A LOW POWER SELECTIVE MEDIAN FILTER DESIGN

A THESIS REPORT SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

Master of Technology

In

Computer Science and Engineering

By

Radhamadhab Dalai

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING,

NATIONAL INSTITUTE OF TECHNOLOGY, ROURKELA,

PIN-769008 (ORISSA)

JUNE-2008

A LOW POWER SELECTIVE MEDIAN FILTER DESIGN

A THESIS REPORT SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

Master of Technology

In

Computer Science and Engineering

By

Radhamadhab Dalai

Under The Guidance of

Prof. Banshidhar Majhi

Dept. of Computer Science. & Engg.

&

Prof. Kamala Kanta Mahapatra

Dept. of Electronics & Communication Engg.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING,

NATIONAL INSTITUTE OF TECHNOLOGY, ROURKELA,

PIN-769008 (ORISSA)

JUNE-2008

National Institute of Technology Rourkela
CERTIFICATE

This is to certify that the thesis entitled, **"A Low Power Selective Median Filter Design"**, submitted by Sri.Radhamadhab Dalai (Roll No:-20606002) in partial fulfillment of the requirements for the award of Master of Technology Degree in Computer Science and Engineering at National Institute of Technology, Rourkela (Deemed University) is an authentic work carried out by him under our supervision and guidance.

To the best of our knowledge, the matter embodied in the thesis has not been submitted to any other University / Institute for the award of any Degree or Diploma.

Prof. Kamala Kanta Mahapatra
Professor
Dept. of Electronics and Commn. Engg.

Prof. Banshidhar Majhi
Professor
Dept. of Computer Science and Engg.

National Institute of Technology
Rourkela-769008
May 30 2008

Acknowledgment

No thesis is created entirely by an individual, many people have helped to create this thesis and each of their contribution has been valuable. I express my sincere gratitude to my thesis supervisors, Dr. Banshidhar Majhi, Professor and Head of the Department, Computer Science and Engineering, and Dr. Kamala Kanta Mahapatra, Professor, Electronics and Communication Engineering for their kind and valuable guidance for the completion of the thesis work.

I would also take this opportunity to express my gratitude and sincere thanks to my honorable teachers Dr. S. K. Rath, Dr. D P Mahapatra, SL. Bibhudatta Sahoo, Prof. R Baliarsingh, Dr. A. K. Turuk, Dr. S .K. Jena for their invaluable advice, constant help, encouragement, inspiration and blessings.

I would also like to convey my special thanks to Sri P K Sahu, Lecturer, Department of Electrical Engineering and Jitendtra kumar Sathpathy,Ayaskant Swain,Sushant Mahapatra for providing incredible support and valuable resources during my project implementation.

Submitting this thesis would have been a Herculean job, without the constant help, encouragement, support and suggestions from my friends, especially Atanu S. Bal, Tankadhar Mahanta, Saurav Ganguli, Jitendra K. Singh, Arun Kumar, Prem S. Tondon,E Ashwin Kumar, K S Babu for their time to help. Although it will be difficult to record my appreciation to each and every one of them in this small space; I will feel guilty if I miss the opportunity to thank Tarun Sureja ,Arindam Makur, Yogesh Bani**,** Pushpendra Chandra**,** G Raghav Rao, and others. I will relish your memories for years to come.

Radhamadhab Dalai
Roll No. 20606012
M. Tech, Department Of Computer Science and Engineering

Contents

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

A selective median filter which consumes less power has been designed and different logics for majority bit evaluation has been applied and simulated in VHDL .It is rightly called as selective because an edge pixel detector [2] has been used to select those pixels which are to be processed through median filter. As for median value calculation; sorting of 3 x 3 window's pixel values has been done using majority bit circuit [4]. Different majority bit calculation method has been implemented and the result sorting circuit has been analyzed for power analysis. In this work a general median filter which uses binary sorting method known as Majority Voting Circuit (MVC) has been designed using VHDL and optimized using SYNOPSIS which has used 0.13µm CMOS technology .The digital design of sorting circuit saves approximately 60% of power comprising of cell leakage and dynamic power comparing to a mixed signal design of Floating gate based Majority bit median filter [4]. Before operating median filter on each pixel double derivative filter [2] has been applied to check whether it is an edge pixel or not. Overall this is a digital design of a mixed filter which preserves edges and removes noises as well. Low power techniques at logic level and algorithmic level have been embedded into this work. In our work we have also designed a small microprocessor using VHDL code. Later a memory (for the purpose of image storing) based Control Unit for single median value evaluation has been designed and simulated in XILINX. Here for sorting circuit a common logic based circuit (component) has been put forward. The power, latency or delay, area of whole design has been compared and tested with other designs.

Introduction

A selective median filter is a mixed filter which removes spike noise (impulsive noise or most commonly called as salt and pepper) from a noisy image while preserving sharp edges. It is rightly called as 'selective' because this filter first checks each pixel whether it is a noisy or non-noisy. For this purpose we have used a double derivative filter or Laplace filter which acts as a noise detector. If it is found noisy then we apply a general median filter as described in [2]. The main objective of both of these filters is to denoise the noisy image and restore the original one while keeping the high intensity details intact. The median filter is a major denoising technique used in image restoration technique as described below. The following subsection also describes the degradation process and some common noise models along with double derivative technique used for sharpening the images.

1.1. Image restoration

Preview

Image is a two dimensional operation or function of light intensities. As light falls on object and gets reflected back; hence we can see the object.

Mathematically it can be represented as

$Y(x, y) = I(x, y).R(x, y)$……. (1.1)

Here $I(x, y)$ = amount or intensity of light incident on object.

$R(x, y)$ = amount or intensity of light reflected from that object.

$Y(x, y)$ = resultant image in intensity (Unit:-Candela).

The ultimate goal of restoration technique is to improve image in some predefine sense. Although there are areas of overlap, image enhancement is largely a subjective process, while image restoration is for the most part an objective process. In restoration technique attempts to reconstruct or recover an image that has been degraded by using prior knowledge of the degradation phenomenon. Thus restoration techniques are oriented

toward modeling the degradation and applying the inverse process to recover the original image.

1.1.1. A Model for the Image Degradation Process

The general degradation process has been shown in figure:-1. In this chapter as a degradation function that, together with an additive noise term, operates on an input image f(x, y).Given g(x, y), some knowledge about the degradation function h, and some knowledge about the additive noise term η(x, y), the objective of restoration is to obtain an estimation of the original image.

The degraded image model in spatial domain is given by

$g(x, y) = h(x, y) * f(x, y) + \eta(x, y)$…….(1.2)

where h(x, y) is the spatial representation of the degradation function and the above function on"*" is called convolution operator. This convolution is multiplication in frequency domain. So it can be shown as

$G(u, v) = H(u, v) F(u, v) + N(u, v)$……... (1.3)

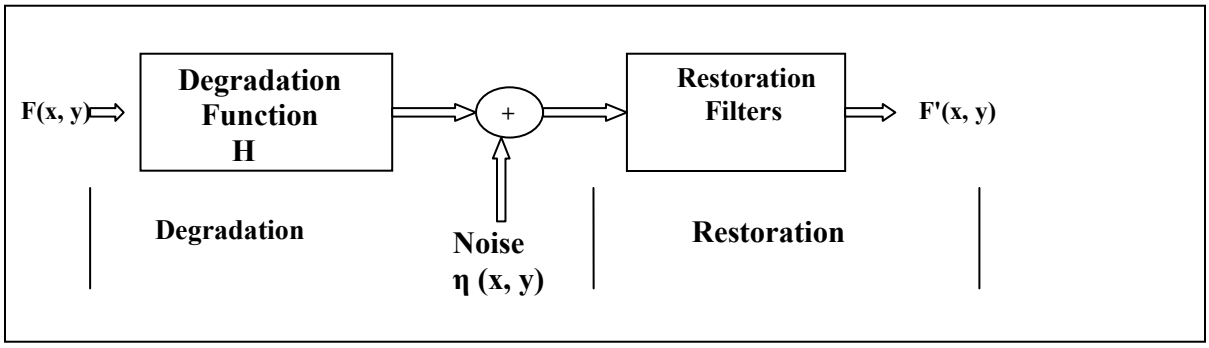


Figure 1.1 Figure of Image Degradation Model

Sometimes image are degraded by external noises as noises are additive in nature. The general degradation process of an image is given below

1.1.2. Some Noise Models

The main sources of noise in digital images arise during image acquisition (digitization) and transmission. The performance of imaging sensors is affected by a variety of factors, such as environmental conditions during image acquisition and by the quality of sensing elements themselves. For example sensor temperature, light levels are major factors affect the amount of noise in the resulting image. Images are corrupted during transmission principally due to interference in the channel used for transmission. For example; due to lightning effect or other atmospheric disturbance image transmitted by wireless network is corrupted. Images acquired by optical, electro-optical or electronic means are mostly degraded by the sensing environment. The degradations may be in the form of sensor noise.

Some of basic noise models are defined over spatial domain and are characterized by probability density function (PDF) as given below.

i. Impulsive Noise

Impulse noise shown in figure 1.2 is characterized by a noise spike replacing the actual pixel value. Impulse noise is further divided into two classes. Random valued impulse noise (RVIN) and salt and pepper noise (SPN). In RVIN, the impulse value at a particular pixel may be a random value between particular intervals. But in SPN the impulses are either the minimum value or the maximum value allowed in the intensity values. For example 0 and 255 in the case of 8-bit image.

$$p(z)\Big| = \begin{cases} p_a \text{.....when..} z = a \\ p_b \text{.....when..} z = b \\ 0\text{.....otherwise} \end{cases} \quad \text{.....(1.4)}$$

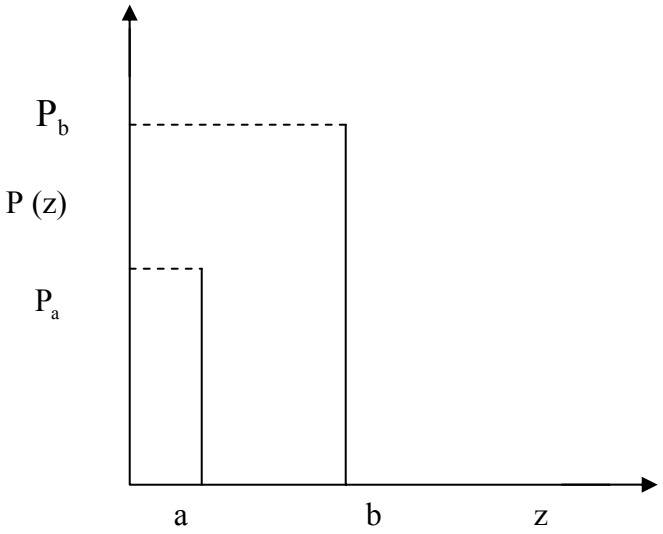


Figure 1.2 PDF of Impulsive Noise

ii. Uniform Noise

It is another common noise model shown in figure. In this case the noise can take on values in an interval [a, b] with uniform probability. The PDF of uniform noise is given by

$$P(z) = \begin{cases} \frac{1}{b-a} \text{....when..} a \le z \le b \\ 0\text{.....otherwise} \end{cases} \text{.....(1.5)}$$

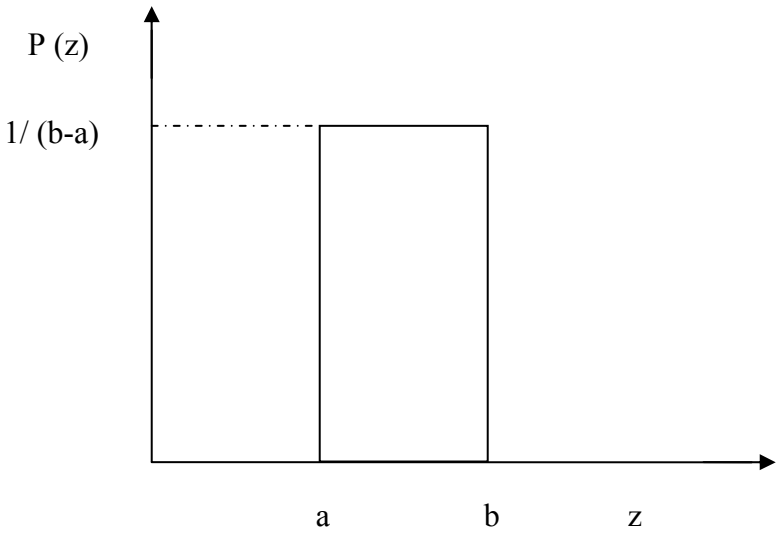


Figure 1.3 PDF of a Uniform Noise

iii. Gaussian noise

Due to mathematical tractability in both the spatial and frequency domains Gaussian noise models are used in frequent in practice**.** The PDF of Gaussian random values z is given by

$$P(z) = \frac{1}{\sqrt{2\Pi\sigma}} e^{-(z-\mu)^2/2\sigma^2} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (1.6)$$

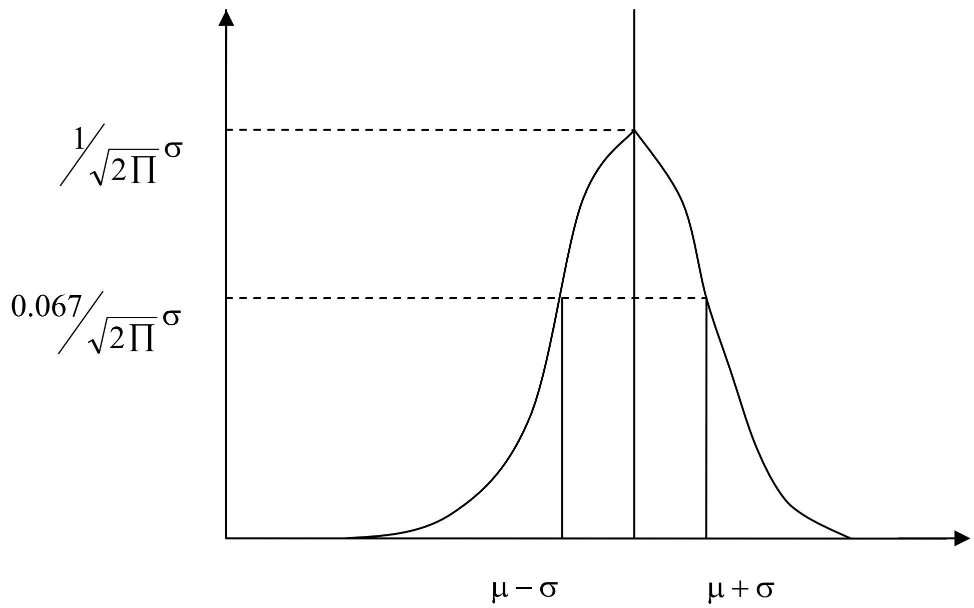


Figure 1.4 PDF of Gaussian Noise

1.1.3 Denoising Techniques

Another type of image degradation is due to additive noise. Random values get added to the intensity values. Denoising involves the application of some filtering technique to get the true image back. The denoising algorithms [1] vary greatly depending on the type of noise present in the image. Each type of image is characterized by a unique noise model. Each noise model corresponds to a probability density function which describes the distribution of noise within the image.

When the only degradation present in an image is noise, then equation (1.2) becomes

$g(x, y) = f(x, y) + \eta(x, y)$ ……..(1.7)

The Fourier transform of 1.4 will be

$G(u, v) = F(u, v) + N(u, v)$ …… (1.8)

1.1.4. Spatial Filtering

Denoising techniques exist in both spatial domain as well as frequency domain.

Spatial Filtering is preferred when only additive noise is present. The different classes [9] of filtering techniques exist in spatial domain filtering.

- Mean Filters
- Order-Statistics Filters
- Adaptive Filters

Mean filters

In this method the middle pixel value of the filter window is replaced with the arithmetic mean or geometric mean or harmonic or contra-harmonic mean of all the pixel values within the filter window. A mean filter simply smoothes local variations in an image. Noise is reduced as a result of this smoothening, but edges within the image get blurred.

Examples: - Arithmetic Mean Filter, Geometric Mean Filter, Contra harmonic Mean Filter

Order statistics filter

Median Filter comes under the class of order-statistics filters. Response of Order-statistics filters is based on ordering the pixels contained in the filter window. Median filter replaces the value of a pixel by the median of the gray levels within the filter window. Median filters are particularly effective in the presence of impulse noise.

Examples: - Median filters, Max-min filters, mid point filter

Adaptive Filters

Adaptive filters change its behavior based on the statistical characteristics of the image inside the filter window. Adaptive filter performance is usually superior to non-adaptive Counterparts. But the improved performance is at the cost of added filter complexity. Mean and variance are two important statistical measures using which adaptive filters can be designed. For example if the local variance is high compared to the overall image variance, the filter should return a value close to the present value. Because high variance is usually associated with edges and edges should be preserved.

1.1.5. Sharpening filters

The main function of sharpening is to highlight the fine detail in an image or enhance the detail that has been blurred. It is used to preserve high intensities value pixel value like edge pixel. In the spatial domain image blurring happens by pixel averaging in a neighborhood. Actually averaging operation is nothing but an integration operation; so sharpening should be accomplished with differentiation operation. In the image at edge there is higher discontinuity; so taking differentiation enhances the edges. Double derivative filter or Laplacian filter is also a good example of sharpening filters.

