High Speed and Area Efficient design of DETMB-FF with clock gating

A
Dissertation
Submitted in the fulfilment of the requirements
For the award of degree
Of
MASTER OF TECHNOLOGY
IN
VLSI DESIGN AND EMBEDDED SYSYEMS

Submitted by:
RAVULA SHIVARAMA KRISHNA
2K19/VLS/14

Under the supervision of
Prof. RAJESHWARI PANDEY
(ECE)

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042
JUNE, 2021

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

CANDIDATE'S DECLARATION

I, **Shivarama Krishna(2K19/VLS/14)** student of **M.Tech (VLSI Design and Embedded Systems)**, hereby declare that the project Dissertation titled "High Speed and Area Efficient Design of DETMB-FF with clock gating" which is submitted by me to the Department of Electronics and Communication Engineering, Delhi Technological University, Delhi in partial fulfillment of the requirement for the award of the degree of Master of Technology, is original and not copied from any source without proper citation. This work has not previously formed the basis for the award of any Degree, Diploma Associateship, Fellowship or other similar title or recognition.

Place: DTU **Shivarama Krishna**

Date:18-06-2021 **2K19 /VLS/14**

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

CERTIFICATE

I hereby certify that the Project Report titled titled "High Speed and Area Efficient Design of DETMB-FF with clock gating" which is submitted by **Shivarama Krishna**, **2K19/VLS/14** of Electronics and Communication Department, Delhi Technological University, Delhi in partial fulfillment of the requirement for the award of the degree of Master of Technology, is a record of the project work carried out by the student under my supervision. To the best of my knowledge this work has not been submitted in part or full for any Degree or Diploma to this University or elsewhere.

Place: DTU

Date: 18-06-2021.

SUPERVISOR

Prof. RAJESHWARI PANDEY

(ECE)

ACKNOWLEDGEMENT

A successful project can never be prepared by the efforts of the person to whom the project is assigned, but it also demands the help and guardianship of people who helped in completion of the project.

With profound sense of gratitude, I thank Dr.Rajeshwari Pandey(Professor) my Project Supervisor, for her encouragement, support, patience and her guidance in this project work. I take immense delight in extending my acknowledgement to my family and friends who have helped me throughout this project work.

Shivarama Krishna

2K19/VLS/14

ABSTRACT

Two efficient low power design strategies are Data-Driven Clock-Gating (DDCG) and Multi-bit Flip-Flops (MBFF), in which various FF is grouped with one clock. Even if the VLSI designers are often used, they are typically handled individually. The previous study has focussed on the use of MBFF in RTL, gate-level, and layout. Conflicts and contradictions were created by studying all aspects of the common design as a whole and trying to wrap all of that up into one architecture of the internal circuit of MBFF, its multiplicity and its synergy with FFs data have not yet been studied. Maximizing the savings by developing a strategy combining DDCG and DET-MBFF. The DET-FFs should be grouped in MBFFs in increasing order of their activities to optimize the power area delay savings. The algorithm was used in a realistic design flow to create a power savings model based on DET-MBFF multiplicities and FF toggling probabilities. By using the Xilinx ISE unit, we were able to save power area delay compared to designs with ordinary FF.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

The use of multi-bit flip-flops (MBFFs) as the modeling method that significantly decreases the strength of digital systems has been stressed by an article recently published. In Flip-Flops (FFs), the data on a digital device is normally stored, each with its own internal clock control unit. In Fig. 1.1 there are two cascaded master and slave latches in a single-bit edge-triggered FF powered by the CLK and CLK opposite clocks. The most important source of energy from the FF is seen by its internal clock drivers, which contribute significantly to the overall energy consumption.

In the case of flip flops (FFs), data on digital systems are generally stored with their own internal clock drivers. A multi-FF (MBFF) module containing the clock control drivers of all underlying FFs can be grouped into an effort to reduce the clock capacity. We indicate that kFFs are grouped into a k-MBFF. Kapoor and others. In the past, digital SMPS monitoring is done by using the Digital Signal Processor general purpose (DSP). Digital algorithm, housekeeping, supervisory duties and correspondence is attempted using DSPs. This method is unsuitable in most industrial cases because of many disadvantages and shortcomings, apart from some small uses. The following are: a single unit that limits the speed of calculation leading to a low bandwidth of control, excessive delays in the case of multiple converters, limited capacity to generate non-sequential pulse as might be necessary in the case of non-linear control, limited capacity for high resolution and reduced number of control channels.

A locked, dedicated controller for a particular application like Voltage Regulator Module is another solution for modern digital power management (VRM). The disadvantage of this solution is that it is restricted to an own use. Applying the device to address other Power Management issues is therefore impractical because, for any cases that registered a 15 per cent reduction in overall dynamic power in the 90-nm processor architecture, a new Application Specific Integrated Circuit (ASIC) design cycle must be initiated. MBFF characterization is assisted by electronic design automation instruments such as Cadence Liberate. The MBFF's advantages are not secure. By exchanging ordinary drivers, the slew rate of the clock is degraded and hence a longer clock-to-Q delay is caused. The internal MBFF drivers should be reinforced for any additional power to correct that. The application of MBFF at the RTL design level is also

advised to escape the time-closing obstacles resulting from the implementation of the MBFF at the backend design stage. And the average FFs toggling rate from data to clock is very small, usually between 0.01 and 0.1. The clock gating isn't easy. For clock enabling signals, additional logic and interconnections are essential to take into account the resulting area and power overheads. Each clock input of an FF may be independently disabled in an extreme case to achieve optimum clock removal. However, this results in a high overhead; thus the aggregation of multiple FFs in an effort to reduce the overhead by sharing a shared clock disabling circuit. In the other hand, the disabled efficiency of such groups can be reduced because the clock is disabled only in the times where the inputs in the system to all the FFs do not shift. In the worst case, the clock disabling likelihood equals the product of individual probabilities while the amount of involved FFs rises quickly to zero when the feedback of the FFs is statistically independent. Signals obtained from FFs are also preferable when they are significantly correlated.

The method for handling the transfer of FFs, such as in microprocessors and controllers, is dependent on the details. It therefore necessitates advanced calculations and statistics, as shown in this article. A great solution is to a circuit containing many FFs (like a register) to use bits that behave similarly is to block the clock entry. registers registered on the same clock are the most power-efficient Additionally, it is a simple task to disable the disabling sound. Clock-gating architecture presents an alternative challenge in this case because the amount of design work required is much larger. This has its origins in the non-reprogressive essence of the control logic An efficient three-dimensional graphics accelerator and a 16-bit microcontroller have both been used as prototypes to test the gating technique presented in this article. These units have been configured to fully inform the internal data dependency and have been specified within the RTL code for suitable clock supporting Signals. After compiling the RTL code and simulating it at the gate stage, significant "secret" disabling possibilities were found.

The savings in clock power still override the short circuits that the data is switched to. Logical, structural and FF considerations can lead to an MBFF classification. Although FFs at the layout-level grouping have been extensively investigated, there was less attention paid to the front-end effects of MBFF groupsize and how they affect the clock-gating. Two questions are answered in this brief. First, the optimum bit number k for the clock-gated (DDCG) clock-gated (MBFF) data. Secondly, how to make maximum savings on the basis of a toggling rate from data

to clock (also termed activity and data toggling probability). You will find an MBFF in the RTL logic synthesis stage Power optimization in modern nanometer IC architecture is still one of the most critical design goals. The efficiency of using multi-bit flops in order to save the clock network power usage has been shown in recent studies. In previous works, however, multi-bit flip flops were used at earlier design stages, which may be very difficult to achieve the balance between power, time and other design goals. This report proposes an innovative method of optimising power by applying more multi-bit flip-flops gradually in the post-placement phase in order to save clock power while taking account of the positioning density and time-slow constraints. Our solution is very powerful and reliable, and can be combined in a seamless way with the modern design flow for a 55-nm 230-MHz device design on a chip, according to experial results on industry benchmark circuits.

A number of flipflops can be clustered into a design such that all similar clock drivers are combined with all flipflops, in an attempt to cut down on the number of time used by clocks. The example in Figure 1 illustrates the concept of 1-bit FFs, also known as double-bit, bit, and quad-bit FF. Further, one can combine FFs into 4-bit MBFFs, as well as 8-bit FFs Thus, the abbreviation "k-MBFFB is denoted as k-MBB after k." Additionally, MBFF minimises the clock tree impedance For more than one fan, only one cable is required. Decreases the depth of the tree, its buffer size, and the number of subtrees, as well. For additional energy savings, the features were limited in the silicone region.

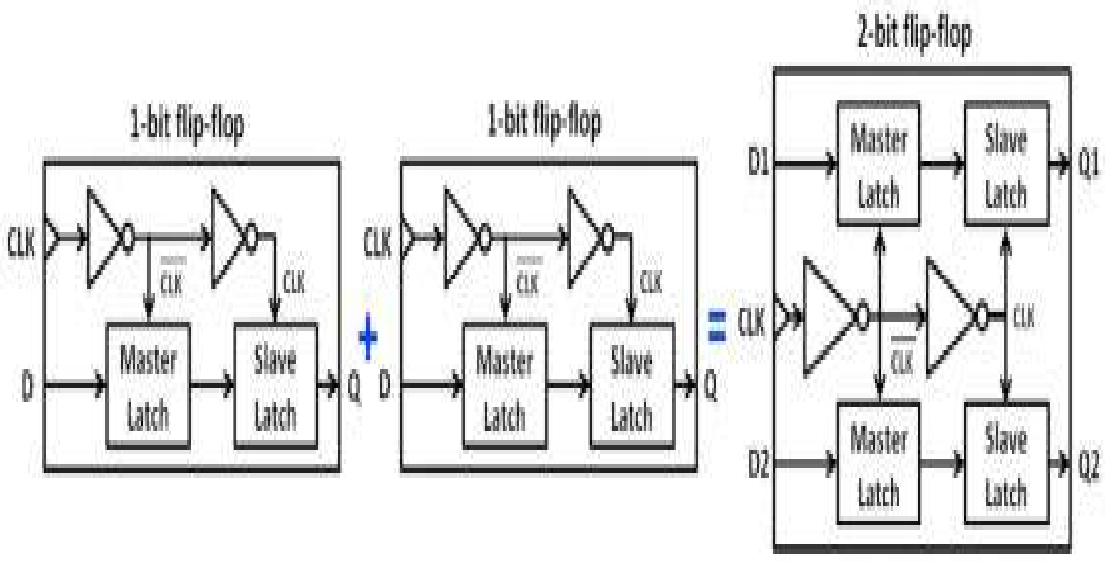


Fig. 1.1. 1-bit FF and 2-MBFF.

Logical, structural and FF considerations can lead to an MBFF classification. The level FF grouping has been well-researly analysed, but the front-end effects of MBFF scale have not been much examined. Two questions are answered in this brief. CDM bit multiples are. Secondly, how to make maximum savings on the basis of a toggling rate from data to clock (also termed activity and data toggling probability). FF 55-nm on the system-on-on-a-a-a-a-chip designs using an MBFF grouping scheme. Both 2-MBFFs and 4-MBFFs with a 20% boost in tpCQ have been used. Convergence was reduced from a strong 13 percent of a stronger dynamic force by losing a small amount of power. The solution was to apply low voltage threshold cells on critical pathways, which increase the leakage capacity slightly. Even because of the correction, the gross area rose by 2.3%

Benefits are not only provided. To incorporate the same logic architecture in 90nm, the clock signals are coupled together, triggering an increased dynamic current and a reduced clock-to-Q propagation delay, as seen in the diagrammed (short-circuit) slope of the PQ t pin. As a result of the extra power consumption the amount of pCQ per bit is increased by 20 percent. On the average, though, due to the fact that the offsetting of a clock is held at an extremely low ratio, clock savings often outweigh the short-power penalty in a frequency-variable design.

This study responds to two questions: how to make use of the experience of the FF's average toggling data clock ratio (also known as activity-toggling probability) in the underlying architecture and what should the optimum bit multiplicity of MBFF's be? The MBFF internal derivers can be strengthened finding a creative way to get around the short-circuit power penalty loss of pCQ t by increasing loads. The larger 2-MBFF drivers as against 1-bit illustrated this figure in Fig. 1.1. The MBFF The toggling of data is dependent on the probability of k. See (2a) the dependence to increase the power savings and to improve the design flow of MBFF. This has not been researched to our best knowledge until now. Tools like Cadency Freed help MBFF characterization, such as ElectronicsDesign Automation (EDA). The new Cadence and Synopsys HDL compilers enable MBFF gate-level architecture. Their internal logic and FF grouping algorithms have not been released in MBFFs. Despite their relevance, the MBFF multiplicity and classification innovators, designers and entrepreneurs at the front end of the concept devoted relatively little coverage in literature. Logical, procedural and FF activities can be guided by the MBFF classification.

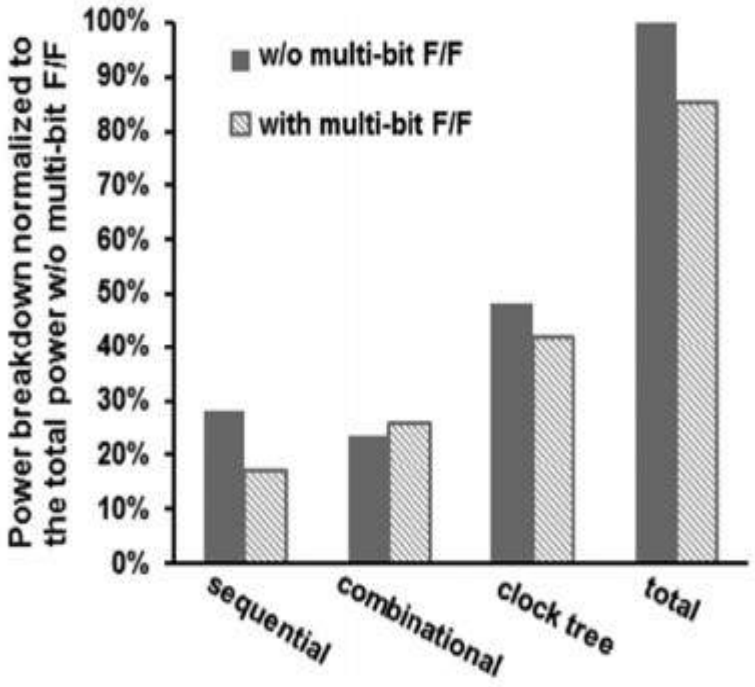


Fig1.2: MBFF power disruption in comparison with popular 1-bit FFs.

The architecture study divided 92 percent of the FFs into MBFs, of which 4-MBFFs was for the most part, whilst the reset was 2-MBFFs. The power disruption of the MBFF as seen in Fig. 1.2 in relation to 1-bit FF. This is normalised for a 1-bit FF core architecture for the overall power consumed (memories and IOs excluded). The overall dynamic power reduction is seen by 15 percent. The sequence logic and clock tree power was expected to decrease since there were a reduction in the overall numbers of clock drivers and the wire load attached to the internal MBFF drivers. The combined logical capacity was improved by upgrading some of the logic to restore pCQ t. We propose to add MBFF at the RTL architecture stage to prevent the time degradation caused by the pCQ t rise. This enables the design stage of the backend and layout to account for pCQ t and prevent time problems at an early stage.

A work was proposed to present the MBFF at the logic synthesis phase, trying to end on the advantages and disadvantages of a synthesis using ordinary FFs. The FF's have been mapped to MBFF using the RTL compiler's Gate Level design. There has been an experimental 55nm 230MHz chip system architecture (SoC). The authors limited MBFF mapping to FFs from the same bus where 2-MBFFs and 4-MBFFs were used and their pCQ was 20% more. Using MBFFs, the number of sinks decreased by 60% and resulted in an easier clock tree, with fewer buffers of 35%. The clock skew was further reduced by 30%. The energy savings are

summarised in Table 1. A 13 percent decrease in dynamic power is presented. It is hardly shocking that energy savings were attributed to timing loss, which was remedied by the introduction of LVT cells on sensitive routes, suggested by increased leakage capacity.

		Single bit	Multi bit	Difference [%]
Number of FFs	1-bit	29437	5267	
	2-bit	0	1860	
	4-bit	0	5216	
Total FF area [u^2]		394047	402955	+2.26
FF's threshold type [%]	Standard	11.5	49	
	High	88.5	51	
Clock buffers in tree		934	624	-33.4
Skew [pSec]		249	176	-29.4
Seq. CKT power [mW]	dynamic	145	104	-28.16
	leakage	4.55	9.89	+117
Combinational power [mW]	dynamic	134	117	-12.4
	leakage	28.9	28.8	-0.35
Clock Tree (sinks incl.) [mW]		195	134	-31.4
Total dynamic power [mW]		446	388	-13.0
Total chip power		682	629	-7.77

Table 1. MBFF feature reduction of power

Though MBFF handles RTL synthesis in the gate-level implementation, the architecture flows provided by EDA tools take very small physical layout information into account. Most important, these tools totally neglect the possibility of data toggling that subsequently impact the MBFF community. The literature which follows is mostly based on the physical execution of MBFF.

These works neglect the behaviours of FFs that this project takes into account. One of the earliest works in the physical design was defined on MBFF grouping. The time limits derived from the 1-bit FFs layout were correlated with each FF. Linked wires with FF data input and output, anchored on the other side to the rest of the logic, were permitted to pass around the direction of the FF, thereby specifying the lay-out area where FF growing be moved without timing breaching. The fusion of FF couples into 2 MBFFs has been formulated to maximise the number of FFs fused, so that there is no breach of the timing of the resulted MBFF sites.

Congestion restrictions also existed, as the silicon field was divided into small bins with minimal MBFF occupancy. With the assistance of region research the problem was solved. Using the above ad hoc technique, an algorithm with improved computer performance proposed later in a work to resolve the same problem with the same timeframe and region restrictions.

While MBFF handles RTL synthesis in the implementation of gate levels, very limited physical layout information is used in the architectural flows supplied by EDA software. Most importantly, the risk of data toggling which would then affect the MBFF population is completely neglected by these methods. The following literature is focused largely on the physical performance of MBFF. The pairing of MBFF with CG was discussed recently. CG cloning, along with a 1-bit FF proximity study, is proposed in the MBFF grouping decision. Considerations of proximity to the layout also determined that it was in order to group 2-MBFF or 4-MBFF. There's a responsibility and a very limiting practicality of using a converted timing layout as a starting point for the MBFF design flow. Timing limitations can be very close, restricting the fusion of possible FFs. In addition, a timely layout eliminates the stimulus to alter the template.

No FF operation is regarded as a driving force in grouping MBFF. Our work suggests a formal MBFF grouping algorithm, with the result that the energy savings predicted can be demonstrably maximised. We believe that MBFF should be implemented at the level of the RTL and logical architecturesuch on functional, on-architectural principles above all, FFs activities. The remainder of the document is arranged like this. The impact on future energy savings of data topping probabilities and Section 3 demonstrate the way Data-Drived Clock-Gating (DDCG) is combined with MBFF. The FFs are to be grouped into an MBFF DDCG 5.

2. Effects of data reversal on energy efficiency

Fig 1.3, obtained from SPICE simulations, shows that possible energy savings of MBFF are dependent upon its toggling chance. It displays the 1-bit ff, 2-MBFF and 4-MBFF energy consumption. Note the volatile "foundation" energy, independent from the input operation, charged per clock. In Figures 1.3 (b) and 1.3 (c), the growth of base energy of 2-MBFF is due to its greater internal load in comparison to one-bit FF in Fig.1.3(a). The energy consumption is expected to increase linearly with the probability of data toggling, which is twice higher if both inputs toggle in comparison with one toggling of the data. For Figs.1. 3(d) and 1.3, 4-MBFF shows similar behaviours (e).

