Focal plane processing for very fast detection of motion vectors

撮像面上での即時動きベクトル検出に関する研究

指導教官:　相澤 清晴 助教授

東京大学 大学院
工学系研究科 電子情報工学専攻

Zheng　Li
李　正　67125

Doctor of Philosophy
of
University of Tokyo

December 1998

平成 10 年 12 月 18 日

論文概要

本論文には、いろいろなビジョンチップとイメジセンサを検討する上で、100–1000 フレームレート 程度の高速フレームレートでの撮像された物体に基づいて、撮像された物体の動きベクトル検出処理を即時に行なうビジョンチップを提案し、試作した。

1　論文の構成

- 第一章　序論 (背景と目的)
- 第二章　ビジョン 及び コンピュテーショナルビジョンチップ
- 第三章　撮像面上での動き検出と新しい提案
- 第四章　エッジ検出とブロックマッチングの分析
- 第五章　撮像面上でのエッジ検出、メモリ、動きベクトル検出の設計
- 第六章　プロトタイプビジョンチップの実装と仕様
- 第七章　プロトタイプビジョンチップの実験・評価・結果
- 第八章　結論

2　研究の内容と成果

試作されたビジョンチップの出力は撮像、エッジ検出と動きベクトル検出の三つの機能で構成されている。

- 撮像:
  試作されたビジョンチップの Photo Diode 数が 16 × 16 少ないのであるけれども、このような撮像面上で、基本画像が出力できる。

- エッジ検出された画像:
  全画面の画素は同時にリセットするので、撮像された画像がエッジ検出をしてから、二値情報で 二フレームのメモリの中に保存されている。そして、画像は二値のエッジ画像として出力もできる。

- 動きベクトル検出:
  一般的な CPU とか DSP とかなくて、撮像面上で Specific ACC と Specific プロセッサーを設計し、小範囲高速の動きベクトル検出はできた、0 から 8 までの Index 数字で表示されている。

試作したチップのエッジ検出部分では一つのエッジ検出器を水平エッジ、垂直エッジ検出に共用して、時間分割に (Time-multiplexed) エッジを処理する。画素当たり、現フレーム (水平、垂直) のエッジ、前フレーム (水平、垂直) のエッジに共に 4-bits メモリで作られている。メモリの書き込み / 読み出しは ツリーのアーキテクチャーによって行なわれる。複数の動きベクトルは存在の場合、中心部のベクトルが優先出力する。高速化が図られるため、ブロックマッチングのサイズは 2 × 2 画素、サーチエリアは $(\pm1, \pm1)$画素の範囲に限定され、ローカル並列グローバル列並列のアーキテクチャー(LPGCP)で、ローカルのブロックマッチングが並列で行なわれている内に、グローバルの動きベクトル検出は同時に列並列で行なわれている。

評価の結果については 次になる：

- センサ部 、エッジ検出部 とメモリ部はテスト回路で全部 正しく評価された。

- 16 × 16 画素での撮影できた。

- 高速ブロックマッチングを含む動きベクトル検出を持つプロトタイプチップが正しく評価された。

- 任意の二値の前フレーム (水平、垂直) のエッジと現フレーム (水平、垂直) のエッジを入力として、 プロトタイプチップに入力して、 4 × 4 範囲での動きベクトル検出の正しい出力波形が得られた。

Content

Chapter 1

INTRODUCTION

In this thesis, we first start with the human vision and the general computational vision chip, then develope our discussion to a specific motion detections on the focal plane, and finally propose and implement prototypes of vision chips concerning about imaging, edge detection and very fast motion vector detection on the focal plane.

1.1 Background and Purpose

The motion vector estimation usually requires high computational complexity and consums much of CPU time in conventional image processing system. In real time imaging and coding application, the reduction of high computational complexity of the motion estimation while preserving reasonable performance is always of a great concern. Many fast algorithms[1],[2] as well as some real time hardware designs[3],[4],[5] have been proposed, however, since a CCD camera, a lowpass smothing filter, A/D converter, as well as powerfull CPU or DSP are needed, not only the cost is expensive, the size is large, but also the highest estimation speed is limited by its bottle neck between a sensor and a processor.

Because of the difficulties of designing a conventional 2D motion vector estimation on the tiny focal plane, it is quite neccessary for us to work out a new way of estimating vectors on the sensor focal plane. Thinking of the imaging process with a high rate of $100 \sim 1000$ frames/sec, this kind of high speed imaging makes the changes in position very small between two succesive frames of neighboring pictures.

By making use of this feature, a fast estimation of 2D motion vector on the CMOS sensor plane is proposed here by using block matching with a small search window which can be reduced to 2×2 and with a small search area of $(\pm 1, \pm 1)$ pixels around the intentinal area. It firstly binarizes the detected edge information after the analog photoelectronics conversion and then performs the block matching by the edge information. In this way, the motion vector detection procedure can be clearly described by the

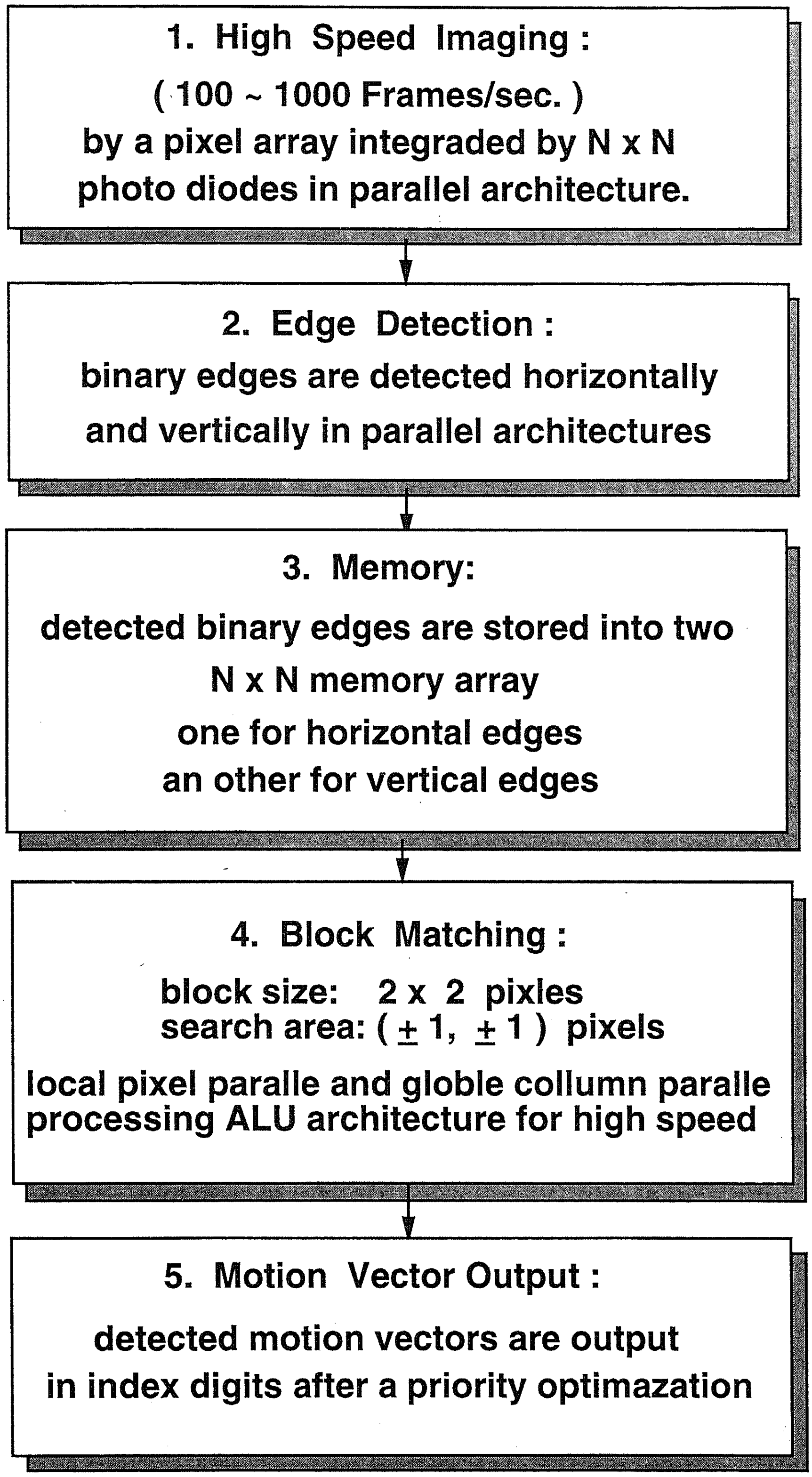

1. High Speed Imaging :
(100 ~ 1000 Frames/sec.)
by a pixel array integraded by N x N
photo diodes in parallel architecture.
2. Edge Detection :
binary edges are detected horizontally
and vertically in parallel architectures
3. Memory:
detected binary edges are stored into two
N x N memory array
one for horizontal edges
an other for vertical edges
4. Block Matching :
block size: 2 x 2 pixles
search area: (± 1, ± 1) pixels
local pixel paralle and globle collumn paralle
processing ALU architecture for high speed
5. Motion Vector Output :
detected motion vectors are output
in index digits after a priority optimazation

Figure 1.1: A general flow of the motion vector detection on a vision chip's focal plane with high frame rate imaging(100 ~1000 *frames/sec*) and small size of block matching. The vision chip can be implemented by *ananlog/digital* mixed CMOS process.

procedure in Fig. 1.1.

By using a submicro analog-digital mixed CMOS techology, the implementation of this vision chip can be illustrated by Fig. 1.2, which consists of the following main fuctions and therefore intends to detect a motion vector for every pixel and results in a motion field of vectors.

1. Imaging by MOS photo diode array:
   The imaging ability on the focal sensor plane can be designed by a $N \times N$ MOS photo diode (PD) array in parallel, all of these PDs are reset at the same time for getting a synchronizing integral time.

2. Edge detection:
   $N \times N$ horizontal edges and $N \times N$ vertical edges are detected in time multiplexed sampling.

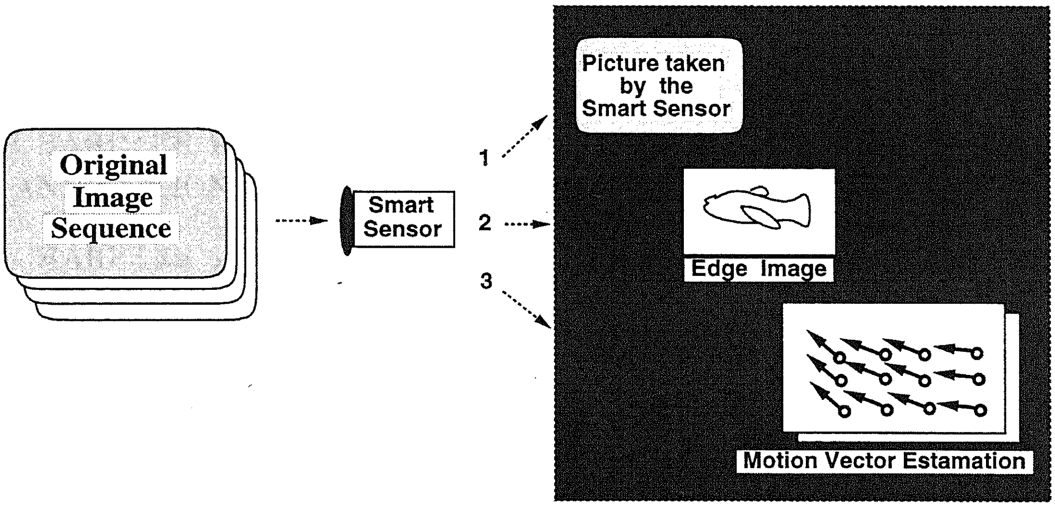


Figure 1.2: A Smart vision chip designed with the ability of imaging, edge detection, and motion vector estimation which can result in a motion field of vectors by estimating the motion vector for every pixel.

3. Memory arrays:
   The current frame edge information is stored dynamically into two $N \times N$ 4-bits memory arrays, one for horizontal edge and the other for vertical edge. They can be switched into two previous frame edge memory array.

4. Block matching for motion vector detection :
   To perform a block matching operation by 2×2 edges in current frame with the corresponding 4×4 edges in the previous frame in a LPGCP architecture.

5. Three kind of outputs:
   As is shown in Fig. 1.2, there are 3 kind of outputs by this vision chip.
   - imaging
   - edge imaging
   - index numbers among {0,1,2,...,8} are put out as the matched motion vectors after an arbitration optimization.

1.2 Main contents of the thesis

The main construction for this thesis consists of 8 chapters. It is listed bellow:

- CHARPTER 1 INTRODUCTION
- CHARPTER 2 VISION AND COMPUTATIONAL VISION CHIP
- CHARPTER 3 FOCAL PLANE PROCESSING FOR MOTION DETECTION
- CHARPTER 4 EDGE DETECTION AND BLOCK MATCHING ANALYSIS
- CHARPTER 5 CIRCUIT DESIGN OF THE EDGE DETECTION, MEMORY AND MOTION VECTOR DETECTION ON FOCAL PLANE
- CHARPTER 6 PROTOTYPE IMPLEMENTATION AND SPECIFICATION
- CHARPTER 7 EVALUATION AND EXPERIMENT
- CHARPTER 8 CONCLUSION

Chapter 2

VISION AND COMPUTATIONAL VISION CHIP

2.1 Introduction:

Can we simulate ourselves with the up-to-date technology? Clearly speaking, can we make use of electronics, biology and optics to implement some devices in order to imatate some functions of our own? From the history of the development of human beings, we can get a certain answer.

Human eye is one of the first element which the electronic engineerers are interested in. By a comparative studies between the human eyes and the vision silicons, we are making a great effort to realize some primary but marvellous functions of the precise eyes with the combinations of simple silicon element and signal processing technology.

T. Poggio, V. Torre, and C. Koch[6] once stated that vision chip or computational vision is the field of visual information processing. The main objectives of computer vision are to develop image understanding systems and to understand the human vision. Early vision is the set of visual models which can extract such physical properties of the object under view as distance, orientation, reflectance, color, texture, and so on. Several obstacles have been circumvented and efficient algorithms have been developed in implementing the hardware of the early vision system[6],[7],[8]. Image acquisition and the spatio-temporal smoothing for noise removal can be performed in the human-eye like silicon retina chip[9]. But the feature extraction processing is the next step which emphasizes some important properties of the image by more complicated algorithms. In this Chapter, we will discuss some fundmental vision ability of silicons compared with the human eyes and give a short survey[30] on this subject.

2.2 Properties of our eyes

As we know that our human eyes have three invarant features, since all these charicristics are important in implementing vision chips, we should keep these features in mind when we devise any algorithms on the vision chip. Therefore, let's study our eyes first.

1. Space Invariant

   When we make a servey on an object and make our eyes close to the object, we can see that only the concentrate center is enlarged, while the other part reminds almost unchanged so as not to interfere our attention, this is called space invariant. It explains that the eyes can pre-precess the image on the retina with a logarithm operation. When the distance between the eyes and the observed image changed, the size of the image on the retina will change accordingly, but after the logrithmic operation, the image input the brain will remain almost the same as before, that is the effort of the space invariant.

2. Brightness Invariant

   When an object is getting close to our eyes, we can fell that the brightness of the object is much less than the brightness that the Radium Theory measures, this feature of the human eyes is called the brightness invariant.

   If the general energy on the retina is

   $$E = \int G(r)dr \tag{2.1}$$

   then the general energy on the vision neuron is

   $$E' = \int F(u)du \tag{2.2}$$

   here, $u = ln(r)$

   We can see that the density $F(u)$ of the output is correspond to the brightness distribution $G(r)$ of the image on the retina. The logarithm opration makes the intgration area in formula (2.2) change a little when the observed object moves. Therefore the general energy in formula (2.1) remaims almost unchanged.

3. Time Invariant

   Time invariant is a kind of quick tracing ability, it makes the twinckle moving object form an image on the tissue in a fixed position in the eyes. The human eyes have an averaging ability when observing object, so we can find some target by a longe time staring even in the dark night. An experiment can be made with a battle of water. Since the statistical average of the water level is quite, when the center of the statistical picture is found, we can put all the centers together and convolute all the pictures taken by the camera. In this way, we can finally obtain a clear edge picture. This experiment properly simulates the time invariant feature of the human eyes.

2.3 Light Detecting and Transducing by Seeing Silicons

From now on, let's turn our attention to the magic silicon. In a similar way like the eye's system, the primary tasks of vision chips are detecting the light intensity and transducing it into some electrical parameters, voltage or current, and subsequently processing the signals from an array of detectors, in spatial and temporal domain.

2.3.1 Logarithmic Sensor Using MOS Diodes

The simplest circuit for converting the photocurrent to voltage is the logarithmic conversion circuit[10] shown in Fig. 2.1. The logarithmic function is a result of the subthreshold operation of the diode connected MOS transistors. As the input photocurrent is usually very small and falls within the subthreshold region of a MOS diode, the current-voltage relationship is determined by

$$\begin{aligned} I &= \frac{W}{L} I_{D0} exp(\frac{V}{U_T}\frac{1}{n}) \qquad (1\ diode) \\ I &= \frac{W}{L} I_{D0} exp(\frac{V}{U_T}\frac{1}{n^2+n}) \qquad (2\ diodes) \\ I &= \frac{W}{L} I_{D0} exp(\frac{V}{U_T}\frac{1}{n^3+n^2+n}) \qquad (3\ diodes) \end{aligned}$$

where W and L are the width and length of the transistor, respectively. I is the input photocurrent, n is the subthreshold slope factor, I_{Do} is a process dependent parameter, and V is the output voltage. This circuit has been the workhorse of many vision chips. In many designs this circuit is used because of its small size and large dynamic range. At low light levels, however, the circuit illustrates a very slow response, which necessitates longer settling times.

A disadvantage of this circuit is the extreme compression on the input signal. If the implemented algorithm needs to differentiate between signals, the logarithmic compression reduces the chance to detect such differences. This is specially true for motion detection chips. When the contrast of the input image is low, the logarithmic compression reduces the contrast to such low levels that in many cases only very large contrast edges can be detected.

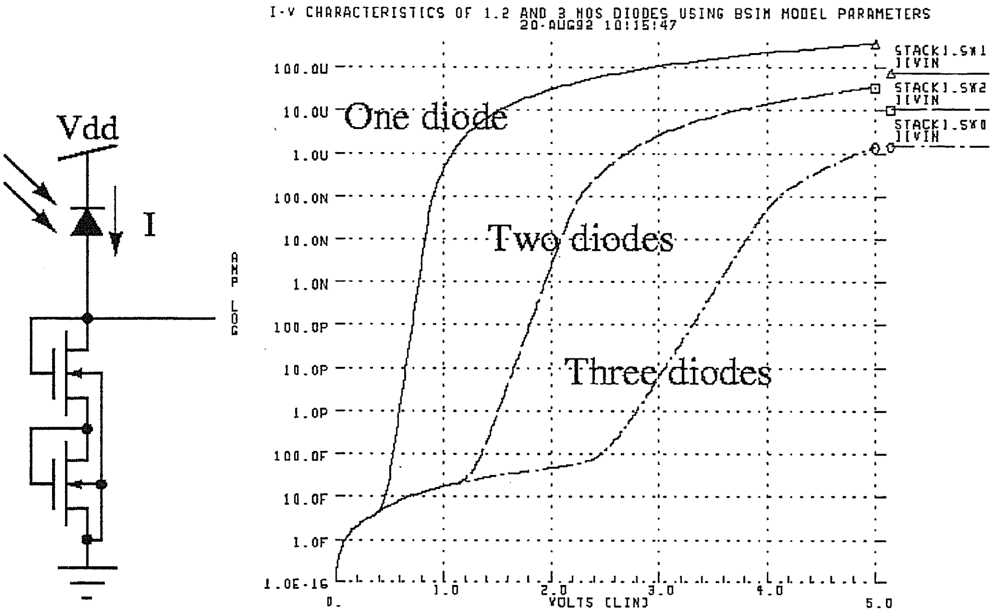


Figure 2.1: A logarithmic photocircuit with two series MOS diodes. Simulation results[10] for three cases.

2.3.2 Pixel Output: 1T-pixel, 2T-pixels and 3T-pixels

The architectures for pixel output are tipically represented by using 1T-pixel, 2T-pixels and 3T-pixels circuits, as shown in Fig. 2.2.

The 1T-pixel requires column amplifiers which introduce offset non-uniformity.

The 2T-pixel avoids charging and discharging of the bus-lines.

The 2T-pixel can cancellate the offset and the correlated double sampling(CDS).

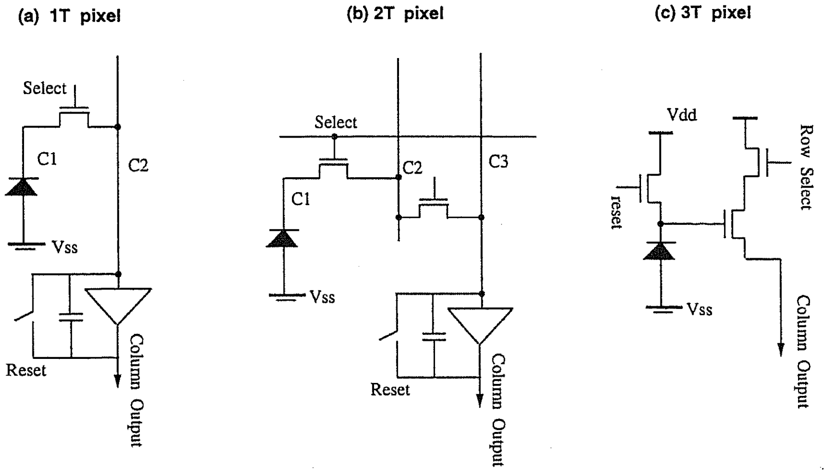


Figure 2.2: (a) 1T-pixel; (b) 2T-pixels; (c) 3T-pixels.

2.4 Computational Vision Chip

Computatinal/Smart vision systems on one chip[9],[10] will be an inevitable component of future intelligent systems. Conventional vision systems, based on the system level integration or even chip level integration of an imager (usually a CCD camera) and a digital processor, do not have the potential for application in general purpose consumer electronic products. This is simply due to the cost, size, and complexity of these systems. Because of these factors, conventional vision systems have mainly been limited to specific industrial and military applications. With the development of VLSI, some processing abilities, which include both the photosensors and parallel processing elements (analog or digital) attached to the sensor itself[7],[8],[17], have been under researchs in recent years and illustrate promising capabilities.

2.4.1 Study the Advantages and Disadvantages of Vision Chip

When compared to a vision processing system consisting of a camera and a digital processor, a vision chip provides many system level advantages.

- Speed:
  The processing speed achievable using vision chips exceeds that of the CCD camera-DSP combination. A main reason is the information transfer bottleneck between the imager and the processor. In vision chips, information between various levels of processing is processed and transferred in parallel.

- Dynamic range:
  Many vision chips use photodetectors and photocircuits which have a large dynamic range over at least 7 decades of light intensity.

- Size:
  Using single chip implementation of vision processing algorithms, very compact systems can be realized.

- Power dissipation:
  Vision chips often use analog circuits which operate in subthreshold region. There is also no energy spent for transferring information from one level of processing to another level.

- System integration:
  Vision chips may comprise most modules, such as image acquisition, and low level and high level analog/digital image processing, necessary for designing a vision system. From a system design perspective this is a great advantage over CCD camera-DSP option.

On the other hand, a single-chip imager systems has its own shortcomings and limitations:

- Sensitive reliability of processing
- A low fill-factor and a low resolution
- Difficulty of the design, time consuming and error-prone
- Less Programmable ability

2.4.2 Spatio-Temporal Image Processing Vision Chips

Recently, new technology and process are introduced to implememt variety of vision chips. The typical processes are:

- CMOS
- BiCMOS, i.e. pnp-type bipolar transistor(high current driving capability) jointly with CMOS transistor(low static power consumption)
- GaAs MESFET, HEMT, etc.

The typical vision chip for spatio and spatio-temporal image processing are:

- Mead's adaptive retina
- Standley's orientation detection chip
- Moini et al.'s insect vision-based motion detection chip
- Meitzler et al.'s 2D position and motion detection chip
- Etienne-Cummings et al.'s Motion Detector Chip, etc.

which are implemenmted by the above new processes.

But, research shows that there are some problems in them when using these new processes, such as those:

- Complexity
- Mismatch
- Digital noise

The details about these subjects are fully discussed in the references $[9], [10], [18], and [26]$.

2.5 Spatial Processing with Linear Resistive networks

2.5.1 Resistive 2D Gaussian filter

A large number of vision chips, either spatial or spatio-temporal, require processing of information within a neighborhood. Resistive networks have been known as a method of providing local interaction between cells with minimum requirement in terms of space and interconnection. Although general network theories are very helpful in understanding the type of functions realizable using resistive networks, it is in general very difficult to find a resistive network suitable to a specific problem. There are a number of resistive networks that have been fully analyzed and characterised.

1. Linear Resistive networks

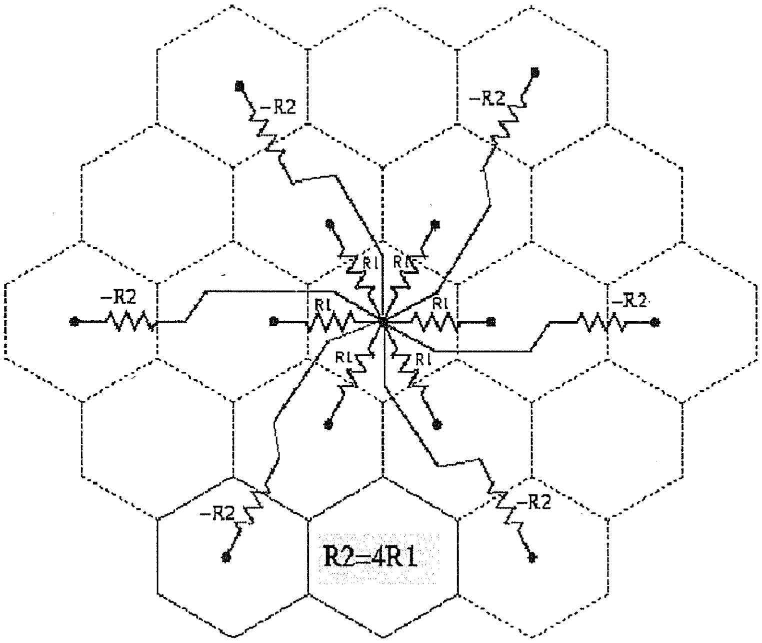


Figure 2.3: resistive 2D mesh network using negative impedance converters for implementing a Gaussian filter

The exponentially decaying smoothing function is not generally used in image processing algorithms, but as it is the simplest network providing spatial smoothing, it has been preferred over more complicated and area consuming networks, for example the Gaussian filtering in VLSI implementations. The errors due to device mismatch are usually prevalent and exceed the difference between an exact Gaussian function and an approximated exponential function. Therefore, it

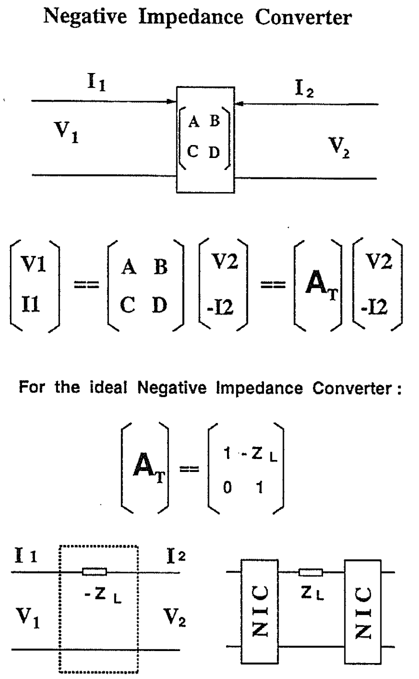

Negative Impedance Converter
I1
I2
V1
V2
A B
C D
V1 I1 == A B C D V2 -I2 == A_T V2 -I2
For the ideal Negative Impedance Converter :
A_T == 1 -Z L 0 1
I 1
I2
-Z L
V1
V2
NIC
Z L
NIC

Figure 2.4: NIC scalling transformation. When a negative resistor is realized by NIC, the circuit size and sensitivity are two big factors for the VLSI integration.

is rather unnecessary to use more accurate approximates, unless the specific algorithm heavily relies on the shape of the function, and device mismatch can be controlled within the desired range. The resistive network can be extended to two dimensions.

Kobayashi et al[11] used a hexagonal resistive network, shown in Fig. 2.3, realised a Gaussian filter, which has a 45 × 40 array of photodetectors and resistive grid on a 7.9 × 9.2mm chip, using a 2 μ m CMOS process.

2. NIC Theory and Negative Resistive networks

   Linvill started the use of NIC in active network application as early as in 1954. But in using of NIC may result in high sensitivity and unstable[12],[13]. The research of NIC has acquired many progresses recent years and it has developed a variety of usages in active filtering as well as image processing. As a review, the fundmentals of NIC scalling and transformation are shown in Fig. 2.4.

   For the purpose of image detection and smothing, the architecture of an active resistive mesh containing both positive and negative resistors to implement a Gaussian convolution in two dimensions is used in vision chips. Kobayashi et al's 2D

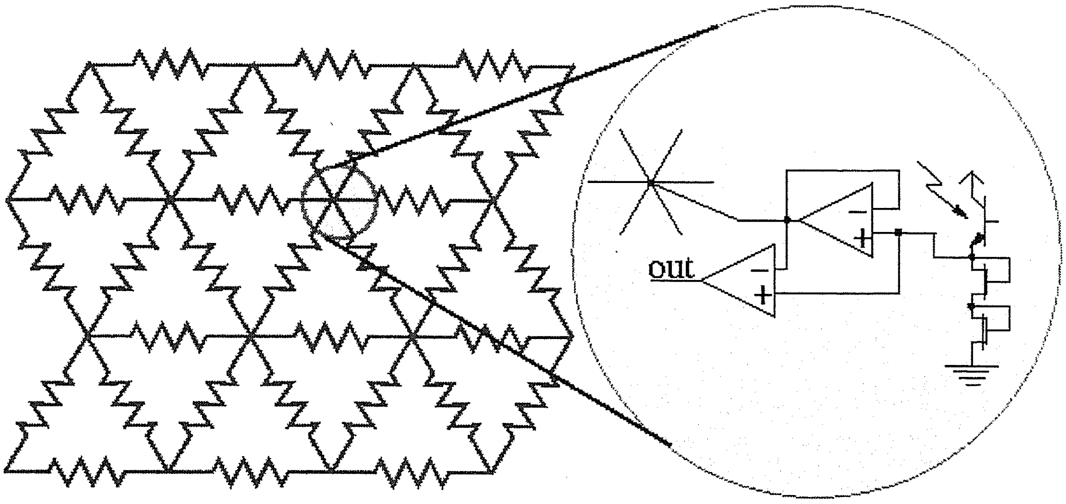


Figure 2.5: Architecture of Mahowald and Mead's silicon retina.

mesh with negative resistors between second nearest neighbores, shown in Fig. 2.3, illustates a typical example of the application of negative resistors in this area. The realization of the negative resistors is generally by NIC. But the simplification to the general NIC is neccessary if we try to realize smaller size, low power dissipation Gaussian filters on the focal plane.

2.5.2 Mahowald and Mead's Silicon Retina

Mahowald and Mead's silicon retina chip[9] is one of the first vision chips which implemented a biological facet of vision on silicon. This retina is based on models of computation in distal layers of the vertebrate retina, which includes the cones, the horizontal cells, and the bipolar cells. In this silicon retina the cones have been implemented using parasitic phototransistors and MOS-diode logarithmic current to voltage converters. In those implementations two separate smoothing networks with different smoothing constants are used. The corresponding outputs of the two smoothing networks are then compared using a differentiating function, such as division or subtraction.

2.5.3 Central region of foveated CMOS Retina

A foveated CMOS vision chip designed by IMEC[19] is shown in Fig. 2.6. The chip has a foveated rectangular region in the middle with high resolution and a circular outer layer with decreasing resolution.

The foveated image sesnor[25],[19] can be best utilized for robotic applications in which the low resolution periphery of the fovea finds areas of interest, and then directs the foveated part to get the details of those areas.

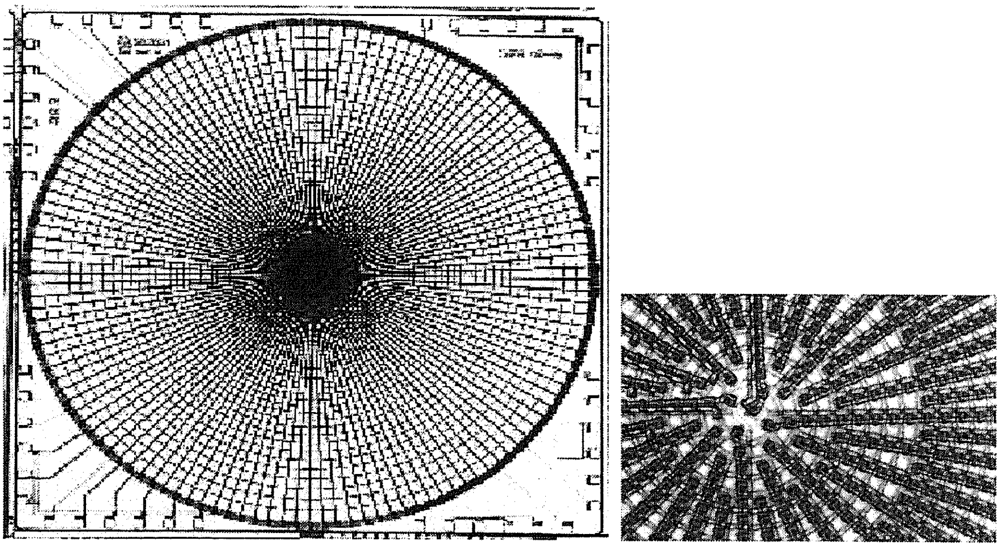


Figure 2.6: Photograph of the foveated CMOS retina and its central region enlarged.

2.6 The Combination of CCD Camera with DSP

The traditional imager system is a combination of CCD camera with a micro-processor or DSP, like the upper part of Fig. 2.7 shows. Although it can fulfil some complicated image processing tasks, the processing speed of the CCD-DSP system is limited not only by the charge accumulation speed of the CCD device but also limited by the micro code and access speed of the DSP.

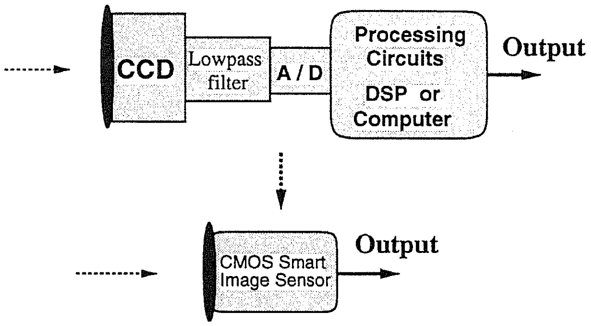


Figure 2.7: CCD camera v.s. CMOS Image Sensor

For instance, if the DSP TMS320C30 is used for processing of a methematical function with N instruction codes. Since the instruction speed of TMS320C30 is about 40 ns Instruction Cycle Time for floating point operation, and let's suppose that the fast memories are used for accessing data with a speed of 30 ns as well as M memory cells are needed for storage of these data. Then the time needed for fulfilling this methematical function will aproximate:

$$t = 40N + 2 \times 30M + t_{overhead} \quad [ns] \tag{2.3}$$

Or, if the DSP system adopts the DMA (direct memory access) mode, then the time will be

$$t = max\{40N\ ,\ \ 2 \times 30M\} + t_{overhead} \quad [ns] \tag{2.4}$$

For simplicity, in the above analysis we only have considered the core operations of the DSP. We ignored the speed of the off-chip A/D or D/A, but sometimes the A/D or D/A conversion would be an other fatal factor which limits the speed of the system.

While on the other hand, the CMOS imager system is entirely dependent on the integral time limitation of the photo diode (reference to Appendix A) and the speed of switching or transition of the transistor itself. The design and implementation of the prototype vision chip proposed by us in this thesis will fully base on the analog/digital mixed processing technology at the CMOS transistor level. So that it can perform the fastest signal processing tasks on the vision chip's focal plane.

Chapter 3

FOCAL PLANE PROCESSING FOR MOTION DETECTION

In this chapter, we will study some of the typical focal plane motion estimation methods and propose our own one for motion estimation on the vision chip's focal plane.

3.1 Correlating Motion Detection

Tanner and Mead[14] implemented a vision chip with the ability of motion detection in NMOS process. And then other groups[15],[24] developed some similar work in this area. It uses correlation to determine the direction of motion. Phototransduction is performed by an integration based photocircuit using a photodiode. The image at one sampling time is digitized to a one bit image pattern and stored in latches. It is then correlated with the analog signal detected in the next sample using the architecture shown in Figure 3.1. The digitization of the image to one bit in the latch branch is done for simplifying the design. Multiplication is performed using a simple current mirror switched by the output of the latch. The currents are summed by hard-wiring the outputs of all current mirrors.

When there is a right movement, the Right Move Correlation unit can detected a peak. If there is no movement, the Unmoved Correlation unit shows an output of peak. And when there is a left movement happened, the Left Move Correlation unit detects a peak.

Therefore, this motion detector can only detect 1D motion.

It was reported[14] that a one dimensional array of this motion detector chip has been fabricated by using a 4 μm NMOS process in a 5.7mm × 1.73mm die.

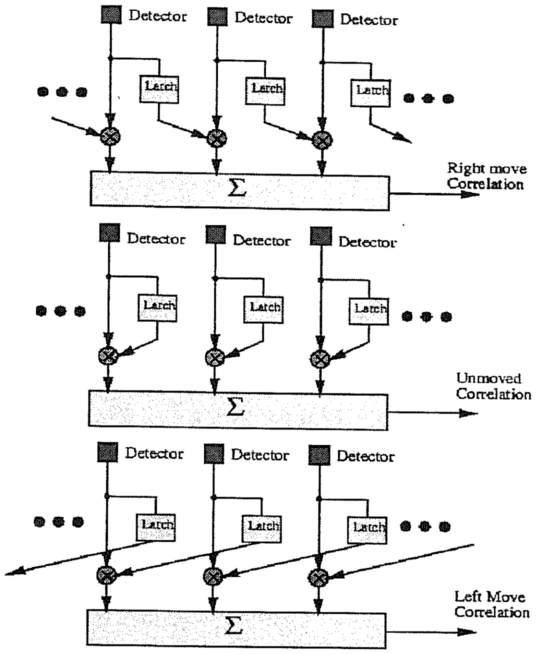


Figure 3.1: Architecture of Tanner and Mead's correlating motion detector.