A basic mathematical definition of first order derivative is

$$\frac{\delta f}{\delta x} = f(x+1) - f(x) \quad \text{..... (1.9)}$$

And second order derivative

$$\frac{\delta^2 f}{\delta x^2} = f(x+1) + f(x-1) - 2\,f(x) \quad \text{..... (1.10)}$$

1.1.6. Double Derivative Filter-The Laplacian

The Derivative or gradient or difference taken at boundary or edge results a sharp contrast. First order derivative produces thicker edges in an image and so second order produces more detail information. The main principle of using first order is to extract the edges in an image. But second orders are mostly preferred as it produces even better

enhancement results. Usually these filters along with other enhancement or restoring filters yield impressive results.

The Laplacian for a function image f(x, y) of two variables x , y defined as

$$\nabla^2 f = \frac{\delta^2 f}{\delta x^2} + \frac{\delta^2 f}{\delta y^2} \quad \text{.....(1.11)}$$

In X-direction

$$\frac{\delta^2 f}{\delta x^2} = f(x+1, y) + f(x-1, y) - 2f(x, y) \text{..... .....(1.12)}$$

In y-direction

$$\frac{\delta^2 f}{\delta y^2} = f(x, y+1) + f(x, y-1) - 2f(x, y) \text{..... ..... (1.13)}$$

$$\nabla^2 f = f(x+1, y) + f(x-1, y) + - f(x, y+1) + f(x, y-1) - 4f(x, y) \text{..... (1.14)}$$

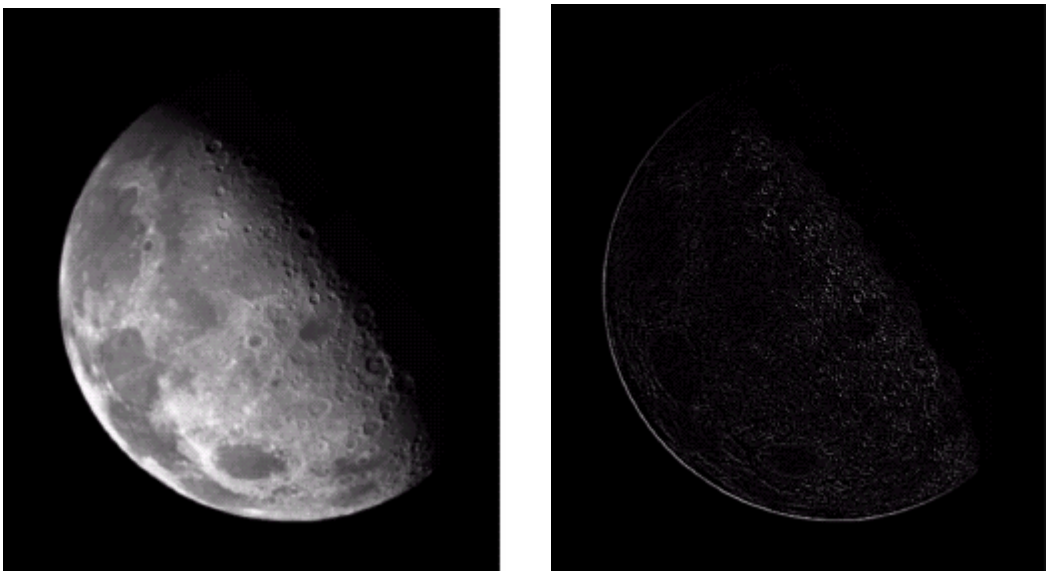


Figure1.5 (a) The Original Image and (b) Filtered Image Applying Laplacian

As we can see from above figure Laplacian of the image produces a sharp edges; it is useful when we need to preserve the sharp edges in a noisy image.

1.2. Low power VLSI Design

The VLSI low power design problems can be broadly classified into two: analysis and optimization. Analysis problems are concerned about the accurate estimation of the power or energy dissipation at different phases of the design process. Optimization is a process of generating the best design given an optimization goal, without violating design specification. Analyze techniques also serve for design optimization but major criteria to be considered are the impact of circuit delay which affects throughput and performance of circuit. Other factors are design cycle time, reliability, testability, quality, reusability. Power efficiency cannot be achieved without yielding to one or more of these factors because they are complementary to each other.

Performance of processors has been growing at an exponential rate, doubling every 18 to 24 months which obeys Moore's' law. However, at the same time the power dissipated by these processors has also been growing considerably. Although the rate of growth of power dissipation is perhaps not quite proportional to performance and size (the number of transistors) it still has led to processors which dissipated more than 50W. For such processors cooling becomes an absolute necessity and at high power dissipation level this is even difficult and expensive. If this trend continues processors will soon dissipate hundreds of watts, which would be unacceptable in most systems. Thus there is great interest in understanding how to continue increasing performance without increasing the power dissipation. The increase in device density and operating frequency of various transistors increases power dissipation also. The current demands for portable electronics devices having features like smaller size, lighter weights and more durability need good energy source or battery. The recent battery technology using Li^+ or Ni-Cd has not improved much over this area to overcome the power problem.

Another factor for low power chips is the increased market demand for portable consumer electronics powered by batteries. The craving for smaller, lighter and more durable electronic products indirectly translates to low power requirements. The power dissipation of high performance computing system or microprocessors is now approaching a considerable amount (in terms of watts).power dissipation has direct impact on the packaging cost of the chip and the cooling cost of the system. Some of

personal computer's CPUs require cooling fans directly mounted on the chip due to high power dissipation.

Another major demand for low power chips and systems comes from environmental concerns. A study by American Council for an energy efficient economy estimated that office equipment doubles of in the period 1993 to 2000 and hence fastest–growing electricity load in commercial sector.

The major power sources and the solution to reduce power are given below.

1.2.1. Power Dissipation Sources

In CMOS circuits, the main contributions to the power consumption are from short-circuiting current, leakage current, and switching currents [18, 21]. In the following subsections, we have introduced them separately.

i. Short-Circuit Power

In a static CMOS circuit, there are two complementary networks: p-network (pull-up network) and n-network (pull-down network) as shown in figure 1.6. The logic functions for the two networks are complementary to each other. Normally when the input and output state are stable, only one network is turned on and conducts the output either to power supply node or to ground node and the other network is turned off and blocks the current from flowing. Short-circuit current exists during the transitions as one network is turned on and the other network is still active. For example, the input signal to an inverter is switching from 0 to V_{dd}. During this transaction, there exists a short time interval where the input voltage is larger than V_{tn} but less than $V_{dd} - |V_{tp}|$. During this time interval, both PMOS-transistor (p-network) and NMOS-transistor (n-network) are turned on and the short-circuit current flows through both kinds of transistors from power supply line to the ground.

The exact analysis of the short-circuit current in a simple inverter [18] is complex; this is analyzed by SPICE simulation. It is observed that the short-circuit current is proportional to the slope of input signals, the output loads and the transistor sizes [19]. The short-

circuit current consumes typically less than 10% of the total power in a "well-designed" circuit [19].

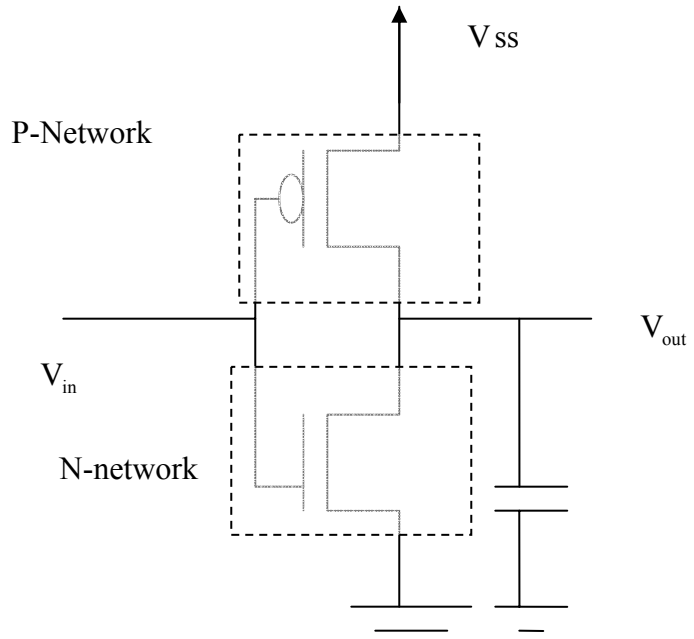


Figure 1.6 Figure of CMOS Inverter

ii. Leakage Power

Leakage currents are due to two sources: one from the currents that flow through the reverse biased diodes (reverse biased PN-Junction current), the other from the currents that flow through transistors that are non-conducting (subthreshold channel conduction current).

The leakage currents depends upon temperature and are proportional to the leakage area and exponential of the threshold voltage.Sub threshold leakage and reverse-biased junction leakage, both increases dramatically with temperature and are independent of the operating voltage for a given fabrication process.

The leakage current is in the order of pico-Ampere, but it increases as the threshold voltage is reduced. In some cases, like large RAMs, the leakage current is one of the main concerns. The leakage current is currently not a severe problem in most digital designs. However, the power consumed by leakage current can be as large as the power consumed by the switching current for 06.0 *m*μ technology. The usage of multiple

threshold voltages can reduce the leakage current in deep-submicron technology. Leakage current is difficult to predict, measure or optimized.

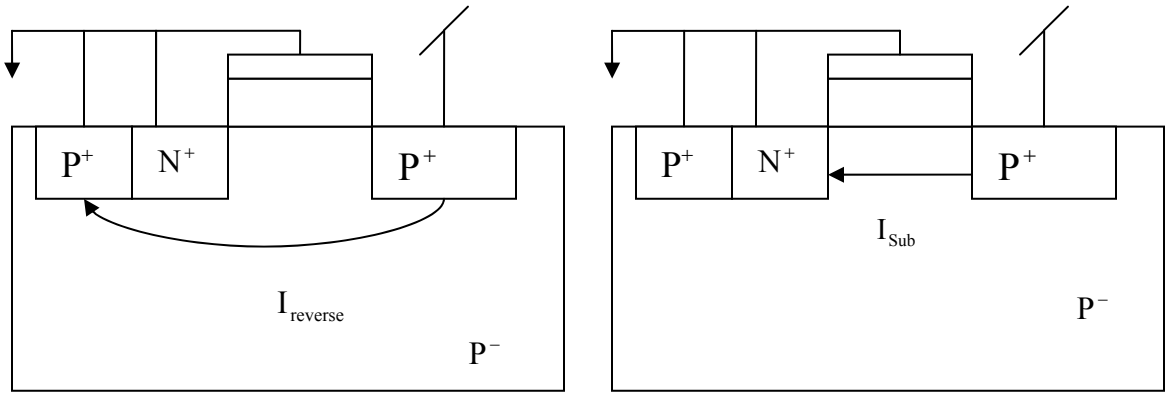


Figure 1.7 Leakage Current types (a) Reverse Biased Current (b) Sub-threshold Current

Generally, leakage current serves no useful purposes, but some circuits do exploit it for i ntended operations, such as power-on reset signal generation. The leakage power problem mainly appears in very low frequency circuits or ones with "sleep modes" where dynamic activities are suppressed.

iii. Switching Power

The switching currents are due to the charging and discharging of node capacitances. The node capacitances mainly include gate, overlapping, and interconnection capacitances. The power consumed by switching current [9] can be expressed as

$$P = \alpha C_L f V_{dd}^2 / 2 \quad \text{..... (1.15)}$$

Where α is the switching activity factor, C_L is the load capacitance, f is the clock frequency, and V_{dd} is the supply voltage. The above equation (1.15) shows that the switching power depends on a few quantities that are readily observable and measurable in CMOS circuits. It is applicable to almost every digital circuit and hence provides guidelines for the low power design. The power consumed by switching current is the

dominant part of the power consumption. Reducing the switching current is the focus of most low power design techniques. For large capacitance circuits, reduction of the frequency is the best way to reduce the switching power. The use of different coding methods, number representation systems, continuing sequences and data representations can directly alter the switching frequency of the design, which alters the switching power. The best method of reducing switching frequency is to eliminate logic switching that is not necessary for computation.

1.2.2. Low Power Techniques

Low power techniques have been discussed at various levels of abstractions: system level, algorithm and architecture level, logic level, circuit level, and technology level [13,14]. Figure 1.8 shows some examples of techniques at the different levels. In the following sections; an overview for different low power techniques has been described in detail. This is organized on the basis of abstraction level.

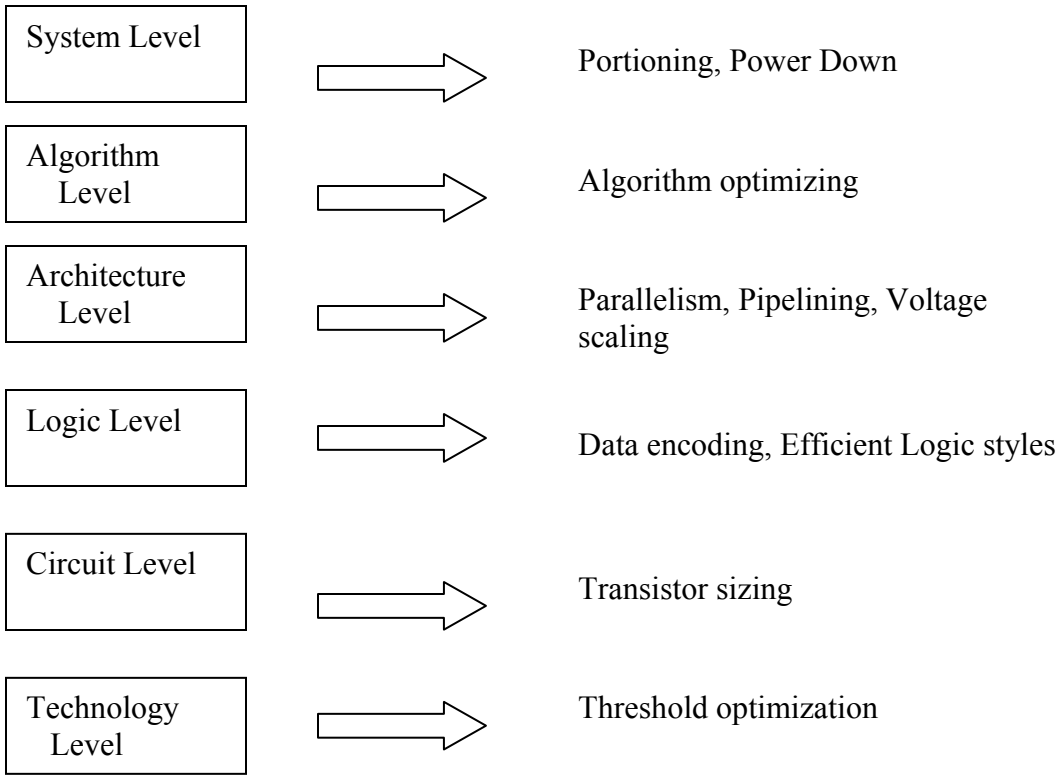


Figure 1.8 Low-power Design Methodology at Different Abstraction Levels

i. System Level

A system typically consists of both hardware and software components, which affect the power consumption. The system design includes the hardware/software partitioning, hardware platform selection (application-specific or general-purpose processors), resource sharing (scheduling) strategy, etc.

At the system level, the faster code and instruction level optimization using different software tool power reduction can be possible. The power-down and clock gating are two of the most used low power techniques at system level. The non-active hardware units are shut down to save the power. The clock drivers, which often consume 30-40% of the total power consumption, can be gated to reduce switching activities as illustrated in figure 1.9. In clock gating technique the unnecessary blocks are remained powered off when they are not doing any operation.

The power-down can be extended to the whole system. This is called sleep mode and widely used in low power processors. The system is designed for the peak performance. However, the computation requirement is time varying. Adapting clocking frequency and/or dynamic voltage scaling to match the performance constraints is another low power technique. The lower requirement for performance at certain time interval can be used to reduce the power supply voltage. This requires either feedback mechanism (load monitoring and voltage control) or predetermined timing to activate the voltage down-scaling.

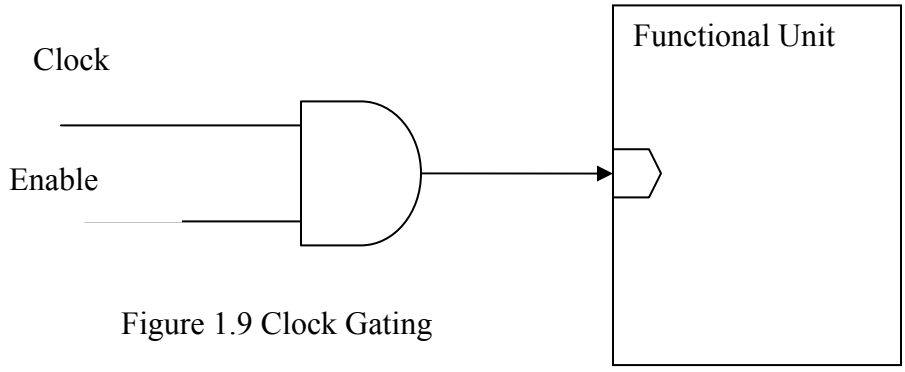


Figure 1.9 Clock Gating

Asynchronous design of the circuit can also be used as another low power designing technique. The asynchronous designs have many attractive features, like non-global

clocking, automatic power-down, no spurious transitions, and low peak current, etc. It is easy to reduce the power consumption further by combining the asynchronous design technique with other low power techniques, for instance, dynamic voltage scaling technique, as shown in the following figure 1.10.