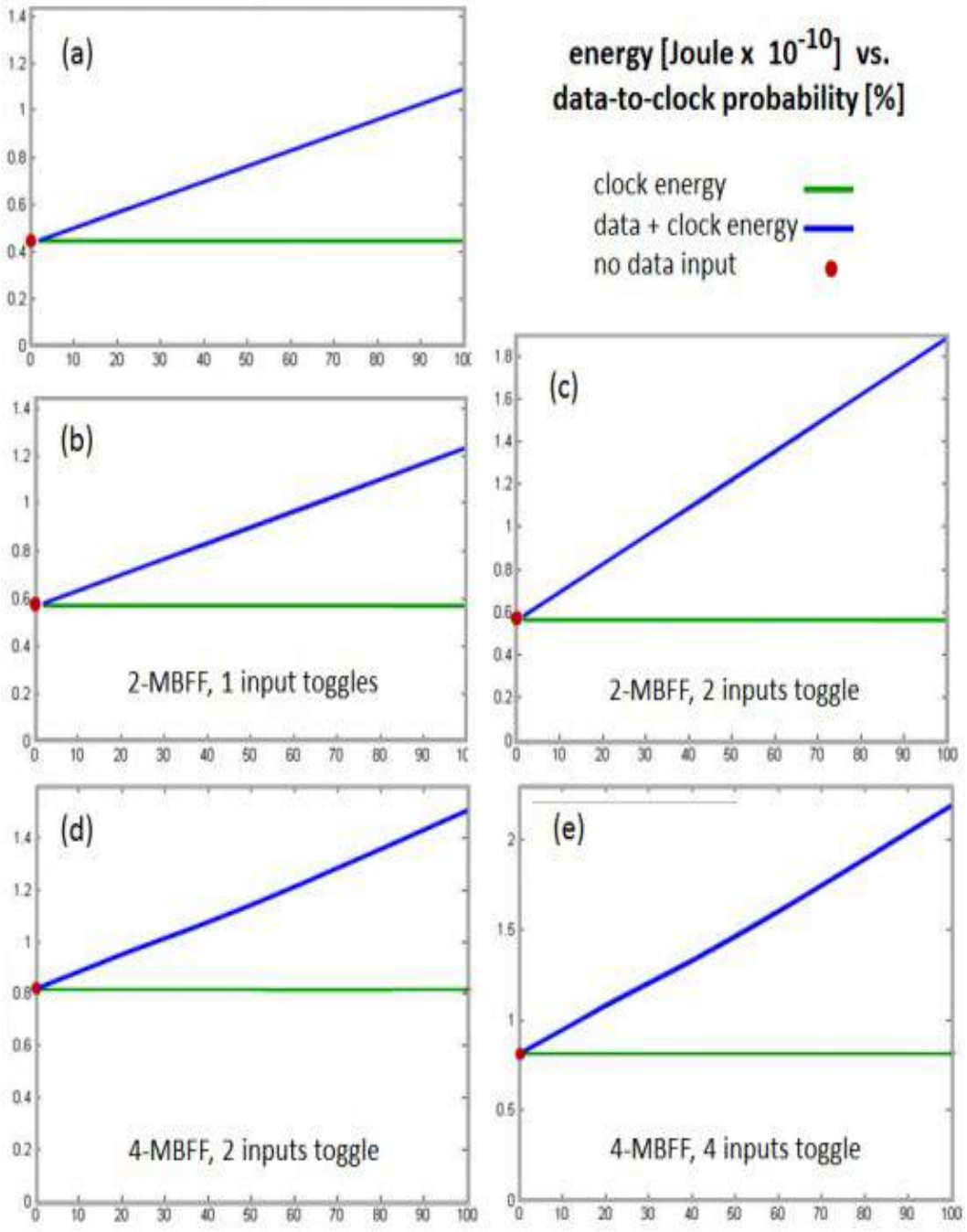


Fig. 1.3: The reliance on the toggling chance of the MBFF energy savings.

Let p be the possibility to move from data to clock. Notice that 1-bit FF consumption is anticipated by E1. In Fig. 1.3(a) we assume that

$$E_1(p) = \lambda_1 + \mu_1 p, \qquad \text{……..(1)}$$

Where 1λ bis is the flipflop internal clock driver's energy, and 1 bis is the data toggling energy. There are three scenarios for 2-MBFF: no FFs switch, just one FF switch and the two FF switches. The predicted energy usage E2 is assuming data to rotate independence

$$E_2(p) = \lambda_2(1-p)^2 + 2(\lambda_2 + \mu_2)p(1-p) + (\lambda_2 + 2\mu_2)p^2 = \lambda_2 + 2\mu_2 p, \quad \text{……..(2)}$$

When 2λ is the internal driver power of the clock, and 2μ is the per Bit data energy supermoderation k-FF: in the majority of applications is derived directly from data sent to the hardware. MBFF should be kλ and per-bit data toggling power should be kμ For all the variations of toggling FFs, the energy required is

$$E_k(p) = \sum_{j=0}^{k} (\lambda_k + j\mu_k)\binom{k}{j} p^j (1-p)^{k-j} = \lambda_k + k\mu_k p. \quad \text{……(3)}$$

In equation (3), a number of rearrangements is applied

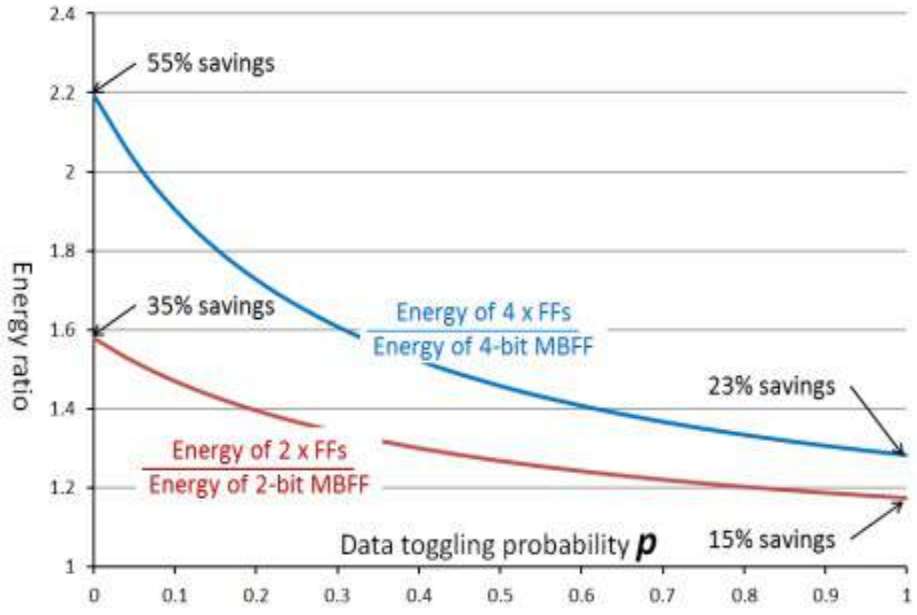


Fig. 1.4Software toggling's power savings [8].

Fig. 1.4 displays the energy ratio of 2 and 4 1-bit FFs to that of 2-MBFFs and 4-MBFFs for assessment of possible MBFF energy savings. Via the power of one's imagination k human FFs, we split the a small amount of energy disparity k and the k-MBFFs. In case of limited p,

savings of (1.6 − 1)/1.6 = 30% for 2 pikes and (2.2 − 1)/2.2 = 55% for 4 pikes was seen. Savings for high p are (1.18 − 1)/1.18 = 15% for 2 pounds and (1.3 - 1)/1.3 = 23% for four pounds.

The mean P does not surpass 0,05 for modern VLSI schemes, so that large savings can be achieved. Section 4, which takes account of MBFF energy savings by adding DDCG, generalises a power usage model for seven different probabilities of data toggling. It can also be noted that the overriding freedom is the weakest presumption that the association of toggling FFs will really be used to increase energy consumption.

CHAPTER 2
LITERATURE SURVEY

[1] High performance mixed signal platforms digital systems power management

Mixed signal (HPMS) high performance platforms need stringent overall device performance and subsystem performance. A wide variety of platforms, including consumers, mobile systems, recognition, health goods and microcontrollers are used to build high-performance systems. In this article, we provide an overview of strategies, difficulties and possibilities of low power design in an industrial testing environment. The project addresses solutions to implement low-power technology ranging from power optimisation scenarios that cover active and standby operating modes to developing a multi-core, low-voltage architecture.

[2] The Optimal Fan-Out of Clock Network for Power Minimization by Adaptive Gating

The clock signal measurement on VLSI chips is currently a mainstream architecture approach to reduce power consumption in switching systems. In this article, we build a probabilistic clock gating network model that enables us to measure the energy savings and the overall implied. Power savings expressions are provided in a gated clock tree and the optimum gater fan-out is obtained from the flip flops that topass process technology parameters and probabilities. The corresponding technique of clock gating saves the switching capacity of the overall clock tree by 10%. There are discussions about the scheduling ramifications of the new gating system. There is also a discussion on the grouping of FFs for a combined gating. These research and findings fit the test data gathered on the 65-nanometer technology of a 3D graphics processor and the 16-bit microcontroller.

[3] The effect on physical synthesis of multi-bit flip-flop use.

Reducing network clock power is an effective way to minimise high-frequency ASIC's energy consumption since it accounts for an extensive amount of dynamic chip power. Recently it has been shown that multibit flipflops (MBBF) are an important design technique for improving synthesis of clock tree and can either be used as an alternative solution, or as part of the well-known clock gate approach, aimed at reducing clock capacity. The concept behind this strategy is to save clock tree power by using optimised designed flip flop and even a reduced clock tree until the number of clock sinks is smaller, in a cell configuration. There have been several recent

works suggesting methods to use MBFFs in traditional cell based models, which substitute single-bit flip flops in logic and/or physical systems with MBFF cells. However, for various design phases it is also necessary to analyse the effect of MBFFs on physical design.

[4] Construction of constrained multi-bit flip-flops for clock power reduction

Given the congested congested constraints of unreleased bins, the length restrictions of the input and output signals of all 1-bit flip flops, an effective two-phase approach to obtain the final multi-bit flip flops is suggested based on the disposal function of the redundant inverters when melting 1-bit flip flops into multi-bit flip flops. The experimental results show that our proposal removes 68 percent of inverters in order to preserve their synchronous designs and saves on average 19,75 percent in clock capacity for two examples in a fair CPU period, compared with the original design in the number of inverters for two examples checked.

[5]INTEGRA: Clock Power Saving Fast Multibit Flip-Flop Clustering

Clock power is the most dynamic power contributor to the new integrated circuit architecture. A traditional single-bit paddle cell uses a high-drive clock signal inverter chain. By clustering several of these cells and creating a multiplex flip flop, the driving force, dynamic power and inverter chain region can be shared and even the power of the clock network can be saved and skewed. Therefore, we are focusing in this project on multi-bit postponement clusters in order to achieve these advantages. We model the problem instance by two interval diagrams and use a pair of linear sequences to describe it, using the properties of Manhattan distance and co-ordinate transformation. We only define partial sequences required for clustering flip flops without listing all possible combinations, which leads to a successful compositional arrangement. In addition, our rapid coordinate transformation is also very effective in implementing our algorithm. Experiments on industrial circuits are performed. Our findings suggest that succinct representation is more efficient and accurate. Clock power savings from multibit flip-flop clustering can also be significant at postplacement, even though timing and placement density are constrained.

[6] Pulsed-Latch Replacement Using Concurrent Time Borrowing and Clock Gating

Flip-flops are the most common sequencing components, but with their overheads, delay, power and field, are considerably higher than latches. Pulsed latches are also an excellent way of reducing power on high-performance circuits. In this project we use the inherent time borrowing property of pulsed locks to conserve power and to compensate for timing violations entirely and

we take a clock gating into account when replacing pulsed lock. Experimental findings demonstrate that our strategy will produce powerful results.

[7] Design Flow for Flip-Flop Grouping in Data-Driven Clock Gating

The gating of the clock is a dominant power saving strategy. It is found that there is a significant number of redundant clock pulsations in the widely employed synthesis dependent gating. The aim is to disable data-driven gating. Flip Flops (FFs) are clustered to share a shared clock allowing signal to reduce the hardware overhead involved. A previous project answers the issue of the group size that maximises electricity savings. Here we reply to the question of which FFs should be grouped to optimise the reduction of power. We propose a realistic solution based on the associations between the behaviour and physical location of FFs within the design. We propose a practical solution. Our data controlled clock ratio is embedded into the commercial backend design flow of Electronic Design Automation, achieves a total power reduction of 15%-20% in 40 and 65 manometer process technologies for different kinds of modern industrial and academic design. These savings are obtained in order to achieve an overall power saving technique on the sClock gate. It is found that a significant number of redundant clock pulses remain in the normal synthesis-based gating. The goal is to discapacitate data-driven gating. Flip-flops (FFs) are clustered to share a similar clock activation signal to reduce the hardware overhead involved. A previous project answers the question of the group size which maximises power savings. Here we answer the challenge, to optimise power reduction, of which FFs should be put in the set. We provide a realistic approach based on the association between operation of FFs and the restrictions on their physical location in the structure. Our data-driven clock gating is built into a business backend design flow for Electronic Design Automation (EDA). The power reduction for different forms of state of the art industrial and academic design in 40 and 65 manometer process technologies is 15 percent to –20 percent overall. This saving is made by means of the synthesis of clock gating done by commercial EDA instruments and the gating inserted manually in the architecture of the register level switch. Savings achieved by synthesising clock gates using commercial EDA techniques and by manually inserting gate into the transference level register architecture.

[8] Effective and Efficient Approach for Power Reduction by Using Multi-Bit Flip-Flops

Power in modern VLSI architecture has become a raging issue. The power used by clocks increasingly takes on a dominant role in modern integrated circuits. Given a specification, the power consumption can be reduced by replacing a few flip-flops with less flip-flops with several bits. The output of the initial circuit could be affected by this process. Therefore, replacing the flip-flop without breaching timing and positioning capacities is a very complicated problem. We also suggested a number of methods to deal with the problem effectively. Initially, we carry out a coordinated transformation to recognise certain twists that can be combined with their legal regions. Furthermore, we demonstrate how we can build a combination table for listing potential combinations of library flip flops. We use a structured way finally to fusion flip flops. The goal of minimising the overall wire length is also considered in addition to power reduction. Our algorithm's time complexity is less than that of the empirical complexities of the term (n1.12) (n2). Our algorithm cuts clock power dramatically by 20-30 percent, and the working time is very low, according to experimental findings. Our algorithm takes just about 5 minutes to replace flip flops and the power reduction may reach 21 per cent in the most important test case, which includes 1 700 000 flip flops.

CHAPTER 3

CLOCK GATING FLIP-FLOPS

Flip-Flops: Flip-flops are a logical gate application. You can construct memories for them using Boolean logic. The most fundamental concept of a random access memory [RAM] can also be considered as Flip Flops. If a certain input value is given, the logical gates are correctly programmed to be remembered and implemented. Greater use of flip flops helps to design improved electrical circuits.

The most famous use of flip flops is where a feedback circuit is implemented. As the input principle is used to design a memory, flip flops may be used.

MULTI-BIT FLIP-FLOPS

Multi-bit Flip-Flops reduces power use since the inverter is shared in the flipflop. Clock skew is also minimized at the same at the same time. Multi-bit flip-flop and sngle have the same clock status. The state set and reset is the same too. Figure 1 shows the case of multi-bit flops. 1 - 2-bit flip-flop consists of a single bit flop blending. It shares the buffer with the clock and can reduce power.

Advantages of flip-flop

1. Exclude inverters repeat.
2. The total flip-flop area can be decreased.
3. The mutual inverters optimize the electricity.

ALGORITHM.

It is divided into three stages, as first the combined flip-flops are identified. Secondly, in the first step we will build the combined table by overlapping area. For easy representation we can construct the combination table in binary tree display. In the third stage, combining flip flops are created based on the combination table.

IDENTIFICATION OF MERGEABLE FLIPFLOP.

The selection of flip flops for combining is achieved on the basis of the flip flops used in digital systems. Each flip flop has its own separate clock during identification

COMBINATIONAL TABLE.

By putting the combinational circuit together, we achieve the optimal result. If the flip flops are fused with no mixing table, so this is not effective since the intersection value of meldable flip flops is not present. Based on the library initialization value, the combination table is made. We create potential flip-flops based on library bit values. The algorithm is called L, and denotes the one-bit table distance. The sum is minimal and the size of the library is initialised since we combine the number of a bit of flip flops In the following figure, dual-bit cell 3.1 is shown. It has two entry data pins, two pins, a pin for a clock and a reset pin. Dual-bit flipflops, on the other hand, provide better value and pay virtually no extra in weight. The truth table of the two-bit flip-flop cell can be seen in Figure 3.2. When it's 6, the values of Q1 and Q2 will flow into D1, and the value of Q will stay the same for Clock.

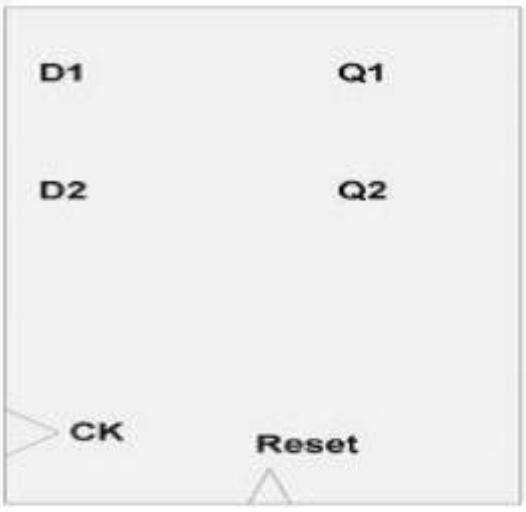


Figure 3.1: A dual-bit flip-flop cell

CK	D1	Q1	D2	Q2
↑	L	L	L	L
↑	L	L	H	H
↑	H	H	L	L
↑	H	H	H	H
↓	X	D1	X	D2

Fig.3.2 Table : The true table of dual-bit flip-flop cell

Clock Gating to Flip-Flops.

In the phase of device level, clock signals are to be activated and working block modules can effectively be captured. This could not be necessary. These signals are triggered in the form of gate levels in the clock. The architecture consideration immediately adds the clock signal to the other instruments. The circuit already has those at the highest degree floating. In this case, when clock signals are enabled, we must quantify complex electricity usage in the circuit. This time assesses the clock gate which calls for the FF Pre-Load and Assessment Statement review and specifications.

By XOR-ing the output of the current data supply, the clock will be deactivated in the next loop and shown in the next cycle. Then the display of the XOR gates is OR-ed for the FF gate signal to be used to prevent crashes. By combining LATCH with AND gate, the Integrated Clock Gate (ICG) may be used by environmental instruments. These latches may be used for optical filter in ultra-low energy applications. In these implementations the data-driven clock gating signal is used as an allowable signal. The number of clock pulses may be disabled. There would be a trade off with ICG. The bursts may also be a compromise for the overhead circuitry. While the flipflops increase, the overhead hardware decreases to get signals through ORing.

Signals are not permitted as free of charge. Logics and interconnections can be desired to allow such signals and range and power concern can be taken into account in performance. In certain operations, the FF has been provided with individual clock input and absorbs more power. The division of these clocks has also increased in scale. This can lead to high output

overhead. This reduces the clock load by using the Flip-Flops shared circuits. Tiny amount of electricity may be consumed.

The clock registries and the state used for clock gating are connected. To obtain the clock gating from the situations under which the imperative design can be used. This method saves the strength of the logic circuit and a vast amount of MUX's. The Clock Gating Signals from the CDN can be substituted for these circuits. The overall shape of the ICG also allows these signals to be distributed as part of the CDN to the interchange levels. Since the clock level changes the arrangement of the clock tree and it is kept in the same tree.

The level of the clock match logic with the tactic is the following:

1) The RTL level code must allow the logic level synthesis to be reached.
2) Special modules or registers that ICG can manage as a library feature may be designed.
3) Semi-automatic insertion of the electronic clock gate and generation as ICG cells. So the RTL level is allowed or the ICG for optimisation is included in the level

Data Driven Clock Gating for single Flip-flop.