3.2 2D Motion Detection by Optic Flow

For a two demensional image, its intensity surface can be depicted by Fig. 3.2.

The expression that ralates the intensity derivatives to the velocity can be described by the optic flow equation:

$$\frac{\partial I}{\partial t} = -\frac{\partial I}{\partial x}v_x - \frac{\partial I}{\partial y}v_y \tag{3.1}$$

This optic flow equation can be globally solved by feeding back the error

$$e = D \cdot \frac{\nabla I}{\nabla I} \tag{3.2}$$

where

$$D = -\frac{\frac{\partial I}{\partial x}v_x + \frac{\partial I}{\partial y}v_y + \frac{\partial I}{\partial t}}{\sqrt{(\frac{\partial I}{\partial x})^2 + (\frac{\partial I}{\partial y})^2}} \tag{3.3}$$

Horn, Schunck and Hildreth detailly studied this optic flow equation in mathematics, and then Tannaner, Mead and Wyatt[9],[16] implemented it on the vision chip by CMOS technology.

As it is known that for binary-valued images, there is an aperture problem as shown in Fig. 3.3. A binary-valued image containing a single straight edge is moving at some

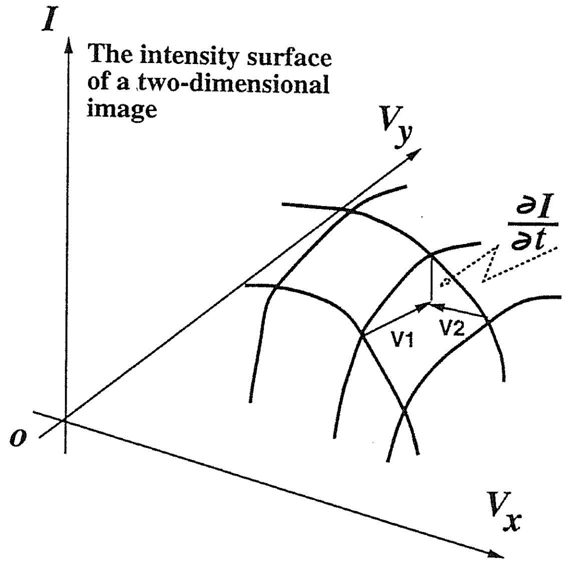

I
The intensity surface of a two-dimensional image
V_y
$\frac{\partial I}{\partial t}$
V1
V2
o
V_x

Figure 3.2: The intensity surface of a 2D image. The change in intensity from a present high value to a future lower one, ∂I / ∂t, could be the result of many possible motions of the intensity surface. The arrow V1 and V2 represent 2 possible motions.

velocity, the postition of the edge at later time is shown by the dashed line. The velocity connot be uniquely determined from these two snapshots in the views of square aperture.

There is an infinite family of possible velocities, as illustrated by the arrows. The image velocity components v_x and v_y can be viewed as the x and y coordinates in a velocity plane.

In this plane, the actual velocity of the image defines a point. The family of possible image velocities defines a constraint line in velocity space that has the same orientationas does the edge in physical space. To be consistent with the visual information from the local aperture, the actual velocity point is constrained to lie on the line in velocity space.

In the optic flow motion velocity detection, the ambiguity of a single local set of measurements is solved by using an other set of local values from a nearby location. These values define another line in the velocity plane. As shown in Fig. 3.4, the intersection of these two lines uniquely determines the actual velocity.

The designed chip contains phototransistors, and temporal and spatial differentiation circuitry for computing the spatio-temporal gradients of the input image.

The spatio-temporal information is collectively computed across the chip. The output of the chip is a global signal indicating the motion flow of the whole image, as showm in the left side of Figure 3.5.

And the right side of Fig. 3.5 shows the block diagram of each motion processing

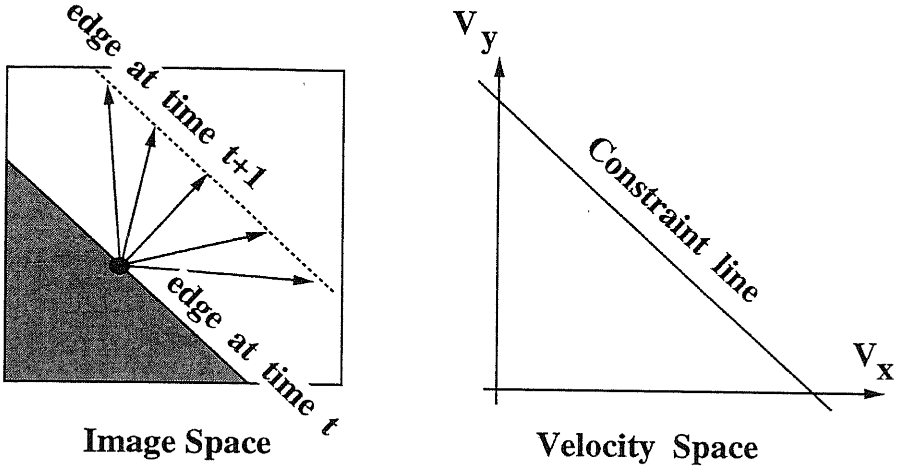


Figure 3.3: According to Professor C. Mead[9], there is aperture problem in the 2D velocity detection that the local information is not sufficient to determine 2D velocity uniquely. The family of possible velocities, denoted by the arrows in image space, define the constraint line in velocity space. The constraint line has the same orientation as does the edge in physical space.

element in Fig. 3.5.

Therefore, the chip is only capable of reporting a global motion. That is an imperfection of this method. We will solve this shortcoming by using a method of small size block matching.

Despite the imperfection, since it is one of the first vision chips being realized, the significance is this chip demonstrated and proved that:

the low level vision processing based on mathematical algorithms can be implemented in VLSI.

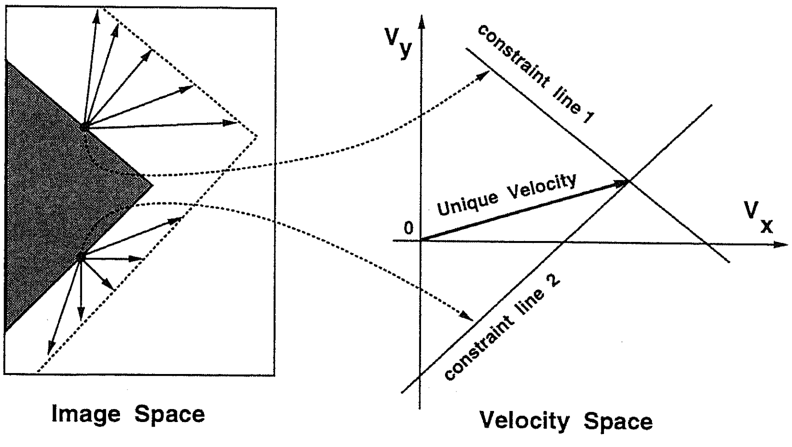


Figure 3.4: Uniquely determines the actual velocity by intersection of constraint lines.

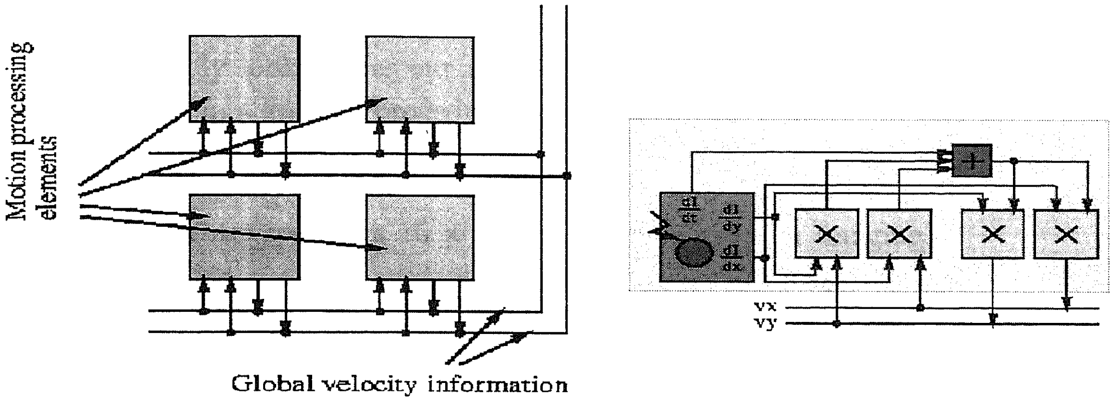


Figure 3.5: Left side: Architecture of the optic flow motion detector.(b)Right side: Block diagram of the motion detector elements.

3.3 2D Motion Vector Detection by Block Matching

Although block matching is a typical method for motion vector detection, focal plane processing need special architecture and algorithm arrangement[20],[28],[29],[31].

For a high frame rate imaging system, we can apply a small size block matching method to estimate the 2D motion vectors not only globally but also locally on the focal plane. We proposed and designed the block matching method on our vision chip based on the minimum-absolute difference(MAD) criterion:

$$D_{x,y} = \sum_{i=-m}^{m} \sum_{j=-n}^{n} |T(x,y) - S(x+i, y+j)|. \tag{3.4}$$

$D_{x,y}$ is the Hamming distance when T and S are binary images, which also represents the similarity between a block pattern T in present frame and a shifting block S in the previous frame. $(-m \leq i \leq m; -n \leq j \leq n)$ is the searching area around the position (x, y).

The size of the block matching will be a big factor which influences the design on the focal plane.

1. Increasing the size of the block matching will increase the complexity of the design and decrease the precise of the motion vector detection in a local place.

2. Decreasing the size of the block matching will decrease the complexity of the design and increase the precise of the motion vector detection in a local place.

We adopted the later by using a 2 × 2 pixels' small size of block matching in a $(\pm 1, \pm 1)$ local searching area.

We once briefly looked over our method in Chapter 1, here for a comparation, we describe it again in a specific procedure, as shown in Fig. 3.6.

1. Imaging:
   Our first prototype has a 16 × 16 photo diode array for imaging.

2. Motion Tracking by Edge Information:
   A 16 × 16 edge detectors parallel designed together with the photo diode array is used. Each pixel has an edge detector which gets the horizontal and vertical edge in different time slot. A 4-bits memory, which holds the horizontal and vertical edge information of current and previous frame, is also rparallel designed together with the pixel and the edge detector. The binary edge information is devided into two main groups, i.e. those for current frame's edges and those for previous frame's edges in a tree structure.

3. Block matching for motion vectors:
   The edge image information will be readout when it is needed for the correlative block matching. A 2 × 2 edge information in the current frame is used for performing a block matching operation with the corresponding blocks of $(\pm 1, \pm 1)$ 1 search areas in the previous frame by a Local pixel (PD-edge-memory) Parallel and Global Collum Parallel(LPGCP) processing architecture.

After an arbitration optimization, the motion vector index among {0, 1, 2, ...,8 } is taken as the output of a best matched vector.

By using a method of small size block matching, we can detection not only the global motion vectors but also the local motion vectors. This this the advantage of this method over those introduced in Section 3.1 and Section 3.2.

We will discuss the implementation of this motion vector detection method on a primary 16 × 16 pixels focal plane step by step in the following chapters.

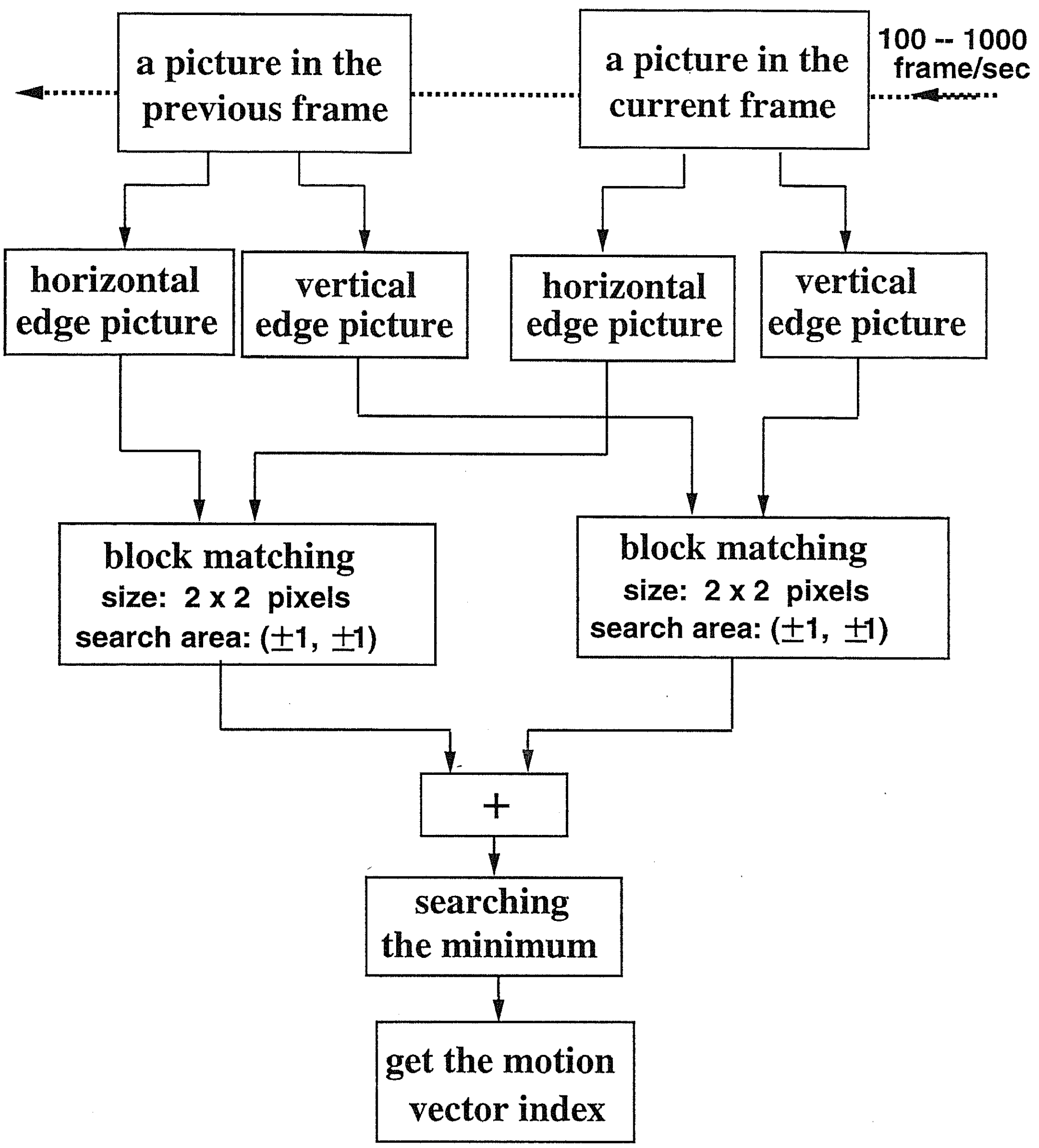


Figure 3.6: A motion vector detection flow with high frame rate imaging and small size block matching. First, the horizontal and vertical edges are abstracted from both the previous frame and current frame of images, then the edge images are binarized and stored into a 4-bits memory, finally a small size block matching is performed to search for the motion vectors on the whole focal plane.

Chapter 4

EDGE DETECTION AND BLOCK MATCHING ANALYSIS

4.1 Edge Detection Algorithms

As we know that the gradient has a large peak centered around the edge, by comparing the gradient to a threshold, we can detect an edge whenever the threshold is exceeded.

Since we know the edge occurs at the peak, we can localize it by computing the laplacian (the second derivative with respect to the time t) and finding the zero crossings.

There are several typical edge detection algorithms for image processings. Such as:

- 1. Roberts gradient method
- 2. Prewitt gradient method
- 3. Sobel gradient method
- 4. Marr-Hildreth method[22]
- 5. Wavelet transform method

The operaters for the first three methods can be represented by Fig. 4.1.

And the 4th one, pictured in Fig. 4.2, is a kind of 'Mexican hat' operators - also called 'difference-of-Gaussian' (DOG), which is based on an initial smoothing with a Gaussian lowpass filter window, to remove noise and spurious edges, and then apply the Laplacian, followed by detection of zero crossings.

Therefor, these first 4 operators can be grouped into two categories, gradient and Laplacian. The gradient method detects the edges by looking for the maximum and minimum in the first derivative of the image. The Laplacian method searches for zero-crossings in the second derivative of the image to find edges.

(a) Roberts Operator

1	0
0	-1

0	1
-1	0

(b) Prewitt Operator

-1	-1	-1
0	0	0
1	1	1

-1	0	1
-1	0	1
-1	0	1

(c) Sobel Operator

-1	-2	-1
0	0	0
1	2	1

-1	0	1
-2	0	2
-1	0	1

Figure 4.1: Typical gradient edge detection operaters(Roberts, Prewitt, Sobel).

In the 5th method, the Wavelet transform[26], the edge can be detected by observing the amplitue of the Wavelet transformed signal along its responding phase. The result of edge detection[26] is better then those of the direct gradient or Laplacian.

With the additional consideration of the hardware simplicity on the focal plane, in our prototype vision chip which will be described in Chapter 5 and Chapter 6, we use the algorithm of Equation (4.1) and (4.2) or the operator in Fig. 4.3 to get the horizontal edge and vertical edge by additionally using its absolute values and setting a proper threshold.

0	-1	0
-1	4	-1
0	-1	0

Figure 4.2: Mexican hat or DOG Laplacian operator(Marr-Hildreth).

$$
\begin{aligned}
Edge_{horizontal} &= f(i,j) - f(i+1,j) \qquad (4.1)\\
Edge_{vertical} &= f(i,j) - f(i,j+1) \qquad (4.2)
\end{aligned}
$$

1	0
-1	0

1	-1
0	0

Figure 4.3: An operator of edge detection which is used on our prototype vision chip by using its absolute values and setting a proper threshold.

By comparing Equation (4.1)/(4.2) or Equation (5.1)/(5.2) with Equation (B-10)/(B-11) in Appendix B, we can find that the simple edge detection method we adopted in our prototype vision chip is belong to the 5th method above.

4.2 Motion Tracking by Edge Information

Let us examine a moving object shown in Fig. 4.4. Its movement can be basicly tracked by its horizontal edge images and the vertical edge images as shown in the 3rd & 4th rows and 5th & 6th rows, respectively, in Fig. 4.4.

Among these pictures, we can examine two succesive neighboring pictures and their correspondance horizonatal edges and vertical edges in Fig. 4.5. It is obviously that the information contained in the edge image is greatly reduced than that in the original image.

Instead of using the whole original picture, edge image can be used for tacking the movement of a moving object when we put two succesive edge pictures together. Here in Fig. 4.6, picture 2 to 8 are the motion tendency of the edge image. The smaller tendency can be detected by a 2 × 2 block matching. This is the main reason why we will use a 2×2 block matching and a $(\pm 1, \pm 1)$ searching area for motion vector detection on our vision chip.

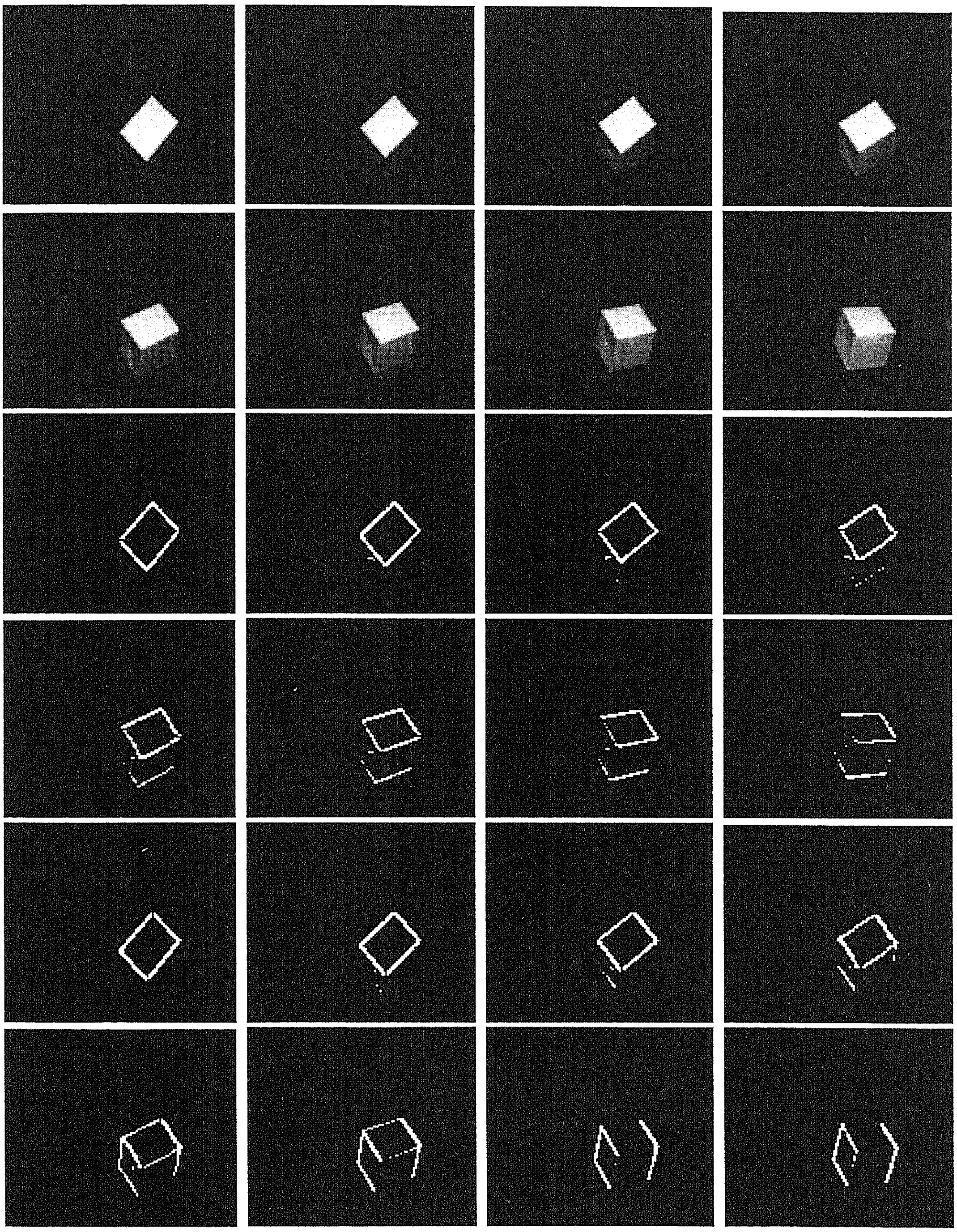


Figure 4.4: The sequences of 8 moving pictures and its 8 horizontal edge pic'ures(in the 3rd and 4th rows) and 8 vertical pictures(in the 5th and 6th rows), in the order of from the left to the right and from the top to the bottom.

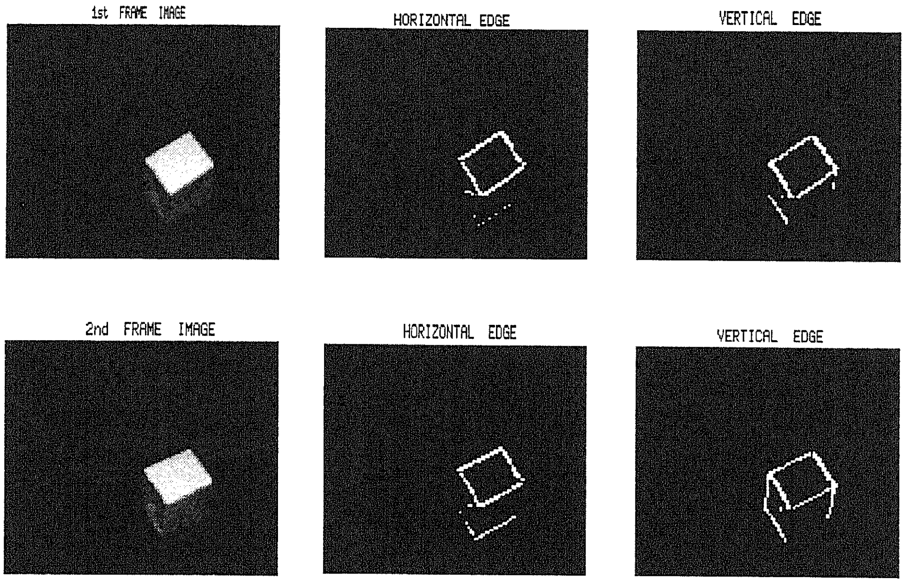


Figure 4.5: Two specific neighbouring moving pictures, and their conrespondence horizontal and vertical edge information which will be used for block matching.

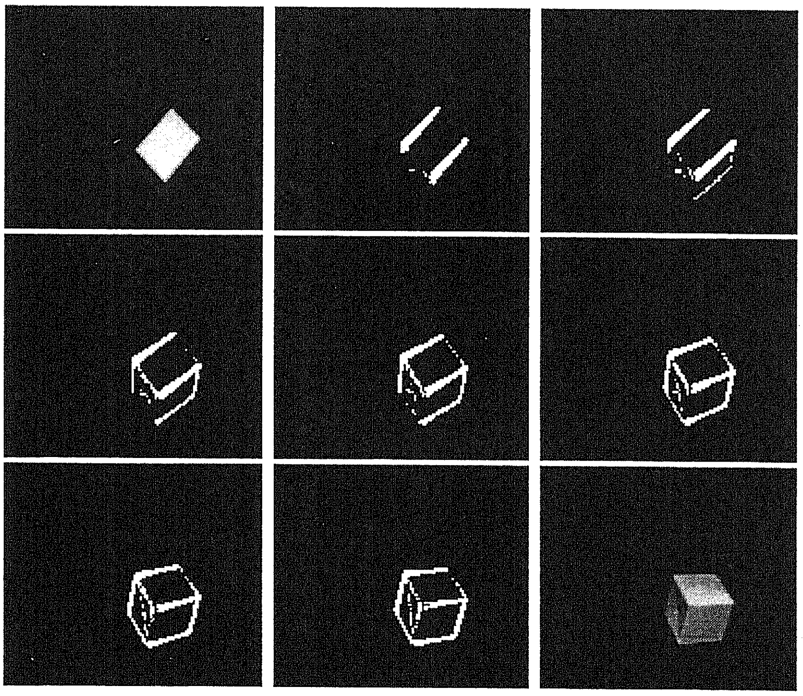


Figure 4.6: Moving image and its moving edge images, which include both horizontal and vertical edge information

4.3 Algorithm Analysis for 2D Block Matching

Since our purpose is to design a vision chip with strong computational abilities, we should study the relative algorithms and then choose the one which is suitable for designing on the focal plane.

4.3.1 Basic Theory of Matching

Suppose that each pattern class is represented by a prototype or mean vector:

$$\mathrm{m_j} = \frac{1}{\mathrm{N_j}} \sum_{\mathrm{s} \in \omega_\mathrm{j}} \mathrm{s} \qquad \mathrm{j} = 1, 2, ..., \mathrm{M} \tag{4.3}$$

where N_j is the number of pattern vectors from class ω_j and the summation is taken over these vectors. One way to determine the class membership of an unknown pattern vector s is to assign it to the class of its closest prototype. Using the Euclidean distance to determine closeness resuces the problem to computing the distance measures:

$$D_j(\mathrm{s}) = ||\mathrm{s} - \mathrm{m_j}|| \qquad \mathrm{j} = 1, 2, ..., \mathrm{M} \tag{4.4}$$

where $||a|| = (a^T a)^{(1/2)}$ is the Euclidean norm. We then assign s to class ω_j if $D_i(\mathrm{s})$ is the smallest distance. i.e. the smallest distance implies the best match in this formulation.

4.3.2 Fast Block Matching

The pattern of a rectangular block is always considered first in the image processing area. The basic principle of block matching in moving picture processing is using a proper size of rectangular block in the current frame of picture to perform a corelation operation or minimum mean-square error (MMSE) or minimum-absolute difference(MAD) detection on the other frame of picture, usually the previous frame. The direction and position of the relative block in the previous frame of picture is chosen as the motion vector if it is with the maximum value of corelation or with the minimum value of the MMSE. Generally, the operation in MMSE form can be calculated by

$$D_{x,y} = \sum_{i=-m}^{m} \sum_{j=-n}^{n} [T(x,y) - S(x+i, y+j)]^2. \tag{4.5}$$

For the purpose of a fast operation, the minimum-absolute difference(MAD) will be taken as the criterion for the block matching in our prototype. It can be writen as:

$$D_{x,y} = \sum_{i=-m}^{m} \sum_{j=-n}^{n} |T(x,y) - S(x+i,y+j)|. \tag{4.6}$$

$D_{x,y}$ is the Hamming distance when T and S are binary images, which also represents the similarity between a block pattern T in present frame and a shifting block S in the previous frame. $(-m \leq i \leq m; -n \leq j \leq n)$ is the searching area around the position (x,y).

4.3.3 Motion Vector Detection with a Small Size of Block Matching

The motion of a 2D object can be detected by the above block matching method either by their pixel values or by their binary values.

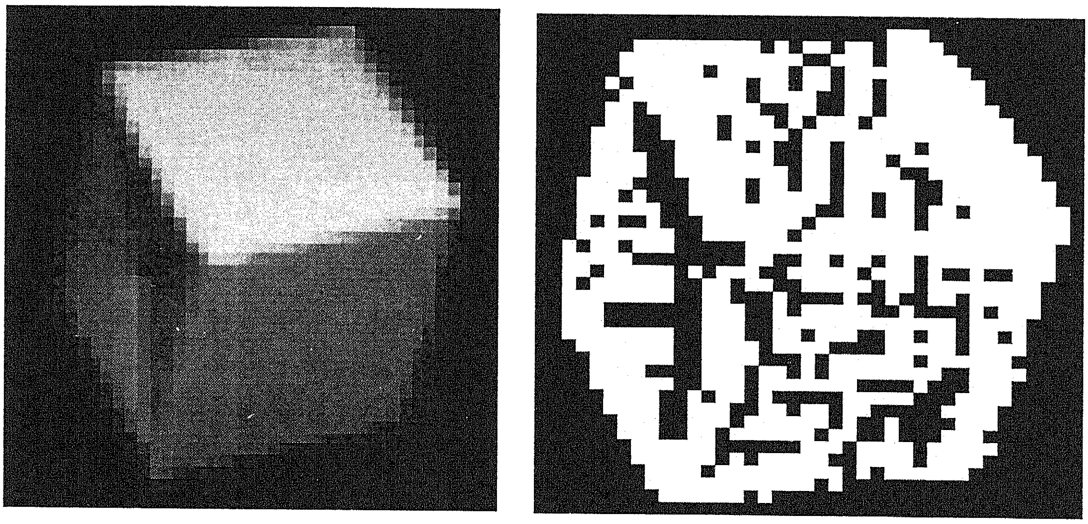


Figure 4.7: Left figure: a 2D moving object; Right figure: the motions detected by block matching with pixel values between 2 successive frames. The bright blocks indicate that motion vectors are detected.

Digital simulations in Fig. 4.7 shows the motion detected by using a block matching with the gray pixel values of the original images. The detected motion area is large and unclean since too much of the tiny details be detected.

Fig. 4.8 shows the motion detection and motion vector detection by using a block matching with the binary edge information instead of the pixels of the original images. We can see that it can get clean motion detections with clean motion dentency and obviously it greatly reduces the arithmetic and logic operations. This method is reasonable to be applied on the tiny focal plane with simple edge detectors and simple specific arithmetic and logic processors.

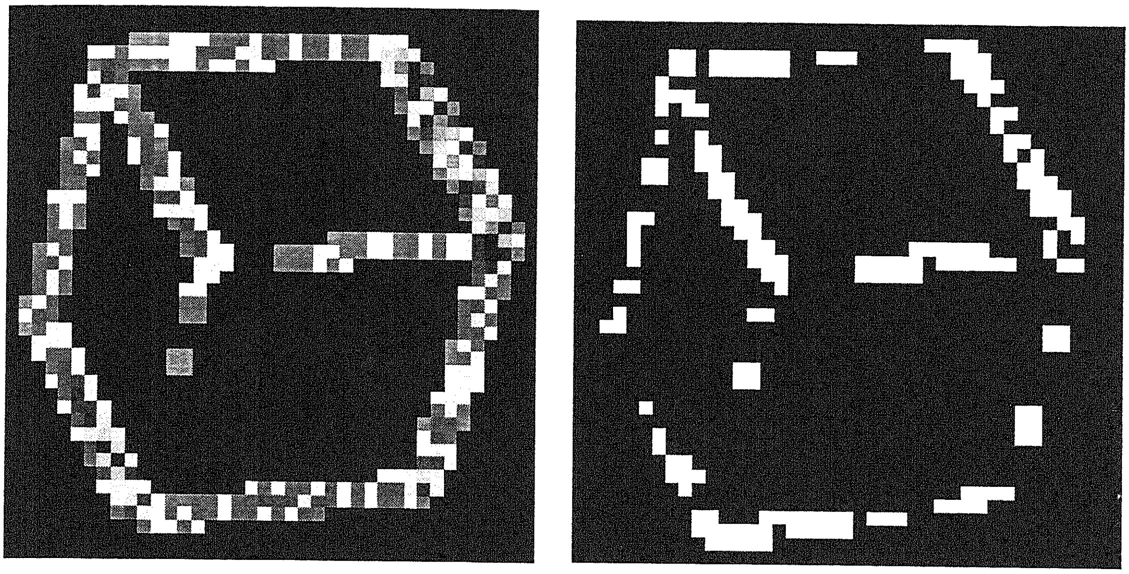


Figure 4.8: Left figure: Motion is detected between 2 successive frames by block matching with the binary values; Right figure: Motion vectors are detected between 2 successive frames by block matching with the binary values after the priority decision. The bright blocks indicate that motion vectors are detected.

If there is no general CPU and A/D converter integrated on the focal plane, the implementation of a motion vector detection by block matching with the gray pixel values is almost impossible. Even with a general A/D converter integrated on the focal plane, a Full Search Block Matching Algorithm(FSBMA) [2] is an extreamly time consuming task of implementation of the Equation (4.6) since $N^2 \times (2 \times m)^2$ computations are needed for $D_{x,y}$. Here we suppose that there are $N \times N$ pixels on the focal plane, and $m = n$.

By using of a block matching with a small search window of 2×2 pixels and a small search area of $(\pm 1, \pm 1)$ pixels around the intentinal area, we can realize a 2D motion vector detection in 9 directions within a step of 1 pixel in length. As Fig. 4.9 shows that the motion vectors with any angle and length can also be obtained if we estimate the motion vector contineuesly in its following successive frames.

Comparing with the optical flow method, we can conclude that the block matching method not only can detect the global motion/motion vectors, such as the movement of the borders or boundary lines, but also can detect the local motion/motion vectors, such as the movement of the points within the borders of the 2D object. It is always effective no matter what the block size is, large or small.

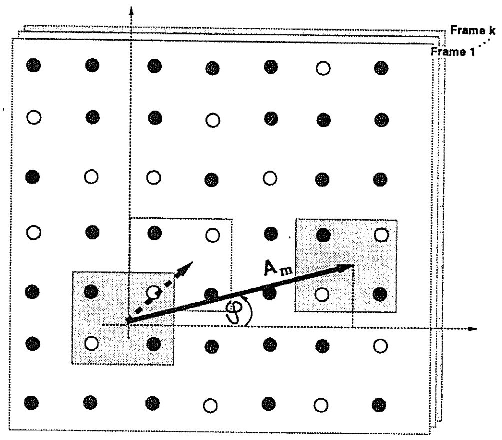
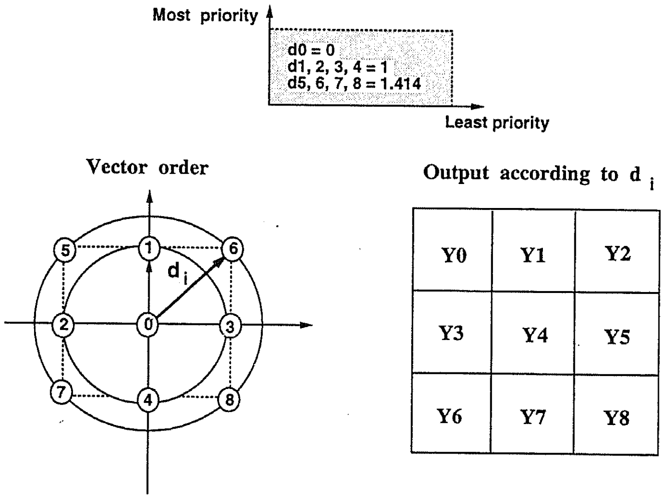


Figure 4.9: Motion vector: its amplitude and direction on the sensor plane. Motion vectors with any angle and length can be estimated if we trace the motion vector contineuesly in its following successive frames.



Y0	Y1	Y2
Y3	Y4	Y5
Y6	Y7	Y8

Figure 4.10: The priority dicision for the motion vector output.

4.3.4 Priority Decision of the Small Size Block Matching

In the above digital simulation, we chose the block pattern T and S of a size 2×2 pixels and a searching area of 4×4 pixels with 2 kinds of different information standing for horizontal and vertical pixels. 9 blocks of 2×2 pixels produce 9 candidates { Y0, Y1, Y2, Y3, Y4, Y5, Y6, Y7, Y8 } after the local area block matching. Among these candidates, the most smallest one is chosen as the vector, but when several same minimum values come up, an arbitration logic is needed to select the one as the final vector, which has the shortest distance d_i to the center of the searched area in the order of priority shown in Fig. 4.10. This arbitration logic ensures a unique vector can be detected.

Chapter 5

CIRCUIT DESIGN OF THE EDGE DETECTION, MEMORY AND MOTION VECTOR DETECTION ON FOCAL PLANE

In this chapter, we put forward some specific circuits for the edge detection and motion vector detection and give out the designs for the focal plane processing.

The pixel which includes one photo diode, one edge detector and 4-bits memory cell will be designed in a compacted local parallel structure, as shown in Fig. 5.1, This structure benifits for data transition among the sensor, the edge detector, the memory and as well as the data feeding to the block matching processor.