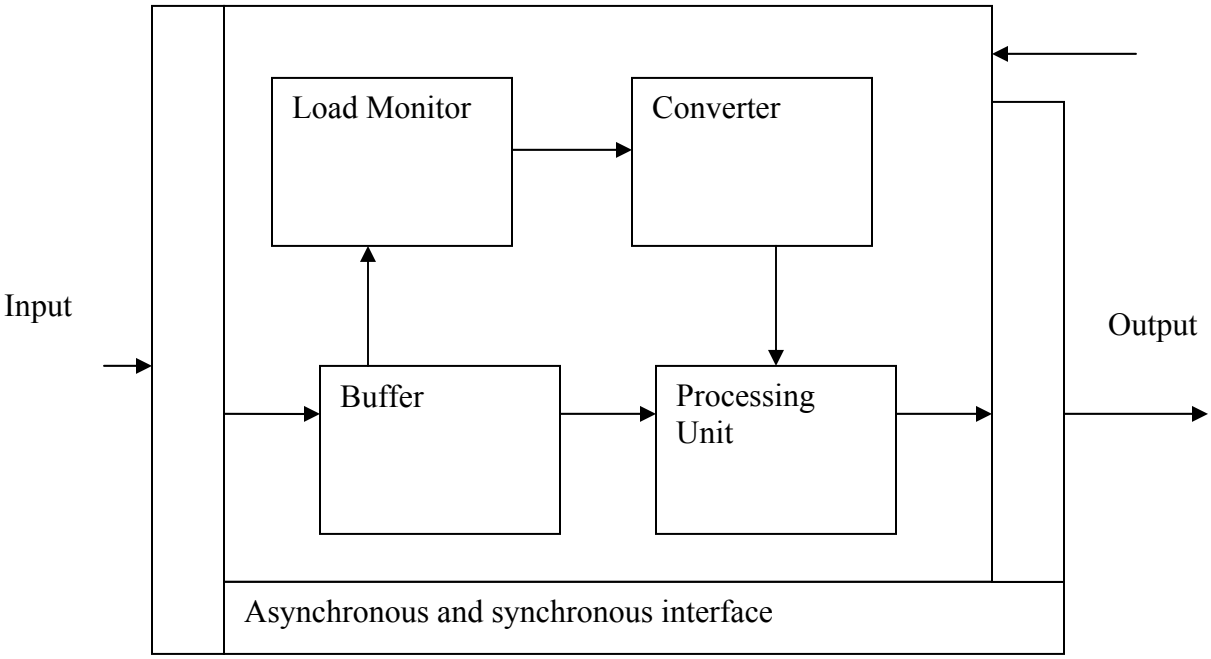


Figure 1.10: Asynchronous Design with Dynamic Voltage Scaling.

ii. Algorithm Level

The algorithm selection plays a vital role in power reduction as it decides various usage resources of system such as communication, storage and delay. The complexity of the algorithm decides the optimum usage of resources. As example quick sort algorithm for stand alone computer is fastest sorting algorithm but it takes extra space compared to shell sort and heap sort. Lesser complexity means lesser number of operations and hence lesser switching activities and lesser power consumption.

Another algorithmic level technique is algorithmic transformation. The loop unrolling technique is a one of this transformation that aims to enhance the speed. This technique can be used for reducing the power consumption. The With loop unrolling, the critical path can be reduced and hence voltage scaling can be applied to reduce the power consumption.

Example of loop un-rolling:

Y = ((x1 + x2)*x3 + x4) + x5……. (1.16)

Y=(x1*x3+x2*x3+x4) + x5…… (1.17)

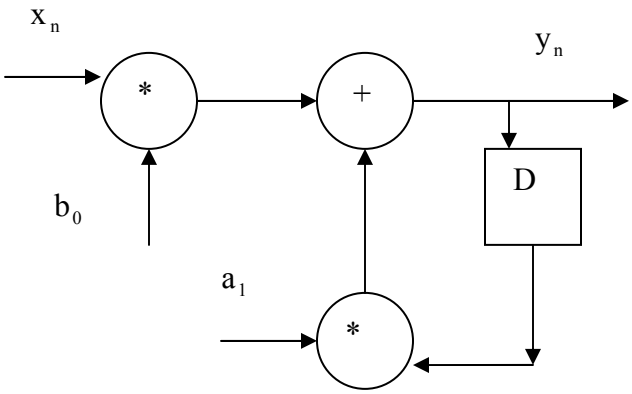


Figure 1.11 The Normal Operation Using Addition and Multiplication

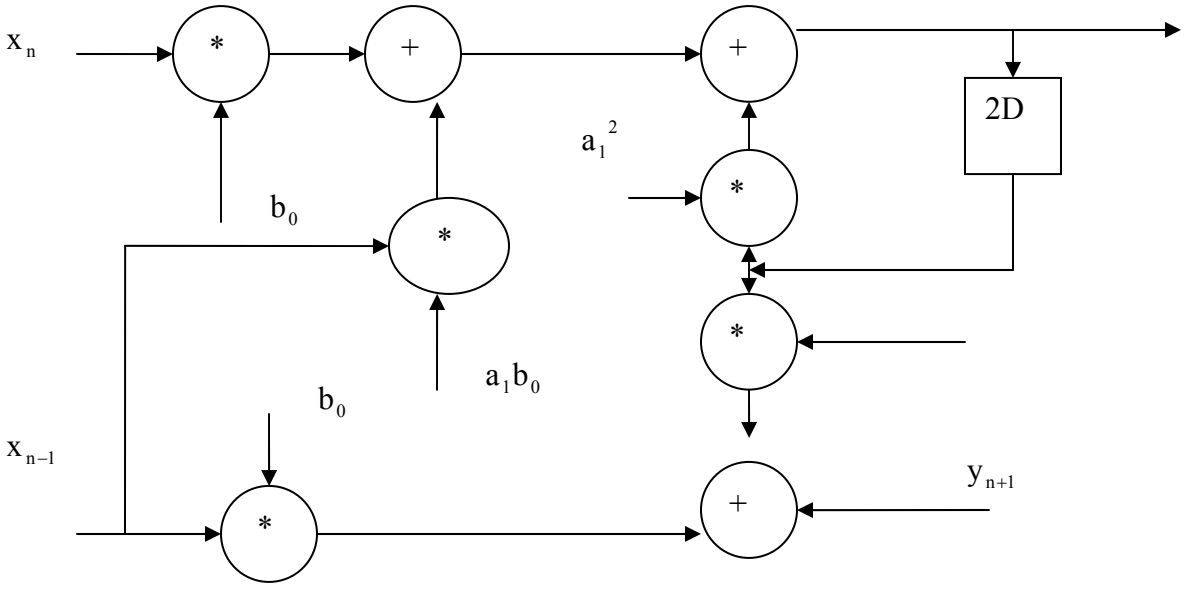


Figure 1.12 The Operation Using Loop Unrolling Technique

In figure 2.5, the unrolling reduces the critical path and gives a voltage reduction of 26% [18 – 23]. This reduces the power consumption with 20% even the capacitance load is increases with 50% [13]. Furthermore, this technique can be combined with other techniques at architectural level, for instance, pipeline and interleaving, to save more power. In some cases, like digital filters, the faster algorithms, combined with voltage-scaling, can be used for energy-efficient applications [18].

iii. Architecture Level

According to the selection of the algorithm, the architecture can be determined for the given algorithm. From equation (2.1) we can say that, an efficient way to reduce the dynamic power consumption is the voltage scaling. When supply voltage is reduced, the power consumption is reduced. However, this increases the gate delay. To compensate the delay, low power techniques like parallelism and pipelining [14] architectures were used.

The use of two parallel datapath is equivalent to interleaving of two computational tasks. A datapath to determine the largest number of C and (A + B) is shown in figure 1.13. It requires an adder and a comparator. The original clock frequency is 40 MHz.

In order to maintain the throughput while reducing the power supply voltage, we use a parallel architecture. The parallel architecture with twice the amount of resources is shown in figure 1.14. The clock frequency can be reduced to half, from 40 MHz to 20 MHz since two tasks are executed concurrently. This allows the supply voltage to be scaled down from 5 V to 2.9 V Since the extra routing is required to distribute computations to two parallel units, the capacitance load is increased by a factor of 2.15 .

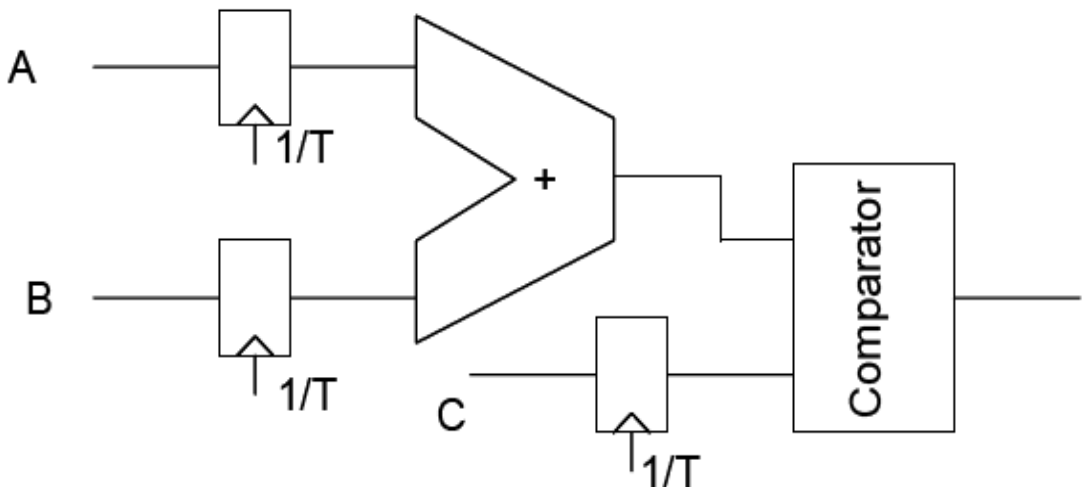


Figure 1.13 Original Data path.

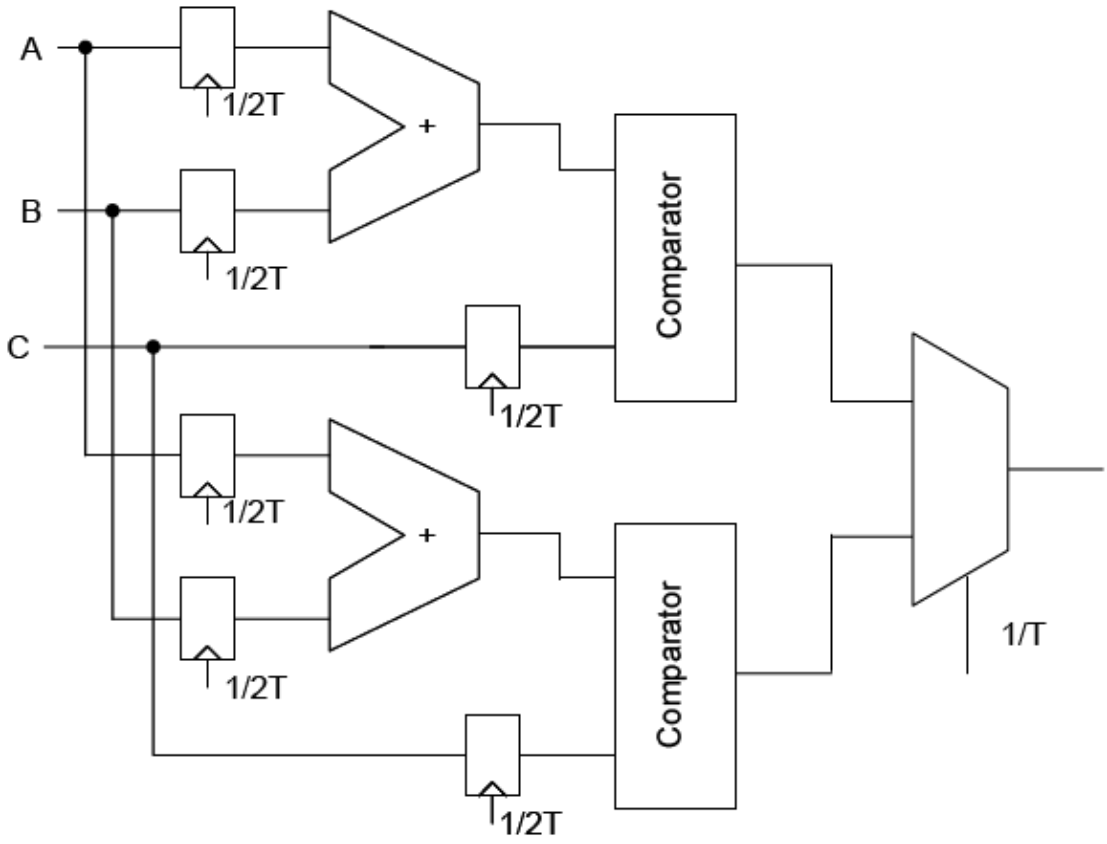


Figure 1.14 Parallel Implementation

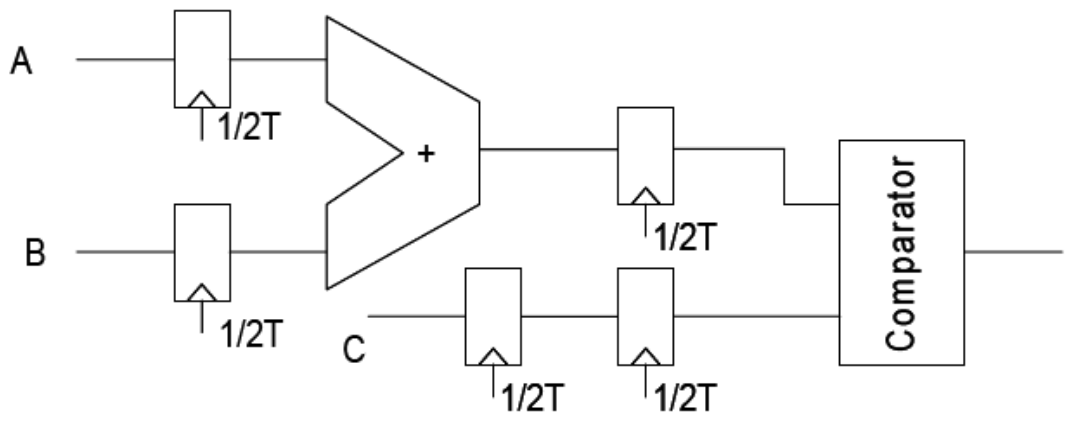


Figure1.15 Pipeline Implementation

Pipelining is another method for increasing the throughput. By adding a pipelining buffer or register after the adder in figure 1.15, the throughput can be increased and also save power. The power is calculated in three data paths using 1.15 equation and compared; which shows in pipeline architecture it consumes lesser power.

Main advantage of pipelining is the low area overhead in comparison with using parallel data paths. Another benefit is that the amount of glitches can be reduced. However, since the delay increases significantly as the voltage approaches the threshold voltage and the capacitance load for routing and/or pipeline registers increases, there exists an optimal power supply voltage. Reduction of supply voltage lower than the optimal voltage increases the power consumption.

iv. Logic Level

The low power techniques at the logic level, however, focus mainly on the reduction of switching activity factor by using the signal correlation and the node capacitances.

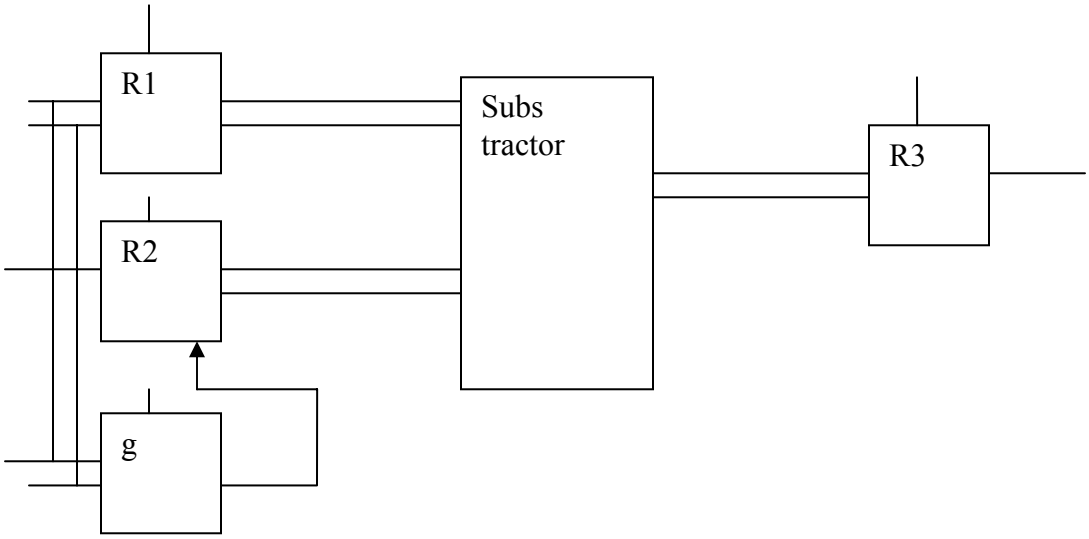


Figure 1.16 A Precomputation Structure for Low Power.