Data powered gating causes area and overheads of power to be taken into consideration. It is suggested to group multiple FFs to be controlled by the same clock signal, provided by the activating signals of the individual FFs, to try to minimise overhead. However, this may reduce the disabling performance. The FFs whose shifting activity is strongly correlated with a joint activating signal are also advantageous. A recently published project develops a model for datadriven gating based on the toggling operation of the component FF. The optimal clock gate fan out which produces a maximum energy saving is extracted from the average switching statistics of each FF, process engineering technology and cell library. In general, FF's state transformations rely on data processing in digital systems. Evaluation of the efficiency of clock rated data therefore involves detailed simulations and comparative analysis of the operation of the FFs.

By using the clock gating strategy, dynamic power consumption may be minimised. Clock gate regulated by this data indicate that the clock signal can be switched on. Via the gate signs, flip flops and latches must also be permitted. The X-OR outputs are OR ed for combined output gate signals from the flip flops and latched to prevent any flaws in the required modules.

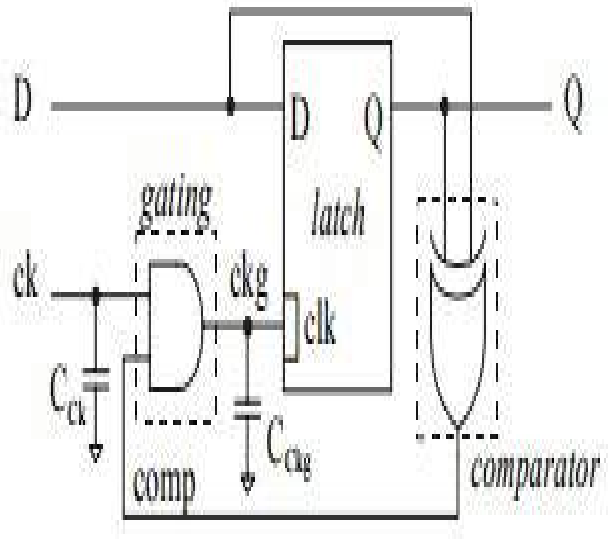


Fig 3.3 Clock Gating using for single Flip-flop.

Figure 3.3 shows the scheme of a gated latch. The lock is positive (it is clear if ckg=1 and holds ckg=0). A XOR gate compares D to Q while the logic of gating is a plain AND gate. The circuit is controlled accordingly. If ck is 0, ckg is 0 and the lock is in holding condition correctly. In the other hand, the gating logic allows the CKG signal to turn correctly when ck is high and D is different from Q. Notice that the gating logic prevents the spread of switching activities from ck to ckg if D is equal to Q.

Clock gates should be used to reduce the current control of the device. The signals of the Clock are multiplied by an AND gate to a predefined signal. With clock-gating. However, this gating timepiece leaves a lot of redundant clock pulses. While the design efficiency is significantly increased, such instruments require the use of an automated chain of synthesis algorithms from the registry transfer level (RTL) to the gate level and net list. Unfortunately, such an automated process contributes to several needless clock switching and, as seen in this project in many industrial scenarios, thus the the amount of waste clock pulses on flip-flops (FFs). It is also beneficial to create automated and efficient processes to reduce this inefficiency. In the following we interchange the expressions swapping, switching and operation.

The internal MBFF drivers may be enhanced for some extra power in order to remedy the problem. The MBFF at RTL design level is therefore advised to prevent the time closing hazards due in the back end of the design phase to the implementation of the MBFF. The chrome power

savings still outweigh the short-circuit power penalties of data togling because FFs' average clock-to-clock ratio is very small and generally rates from 0.01 to 0.1. Logical, structural and FF considerations can motivate an MBFF classification. During a detailed investigation into FF's grouping on the structure, the front-final effects of the MBFF group size and the effect it has on clock gating (CG). Two questions are addressed in this brief. The first of these is what can be the best bit of multiplicity of clock-driven (DDCG) MBFFs. Secondly, how to optimise power consumption depending on the toggling ratio between the data and the clock (also termed activity and data toggling probability).

DISADVANTAGES:

- High energy consumption

CHAPTER 4

PROPOSED SYSTEM

The Multi-bit Flip-Flop system removes the cumulative amount of the inverter by exchanging the flip flop inverters. The clock gating guided data reduces the pulse of redundant clock. The additional power saving would be increased by combining multi-bit flip-flop with data driven clock gates. This proposed work is used for Xilinx tool. This project studies clock gating, which is the most offensive possible and is used by FFs at gate level. If the FF status does not adjust in the next clock cycle, the clock signal drived by the FF is disabled (gated). Data powered gating causes overheads in area and control that must be taken into account. It is suggested to group multiple FFs to be controlled by the same clock signal, provided by the activating signals of the individual FFs, to try to minimise overhead. However, this may reduce the disabling performance. It is also advantageous to group FFs with strongly correlated switching operations and a combined activating signal. A model for data gating based on the toggling operation of component FFs is built in a recent project.

As the number of transistors per application increases, power minimization is becoming more critical in modern SOC designs. High power dissipation of a SOC increases the device costs as well as product life and reliability. reducing energy usage in electrical and physical architecture such as generating multi-voltage designs while avoiding pacing issues using programmable voltage regulators A methodology and design technique of electrical and physical design power optimization to build the IS with the microprocessor ARM 1136JF-S was presented in 90nm CMOS standard. Improvements to the design techniques and methodology for multiple supply voltage operations, the optimization of leaking current and clock rate, one-pass RTL synthesis, VDD selection, torque optimization and electrical closure are defined in a multi-VDD design. Dynamics were reduced by 40 percent and leakage dissipation was decreased by 46 percent while retaining a normal working clock rate of 355-MHz. The first silicone has fulfilled functional and electrical design criteria, As the technology is scaling down, power discharge is rapidly becoming one of the most critical limits to the nanometer IC architecture. However, power and timing frequently clash with optimization targets. We give a new absolute flow of optimization of electricity under output limit in this article. We integrate them along with slack delivery management to optimise the power reduction efficiency instead of using placing, door

scaling and multiple-Vt assignment methods separately. We suggest the use of linear programming (LP) and GP formulation to increase the slack distribution, which helps to optimise the overall power reduction during the Vt assignment stage. Our formulations provide important, frequently overlooked practical design restrictions such as slew, noise and short-circuit power. We tested our algorithm on a series of manually optimised industrial strength circuits from a 65nm microprocessor with several GHz and got very promising performance. To our best understanding, it is the first work that integrates positioning, gate sizing and Vt swapping systematically for complete power (and in particular leakage) control. Clock networks minimization Server workload positioning has historically been motivated primarily by success goals. In this study, we study the architecture, deployment and evaluation in a heterogeneous, virtualized server clusters of the power-aware application placement controller.

In addition to the performance advantage of the data containers placed on the physical servers, the positioning element of the application management middleware takes care of the cost of energy and migration. This analysis has a twofold contribution: first, we have some means of capturing the issue of the cost-conscious application positioning that can be applied to different environments. Details of the kind and the model assumptions and practicality of the presuppositions on actual servers are given for each formulation. In the second part of our analysis we present the design and positioning algorithms of pMapper to solve the problem in realistic ways: to minimise power under the constraint of fixed output. We introduce an automatic recording technology to synthesise low-power clock trees for low-power ICs. On 7 industrial designs, the clock-tree capacity was decreased by 19.0% and 14.9% respectively by (1) commercial base and (2) electricity-aware positioning technology, and by 1.8% and 1.5% on average of WNS, respectively, by 15.3% and 5.2%.

There are two limiting factors for the advancement of VLSI technology: strength and variance. Minimizing the size of the clock will result in lower power consumption, less noise in the power supply, a lower clock buffer size and less risk of change. Previous work on clock network reduction focuses mostly on clock routing and changes are also reduced by location in the Input Register. We propose to browse the cell location registrys in order to further reduce the scale of the network clock. We propose the following strategies in a quadratic placement system in order to resolve the discrepancy between clock network minimisation and standard placement

goals: (1) instructions for registry registries based on ring in Manhattan; (2) a centre of gravity restrictions for registries; (3) pseudo-pin and net;

These strategies work both for null and recommended skew designs in wired and time-driven positioning. Experimental evidence shows that our system will cut net clock wirelength by 16% -33% with no higher signal net wirelength than standard methods or multibit registers by as much as 0.5%.

We present an automated record placement method which allows low-power clock trees for low-power ICs to be synthesised. Combining 1-bit flip-flops into multi-bit flip-flops is one of the most efficient technologies for reducing clock power. The barriers to the fusion of multi-bit flip flops are (1) the limit on input and output times on each flip flop; (2) the restriction of the area on any divided bin in the positioning stage. One of the most powerful methodologies for both saving chip areas and power use is the use of multi-bit flip-flops or multi-bit registers[6], or registering banks[4].

The ideal fan out of a clock gater that provides maximum power savings is dependent upon average toggling statistics from the various FFs, process technologies and used cell library. The state transformations of FFs in digital systems are generally dependent on the data processed. Thereby comprehensive simulations and statistical analyses of the activities of the FF are essential to evaluate the effectively data-driven clock gating. Another grouping of FFs called multi-bit FF for clock change reduction (MBFF).

In order for the inverters that drive the clock pulse in their master and the slave latches, MBFF tries to mergate FFs into a single cell physically. The MBFF classification is mostly driven by the physical location of each FF, while data-driven clock gatting can incorporate toggling similarities with considerations of physical position. This project examines the group size to optimise savings in power, 1) the FFs are to be positioned in a group for maximising power reduction, and 2) how to extract these groups algorithmically. We also define the execution of a backend design flow.

Introducing clock-gating into MBFF

The discussed MBFFs have been powered by an un-going clock signal free-running. The DDCG incorporated in the k-MBFF is seen in Fig. 4.1. Indoor library cells are all shaded circuits. the group size to k, which achieves maximum creativity energy saving solves the equation, given operation p, was shown in [2].

$$(1-p)^k \ln(1-p) C_{FF} + \frac{C_{latch}}{k^2} = 0, \quad \ldots\ldots\ldots\ldots\ldots\ldots(1)$$

Where the clock input loads of the FF and a latch are the CFF and Clatch. Table 2 for standard CFF and Clatch shows the solution of (4) for different operations.

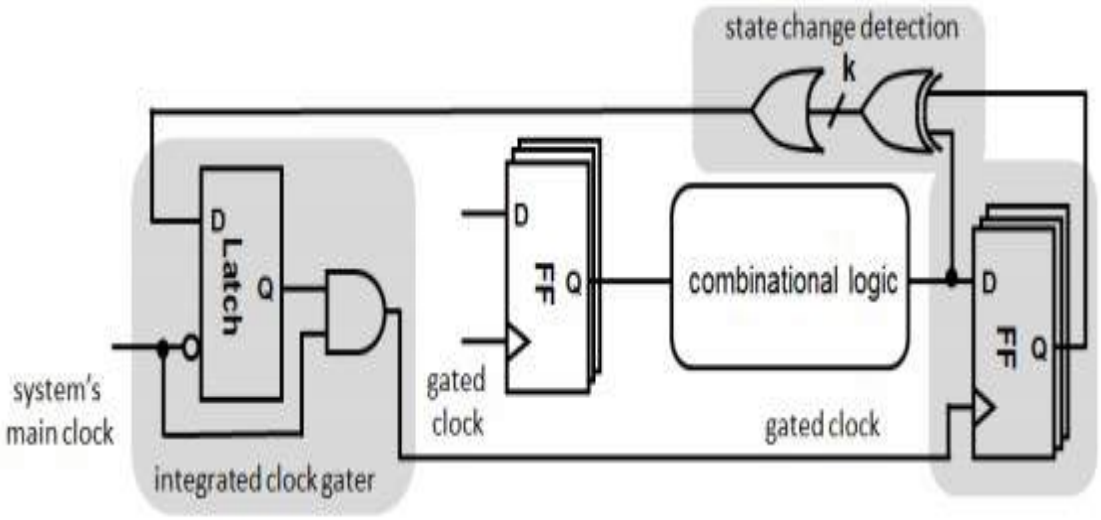


Fig 4.1: DDCG integrated into k -MBFF.

p	0.01	0.02	0.05	0.1
k	8	6	4	3

Table2: Dependence on the toggling likelihood of the optimum MBFF diversity.

The MBFFs addressed in the sequel are DDCG unless otherwise specified. Fig. 4.1 was tested with SPICE for various kinds of p and k and DG architectures to gain an understanding of the various ways in which the DDC can improve power consumption In Fig. 4, the 2-MB power reduction is shown FF2's bits display the amount of energy used by each line (Fig. 2 illustrates).

Due to the on-and-off disabling of the clock motor, 0 operations require power usage irrespective of operation. Line (b) is the optimal case where both FFs swing at the same time. In this case the two FFs share the clock driver either toggle for each of them, or the internal gater seen in Fig. 4.1 disabled them. The electricity consumed for zero operation is expected to be about half that of two 1-bit FFs. If the action increases, the power of (b), as power absorbs proportionately to the activity, grows more rapidly than (a).

A 2-MBFF is useless after the crossing point of operation 0.17, when power begins to be wasted. For the same time

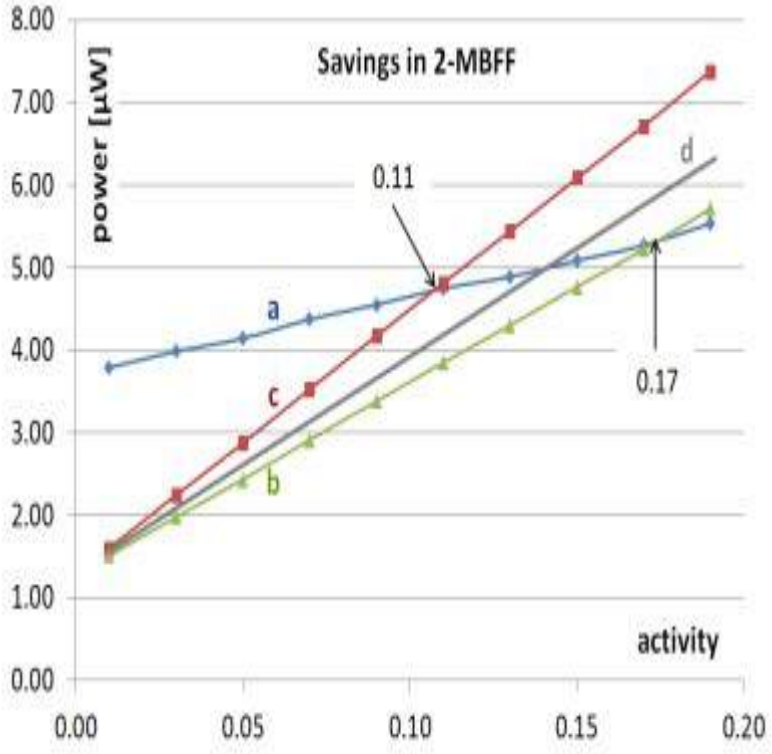


Fig 4.2: Power consumption of 2 FFs vs. 2-MBFF.

Line (c) indicates the disjointing case of flipflops. This is the worst case, of course, because the clock driver functions on both FFs, and only one wants it. Concerning (b) the use of 2-MBFF would not make sense if FFs are greater than 0.11 for disjoint toggling. The power savings in the range from line b) or c) to line 2-MBFF are based on operation (a). Note that the every bit power savings is = (3.8- 1.8)/2=1.0µw for zero activities. Please note:

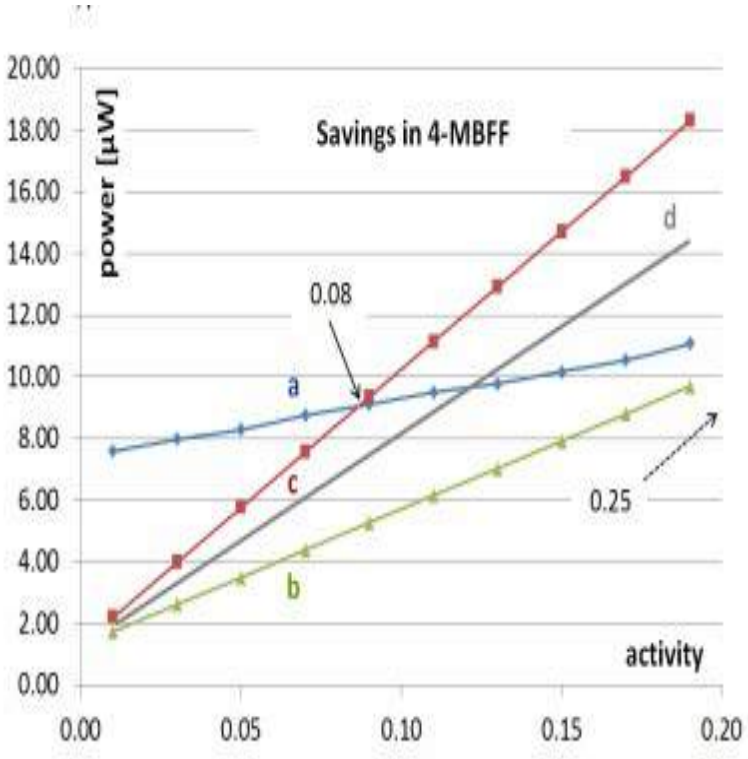


Fig 4.3: Energy consumption of 4 FFs vs. 4-MBFF.

Fig. 4.3 showed the energy consumption of 4-MBFF, which is the better case of the synthesis toggling of 4-MBFF FFs in line (a), four 1-Bit FFs, powered separately from everything else and line (b) the worse case for a disjointed toggling in line (c). The per-bit energy savings per null operation are: l'alloy l'alloy' (dir) l'alloy's length is µ) 7.4 2.2 4 1.3 W. Remember, though, that 4-MBFF does not have to save at 0.08 operation until 0.09 in 2- MBFF in the worst case of disjoint toggling. Anyway 4-MBFF is preferred over 2-MBFF in the best case of simultaneous toggling. Similar findings can be seen in Figure 4.4 for the 8-MBFF. Per-bit power savings are . (15.3-2.5)/ 8=1.6 µW. The power savings are not available

8-MBFF savings interrupt 9 under the worst scenario of disjoint toggling, at 0.06 operation, and in the best case of concurrent toggling at 0.40.

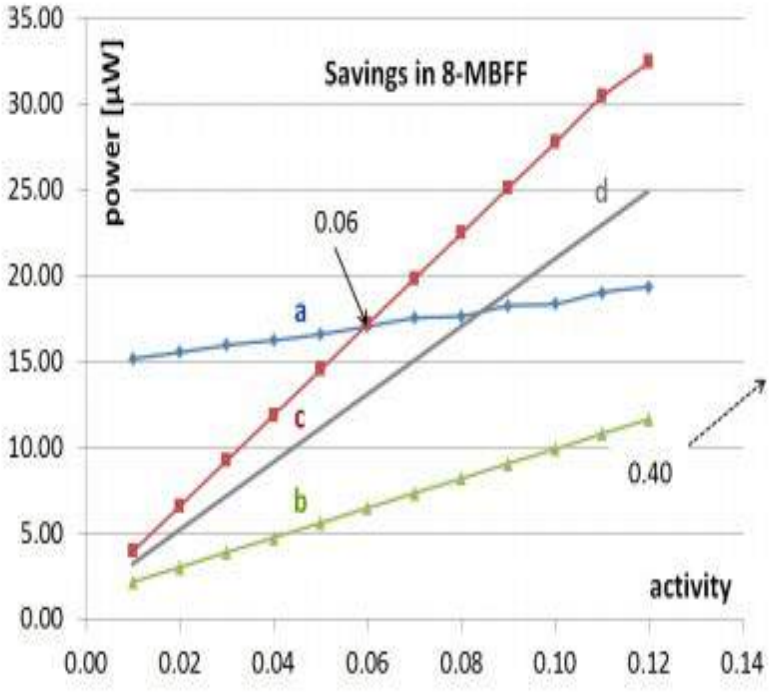


Fig 4.4: Power requirement of 8 FFs vs. 8-MBFF.

Filp- Flops that grouped in a DDCG MBFF.

