and its speed grade etc. Individual optimization is not considered as main purpose. So, main issue in serial/parallel multipliers is to reduce the internal loop delay which is the bottleneck of throughput.

Shailesh Shah [1] have made a comparison of different 32-bit integer multipliers based on different performance measures by using same design constraints and keeping the target device same for all multipliers. It is observed that area of twin pipe hybrid multiplier is the least but has more delay as compared to non-additive cascade multiplier. In this twin pipe hybrid multiplier, the delay is mainly caused by the internal loops in each multiplication stage.

So, main issue in hybrid multipliers is to reduce the internal loop delay which is the bottleneck of throughput.

Soojin Kim [2] proposed the pipeline technique for the 8-bit and16-bit modified booth multipliers to improve performance but operating at very high frequency (in GHz) which results in more power consumption. He discussed that the overall delay of the multiplier circuits decrease as the number of pipeline stages increase but that results in an increase in area. He compared his proposed multiplier with others [3,4,5,6,7,8] and concluded that proposed one shows the best result. However, single pipeline using the concept of hyper-pipelining and DET clock pulse has not been designed in 32-bit hybrid multipliers by any researcher which is presented in this work.

Twin pipe hybrid multiplier is not implemented in any real application as it operates on 120 nm technology device and 400 MHz clock frequency.

ARM7TDMI 32-bit Controller [14] uses the barrel shifter for designing the Arithmetic & Logic Unit in the core. IBM & Motorola launched PowerPC 603e & 603ev Processor [13] using the ARM core. This processor works on 90 nm technology & operates at 333 MHz and uses barrel shifter to obtain low-power design.

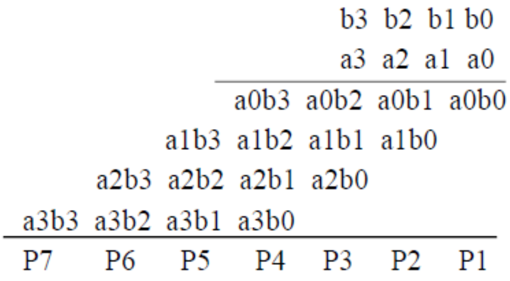
The ARM11^{TM} processor family provides the engine that powers many smartphones in production today.It delivers extreme low power and a range of performance from 350 MHz in small area designs up to 1 GHz in speed optimized designs.

ARM11 32-bit processor core [14] is implemented in Nokia 6120 Classic with a clock frequency of 350 MHz and designed on 45-65 nm technology. But in this present work, single pipeline 32-bit hybrid multiplier with DET memory is designed on 65 nm technology device at 200 MHz clock frequency to improve the overall performance.

2. Related work and Design Analysis

Assuming B is multiplicand and A is multiplier then product B x A is shown as:

A= {a0, a1, a2, a3} B= {b0, b1, b2, b3} & Product P = {p1, p2, p3…..p7}



			b3	b2	b1	b0
			a3	a2	a1	a0
			a0b3	a0b2	a0b1	a0b0
		a1b3	a1b2	a1b1	a1b0	
	a2b3	a2b2	a2b1	a2b0		
a3b3	a3b2	a3b1	a3b0			
P7	P6	P5	P4	P3	P2	P1

The Equation is:

$$P\,(m+n) = A\,(m)\,B\,(n) = \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} 2^{i+j}\, a_i b_j$$

The operating method of a multiplier is ANDing, Adding & Shifting. ANDing operation is performed in parallel using AND gate to obtain partial products. Adding & Shifting operations are sequential operation which is time-consuming. Adding operation is performed using full adder circuit to obtain intermediate sum of partial products. Shifting operation requires group of memory cells (storage element). Shifting and Instruction overlapping is equivalent operation in Instruction Pipeline of any processor.

Using this logical and analytical approach, multiplier performance can be improved by designing a pipeline for shifting operation. If Adding operation is performed along with shifting operation in single pipeline then pipeline stage reduction can be achieved and thus multiplier performance can be further optimized.

A commonly used method to improve performance is to increase the clock frequency. However, use of high clock frequency has a number of disadvantages such as more power consumption. An alternative strategy relies on the use of storage elements capable of capturing data on both clock edges (rising and falling edge). [9]

2.1. Anatomy of Hybrid Multiplier

Hybrid multiplier is a composistion of Serial/parallel multiplier. An architecture of hybrid multiplier has three components i.e. Paprallel to Serial Converter, Hyper-pipeline multiplier & Serial to Parallel Converter. PTSC & STPC are working on master clock frequency.

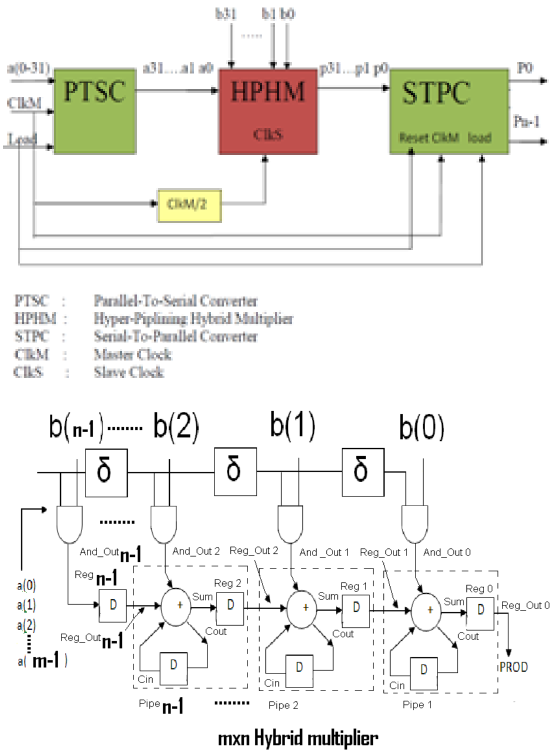


Figure 2. Architecture of Hybrid Multiplier

2.2. Anatomy of Pipeline Multiplier

Figure 3. Generic Architecture of pipelined Multiplier

2.2.1. Anatomy of Pipeline in Processor Micro Architecture

Pipeline is the combination of pipe. The pipe is component which consists of flip-flop, adder and various signals for controlling. The main two types of pipe shown below.