Precomputation [18] uses the same concept to reduce the switching activity factor: a selective precomputing of the output of a circuit is done before the outputs are required, and this reduces the switching activity by gating those inputs to the circuit. As shown in figure 2.9, the input data is partitioned into two parts, corresponding to registers R_1 and R_2. One part, R_1, is computed in precomputation block g, one clock cycle before the computation of A. The result from g decides gating of R_2. The power can then be saved by reducing the switching activity factor in A.

Lets consider a comparator as an example of precomputation for low-power. The comparator takes the MSB of the two numbers to register R_1 and the others to R_2. The comparison of MSB is performed in g. If two MSBs are not equal, the output from g

gated the remaining inputs. In this way, only a small portion of inputs to the comparator's main block A (subtracter) is changed. Therefore the switching activity is reduced Gate reorganization [14] is another technique used to restructure the circuit. This can be decomposition a complex gate to simple gates, or combines simple gates to a complex gate, duplication of a gate, deleting/addition of wires. The decomposition of a complexgate and duplication of a gate help to separate the critical and non-critical path. Which reduce the size of gates in the non-critical path, as a result reduces the power consumption. In some cases, the decomposition of a complex gate increases the circuit speed and gives more space for power supply voltage scaling. The composition of simple gates can reduce the power consumption. The complex gate can reduce the charge/discharge of high-frequently switching node. The deleting of wires reduces the circuit size as a result, reduces the load capacitance.

Traditionally, the logic coding style is used for enhancement of speed performance. Careful choice of coding style is important to meet the speed requirement and minimize the power consumption. This can be applied to the finite state machine, where states can be coded with different schemes.

A bus is the main on-chip communication channel that has large capacitance. As the on-chip transfer rate, increases, the use of buses contributes with a significant portion of the total power. Bus encoding is a technique to exploit the property of transmitted signal to reduce the power consumption.

2.2.5 Circuit Level

At the circuit level, the power saving techniques is quite limited if compared with the other techniques at higher abstract levels. However, this cannot be ignored. The power savings can be significant as the basic cells are frequently used. A few percents improvement for D flip-flop can significantly reduce the power consumption in deep pipelined memory systems.

In CMOS circuits, the dynamic power consumption is caused by the transitions. Spurious transitions typically consume between 10% and 40% of the switching activity power in

the typical combinational logic. In some cases, like array multipliers, the amount of spurious transitions is large. To reduce the spurious transitions, the delays of signals from registers that converge at a gate should be roughly equal. This can be done by insertions of buffers and device sizing [14]. The insertions of buffer increase the total load capacitance but can still reduce the spurious transitions. This technique is called path balancing.

Many logic gates have inputs that are logically equivalent, i.e., the swapping of inputs does not modify the logic function of the gate. Some examples of gates are NAND, NOR, XOR, etc. However, from the power consumption point of view, the order of inputs does effect the power consumption. Let's consider the figure 1.10, the A-input, which is near the output in a two-input NAND gate, consumes less power than the B-input closed to the ground with the same switching activity factor.

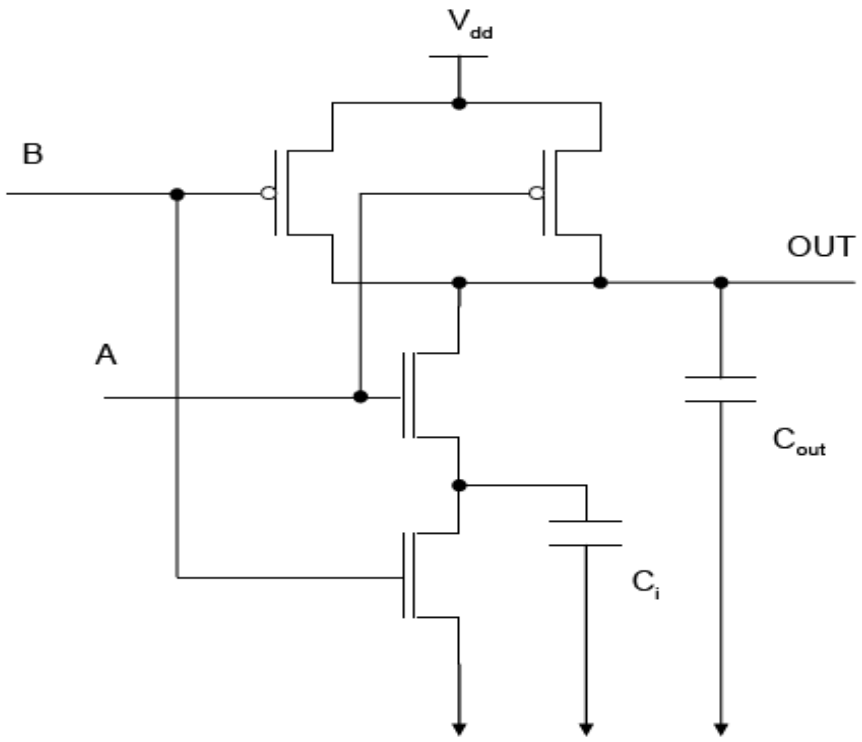


Figure 1.17 A two Input NAND Gate.

Pin ordering is to assign more frequently switching input pins near to the output node, which will consume less power. In this way, the power consumption will be reduced without cost. However, the statistics of switching activity factors for different pins must be known in advanced and this limits the use of pin ordering [13].

Different logic styles have different electrical characteristics. The selection of logic style affects the speed and power consumption. In most cases, the standard CMOS logic is used for speed and power trade-off. In some cases other logic styles, like complementary pass-transistor logic (CPL) [15] is efficient.

Transistor sizing affects both delay and power consumption. Generally, a gate with smaller size has smaller capacitance and consumes less power. To minimize the transistor sizes and meet the speed requirement is a trade-off. Typically, the transistor sizing uses static timing analysis to find out those gates (whose slack time is larger than 0) to be reduced. The transistor sizing is generally applicable for different technologies.

1.3. Performance Measures for low power deign

There are several measures for evaluating the performance of the low power VLSI design;they are discussed as below

Power:-The total power consumption and dissipation during transition or switching of 0 to 1 or 1 to 0.The main sources of power dissipation are leakage power, switching power and static power.

Area: - The total area of whole design; It mainly comprises of total number of cells, gates and flip flops used. For FPGA implementation the total chip area comprises of logical blocks and Look Up Tables (LUTs).

Speed: - It is the measure of rate at which the designed IC produces an output .It depends upon various factors such as gate delays, clock delays, and wire delays and also critical path of design.

1.4. Problem definition:

A low power VLSI design of selective median filter circuit has been our primary goal. This problem has been divided into two phases namely

i. Design of an algorithm for median filter with double derivative filter, (ii) the low power VLSI design of this filter

The main objective is to remove spike or salt and pepper noise from the noisy image while preserving the sharp edges. This is the main reason for why the double derivative filter has been used. Designing complete hardware for Median filter with sorting networks is computationally expensive. The main power consuming component here is sorting circuit and it has been repeatedly used in simulation. Here we have used majority voting logic which uses bit level computation. Taking 3 x 3 windows the median value is calculated using MVL logic. Again preserving edges is also another goal because noise suppressing and edge preserving are side by side affairs. The whole design which consumes lesser and lesser power is a desirable factor.

In VLSI system design the power consumed by switching activities of various gates given by equation $P=CV^2f$ as mentioned in 1.15 has to be analyzed and optimized for low power implementation. By minimizing any of parameters either v-supply voltage, capacitance (diffusion and depletion due to various components of CMOS), frequency of clock of each subsystem we can reduce the power consumption. For example: - If we can minimize switching activity F and C factors of switching power can be reduced. Optimizing circuit also we can minimize Capacitance and hence overall power. In our thesis logic level optimization has been tried in behavior code of forty one 8-bit sorting networks.

1.5. Motivation:

In image processing noise removal plays a vital role as it eases the object recognization and object identification problem. For this purpose median filter is one of the noise suppressing filters which is most commonly used. Median filters are implemented by software in most of the systems .When real time image processing applications, like robot visions and visual feedback control are concerned, the minimum latency in the system's response is of paramount importance, making software approaches un acceptable. How ever it is computationally expensive and hence consumes a lot of power.

There has been tremendous effort on sorting network design and comparator design at circuit level [1, 3]. VLSI median filters have been developed using both digital and analog technologies. There has been a number of application based on median filter in the field of image processing and remote sensing such as medical imaging, satellite imaging, Geographic image surveillance, Seismographic analysis. In image processing field as it is the most commonly used impulsive or spike noise removal filter; so on chip median filter is now days very cost effective. Therefore a lot of VLSI design both Analog and Digital have been implemented and there have been a lot of efforts to build faster and faster sorters.

The fast growing demand for System on Chip (SOC) design requires much more developed but complicated techniques. But there have been heavy power consumptions and dissipations supporting this type of application. So currently there is a heavy demand for reducing this power factors with same functioning capabilities. But in that context battery or power optimizing technology has not grown up to that mark. Most of these products include embedded microprocessors, DSPs and ASICs. Therefore low power design of any VLSI design is now a days a challenging task.

For low power design there are various architecture level design like systolic, bit level design [4-6] have been proposed which are faster in design but in terms of power there have been a little concern . For low power application different level of optimization has been done in VLSI design process as discussed in section 1.2.

1.6 Chapter-wise Organization of the thesis:

The rest of thesis is organized as follows:

(1)Chapter-2: This chapter briefly introduces with the related works of hardwire design median filter, sorting network and their low power designs**.**

(2) Chapter-3: This chapter presents the design of a 16 bit general processor.

(3) Chapter-4: This chapter presents a design of selective median filter and its low power implementation

(4) Chapter-5: In this chapter conclusion has been made and some further research scopes are suggested.

Literature Survey

A double derivative impulsive noise detector has been proposed in [2].In this paper a double differential filter has been applied on each target pixel to check whether that pixel is a noisy one or not. Usually applying double derivative filter restores the sharp edges and high pixel values in the remaining image. Using a 3 x 5 window double derivative or Laplacian filter is applied and the filtered value is compared with threshold value. If it is less then median filter is applied. It concludes that the test pixel is of high intensity value and hence there is less chance of corruption. The complete mechanism has been given in figure 2.1.

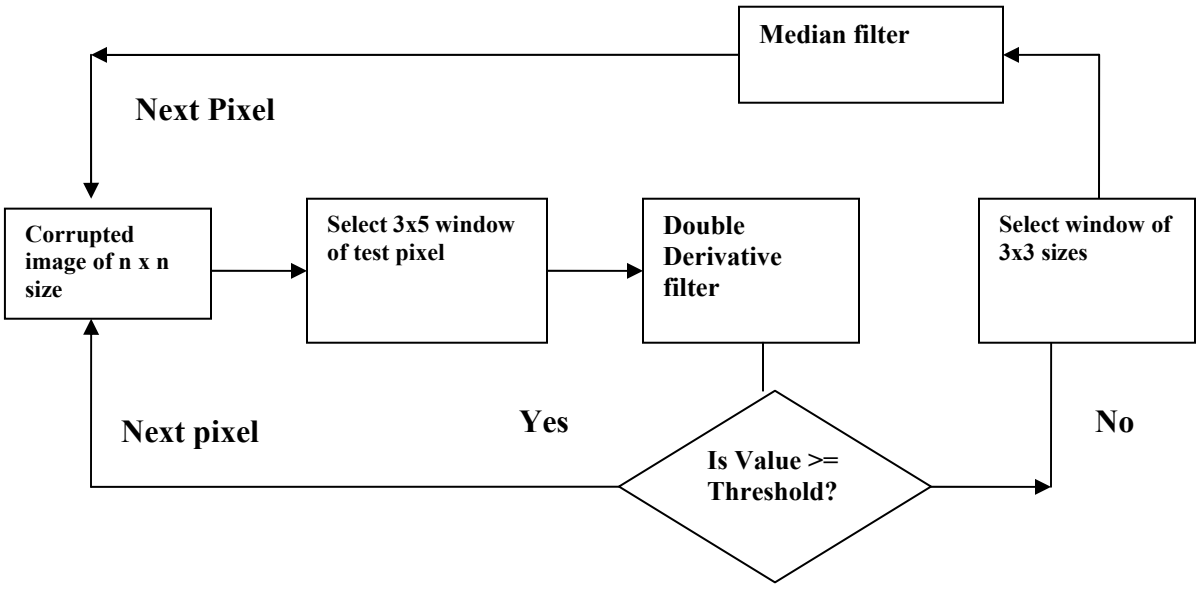


Figure 2.1 Block diagram of Selective Median Filter

Depending upon threshold value the result of designed filter varies. Hence this threshold value can be selected in such a way that it will be suitable to each image depending upon their intensity or pixel values of image and so more optimized result can be possible.

2.1. A Survey on Hardware implementation for Median Value Evaluation

Mustafa Karaman et al [3] present a general purpose median filter in the form of two single-chip median filters: one for extensible and one real time. They have found experimentally that the exact median of elements, in a window size w = 9 with arbitrary word length L, can be found by using only one extensible median filter chip. This can be also extended to any length of L. For w > 9 with arbitrarily L the number of chips required to find the exact medians is no more than the smallest greater integer of $(w/9)^2$.Their simulation showed on a 40 MHz chip the filter can filter 30/L mega medians per second. On the other side real-time median filter with fixed size w=9 and L=8 can generate 50 mega median per second on 50MHz clock.

In this work an odd/even transposition sorting network has been presented which is a pipelined modular structure consisting of w compare and swap stages. The extensible median filter consists of

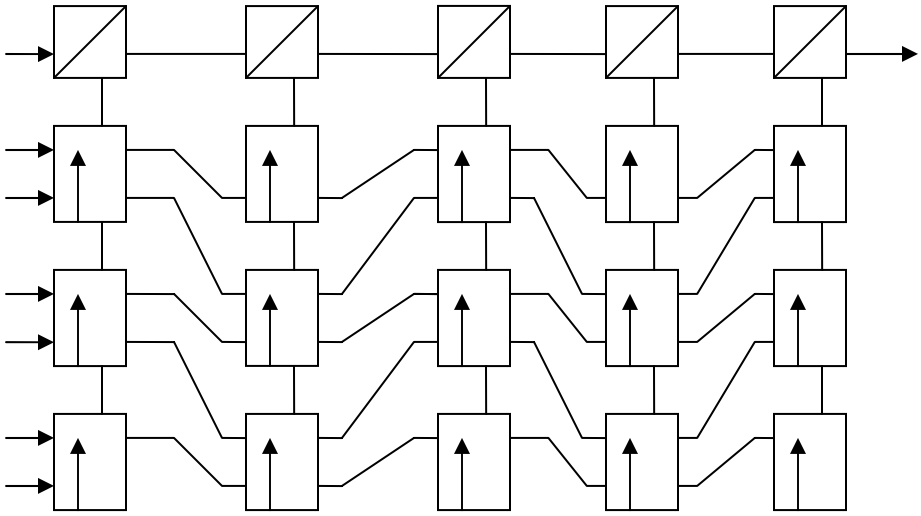


Figure 2.2 Diagram of Sorting Network for Five 3-bit Inputs

Each compare and swap unit as shown in figure 2.3 has three states such as reset, swap or pass and equal/not equal. Two bits A_i and B_i from two different input sets are taken and compared and if $A_i > B_i$ then pass the inputs unaltered to next level, if $A_i < B_i$ then swap the values and pass to next stage. By doing at each clock cycle after initial delay we will get median output as shown in sorting network (figure 2.2).