The k -MBFF predicted power savings — by Section 2— is quantified by k -E p k on the premise that the clock is rolling independent and free-running. Section 3 demonstrated how tobacco correlation influences the risk of breakdowns when an MBFF ceases energy saving. Clearly, for flip-flops whose toggling is almost entirely associated, the strongest set of FFs could be obtained. FFs have shown themselves to be a dilemma of grouping the FFs that yields maximum rebounds, and thus maximum power savings. The practical approach has been introduced with an almost maximum power saving [10]. Its inconvenience is the need for early awareness of the Value Change Dump (VCD) vectors, drawn from numerous modeling of control, which reflect standard process and design applications. In the early design stage, such data does not exist.

More typical is the minimum toggling bulk chance of any flip-flop in the model, which is used to derive an optimally toggling grouping of FFs that is based on probability. The study hitherto presumed that all MBFF groups had the same chance of toggling data p. The toggling chances of FFs are generally different and so the relevant question is how the chances change the classification of FFs. In the past works the structural FFs were either grouped into registers (e.g.,

subsequent bits) or the spatial proximity grouped into post layout. We then demonstrate that data retrieval is important for optimising energy savings and should be considered.

If n FFs 1 is considered, FFs 2-MBFFs would be considered. Let a 2-MBFF, as described in the above description, comprise FFi and FFj, each independently of a likelihood i p and j p and p. If neither of these clock is switching over, the clock is disabled for the entire (FFi j and no dynamic energy is used in the inner ten clock driver. The 10 clock is triggered when FFi and FFj are both toggling), and the energy of the clock driver is entirely utilizable and there is no waste. When an FF is switched on, a waste occurs, but not its equivalent. It can drive both FFs, but only one is currently in use. When the other half of the timekeeper power supply has failed, time will have stopped (see: Power interruption) half the internal power will be futile

$$W_{(i,j)} = \frac{\lambda_2}{2}\left[p_j\left(1-p_i\right) + p_i\left(1-p_j\right)\right] = \frac{\lambda_2}{2}\left(p_i + p_j - 2p_ip_j\right).$$

……(5)

Given FFi , FFj , FFk and FFl Since two 2-MBFFs files is a lot of memory, you end up using a lot of wasted resources

$$W_{(i,j)} + W_{(k,l)} = \frac{\lambda_2}{2}\left[\underbrace{p_i + p_j + p_k + p_l}_{(a)} - 2\underbrace{\left(p_ip_j + p_kp_l\right)}_{(b)}\right].$$

If 'a' 9 is paired, so the 'b' depends. when I and (a)and (W is restricted)at the same time when (k is held constant). If ,≤≤≤i j k l p pp

the pairing $\left\{FF_{(i,j)}, FF_{(k,l)}\right\}$ is favored over $\left\{FF_{(i,k)}, FF_{(j,l)}\right\}$ since $\left(W_{(i,j)} + W_{(k,l)}\right) - \left(W_{(i,k)} + W_{(j,l)}\right) = -\lambda_2\left(p_i - p_l\right)\left(p_j - p_k\right)/2 \le 0$. $\left\{FF_{(i,j)}, FF_{(k,l)}\right\}$ is similarly favored over $\left\{FF_{(i,l)}, FF_{(j,k)}\right\}$. The generalization for pairing of n FFs is straight forward. Let

$n/2$

The generalization for pairing of n FFs is straight forward. Let n be even and })({ 2 , 1 : FF n =s t i ii P be a pairing of FF ,FF , ,FF 1 2 n in n 2 2-MBFFs. The following energy waste)(W P results in

$$W(P)=\sum_{i=1}^{n/2} W_{(s_i,t_i)}=\frac{\lambda_2}{2}\left[\sum_{j=1}^{n} p_j-2\sum_{i=1}^{n/2} p_{s_i}p_{t_i}\right]. \qquad \text{……..(10)}$$

Since 1 n j j=∑p is independent of the couple,) (W P is reduced by maximizing 2 1 n =∑s t i ii p p The following theorem defines the optimum combination <to W P>. The optimum combination) [8].

Theorem 1. Their chances of toggling must be even and FF, FF, FF 1 2 n be ordered to satisfy 1 2 n ≤≤≤p pp. pairing })({ 2 2 1,2 1 : FF n i ii–= P (10). For grouping in k-MBFFs, the above finding is generalised as follows. 11

Theorem 2. The divisible should be n by k and the order FF, FF 1, FF 1, 2 n should be ordered to be toggled.

probabilities satisfy 1 2 n ≤≤≤p pp.

$$P:\ \left\{FF_{(k(i-1)+1,\ldots,ki)}\right\}_{i=1}^{n/k}$$

The composition of successive FFs minimises nk k-MBFF energy waste. In addition, [8] presented where n is not dividable by k.

In the template flow, capture everything together.

It was because of detailed simulations that it is possible to gain knowledge of the toggling capabilities of all FV cards in the most advantageous grouping. Such data are not always available and we presume that the model used is based on the assumption that the FFs are autonomous. The relationship between power savings and FF operation p's and MBFF's multiplicity k, which grouping the power savings in monotonous order p. Consequently, We consider process p and the number of pieces in the design flow when looking for a way to reduce energy consumption. For this, the power-saving figures of the two-MBFF, four-MBFF and eight-MBFF are illustrated in Figure 1.1, 1.2, and 1.3.

The distinction between FFs swinging at the same time but in opposite directions comes close to reflecting the function that all methods can or not do something at all. The decision

under which MBFF is to be clustered would follow intermediate lines, knowing the operation of the FF. Figure 4.5 shows Figure 1(1, 1.2 and 1.3), which is separated by their respective multiplicity, Usual energy consumption scale per-bit.

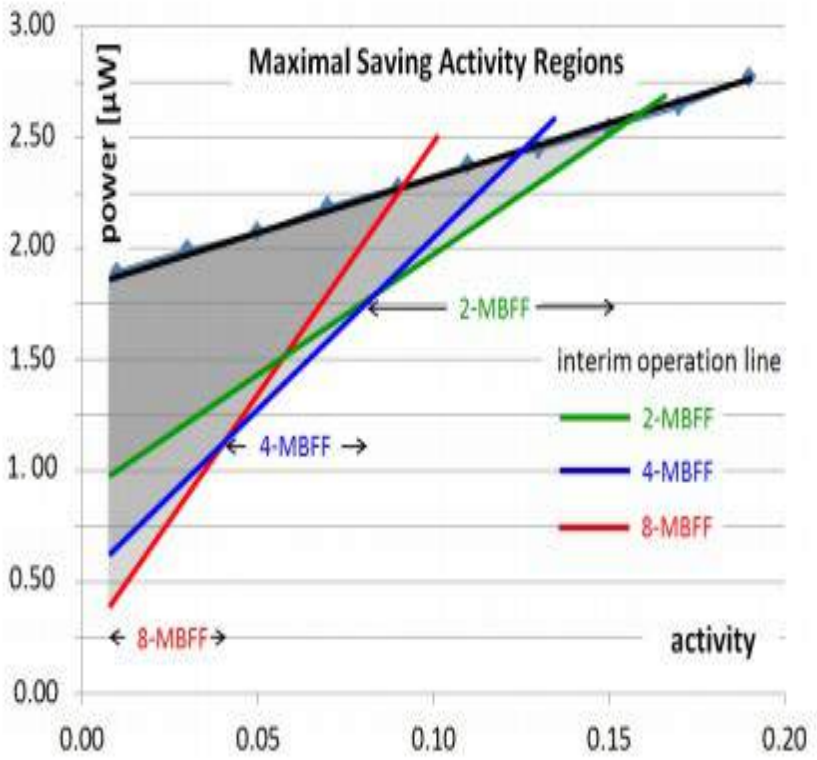


Fig. 4.5. The division into maximum savings ranges

Fig. 4.5 illustrates how the FF operations are divided into regions to maximise energy savings. The shaded area follows the power consumption of a 1-bit ungated FF. By grouping FF into 2 MBFF, 4MBFF, and 8MBFF, triangular areas among black lines, green, blue, and red paths per bit reflect however much power is saved. It exposes the very poor activity paid for by the community FFs in 8-MBFF. There would be a position of exchange where 4 MBFF costs more with growing operations. Two MBFF pay better at a higher stage, until power consumption stops. This behaviour is used in the following MBFF

1. Sort the n FFs such that 1 2 n .≤≤≤p pp
2. Set 1.=i
3. optimize k by i p , based on Fig. 4.5.
4. Group 1 1–++FF ,FF , ,FF i ii k in a k -MBFF.
5. Set .+←i i k

6. If >i n stop. The other way go to 3.

Flip-Flop Group Placement Upon receiving the IS of TSFGs, an appropriate MBFF position for each TSFG should be determined for both positioning density and wirelength considerations. 1) Placement density consideration: The placement bins protected by the inclined rectangular placement region should be obtained before seeking a legal position for MBFF corresponding to a TSFG within the rectangular placement zone. The bins which cross each limit of the inclined rectangular positioning are classified first Thus can be identified and collected correctly, the containers enclosed by these intersected containers. To handle an MBFF that is corresponding to a TSFG, the bin with the lowest placement density is

If the bin with the second lowest placement density doesn't have a correct placement grid, so it will be selected. Unless a correct positioning grid is discovered, the grid-search procedure is repeated. 2) Considering the interconnecting wire length: In addition to the consideration of positioning density, it is often very important to reduce the interconnecting wire length when positioning an MBFF that matches a TSFG. The area bounded by the median coordinates of all the pins attached to the MBFF is first considered for a location for the MBFF with a shorter wire length. The median co-ordinates in the 8 pins are in the directions xp4, xp5, yp4 and yp8 When there is no valid positioning grid in a bin, which is intersected by the pin co-order field, and the tilt rectangular positioning region, it is expanded to the next pitch, which is the nearest of the current pitches, by the pins' co-ordinates zone. Compared to the other adjacent pitches, yp1 is the nearest pitch of yp8. Then xp4-, xp5-, yp4-, and yp1 will surround the expanded field. The procedure continues until the MBFF is defined with a correct placement grid.

The positioning of both flip flops were also optimised on each circuit. Table II lists the bench-marking circuit names ("Circuit"), 1-bit flip-flop numbers ("# of 1-bit flip-flops"), 2-bit flip-flop numbers ("# of 2-bit flip-flows") and 4-bit flip-flop numbers ("# of 4-bit flip-flops"). There are also requirements for their power consumption and areas for a cell library featuring one, two and four-bit flip flops. The Table III lists the flip-flop bit numbers ('Bit # Flip-Flop') and the energy consumption ('Power') and the area corresponding ('Area'). For three different methods we compared the number of 1, 2 or 4 bits, power reduction, HPWL ratio and runtime of flip flops: (1) the proposed solution without using progressive window optimisation (2) the suggested

approach focused exclusively on progression window optimisation, and (3) a proposed approach based on a progressive window optimisation, each of which takes account of the positioning density and the interconnecting wire length. This approach is based on progressive window-based optimisation. Circulating benchmark ("Circuit"), number of 1, 2 and 4-bit flip-flops ("# FF(1), 2, 4-bit)"), reduced power ("Power Red.") and HPWL ratio between the outcome and the input ("HPWL ratio") of the three methods and the run times ("Time"). The findings reveal that Approach (2) and (3) exceeds Approach (1) by at least 37222X and is significantly improved by incremental window optimisation. The runtime based on the Approach 3 is only 79 seconds even for the largest circuit which holds hundreds of thousands of flip flops. Approach (2) is 7% better, while the HPWL ratio is 21% less than approach (2). (3). The suggested solution therefore is very effective and effective on the basis of the incremental window-based optimization taking into account both the positioning density and the interconnecting wire length, which is able to combine current MBFFs incrementally to achieve further power conservation.

There are some realistic remarks. Apart from toggling chances, the MBFF community can also take into account rational relationships and physical location and road restrictions. One example is the microprocessor pipeline registers. Mixing parts of various pipeline phases makes little sense. It is clear and obvious that bits from the same register are similar to each other in the position and track tool, whereas FFs clusters of registers from different pipeline levels are located separately. Therefore, the MBFF does not combine ffs from various pipeline registers, although they might be desired to be grouped from the point of vue of toggling chance. For other device buses and registers such as data storage, adresses, counters, and so on, similar claims are supported. The Finite State Machines (FSMs), whose MBFF grouping does not need to cross regulate logical limits, is another example. Another example. Another example:

The proposal algorithm can be paired with grouping methods even though it is designed for RTL or Gate architecture stages. Original positioning as a "dry run" takes place in order to gain guidelines for initial FF proximity. The toggling probability-based algorithm will then take into account those that direct the MBFF classification. Instead of tearing up the old FFs, the later true position and route will use the library cells of MBFF and attach MBFF substitutes, a repetitive and non-trivial layout operation, saved by our style flow.

The proposed concept flow from MBFF was implemented with the 65nm process technology of TSMC in a 32-bit pipeline MIPS processor. The two programmes shown in Table

3 were used for the workload. The average FF operation in the pipeline register shall be shown in blue is generated in the pipelines. Notice the decline in operation as the pipeline phase of instruction fetch (IF) is progressed (WB).

There are two types of MBFF classification. In the first, FFs are sequentially grouped according to their bit number. Section 4 showed that FFs are optimally clustered by second method 13 in the increased order of their operations, while FFs are expected to knock separate and distinct from each other Both classifications maintain a careful balance with the constrains that clock domain borders are not crossed and FFs from unrelated logical organisations are not mixed. The average operation is shown in table 3, for each k -MBFF, 2, 4, 8. In most cases it is recommended to group by monotonic operation (green), although it has exacerbated in some cases (colored in red). The classification is oblivious to rotating correlation. That will happen.

pipeline stage activity for array sorting program												
grouping method	IF / ID			ID / EXE			EXE / MEM			MEM / WB		
	0.105			0.0856			0.0711			0.0473		
	2-MBFF	4-MBFF	8-MBFF	2-MBFF	4-MBFF	8-MBFF	2-MBFF	4-MBFF	8-MBFF	2-MBFF	4-MBFF	8-MBFF
by index	0.174	0.261	0.353	0.140	0.204	0.275	0.109	0.163	0.214	0.0793	0.117	0.173
by activity	0.169	0.261	0.353	0.134	0.189	0.231	0.116	0.155	0.190	0.0761	0.104	0.131
improve [%]	+2.9	0	0	+4.3	+7.4	+16.0	-6.4	+4.9	+11.2	+4.0	+11.1	+24.3
pipeline stage activity for array matrix multiplication												
grouping method	IF / ID			ID / EXE			EXE / MEM			MEM / WB		
	0.127			0.118			0.0799			0.0582		
	2-MBFF	4-MBFF	8-MBFF	2-MBFF	4-MBFF	8-MBFF	2-MBFF	4-MBFF	8-MBFF	2-MBFF	4-MBFF	8-MBFF
by index	0.203	0.311	0.422	0.169	0.241	0.300	0.128	0.180	0.262	0.0940	0.145	0.208
by activity	0.198	0.294	0.388	0.174	0.246	0.322	0.128	0.174	0.222	0.0938	0.127	0.162
improve [%]	+2.5	+5.5	+8.1	-3.0	-2.1	-7.3	0	+3.3	+15.3	+0.2	12.4	+22.1

Table 3. Pipeline records of 32-bit FF operations. Table 3.

The registers of pipelines were then introduced using MBFFs clustered into a single order. The Grouping begins with 8-MBFF for low tasks, as shown in Fig. 4.5, and then

progresses to 4-MBFF and 2-MBFF for FFs the large bulk of the FFs are fully solo and unnoticed Yes, of course this should be done so as to focus on the feedforward system only. Table 4 illustrates the energy savings for the kind and matrix multiplication weighted working load obtained at each pipeline register. The findings have been calculated in the simulation of the MIPS in 1.1V and 200MHz by SpyGlass[11]. Savings have been reached at 34.6 percent. Complete power decrease (CGHW overhead included) was 23 percent for the pipeline registers consuming 65 percent of the whole MIPS output (not including the memory).

	IF/ID	ID/EXE	EXE/MEM	MEM/WB	total
power [µW]	980	1056	952	916	3904
savings [µW]	284.4	344.0	332.0	388.4	1348
savings [%]	29.1	32.6	34.8	42.4	34.6

Table 4 contains data about power savings for 32-enhanced MIPS.

We also show the effect of the power savings that have been achieved by grouping a full-sized industrial processor for a 28nm technology node in MHz. A to G in Table 6. Unfortunately, it utilises a 6.2 watts, which is 45% for the 3D vSync renderer.The original design consists of unloaded MBFFs, so the power reduction can only be attributable to the inclusion of the clock gate in Fig. 5, in addition to savings made by fewer drivers from the unloadable MBFFs in the original design. In addition, the original design contains detailed logical signals enabled by the clock, as well as manual insertions. The FF's practises were first profiled and later classified. Table 5 has an average net total power saving of 8%, consisting of dynamic and static/hand drawn components and all CGs. Table 5 reveals The power saved of 8 percent was done with an additional 9 percent saving, resulting in a net saving of 17 percent, from 1-bit FFs to non-gated mbffs. The industrial design group VLSI greatly appreciates these savings. With the implementation of a clockgating circuit the field penalty was 2.3%.

unit	FF CLK power [mW]	total CLK power [mW]	total unit power [mW]	FF CLK power save [mW]	FF CLK power save [%]	total CLK power save [%]	total unit power save [%]	area penalty [%]
A	80	1112	1,802	44	57.6	4.09	2.52	1.7
B	304	316	1,638	104	33.4	32.5	6.22	2.8
C	184	268	760	76	41.9	28.6	10.1	2.7
D	72	172	294	32	45.2	19.2	11.2	2.3
E	162	368	884	88	53.4	23.8	9.90	4.3
F	112	204	252	80	69.7	38.2	31.0	1.3
G	124	368	556	72	57.4	19.7	13.0	1.9
total	1,040	2,804	6,186	496	47.5	17.7	8.00	2.3

Table 5. Power savings in an 40nm network processor.

The overheads of the latch and gat (AND gate) are amortised over k FF.

Let the average FF (also known as the activity factor) to be indicated by p (0 <p<1). The worst case assumption of independent toggling FF is that the number k of jointly regulated FFs for which power savings are optimised is shown in[9], on the assumption that a standardized physical clock tree structure is used.

$$(1-p)^k \ln(1-p) C_{FF} + \frac{C_{latch}}{k^2} = 0,$$

Where cFF is the input capacitance of the FFs clock, cW are unit size wire and clatch is the capability of the latch including the wire capacitance of the clck input. Table I shows how p depends on the ideal k. This kind of timing scheme has significant consequences, which are

explored in [9]. When we address data-driven gating as part of a full design flow, we shall return to such.

4.2 Implementation and Integration in a Design Flow.

We illustrate below how data-driven clock gates are implemented as part of a typical backend architecture flow. The following procedures are included.

(1) Estimating the probabilities of FFs is the use of a comprehensive test bench that represents the system's standard operating modes to calculate the FF category size k by solving (1).

2) Run the positioning method in hand to obtain the preferred preliminary FF positions in the layout.

3) The FFs grouping tool used is used, in accordance with the toggling correlation data collected at the step 1 and the FF positions, to apply the models and algorithms present in SectionsII and IV. 3) This move will produce k-sized FF sets, where the ff's are clocked together with a common gater (with manual overrides if necessary).

4) To enter into the hardware definition the data-driven clock-gating logic (we use Verilog HDL). A programme tool automatically adds sufficient Verilog code to execute the logic outlined in the figure. 2. The FFs are connected by the grouping achieved in step 3. A sensitive practical concern is whether the gating logic should be used in the RTL or the gate level definition. This depends on the technique of architecture being used and its topic goes beyond the reach of this document. In the RTL definition, we have included the gating logic.

5) Run a Step 1 test bench to check the complete identities of the outputs of FFs before and après the gating logic implementation. Although datadriven gating does not alter the logic of signals in its own definition, and thus FF toggling should remain similar, it must be implemented by a robuste architecture flow.

6) Completion of regular backend flow. The backend architecture flow continues from this stage by the use of ordinary location and path methods.

Double Edge Triggered Flip-Flop.