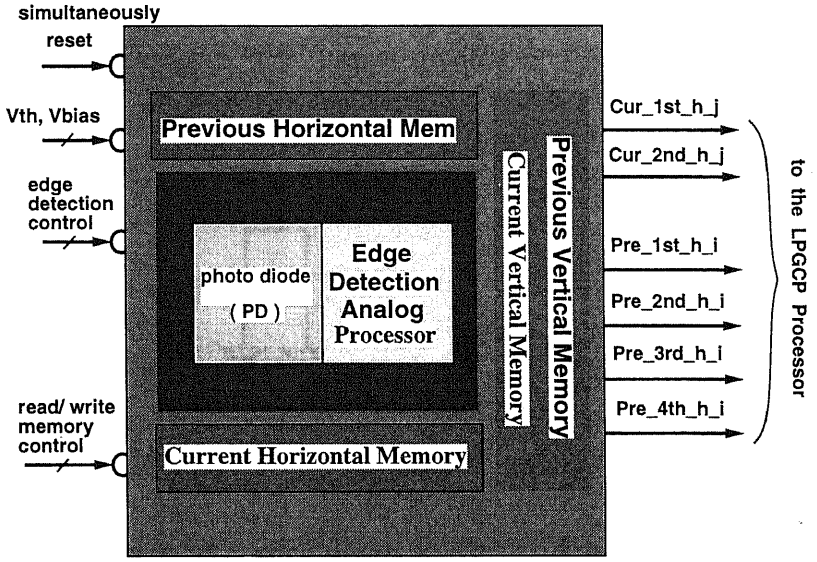


Figure 5.1: Compacted parallel structure for PD, edge detector and 4-bits memory

5.1 Edge Detection and the MOS Circuits

The gereral method of obtaining the edge information of image is to estimate the Gradient or Laplacian of the concerning images, as in Chapter 4 described. For an image pixel represented by $f(x, y)$, its first partial derivative along x axis and y axis can be represented by $\frac{\partial f}{\partial x}$ and $\frac{\partial f}{\partial y}$, the digital form can be approximated by Equation (4.1) and (4.2), i.e. the first order differential operation between 2 neighboring pixels in the directions of horizontal and vertical, respectively.

Edge detection Equation (4.1) and (4.2) should be changed into the form of Equation (5.1) and (5.2) when we try to implement them on the focal plane by MOS circuit.

Equation (5.1) and (5.2) are the absolute forms of Equation (4.1) and (4.2) with a proper threshold "Thr". Therefor, we can choose some absolute value differenciator circuits for implementation of these Equations.

$$Edge_{horizontal} = \begin{cases} 1 & (|f(i,j) - f(i+1,j)| \geq Thr\ ) \\ 0 & (\text{otherwise}) \end{cases} \tag{5.1}$$

$$Edge_{vertical} = \begin{cases} 1 & (|f(i,j) - f(i,j+1)| \geq Thr\ ) \\ 0 & (\text{otherwise}) \end{cases} \tag{5.2}$$

5.1.1 Edge Detector with Differencial Pair

How to design the edge detector by VLSI has been attempted for years[21],[23]. Here after we discussed the edge detection algorithms, let's see how to design a proper edge detector for our prototype vision chip by the MOS circuit.

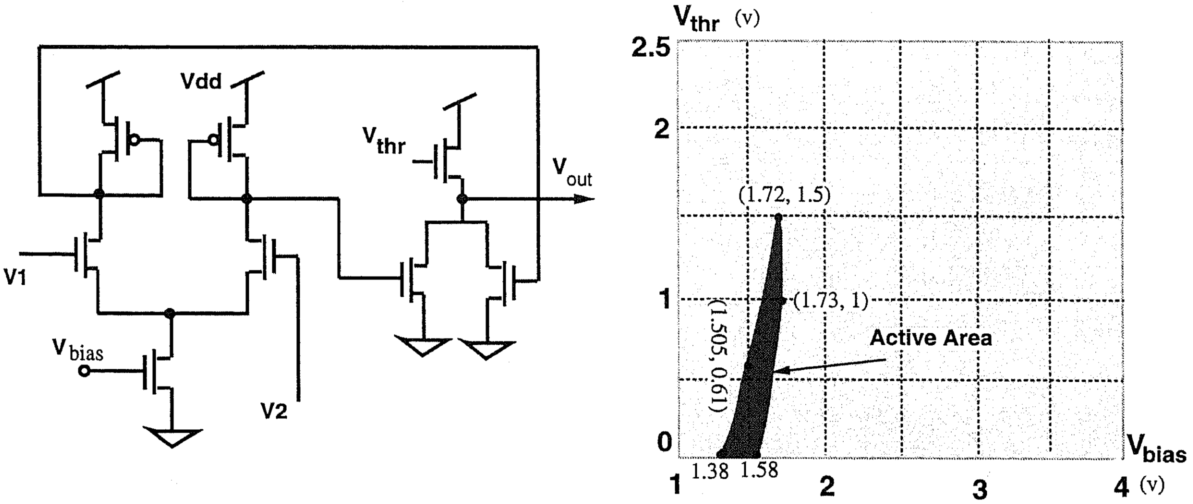


Figure 5.2: Edge detector with Differencial Pair

The left side of Fig. 5.2 shows an edge detector with a Differencial Pair structure[20]. The right side figure of Fig. 5.2 shows its typical dynamic working range for bias voltage

V_{bias} and threshold voltage V_{thr}. HSPICE simulation(19) shows that the dynamic range of this circuit is very small.

In the case of smaller range of V_{bias} and voltage V_{thr}, when the transistor's parameters of W/L has a slight change, the working point of the Absolute Value Differenciator might be out of the working range. This would lead to the failure of designing the edge detector.

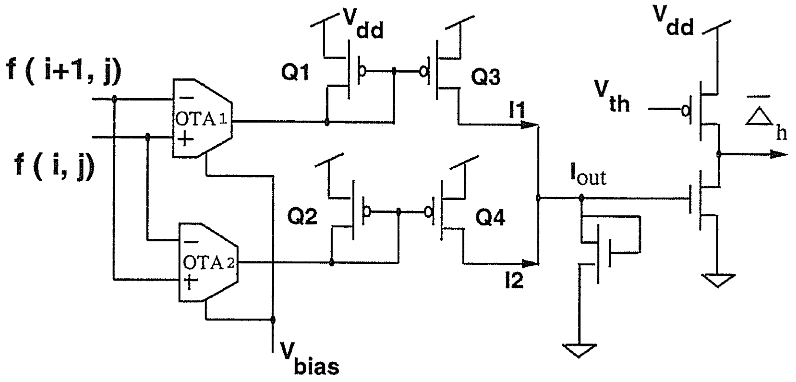


Figure 5.3: Edge detector with double OTAs.

5.1.2 Edge Detector with Basic CMOS OTA

An Absolute Value Differenciator(1),(3),(19) operating with current mode is structed in Fig. 5.3. By making use of 2 Basic CMOS OTA (Transconductance Operational Amplifier)[9], shown in the left side of Fig. 5.4, the dynamic working range for bias voltage V_{bias} and threshold voltage V_{thr} will be made large, as shown in the right side of Fig. 5.4. Searching for the parameter insensitive circuit is always a target of the circuit design.

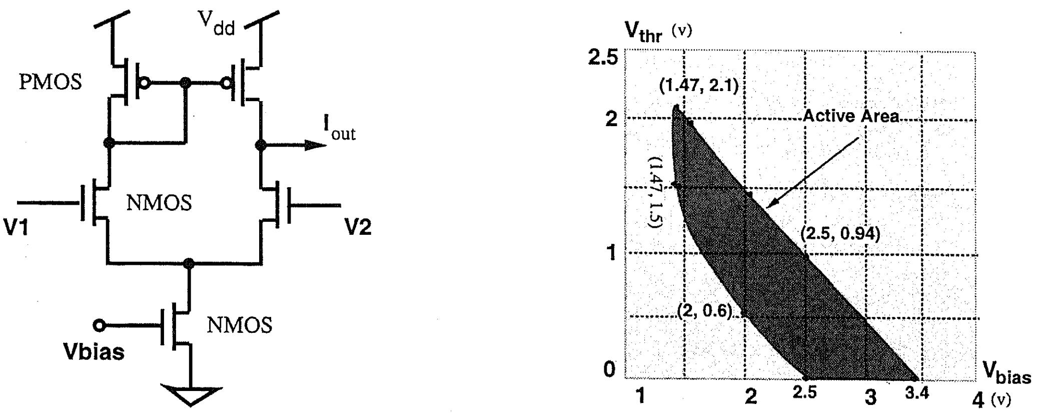


Figure 5.4: Basic CMOS OTA

5.1.3 Edge Detector with Symmetrical CMOS OTA

When we use the Symmetrical CMOS OTA[18], shown in the left side of Fig. 5.5, instead of the basic CMOS OTA, the dynamic range of the circuit input/output ralation can be from the min = 0 to the maxmum max =V$_{dd}$ = 5V. In the case of the wide dynamic range signal application, the Symmetrical CMOS OTA is prefered. While the range of bias voltage V_{bias} and threshold voltage V_{thr} reminds almost the same as that of the Basic CMOS OTA's.

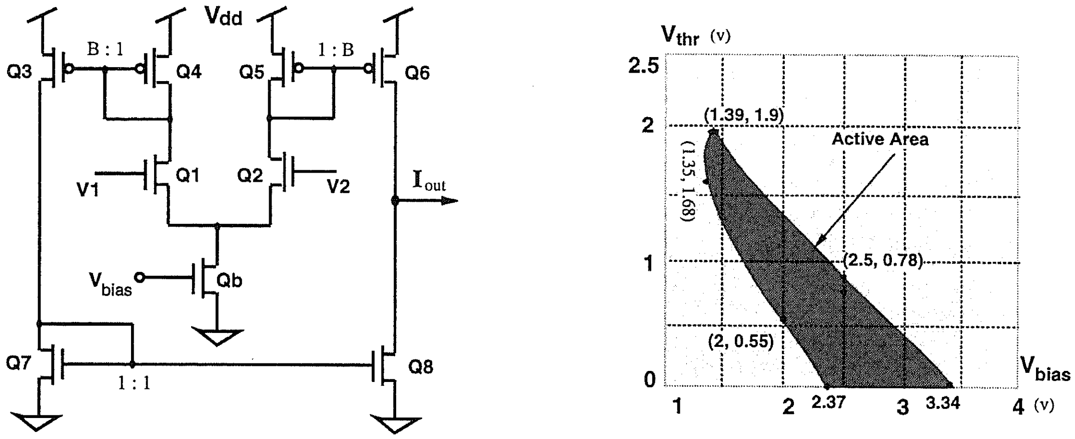


Figure 5.5: Symmetrical CMOS OTA

5.1.4 Comparison and Summary for the Absolute Value Differentiator

1. Differencail Pair:
   The edge detector by using of the Differencail Pair has a small circuit size and consumes smaller power dissipation, but the dynamic working range of V_{bias} and V_{thr} is very small which may introduce a high possibility of design failure.

2. CMOS OTA:
   In the double OTA edge detector structure, Basic CMOS OTA occupies large size than Differencail Pair, but with a smaller size than the Symmetrical CMOS OTA. The power dissipation is determined by the factor B of the Current Mirror.

3. Other points:
   HSPICE simulation in Fig. 5.6 shows that we can find a proper edge detection Symmetrical center with the double OTA edge detector. This is important if we want to produce a symmetrical waveform for both the input 1 is larger than input 2 and input 1 is less than input 2.

By making use of the Symmetrical CMOS OTA, we can obtain a workable range of V_{bias} and V_{thr} almost the same as that of the Basic OTA, and we can also get a

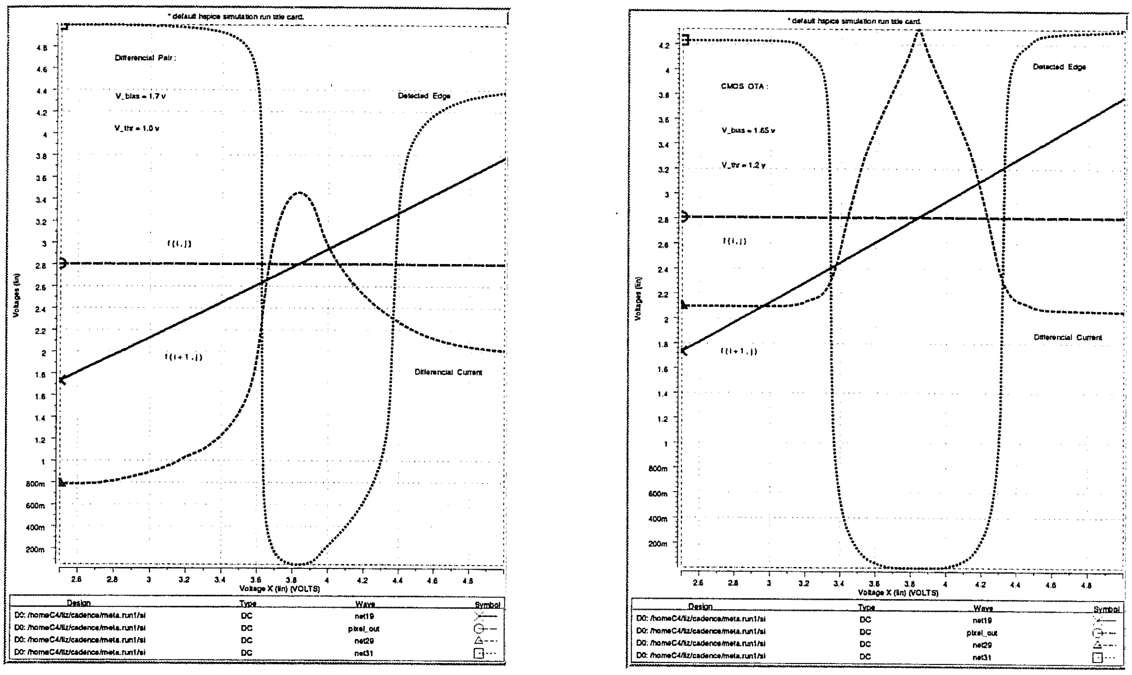


Figure 5.6: Detected edge and output current symmetry

wide dynamic range of the differential inputs and edge output. A summary of the characteristics of these edge detection circuits[18] is listed in Table 5.1.

Table 5.1: summary of 3 edge detection circuits

CIRCUITS	CIRCUIT SIZE	POWER DISSIPATION	EVALUATION
Differencial Pair	small	small	small bias range, small dynamic range
Basic OTA	midle	midle	large bias range, midle dynamic rang
Symmetrical OTA	large	large (depends on current Mirror factor B)	large bias range, large dynamic rang

5.2 Time Multiplexed Edge Detection on Focal Plane

We can use only one edge detector for obtaining both horizontal edge and vertical edge. In this way Equation (5.1) and (5.2) are changed into Equation (5.3) and (5.4) which are excuted in time multiplexed samplings at time $t1$ and $t2$, respectively. We have adopted this Time Multiplexed Edge Dedection (TMED) architecture, shown in Figure 5.7, in our protoype chips.

The detected horizontal and vertical edge information is then stored into memory arrays described later.

$$Edge_{horizontal} = \begin{cases} 1 & (|f(i,j) - f(i+1,j)| \geq Thr\ )_{t1} \\ 0 & (\text{otherwise})_{t1} \end{cases} \tag{5.3}$$

$$Edge_{vertical} = \begin{cases} 1 & (|f(i,j) - f(i,j+1)| \geq Thr\ )_{t2} \\ 0 & (\text{otherwise})_{t2} \end{cases} \tag{5.4}$$

5.3 Circuit Design for Sensor, Edge Detection and Memory

The circuit for 1 pixel, which includes 1 Photo Diode, 1 edge detector and 4-bits memory cell designed in parrallel, is shown in Fig. 5.8 and Fig. 5.9. i.e. the circuits in Fig. 5.8 and Fig. 5.9 consist of one pixel in our prototype design. In our prototype, the edge detector abstracts the edge information between 2 neighboring pixels f(i,j) and f(i+step,j) or f(i,j+step) with the *step* = 1. Of course, if we try to implement other edge detection algorithms, such as those in Fig. 4.1 or Fig. 4.2, different step length may be used.

Fig. 5.8 is an instruction for pixel connection, which can be applied to connect the single pixel circuit into a 16 × 16 or any N × N array on the focal plane.

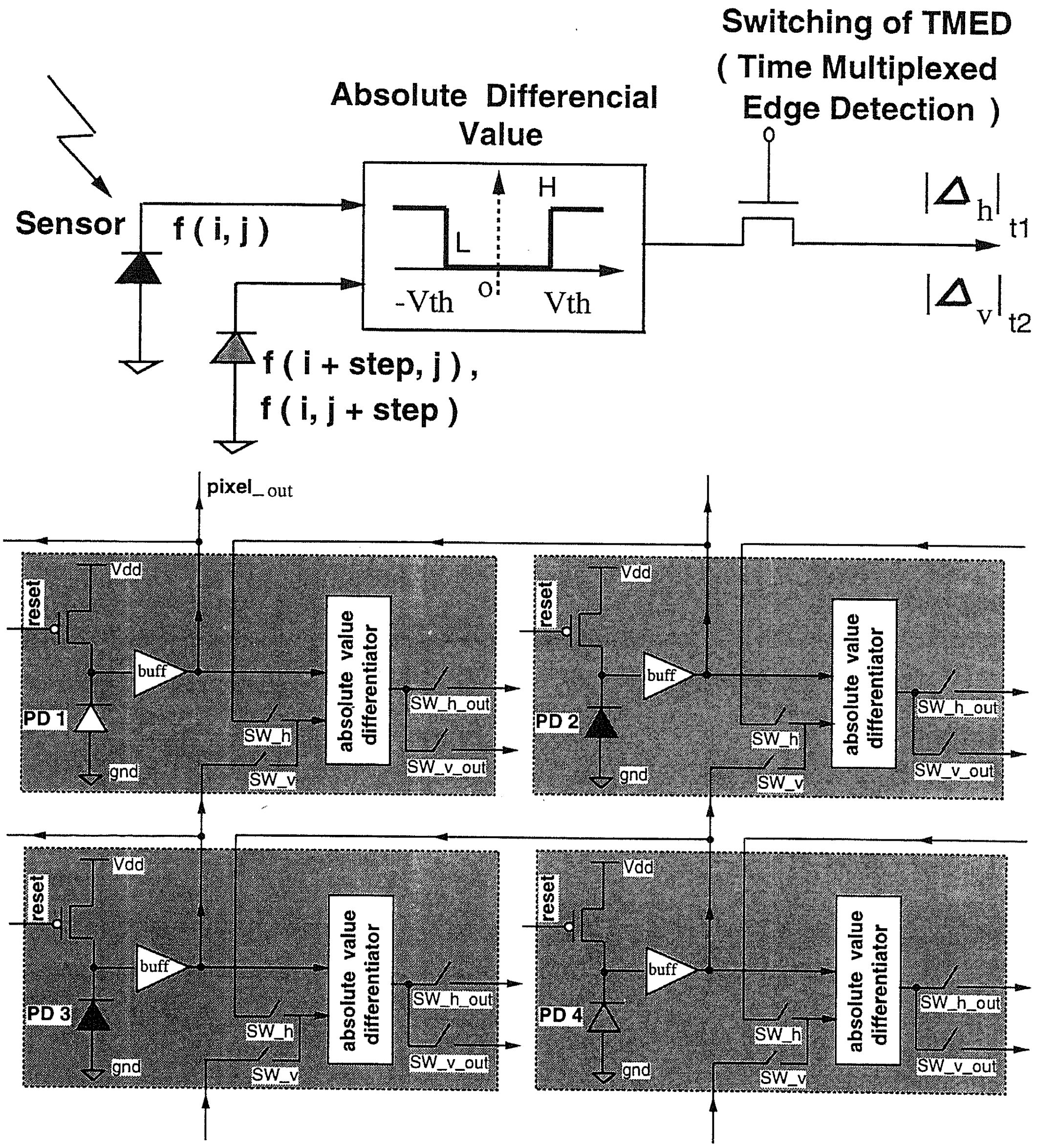


Figure 5.7: TMED - Time Multiplexed Edge Detection (step = 1). This circuit abstract the edge information between 2 neighboring pixels f(i,j) and f(i+step,j) or f(i,j+step). A threthold value Vth should be properly set for getting the edges.

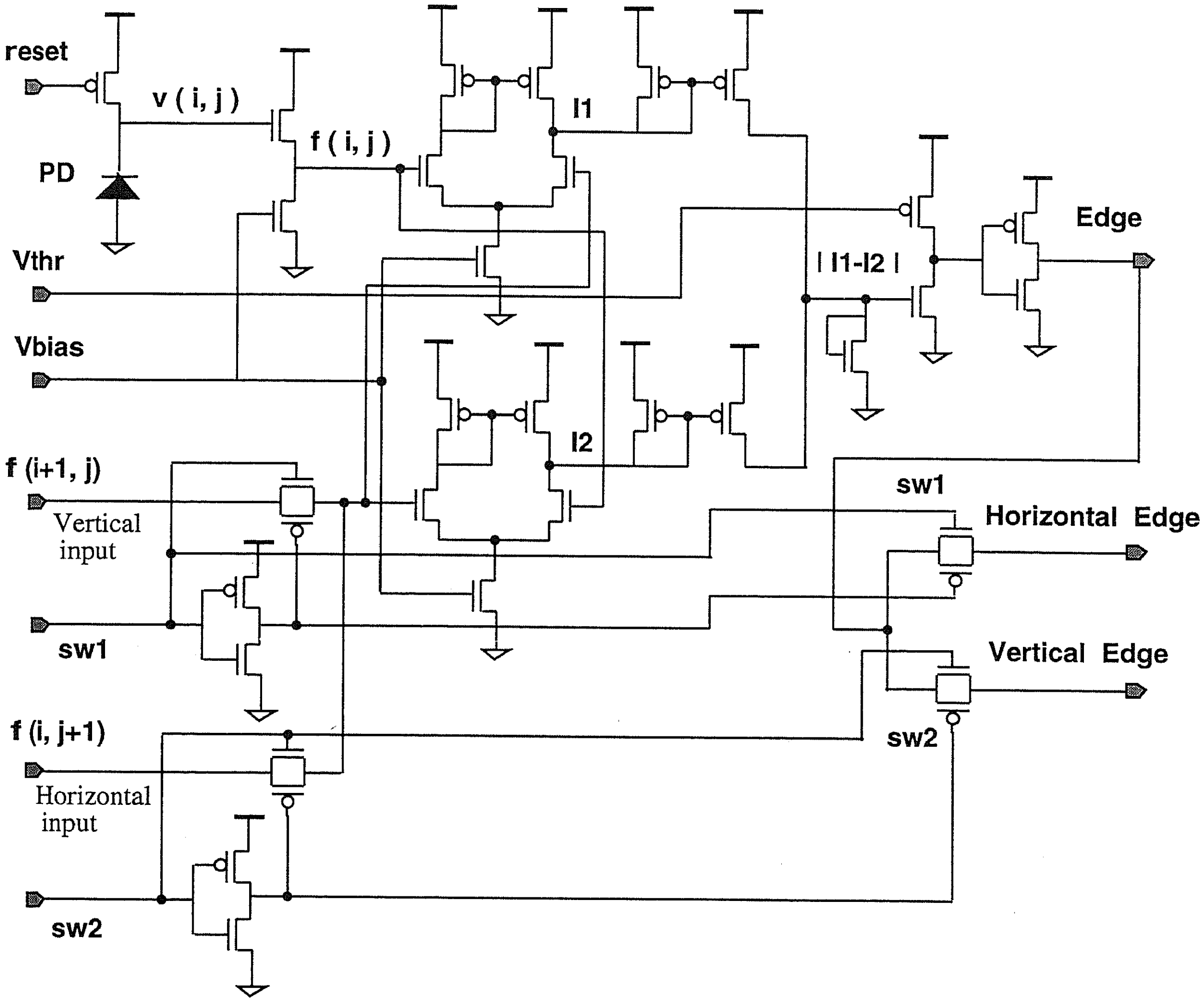


Figure 5.8: The Photo diode, edge detector circuit used in our prototype design.

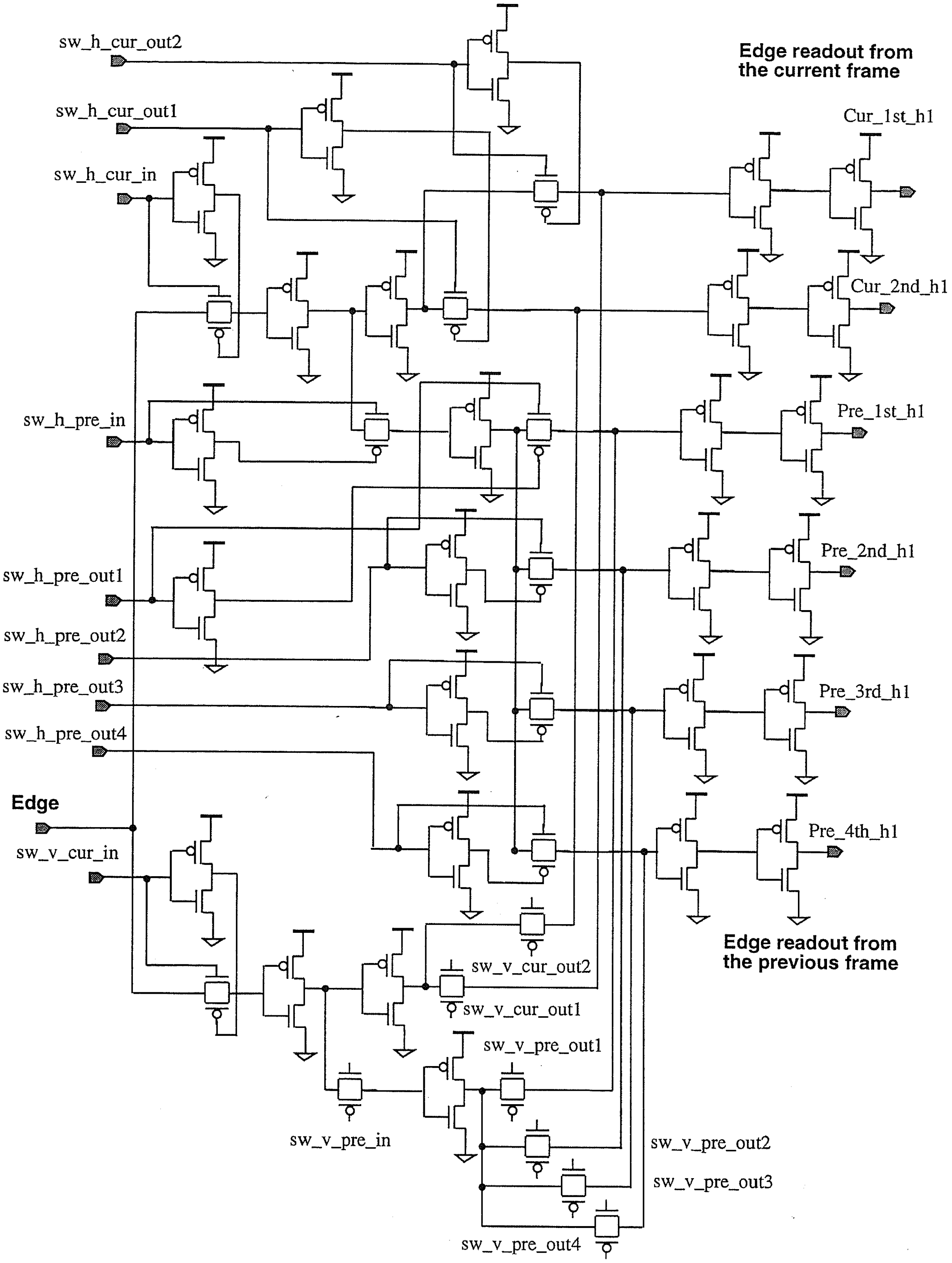


Figure 5.9: Memory cell circuit and its input/output control.

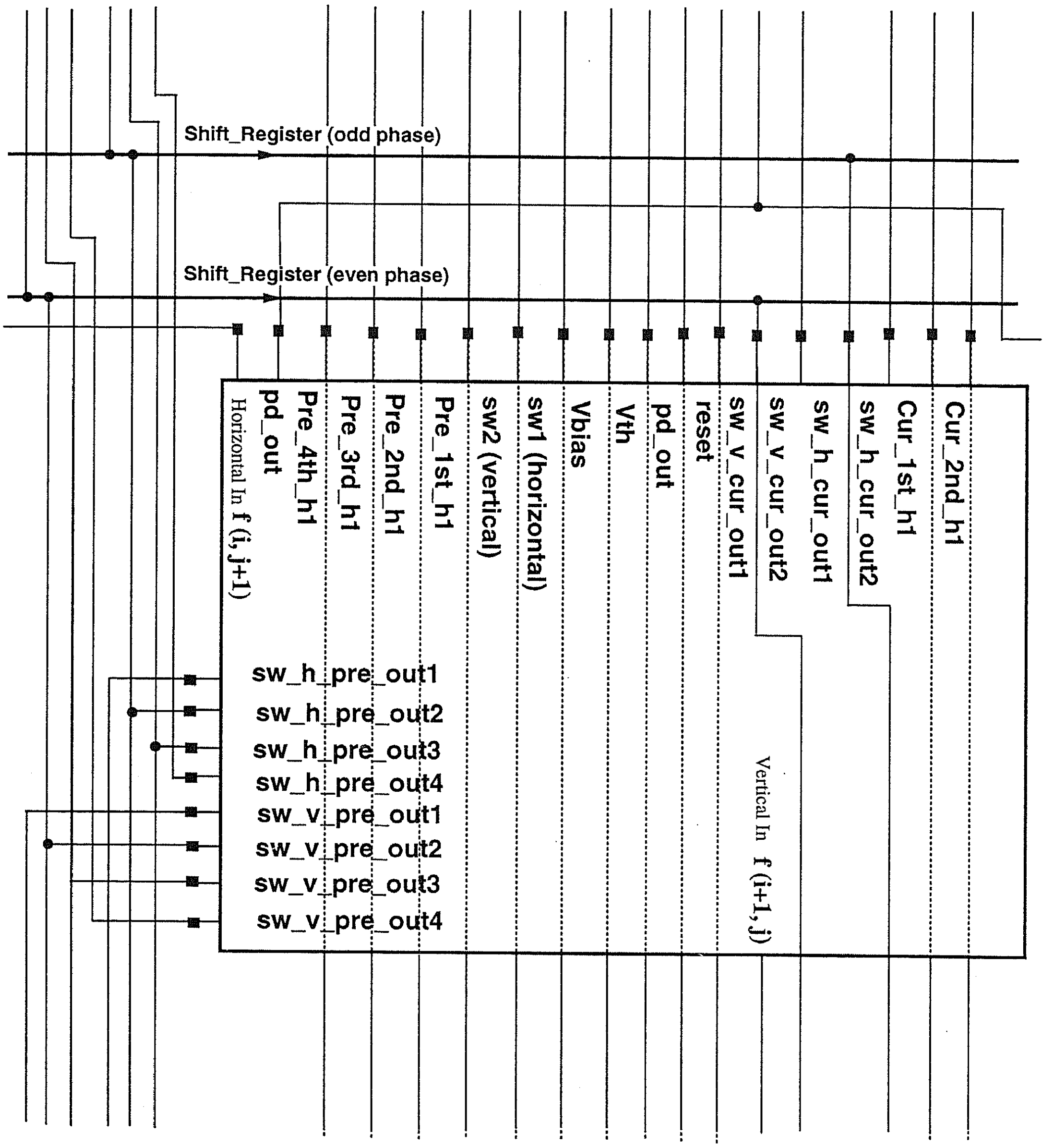


Figure 5.10: A connection diagram of the pixles. Every pixel includes a photo diode, a TMED edge detector, 4-bits memory cells and the controls. The signals' arrangement on this diagram appropriately helps to connect N × N pixels into an array on the focal plane in our prototype design.

5.4 4-Bits Transmission Gate Memory Array

Fig. 5.11 shows memory array arrangement for storage of 4 bits information of the horizontal edge and vertical edge in the current frames and the horizontal edge and vertical edge in the previous frames.

4-bits memory cell:

3	2	1	0
a bit for horizontal current edge	a bit for vertical current edge	a bit for horizontal previous edge	a bit for vertical previous edge
0	1	0	1
1	1	1	0
1	0	0	1
1	0	0	1
0	0	1	0
0	1	0	1
1	1	0	0

16 x 16

Figure 5.11: 4-bits memory arrangement

The architecture of designing the pixel, edge detector and memory cell parallelly together benefits for edge detection, storage and transition.

5.4.1 How to Design a 4-bits Memory on Focal Plane

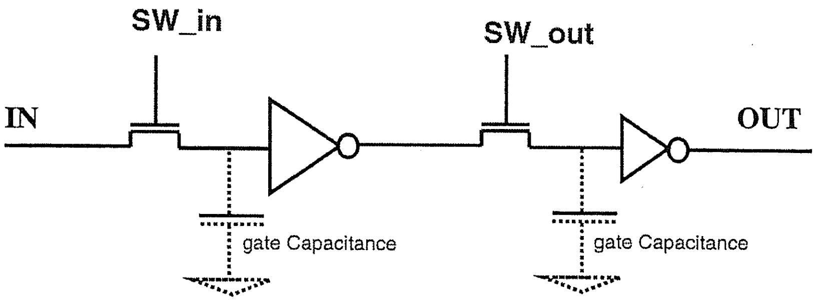


Figure 5.12: The principle circuit of the transmission gate memory cell.

By making use of the gate capacitance, we implemented a 4-bits memory cell together with the pixel and edge detection. Any 1 bit information can be easily written in or read out by ON and OFF operation of the corresponding CMOS switches. Fig. 5.12. shows the principle circuit of the transmission gate memory cell.

5.4.2 A Tree Structure of the Memory Cell for Block Maching

The memory cell circuit and its input/output control have been shown in Fig. 4. 9 in the previous section.

The stored binary edge information is devided into two main groups preparing for block matching, as shown in the memory tree in Fig. 5.13, i.e. those for current frame's edges and those for previous frame's edges.

The signals of Cur_1st_h_j and Cur_2nd_h_j (j=1,2)
are used for controlling the output of current frame edges, and the signals of
Pre_1st_h_i, Pre_2nd_h_i,Pre_3rd_h_i, and Pre_4th_h_i (i=1,2,3,4)
for previous frame edges. All of the output of the memory cell controlled by the above signals will be connected to the block matching architecture for the local parrellel processing.

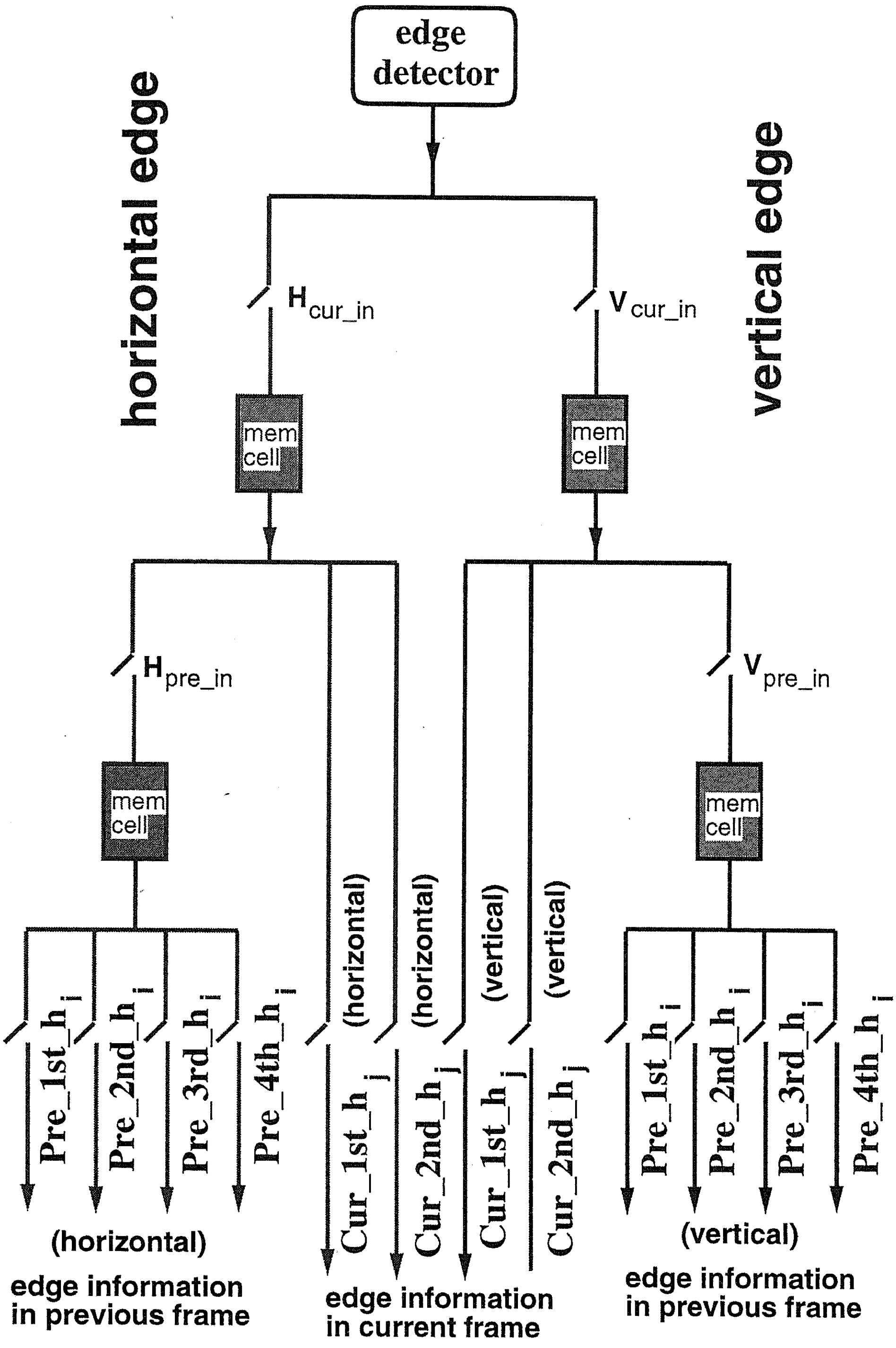


Figure 5.13: 4-bits memory cell and its tree structure for read-out. i=1,2,3,4. j=1,2.