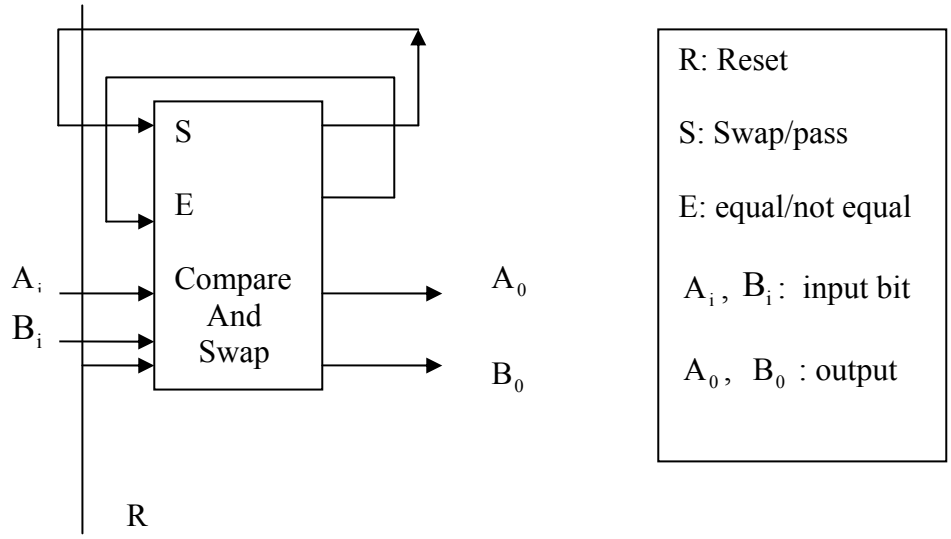
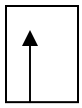


Figure 2.3 Diagram of Compare and Swap Unit



➔ comparator and swap unit

➔ One bit delay unit

The VLSI implementations of algorithms consist of circuit designs and generation of layouts using 3-μm CMOS technology. It is completely pipelined as first data set is generating output rest four data sets are in network. The circuits are designed using CAD tools and layout is designed in SPICE tool.

Hideo Yamasaki et al in [4] presented a low power and high speed mixed signal VLSI median filter that has used majority voting circuits. In this design an eight-bit 41-input median filter circuit was designed and fabricated 0.35 μm technology. It has used Binary search algorithm which is a bit-comparison based technique and it is compatible to direct hardware implementation.

The mechanism that has been used is given below:

Majority Voting Technique:-

Example:-

1101 ->	1**1**01* ->	11**1**1* ->	111**1**
1001 ->	1**0**01 ->	10**0**1 ->	100**1**
1010 ->	1**0**10 ->	10**1**0* ->	101**1**
:	:	:	:
0010* ->	0**0**00 ->	00**0**0 ->	000**0**
0001* ->	0**0**00 ->	00**0**0 ->	000**0**
---	---	---	---
1xxx	10xx	100x	1001

Above example explains the median detection by the binary search algorithm taking the search from five 4-b input data as an example. In the first place, majority voting for the MSB's of all data is carried out. Namely, input data are divided into two groups, one with MSB of "0" and the other with MSB of "1", and the majority group wins. Since the median value is necessarily in the winner group, the MSB of the median is identical to the winner bit ("1" in this example). Secondly, the remaining bits of the data in the loser group are all converted to the loser MSB ("0" in this case) in order to propagate the information that losers are all smaller (or larger) than the median value.(In the present example, all the data in the losers group are converted to "0000".) Then the majority voting for the second bits is carried out in a similar way, and the winner bit yields the

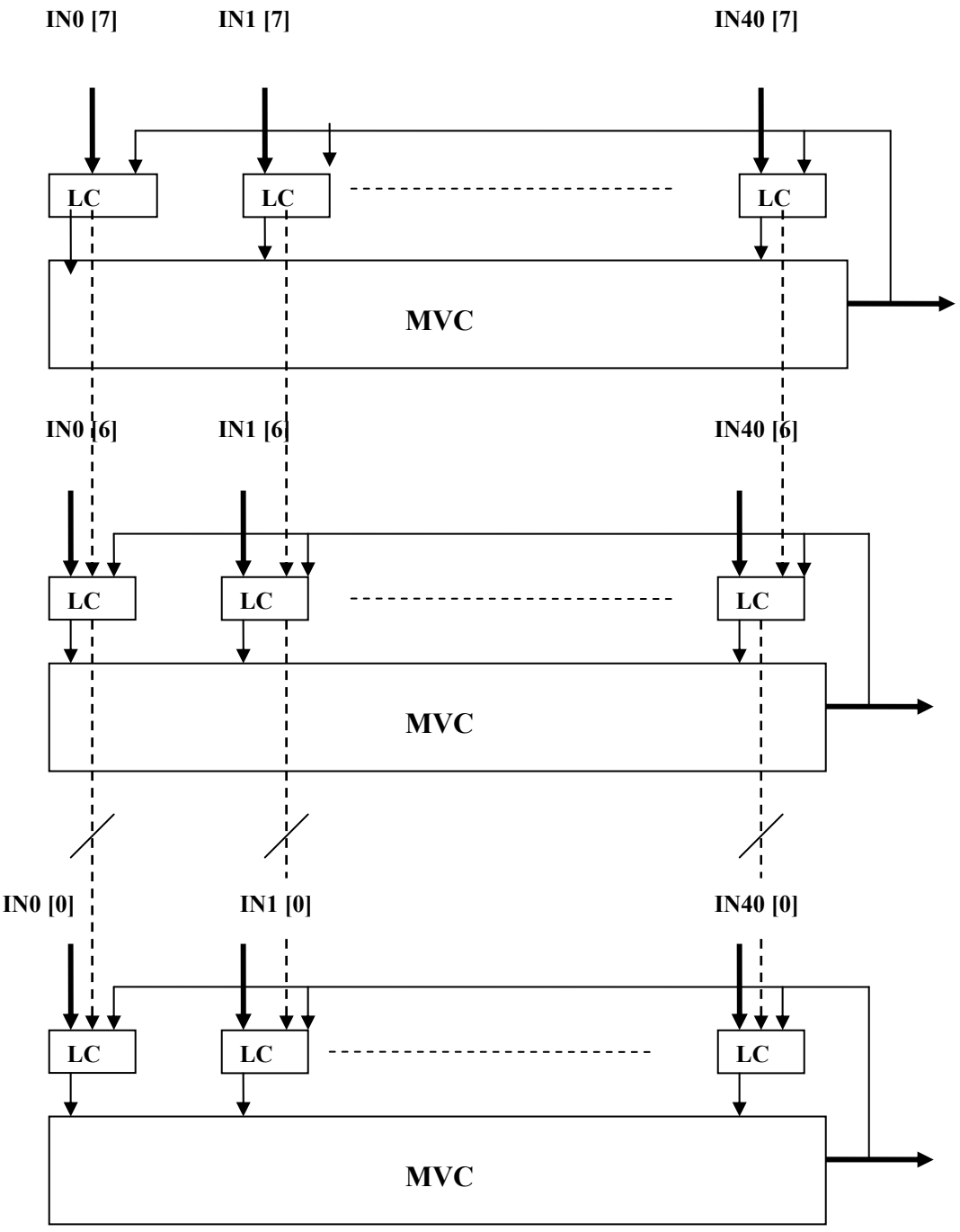


Figure 2.4 Block Diagram of Majority Voting Technique

second bit of the median. The procedure is continued down to LSB's. In this manner, the median value is determined by repeating the majority voting with simple logic control.

The circuit organization for majority voting circuit of forty one 8-bit data has been presented in figure 2.4.Every bit of the 41 data is fed to the circuit simultaneously and the median value is calculated asynchronously. LC unit are for logic control which turns the the data of loser bit into corresponding winner bit as described in the algorithm. The MVC unit has been implemented using floating gate technology as shown in figure 2.5.

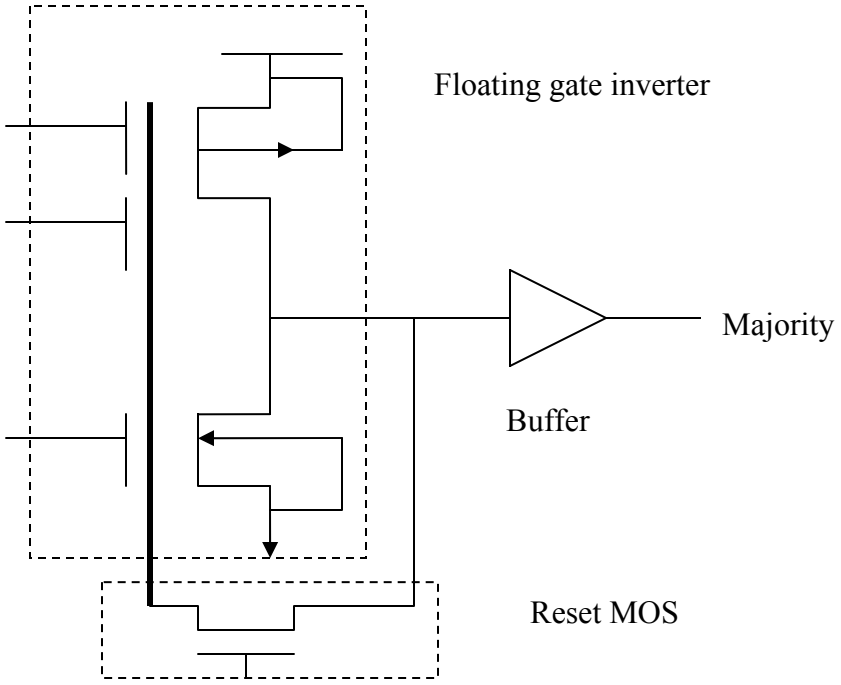


Figure 2.5 Majority Voting Circuit in This Work

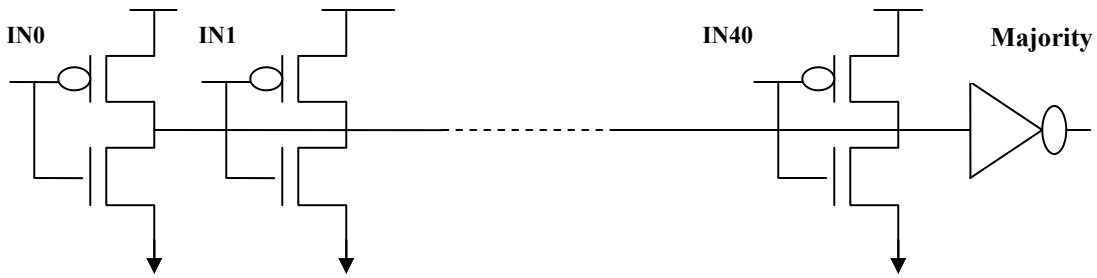


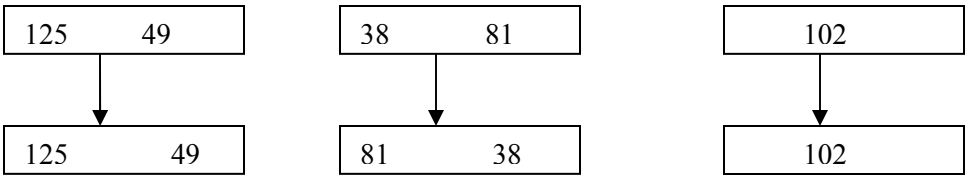
Figure 2.6 Conventional MVC Circuit

Their experimental work shows 74% reduction in power as compared to their previous work.

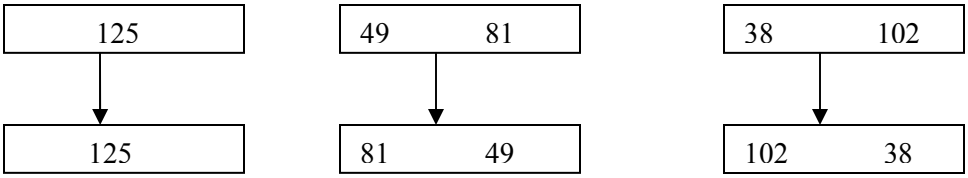
Krishna J. Palaniswamy et al has designed an 8-bit VHDL based 2-D median filter using Mentor Graphics tools. The algorithm is based on sorting mechanism that was proposed in [1].The VHDL code has been written, synthesized and optimized for an IC layout using CMOS 2–μm technology. This algorithm is implemented in MATLAB and then synthesized in CAD tools. This paper is a standard reference for building block for the development of post processing Application Specific Circuits (ASIC) based system for video signals. The main motivation of this project is to combine the design methodologies and environments for digital signal processing architectures and applications with VLSI design.
The mathematical model for sorting:

Example: 125 49, 38, 81,102

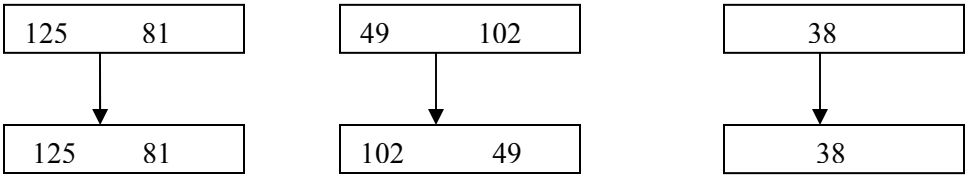
A comparator and swap unit is used as in [1] .First starting from left number pairing has been done. Here there are odd numbers of values and hence one number has been remained atlast. Again after each step of comparing the pairing starts from right most side .This has been clearly shown in example mentioned below. Each compare and swap stage compares the values and the larger value occupies left most side of block and the smaller in the right side.



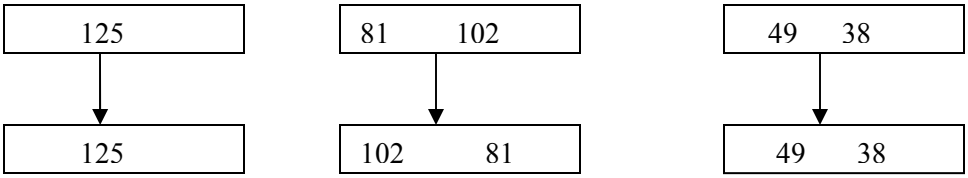
Compare and swap stage I



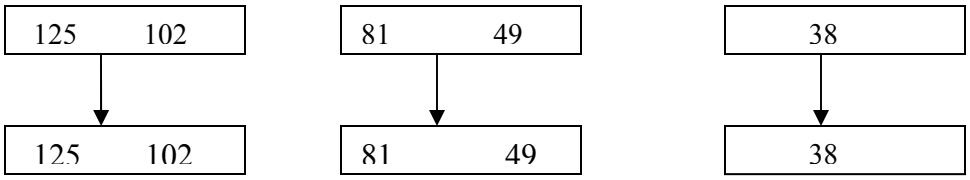
Compare and swap stage II



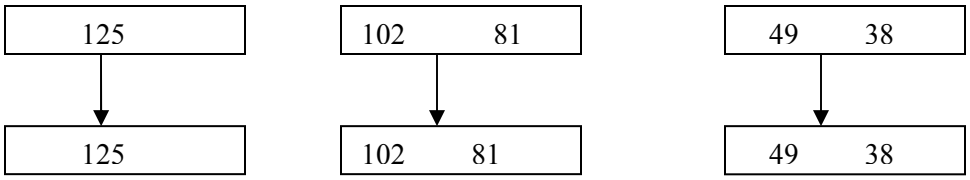
Compare and swap stage III



Compare and swap stage IV



Compare and swap stage V



Krishna et al have proposed a median filter [7] which has used a fast memory with multi ported enable signals. This was also implemented in VHDL in Mentor Graphic platform. This special RAM cell has used ass transistor logic. The previous 1-d median filter has been extended to two dimensional architecture using registers as shown in figure

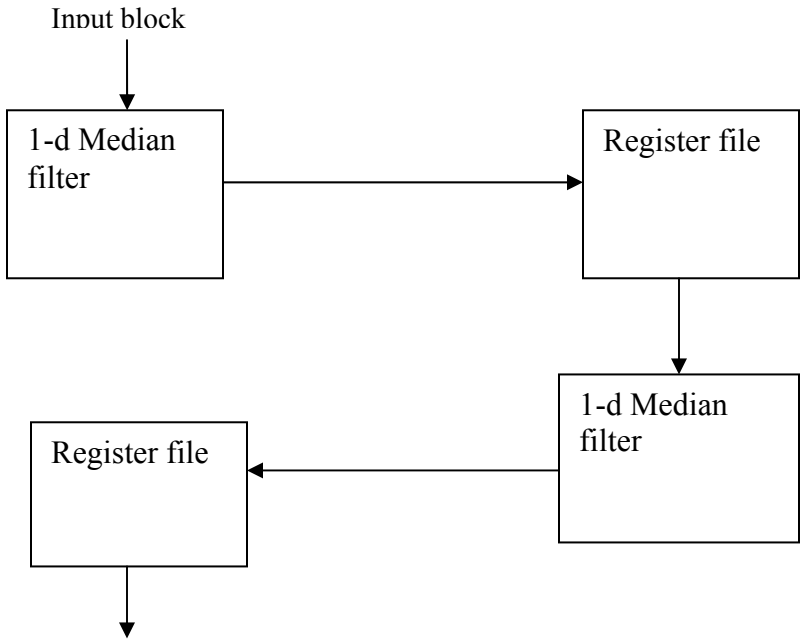


Figure2.7 Block Diagram of 2-D Filter

VHDL implementation shows the control register file has been handled by address, read, write command and all these signals are synchronized by single clock.

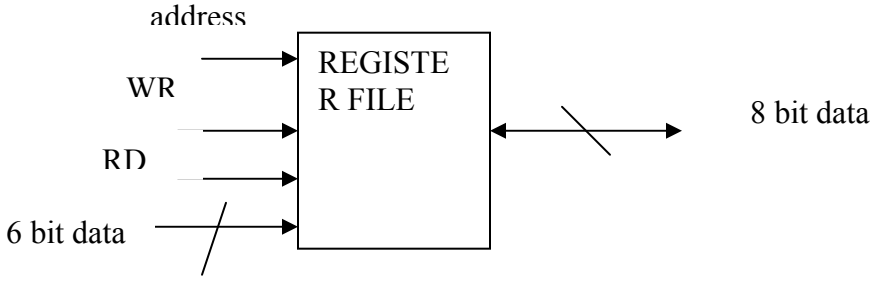


Figure 2.8 Block Diagram of Register Block

In paper [12] FPGA implementation median filter has been described .This paper gives the algorithm and implementations details of a sliding real time 3x3 median filter. The design is implemented in a XILINX XC4010 FPGA chip.