A D-flip (D-FF)[1] edge-triggered flop has a clock input and a data entry (D). The output Q takes the D value and retains the value to the next positive C-signal immediately after the C signal moves from 0 to 1. A FF like this is said to occur at the clock pulse's leading or positive rim. Some FF's are intended to activate the C-signals on the trailing, or on the negative. JK-FFs

that react to the J and K signal after the C-transition are also activated at the end (again they may be of either the positive or negative edge sensing type). If j = 1 (depending on k) and changes 1-0 if k = 1 the JK-FF output changes from 0 to I. (independent of J).

The benefit of triggering the edge is that the inputs of regulation (D or J and K) can be modified in the area of a triggering edge of the signal C at any moment. It also decreases noise response.

When in a synchronous device the minimum time between successive shifts in the edge enabled FF state is L, then the frequency of the pulse clock shall be at least 1/L. One of the two C-signal transformations does nothing during each clock pulse cycle, although it will adjust some of the logic elements' external outputs. Such operation is unwanted since it leads to a greater dissipation of energy by almost all technologies currently used for the implementation of logical circuits. (The C-MOS logic basically has no dissipated power until a transformation takes place.) The clock pulse Generator works at the half-frequency at the same data rate if FF is used on both sides of C-pulses. This alone would minimise the cost and power dissipation of the clock pulse and clock pulses and would also remove significant differences in the states at the outputs of different doors. It will also be expected to a degree increase data rates.

Various prototypes for dual-edge induced (DET) D-FFs and DET JK-FFs are available here. The simplest logical complexity designs involve retarding elements that limit the permissible running speed. In the other architectures, approximately 50-100% more complex than the corresponding single-edge circuits, no time components are required to achieve optimum running speeds. Only basic operations are carried out; no fixed or specific operations are introduced and no production supplements are made. It would not be hard to design these features. Convenient implementations, instead of the AND-OR-INVERTOR logic seen here are in most cases often to use elements such as the NAND and NOR gateways or transfer transistor networks.

DET FF's architecture is a strong use of asynchronous sequential circuit theory and the use of decomposition techniques may be of particular concern..

FLOW TABLE DESCRIPTIONS OF DET-D-FF's:

Table 6 represents the primitive DET-D-FF flow table. Note that concurrent D and C modifications are processed as if one of these variables has first modified, but it doesn't matter which variations were first made. This is a rational assertion that in any pair of events we can

never depend on perfect simultaneity. In the design phase, the possibility of addressing a concurrent transition in any direction is left available for further use.

It is not difficult to demonstrate that there are two minimal rows of this table with the well-known procedure, as shown in Table 11(A) and (B). Table I rows occupied by the reduced-tables row are shown in the parentheses sets of numbers on the right of individual rows (A) and (B)

	CD 00	01	11	10
1	(1),0	2,0	4,0/5,-	3,0
2	1,0	(2),0	5,-	3,0/6,-
3	1,0	2,0/7,-	4,0	(3),0
4	1,0/8,-	7,1	(4),0	3,0
5	1,-/8,1	7,1	(5),1	6,1
6	1,0	2,-/7,1	5,1	(6),1
7	8,1	(7),1	5,1	3,-/6,1
8	(8),1	7,1	4,-/5,1	3,-

TABLE 6 PRIMITIVE FLOW TABLE FOR DET-D-FF

	CD 00	01	11	10	
1	(1),0	(1),0	4,-	2,0	(12)
2	1,0	3,-	(2),0	(2),0	(34)
3	(3),1	(3),1	4,1	2,-	(78)
4	1,-	3,1	(4),1	(4),1	(56)

(A)

	CD 00	01	11	10	
1	(1),0	2,0	4,0	(1),0	(13)
2	1,0	(2),0	3,1	(2),1	(26)
3	4,1	(3),1	(3),1	2,1	(57)
4	(4),1	3,1	(4),0	1,0	(48)

(B)

Table 7 Minimal Table Covers

In relation to single-input shift (SIC), Tables 7(A) and (B) are equal to each other. But the answers to various input shifts, a variation that is not realistic but leads to very different

implementations, are defined differently. There are no significant dangers, but (A) has d-transitions (for example row 1 with CD = 01, if C changes), whereas B has zero. There are no significant risks.

CHAPTER 5

HARDWARE REQUIREMENTS

GENERAL

Integrated circuit (IC) technology enables a wide range of groundbreaking technologies and applications that have transformed our way of life. The Nobel Prize in Physics 2000 was awarded to Jack Kilby and Robert Noyce for their invention of the integrated circuit; neither transistors nor computers are as significant without the integrated circuit as they are now. VLSI devices are much smaller than the individual modules used by electronic systems in the early 1960s and use less energy.

Integration enables one to develop devices with far more transistors to solve a problem with far more computer power. Integrated circuits can be designed and manufactured even more easily and are more durable than standalone devices. These systems allow special use systems to be developed more efficiently than general-use computers.

5.1 APPLICATIONS OF VLSI

Electronic devices are also carrying out a wide range of everyday activities. In some instances electrical systems have replaced electric, hydraulic or other systems that typically have smaller, more flexible and easy to use electronics. In other ways, new applications have been developed by electronic devices.

- People are building and purchasing complex and power-hungry personal entertainment devices, including mobile MP3 players and DVD players, with remarkable levels of inefficiency.
- Electronic devices are installed in stereos and displays, monitor the systems and vary the height of fuel injection. Electronic systems perform many tasks, some more visible, others hidden:
- Digital electronics compress and decompress video in consumer electronics, including at high-definition data speeds.
- Despite their dedicated purpose, low-cost web browsing terminals continue to demand sophisticated electronics.

- Full word processing, financial reporting and games are included on personal computers and stations. Computers both have central processing units (CPUs) and disc access, fast screen monitor, and special hardware.

Medical electric systems calculate body functions to prevent irregular symptoms by performing dynamic computing algorithms. The availability of such complex systems does not generate demand for even complex systems, far from overwhelming customers. The increased complexity of applications increasingly leads to new stages of complexity in the design and manufacture of the integrated circuits and computer devices.

And maybe the most astonishing feature of this system set is its diversity as systems get ever more sophisticated, because we are not building just any common computers but an increasing number of specialist systems. The ability to achieve this reflects our rising understanding of integrated circuit manufacture and construction, but increased market demand continues to challenge design and output boundaries.

5.2 VLSI Advances:

- Whereas we concentrate on integrated circuits in this book, the architecture of the system is essentially determined by what we can and cannot easily incorporate into an integrated circuit. In numerous ways, integrated circuits are critical for improving system features. ICs have three main advantages over optical discrete circuits:
- **Size:** Electronic components are that much thinner, with all wires and transistors tightening to measurements of micrometres compared with the scales of individual devices in the millimetre or centre. The smaller components have lower parasite resistances, capacitances and inductances, which result in advantages for speed and power consumption.
- **Speed**: The waveforms can be modified from logic 0 to logic 1 in much greater numbers of chips than in a circuit. Chip-to-to-chip communications will happen several times faster than chip-to-to-board communications.The board is built with higher speeds in mind because of its restricted dimensions..
- **Power consumption**: The efficacy of rational chip operations has increased dramatically. the primary reason for the decreased energy requirement is that small structures on a processor have lower capacitances.

5.3VLSI AND SYSTEMS

These benefits of integrated circuits transform into system-level advantages:

- **Smaller physical dimensions**: Smallness is always an asset per se—consider mobile or mobile phones.
- **Lower power consumption**:The substitution of one chip for a handful of regular parts eliminates overall power consumption. Reducing power is a ripple effect on the rest of the system: it may be used for smaller, cheaper power supplies; because lower power usage means lower heat, the ventilator will not be needed any more.
- **Reduced cost:** Reduced part number, energy supplies and cabinet prices, etc., would decrease device costs eventually. The sweeping result of integration is that the costs will be lower with a device made from custom ICs, even though the individual ICs cost more than their normal components. To explain why the embedded system technology has such a deep impact on the architecture of modern systems, both IC technology and IC and digital systems economies need to be understood.

5.4 TYPES OF CHIPS

The predominance of standard components revealed to the board developers, who used standard components the problems with the construction of custom systems.

Although conventional functions can require many more components where the ICs are used, designers tend to use simpler ICs because this will speed up the design process. While on the one hand, though, the industrial movement is toward making a wider variety of Icas available. The availability of microchips incorporates more options (or feature sets)

More specialized standard parts:

Normal modules were logic gates in the 1960's and LSI components in the 1970's. Today, common products include relatively advanced components: Collaboration, graphical processing, and floating-point accelerator boards Despite all these components being less complex than microprocessors, they are put in volume enough to justify developing chips tailored to them.

In reality, a complicated, high-performance feature on one chip also allows other applications – e.g. one-chip floating point processors make high-speed digital computations even on cheap personal computers available.

• Application-specific integrated circuits (ASICs)

The programmers will now produce a single chip for their individual use, rather than building a circuit out of standard components. Due to the processor specialisation, several common components may also be crammed onto one chip to reduce device capacity, strength, heat and expense. Relevant ICs are possible thanks to programming resources that help people build chips even faster

• Systems-on-chips (SoCs).

Manufacturing technology has progressed so far that a single processor may provide a full device. For instance, CPU, bus, I/O and memory may feature a single-chip computer. SoCs facilitate the production of systems at far lower costs than the comparable system at the board stage. SOCs can also be much maximum and power mimimun than board levels, as chip-to-chip links are more efficient.

Because of better-known and more-accurate manufacturing methods, there are also several more different types of chips. Conceptually, as the number of transistors on a chip increases, customised special-use ICs become easier and less expensive to produce. As few transistors as possible were incorporated on the chip, even simple jobs had to be designed with exquisite care. Nowadays, it is possible to insert thousands of imprecise transistors into a specification but be able to produce millions of precise ones in mass production.

While the chip can be smaller or quicker with more design work, the benefits of a quick-design single-chip function frequently overshadow the possible loss.

Production – the ability to use the millions of transistors effectively to fulfill a good purpose on a chip is the issue and the difficulty of producing such big chips.

5.5 FIELD-PROGRAMMABLE GATE ARRAYS(FPGA):

A gate array for field programming (FPGA) is a programmable logic block that can perforate multilevel logic functions. FPGAs are also employed to incorporate broad functions as separate commodity chips.

Small blocks of FPGA logic can, however, be useful on chip components for the consumer to configure part of the logical chip operation. A FPGA block must incorporate both combination logic and interconnect functions in order to build multilevel logic functions. There are several While there are several various FPGA coding techniques, we generally concentrate on SRAM-

programmed FPGAs for SRAM-based logic because anti-fuses and hard techniques are not used in most of them.

CHAPTER 6

TOOLS

6.1 Introduction:

There are two broad categories of resources available for this project.

- Need for hardware
- Required software

6.2 Requirements for Hardware:

- KIT FPGA

In the hardware compartment the ordinary machine needs a simple operation of Xilinx ISE 13.2 programme, i.e. Pentium III, 1 GB RAM, 20 GB hard drive device setup.

6.3 Requirements for Software:

- XILINX 13.2

Includes software version Xilinx ISE 13.2 that you can use Verilog source code to execute the specification.

6.4 XILINX ISE Introduction:

This instrument can be used for the development, execution, regeneration and integration of Verilog outlines on FPGA chips.

ISE: Environmental integrated software

- Environment for the refinement and testing of FPGA or CPLD-oriented computerised machine configuration
- Integrated device collection accessible via the GUI (XST: Xilinx Synthesis Technology)

Diverse dialects are based on the XST:

  - Verilog
  - VHDL

- XST creates a net rundown incorporating specifications •
- Any means necessary for the completion of the proposal is supported:
- Translating, guiding, placing and training
- It was a bit stream

In this case, Verilog can be used to create a test seat to validate the utility of a contour using host PC documentation for characterising jolts, for interfacing with the customer and for contrasting these natural.

A display in Verilog becomes the 'doors and wires' that are mapped to a sounding device such as a CPLD or FPGA, and it is then built the actual machine instead of the Verilog code being "executed" as if on some kind of processor chip.

6.4.1 Implementation:

- Synthesis (XST)

  -Produces an HDL definition netlist file

- Translate (NGDBuild)
- Conversion of all input template netlists to a single merged file, which explains logic and limitations.
- Take a netlist and group in the CLBs and IOBs logical elements.
- Mapping
- mapping the logic on system components (components of FPGA).
- Place and Route (PAR)
- Putting FPGA cells and connecting cells to
- Bit stream generation

XILINX Design Process

Step 1: Design entry

– HDL (Verilog or VHDL, ABEL x CPLD), Schematic Drawings, Bubble Diagram

Step 2: Synthesis

– Put files.v,.vhd,.sch in a newsproject file (.ngc)

Step 3: Implementation

– FPGA: Translate/Map/Place & Route, CPLD: Fitter,

Step 4: Configuring/Programming

– Download a BIT file in FPGA

– JEDEC programme file to Flash PROM

– MCS programme file

After steps 1, 2, 3 simulation will occur

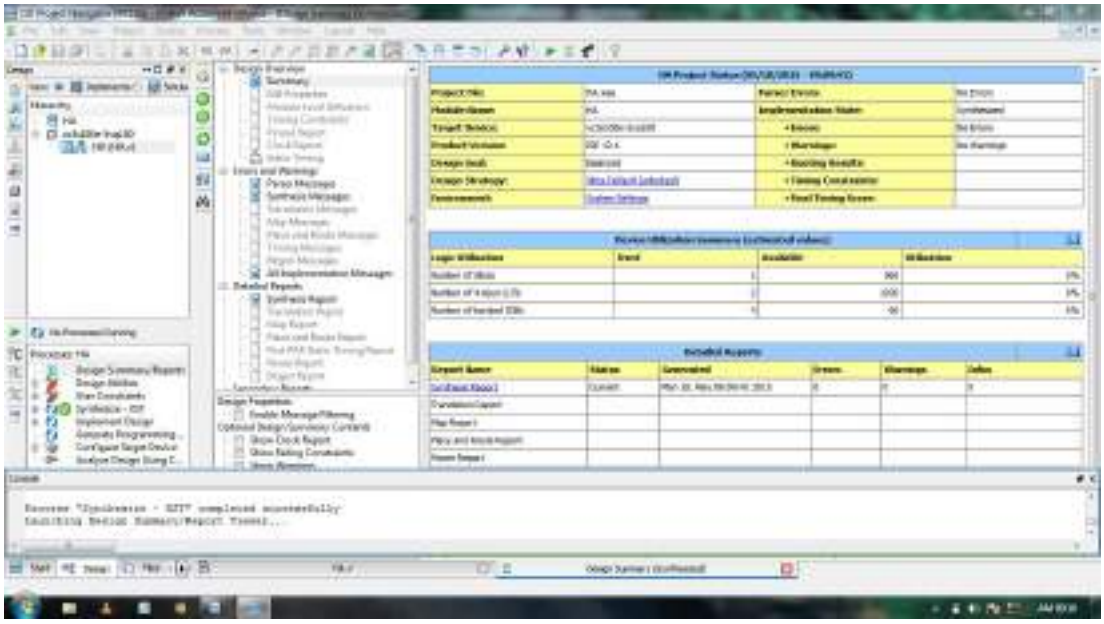
XILINX ISE 13.2 for simulation and synthesis is the methods used in this thesis. The programmes will be published in verilog.

6.5 Xilinx Software

Creative poly: Azipper Software a series of automated circuit architecture and FPGA (FPGA) designs (CPLD). The design process consisted of three parts: design, (creative)input, (practical implementation), execution (safe testing), and validation (e). Design protocol The above CAD methods are used for entering digital designs, either by a schematic input tool, the HDL (Hardware Definition Language) – Verilog, VHDL or both. We can use only the design flow involving Verilog HDL in this thesis.

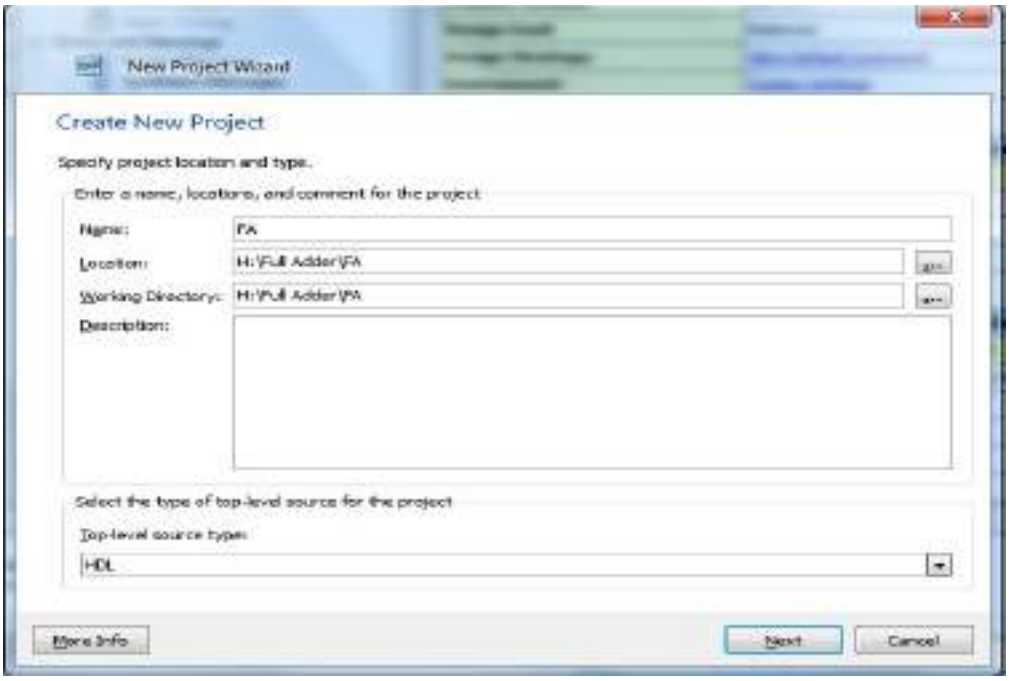
6.5.1 Creating a New Project

You can launch Xilinx Tools with a click on Windows desktop on Project Navigator Icon. The Project Navigator window should be opened on your computer. The last project accessed is seen in this window (See Figure 1).



6.5.2. Project startup

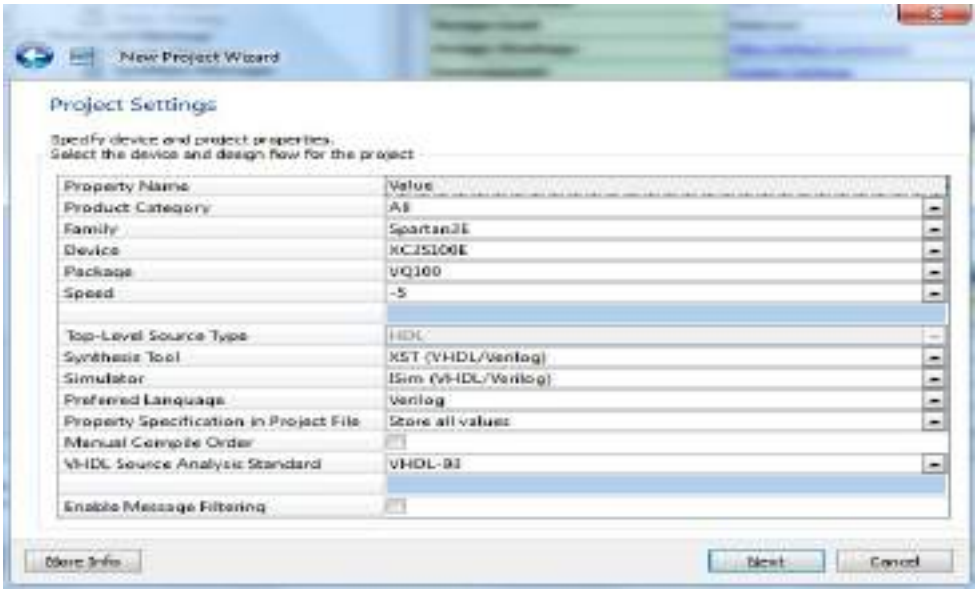
To create a new project, click on the "New Project" and select the template of choice. a new window will be opened on the screen (see Figure 2) Study up and complete the following items:



Name of the project: Write the next project's name

Project Path: the directory in which the current project is to be stored (Note: Do not indicate the location of the project as a desktop folder or a xilinx\bin folder. Your H: the safest place to put it is run. There are no spaces in the project position direction, e.g.: H:\Full Adder\FA should NOT be included. Leave the HDL module of the highest level kind.