5.5 Local Parallel and Globla Collumn Parallel(LPGCP) processing architecture on focal plane

Since the resources on a tiny focal plane is so limited that it is very difficult to do the same kind of operation of blocking matching as the DSP does.

A high speed block matching on the focal plane is to try to effectively execute an operation in a way like Fig. 5.14 shown.

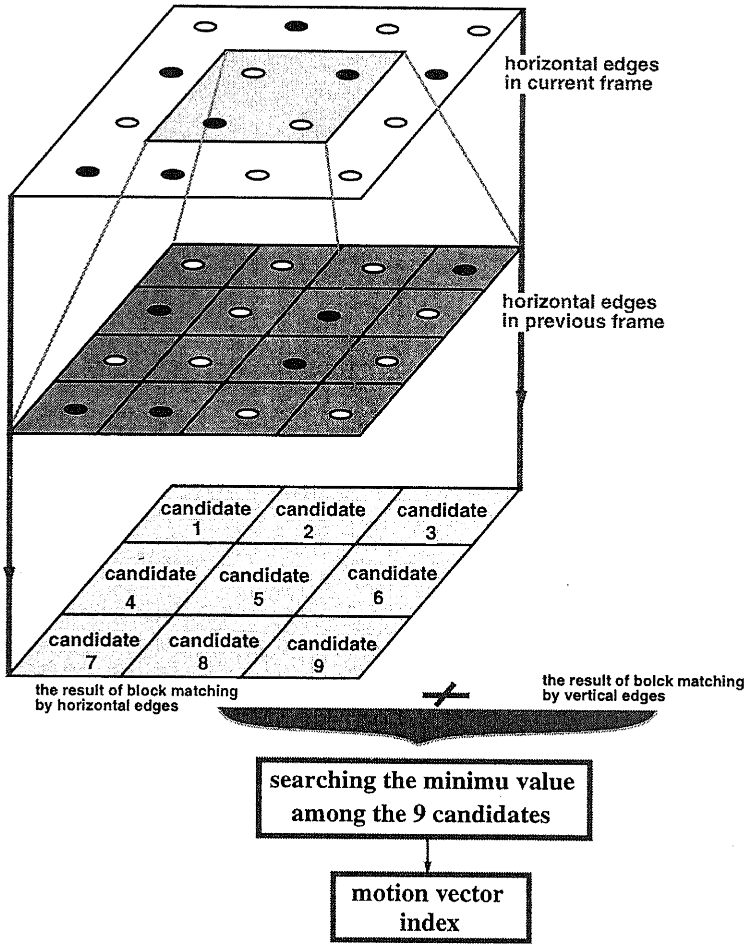

horizontal edges in current frame
horizontal edges in previous frame
candidate 1
candidate 2
candidate 3
candidate 4
candidate 5
candidate 6
candidate 7
candidate 8
candidate 9
the result of block matching by horizontal edges
the result of bolck matching by vertical edges
searching the minimu value among the 9 candidates
motion vector index

Figure 5.14: A Local Parallel and Globla Collumn Parallel(LPGCP) processing architecture is used for a high speed block matching and motion vector detection.

In order to perform the high speed block matching in the pyramid architecture and fully make use of the parallel nature of image signal, we adopted the highly compacted parallel structure of edge detection and short-time memory in Fig. 5.1, which is considered as an optimal pre-arrangement for the whole focal plance processing in Fig. 5.15. Otherwise, the layout lines will be increased immensely.

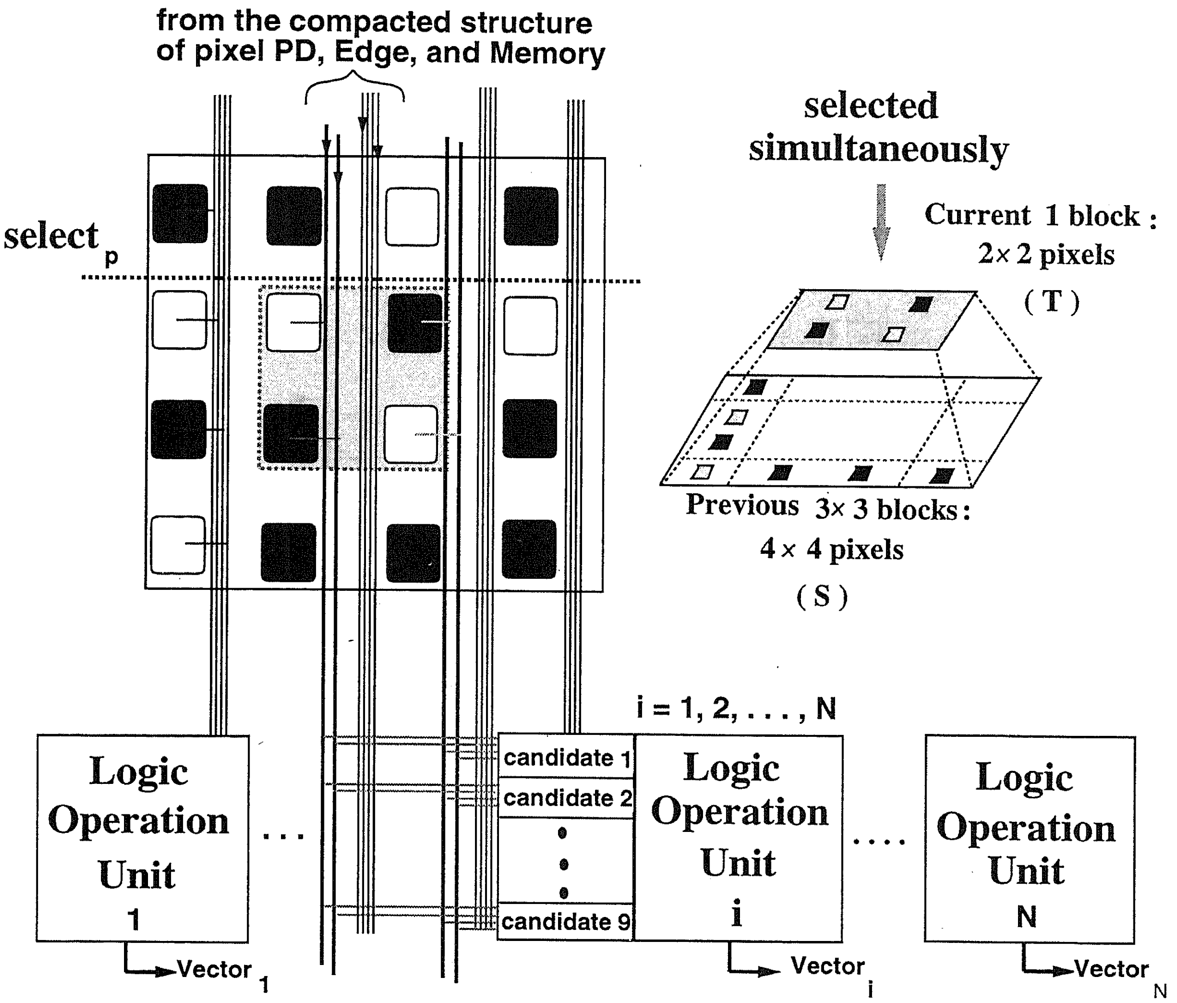


Figure 5.15: High speed block matching with local pixel parallel and global collumn parallel processing (LPGCP) architecture. Only $1 \times (N-3)$ block matching processing units are needed on the whole sensor plane with $N \times N$ pixels.

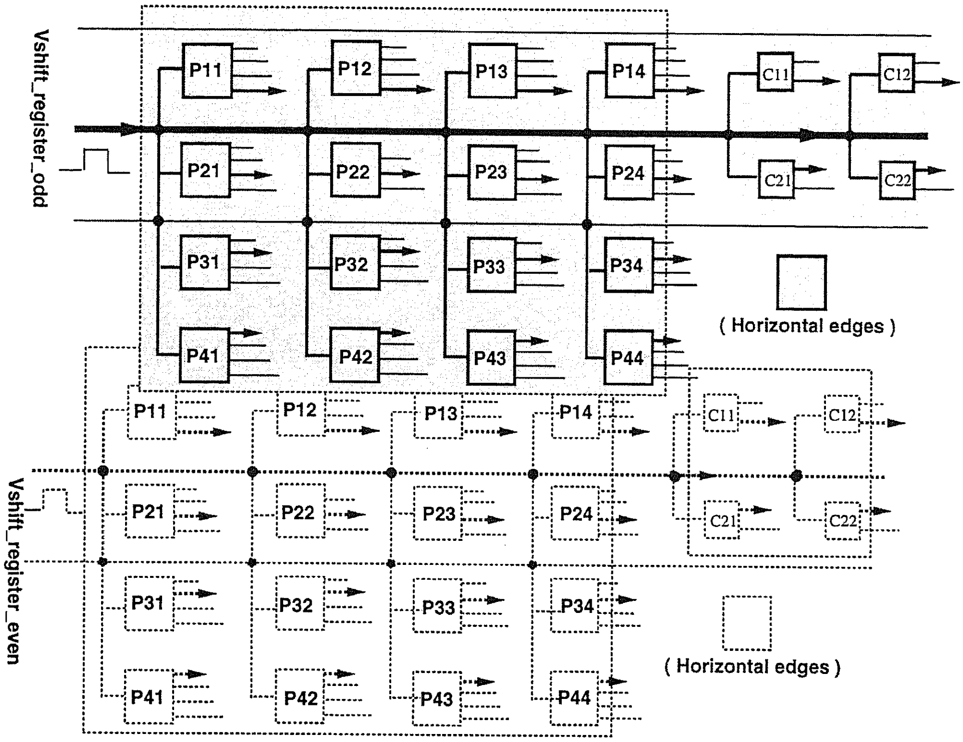


Figure 5.16: The pipeline process of reading of the binary edges from the array of 4-bits memory tree is driven by the Vertical Shift Register but with its odd-phase output and even-phase output, respectively.

Fig. 5.1 and Fig. 5.16 together will construct the Local pixel Parallel and Global Collumn Parallel processing(LPGCP) architecture for the fast 2D motion vector estimation on the CMOS sensor focal plane.

As it is shown in Figure 5.15, the block matching around the selected blocks, denoted by $select_p$ line, is taking place simultaneously with all the pixels within the searched local area $(-m, -n \leq i, j \leq m, n)$. And globally, only $1 \times (N-3)$ block matching processing units are needed on the whole sensor plane with $N \times N$ pixels. Here in our prototype vision chip, we chose m=n=1, and N=16.

In performing the high speed block matching and motion vector detection, the reading of the binary edges from the 4-bits memory array is performed by the Vertical Shift Register but with its odd-phase output and even-phase output, respectively. The pipeline process of reading the binary edges from the memory tree is shwon in Fig. 5.16.

5.6 Specific Accumulator(ACC)

Here we will design a specific Accumulator(ACC), which will be used for performing a small size block matching and for searching the best motion vectors.

5.6.1 ACC for Block Matching on the Focal Plane

As shown in Fig. 5.18, the binary edges stored in the current frame memory and previous frame memory will be used for performing the block matching. The result of the block matching is queued by 8 samplings and then converted into a 8-bits register in the ACC (accumulator) in a form of all the 1's are on the left(MSB) and all the 0's are on the right side(LSB). This special arrangement for the content of the 8 bits register greatly eases the vector's priority decision.

5.6.2 Searching for the Minimum and the Best Vector

Fig. 5.18 best shows the mechanism of searching for the minimum and the best vector among 9 candidates by using our specific accumulator.

First, all the contents of the 8-bits register which holds the candidate are performed an AND operation with each other bit by bit. This operation captures the minimum value among the 9 candidates. Then by using this minimum value, to perform a XNOR operation with all the 9 candidates. Those registers which keep the same minimum value output "1", while those registers which keep larger values output "0". The most smallest candidate is chosen as the vector.

But when several same minimum values come up at the same time, an arbitration logic is needed to select the one as the final vector, which has the shortest distance d_i to the center of the searched area according to the priority dicision rule shown in Fig. 4. 10. The specific accumulator companied with a priority dicision/arbitration logic ensures a unique vector can be detected.

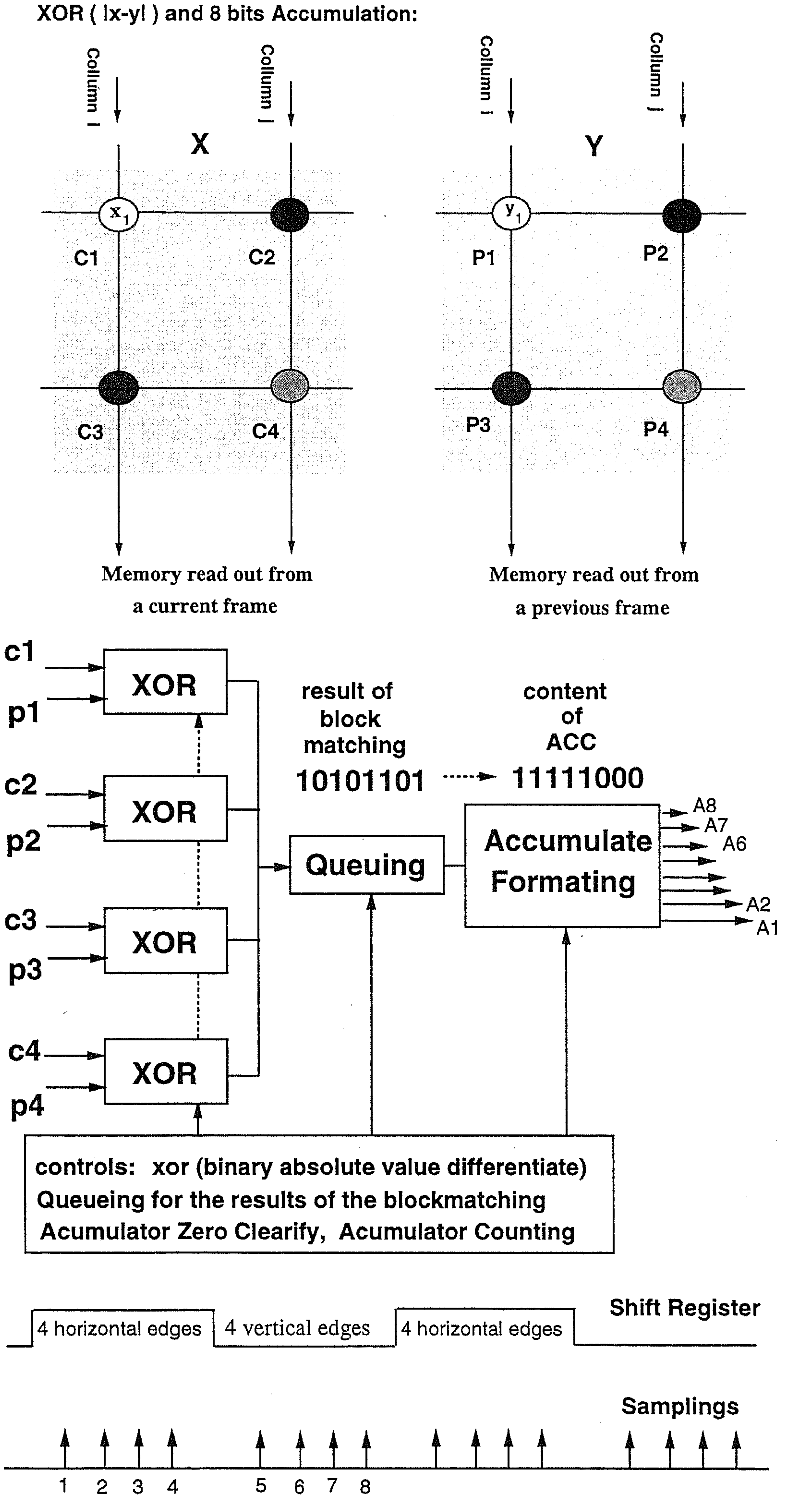


Figure 5.17: xor(x,y) for 8 bits ACC and the blockmatching accumulation control diagram.

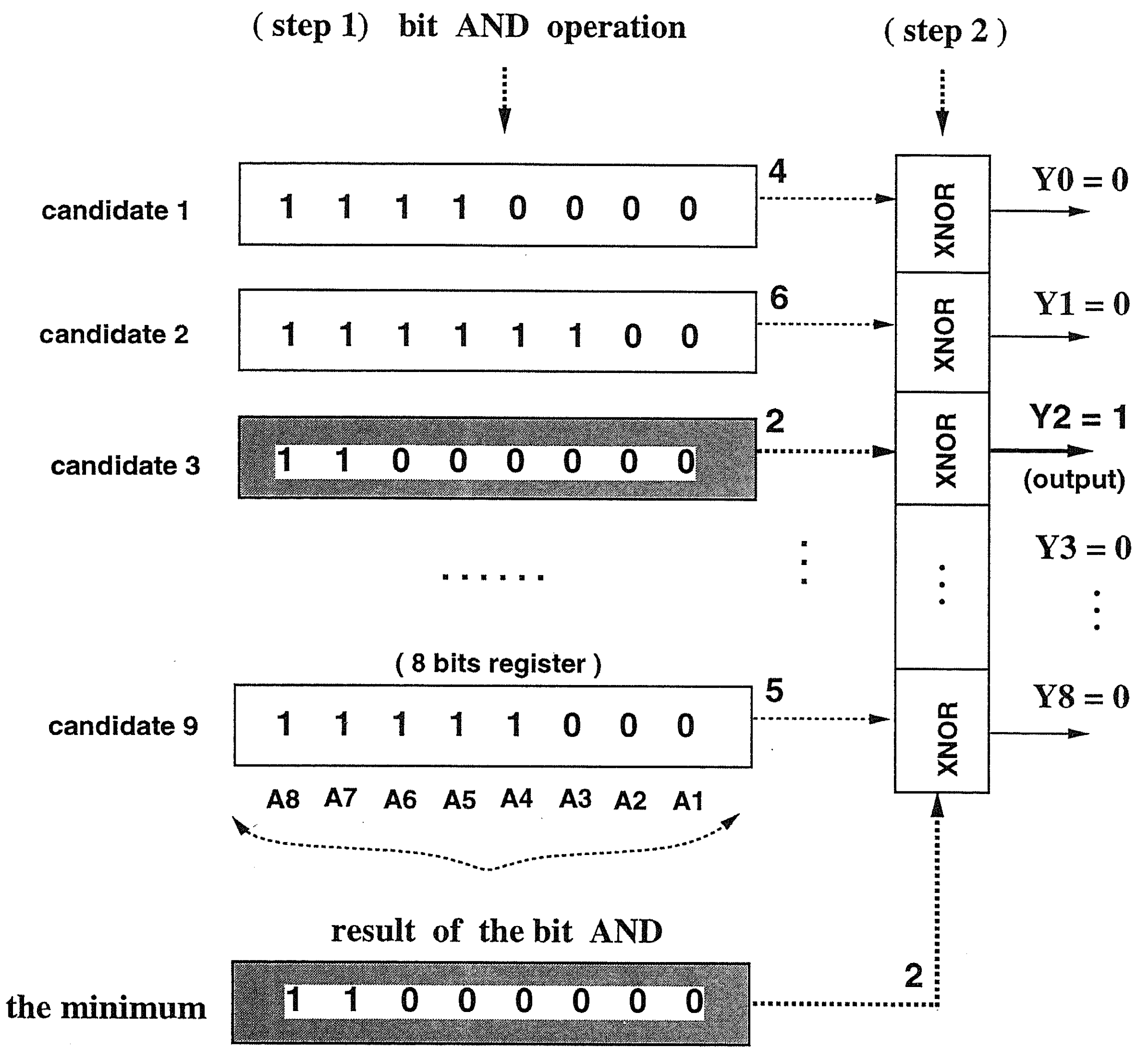


Figure 5.18: Searching for the minimum and the best vector among 9 candidates according to the priority decision rule in Fig. 5.19. The result of the block matching is queued by 8 samplings and then converted into the ACC in a form of all the 1's are on the left(MSB) and all the 0's are on the right(LSB) in a 8-bits register.

5.7 Circuit Simulation for the ACC(Accumulator)

The results of the block matching should be accumulated in an accumulator, i.e. ACC. Since there is no CPU on the solide silicon surface, a special ACC is neccessary for the focal plane processing. Fig. 5.19 is the specific ACC designed in our vision chip. Fig. 5.20, Fig. 5.21, Fig. 5.22, and Fig. 5.23 are HSPICE simulations for the specific ACC.

The inputs in Fig. 5.19 correspond to the inputs in Fig. 5.17. They represent the relative edge information in the current frame and previous frame, respectively. These group of operations are used twice for the horizontal block matching and vertical block matching under the control of the Even-phase output and Odd-phase output of the horizontal shift register.
(c1, p1) ⇒ (Cur_1st_h1, Pre_1st_h1)
(c2, p2) ⇒ (Cur_1st_h2, Pre_1st_h2)
(c3, p3) ⇒ (Cur_2nd_h1, Pre_2nd_h1)
(c4, p4) ⇒ (Cur_2nd_h2, Pre_2nd_h2)

The simulation fulfils the following operation as supposed to be on the focal plane.

$$ACC_k = (\sum_{i=1}^{4} c_i \oplus p_i)_{even-phase} + (\sum_{j=5}^{8} c_j \oplus p_j)_{odd-phase}$$

$$\mathrm{i} = 1, 2, 3, 4; \qquad \mathrm{j} = 5, 6, 7, 8$$

There are 9 accumulators used for block matching and motion vector detection, therefore k = 1, 2, 3, 4, 5, 6, 7, 8, 9;
$\oplus$ is the XOR operator for binary information.

And other signals are controls for this specific ACC.

The simulation is taking palace under CADENCE software environment after running icfb on a SUN Workstation. Since this software is linked with MetaWaves HSPICE simulator, the initiate command (source /MetaSoft/96/bin) should be included in the .cshrc file.

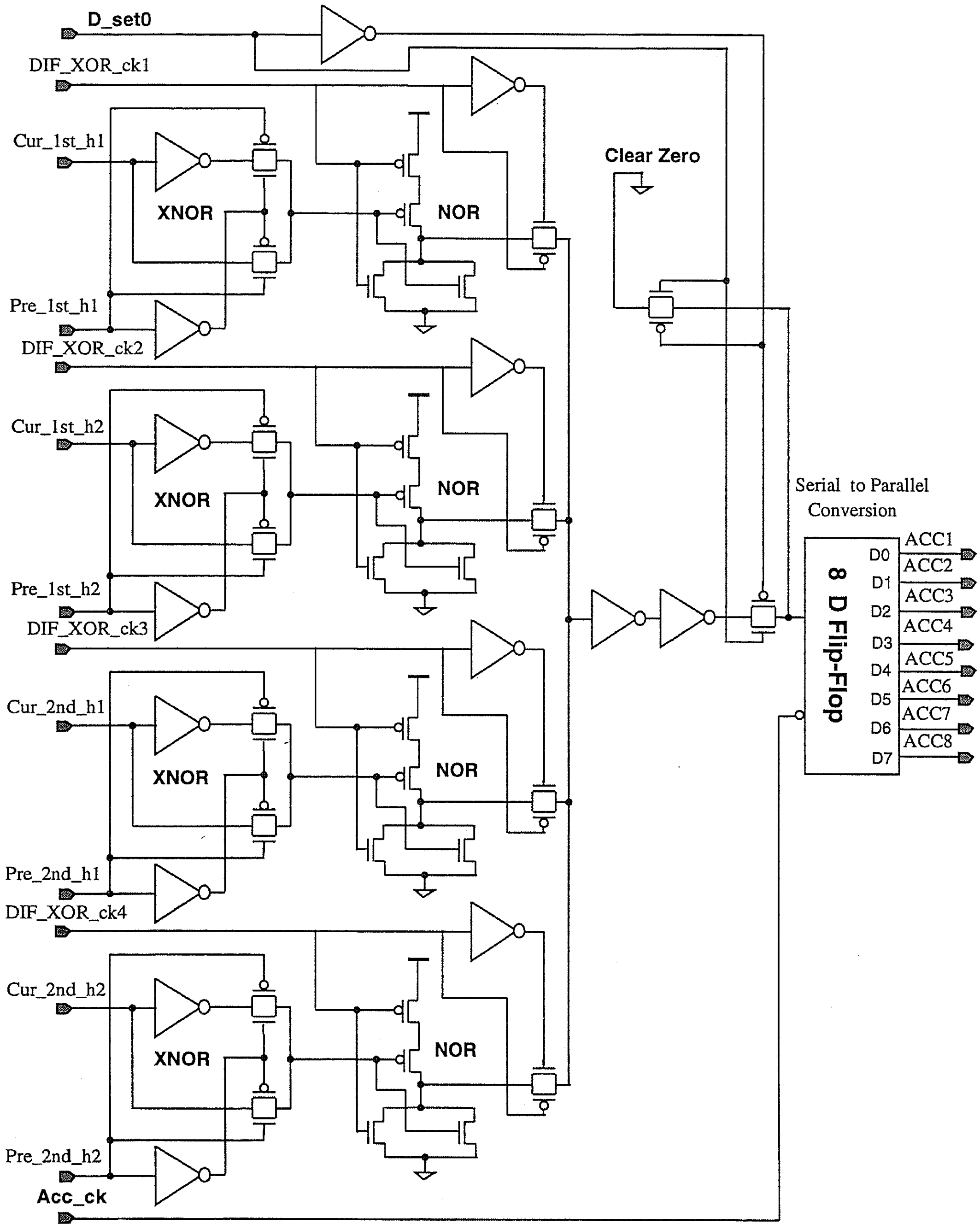


Figure 5.19: Accumulator circuit: ACC_i, i=1,2,3,4,5,6,7,8,9. The output of ACC_i is stored in a 8-bits register: {A8, A7, A6, A5, A4, A3, A2, A1 }.

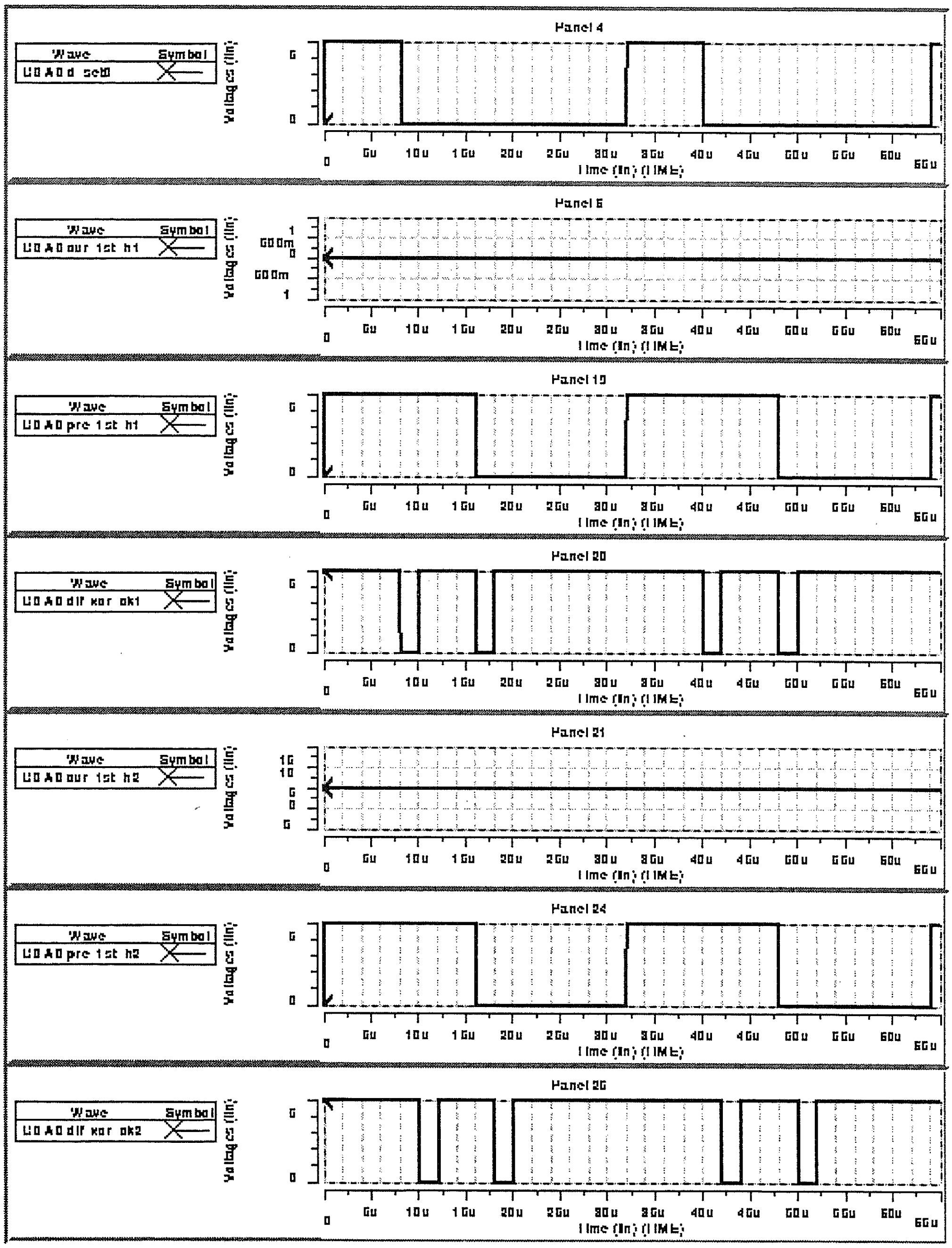


Figure 5.20: Accumulator simulation 1: From top to down are the signals of D-set0, Cur-1st-h1, Pre-1st-h1, DIF-XOR-ck1, Cur-1st-h2, Pre-1st-h2, DIF-XOR-ck2.

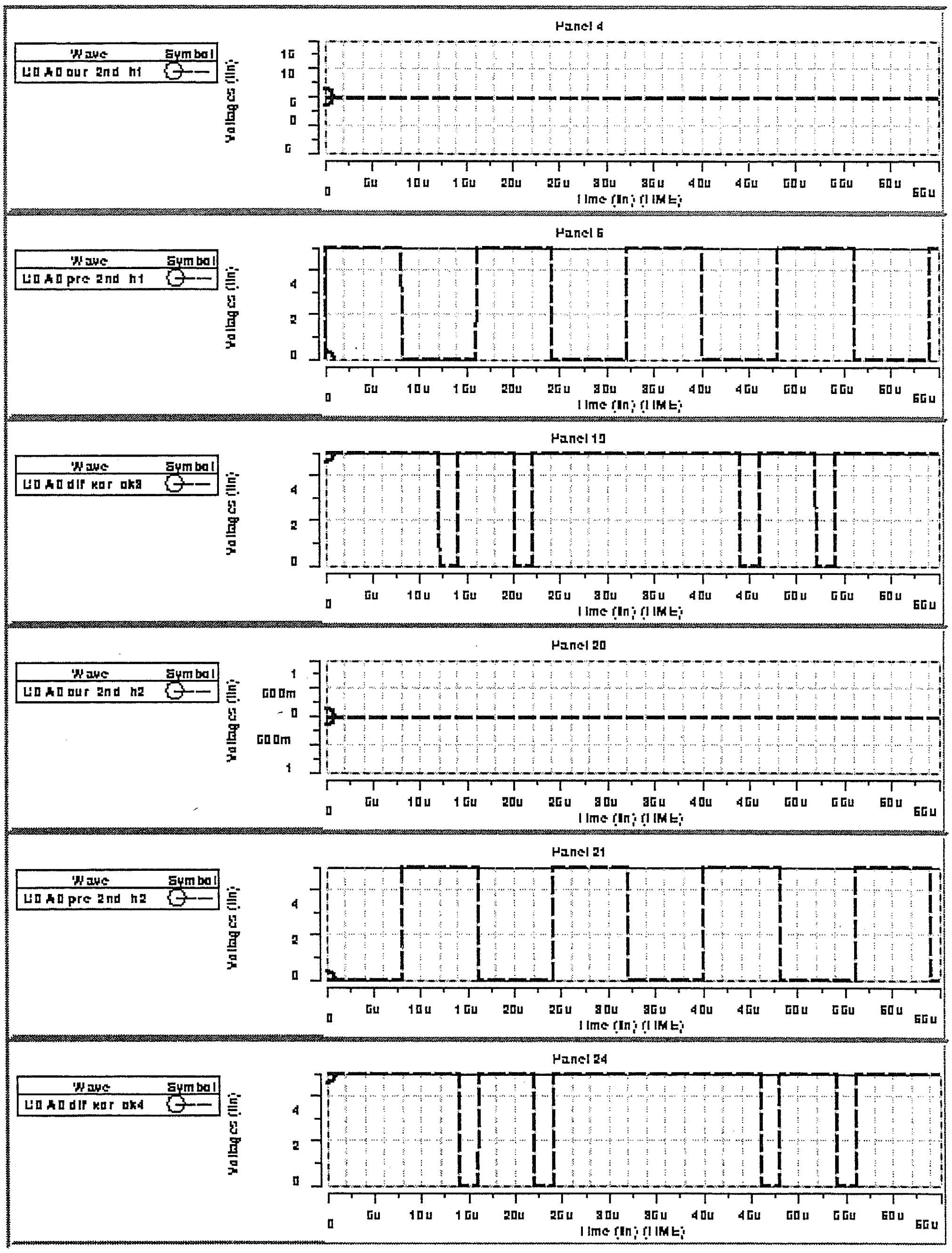


Figure 5.21: Accumulator simulation 2: From top to down are the signals of Cur-2nd-h1, Pre-2nd-h1, DIF-XOR-ck3, Cur-2nd-h2, Pre-2nd-h2, DIF-XOR-ck4.

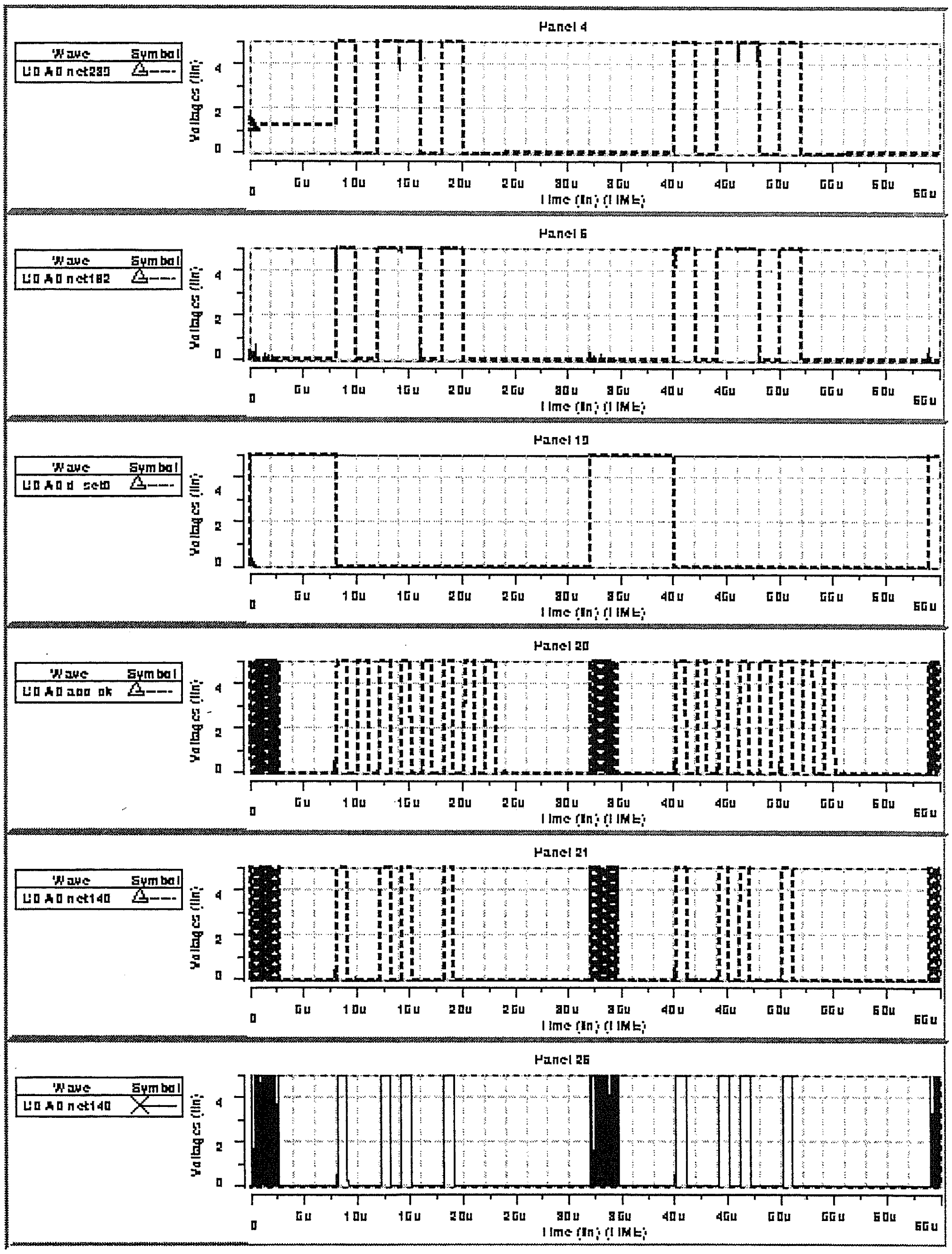


Figure 5.22: Accumulator simulation 3: The 3rd pannel is D-set0, the 4th pannel is ACC-ck.