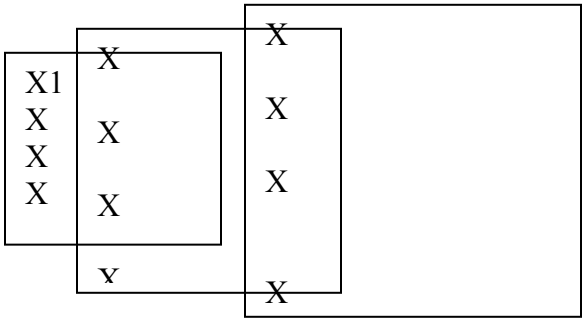


Figure 2.9 Block Diagram of Sliding Window

On each clock cycle window of size 3 X 3 moves on each pixel and only one column has to be slided as other two columns are same and hence using parallel-serial input scheme it can be used only vertical sorting. The result shows a few number of gates and flip flops compared to [1] with same throughput.

A combined VLSI architecture for non linear image filters has been presented in paper [9] .The types of filters consist of weighted order statistics, stack, nonlinear mean, Teager, polynomial and rational filters. The VLSI architecture has been in given in figure. This architecture consists of memory, registers, the individual filters, a multiplexer, control lines, a normalizer as well. The memory contains the data taken from image and the multiplexer is used to select any six of these filters. This is a multifunctional image processing solution for applications that requires the output of nonlinear filters such edge preservation smoothing, noise filtering and image segmentation.

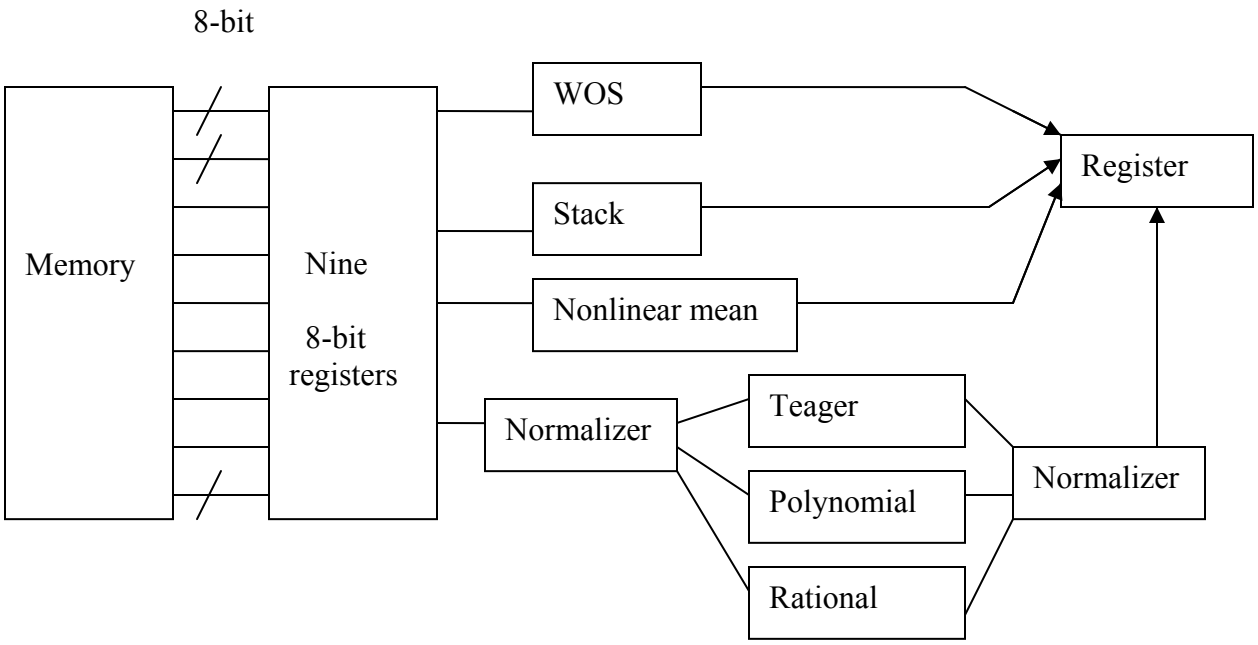


Figure 2.10 Universal Filter Block Diagram

A VLSI rank Order filtering using DCRAM architecture addresses a VLSI design of rank order filtering (ROF) technique adapting a mask able memory for real time speech and image processing.Meng chun Li et al have proposed a bit wise sorting based algorithm as described in [3] with a special defined memory called Dual Cell Random Access Memory(DCRAM). Meng et al have designed a complete 16 bit image processor having architecture as shown in figure 2.11.The main component of this architecture is a dedicated memory whose structural model has been shown in figure given below. The mechanism consists of three main steps namely i) bit counting ii) Threshold decomposition iii) Polarization. As followed in the MVC circuit the scan starts from MSB and bit counting is done to calculate the major bit. This is also called as threshold decomposition. Depending upon winner bit other bits of looser data set has been modified .This step is called polarization. This is very much similar to MVC technique [1].Here these three steps are implemented in pipelined manner as the DCRAM can perform the bit-sliced read which can read data from memory, partial write, computing the majority bit. The bit-sliced read and partial write are driven by mask able registers. With recursive

execution of the bit-slicing read and partial write, the DCRAM can effectively realize ROF in terms of cost and speed

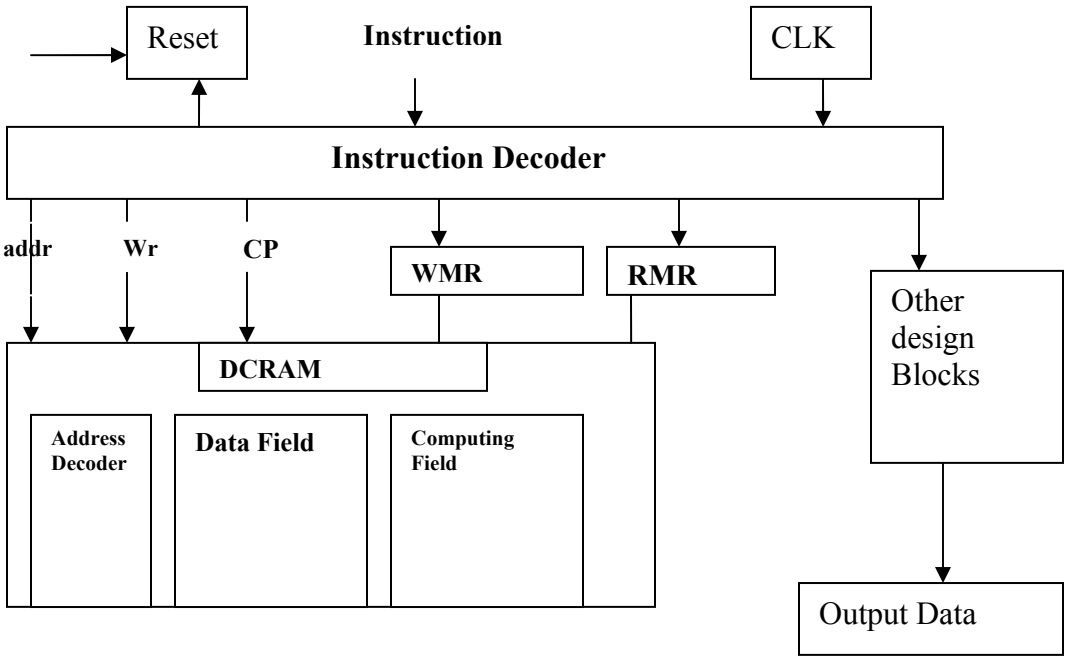


Figure 2.11 Diagram of Architecture of Memory Based ROF Filter

Different techniques for low power VLSI design have been illustrated in [10, 11].The different levels of abstraction for low power solution have been provided elaborately in[15]. A pass transistor logic based design of a 32 bit adder has been illustrated in [15].A digital system deigns of combinational and sequential circuits have been given in detail in [16].Basics of VHDL language has been given in VHDL Primer [17] and Various system level design (sequential and combinational) A CPU design and its behavior model has been given in "VHDL Programming by Example" by Douglas Perry [18] and "Digital System Design" by Tomas Lang et al.

2.2. Performance Evaluation

The resulted images are taken from MVC logic [4].

Figure 2.12 Noisy LENA Image

Figure 2.13 Filtered Image by MVC [4]

The implementation result based on voltage level Floating Gate based majority voting circuit [3] has been presented and compared with previous design of general purposed median filter [3].

Table 2.1

Logic	operation	Technology	Latency	Chip area	Power
MVC[1]	Mixed signal, Current	0.35µm	12ns	0.51mm²	72.6mW

Characteristic parameters of the chip implemented by general purpose median filter design in [1] are shown in table given below (using SPICE tool).

Table 2.2

Parameter	The real time median filter chip [1]
Die size	2.8 x 4.55 mm²
Transistor count	22000
Transistor/mm²	450
Max. Power	800 mW

Chapter 3

Processor Design

There are generally two types of microprocessors: general purposed microprocessors and dedicated microprocessors. General purposed Microprocessor such as Intel Pentium works under the control of software or operating system. On the other hand dedicated microprocessors are less complicated and specially built for certain tasks. They are most commonly known as Application Specific Integrated Circuits (ASIC).The digital circuit for microprocessor is called as Central Processing Unit. The logic circuit for CPU is of two types: data path and control path circuits.

The general purpose microprocessor process consists of building different blocks or components of microprocessor like ALU, Control unit, instruction register and memory unit with common data bus. Now the data path for various instructions set is designed and it is checked whether the instruction set is working efficiently under control unit. There are various components which are controlled by signals generated from control unit which is based on purely sequential circuits (Finite State Machines).

Here a simple memory mapped processor as shown in figure 3.1 has been designed and its operation based on 16-bit instruction set has been shown in figure 3.2 drawn below.

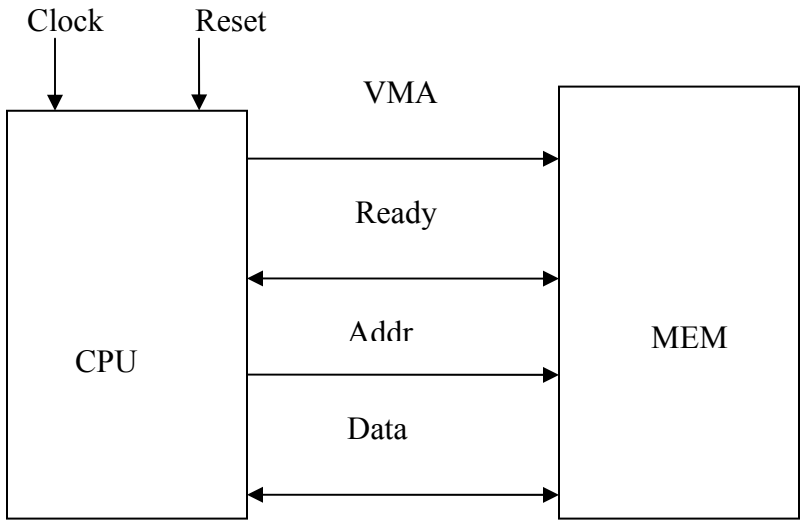


Figure3.1 Diagram of Top level design of CPU

3.1. Instruction set for above CPU

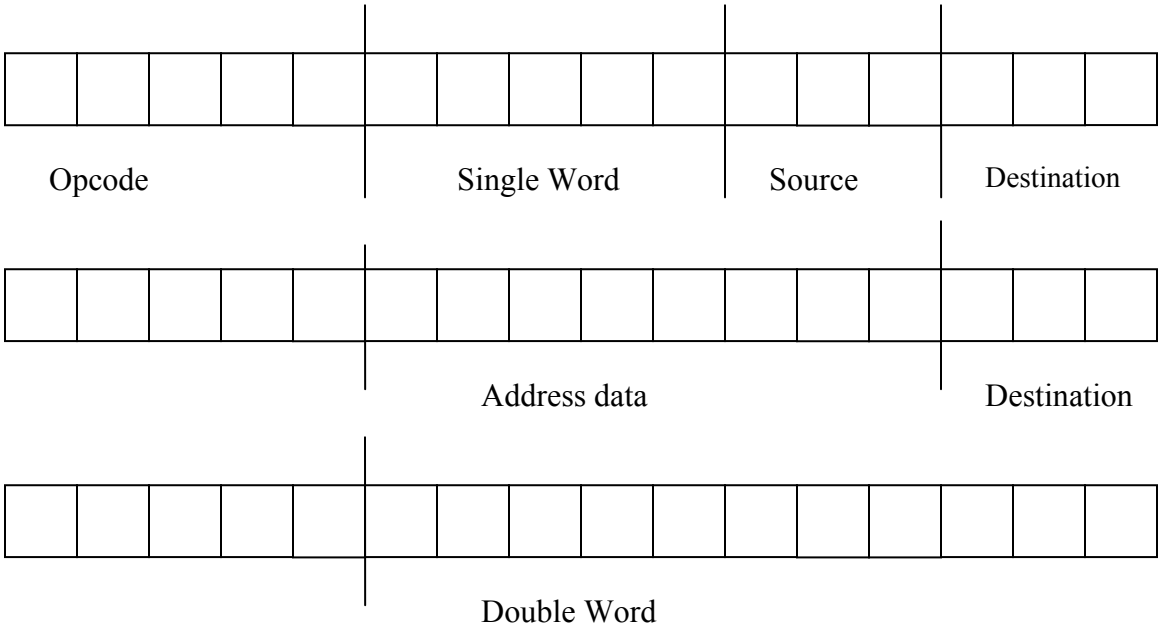


Figure 3.2 16-bit instruction set

Instruction set:

The designed CPU supports instructions mentioned below

Limited instructions:

i. Single word instruction

Arithmetic Add R0, R1, Sub R0, R1, Mul R0, R1

Logical Xor R0, R1, And R0, R2, Or R4, R5

ii. Double word instruction: (first word for opcode and destination and second word for location

Loadi (load immediate), movi (move immediate), bra (branch insrtruction)

3.2. Design Specifications

1.16-bit microprocessor and common data bus for 16-bit

2. Fixed point type arithmetic instructions

3. Memory unit (RAM type) can store sixty four 8-bit data

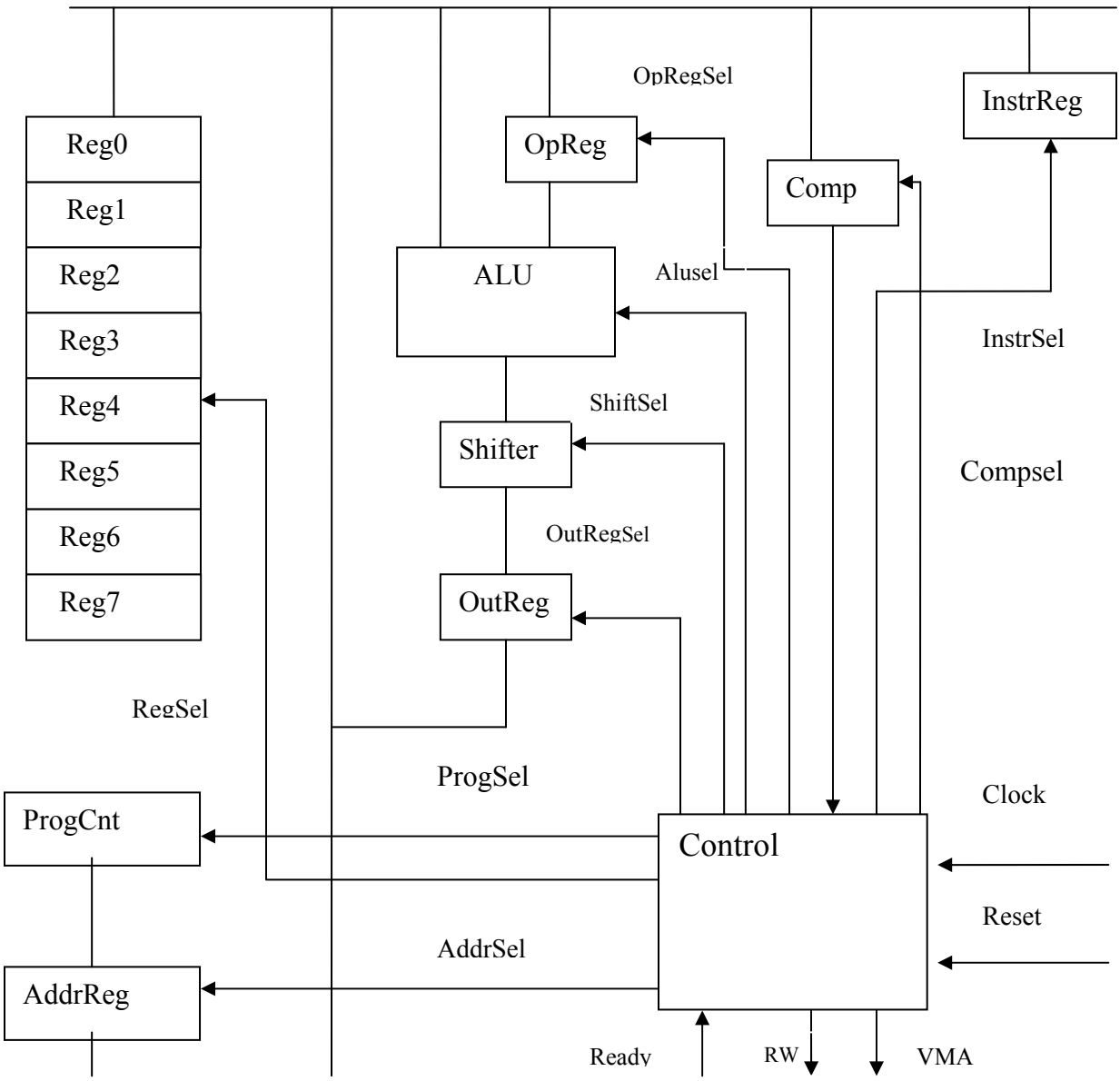


Figure 3.3 Detail block diagram of CPU

Designed a general processor having components as follows

1. ALU: This component does all arithmetic and logical operations.
Arithmetic operation: Addition, sub straction, multiplication Logical: xor and or
2. Control Unit: contains main clock and all the signals that triggers all the other components like address register, ALU , REG, Comparator ,program Counter etc.
i. AddreSel : signal for activating address register unit
ii. AluSel: signal for activating ALU for arithmetic operation
iii. RegSel: used for signaling reading or writing to registers
iv. InstrSel: used for enabling the instruction register
v. CompSel: used for enabling comparator unit
3. Comparator: compares two operands
4. Shifter: right or left shift by one bit
5. Instruction Register: contains the instruction of 16 bit
6. Address register: contains the address of instruction
7. Program Counter: contains the address of current instruction location
8. Registers: Array of eight 16 bit registers
9. Data bus: It is 16-bit bus and connects each component for triggering the signals