Click NEXT and the following window should be displayed**:**



Search for a property in the registry and choose one of the list values.

Family Device: FPGA/CPLD family used. We use the In this study

Spartan3E FPGA's.

Unit: the current device number. You can join XC3S100E for which laboratory (on the board in addition to this, the board often seen in the creative section).

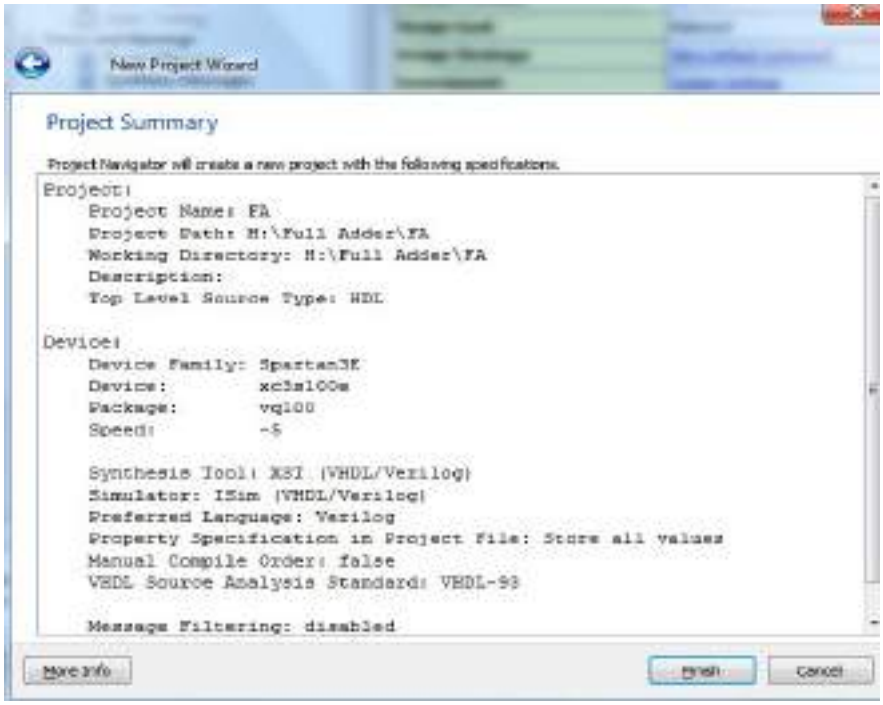
delegation Do sort to have a package. In this testing laboratory, the Spartan FPGA is kept in a VQ100 box.

Degree Speed: The degree Speed is "-5."

Synthesis Tool: XST [VHDL/Verilog].

Simulator: the platform for simulating and verifying design features. The Xilinx ISE is built into the Modelsim simulator. So choose "Modelsim-XE Verilog" as the simulator or you can use also Xilinx ISE Simulator.

To save entries, click NEXT**.**

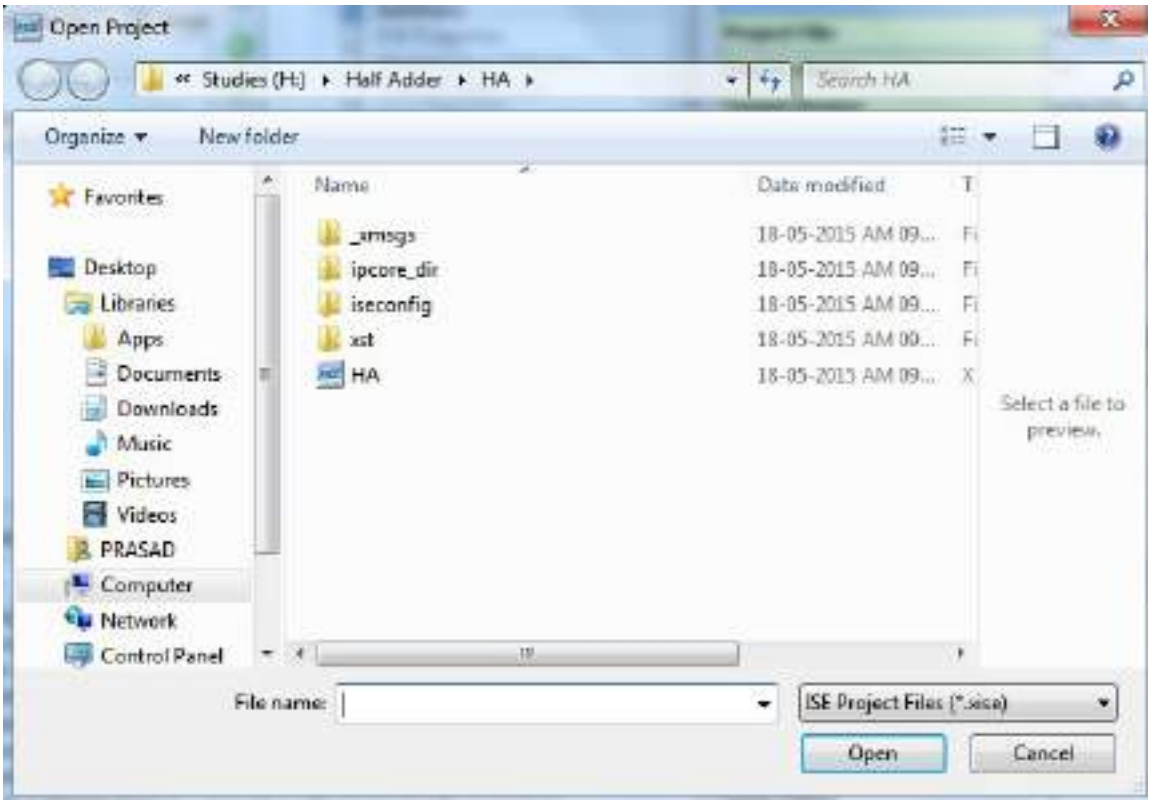


Select Finish to open a project review window.

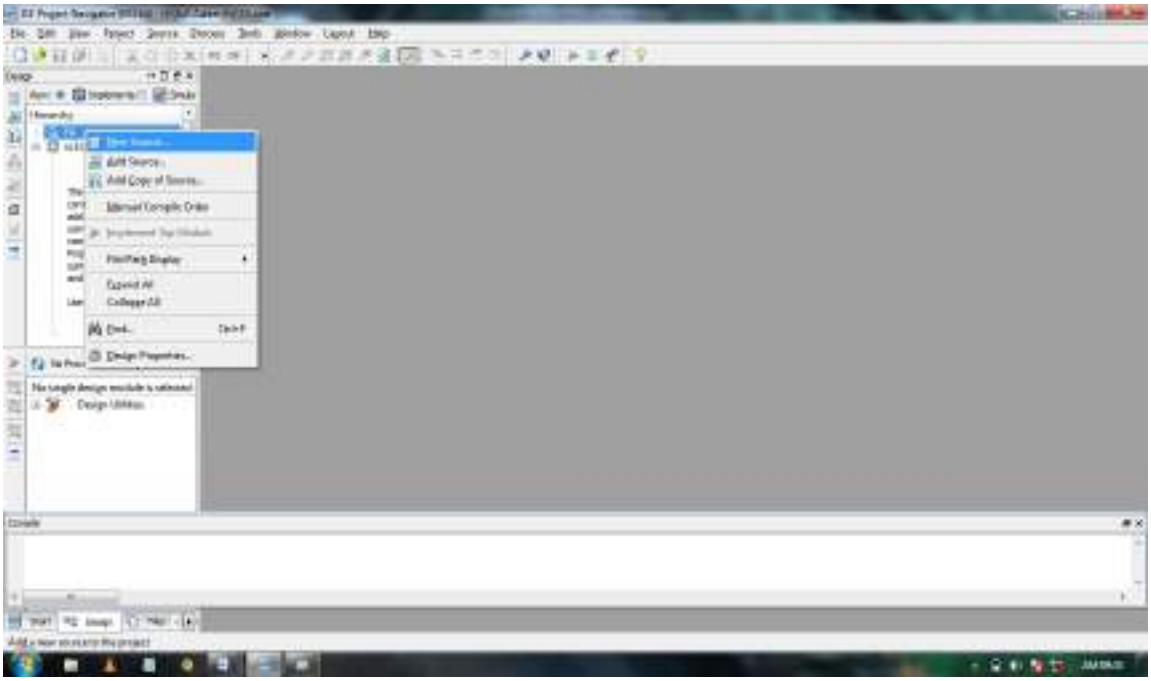
Let's access the project by selecting "File -> Open Projects on Machine".

project in Xilinx Tools. Select your project and press OK.

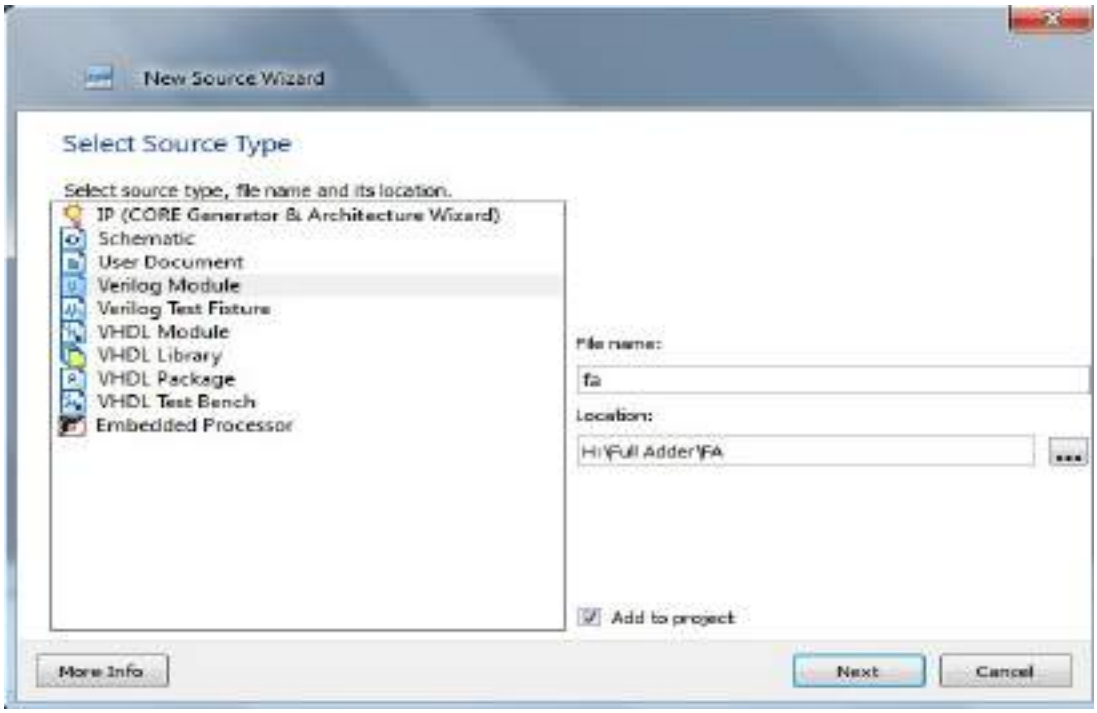
The following window can be found by clicking on NEXT in the window above:



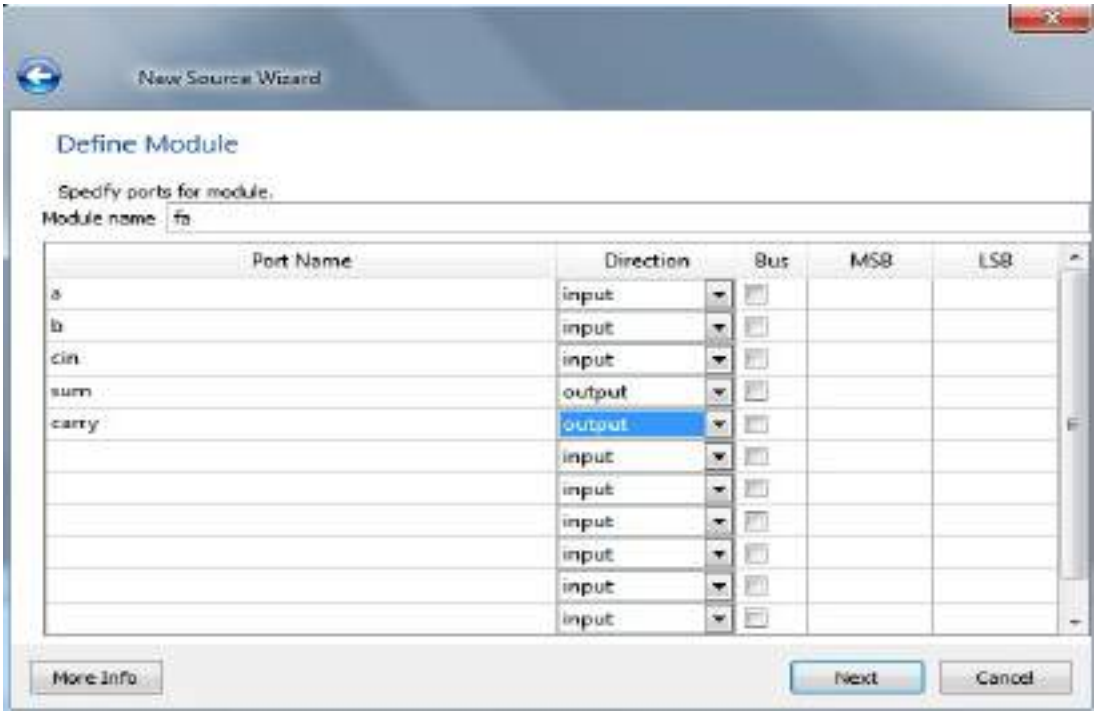
When making a new source file, choose NEW and then choose the Source.



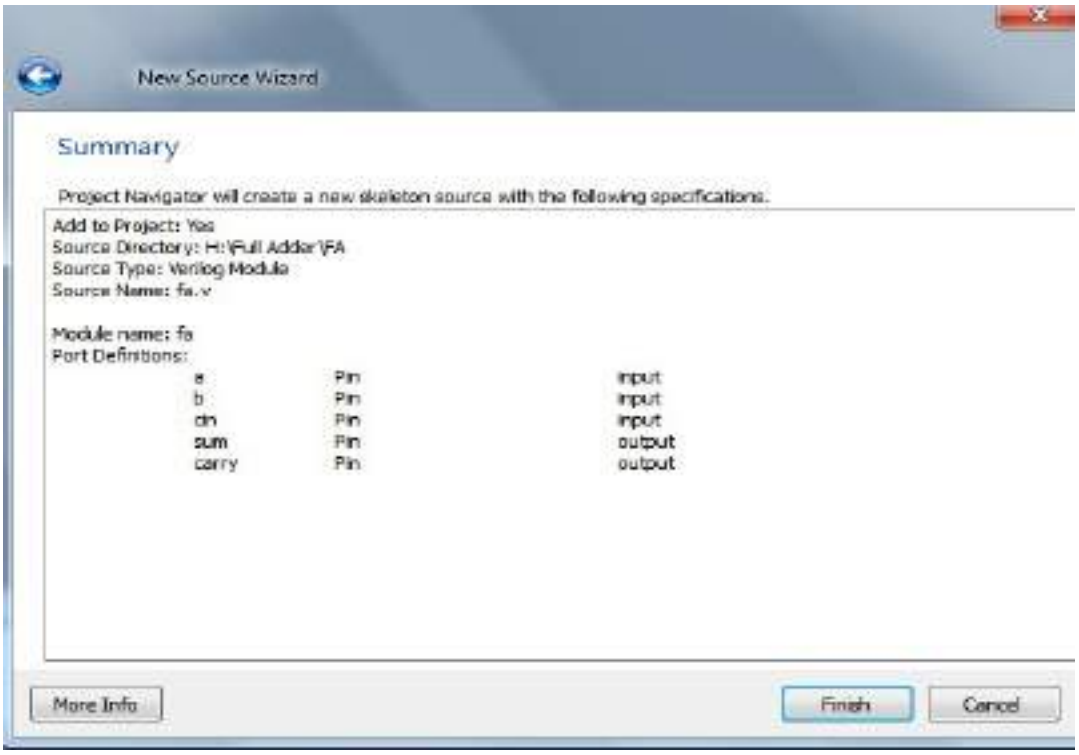
A pop-up window appears..



Enter Verog module and in the “Input:” box, type a new filename to use. Go to the Project Settings section. To acknowledge the submissions, press the Next button. The following window is now open.

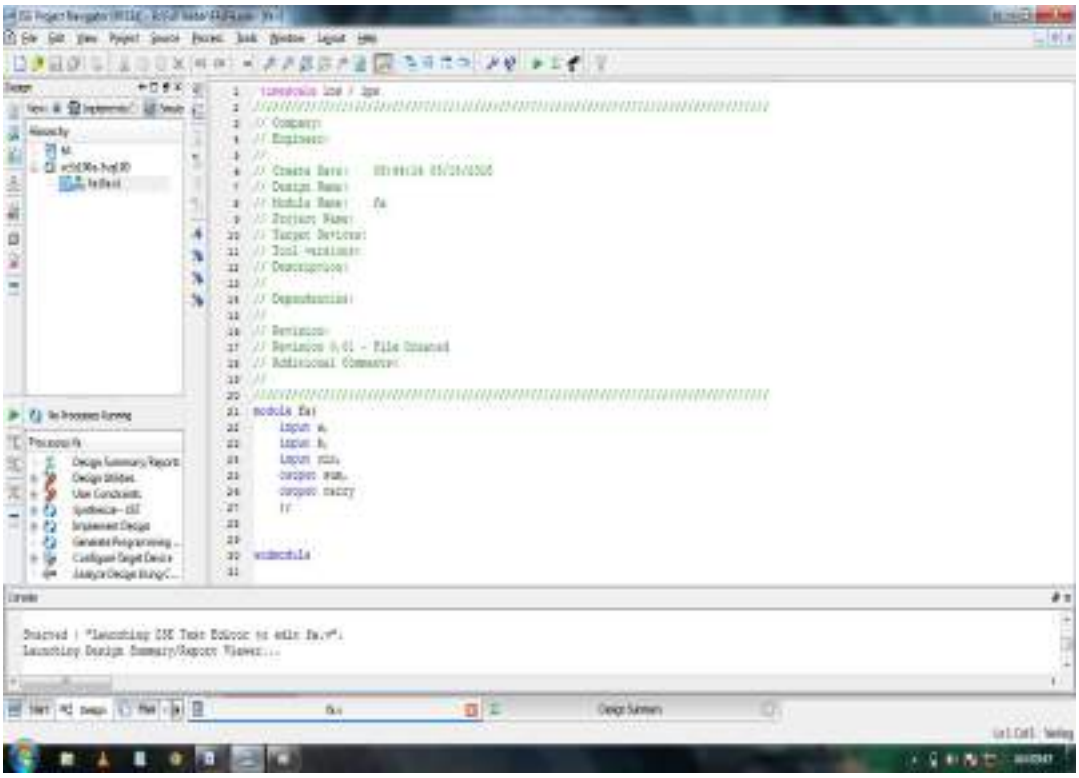


Enter the names of all input and output pins and state the address accordingly in the Port Name column. By entering the necessary bit numbers in the columns of **MSB/LSB**, a Vector/Bus can be specified. To see all of the latest sources, click Next> to get a window.