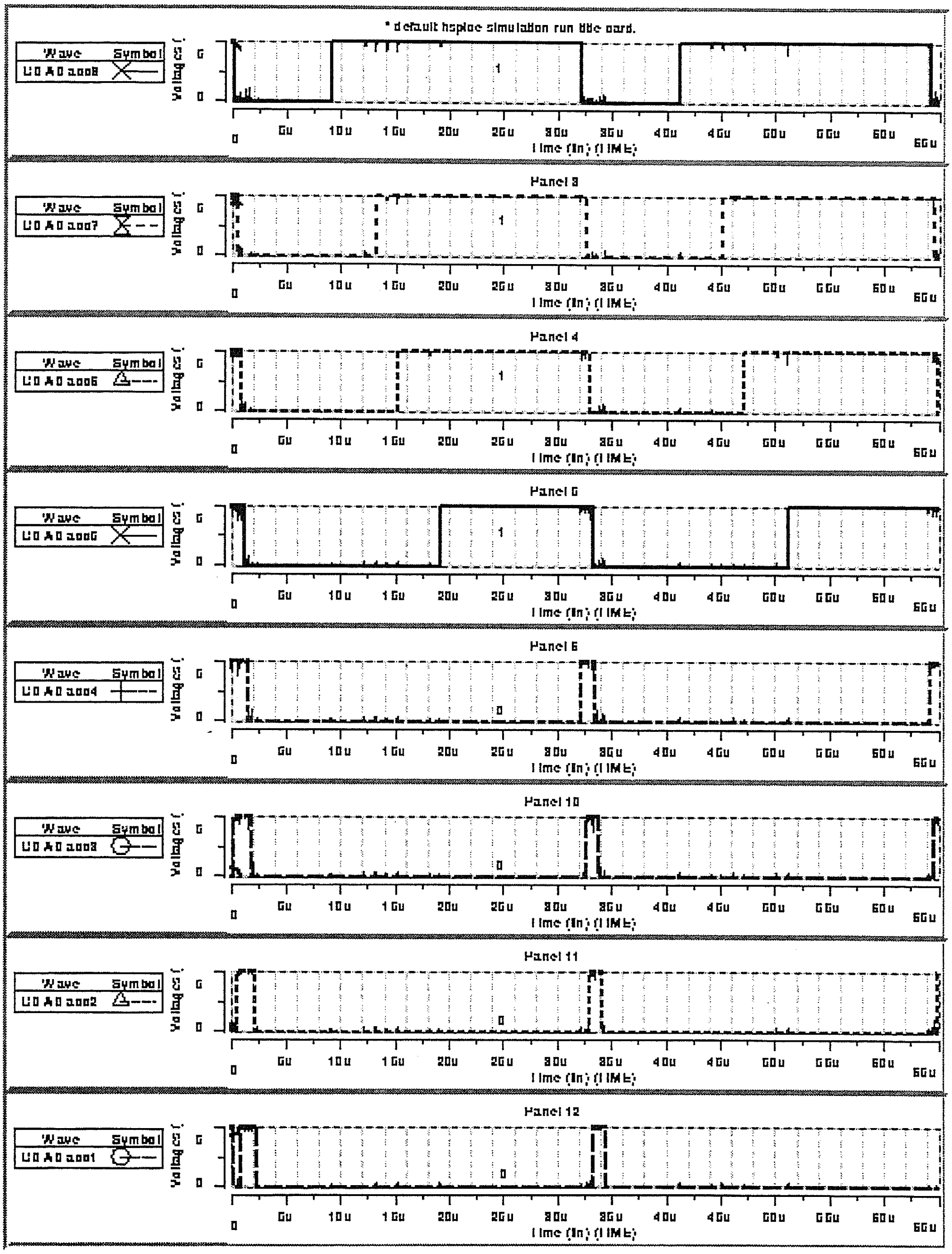


Figure 5.23: Accumulator simulation 4: From the 1st pannel to the 8th are the output of an ACC circuit: A8, A7, A6, A5, A4, A3, A2, A1. The ACC content of { 1 1 1 1 0 0 0 0 } can be obtained at the sampling time of 28 μsec.

5.8 Circuit Simualation for the Specific Processor of Block Matching and Motion Vector Detection

The logic circuit architecture for block matching and vector detection is shwon in Fig. 5.24. It consists of a specific processor when integrated on the focal plane of our vision chip for a very fast motion vector estimation.

A simulation procedure by HSPICE(Metawave) for bock matching and vector detection are shown in Fig. 5.25, Fig. 5.26, Fig. 5.27, Fig. 5.28, and Fig. 5.29. As a result, among the output lines of { Y0, Y1, Y2, Y3, Y4, Y5, Y6, Y7, Y8 } Y4 has a "1" output at the sampling time (activated by Syn-Yi-reg), so that a motion vector at Y4 is detected in this example.

Further more, a series of simulations by using random data for a 2 × 2 pixels in the current frame image and 4 × 4 pixels in the previous frame images are correctly detected and summarized in Fig. 5.30.

This simulation summary clearly shows that there are 9 vector candidates after the blockmatching operations, the most smallest one is chosen as the best vector. But sometimes when multiple minimum values appear, a priority arbitration logic based on Fig. 4.10 is applied to select the one as the final vector which has the short distance from the center of the Vector Order dicision diagram and with the smallest index number.

The final vectors detected in Fig. 5.30 are No. 0, 4, 0, and 2, i.e. Y4, Y7, Y4, and Y3 in series.

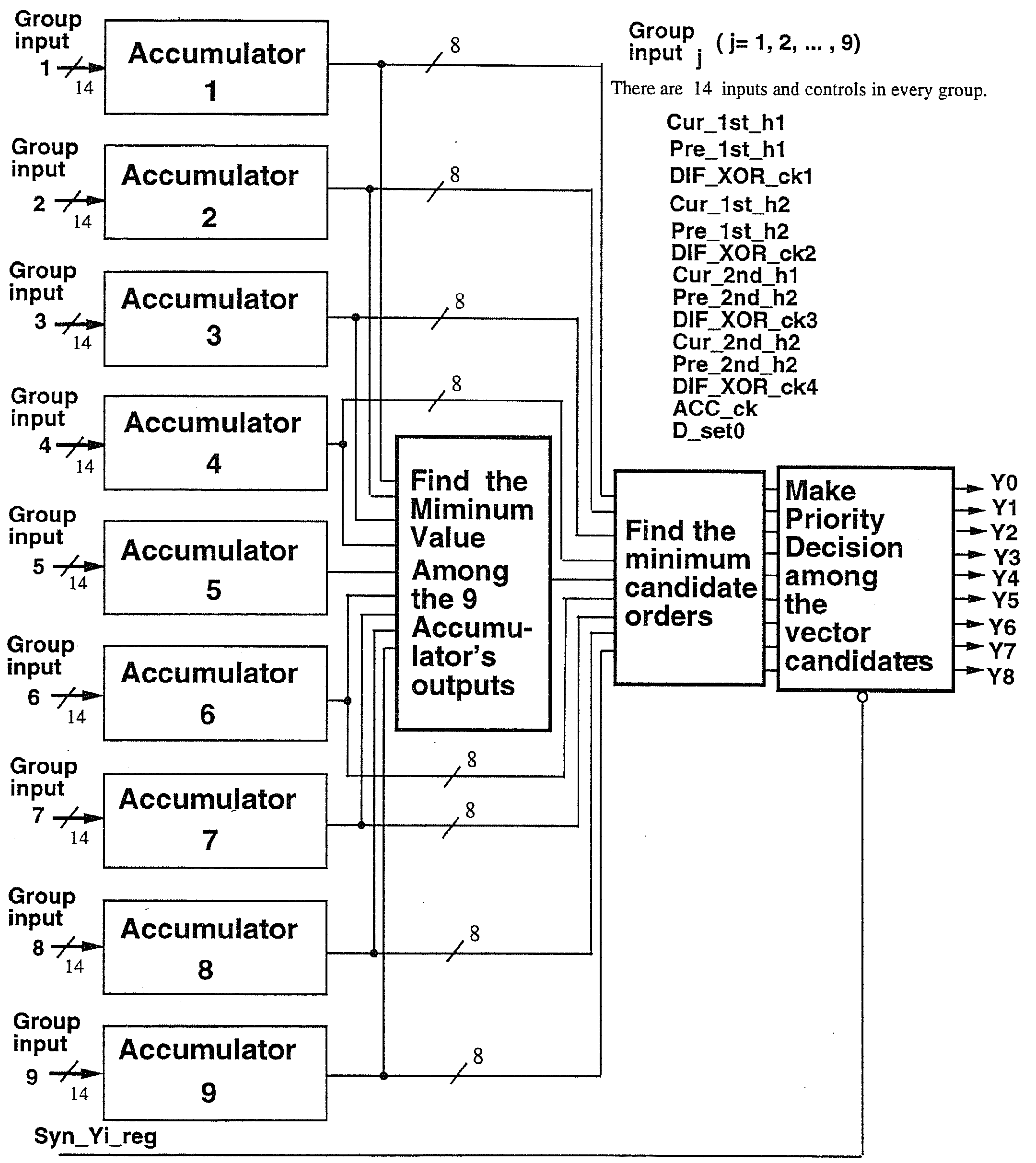


Figure 5.24: A specific processor integrated on focal plane of the vision chip for very fast block matching and motion vector detection. There are 9 groups of inpus for the accumulators. Among every group of accumulator, there are 4 inputs for current edges, 4 inputs for previous edges, and 7 control signals. The operation takes place according to the procedures shown in Fig. 5.17, Fig. 5.18 and Fig. 4.10. The combination of 9 accumulators forms the result of the block matching. After the minimum values are found, priority decided vectors are output by 9-bits lines.

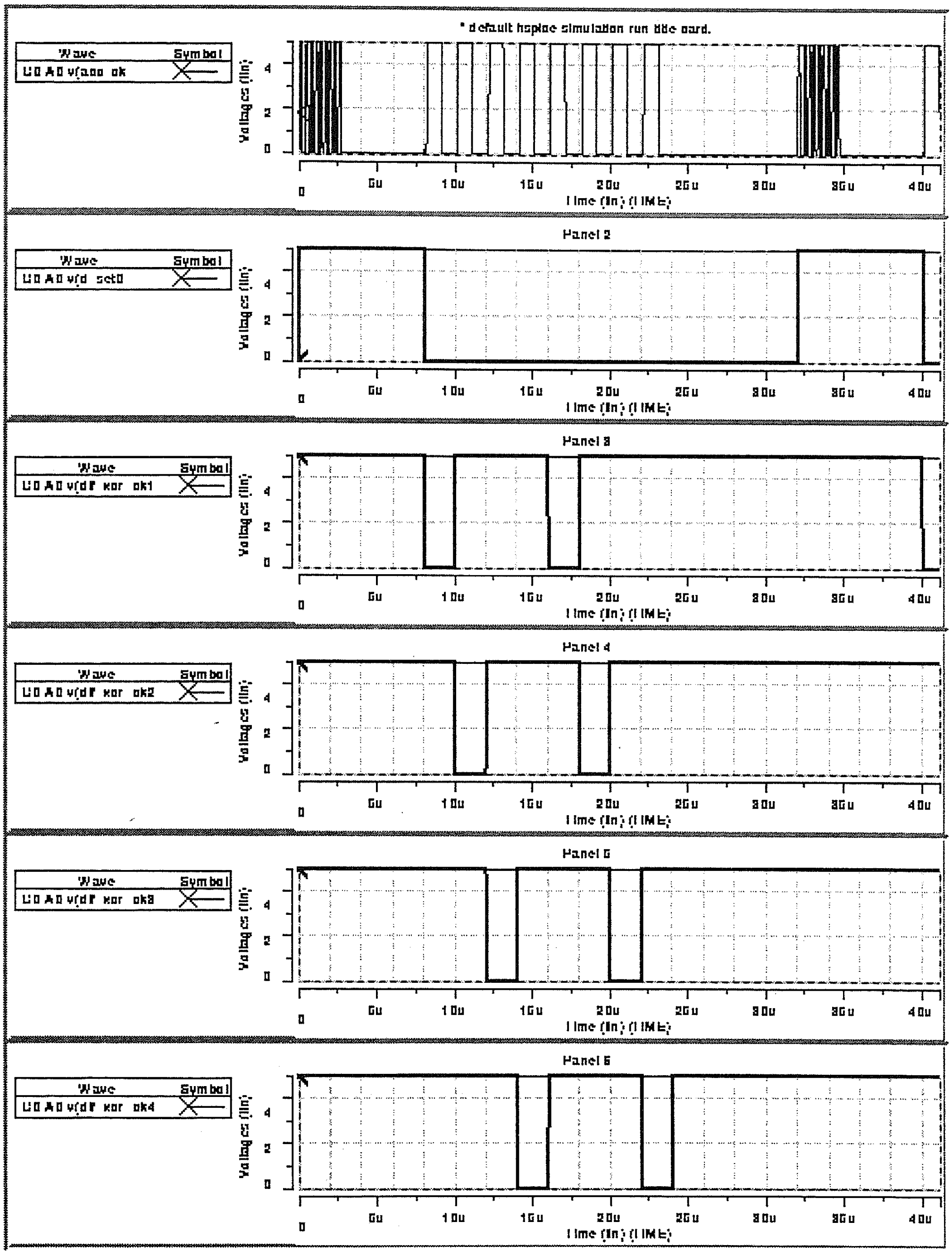


Figure 5.25: Block matching and vector detection simulation 1: the 1st Pannel is ACC-ck, the 2nd Pannel is D-set0, the 3rd Pannel is DIF-XOR-ck1, the 4th Pannel is DIF-XOR-ck2, the 5th Pannel is DIF-XOR-ck3, the 6th Pannel is DIF-XOR-ck4.

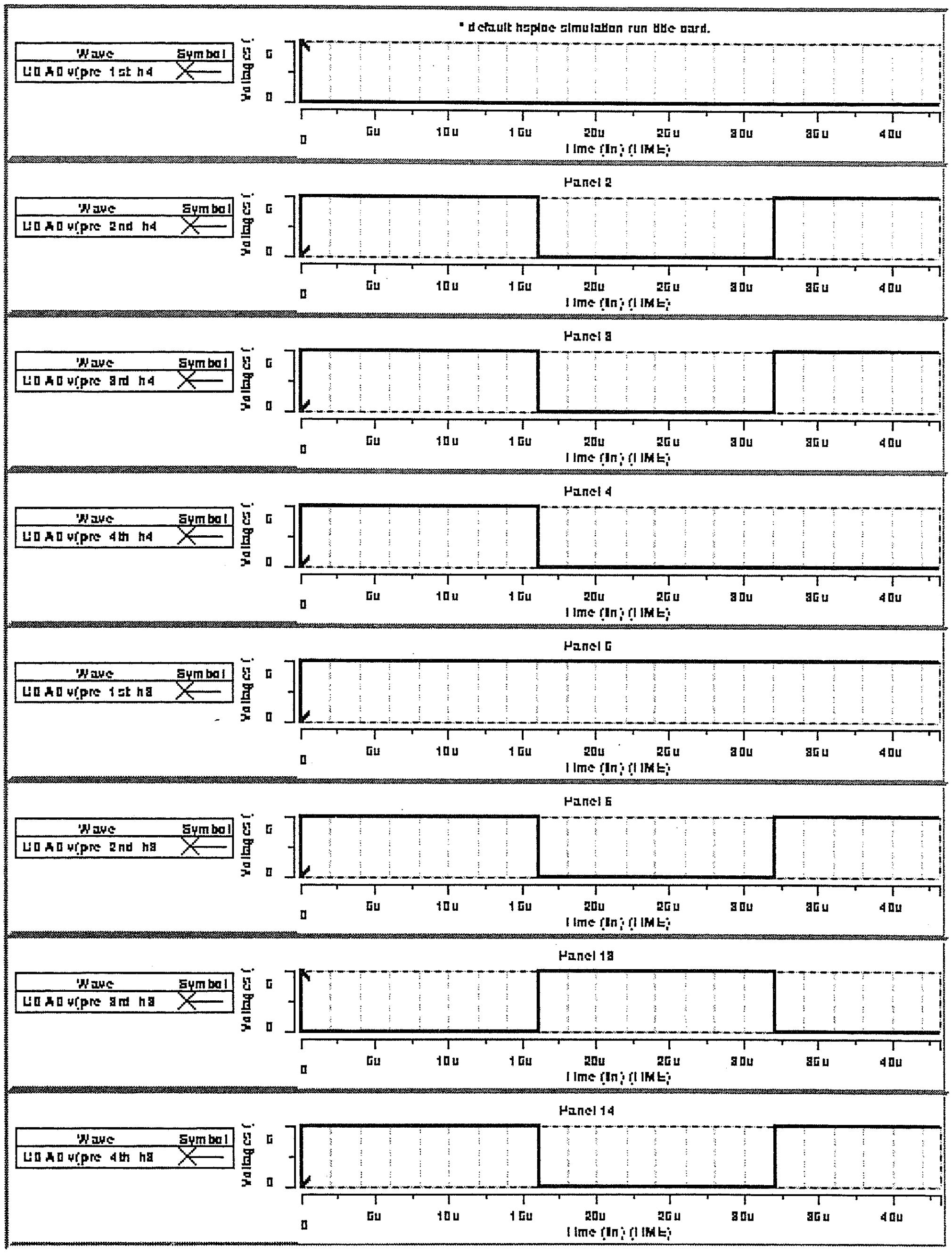


Figure 5.26: Block matching and vector detection simulation 2: the 1st to 4th Pannels are Pre-1st-h4, Pre-2nd-h4, Pre-3rd-h4, Pre-4th-h4; the 5th to 8th Pannels are Pre-1st-h3, Pre-2nd-h3, Pre-3rd-h3, Pre-4th-h3.

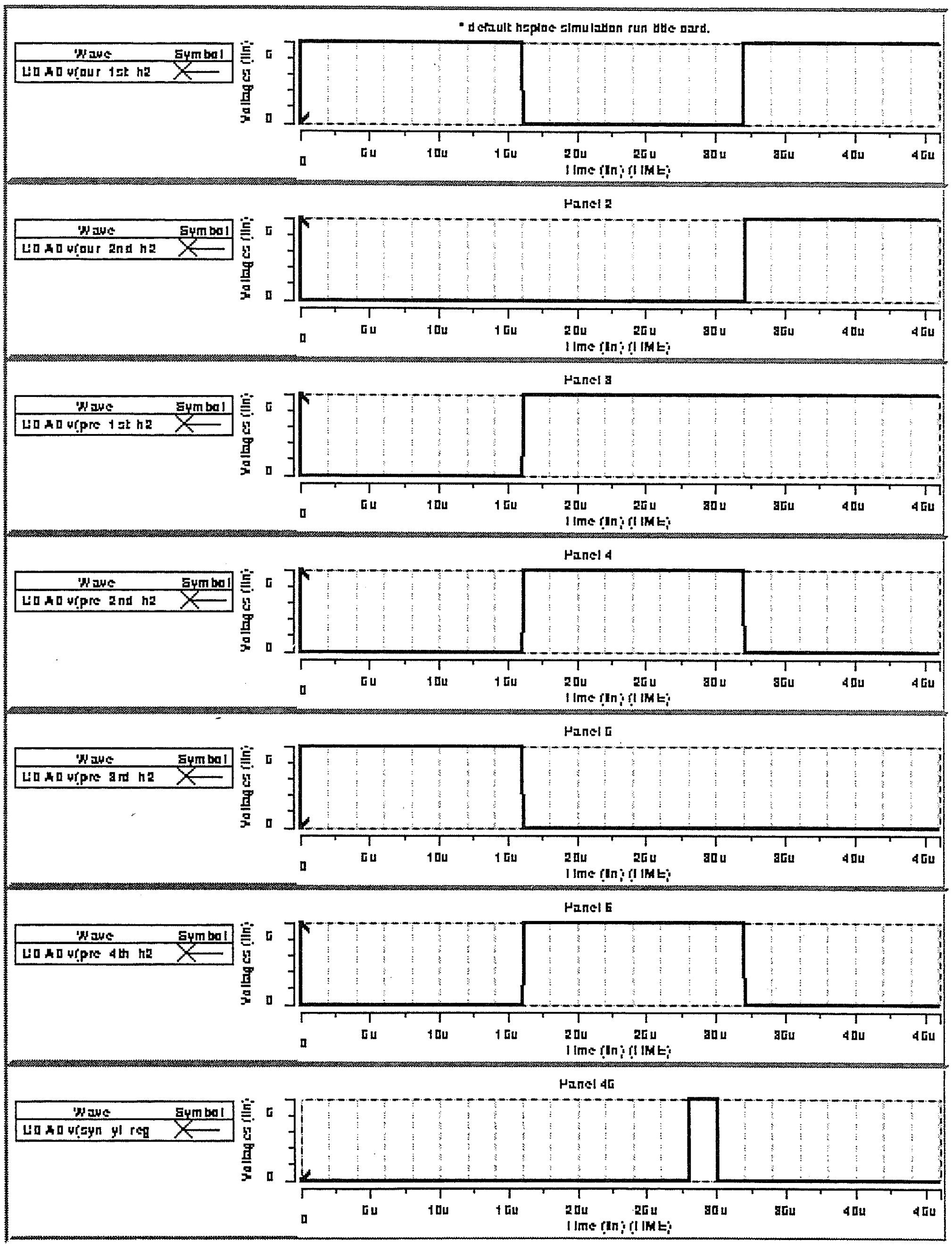


Figure 5.27: Block matching and vector detection simulation 3: the 1st to 2nd Pannels are Cur-1st-h2, Cur-2nd-h2; the 3rd to 6th Pannels are Pre-1st-h2, Pre-2nd-h2, Pre-3rd-h2, Pre-4th-h2; the 8th Pannel is Syn-Yi-reg.

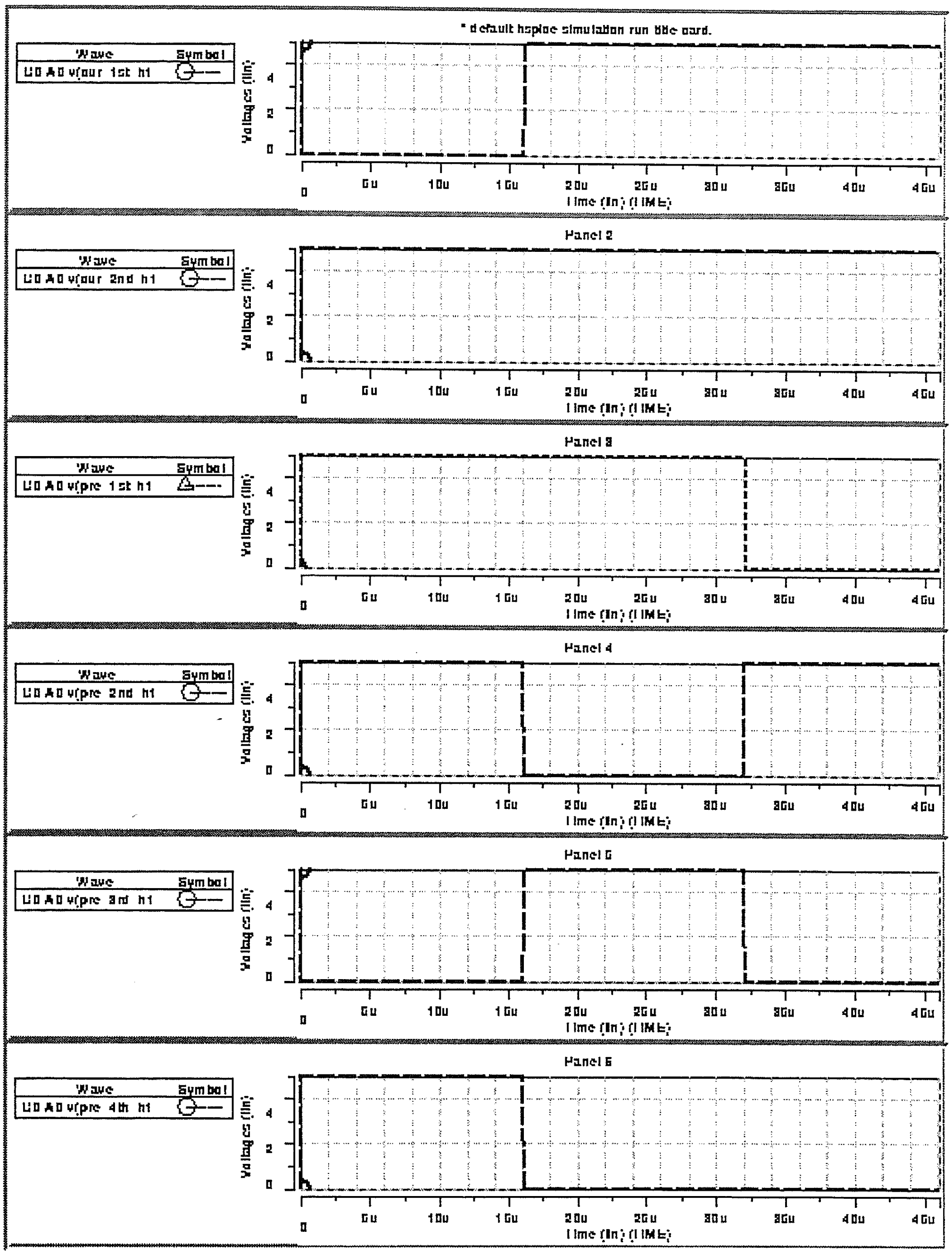


Figure 5.28: Block matching and vector detection simulation 4: the 1st to 2nd Pannels are Cur-1st-h1, Cur-2nd-h1, the 3rd to 6th Pannels are Pre-1st-h1, Pre-2nd-h1, Pre-3rd-h1, Pre-4th-h1.

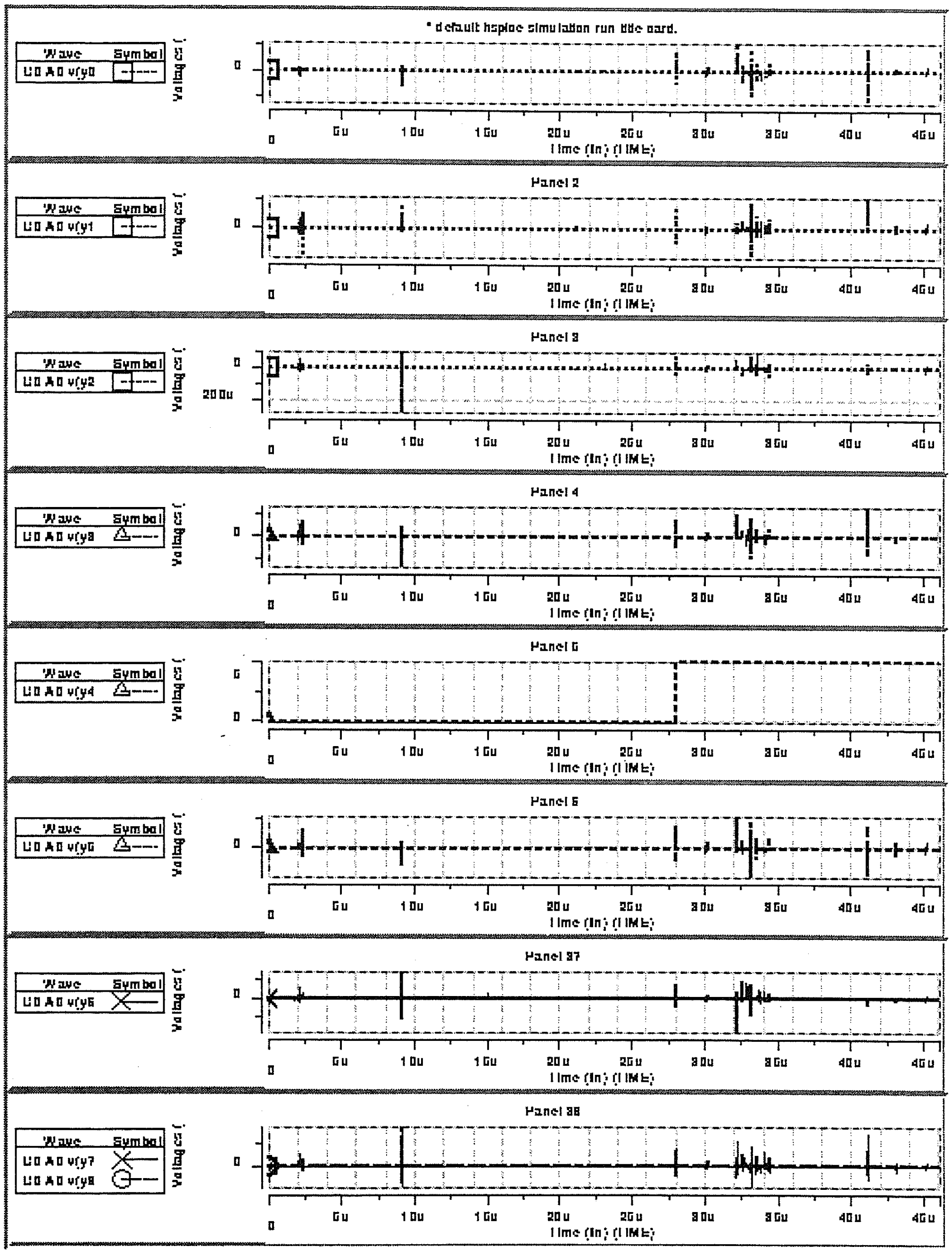
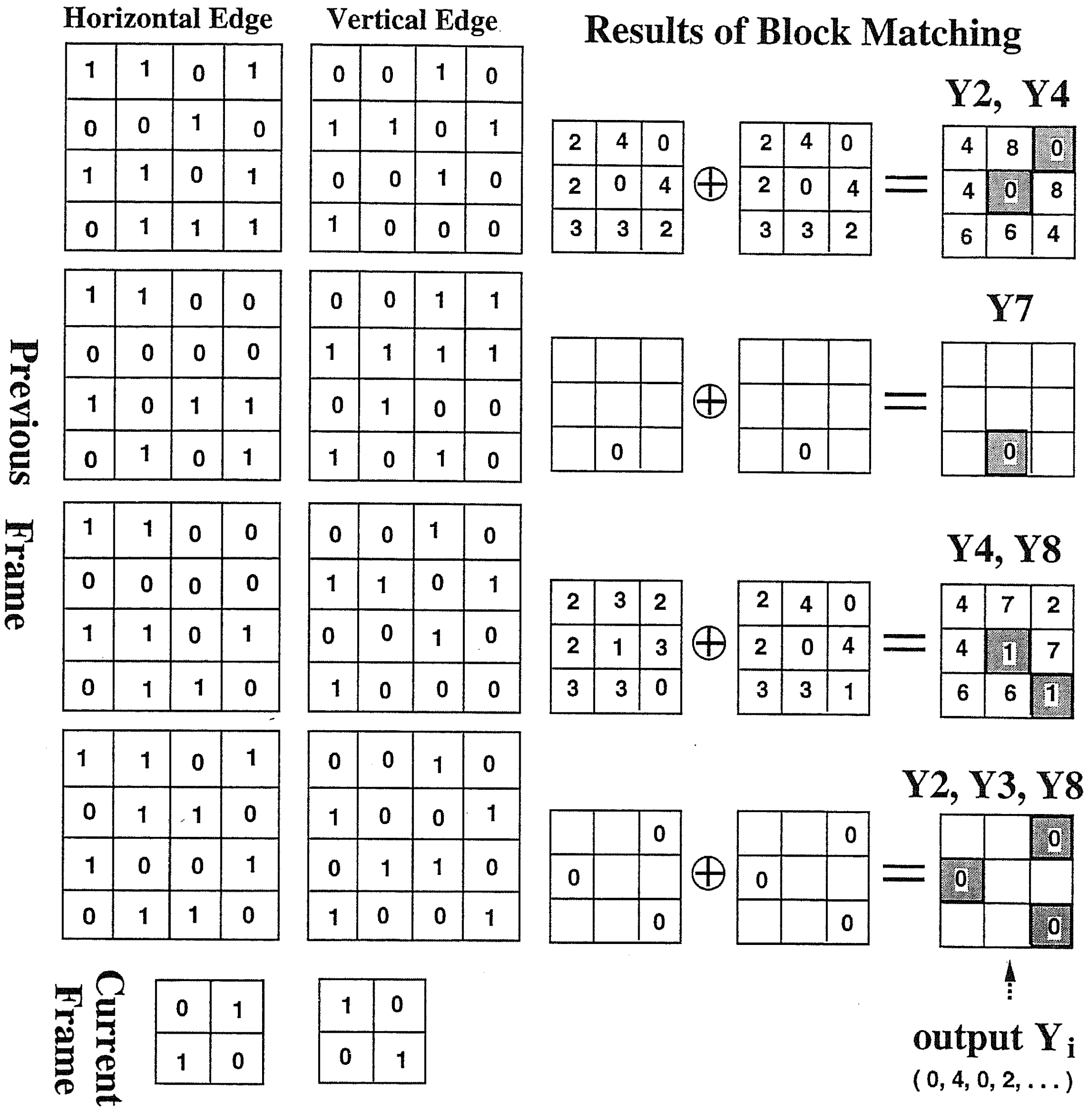


Figure 5.29: Block matching and vector detection simulation 5: the 1st to 7th Pannels are Y0, Y1, Y2, Y3, Y4, Y5, Y6; the 8th Pannel is for Y7 and Y8. We can notice that Y4 has a "1" output at the sampling time of Syn-Yi-reg, so that we detected a motion vector at Y4 in this example.



Previous Frame, Horizontal Edge

1	1	0	1
0	0	1	0
1	1	0	1
0	1	1	1

1	1	0	0
0	0	0	0
1	0	1	1
0	1	0	1

1	1	0	0
0	0	0	0
1	1	0	1
0	1	1	0

1	1	0	1
0	1	1	0
1	0	0	1
0	1	1	0

Previous Frame, Vertical Edge

0	0	1	0
1	1	0	1
0	0	1	0
1	0	0	0

0	0	1	1
1	1	1	1
0	1	0	0
1	0	1	0

0	0	1	0
1	1	0	1
0	0	1	0
1	0	0	0

0	0	1	0
1	0	0	1
0	1	1	0
1	0	0	1

Current Frame

0	1
1	0

1	0
0	1

Results of Block Matching

Y2, Y4

2	4	0
2	0	4
3	3	2

⊕

2	4	0
2	0	4
3	3	2

=

4	8	0
4	0	8
6	6	4

Y7

	0	

⊕

	0	

=

	0	

Y4, Y8

2	3	2
2	1	3
3	3	0

⊕

2	4	0
2	0	4
3	3	1

=

4	7	2
4	1	7
6	6	1

Y2, Y3, Y8

		0
0		
		0

⊕

		0
0		
		0

=

		0
0		
		0

Figure 5.30: Circuit level simulation summary. The results of the motion vectors detected by the block matching of horizontal edges and vertical edges(both are from current frame and previous frame) are the same as the theoritical analysis.

5.9 Shifting Windows of the LPGCP Processors on the Focal Plane

In our design, only 4 LPGCP ALUs or Processors are adopted for the motion vector detection on the whole focal sensor plane. Therefor a shifting window having 3 controllable positions on the focal plane is determined by a 3-bits control logic outside of the chip. The principle of the shifting window technigue is shown in Fig. 5.31, which has the advantages of reducing the layout size and programmablly locating an interested area of motion vector detections. The area where motion vector detection is taking place is controled by the 4LPGCP-ALU shifting window according to the following logic:

$$Window(_{position\ j}) = P_{i+4(j-1)}$$

$i = 1, 2, 3, 4, 5, 6, 7$ is the number of collums of the pixel array;
$j = 1, 2, 3$ is the order of position of the 4LPGCP-ALU Window.

Within the window, the motion vector detection is in a full speed, while on the whole surface of the focal plane, the speed of the motion vector detection is 1/3 of the full speed.

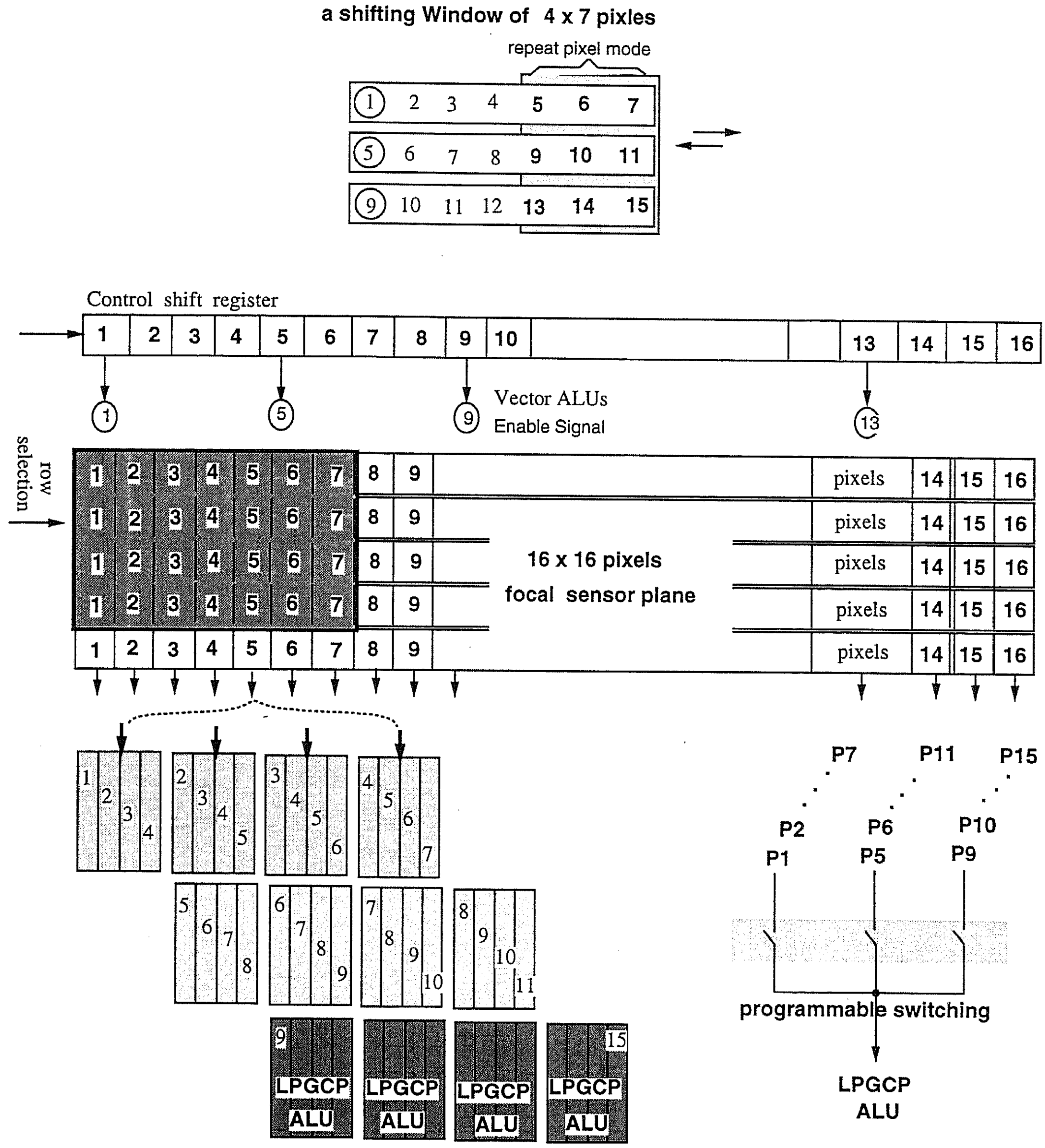


Figure 5.31: Shifting window of LPGCP for motion vector estimation on focal plane. Here $P_{[i+4j]}$ represents 3 groups of pixels. i=1,2,3,4,5,6,7; j=0,1,2.