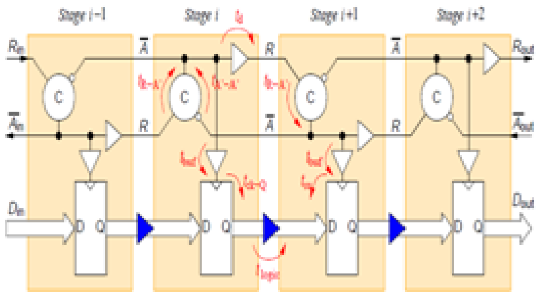


Figure 4. Architecture of Pipeline Model

Pipelining is also called as Pipeline Processing. This chapter deals with the concept of advanced pipelining and superscalar design in processor development. It begins with a discussion of conventional linear pipelines and analyzes their performance. A generalized pipeline model is introduced to include non-linear interstage connections. Further, Super- pipelining and Superscalar design techniques are analyzed based on their performance.

It is a cascade of processing stages which are linearly connected to perform a fixed function over a stream of data flowing from one end to other. A linear pipeline processor is constructed with k processing stages. External inputs (operands) are fed into the pipeline at the first stage S1. The processed results are passed from stage S_i to $S_{i+1,}$ for all i = 1, 2… k-1. The final result emerges from the pipeline at the last stage S_i.

Linear pipelines are applied for instruction execution, arithmetic computation and memory access operations. Depending on the control of data flow along the pipeline, linear pipelines are modeled into two categories:

- Asynchronous model
- Synchronous model

2.2.2. Synchronous Linear Pipeline Model

Clocked latches are used to interface between stages. The latches are made with master slave flip flops, which can isolate inputs from outputs. Upon the arrival of a clock pulse, all latches transfer data to the next stage simultaneously. Pipeline stages are combinational logic circuits. It is desired to have approximately equal delays in all stages. These delays determine the clock period and thus the speed of pipeline.

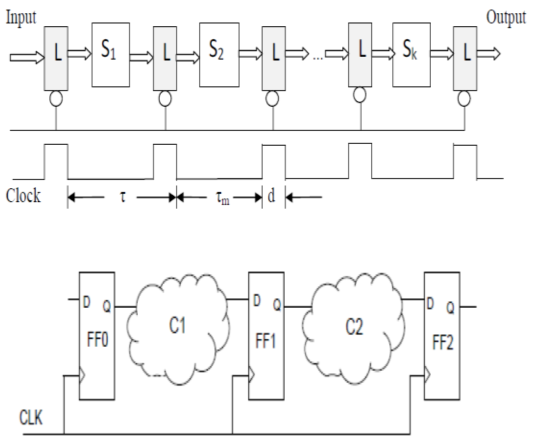


Figure 5. Architecture of Linear Pipeline Model

2.2.3. Clocking and Timing Control

Let τ_i be the time delay of the circuitry in stage S_i and d the time delay of a latch, then clock cycle (period) τ of a pipeline is:

$$\tau = \max\{\tau_i\}_{1k} + d = \tau_m + d$$

where, τ_m denotes the maximum stage delay. At the rising edge of the clock pulse, the data is latched to the master flip flops of each latch register.

The clock pulse has a width equal to d. The pipeline frequency (f) is defined as the inverse of the clock period:

$$f = 1/\tau$$

If one result is expected to come out of the pipeline per cycle, f represents the maximum throughput of the pipeline. The actual throughput may be lower than f depending on the initiation rate of successive tasks entering the pipeline.

When **clock skew** takes effect i.e. same clock pulse may arrive at different stages with a time offset of s. Let t_{max} be the time delay of the longest logic path within a stage and t_{min} that of the shortest logic path within a stage. To avoid a race in two successive stages, we must choose:

$$\tau_m \geq t_{max} + s \text{ and } d \leq t_{min} - s$$

These constraints translate into the following bounds on the clock period:

$$d + t_{max} + s \leq \tau \leq \tau_m + t_{min} - s$$

In the ideal case, s = 0, $t_{max} = \tau_m$ and $t_{min} = d$. Thus, we have $\tau = \tau_m + d$, consistent with the definition in above equation without the effect of clock skewing.

2.2.4.Anatomy of Hyperpipelining or Register Retiming

The concept of Hyper-pipelining or register retiming [11,12] is used to achieve performance requirements that hinges on sequential elements being optimally placed so as to minimize and balance path delays. It works by literally “moving” registers across portions of combinational logic such that the worst-case combinatorial delays on each the input and output sides of the register are more balanced. It is an optimization strategy that leverages positive slack on one side of a sequential element to address or balance negative slack on the other. It can lead to either an increase or decrease in the number of flip-flops in the design.But number of pipeline stages does not change. It is constrained to operate only in such a way that preserves design functionality at the top-level design ports. It results in less area usage and has a beneficial impact on the system performance. In this work, Hyper Pipelining is used in such a way that registers are used on top of sequential logic in the design of single pipeline and then registers are merged to form one combine memory within each stage of pipeline. It takes a timing-driven approach in which static timing analysis is performed to obtain time delay & total area of Hybrid Multiplier.