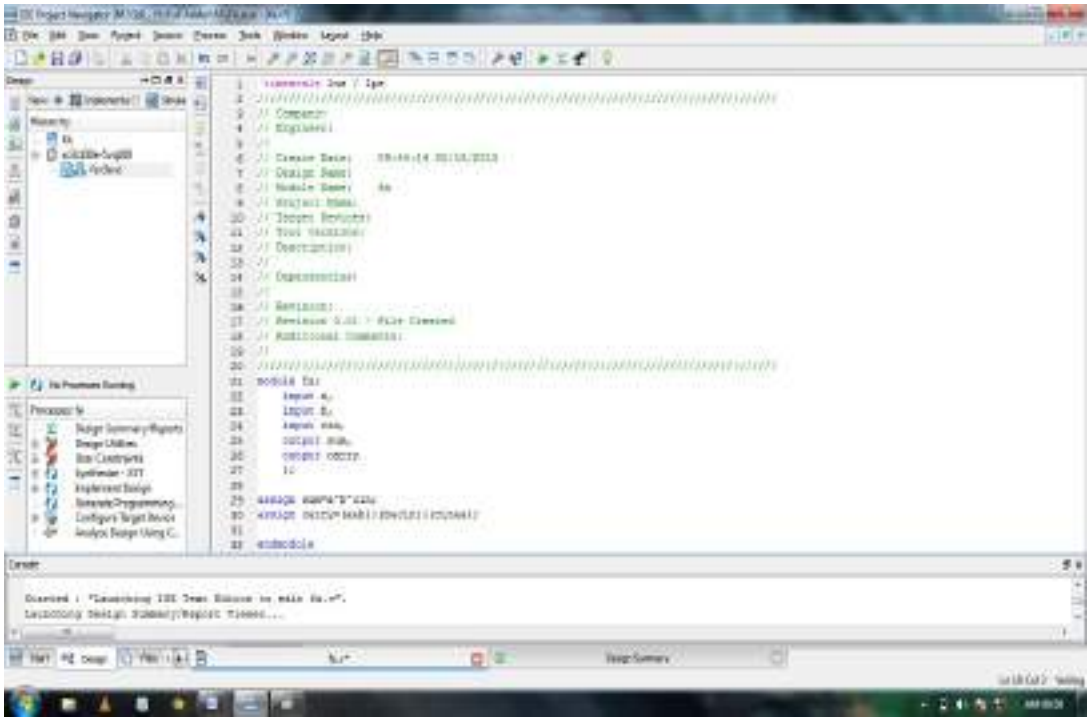


click on **Finish** to continue.

The project navigator window now shows the file root file.

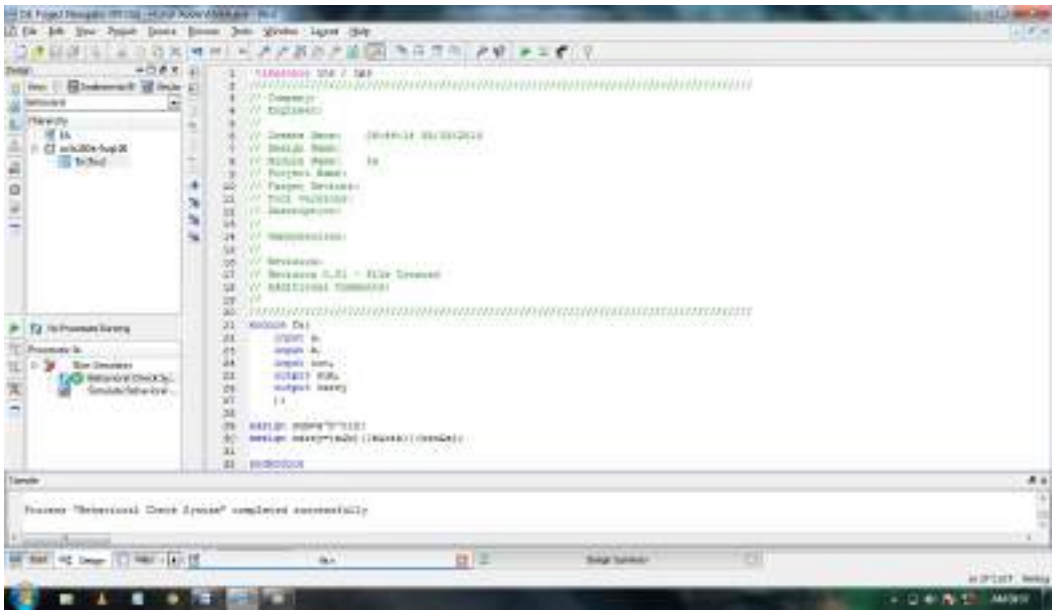


When the source file is modified as a text editor, the source window may be used as a text tool. All of the pins on the input/output will be represented. By selecting the Save menu option, you automatically save your Verilog. You can also add programmes to any text editor and include them in the project by selecting "Link to project folder" and adding "Add Copy".

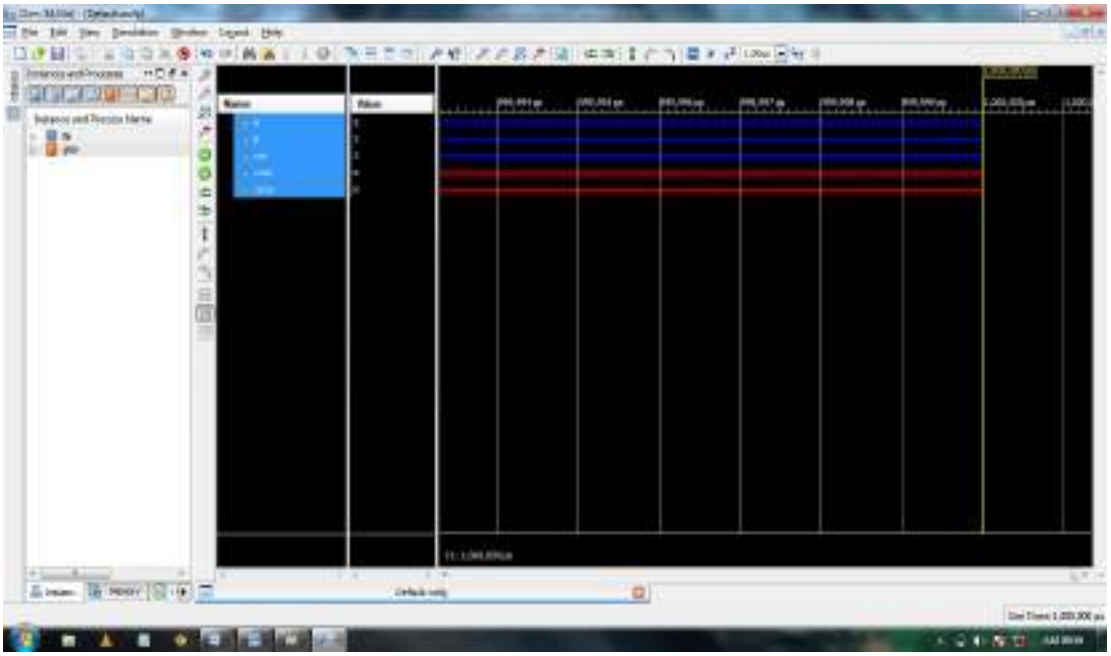


6.5.3. Output Waveforms Simulation and View

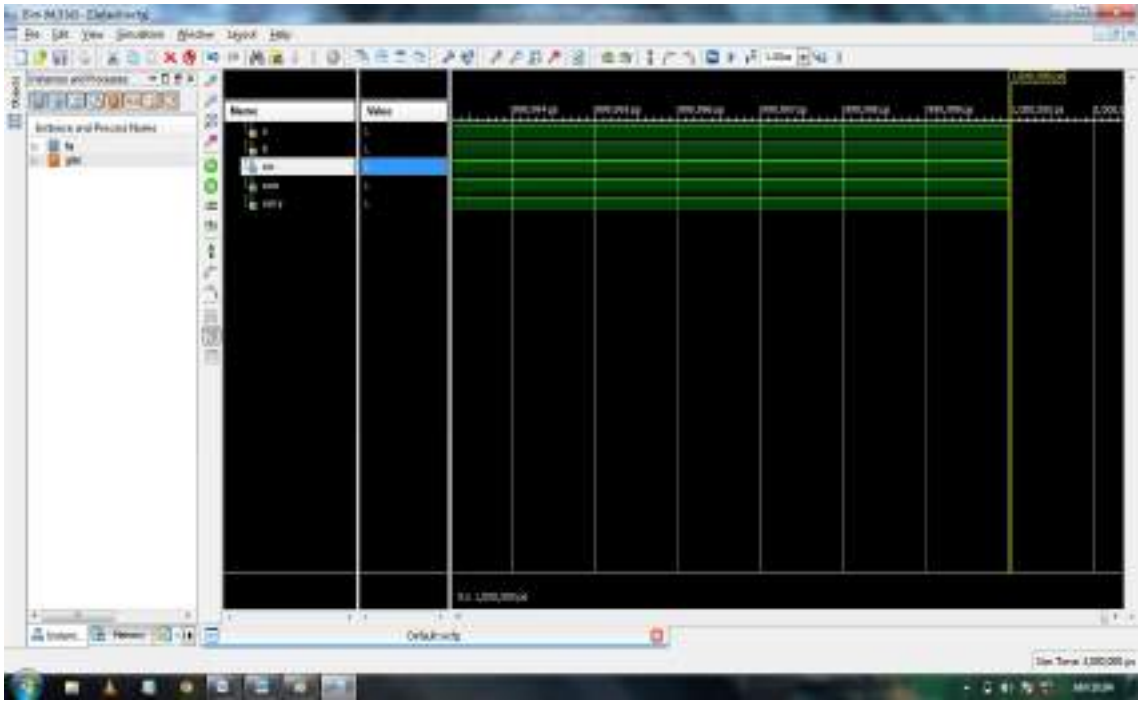
and dial intensity. Analyze to pick an existing file and review ISIM Syntax and then click on the Errors menu item.



Simulate behavioural models where there are no defects. and a pop-up appears..

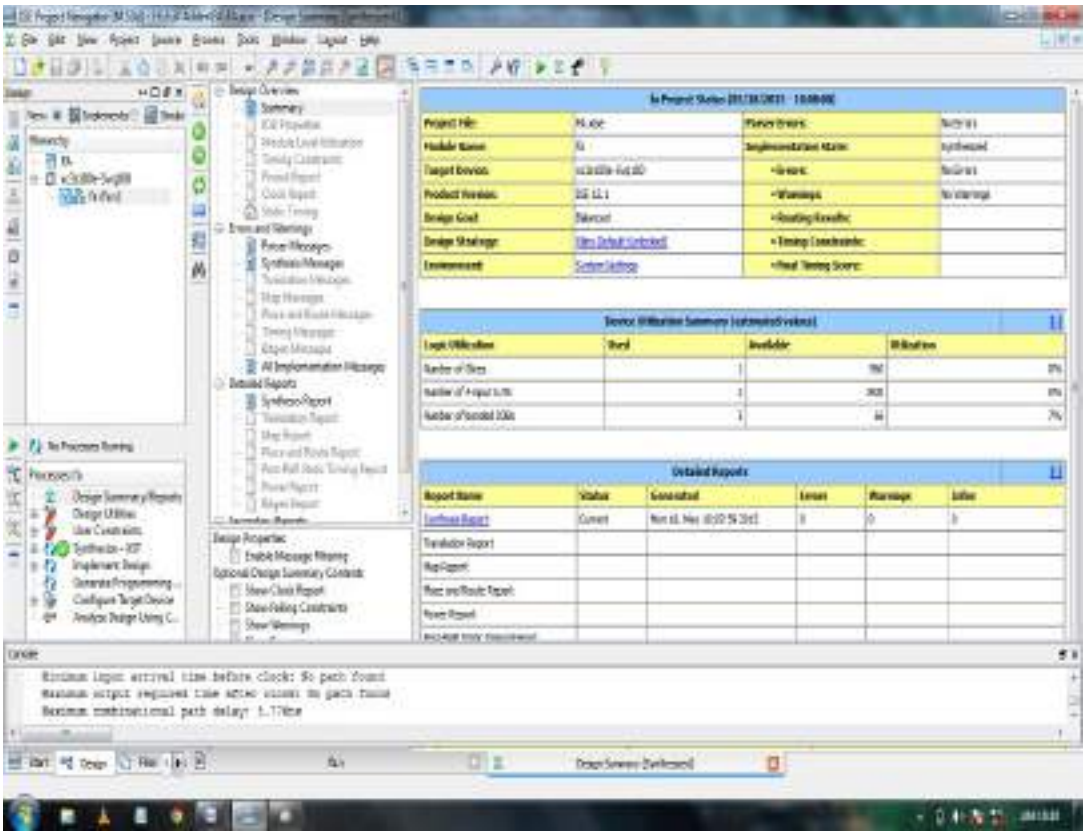


Here we will input the parameters. Use the right-and-click feedback constant tool to edit the input. The play button in the toolbar will check your sound input and output.

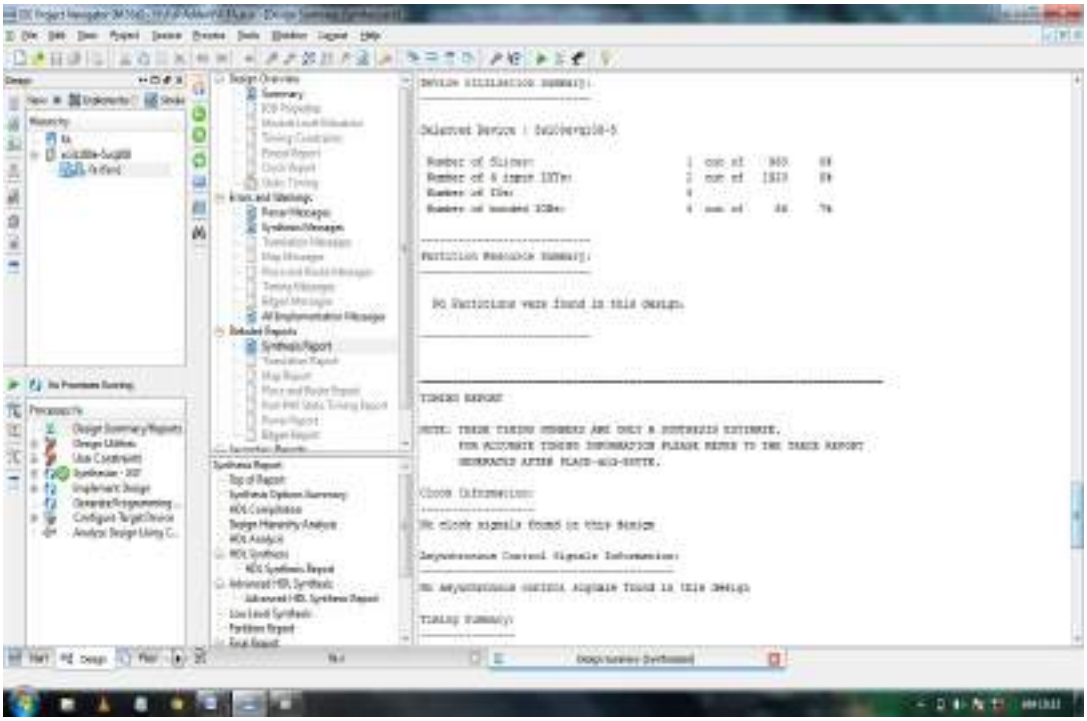


6.5.4. Design synthesization and implementation

Click on Practical Applications Pick and right-and-click a file and choose "Synthesize" to generate XML from XSD If mistakes are made, fix them. If all is right, click on Design & Statistics Overview and try results.

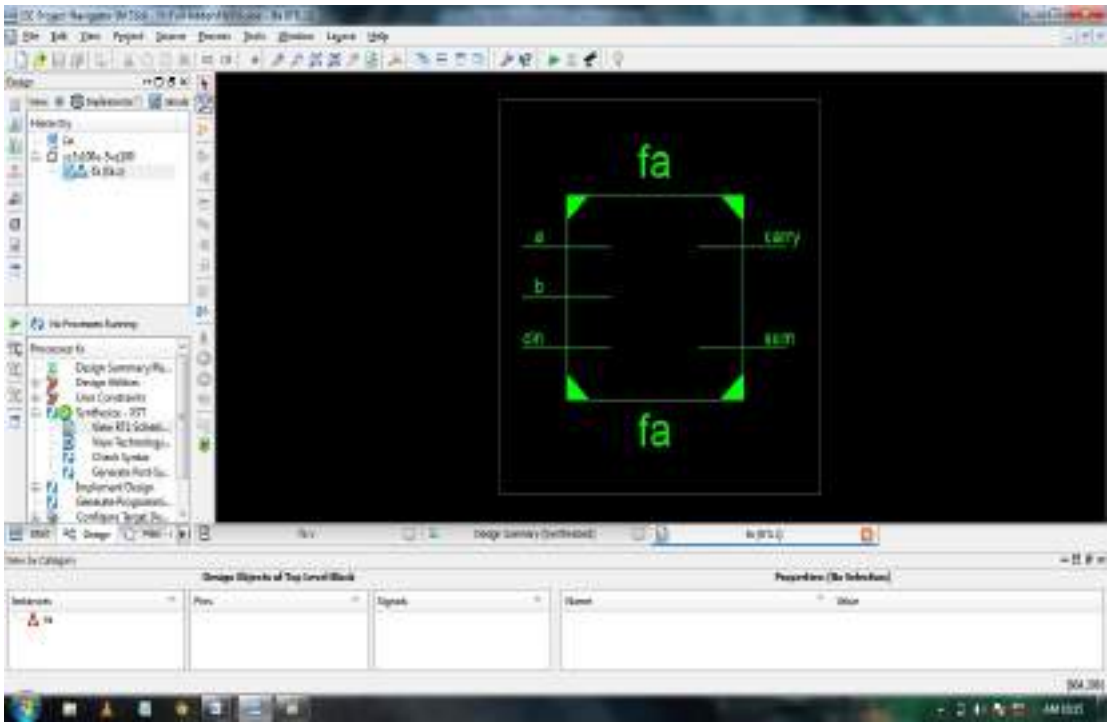


View the Synthetic report in the System Summary and the results will appear.

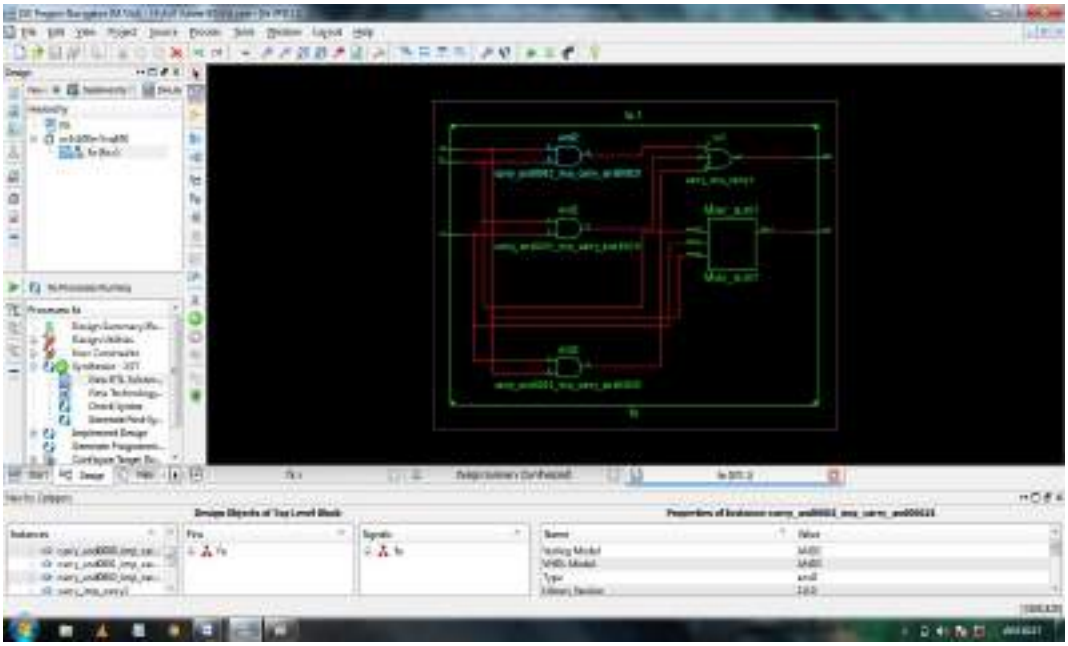


6.5.5. See RTL Schematic:

return to RTL viewExpand Synthesizer and then click on OK.



internal modules are shown in the Top module window Go to the upper module.