Chapter 6

PROTOTYPE IMPLEMENTATION AND SPECIFICATION

This chapter will presents the designed prototype chips from schematic circuit to layout. At the beginning of this chapter, some of the typical layouts are summarized.

6.1 Typical Circuits Layout

1. The layout for 1 pixel:
   The layout for 1 pixel (PD, Edge Detector and memory) is shown in Fig. 6.1. It is in a parallel compact structure regarding Fig. 5.1.

   Since the whole size of the pixel is of 241 × 221 μm, and the size of the Photo Diode is 30.9 × 30.9 μm, therefore its fill factor is 1.79 %.

2. The layout for 1 Transmission Gate memory cell:
   By making use of the gate capacitance, we can implement a short-time memory cell with small layout area. Fig. 6.2 is a layout example for designing this kind of memory cell.

3. The layout for shifting window control logic:
   Fig. 6.3 shows the layout for shifting window control logic. The layout complicity is increased when dealing with a large number of pixels on the focal plane by less number of LPGCP processors.

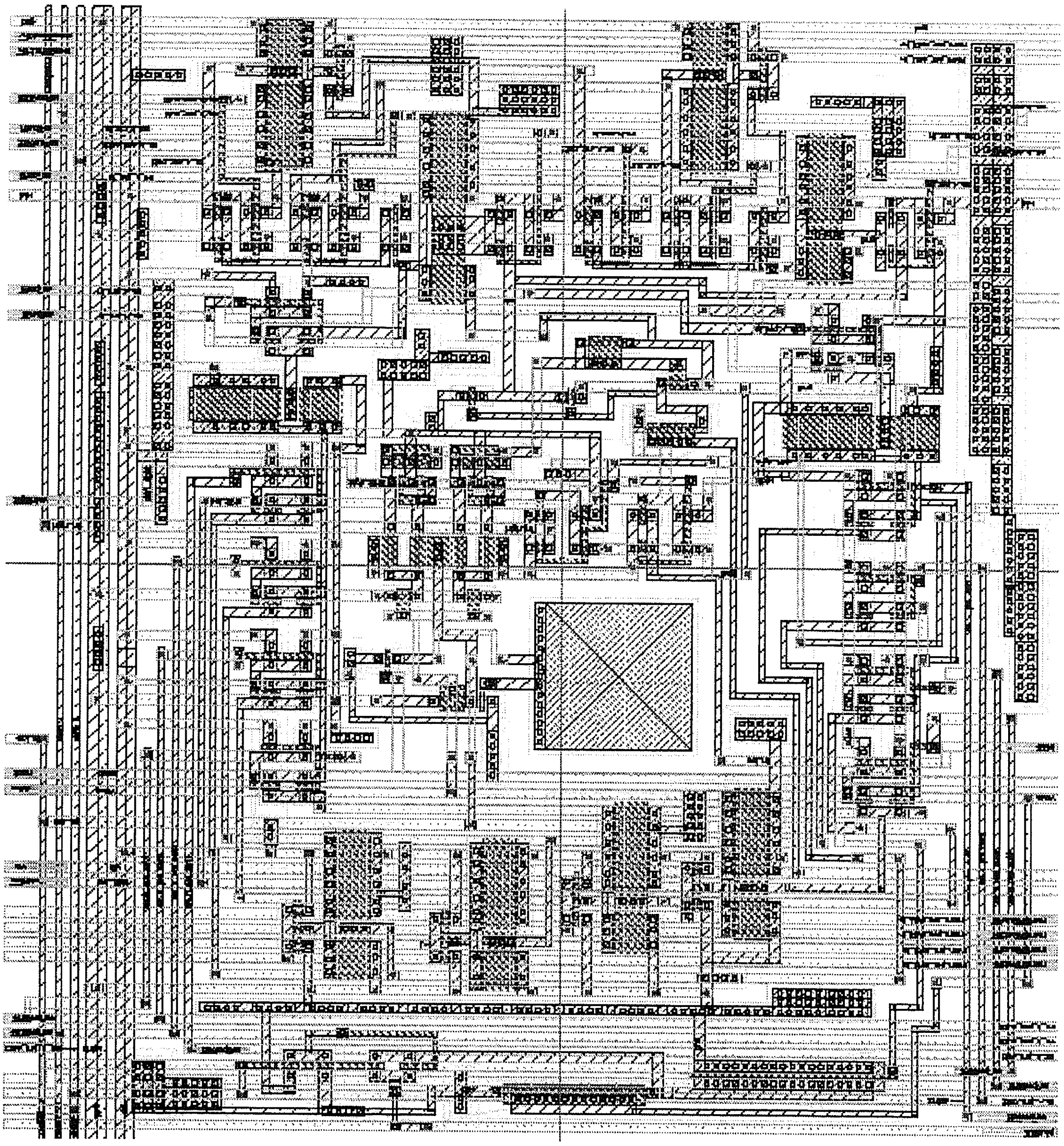


Figure 6.1: Pixel (PD, Edge Detector and memory) layout

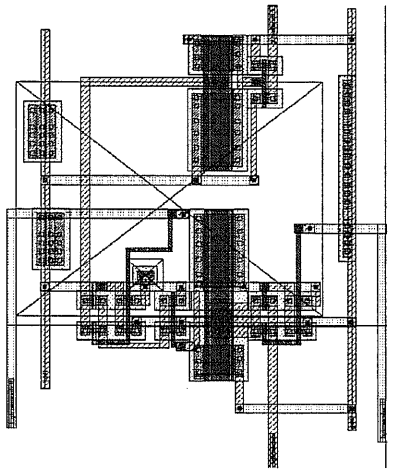


Figure 6.2: Transmision Gate memory cell layout

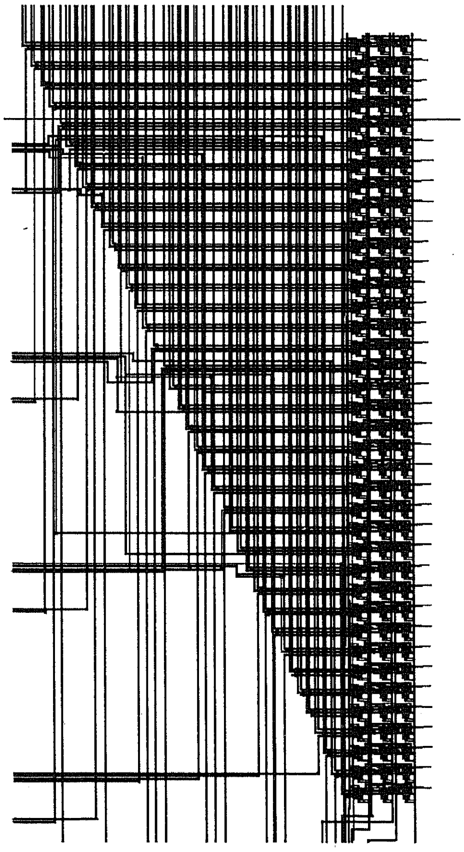


Figure 6.3: Shifting window control logic layout

6.2 Chip Design Summary

3 prototype chips have been designed for this CMOS imager project concerning about motion vector detection.

- The first chip(CHIP-1), shown in Fig. 6.4(schematic) and Fig. 6.5(the layout and the prototype chip) containing 4 × 4 pixel array with PDs, edge detection circuits and memory cells, has been designed by 1-poly 2-metal 0.7 μm CMOS process with a fill factor of 1.7%

- The second chip(CHIP-2) is a LPGCP processor chip, shown in Fig. 6.6 (the layout and the prototype). It can be used for 2-D high speed block matching and for a 2 × 2 pixel v.s. 4 × 4 pixels motion vector estimation.

- The third chip(CHIP-3) is a full functional vision chip with the ability of imaging, edge detection and 2-D motion vector detection. It contains 16 × 16 pixels, 16 × 16 edge detectors, 1K bits memory and 4 LPGCP-ALUs. Its schematic diagram is shown in Fig. 6.7, and the layout and manufactured chip are shown in Fig. 6.8.

The whole chip including pixel, edge detection, memory and LPGCP block matching is designed by 1-poly 2-metals 0.7 μm CMOS process. The main descriptions and design parameters are listed in the Table 6.1.

Table 6.1: summary of the prototype of a 2D motion vector detection sensor

number of pixels	16 × 16 pixels
size of PD (photo diode)	30.9 × 30.9 μ m
size of PD + edge detector + memory	241 ×221μ m
size of the die chip	6499 ×8218μ m
size of block matching	2 × 2 pixels
size of search area	$(\pm 1, \pm 1)$ *pixels*
size of 1 LPGCP ALU	1334.6 ×2924.4μ m
imaging frame rate	100 - 1000
number of trans.	PD & edge: 30 /pixel mem. & control: 100 /1 cell 1 LPGCP Processor: 2486
fill factor	1.79%
power supply	single +5 V
power dissipation (static)	0.3 m W/ 1 LPGCP processor 90 m W/ chip

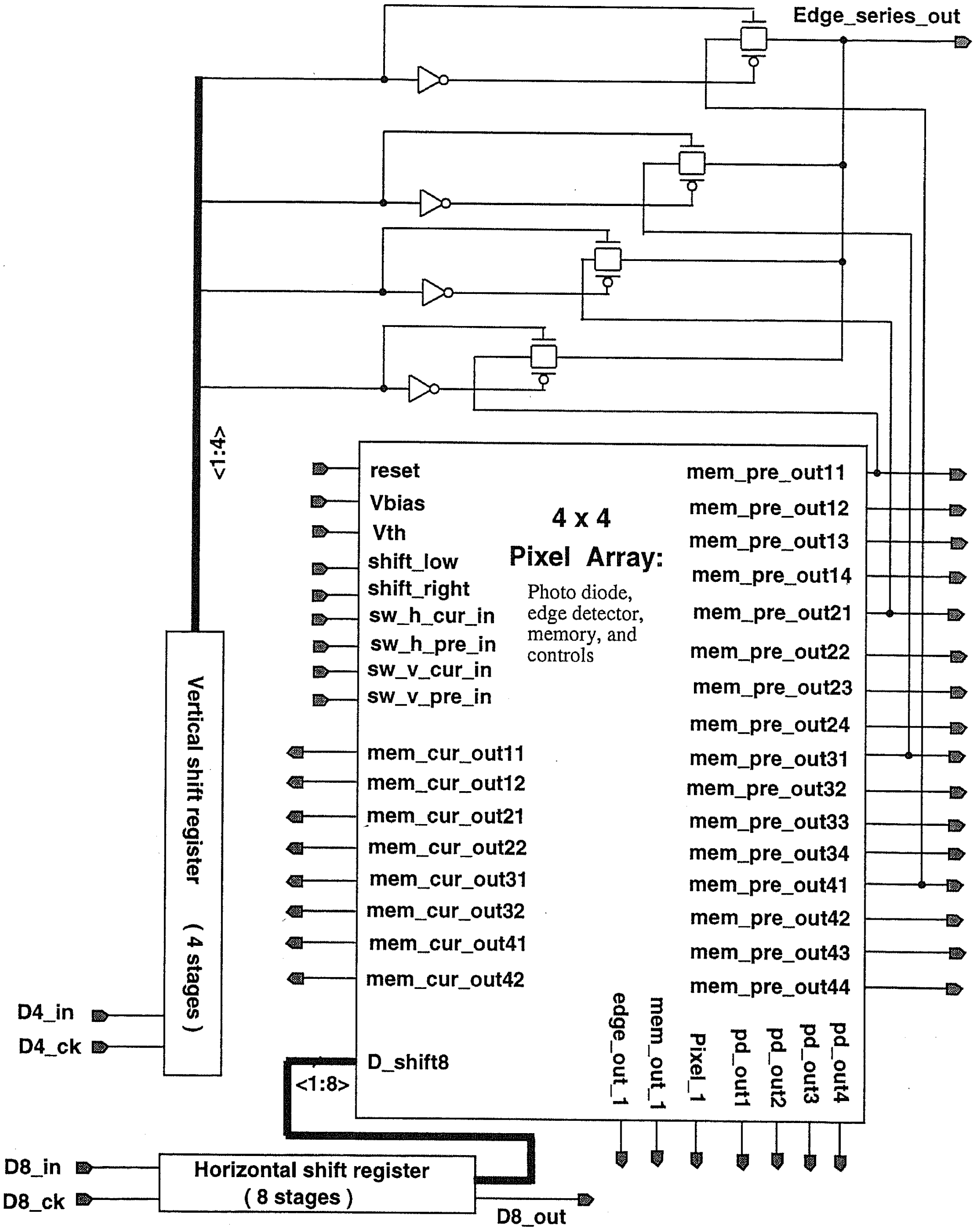


Figure 6.4: 4 ×4 pixel array construction of CHIP-1. It consists of a 4-stage vertical shift register, a 8-stage horizontal shift register, a single circuit of the edge detector, a single circuit of the transmission gate memory, a single circuit of the pixel, and a 4 × 4 pixel array. In every pixel, it includes a photo diode, an edge detector, and a transmission gate memory cell. The fill factor is 1.7% because of the compact parallel architecture.

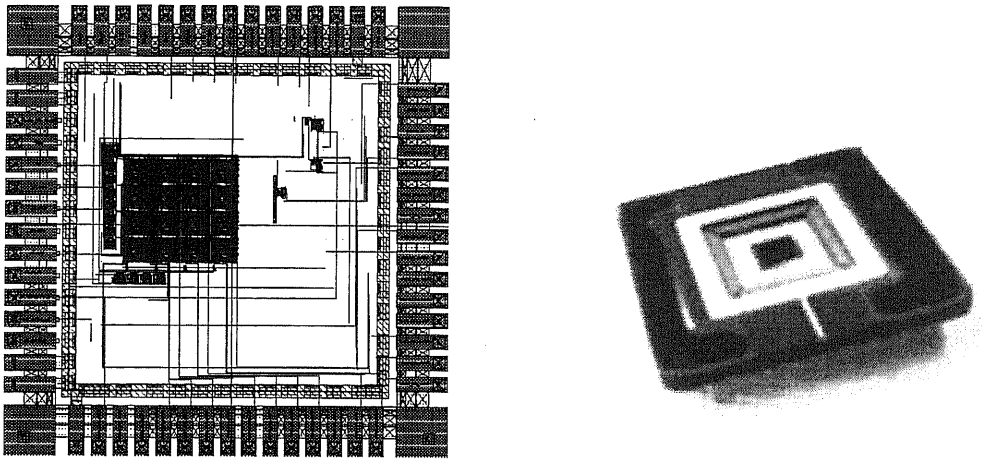


Figure 6.5: The layout and the prototype chip for the 4x4 PD, edge detection, memory

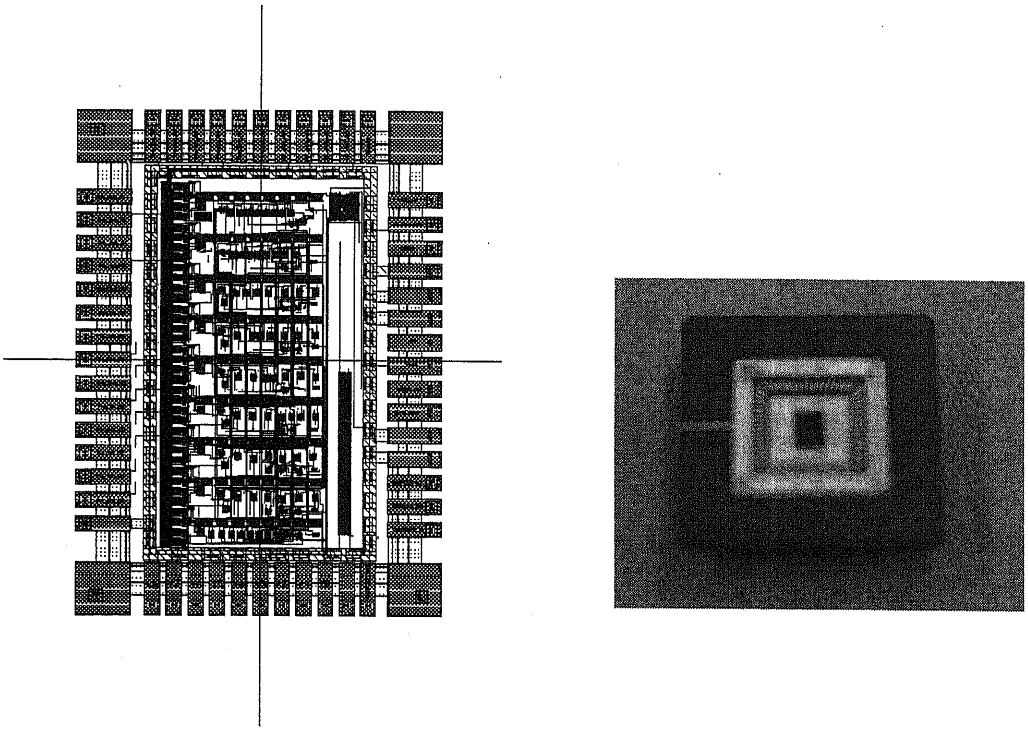


Figure 6.6: The layout and prototype chip for the 2D high speed block matching and motion vector detection in a small area of 2 × 2 pixels in the current frame v.s. 4 × 4 pixels in the previous frame.

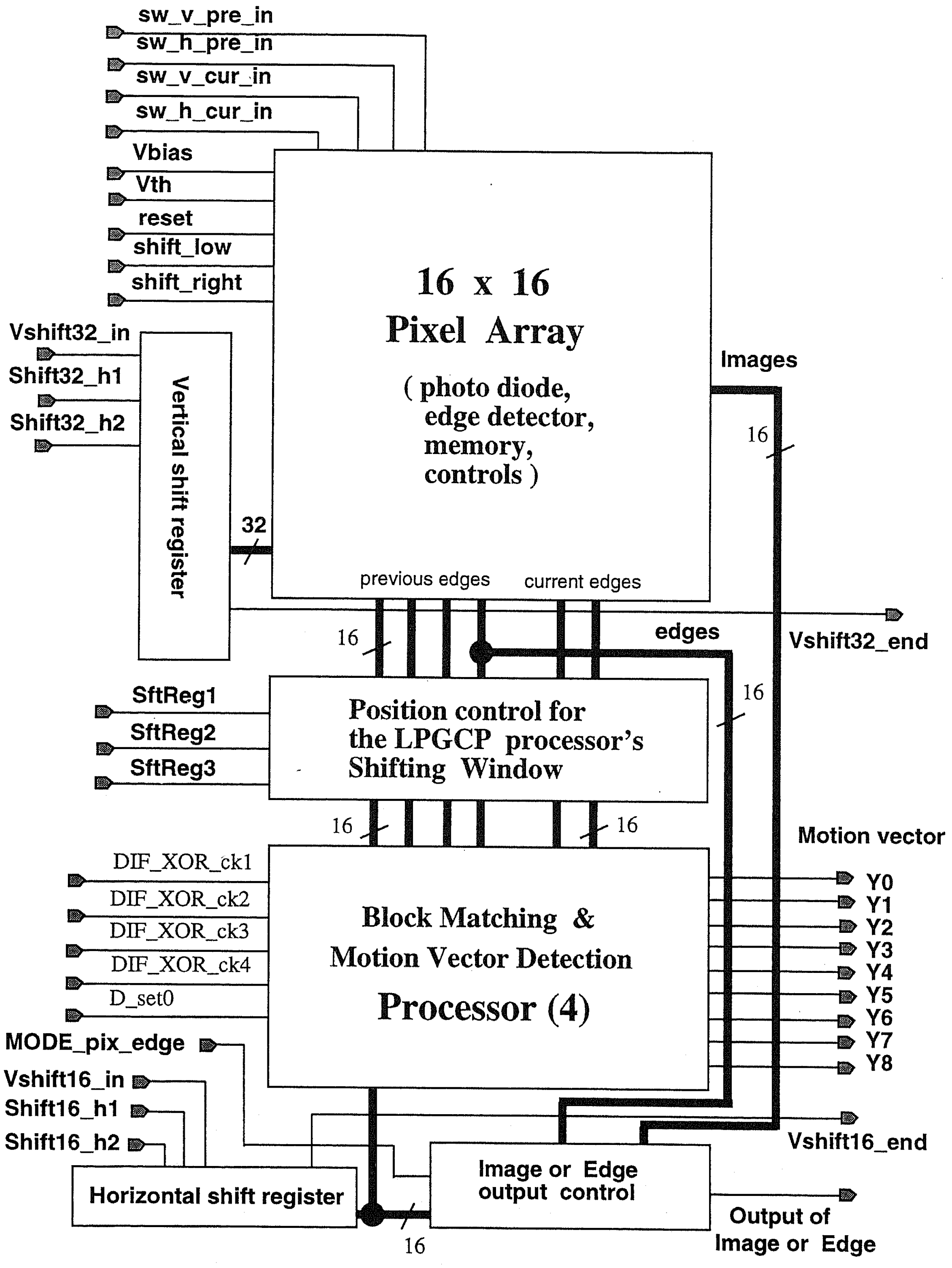


Figure 6.7: 2D motion vector detection construction of CHIP-3: inputs, outputs, and controls for 16 × 16 pixel array, LPGCP processor's shifting window, block matching, motion vector detection and images or edges. There are 3 kinds of output on the chip: the output of images, the output of edges and the output of the motion vectors.

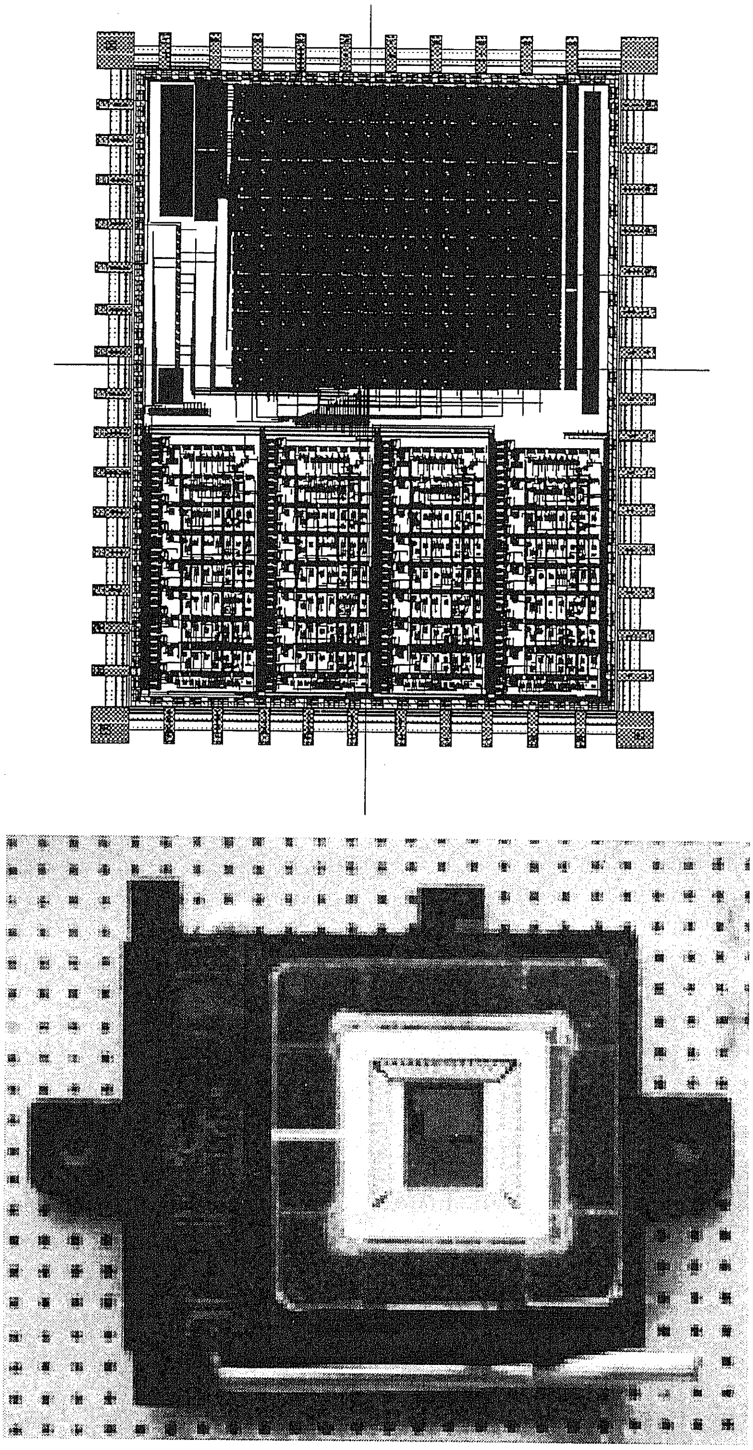


Figure 6.8: The layout and manufactured chip for the 2-D motion vector detection with 16x16 PDs

Chapter 7

EVALUATION AND EXPERIMENT

Three chips regarding about pixels, edge detection, memory, imaging, block matching and motion vector detection have been tested on a working desk as shown in Fig. 7.1. The relavent results are given in this chapter.

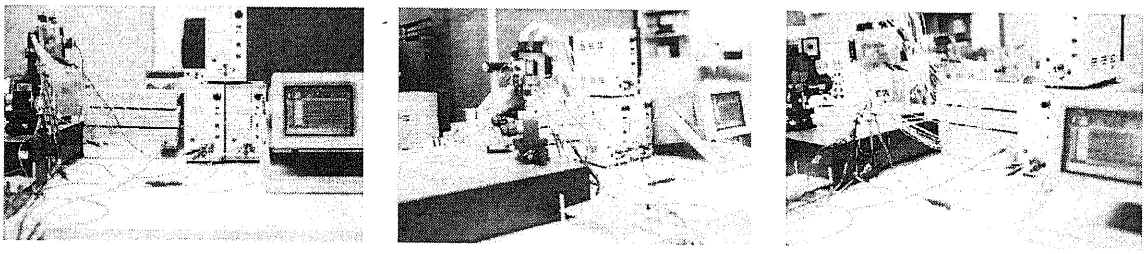


Figure 7.1: The chips evaluation and experiment environment

7.1 Prototype Testing for Pixel: PD, Edge Detection and Memory Cell

A prototype chip(CHIP-1) for the pixel array including PD, edge detection and memory have been designed by 1-poly 2-metals 0.7 μm CMOS process with a fill factor of 1.7%

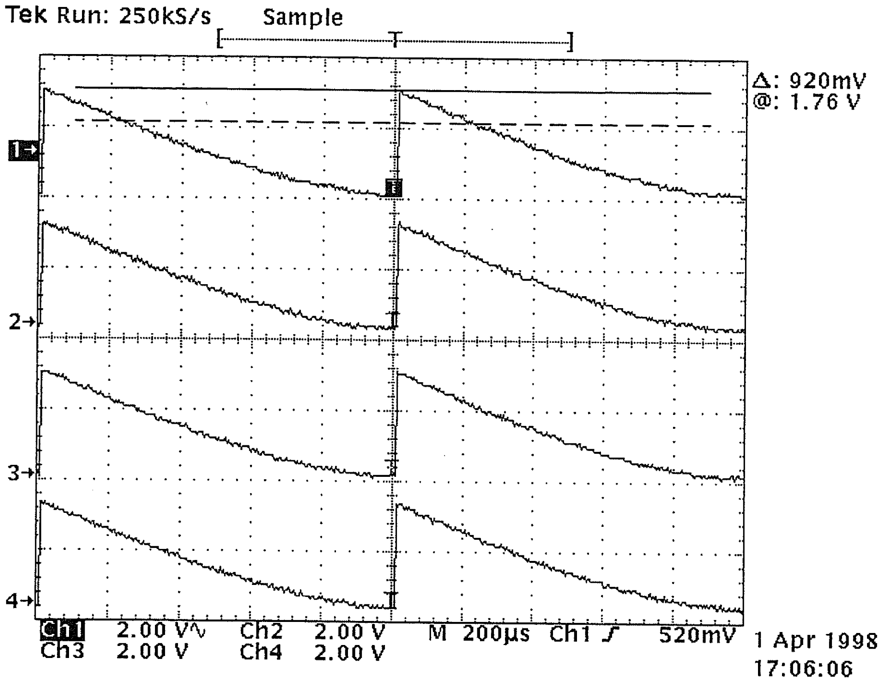


Figure 7.2: the responses of PD1, PD2, PD3, and PD4 to a homogeneous light

according to the highly compacted parallel structure shown in Fig. 5.1.

The first experiment is made for testing the Photo Diode's function of 4 neighboring pixels PD1, PD2, PD3 and PD4 shown in Fig. 5.7 (the lower side figure). The respones to light beam are tested in Fig. 7.2, Fig. 7.3, Fig. 7.4, and Fig. 7.5.

The second experiment is made in such a way that, one input of the absolute value differentiator is fixed at $V_{in1} = 1.761V$ (channel 2), the other input is a discharging output of a PD(channel 3), when we change the threshold value (V_{th}) of the edge detection circuit in Fig. 5.3, a series results, shown in Fig. 7.6, Fig. 7.7, Fig. 7.8, and Fig. 7.9, are obtained. Here Channel 1 is the reset sequence, and channel 4 represents the detected edge.

The third experiment is made for testing the single short-time GT memory cell, reference to Fig. 5.12. The test results are shown in Fig. 7.10 and Fig. 7.11. The input of the memory is a bit stream 1 or 0. Bit signal '1' or '0' kept in the short-time TG memory as long as 1ms and 10ms are tested successfully on the prototype chip shown in Fig. 6.5, where channel 1 is the input, channel 2 is the $Switch_{in}$, channel 3 is the $Switch_{out}$, and channel 4 is the delayed output. This result shows that the short-time memory cell can be used for the 1000 - 100 frame/s imaging and motion vector estimation application.