2.2.5. Anatomy of Power & Speed Analysis

With advances in technology, many researchers have tried & are trying to design multipliers which offer either of following high speed, low power consumption, regularity of layout or less area and even combination of them in one multiplier.

In applications where area and power consumption is strictly restricted and speed is not much important, it is not wise to use Parallel Multiplier as compared to Hybrid Multiplier. P.R. Panda [10] discussed the designing of circuits, which are conceived as Finite State Machines (FSM). Depending on the complexity of the circuit, a large fraction of the power is consumed due to the switching of the state register. The objective of low power approaches is therefore to choose a state encoding that minimizes the switching power of the state register.

Given a state encoding, the power consumption can be modeled as:

$$P = \tfrac{1}{2}V^2F \times C_{sr} \times E_{sr}$$

Where, F is the clock frequency,
V is the supply voltage,
C_{sr} is the effective capacitance of the state register, and
E_{sr} is the expected state register switching activity.

If S is the set of all states, we can estimate E_{sr} as:

$$E_{sr} = \sum_{i,j \in S} p_{ij} \times h_{ij}$$

Where, p_{ij} : probability of a transition between states i and j, and
h_{ij} : Hamming Distance between the codes of states i and j.

To achieve low power and high performance design, the value of E_{sr} need to be minimized.

This can be done by using one-hot method [15] of designing circuits, in which only one register is used in each state of pipeline.

P.R. Panda [10] discussed that the power consumption of any circuit can be divided into three different components: dynamic, static (or leakage) and short circuit power consumption. Thus,

$$P_{total} = Pdynamic + Pshort_{-circuit} + Pstatic$$

Switching power, which includes both dynamic and short circuit power, is consumed when signals change their logic state, resulting in the charging & discharging of load capacitors. The dynamic power of a circuit can be estimated as:

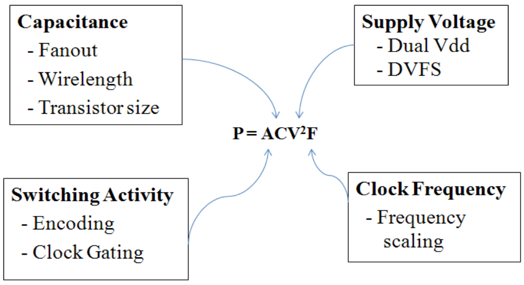
$$P = ACV^2F$$

Where, C is the switched capacitance,
V is the supply voltage,

F is the clock frequency, and
let A be a constant called the activity factor ($0 \leq A \leq 1$).

The clock frequency (F) is related to Time delay as:

$$F = 1/Time$$



Thus, with the increase in clock frequency, delay is minimized which in turn increases the speed but also consumes more power.
Therefore, a good hybrid multiplier design requires some trade-off between speed and power consumption by optimizing clock frequency. This can be done by using Hyper Pipelining approach in single pipeline hybrid multiplier by using DET storage elements. But most libraries do not provide and support DET memory, so a new library is coded in VHDL which provides dual edge behavior.

2.2.6. Genisis of Doule Edges Triggering Memory Element (D-flipflop)

Double edge-triggered flipflops are becoming a popular technique for low-power designs since they effectively enable a halving of the clock frequency [17][18]. The paper by Hossain et al [21] showed that while a single-edge triggered flipflop can be implemented by two transparent latches in series, a double edgetriggered flipflop can be implemented by two transparent latches in parallel. The clock related power is one of the most significant components of the dynamic power consumption [20][22].
The total clock related power dissipation in synchr--onous VLSI circuits is further divided into three major components: (i) power dissipation in the clock network, (ii) power dissipation in the clock buffers, and (iii) power dissipation in the flip flop. The total power dissipation of the clock network depends on both the clock frequency and the data rate, and can be computed based on Eq.:

$$P_{CLK} = V^2_{dd} \, [f_{CLK} \, (C_{CLK} \, C_{f\,f,\,CLK} + f_D \, C_{f\,f,\,D}]$$

Dual Edge Triggered Flip-Flop

Where

f_{CLK} is the clock frequency;
f_D is the average data rate;
C_{CLK} is the total capacitance seen by the clock network;
$C_{f\,f,\,CLK}$ is the capacitance of the clock path seen by the filp-flip;
$C_{f\,f,\,D}$ is the capacitance of the data path seen by the filp-flip;

From above Equation, it is obvious that the clock power can be reduced if any of the parameters on the right hand side of the equation is reduced. The reduction of Vdd is already the trend of contemporary design, and it has the strongest impact on the P_{CLK} expression.
By reducing the overall capacitance of the clock network, C_{CLK}, the power dissipation may also be reduced. For instance, the capacitance can be reduced by proper design of clock drivers and buffers. Similarly, by reducing the capacitance inside a flip flop, $C_{ff,\,CLK}$ and $C_{ff,\,D}$, power may also be reduced. Furthermore, the clock power dissipation is linearly proportional to the clock frequency. Although the clock frequency is determined by the system specifications, it can be reduced with the use of dual edge triggered flipflop (DETFFs). As its name implied, DETFF responds to both rising and falling clock edges. Hence, it can reduce the clock frequency by half while keeping the same data throughput.
As a result, power consumption of the clock distribution network is reduced, making DETFFs desirable for low power applications.