6.6 FPGA MODEL FLOW:

We will have a brief introduction to the FPGA interface flow in this section of the tutorial. The flow chart shows a condensed version of the concept flow

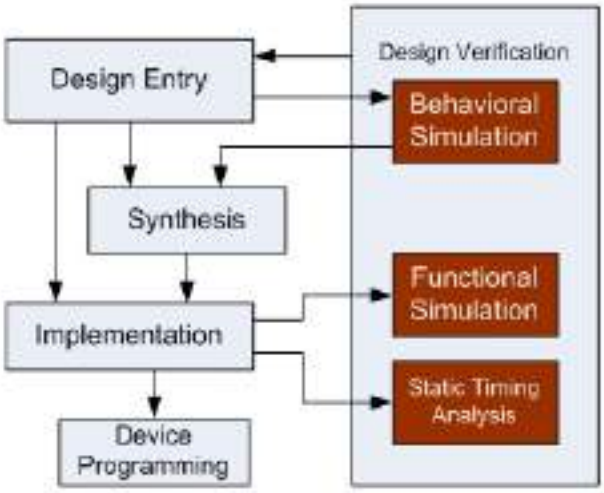


Figure 6.1 FPGA Design Flow

6.7 Design Entry:

There are many concept entry methods. Scheme-based description, hardware language description and combination, etc. The choice of a tool depends on the designer. The schematic entry is the best choice if the designer wishes to deal more with hardware. Even if the architecture can be complex, relatively simple strategies are the easiest. Entry-based compared to scoring but in language and not focused on density and performance.

HDL describes abstracts off of the implementation information, leaving design specs to the programmers. Schematic-based entry provides programmers with even more hardware exposure. It is the best option for hardware-oriented individuals. State machines are another tool, but seldom used.

For designers who see architecture as a set of nations, it is the best option. However, the resources for entering state machines are restricted. We will deal with the HDL concept entry in this documentation.

6.8 Synthesis:

How to translate VHDL or Veril netlist code into a VHDL or Veril netlist file for example, the whole circuit design In the case of several sub-designs, for example to implement the processor, there must be a CPU, and a RAM, so it does not matter which one we have. The synthesis process makes sure that the code is correct syntax, as well as analyses the design hierarchy. The

resulting netlist(s), for Xilinx ® Synthesis Technology (XST)) are saved in NGC (Native Generic Circuit) register.

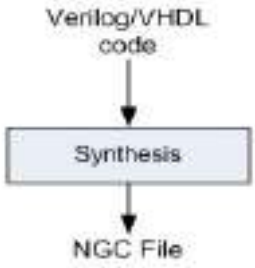


Figure 6.2 FPGA Synthesis

6.9 Implementation:

The construction of the Spartan 3E and IDWT works was carried out using Verilog HDL and are synthesised using the XILINX ISE Tool on the FPGA family. This method comprises:

- Map
- Translate
- Route and place

6.9.1 Translate:

Process integrates all input netlists with logic architecture file restrictions. Saved as an NGD (Generic Native Database) disc. This can be achieved with the software NGD Construct. Defining restrictions here is simply the assignment of the ports for the design to the targeted devices and time specifications for the design physique (e.g. pins, switches, buttons etc). These data are saved in a UCF file (User Constraints File). PACE, Limited Editor etc are the tools used to create or change UCFs.

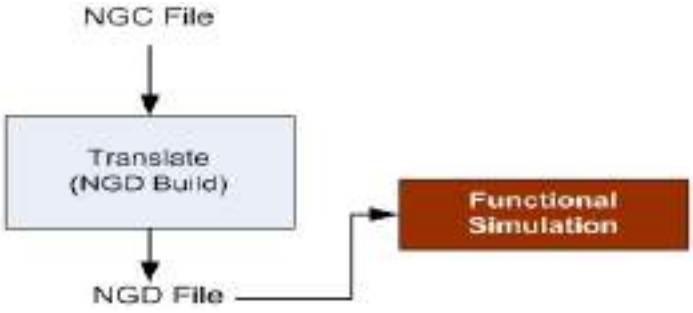


Figure 6.3 FPGA heritance

6.9.2 Map:

Any circuit is decomposed into smaller pieces of logical elements that match with FPGA blocks. The end result of this is that the map method will match the concept of the NGD file, and an NCD will be produced that explains the physical configuration of the mapped components to the FPGA design. Creating an NCD file, or association may also be a viable option (the NGD description).

For that function, the MAP software is used.

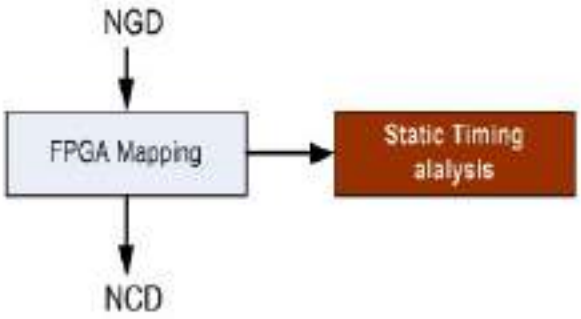


Figure 6.4 FPGA map

6.9.3 Placement and Routing:

PAR is applied here. When the path and path are followed, the sub blocks are placed in the order of the operation. Adjective is saying: For instance, putting a sub-in the IO closer to the IO pin may save time, but can place other constraints on it.

The site and route method therefore takes into account the compromise between all the constraints

The PAR tool uses the traced input NCD file and manufactures the completely routed output NCD file. Routing details is the NCD file output.

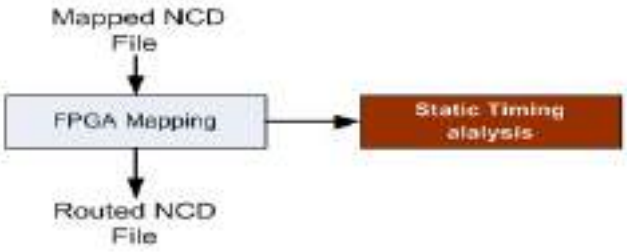


Figure 6.5 FPGA Place and route

6.10 Device Programming:

It is now necessary to load the template on the FPGA. However, the specification has to be translated in a way to be accepted by the FPGA. The BITGEN conversion software. Then the routed NCD file is sent to the BITGEN software to create a bit stream (a.BIT file) for the target FPGA interface to be configured. You can do this with a cable. The cable selection depends on the structure

6.10.1 Design check:

validation of process phases which be carried out at various stages.

6.10.2 Behavioral Simulation (RTL Simulation):

These are first and foremost computation phases, which are found in the concept flow structure. This simulation is done to test RTL (behavioural) code before the synthesis and to ensure that the design works as planned.

Conduct simulation on either VHDL or Verilog designs may be carried out. This is done by observing signals and variables, tracing procedures and functions and setting break points.

This is a really quick simulation, so the author can modify the HDL code in a short time if the desired functionality is not met. Since the architecture has not been synthesised yet, it is still unclear what the time and resource use properties are

6.10.3 Functional simulation (Post Translate Simulation):

Functional simulation provides details on the circuit logic. After the translation process, designers can check the functionality of the template. If the feature is not as intended, then the designer must update the code and follow the design process steps again.

Static Timing Analysis:

This is possible after the processes of MAP or PAR The MAP timing summary lists the delays in signal direction of the design resulting from the design logic. The Timing Report includes timing delay information to include a detailed timing description of the design.

CHAPTER 7

RESULTS

Simulation.

Output waveform of existing system

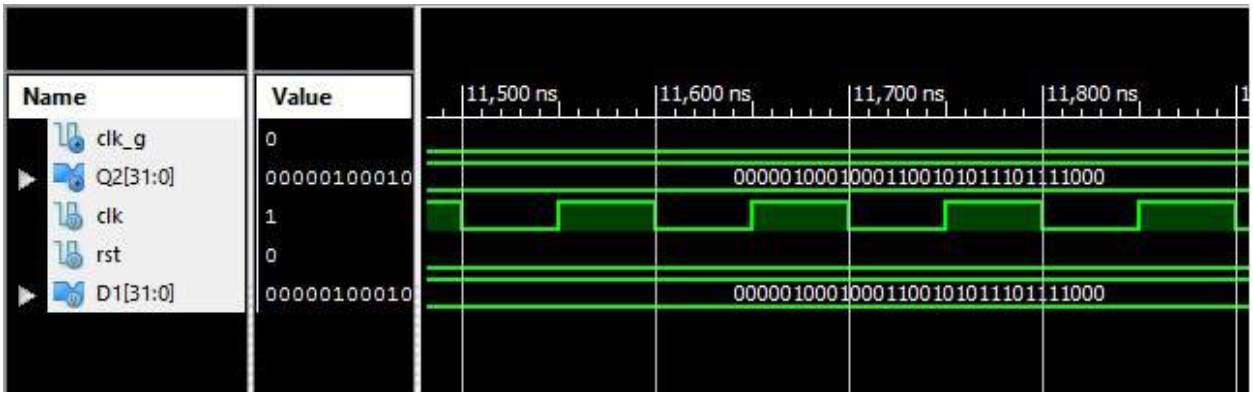


Figure 7.1 Output waveform of existing

Output waveform of proposed system

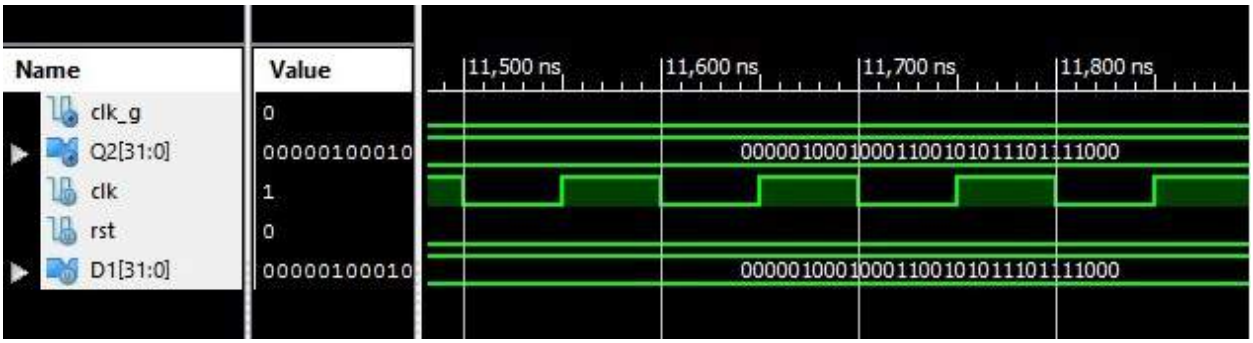


Figure 7.2 Output waveform of proposed

Design Summary.

Design summary and implementation of existing system

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slices	37	4656	0%
Number of Slice Flip Flops	64	9312	0%
Number of 4 input LUTs	32	9312	0%
Number of bonded IOBs	67	232	28%
Number of GCLKs	1	24	4%

Figure 7.3 Design summary and implementation of existing system

Design summary and implementation of proposed system

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slices	35	4656	0%
Number of Slice Flip Flops	64	9312	0%
Number of bonded IOBs	66	232	28%

Figure 7.4 Design summary and implementation of proposed system

Synthesis report of existing system

```
Timing Summary:
---------------
peed Grade: -5

  Minimum period: No path found
  Minimum input arrival time before clock: 2.990ns
  Maximum output required time after clock: 4.040ns
  Maximum combinational path delay: No path found

iming Detail:
-------------
ll values displayed in nanoseconds (ns)

=========================================================================
iming constraint: Default OFFSET IN BEFORE for Clock 'clk'
 Total number of paths / destination ports: 62 / 62
-------------------------------------------------------------------------
ffset:              2.990ns (Levels of Logic = 2)
 Source:            rst (PAD)
 Destination:       m1/m1/Q (FF)
 Destination Clock: clk rising

 Data Path: rst to m1/m1/Q
                              Gate     Net
   Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
   ----------------------------------------  ------------
    IBUF:I->O             93   1.106   1.089  rst_IBUF (rst_IBUF)
    begin scope: 'm1'
    begin scope: 'm1'
    FDR:R                      0.795          Q
   ----------------------------------------
   Total                       2.990ns (1.901ns logic, 1.089ns route)
                                       (63.6% logic, 36.4% route)
```

```
================================================================================
Timing constraint: Default OFFSET IN BEFORE for Clock 'clk_g_OBUF'
  Total number of paths / destination ports: 62 / 62
--------------------------------------------------------------------------------
Offset:              2.990ns (Levels of Logic = 1)
  Source:            rst (PAD)
  Destination:       m2/m1/Q (FF)
  Destination Clock: clk_g_OBUF rising

  Data Path: rst to m2/m1/Q
                                Gate     Net
    Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     IBUF:I->O             93   1.106   1.089  rst_IBUF (rst_IBUF)
     FDR:R                      0.795          m2/m31/Q
    ----------------------------------------
    Total                       2.990ns (1.901ns logic, 1.089ns route)
                                        (63.6% logic, 36.4% route)


================================================================================
Timing constraint: Default OFFSET OUT AFTER for Clock 'clk_g_OBUF'
  Total number of paths / destination ports: 31 / 31
--------------------------------------------------------------------------------
Offset:              4.040ns (Levels of Logic = 1)
  Source:            m2/m31/Q_1 (FF)
  Destination:       Q2<31> (PAD)
  Source Clock:      clk_g_OBUF rising

  Data Path: m2/m31/Q_1 to Q2<31>
                                Gate     Net
    Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     FDR:C->Q              1   0.514   0.357  m2/m31/Q_1 (w1<31>_0)
     OBUF:I->O                 3.169          Q2_31_OBUF (Q2<31>)
    ----------------------------------------
    Total                       4.040ns (3.683ns logic, 0.357ns route)
                                        (91.2% logic, 8.8% route)

================================================================================
```

Synthesis report of proposed system

```
Timing Summary:
---------------
Speed Grade: -5

   Minimum period: 1.139ns (Maximum Frequency: 877.963MHz)
   Minimum input arrival time before clock: 2.982ns
   Maximum output required time after clock: 4.040ns
   Maximum combinational path delay: No path found

Timing Detail:
--------------
All values displayed in nanoseconds (ns)

=========================================================================
Timing constraint: Default period analysis for Clock 'N0'
  Clock period: 1.139ns (frequency: 877.963MHz)
  Total number of paths / destination ports: 32 / 32
-------------------------------------------------------------------------
Delay:               1.139ns (Levels of Logic = 0)
  Source:            m1/m2/q_31 (FF)
  Destination:       m2/m2/q_31 (FF)
  Source Clock:      N0 falling
  Destination Clock: N0 falling

  Data Path: m1/m2/q_31 to m2/m2/q_31
                              Gate     Net
    Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     FDR_1:C->Q            1   0.514   0.357  m1/m2/q_31 (m1/q2<31>)
     FDR_1:D                   0.268          m2/m2/q_31
    ----------------------------------------
    Total                      1.139ns (0.782ns logic, 0.357ns route)
                                       (68.7% logic, 31.3% route)
```

```
================================================================================
Timing constraint: Default OFFSET IN BEFORE for Clock 'N0'
  Total number of paths / destination ports: 96 / 96
--------------------------------------------------------------------------------
Offset:              2.982ns (Levels of Logic = 1)
  Source:            rst (PAD)
  Destination:       m1/m2/q_31 (FF)
  Destination Clock: N0 falling

  Data Path: rst to m1/m2/q_31
                                Gate     Net
    Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     IBUF:I->O             64   1.106   1.081  rst_IBUF (rst_IBUF)
     FDR_1:R                    0.795          m1/m2/q_0
    ----------------------------------------
    Total                       2.982ns (1.901ns logic, 1.081ns route)
                                        (63.7% logic, 36.3% route)


================================================================================
Timing constraint: Default OFFSET OUT AFTER for Clock 'N0'
  Total number of paths / destination ports: 32 / 32
--------------------------------------------------------------------------------
Offset:              4.040ns (Levels of Logic = 1)
  Source:            m2/m2/q_31 (FF)
  Destination:       Q2<31> (PAD)
  Source Clock:      N0 falling

  Data Path: m2/m2/q_31 to Q2<31>
                                Gate     Net
    Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     FDR_1:C->Q            1   0.514   0.357  m2/m2/q_31 (m2/q2<31>)
     OBUF:I->O                 3.169          Q2_31_OBUF (Q2<31>)
    ----------------------------------------
    Total                       4.040ns (3.683ns logic, 0.357ns route)
                                        (91.2% logic, 8.8% route)

================================================================================
```

CHAPTER 8

CONCLUSION

Clock gating is used to reduce power consumption. Sequence circuits are used to help save power by clock gates and multi-bit flops. For the saving of energy, popular clock gating is used. However, clock gating makes many more redundant pulses. Multi-bit flip-flops are often used to minimise battery usage. Using the multi-bit Flip-Flop method, the total inverter number is removed by exchanging the flip-flop inverters. The additional power saving would be increased by combining multi-bit flip-flop with data driven clock gates. This proposed work is implemented in Xilinx tool. In order to achieve additional power savings, the combination of data driven clock gating with DET-MBFF is used.

REFERENCES

1. Kapoor, Ajay, Cas Groot, Gerard Villar Pique, HamedFatemi, Juan Echeverri, Leo Sevat, Maarten Vertregt et al. “Digital systems power management for high performance mixed signal platforms.” Circuits and Systems I: Regular Projects, IEEE Transactions on 61, no. 4 (2014): 961-975.

2. Wimer, Shmuel, and Israel Koren. “The optimal fan-out of clock network for power minimization by adaptive gating.” Very Large Scale Integration (VLSI) Systems, IEEE Transactions on 20, no. 10 (2012): 1772-1780.

3. Santos, Cristiano, Ricardo Reis, GuilhermeGodoi, Marcos Barros, and Fabio Duarte. “Multi-bit flip-flop usage impact on physical synthesis.” In Integrated Circuits and Systems Design (SBCCI), 2012 25th Symposium on, pp. 1-6. IEEE, 2012.

4. Yan, Jin-Tai, and Zhi-Wei Chen. “Construction of constrained multi-bit flip-flops for clock power reduction.” In Green Circuits and Systems (ICGCS), 2010 International Conference on, pp. 675-678. IEEE, 2010. 15

5. Jiang, IH-R., Chih-Long Chang, and Yu-Ming Yang. “INTEGRA: Fast multibit flip-flop clustering for clock power saving.” Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on 31, no. 2 (2012): 192-204.

6. Chang, Chih-Long, and Iris Hui-Ru Jiang. “Pulsed-latch replacement using concurrent time borrowing and clock gating.” IEEE Transactions on ComputerAided Design of Integrated Circuits and Systems 32, no. 2 (2013): 242-246.

7. Lo, Shih-Chuan, Chih-Cheng Hsu, and Mark Po-Hung Lin. "Power optimization for clock network with clock gate cloning and flip-flop merging." In Proceedings of the 2014 on International symposium on physical design, pp. 77-84. ACM, 2014.

8. Wimer, Shmuel, DoronGluzer and Uri Wimer. “Using well-solvable minimum cost exact covering for VLSI clock energy minimization.” Operations Research Letters 42, no. 5 (2014): 332-336.

9. Wimer, Shmuel, and Israel Koren. “Design flow for flip-flop grouping in datadriven clock gating.” Very Large Scale Integration (VLSI) Systems, IEEE Transactions on 22, no. 4 (2014): 771-778.

10. Wimer, Shmuel. “On optimal flip-flop grouping for VLSI power minimization.” Operations Research Letters 41, no. 5 (2013): 486-489.

11.SpyGlass Power [Online]. Available: Using many advanced algorithms and analysis techniques, the SpyGlass platform provides designers with insight about their design, early in the process at RTL. It functions like an interactive guidance system for design engineers and managers, finding the fastest and least expensive path to implementation for complex SoCs.http://www.atrenta.com/solutions/spyglassfamily/spyglass-power.html