This design has been designed and synthesized using VHDL for Spartan 3 FPGA kit (XC3S250).The generated test bench waveform has been shown in figure

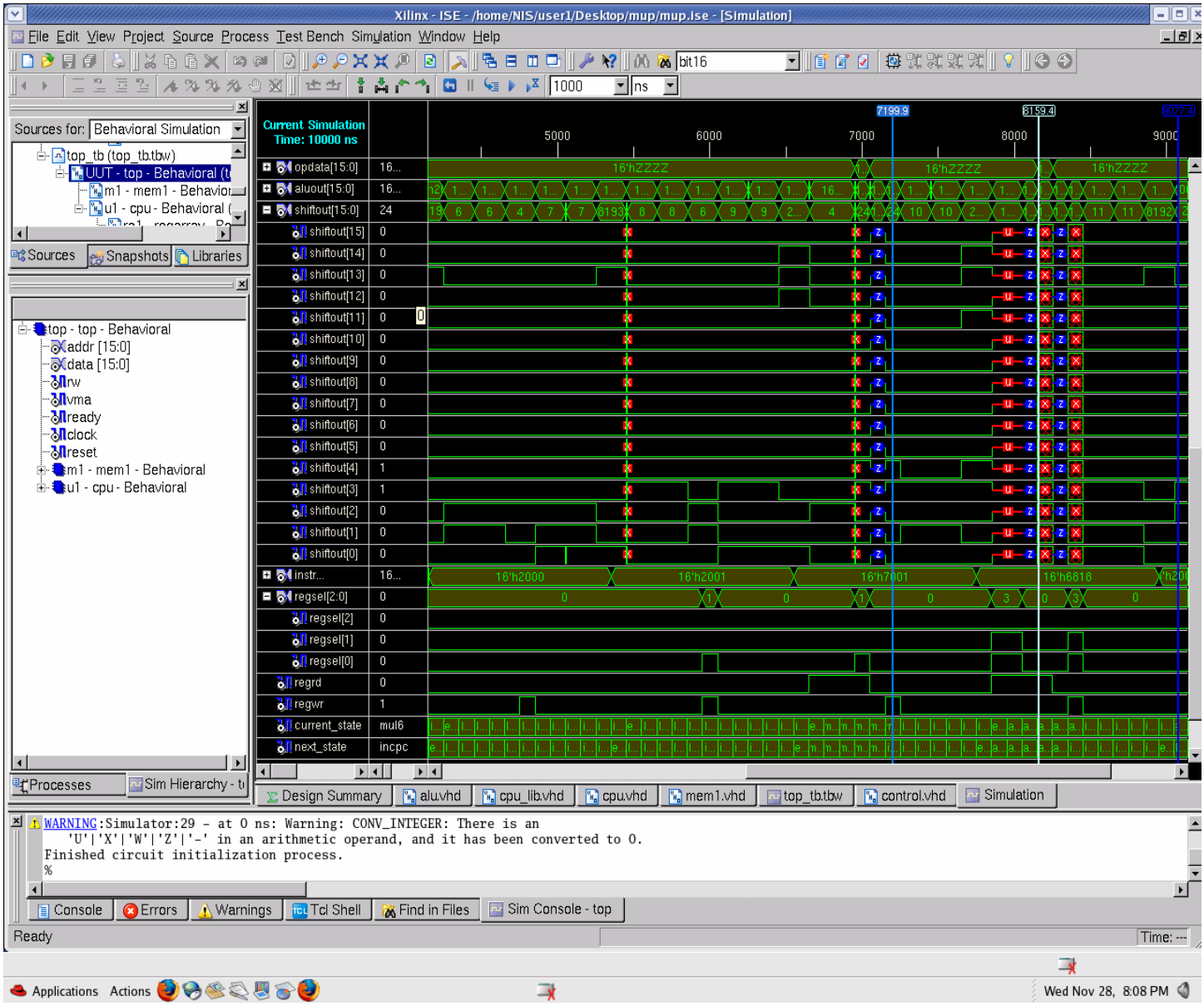


Figure 3.4 Snapshot of Test bench Waveform-1 for CPU Simulation

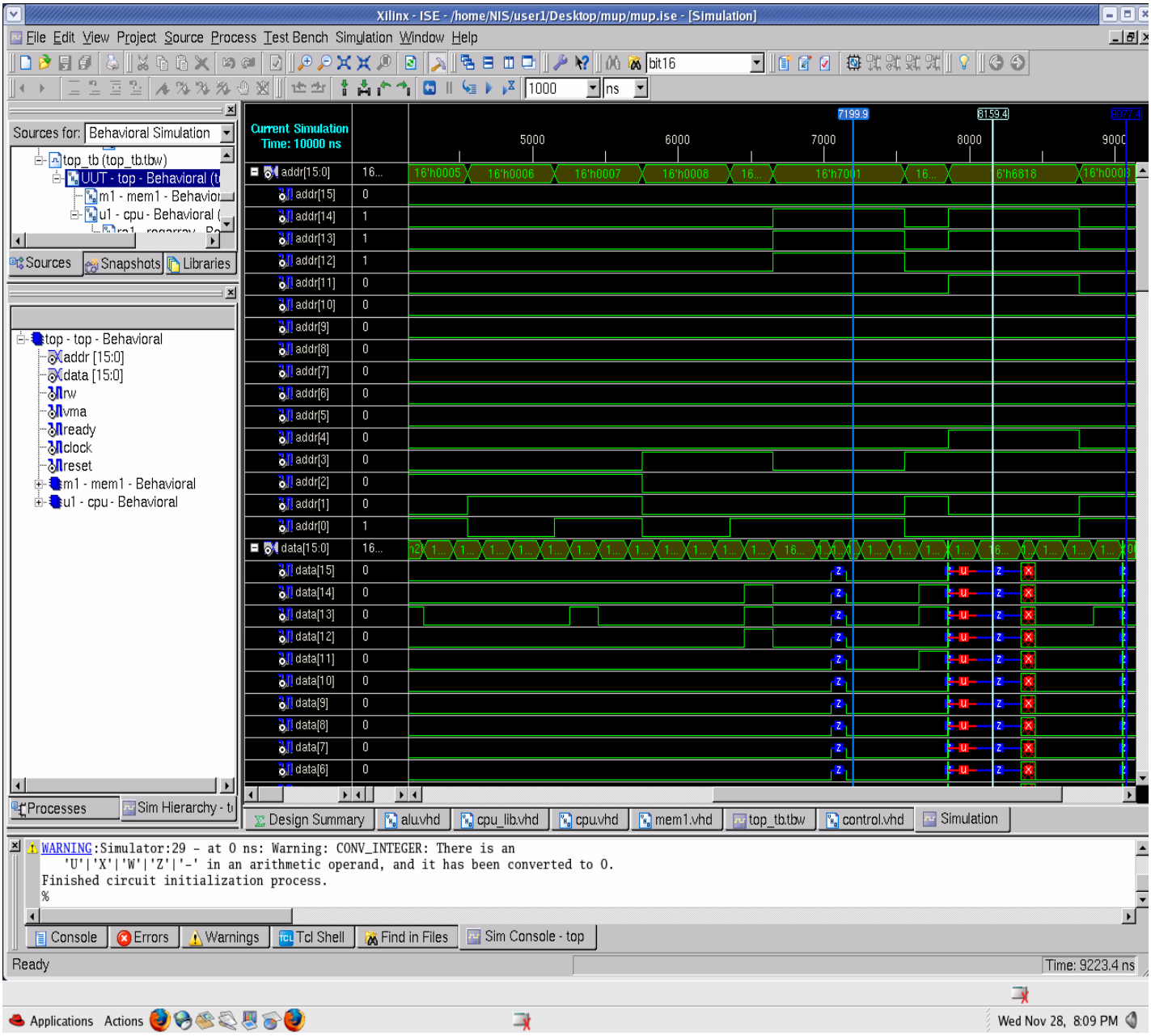


Figure 3.5 Snapshot of Test bench Waveform-2 for CPU Simulation

Chapter 4

Low Power Selective Median Filter Design

4.1. Median Filter Implementation

Here our designed filter has been implemented using various low power techniques at logic level and algorithmic level of majority sorting technique as mentioned in [4]. Different logics have been tried while designing the sorting circuit. In the following section the double derivative mechanism and those majority bit evaluation scheme has been discussed.

A standard image of LENA of size 250 x 250 is taken and the above selective median filter is applied and tested.

In 1-D and 2-D standard median filtering applications a window of size w ;where w is an odd size is usually taken as 3 x 3 or 5 x 5.

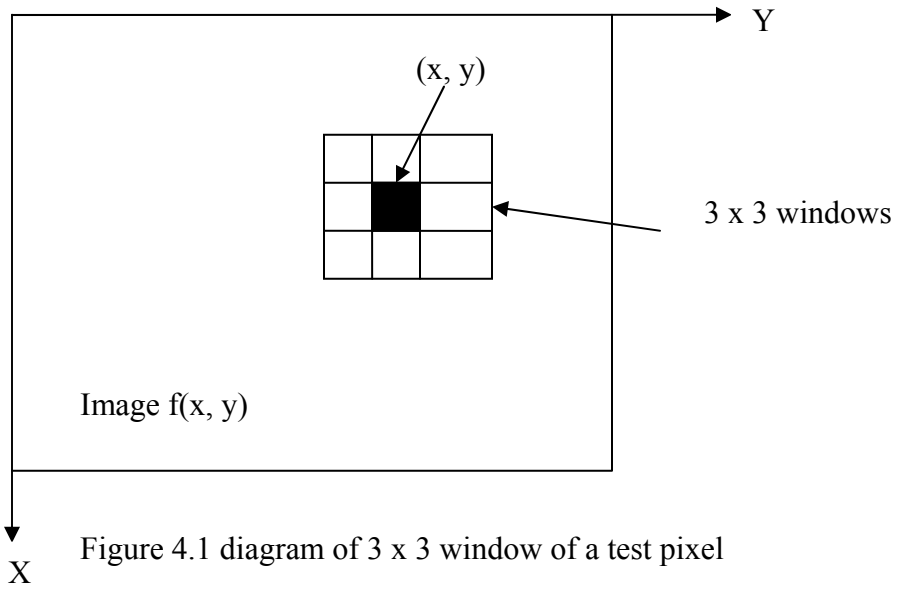


Figure 4.1 diagram of 3 x 3 window of a test pixel

Median filter applied on (x, y) as shown in above figure having 3x 3 window

X11 X12 X13

X21 X22 X23

X31 X32 X33

Median value => F(x11, x12, x13, x21, x22, x23, x31, x32, x33)= median value of sorted values =>SORT(x11, x12, x13, x21, x22, x23, x31, x32, x33)

4.2. Double derivative implementation

Here we have used 3x5 window and accordingly

x11	x12	x13	x14	x15		x'11	x'12	x'13	x'14		x''11	x''12	x''13
x21	**x22**	**x23**	**x24**	**x25**	**=>**	**x'21**	**x'22**	**x'23**	**x'24**	=>	**x''11**	**x''12**	**x''13**
x31	x32	x33	x34	x35		x'31	x'32	x'33	x'34		x''11	x''12	x''13

Here x22 is the target pixel so instead of taking 3 x 5 filter into a memory only five values are taken and difference of their values is calculated.
Where

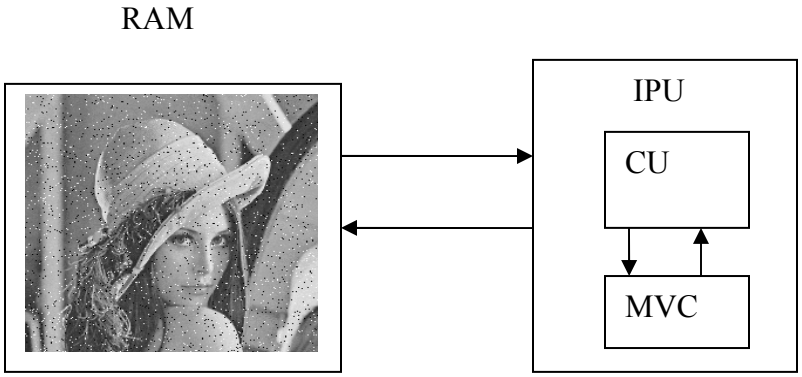
x'11= |x11-x12|, x'12= |x12-x13|, x'13= |x13-x14|,x'14= |x14-x15|
x'11= |x11-x12|, x'12= |x12-x13|, x'13= |x13-x14|,x'14= |x14-x15|
x'11= |x11-x12|, x'12= |x12-x13|, x'13= |x13-x14|,x'14= |x14-x15|

and

x''11= |x11-x12|, x''12= |x12-x13|, x''13= |x13-x14|
x''21= |x11-x12|, x''22= |x12-x13|, x''23= |x13-x14|
x''31= |x11-x12|, x''32= |x12-x13|, x''33= |x13-x14|

If x''22 >=t then apply median filter(x22) else do not
t = Threshold value
Here we have taken t =100 for MATLAB simulation.

4.3. Simple memory mapped Control Unit design for median filter



RAM of 30 X 30 size, IPU:-Image Processing Unit
CU: - Control Unit, MVC:-Majority voting circuit

Figure 4.2 a simple image processor

In this architecture a simple memory based control unit has been designed using VHDL language. As shown in above figure in the IPU component MVC logic has been embedded. CU has control signals like clock, mvc_rd and vma for controlling dataflow and control flow between memory and IPU.

4.4. Sorting circuit implementation

A sorting circuit is an utmost requirement for median value evaluation of large values. We have implemented the same sorting mechanism of majority voting circuit [4] forty one 8 bit sorting circuit has been designed and compared with other results. For majority bit calculation there have been three logics which are given below.