Even for high performance applications, the usage of DETFFs offers certain benefits. Since the clock speed is reduced by a factor of two, one does not need to propagate a relatively high speed clock signal [23].
In this classic configuration, two opposite polarity level-sensitive latches are connected in parallel, the output is then multiplexed at the output stage.

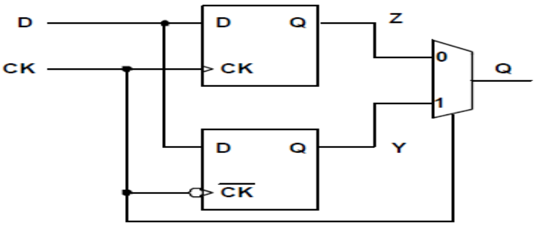


Figure 6. Classical Implementation of DETFF

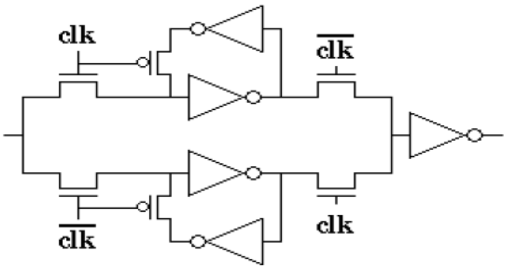


Figure 7. Gate Level CMOS Implementation

2.2.7. Single Edge Triggering (SET) Pipleine Component with Separate Memory

A multiplier pipe is a composition of two main components or elements i.e. Full Adder & Memory Elelments or Storage Elements.One pipe requires two memory elements one for Sum storage and other for Carry out storage. Each memory element consists of D-Flip-Flop with two control signal. In this meory elements design, we have clock signal & asynchronous reset signal as a control signal for both meory elements separately. A Full Adder has two inputs **a** and **b** where **b** is partial product of $\mathbf{a_n}$ and $\mathbf{b_n}$. The behaviour of a pipe is similar to serial adder or sequential adder circuit. The speed of pipe is depended on the speed of adder circuit (combinational circuit) and inter loop delay path of Cout to Cin. The speed of pipe can be improved further by reducing delta delay or transportation delay of path Cout to Cin and propogation delay of D-Flip-Flop.

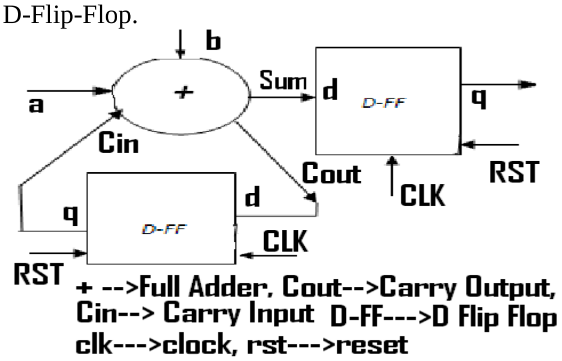

b
a
+
Sum
d
D-FF
q
Cin
Cout
CLK
RST
q
D-FF
d
CLK
RST
+ -->Full Adder, Cout-->Carry Output,
Cin--> Carry Input D-FF--->D Flip Flop
clk--->clock, rst--->reset

Figure 8. Design Adder and Shifter with SET Separate Memory

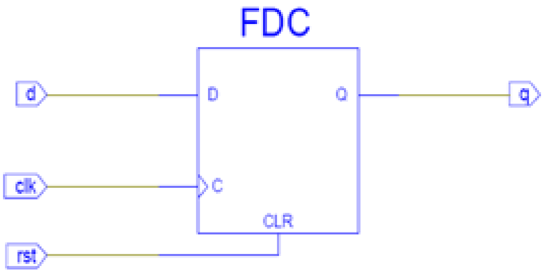


Figure 9. RTL Schematic of SET Separate Memory Element

In this design, the area will increase due to separate memory element as well as path of clock signal also increased.The chances of clock skew more in separate memory element based pipe in multiplier.

2.2.7. Single Edge Triggering Pipeline Component with Combined Memory

To reduce the area of pipe and clock delay of memory element, a combined memory approach is used. This is known as register retiming or hyper pipelining mechanism. This approach is also minimize the clock signal and reduced the chances of clock skew in combined memory element based pipe.