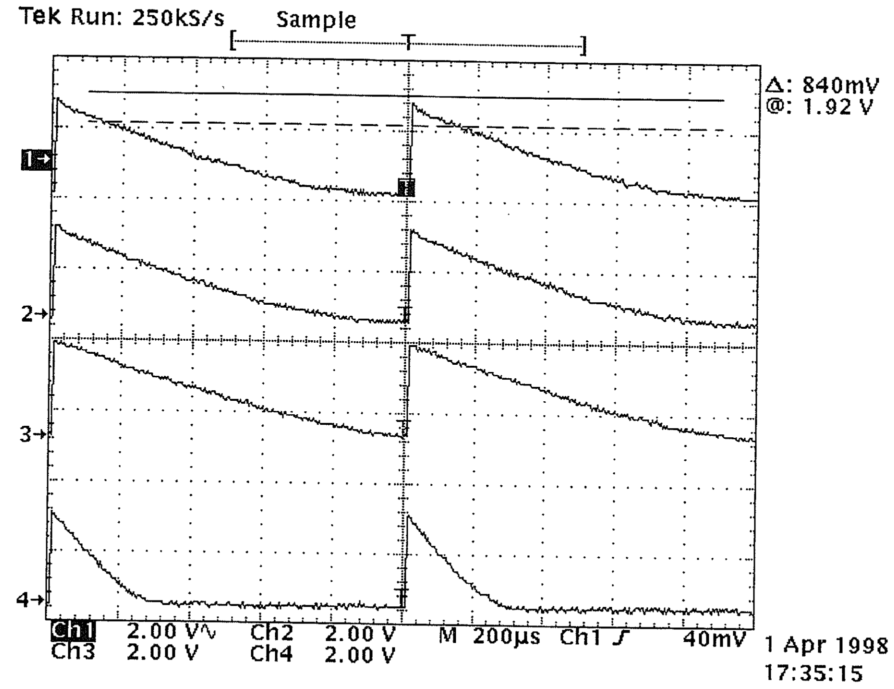


Figure 7.3: Pixel PD4 received stronger light beam

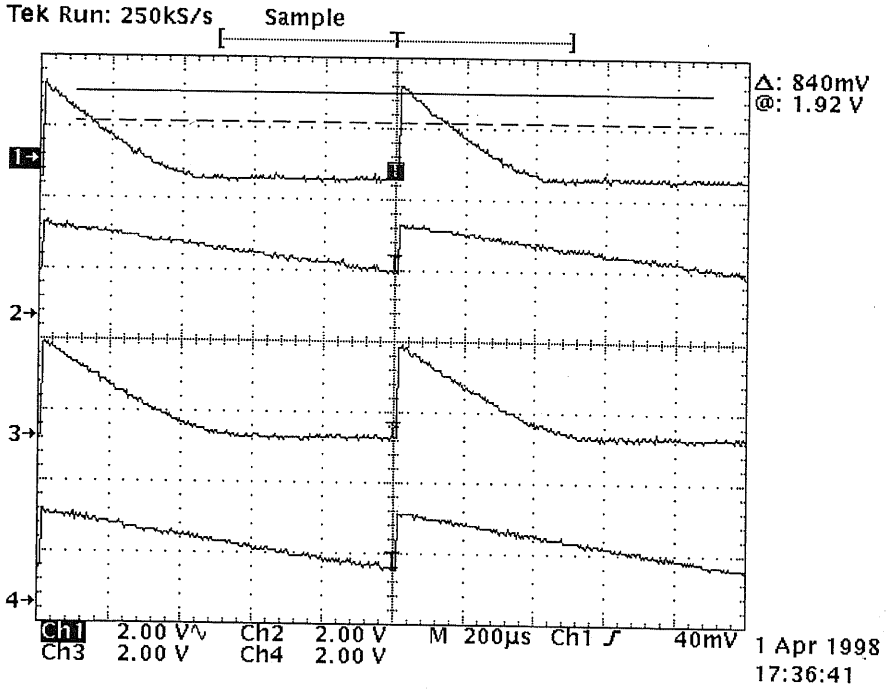


Figure 7.4: Pixels PD1 and PD3 received stronger light beam

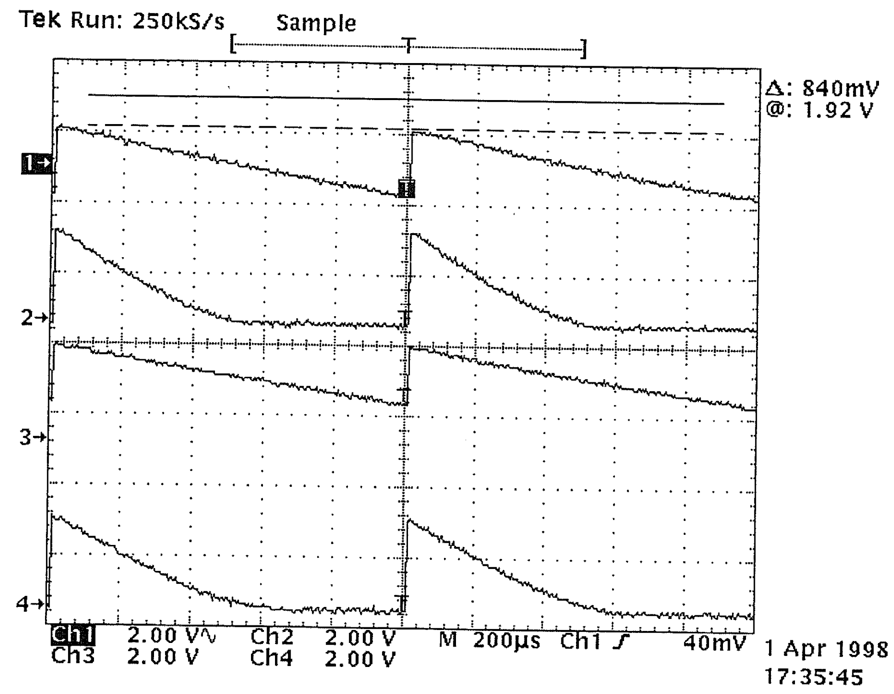


Figure 7.5: Pixels PD2 and PD4 received stronger light beam

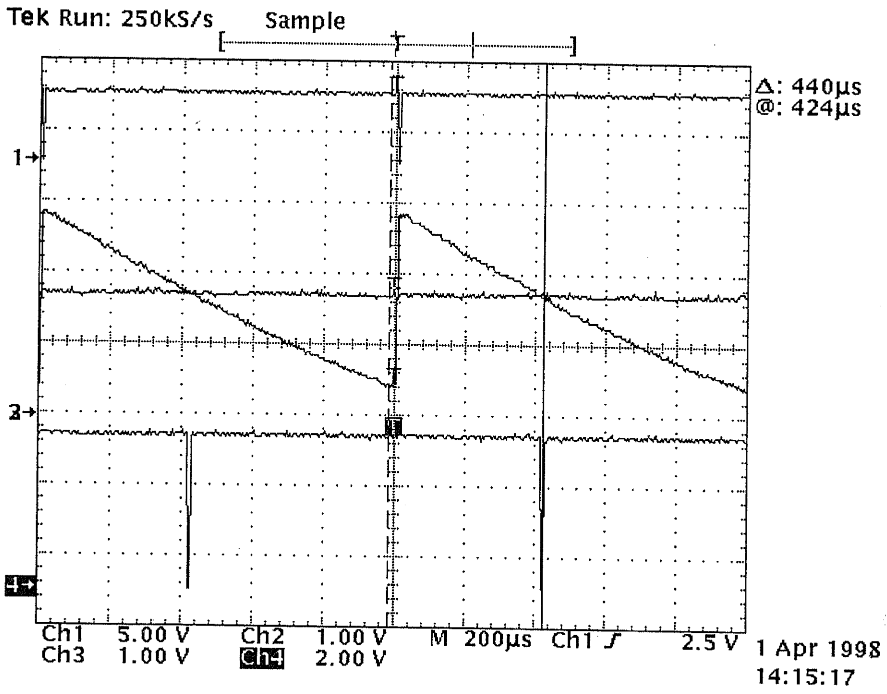


Figure 7.6: edge detection 1: Vth=3.519 V

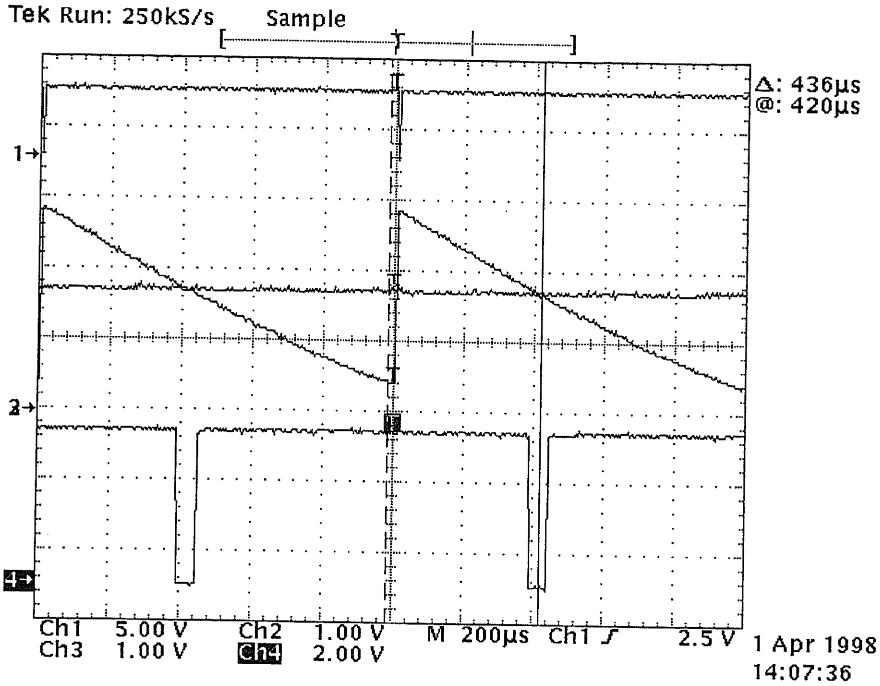


Figure 7.7: edge detection1: Vth=2.917 V

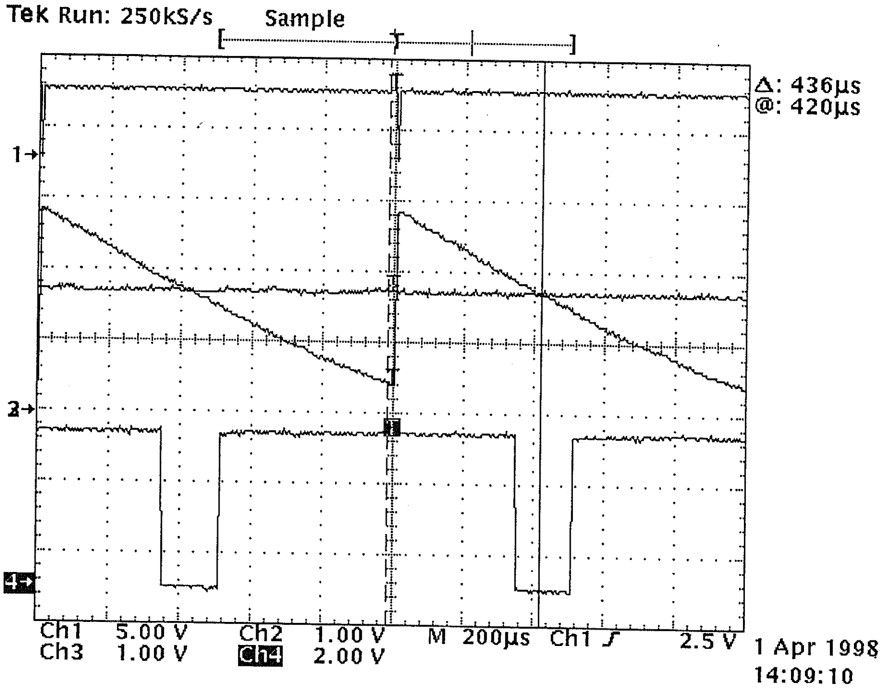


Figure 7.8: edge detection1: Vth=2.150 V

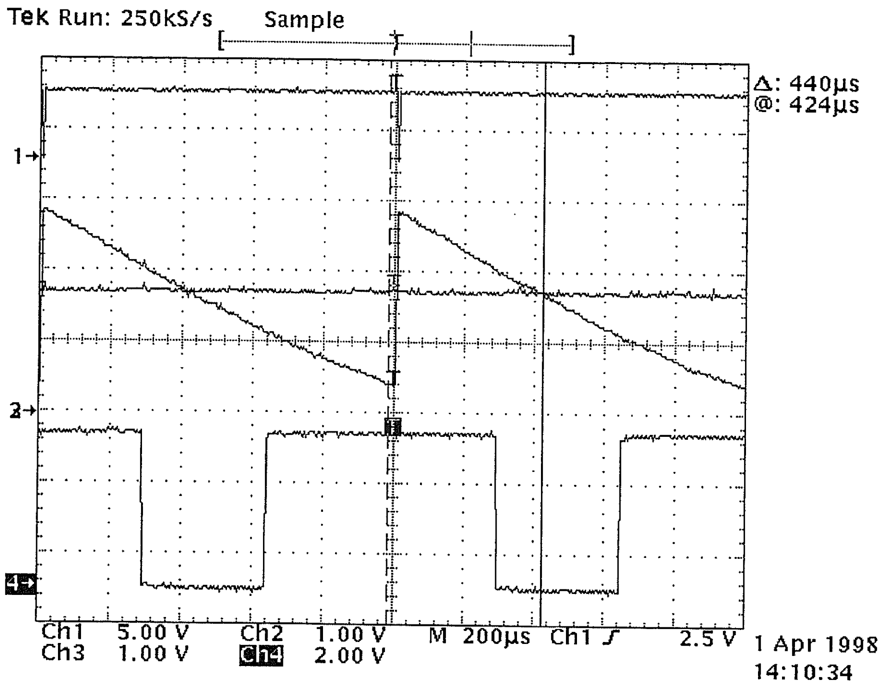


Figure 7.9: edge detection1: Vth=1.380 V

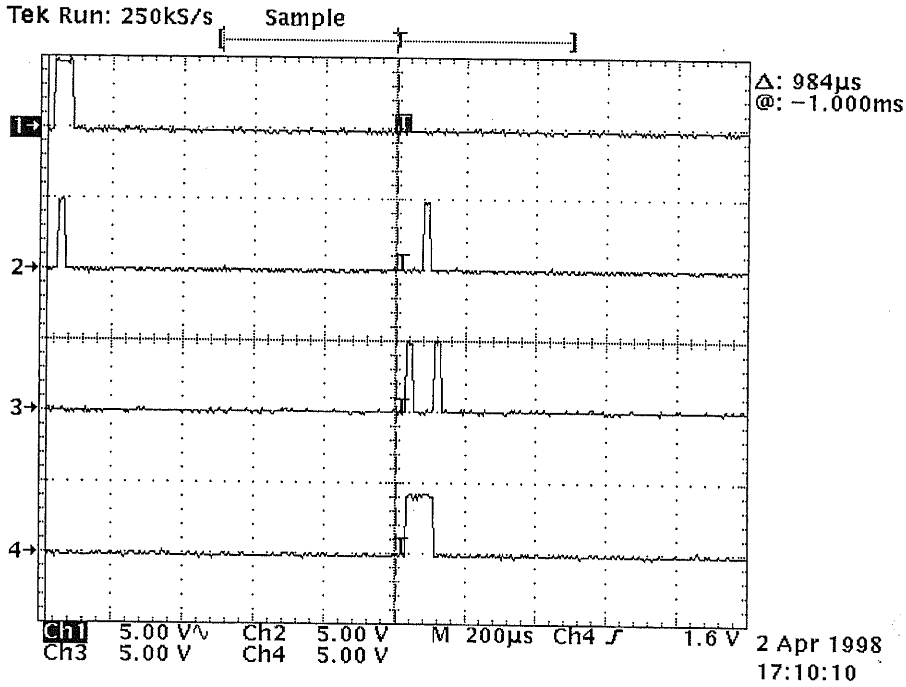


Figure 7.10: a memory cell keeping message “1” for 1ms

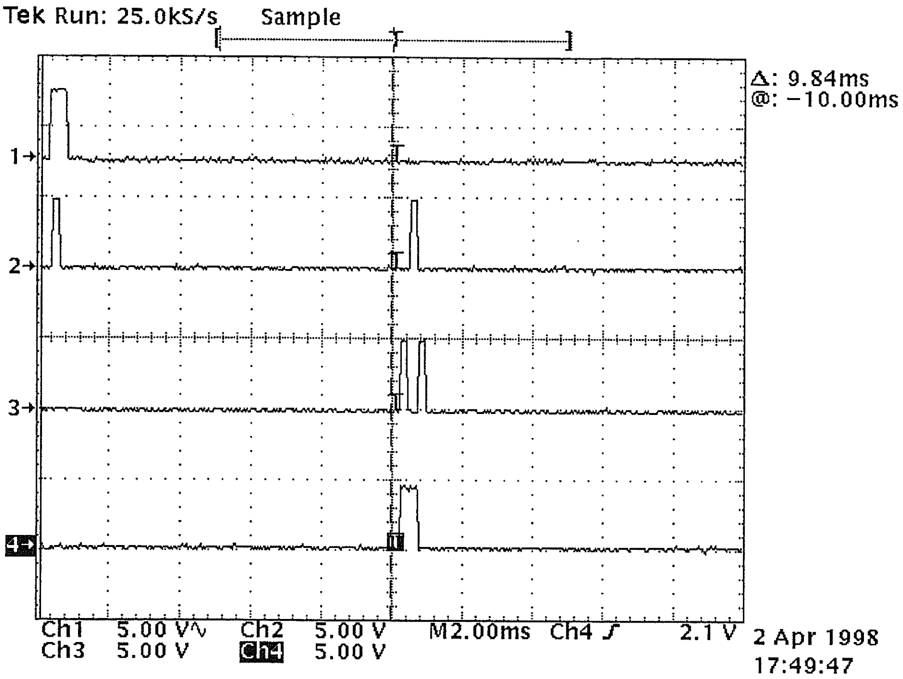


Figure 7.11: a memory cell keeping message "1" for 10ms

7.2 Chip Test for the Shift Registers

The test is made in the CHIP-2 and CHIP-3, the results are shown in Fig. 7.12, 7.13 and 7.14. These results show that the 16 and 32 stages shift registers work well.

The difference of the 32-stages shift register from other traditional ones is that its delayed outputs will be devided into two groups, one is for horizontal processing and the other one is for vertical processing. The load of the shift register is a critical factor. If there are too many of loads connected to the output of the shift register, the shift register may fail to drive the computational circuitry. Therefor, we should pay a great attention to the design of the driving ability of the 32-stages shift register.

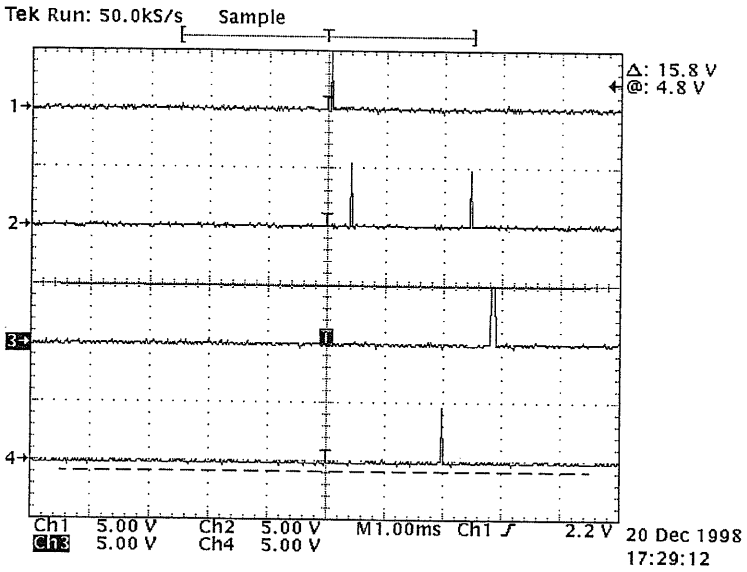


Figure 7.12: Evaluatons for the 32-stages Shift Register and the 16-stages Shift Register. Channel 1 is the input of the 32-stages Shift Register: Vshift32-in; Channel 2 is an other input of the 32-stages Shift Register: Shift-h2; Channel 3 is the output of the 32-stages Shift Register: Vshift32-end; Channel 4 is the output of the 16-stages Shift Register: Vshift16-end.

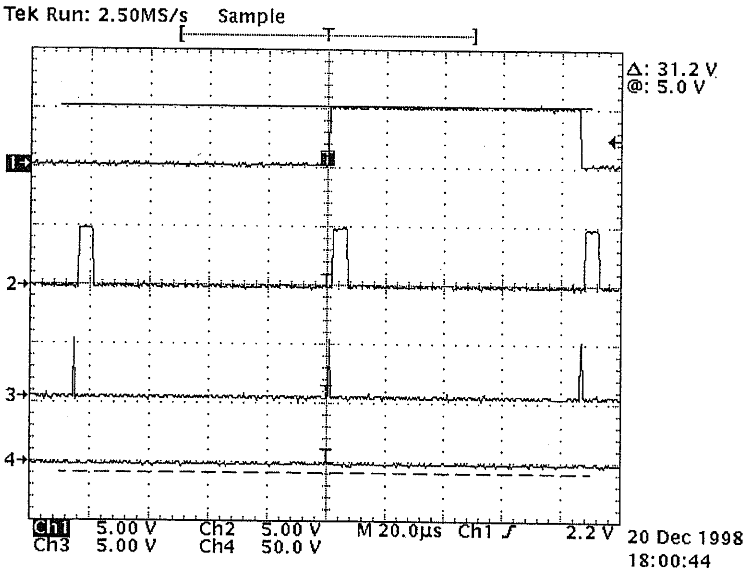


Figure 7.13: An enlarged diagam for testing the 32-stages shift register. Channel 1 is the output of the 32-stages Shift Register: Vshift32-end; Channel 2 is the input of the 32-stages Shift Register: Shift32-h1; Channel 3 is the input of the 32-stages Shift Register: Shift32-h2; Channel 4 is the input of the 32-stages Shift Register: Vshift32-in (out of seeing range).

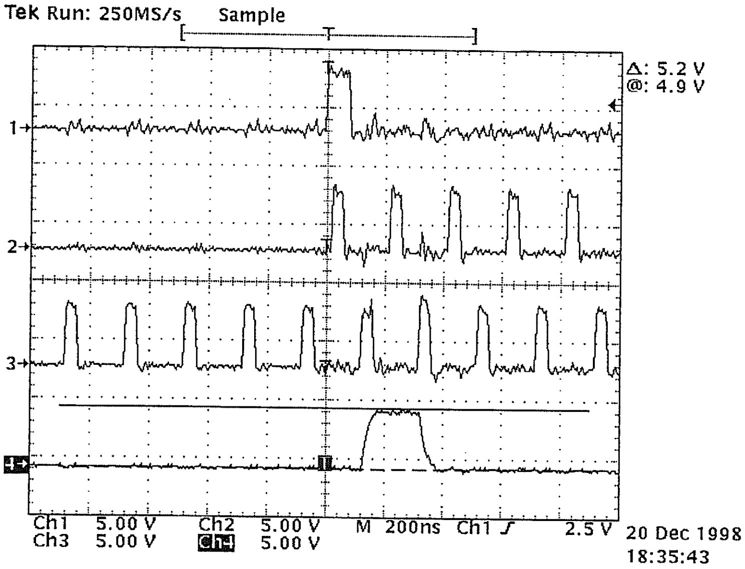


Figure 7.14: An enlarged diagam for testing the 16-stages shift register. Channel 1 is the input of the 16-stages Shift Register: Vshift16-in; Channel 2 is the input of the 16-stages Shift Register: Shift16-h1; Channel 3 is the input of the 16-stages Shift Register: Shift16-h2; Channel 4 is the output of the 16-stages Shift Register: Vshift16-end.

7.3 Chip Test for Imaging

In our prototype vision chip, the electrical level of the output voltage is carefully adjusted to meet the need with a good range. A practical single pixel circuit and a 16 × 16 pixel array constructed by the practical single pixel circuit are tested in this section.

7.3.1 Practical Pixel Circuit and the Output Test

A practical single pixel circuit is shown in Fig. 7.15.

The output of this practical pixel circuit is demonstrated in Fig. 7.16. We can see from these test results that the output has a large dynamic range within 1.6 V ~ 4 V by using two stages of buffers. The first stage is a NMOS buffer, and the second stage is a PMOS buffer.

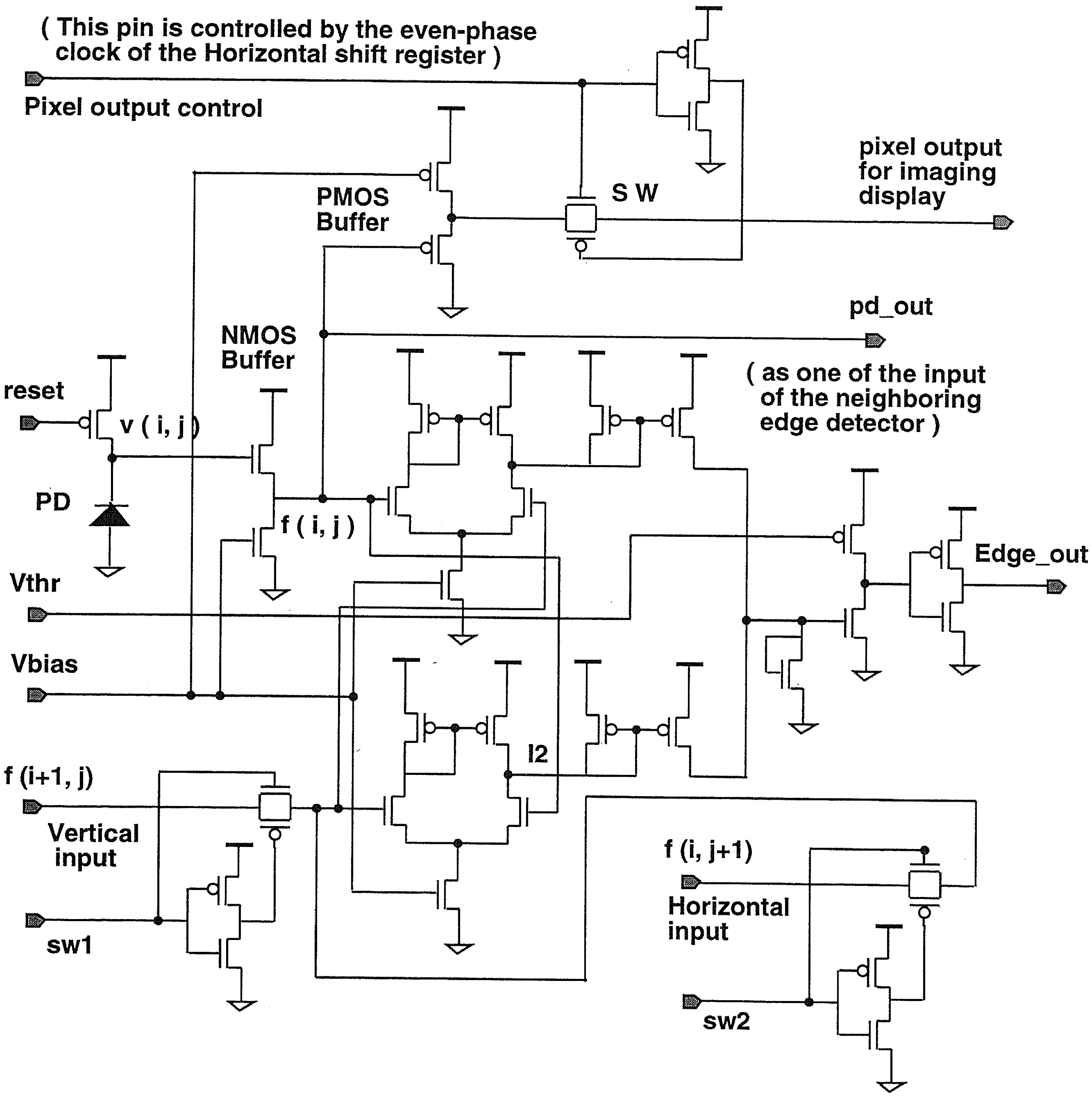

(This pin is controlled by the even-phase clock of the Horizontal shift register)
Pixel output control
PMOS Buffer
S W
pixel output for imaging display
pd_out
(as one of the input of the neighboring edge detector)
NMOS Buffer
reset
v (i, j)
PD
f (i, j)
Vthr
Edge_out
Vbias
f (i+1, j)
I2
Vertical input
sw1
f (i, j+1)
Horizontal input
sw2

Figure 7.15: Practical pixel circuit with a PhotoDiode, 2 buffers(NMOS buffer and PMOS buffer), output switch, and the output.

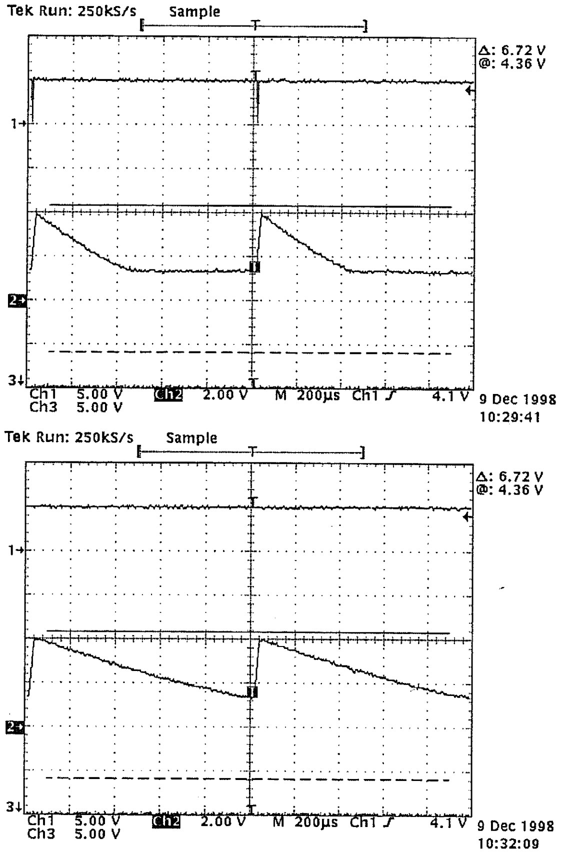


Figure 7.16: Evaluatons for the pixels. The output of the practical pixel circuit has a large dynamic range within 1.6 V ∼ 4 V by using two stages of buffers. The first stage is a NMOS buffer, and the second stage is a PMOS buffer.

7.3.2 Imaging by Using of a 16 × 16 Pixel Array

Our prototype vision chip has multifunctions, the ability of imaging is one of the by-products in the CHIP-3. Since there is no specific Sample/Hold circuit for the imaging function in the CHIP-3, we have to work out an other way to read the pixels from the vision chip.

The imaging circuitry is driven by the even-phase output of the 32-stages Vertical Shifter Registers described in Section 7.2. The principle circuit constructure is shown in Fig. 7.17. The timing sequence driving the chip for imaging is shown in Fig. 7.18.

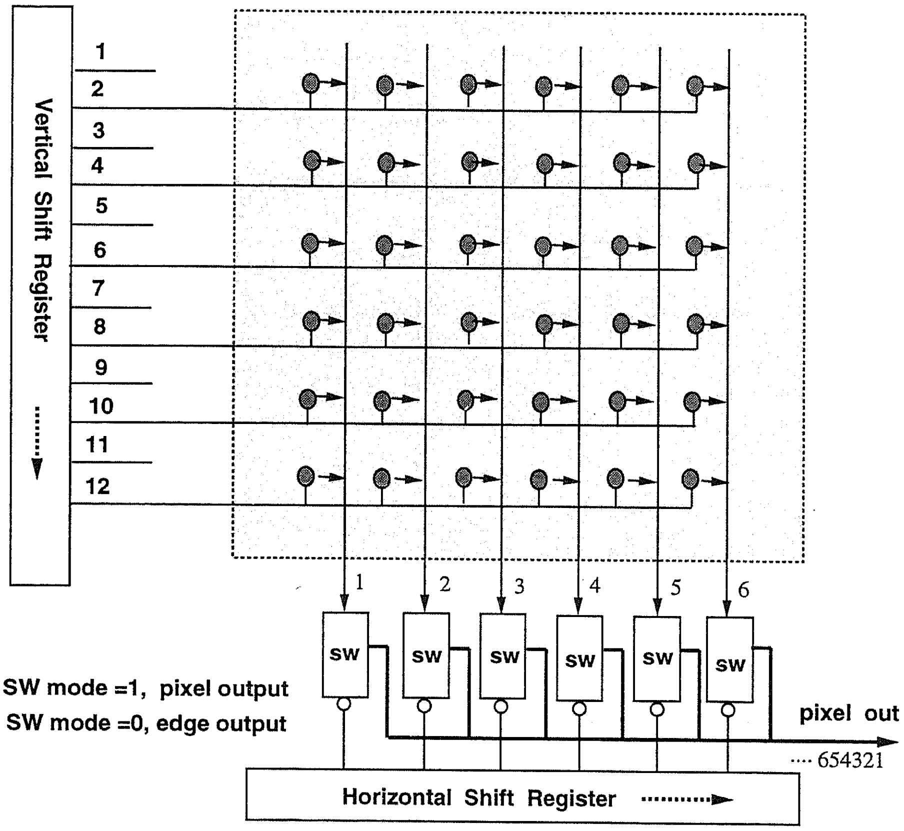


Figure 7.17: Both the even and odd outputs of the Vertical shift register will be used for horizontal/vertical edge detection and blockmatching. But here, only the even output of the Vertical shift register is specially used for driving the imaging output.

When we project a pattern, such as a T shown in Fig. 7.19, on the vision chip, we can get a re-constructed image on a NTSC mode television screen. The output is inverted in such a way that th e brighter area is shown darker, and vice versa. Test shows that although there are only 16 × 16 photo diodes integrated on the focal plane,

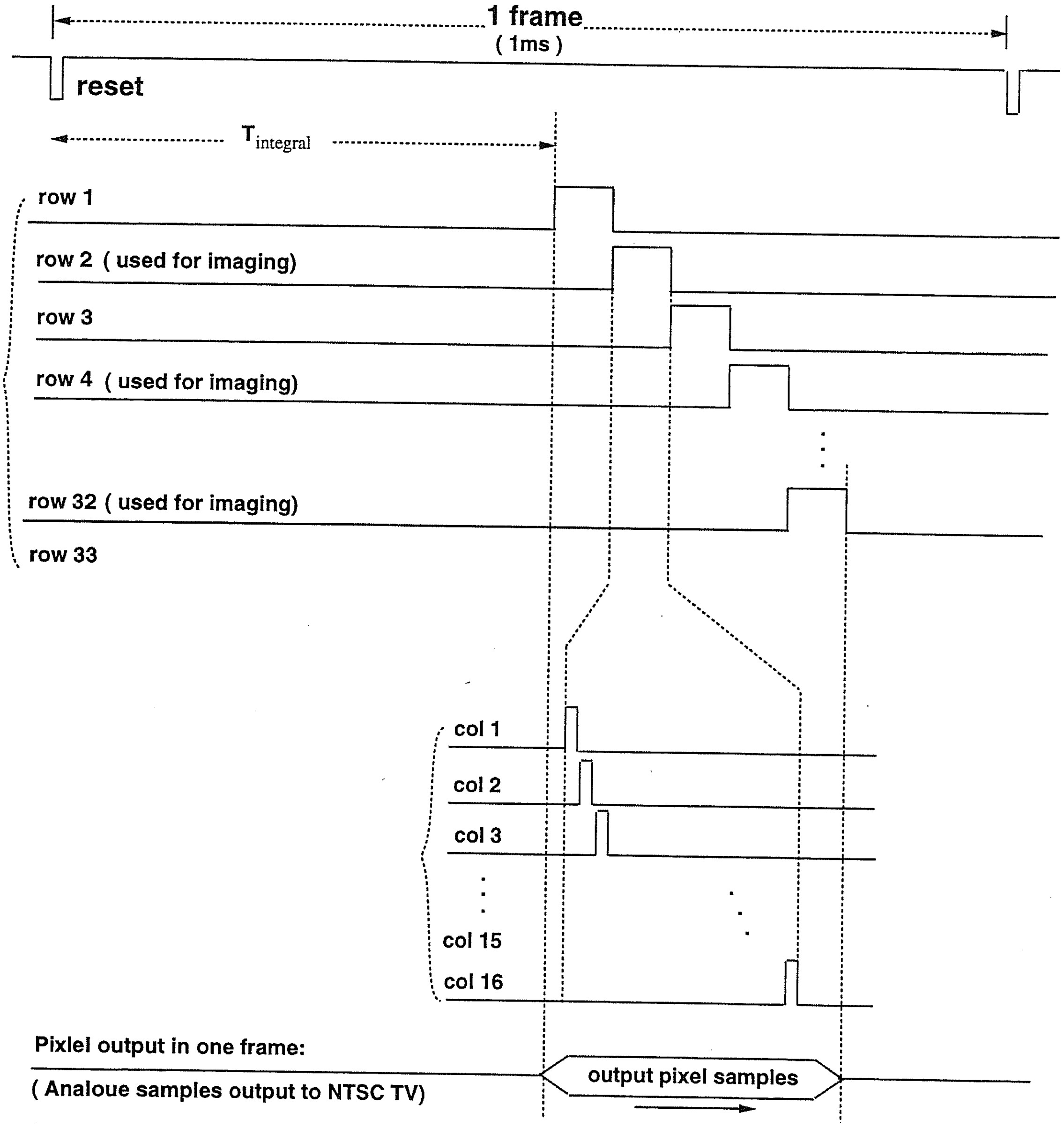

1 frame
(1ms)
reset
T$_{integral}$
row 1
row 2 (used for imaging)
row 3
row 4 (used for imaging)
row 32 (used for imaging)
row 33
col 1
col 2
col 3
col 15
col 16
Pixlel output in one frame:
(Analoue samples output to NTSC TV)
output pixel samples

Figure 7.18: Timing driving sequence for reading out images from the vision chip to a NTSC Television screen.

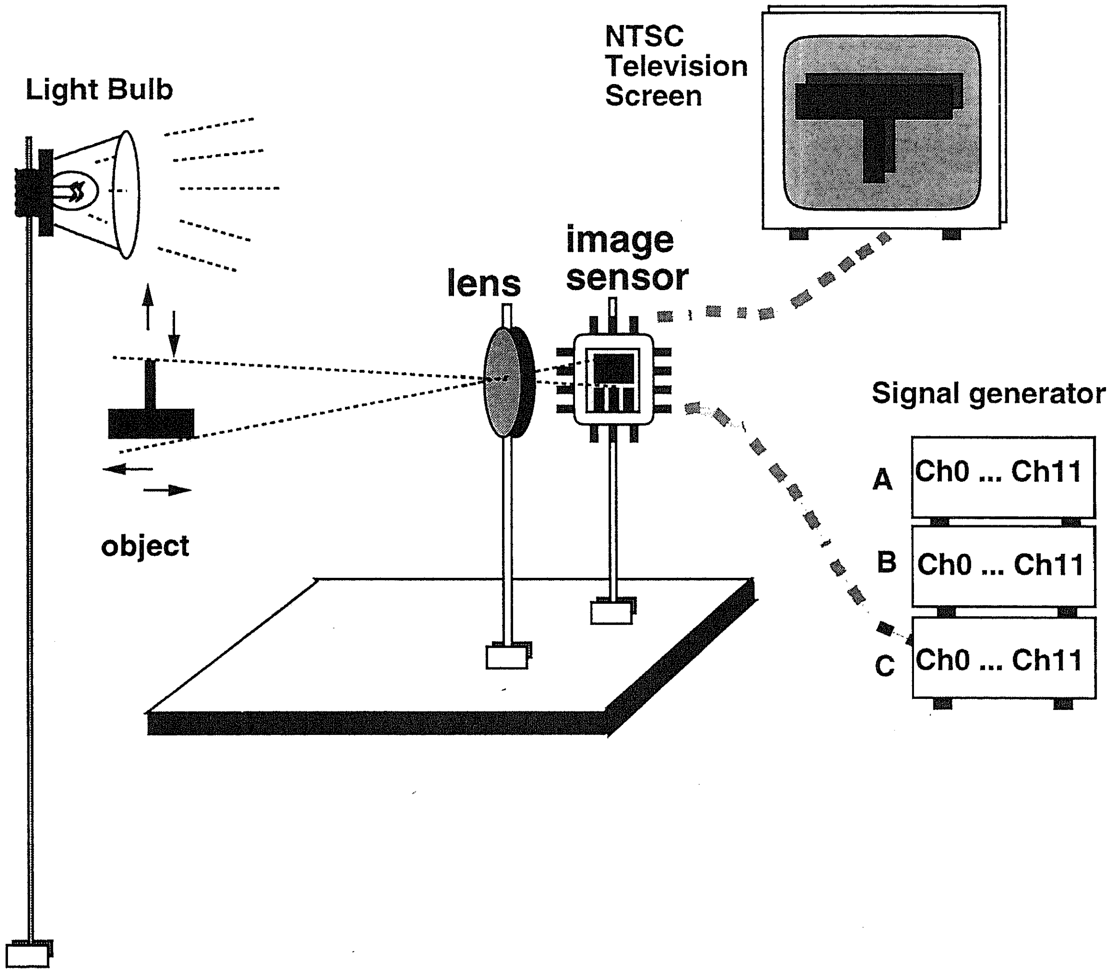


Figure 7.19: This is a working desk for evaluation of the imaging chip. A pattern T is projected onto the sensor focal plane through a suitable lens under a mount of light lumination. There 3 sets of control signals from the signal generator are used for driving the chip. The output is connected to a NTSC mode television for displaying the recovered image.

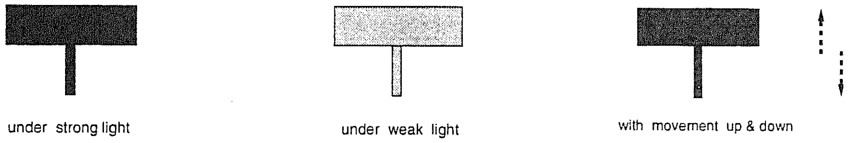


Figure 7.20: Pattern T with different light strength and up & down movement are used to test the vision chip's imaging ability on the focal plane.

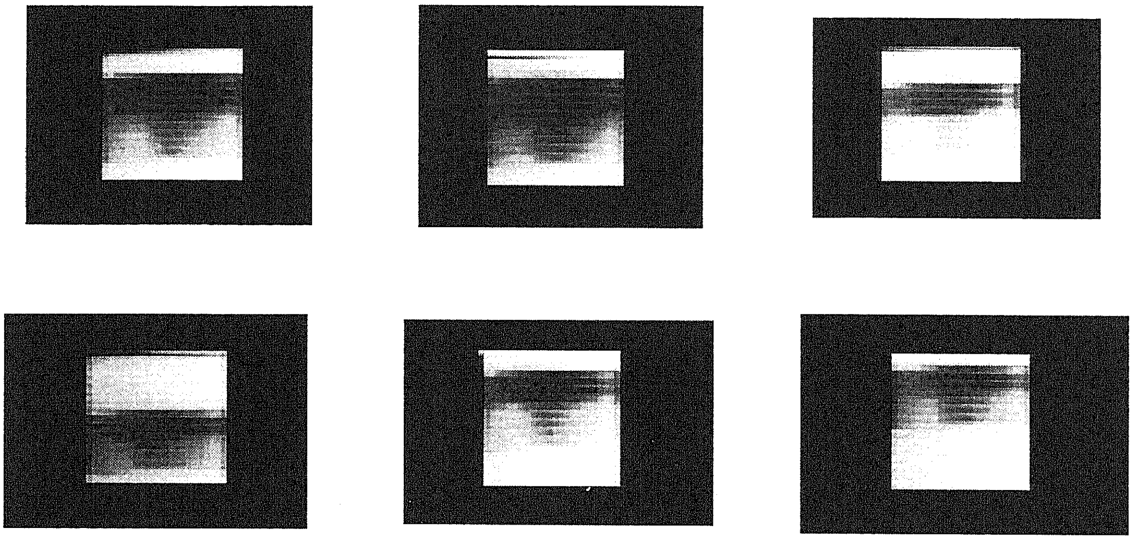


Figure 7.21: The top 3 pictures were taken under a proper light, strong light, and weak light, respectively. The low 3 pictures were taken with the pattern T having an up and down movement. Although the prototype has only 16 × 16 pixels, we can see the T and its movement in the images. Notice that the output is inverted such that the brighter area is shown darker, and vice versa.

we can still obtain a visible recovered image on the TV screen. The other noise source in the recovered image is due to the specific structure which is specially designed for the edge detection and motion vector detection. Fig. 7.20 shows the combination of the pattern T with different light strength and with an up and down movement. Fig. 7.21 shows the test results, which is specially driven by the even-phase output of the Vertical shift register for $SW_{pixelout}$, and by the Horizontal shift register (Col 1, Col 2, ..., Col 16) for pixel-out both depicted in Fig. 7.17 and Fig. 7.18.

7.4 Chip Test for the 2D Motion Vector Detection

The experiment shows that CHIP-2 is successful for 2D block matching and fast motion vector estimations in a small area of 2 × 2 pixels in the current frame v.s. 4 × 4 pixels in the previous frame. The whole vector estimation on the 16 × 16 pixels focal plane is just a combination based on this successful key part.