If a test vector (x1 x2 and x3) of three input bits either 0 or 1 are taken then

i) **Logic1**: Majority*(x1, x2, x3) = 1 if count1 > count0 else 0; where count1 = no. of one in x1, x2, x3 and count0 = no. of zeros in test vector.

ii) **Logic 2**: Majority*(x1, x2, x3) = 1 if $\sum xi > 1$ else 0.

iii) **Logic 3**:Majority*(x1, x2, x3) = 1 if (x1 or x2 and x3) = '1' else '0'

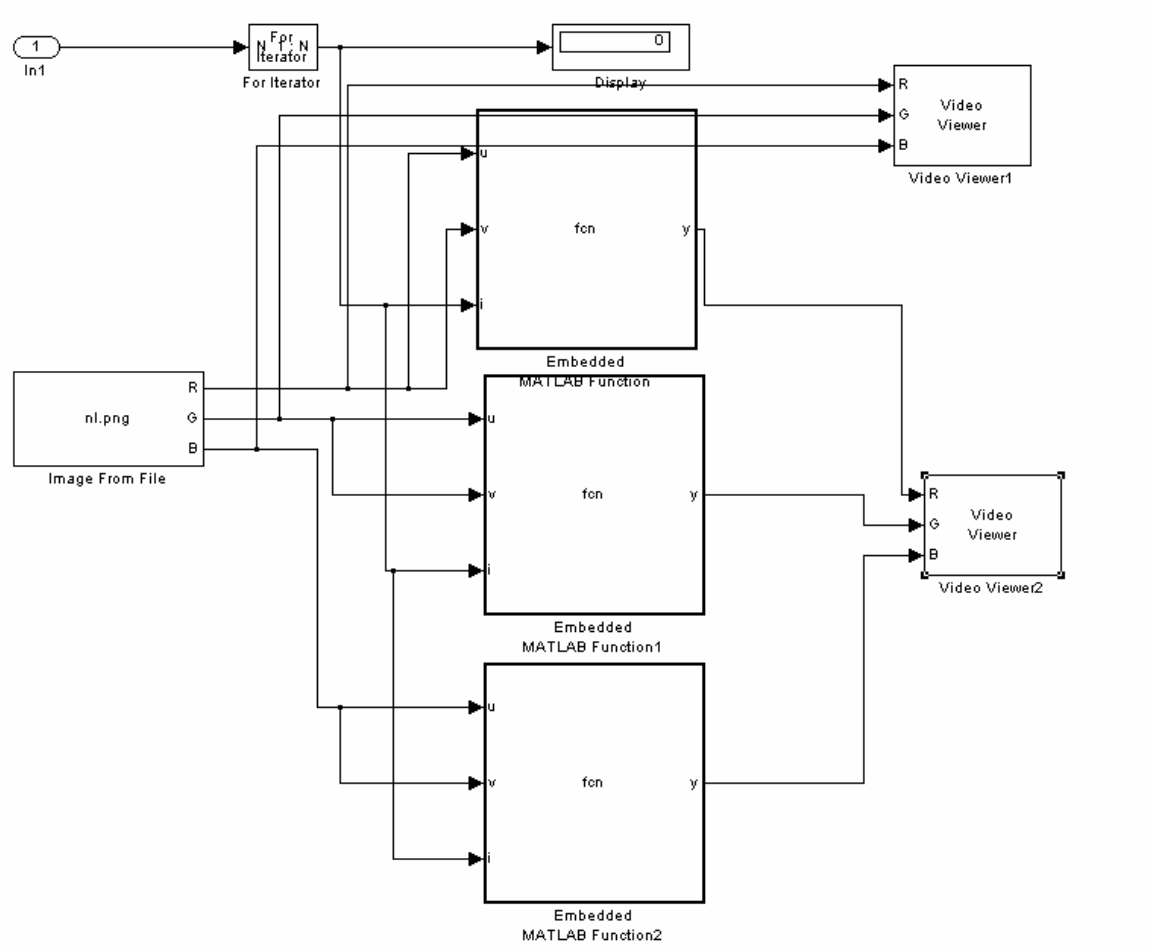


Figure 4.3- MATLAB User Defined function blocks for Median filter

Figure 4.4 (a) Original Image (b) Image corrupted by 10% salt& Pepper noise (c) Filtered Image

Figure 4.5 Image corrupted by 50% salt and pepper noise and filtered image

As shown in figure 4.4 and 4.5 the standard Lena image (250 x 250) has been taken and added by 10 % and 50% salt and pepper impulsive noise Using MATLAB 7.1 and then filtered by selective median filter one by one.

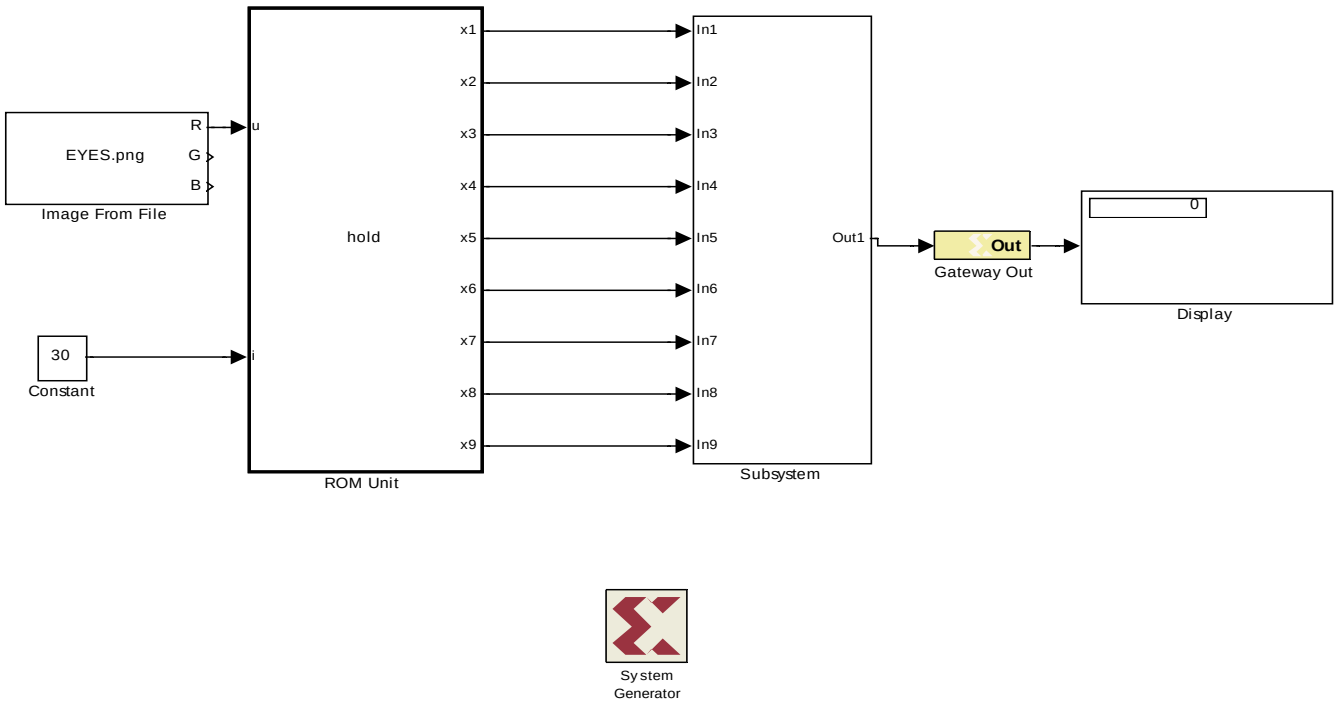


Figure 4.6: Simulink Block Diagram Of median calculator of nine pixel values

The basic operations of following blocks are:

Image from file (Image toolbox) = It takes image file from a user defined location or directory. R, G, B are three output of size of image matrix.

Rom Unit (MATLAB user defined function): It retrieves nine input values of a pixel according to the index value.

System Generator (XILINX block set) : This block is for the control of system and simulation

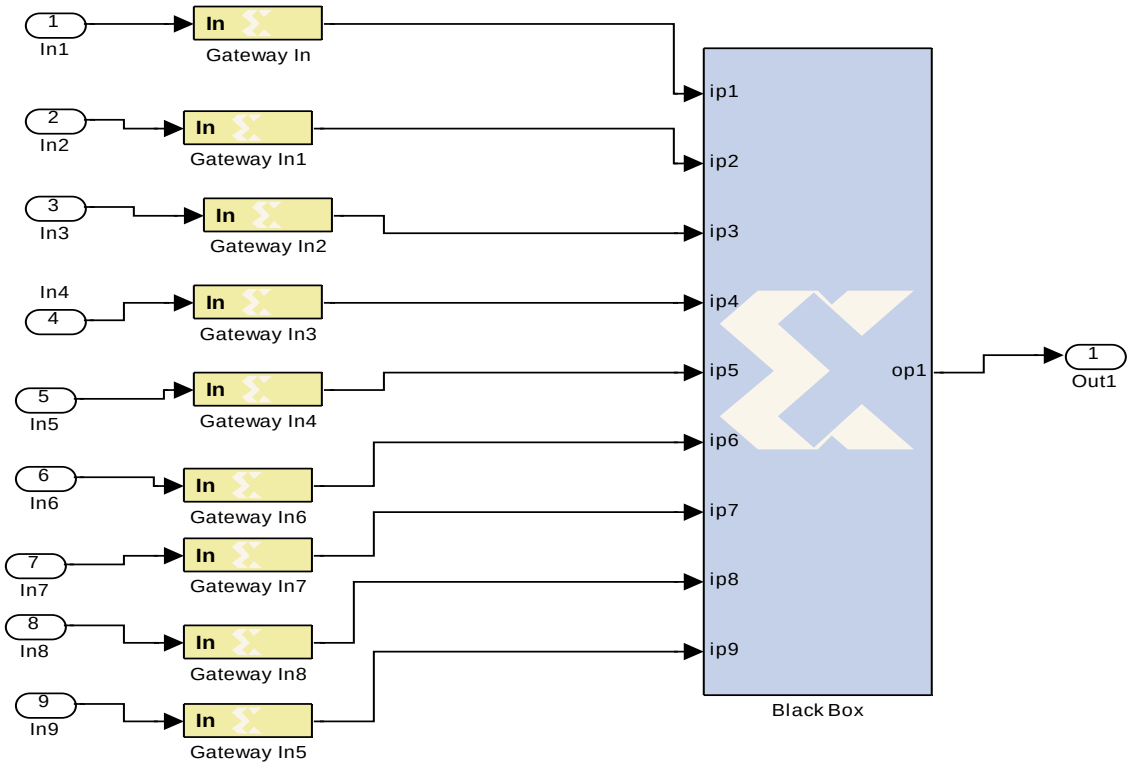


Figure 4.7 Diagram of median value evaluation Subsystem using XILINX Block set

Black box (XILINX block set): It implements the Majority voting circuit in VHDL.

Gateway in and Gateway out (XILINX block set): These are used for type conversion from MATLAB type data types to XILINX type.

4.4. Results

Here the power report of our various designs has been given. In table 4.4.1 the total power consumption of 41 8-bit sorting circuits has been given. and the simulation was done in SYNOPSIS. The same design has produced power report in XPOWER synthesized in VIRTEX 5 as given in 4.8. A sorting circuit has been designed using forty one 8–bit data values using MVC logic as mentioned in [4].This design further re-implemented using Logic 1 and Logic 2 as described in section 4.3 and the power report produced by those design has been shown in table 4.4.2 and 4.4.3. The same code has been re-implemented using nine 8-bit input values in XC5vl250t-3ff1136 and on further applying Logic 1, Logic 2, and Logic 3 shows optimized power report as given in respective tables 4.4.4, 4.4.5 and 4.4.6.

1. Power report generated after optimizing in Synopsis tool

Table 4.4.1 Power report from Synopsis

```
Global Operating Voltage = 1.08
Power-specific unit information:
    Voltage Units = 1V
    Capacitance Units = 1.000000pf
    Time Units = 1ns
    Dynamic Power Units = 1mW    (derived from V, C, T units)
    Leakage Power Units = 1nW


  Cell Internal Power = 21.1893 mW   (59%)
  Net Switching Power = 14.4449 mW   (41%)
                        ---------
Total Dynamic Power    = 35.6342 mW (100%)

Cell Leakage Power     = 164.5556 uW
```

2. The power generated by XPOWER tool in XILINX

Table 4.4.2 Power summary of sorting circuit of 41 pixel values

Power summary:	I(mA)	P(mW)
Total estimated power consumption:		**5254**
Vccint 1.20V:	**4356**	**5228**
Vccaux 2.50V:	**10**	**26**
Vcco25 2.50V:	**0**	**0**

Table 4.4.3 Power summary of sorting circuit of 41 pixel values using logic-2

Power summary:	I(mA)	P(mW)
Total estimated power consumption:		396
Vccint 1.00V:	238	238
Vccaux 2.50V:	62	155
Vcco25 2.50V:	1	3

Table 4.4.4 Power summary of nine input values

Power summary:	I(mA)	P(mW)
Total estimated power consumption:		80
Peak Power consumption::		545342

Table 4.4.5 Power summary of nine input values using logic 2

Power summary:	I(mA)	P(mW)
Total estimated power consumption:		72
Peak Power consumption::		1216730

Table 4.4.6 Power summary of nine input values using logic 3

Power summary:	I(mA)	P(mW)
Total estimated power consumption:		67
Peak Power consumption::		1424754

Table 4.4.7 Table for Comparison of Power Measurements of two designs

Design	Logic style	Power(in mW)	Delay(in ns)	Process technology (in μm)
Our work	Digital	36	2.52	0.13
MVC [4]	Mixed signal	72.6	12	0.35

Chapter 5

5.1. Conclusion

The above results confirms that various logic level and algorithm level optimization can enhance the power efficient design process. The table 5.1 shows the comparison between sorting circuit of MVC [4] and our work. This result has been taken from power reports and design reports of VHDL design of sorting circuit of forty one 8-bit values implemented in SYNOPSIS. This selective median filter is definitely effective because it saves some of extra computational work and it is also fast because it is based on hardwired logic. But the area report of the design shows increments in area and number of gates or cells is higher in comparison to others. Other factor is delay or latency which is shown in table is not an optimized value for clock delay and pin to pin delay. It has only shown the switching delay; although there are various timing analysis report generators available in various CAD tools. Our simulation has been completely based on logic simulation of sorting net work for 3x3 windows.

5.2. Future Work

Since this work is a complete digital design where there can be greater performance in terms of speed, area and power using mixed signal and analog signals circuits. Analog design of various system level design is now a days a major research interest. Various analog designs based on Operational Amplifier based sorters or comparators which are much faster than this hardware can be embedded for sorting circuit. This design has been a simple one optimizing only power factor of sorting circuit of MVC circuit. This design can be made suitable for pipelined architecture which will further increase both speed and power efficiency. At last this architecture can be a extended to general purpose image processor by making it easier for a number of image processing applications.

References

1. R. C. Gonzalez and R. E. Woods, "*Digital Image Processing*", 2nd edition. Englewood Cliffs, NJ: Prentice-Hall, 2001.

2. B.Majhi, P.K.Sa and G. Panda, "Second Order Differential Impulse Detector", *Internatioanl Conference on Intelligent Systems (ICIS-2005),* 1-3 December, 2005, Kuala Lumpur, Malaysia.

3. M. Karman, L. Onural, and A. Atalar, "*Design and Implementation of a General purpose Median Filter* Unit in CMOS VLSI," IEEE J. Solid-State Ciruits,Vol. 25, No.2, April 1990

4. Hideo Yamasaki and Tadashi Shibata,"*A High Speed Median Filter VLSI Using Floating Gate MOS Based Low Power Majority Voting Circuits*",Proceedings Of ESSCIRC,Grenoble,France,IEEE ,2005

5. Long-Wen Chang And Jing-Ho Lin,"*Bit level systolic arrays for real time median filters*",IEEE 1990

6. Krishna J. Palaniswamy, Maher E. Rizkalla,Akhouri C. Sinha and Mohmed El-Sharkawy,"*VLSI implemetation Of Memory Design Applied to median Filter*",IEEE Journal of Solid State Circuits,Vol-32,18 April 1999

7. Meng-Chun Lin,Lan-Rong Dung, "*On VLSI Design of rank order filtering using DCRAM architecture*", Integration VLSI Journal,9 May 2007

8. Orlando J. Hernadez, Tara Keohane, and Julia Steponanko,"*A Combined VLSI Architecture for Nonlinear Image Processing Filters*",IEEE,2006

9. Gary Yeap," *Low power VLSI digital design*", Kluwer Academic Pubblishers, 1st ed.,1998.

10. A. Bellaouar and M. I. Elmasry,"*Low power digital VLSI design circuits and systems*". 2nd ed, 1st Edn., Kluwer Academic Publishers, 1995.

11. Rajul Maheswari,S S S P Rao,P G Pooncha, "*FPGA Implementation Of Median Filter*",10th International Conference On VLSI Design –January 1997.

12. A. P. Chandrakasan, S. Sheng, and R. W. Brodersen, "*Low-power CMOS digital design*",IEEE Journal of Solid-State Circuits, vol. 27, no. 4, pp. 473–483, Apr. 1992.

13. Rabaey JM and Pedram M, "*Low Power Design Methodologies*", Kluwer Publishers, 1996

14. Reto Zimmermann and Wolfgang Fichtner,"*Low power logic styles: CMOS Versus Pass –Transistor Logic*", IEEE Jornal of solid dtate circuits,VOL. 32,NO. 7, July 1997

15. Tomas Lang, Jaime H. Moreno, M Ercegovac "*Introduction to Digital Systems*", John Wiley & Sons, Inc. New York, NY, USA ,1998

16. Douglas Perry, "*VHDL Programming by Example*", 4th edn.,McGraw Hill publication,2002

17. J. Bhaskar, "*VHDL Primer*", 3rd edition, Prentice Hall PTR, 1999

18. Power calculation site: http://www.xilinx.com/cgi-bin/powerweb.pl

19. The Xilinx Home page: www.xilinx.com

Dissemination of Work

1. Radhamadhab Dalai,Banshidhar Majhi and K. K. Mohapatra "*A Low Power Median Filter Using Selective Double Derivative Filter*", 1st International Conference on Advances in Computing, Chikhli, India, pp 1-4, 21-22 February 2008.