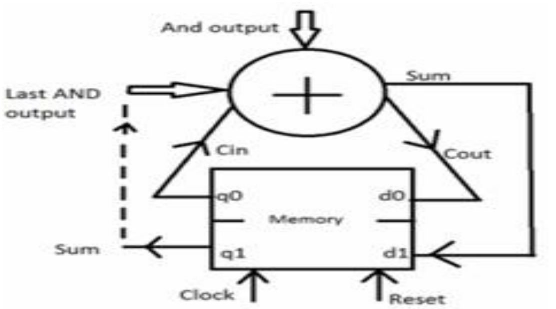


Figure 10. Design of Adder and Shifter with SET Combined Memory

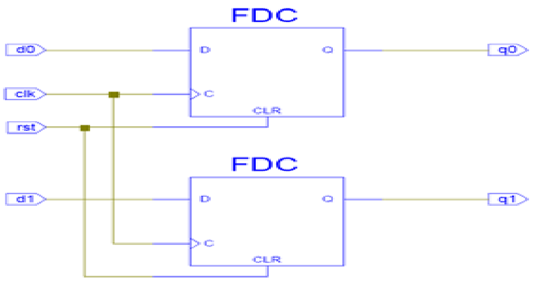


Figure 11. RTL Schematic of SET Combined Memory Element

2.2.8. Double Edge Triggering Pipeline Component with Separate Memory

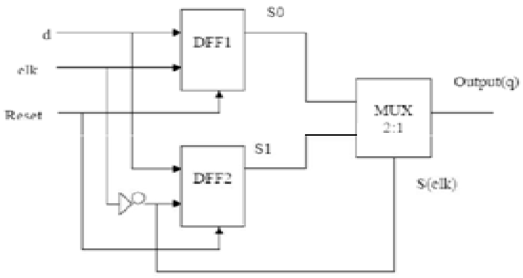


Figure 12. Design of Double Edge Triggering (DET) Memory

2.2.8.1. RTL Modeling of Double Edge Triggering Memory Elememnt of Separate Memory

```
 DFF0: process(d,clk,rst)
  begin
   if rst ='1' then
       s(0) <= '0';
   elsif clk'event and clk = '1' then
           s(0)<= d;
   end if;
   end process;
DFF1:  process(d, NegClk, rst)
   begin
   if rst ='1' then
          s(1) <= '0';
   elsif NegClk'event and NegClk = '1' then
           s(1)<= d;
  end if;
  end process;
 Mux2x1: process(NegClk,S(0),S(1))
  begin
   q <= 'Z';
   if NegClk ='0' then
        q <= s(0);
   else
        q <= s(1);
   end if;
  end process;
```

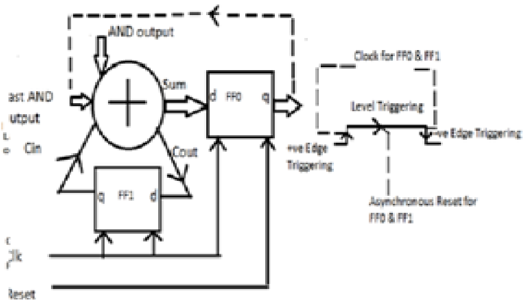


Figure 13. Design Adder and Shifter with Double Edge Triggering (DET) Memory

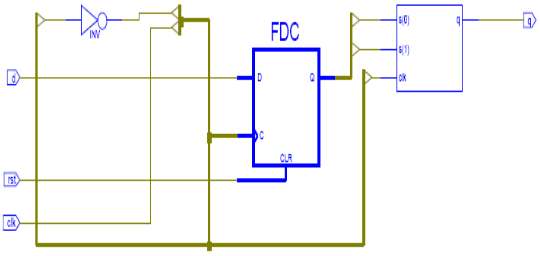


Figure 14. RTL Schematic of Double Edge Triggering Memory Element

2.2.8.2. RTL Modeling of Double Edge Triggering Memory Elememnt of Combined Memory

```
DFF0: process(d0, d1, clk, rst)
  begin
   if rst ='1' then
          s(0) <= '0';
           s(2) <= '0';
   elsif clk'event and clk = '1' then
          s(0)<= d0;
          s(2)<= d1;
   end if;
   end process;
 DFF1:  process(d0, d1, NegClk, rst)
   begin
   if rst ='1' then
          s(1) <= '0';
          s(3) <= '0';
  elsif NegClk'event and NegClk = '1' then
          s(1)<= d0;
          s(3)<= d1;
  end if;
  end process;
 Mux2x1: process(NegClk, S(0), S(1), s(2), s(3))
  begin
          q0 <= 'Z';
          q1 <= 'Z';
   if NegClk ='0' then
          q0 <= s(0);
           q1 <= s(2);
   else
          q0 <= s(1);
           q1 <= s(3);
   end if;
  end process;
```

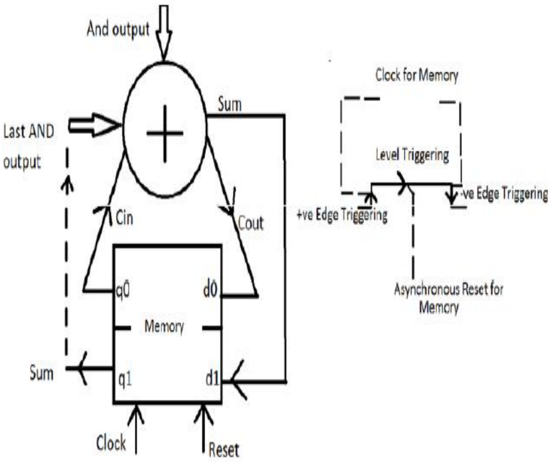


Figure 15. Design Adder and Shifter with Double Edge Triggering Combined Memory

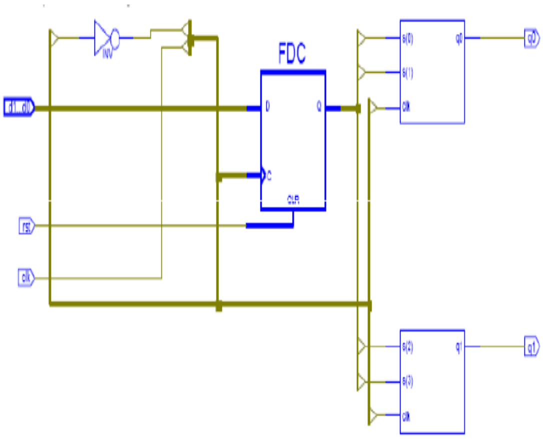


Figure 16. RTL Schematic of DET Combined Memory Element

3. Observation

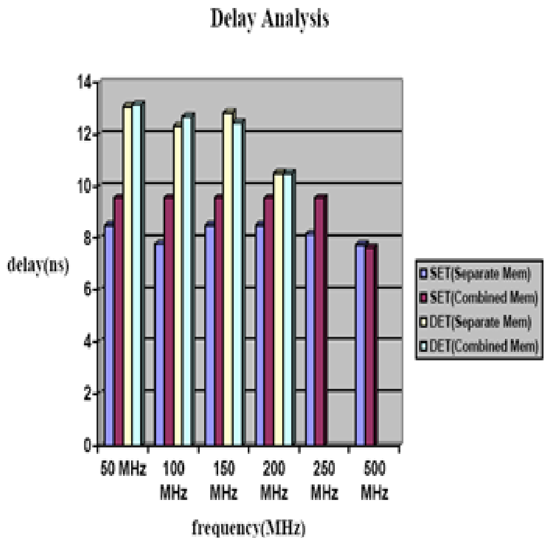
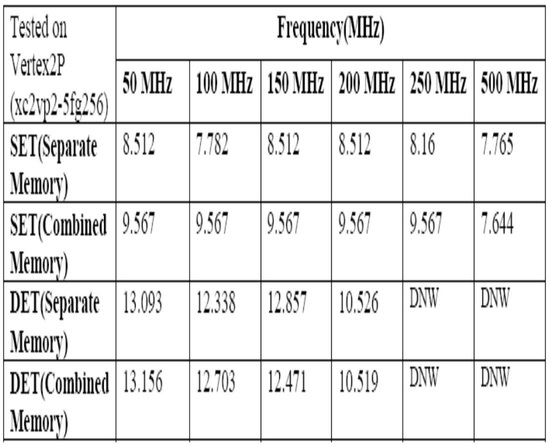


Figure 17. Delay Analysis Chart



Tested on Vertex2P (xc2vp2-5fg256)	Frequency(MHz)					
	50 MHz	100 MHz	150 MHz	200 MHz	250 MHz	500 MHz
SET(Separate Memory)	8.512	7.782	8.512	8.512	8.16	7.765
SET(Combined Memory)	9.567	9.567	9.567	9.567	9.567	7.644
DET(Separate Memory)	13.093	12.338	12.857	10.526	DNW	DNW
DET(Combined Memory)	13.156	12.703	12.471	10.519	DNW	DNW

Figure 18. Delay Analysis of Singleton Pipelined Hybrid Multiplier

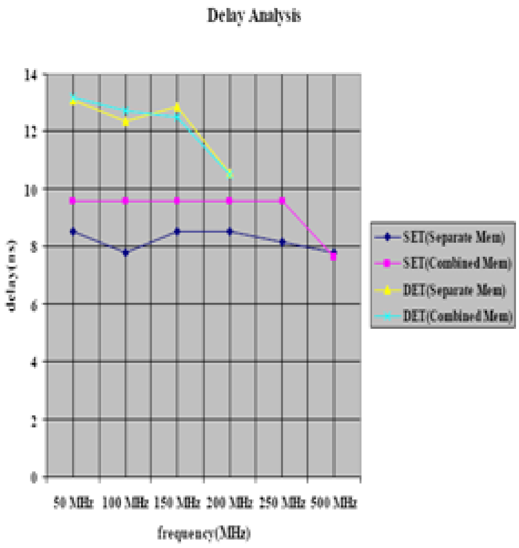
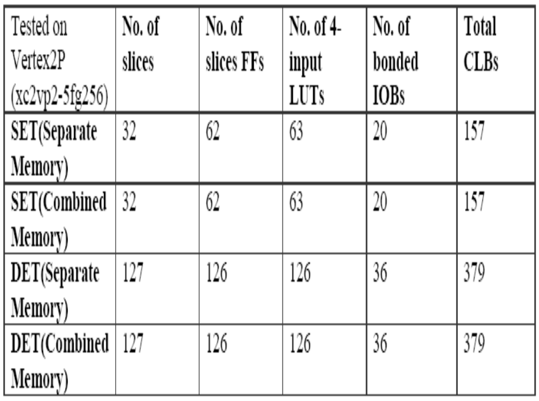


Figure 19. Delay Analysis Graph



Tested on Vertex2P (xc2vp2-5fg256)	No. of slices	No. of slices FFs	No. of 4-input LUTs	No. of bonded IOBs	Total CLBs
SET(Separate Memory)	32	62	63	20	157
SET(Combined Memory)	32	62	63	20	157
DET(Separate Memory)	127	126	126	36	379
DET(Combined Memory)	127	126	126	36	379

Figure 20. Area Analysis Singleton Pipelined Hybrid Multipler

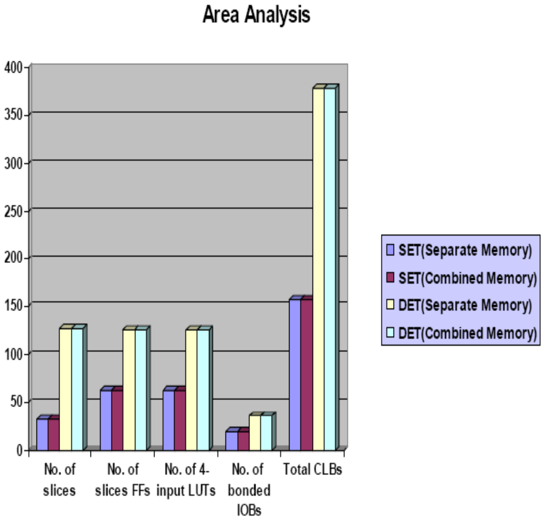


Figure 21. Area Analysis Chart

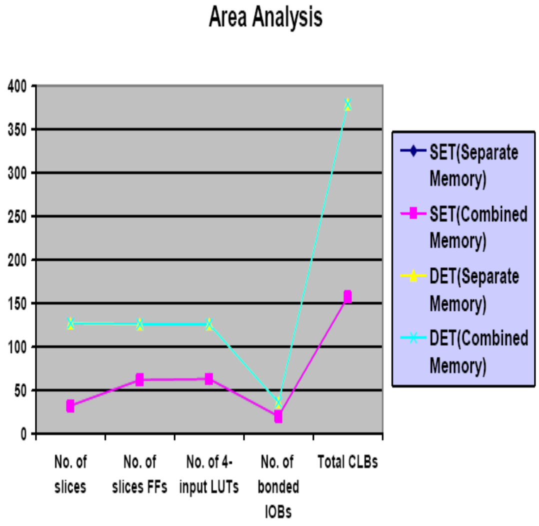
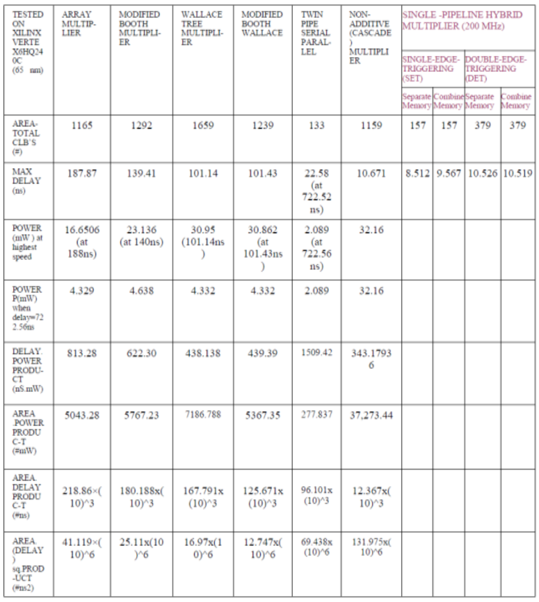


Figure 22. Area Analysis Chart

4. Result



TESTED ON XILINX VERTE X6HQ240C (65 nm)	ARRAY MULTIPLIER	MODIFIED BOOTH MULTIPLIER	WALLACE TREE MULTIPLIER	MODIFIED BOOTH WALLACE	TWIN PIPE SERIAL PARALLEL	NON-ADDITIVE (CASCADE) MULTIPLIER	SINGLE -PIPELINE HYBRID MULTIPLIER (200 MHz)			
							SINGLE-EDGE-TRIGGERING (SET)		DOUBLE-EDGE-TRIGGERING (DET)	
							Separate Memory	Combine Memory	Separate Memory	Combine Memory
AREA-TOTAL CLB'S (#)	1165	1292	1659	1239	133	1159	157	157	379	379
MAX DELAY (ns)	187.87	139.41	101.14	101.43	22.58 (at 722.52 ns)	10.671	8.512	9.567	10.526	10.519
POWER (mW) at highest speed	16.6506 (at 188ns)	23.136 (at 140ns)	30.95 (101.14ns)	30.862 (at 101.43ns)	2.089 (at 722.56 ns)	32.16				
POWER P(mW) when delay=722.56ns	4.329	4.638	4.332	4.332	2.089	32.16				
DELAY. POWER PRODUCT (nS.mW)	813.28	622.30	438.138	439.39	1509.42	343.17936				
AREA .POWER PRODUC-T (#mW)	5043.28	5767.23	7186.788	5367.35	277.837	37,273.44				
AREA. DELAY PRODUC-T (#ns)	218.86×(10)^3	180.188x(10)^3	167.791x (10)^3	125.671x (10)^3	96.101x (10)^3	12.367x(10)^3				
AREA. (DELAY) sq.PRODUCT (#ns2)	41.119×(10)^6	25.11x(10)^6	16.97x(10)^6	12.747x(10)^6	69.438x (10)^6	131.975x(10)^6				

5. Conclusion

In the above results, performance measures of all six multipliers and Single Pipeline Hybrid Multiplier are summarized and compared based on the Synthesis and Timing analysis reports. These results were obtained after synthesizing each architecture targeting towards Xilinx Vertex FPGA xc2vp2-5fg256. All these results are for processing 32-bits of data. It is observed that delay of SET & DET Pipeline Hybrid Multiplier is the least. Hence it is fastest among six multipliers.

The number of clock pulses gets reduced by half in DET Pipeline Hybrid Multiplier as compared to SET Pipeline Hybrid Multiplier. However, total area usage is more than the area in Twin-Pipe Serial Parallel Multiplier but less than all other multipliers. Also low frequency is used in this design which consumes low power and speed is improved through DET memory & Hyper-Pipelining concept which reduces the internal loop delay in single-micro-pipeline Hybrid Multiplier.
The protyping & verification has been done using following tools & technology by considering various design constraints:

- Target Device: xc2vp2-5fg256
- On Board frequency: 50 MHz
- Supply Voltage: 3.0 volts
- Worst Voltage: 1.4 to 1.6 volts
- Temperature: -40^{o}C to 80^{o}C
- Speed Grade: -5
- FPGA Compiler: Xilinx ISE 10.0i
- Simulator used: ModelSim10.0a
- Static Timing Analysis Tool: Prime-Time Tools

The limitation of this design is that the maximum frequency on which it operates is 500 MHz for SET Singleton Pipeline and 200 MHz for DET Singleton Pipeline.
To further enhance the High-Speed, Low-Area & Ultra-Low-Power of the Hybrid Multiplier, we need to design other variant of Singleton Micro-Pipeline and its DET Memory element by considering VLSI design matrix paradigm.

6. Reference

[1] Shailesh Shah, "Comparison of 32-bit Multipliers for Various Performance Measures", in Tehran 12th Int. Conf. on Microelectronics, MEng. Project Report, Concordia University, January 2000.
[2] Soojin Kim, Kyeongsoon Cho, “Design of High-speed Modified Booth Multipliers Operating at GHz Ranges”, World Academy of Science, Engg. and Tech. 61 2010.
[3] A. A. Khatibzadeh, K. Raahemifar and M. Ahmadi, “A 1.8V 1.1GHz Novel Digital Multiplier,” Proc. of IEEE CCECE, pp. 686-689, May 2005.
[4] S. Hus, V. Venkatraman, S. Mathew, H. Kaul, M. Anders, S. Dighe, W.Burleson and R. Krishnamurthy, “A 2GHZ 13.6mW 12x9b mutiplier for energy efficient FFT accelerators,” Proc. of IEEE ESSCIRC, pp. 199-202,Sept. 2005.
[5] Hwang-Cherng Chow and I-Chyn Wey,“A 3.3V 1GHz high speed pipelined Booth multiplier,” Proc. of IEEE ISCAS, vol. 1, pp. 457-460,May 2002.
[6] M.Aguirre-Hernandez and M. Linarse-Aranda, “Energy-efficient high-speed CMOS pipelined multiplier,” Proc. of IEEE CCE, pp. 460-464, Nov. 2008.

[7] Yung-chin Liang, Ching-ji Huang and Wei-bin Yang, “A 320-MHz 8-bit x 8-bit pipelined multiplier in ultra-low supply voltage,” Proc. of IEEE A-SSCC, pp. 73-76, Nov. 2008.
[8] S. B. Tatapudi and J. G. Delgado-Frias, “Designing pipelined systems with a clock period approaching pipeline register delay,” Proc. of IEEE MWSCAS, vol. 1, pp. 871-874, Aug. 2005.
[9] Nikola Nedovic, Marko Aleksic,Vojin G. Oklobdzija, “Comparative Analysis of Double-Edge versus Single-Edge Triggered Clocked Storage Elements,” IEEE Adv. Comp.Sys. Engg. Lab., University of California Davis, CA 95616, 2002.
[10] P.R. Panda et al., Power-efficient System Design, chapter 2, pp. 11- 39. Springer Science+Business Media, LLC 2010.
[11] Darren Zacher, “How to use register retiming to optimize your FPGA designs,” Mentor Graphics, Dec. 14, 2005.
[12] Tobias Strauch, “Hyper Pipelining of Multicores and SoC Interconnects,” Germany, Oct. 2010.
[13] Seongmoo Heo, “A low-Power 32-bit Datapath Design,” © August 2000.
[14] http://www.arm.com/arm_arm.pdf
[15] John P. Hayes, “Computer Architecture and Organization,” Mc-Graw Hill, 3rd edition, pp. 308.
[16] G.E. Tellez, A. Farah and M.Sarrafzadeh, “Activity –driven clock design for low power circuits”, in Proc. IEEE ICCAD, San Jose, pp.62-65 Nov. 1995.

[17] S.H. Unger, “Doule-edge-trigger filp-flops”, IEEE Trans. Computers, vol.30,no.6, pp.447-451, June 1981.

[18] S.L.Lu and M. Ercegovac, “A novel CMOS implementation of doube-edge-triggered flip-flops”, IEEE J. Solid-State Circuits, vol.25, pp.1008-1010, Aug.1990.

[19] M.Afghahi and J.Yuan, “Double-edge-triggered D-flip-flops for high-speed circuits”, IEEE J. Solid-State Circuits, vol.25, pp.1168-1170, Aug.1991.

[20] A.Gago, R.Escano and J. A. Hidalgo, “Reduced implementation of D-type DET flip-flops”, IEEE J. Solid-State Circuits, vol.28, no.3,pp.400-402, Mar. 1993.

[21] R.Hossain, L. D. Wronski and A. Albicki, “Low power design using double edge triggred flip-flops”, IEEE Trans. VLSI Systems, vol.2, no.2, pp.261-265, June 1994.

[22] Massoud Pedram, Qing Wu, Xunwei Wu, “A new design for double edge triggered flip-flops”, Dept. of Electrical Engineering- Systems, Univeristy of Southern California Los Angles,CA 90089, USA.

[23] Wai Man Chung, “The Usage of Dual Edge Triggered Flip-flops in Low Power, Low Voltage Applications”,A thesis presented to the University of Waterloo, Master of Applied Science in Electrical and Computer Engineering Waterloo, Ontario, Canada, January 2003.

[24] G. M. Blair, “Low power double edge-triggered flip-flops,” Electron. Lett. , vol.33, no.10 , pp.845-847, May 1997.