7.4.1 The Timing Sequence Driving the 2D Motion Vector Detection

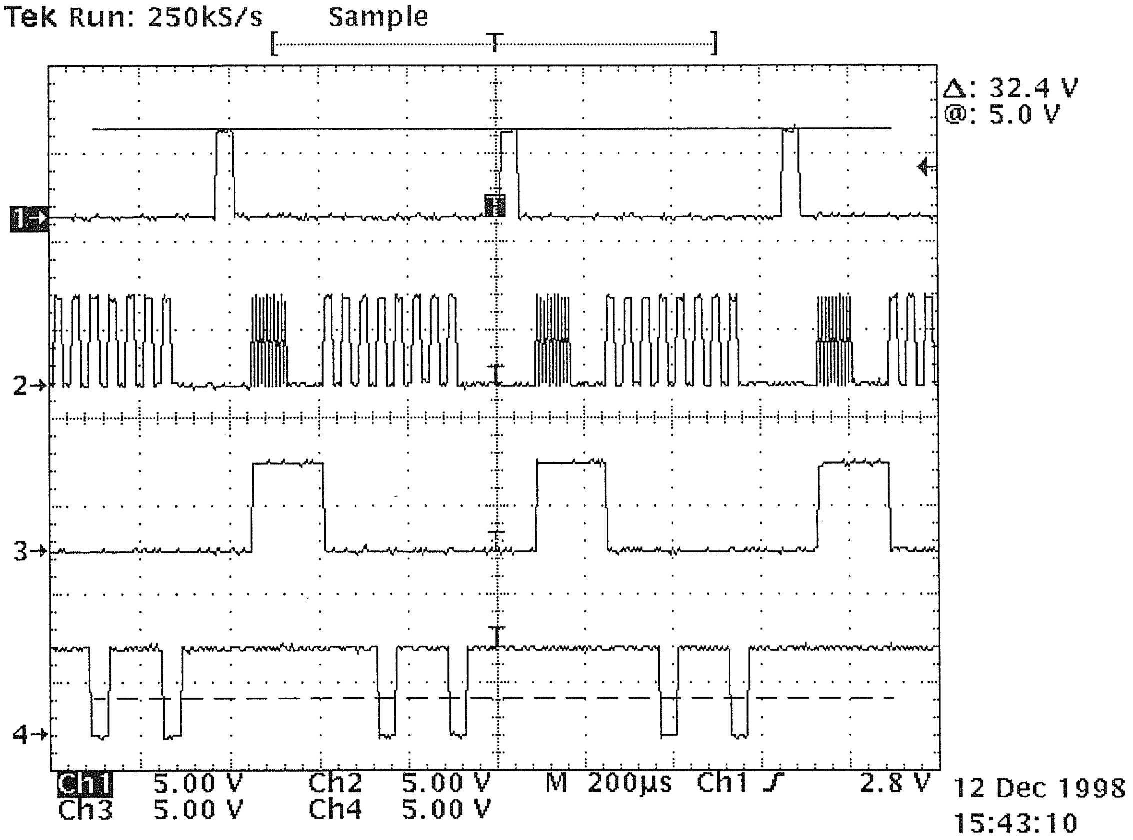


Figure 7.22: The timing sequence driving the 2D motion vector detection. Channel 1 is the sampling sequence $\{Syn - Y_i - reg\}$ for output motion vectors; Channel 2 is the clock $\{ACC - ck\}$ for the accumulator ACC; Channel 3 is the a clear Zero clock $\{D - set0\}$ for the registers; and Channel 4 is the clock $\{DIF - XOR - ck_i\}$ (i=1,2,3,4) for binary block matchings.

The timing sequence driving the 2D block matching and motion vector detection is shown in Fig. 7.22. These timing sequences are a fatal factor for designing a successful motion vector detection. Any small mistakes in designing any of these timing sequences may bring a big failure. The corresponding functional circuit can be refered back from Fig. 5.24.

7.4.2 Motion Vector Estimation by Using Fixed Input of Edge Images in the Current Frame and Random Input of Edge Images in the Previous Frames

According to the LPGCP architecture, the function of motion vector detection needs 4 signal lines for the edges in previous frame and 2 signal lines for the edges in the current frame. Reference to the tree structure of the 4-bits memory in Fig. 5.13, all the input signals in Fig. 7.23 and the 1st figure in Fig. 7.24 will be represented by 5 groups, i.e.
{ Pre-4th-h1, Pre-3rd-h1, Pre-2nd-h1, Pre-1st-h1 } in the 1st picture in Fig. 7.22,
{ Pre-4th-h2, Pre-3rd-h2, Pre-2nd-h2, Pre-1st-h2 } in the 2nd picture in Fig. 7.22,
{ Pre-4th-h3, Pre-3rd-h3, Pre-2nd-h3, Pre-1st-h3 } in the 3rd picture in Fig. 7.22,
{ Pre-4th-h4, Pre-3rd-h4, Pre-2nd-h4, Pre-1st-h4 } in the 4th picture in Fig. 7.22,
{ Cur-2nd-h1, Cur-1st-h1, Cur-2nd-h2, Cur-1st-h2 } in the 1st picture in Fig. 7.23.

For instance, among the 4 signal lines for the edges of previous frame, if one of the signal lines has the bit stream of {10010010}, the horizontal edge stream is its odd position's digit {1001} and the vertical edge stream is its even postion's digit {0100}, which is illustrated in the first figure of Fig. 7.23. The other inputs have the same arrangement for the bit streams in Fig. 7.23. The 1st figure in Fig. 7.24 is a constant input of edges in the current frame, i.e. {1, 0, 0, 1}.

The 2nd figure in Fig. 7.24 shows the output result of the motion vector detection. According to the priority decision rule in Fig. 4.10, Y3 is detected first, and then Y0, Y1, Y0, Y3, etc.

These experiment results are identical to the theoretical analysis by using of a fixed input edge images in the current frame and random input edge images in the previous frame in Fig. 7.25. The 1st array indicates that Y3 (index No. 2) is detected, the 2nd array indicates that Y0 (index No. 5) is detected, and the 3rd array indicates that Y1 (index No. 1) is detected, and so on. In this way an array trace can be obtained on the focal plane, i.e. a series of motion vectors can be estimated by using a specific fixed block in the current frame block matching with any random input edge images in the previous frame.

In the case of the application of real time image coding off-chip, we can use an auxiliary high speed memory or buffer to store the output of the motion vector indexes.

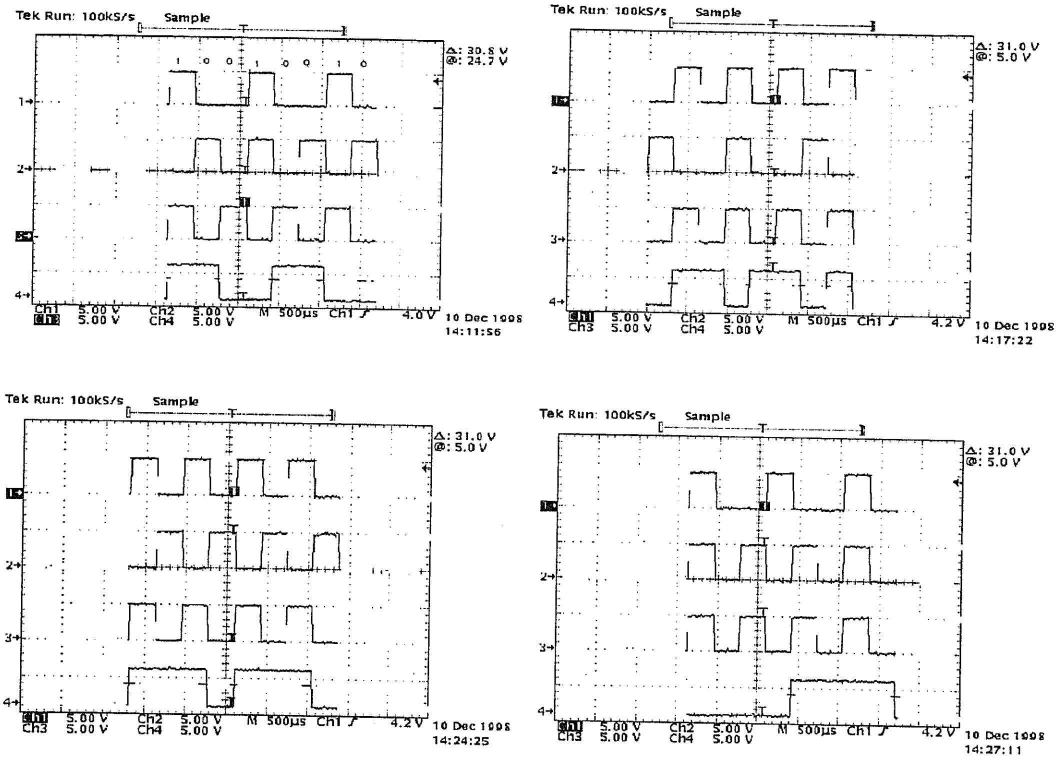


Figure 7.23: These 4 pictures are the inputs of the previous frame edge image, which will be combined together with the inputs of the current frame edge image to estimate the motion vectors.

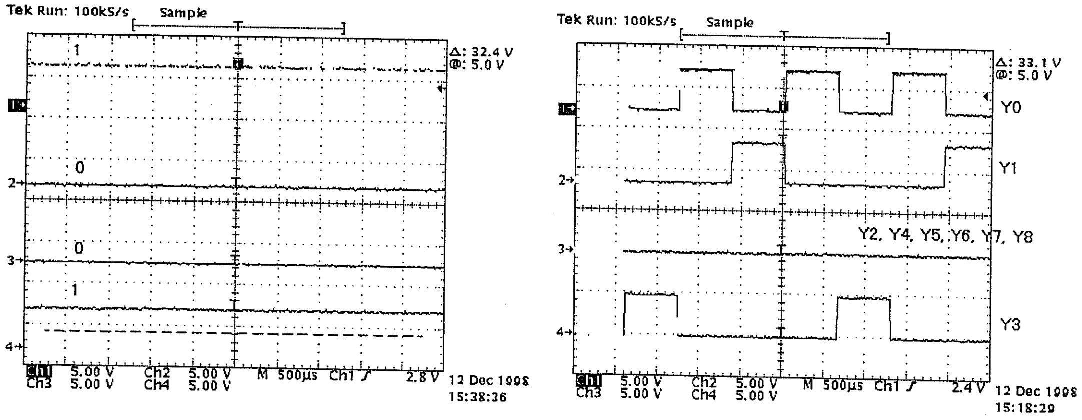


Figure 7.24: The input signals and evaluation results for the motion vector detection. The first picture is a constant input of the current frame edge image. Reference to Fig. 5.18, the motion vectors are detected and output by 9 output lines from Y0 to Y8, In this test, the results are { Y3, Y0, Y1, Y0, Y3, . . .}.

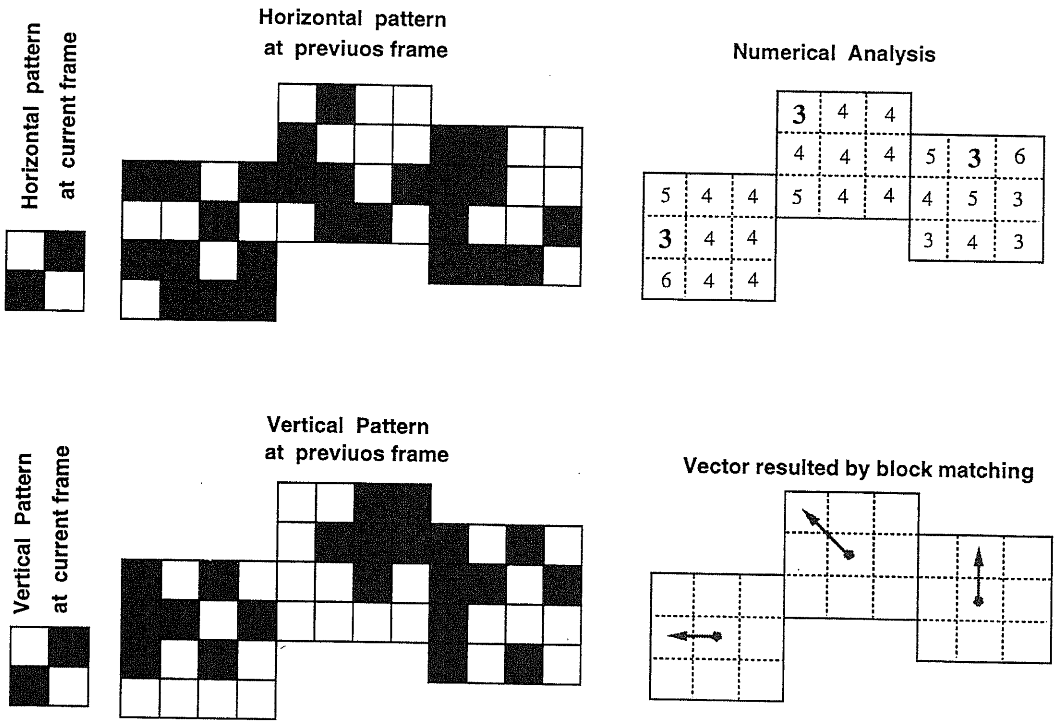


Figure 7.25: The theoretical analysis is consistent to the experiment test in Fig. 7.23 and Fig. 7.24. i.e, the arrays can represent the detected vector { Y3, Y0, Y1}. By tracing the motion vectors, then we can get an equivalent long vector with any possible dgree of angles.

7.4.3 Motion Vector Estimation by Using Random Input of Edge Images in the Current Frame and Random Input of Edge Images in the Previous Frames

We can also use the random input of edge images in the current frame and random input of edge images in the previous frames to make a testing experiment on our processor chip.

Fig. 7.27, Fig. 7.28, Fig. 7.29 and Fig. 7.30 are the binary image matrix regarding inputs and outputs corresponding and identical to the testing signals in Fig. 7.27. The resulted waveforms are shown in Fig. 7.31.

More clearly, for a certain frame, the input signal coming from the horizontal edge memory are

$Cur\_2nd\_h1 = 1;$ $Cur\_2nd\_h2 = 0;$ $Cur\_1st\_h1 = 0;$ $Cur\_1st\_h2 = 1$
$Pre\_4th\_h1 = 1;$ $Pre\_4th\_h2 = 0;$ $Pre\_4th\_h3 = 1;$ $Pre\_4th\_h4 = 1$
$Pre\_3rd\_h1 = 0;$ $Pre\_3rd\_h2 = 1;$ $Pre\_3rd\_h3 = 0;$ $Pre\_3rd\_h4 = 1$
$Pre\_2nd\_h1 = 1;$ $Pre\_2nd\_h2 = 0;$ $Pre\_2nd\_h3 = 1;$ $Pre\_2nd\_h4 = 1$
$Pre\_1st\_h1 = 1;$ $Pre\_1st\_h2 = 0;$ $Pre\_1st\_h3 = 1;$ $Pre\_1st\_h4 = 0$

and the input signal coming from the vertical edge memory are

$Cur\_2nd\_v1 = 1;$ $Cur\_2nd\_v2 = 0;$ $Cur\_1st\_v1 = 1;$ $Cur\_1st\_v2 = 0$
$Pre\_4th\_v1 = 1;$ $Pre\_4th\_v2 = 0;$ $Pre\_4th\_v3 = 1;$ $Pre\_4th\_v4 = 1$
$Pre\_3rd\_v1 = 0;$ $Pre\_3rd\_v2 = 1;$ $Pre\_3rd\_v3 = 0;$ $Pre\_3rd\_v4 = 1$
$Pre\_2nd\_v1 = 1;$ $Pre\_2nd\_v2 = 0;$ $Pre\_2nd\_v3 = 1;$ $Pre\_2nd\_v4 = 1$
$Pre\_1st\_v1 = 1;$ $Pre\_1st\_v2 = 0;$ $Pre\_1st\_v3 = 1;$ $Pre\_1st\_v4 = 0$

We can obtain a block matching results of
{ 5, 6, 2 }
{ 3, 2, 6 }
{ 4, 4, 2 }
then we can get the output of the motion vector Y4 or No. 0, which is located on the center of a 3 × 3 pixel block in the current frame, as shown in Fig. 7.27. In this case, the test result reveals that a 2 × 2 pixel block in the current frame has no relative motion with the corresponding 2 × 2 pixel block in the previous frame.

We can also see that the experiment results are identical to the theoretical analysis by using both of random input edge images in the current frame and random input edge images in the previous frame.

All the test results tell that the designed chip can estimate correct and stable motion vectors in a very fast speed.

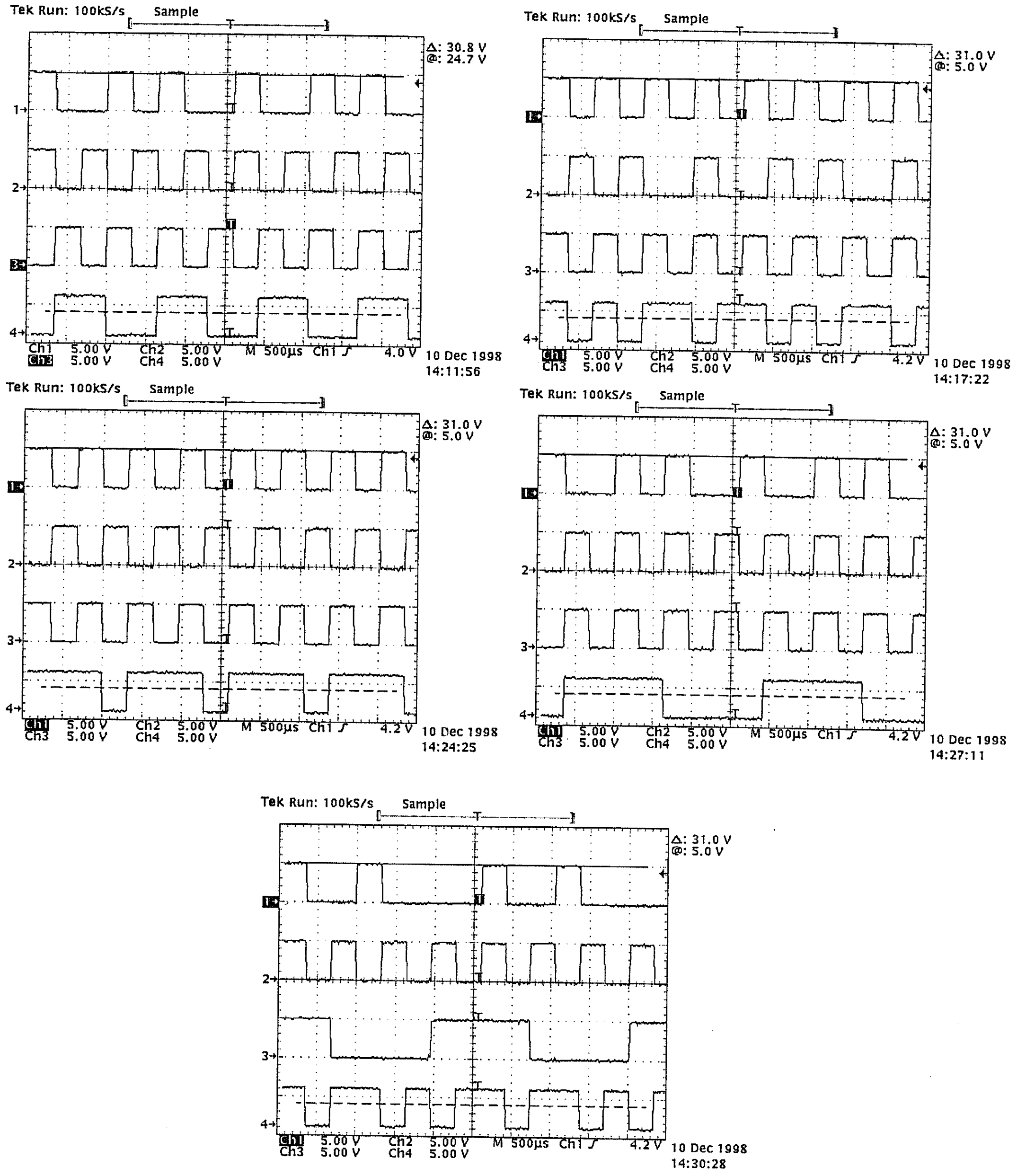
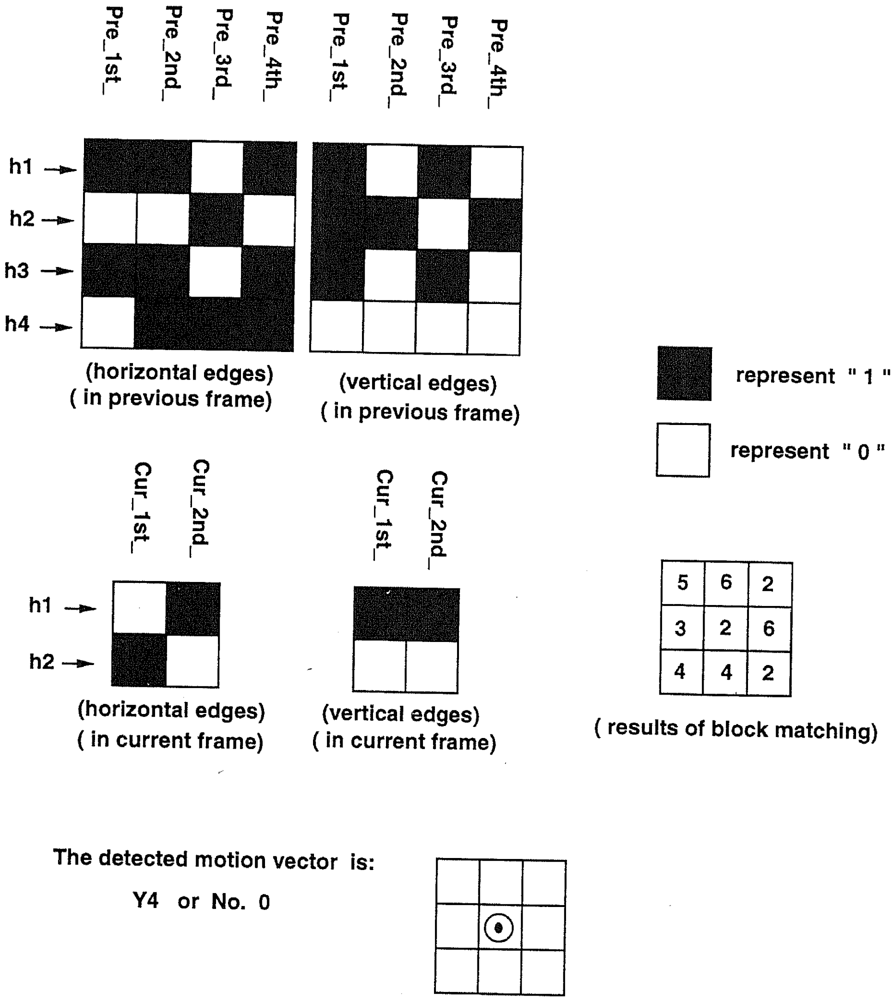
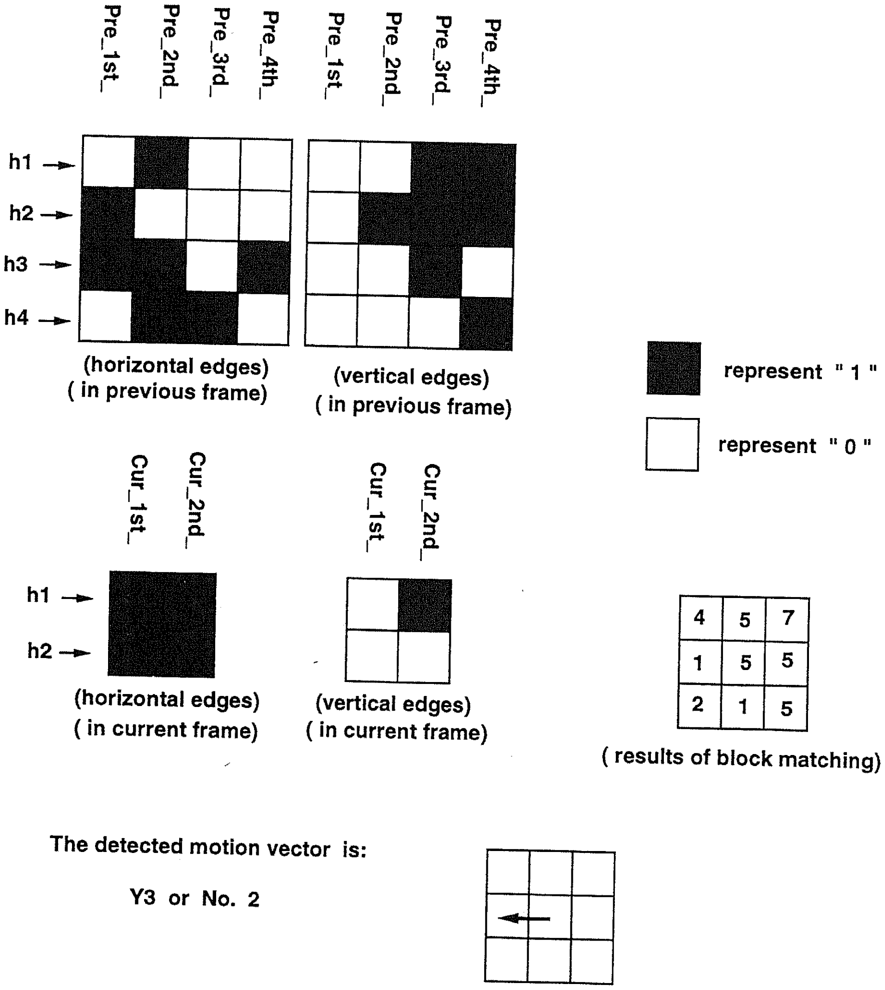


Figure 7.26: Random Input of Edge Images in the Previous Frames (the first 4 pictures): $\{Pre-4th-h1, Pre-3rd-h1, Pre-2nd-h1, Pre-1st-h1\}$ in the 1st picture, $\{Pre-4th-h2, Pre-3rd-h2, Pre-2nd-h2, Pre-1st-h2\}$ in the 2nd picture, $\{Pre-4th-h3, Pre-3rd-h3, Pre-2nd-h3, Pre-1st-h3\}$ in the 3rd picture, $\{Pre-4th-h4, Pre-3rd-h4, Pre-2nd-h4, Pre-1st-h4\}$ in the 4th picture, and Random Input of Edge Images in the Current Frame(the last picture): $\{Cur-2nd-h1, Cur-1st-h1, Cur-2nd-h2, Cur-1st-h2\}$ in the 5th picture.



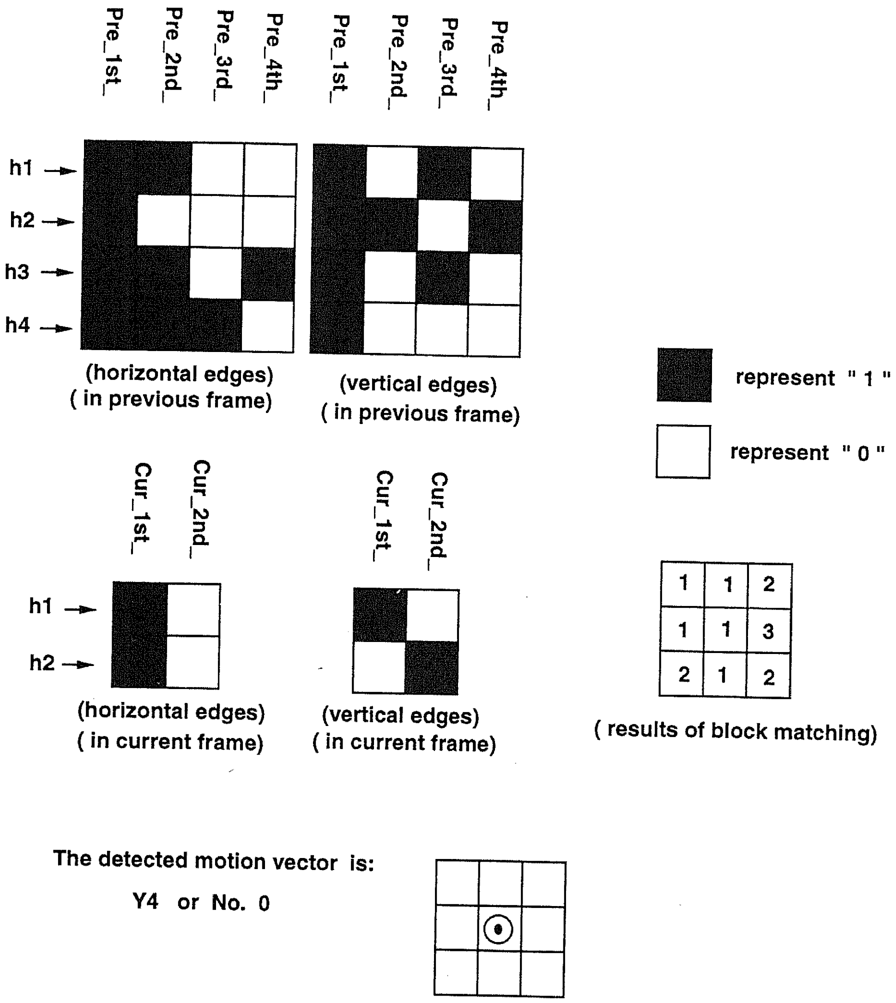
5	6	2
3	2	6
4	4	2

Figure 7.27: Example 1. The analysis for the motion vector detection by block matching of the random input of edge images in the previous frames and the random input of edge images in the current frame. The inputs are corresponding to the 1st phase of Fig. 7.26. The test result is shown in the 1st phase of Fig. 7.31.



4	5	7
1	5	5
2	1	5

Figure 7.28: Example 2. The analysis for the motion vector detection by block matching of the random input of edge images in the previous frames and the random input of edge images in the current frame. The inputs are corresponding to the 2nd phase of Fig. 7.26. The test result is shown in the 2nd phase of Fig. 7.31.



1	1	2
1	1	3
2	1	2

Figure 7.29: Example 3. The analysis for the motion vector detection by block matching of the random input of edge images in the previous frames and the random input of edge images in the current frame. The inputs are corresponding to the 3rd phase of Fig. 7.26. The test result is shown in the 3rd phase of Fig. 7.31.

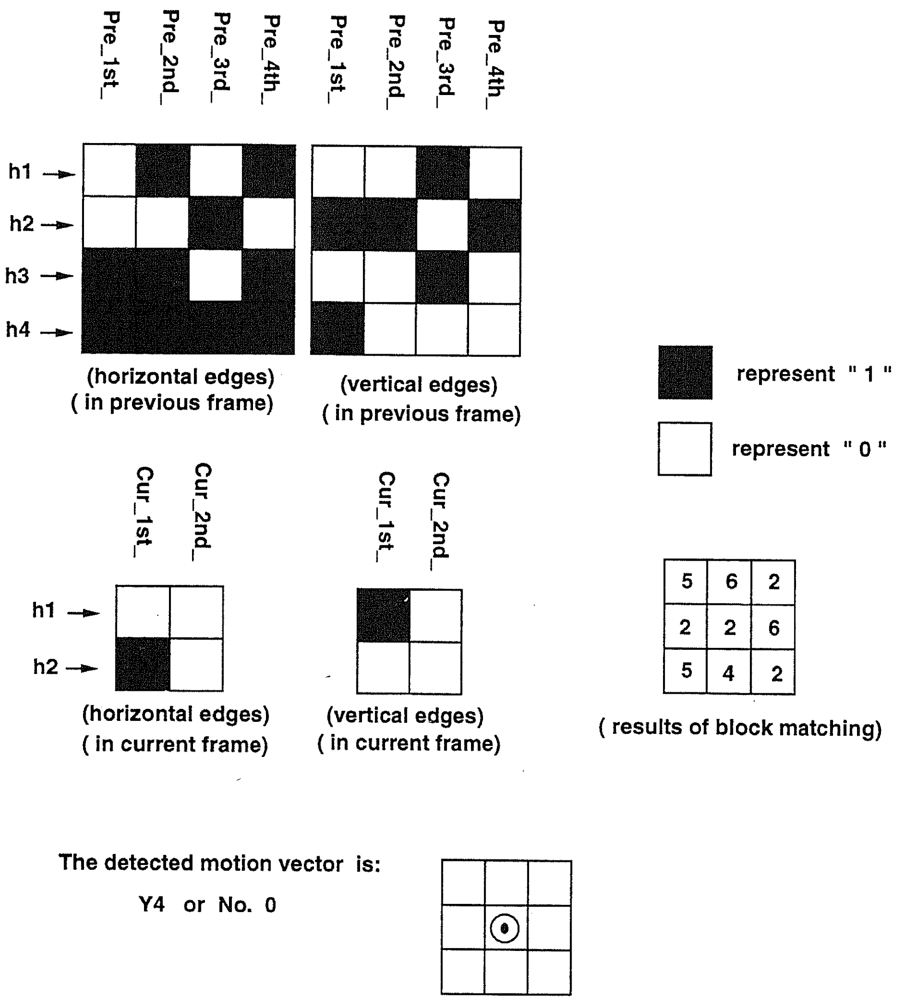


Figure 7.30: Example 4. The analysis for the motion vector detection by block matching of the random input of edge images in the previous frames and the random input of edge images in the current frame. The inputs are corresponding to the 4th phase of Fig. 7.26. The test result is shown in the 4th phase of Fig. 7.31.

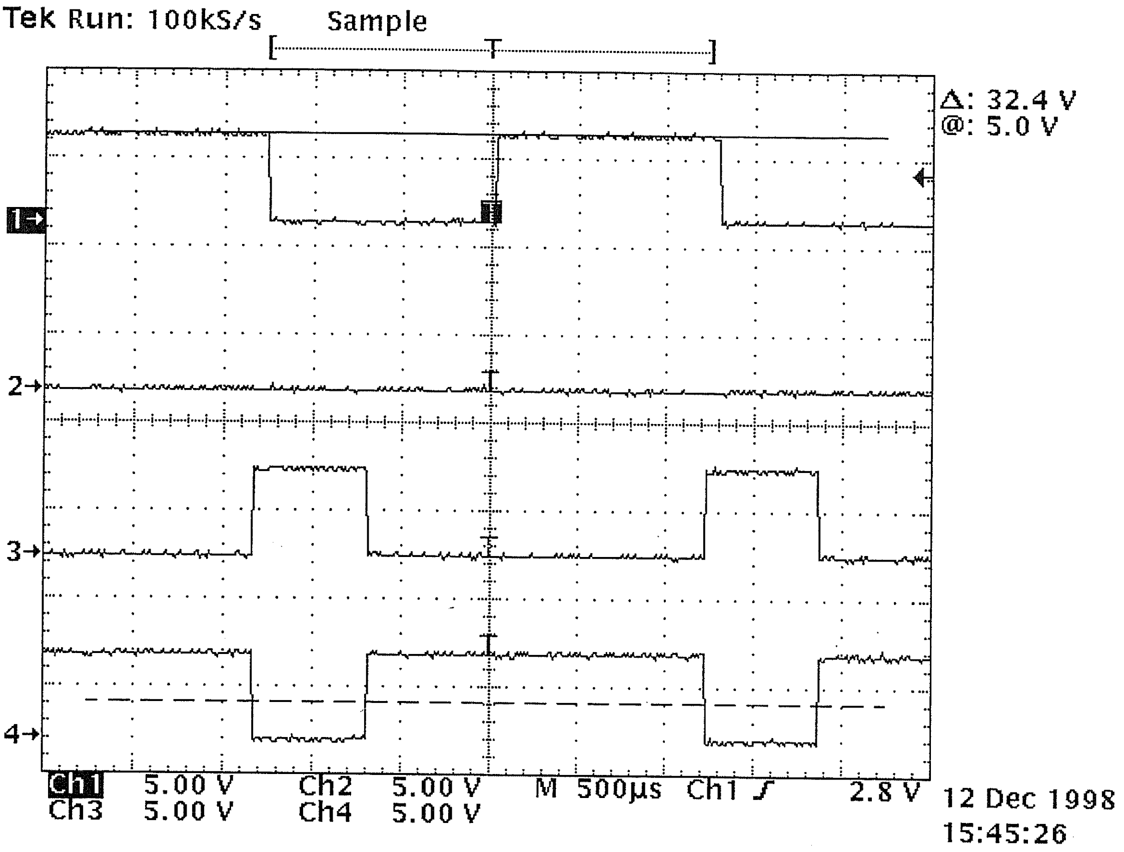


Figure 7.31: Evaluation results for motion vector detection. Channel 1 is Current-2nd-h1; Channel 2 is Y0(as well as $\{Y1, Y2, Y5, Y6, Y7, Y8\}$; Channel 3 is Y3; Channel 4 is Y4. The output waveforms of the motion vectors detected by the chip are identical to the analysis results shown in Fig. 7.27 Fig. 7.28 Fig. 7.29 and Fig. 7.30.

Chapter 8

CONCLUSION

Under a high frame rate of imaging, such as 100 $\sim$ 1000 frame/sec., the changes between two succesive frames are very small. By making use of this feature, vision chips for imaging, edge detection and very fast estimation of motion vectors are developed in this thesis.

By ES2 1-poly 2-metals $0.7\mu m$ CMOS process, we designed 1 prototype chip and 2 testing teg chips. Based on these chips, we successfully evaluated:

- pixel circuit with large dynamic output range
- analog processor for edge detection with large stable working range
- short-time Transmision Gate memory
- shift register with even-phase output and odd-phase output
- 16 × 16 imaging array
- specific accumulator
- specific LPGCP processor for block matching and motion vector detection
- small size 2D motion vector estimation

The prototype chip consists of a pixel array and parallel block matching processors. Each pixel of the pixel array contains a photo detector, an edge detector and a 4-bits memory parallel in local area.

The captured image is binarized by the edge detector and the binary edge data is used in a 2×2 small size block matching within a $(\pm 1, \pm 1)$ searching area. The block matching and motion vector detection are performed in globle column parallel architecture by specific LPGCP procesors designed on the vision chip.

The proposed method intends to detect a motion vector for every pixel and results in a motion field of vectors. In this way, we can obtain a dense motion field in which a motion vector is assigned to each pixel by overlopping 2×2 target blocks.

Further studies in this project are expected to bring in some applicable rusults not only for high speed image coding[32] but also for high speed machine vision in an one-chip imaging system.

Suggestions to a new version chip:

1. A suggestion is given to the further step of this project. When designing a new version of the 2D motion vector detection vision chip with more pixels, such as 32 × 32 pixels, 64 × 64 pixels, or 128 × 128 pixels, the operational logic of LPGCP processor should be designed by Layout Synthesizer or VHDL, which will greatly helpful to reduce the huge amount of manual layout time.

2. New circuit structure is required. Such as to take apart the edge detection and memory from the pixle circuit.

3. High fill factor is required for obtaining high Signal to Noise ratio. One way to get a high fill factor is based on a new circuit structure, the other way to get a high fill factor relys on the development of deep sub-micro CMOS process.

Bibliography

○ 参考文献:

[1] B. Liu, A. Zaccarin, "New Fast Algorithms for the Estimation of Block motion Vectors", IEEE Trans. on Circuits and Systems for Video Technology, Vol. 3, No.2, pp148-157, April 1993

[2] M. Ghanbari, " The cross-search algorithm for motion estimation", IEEE Trans. Commun., Vol. 38, No. 9, pp.950-953, 1990

[3] M. Sung, " Algorithms and VLSI architectures for motion estimation", VLSI Implementations for Image Communications, Edited by P. Pirsh, 1993.

[4] Takao ONOYE, Gen FUJIJTA, Masamichi TAKATSU, Isao SHIRAKAWA, Nariyoshi YAMAI, "Single Chip Implementation of Motion Estimator Dedicated to MPEG2 MP@HL", IEICE Trans. Fundamentals. Vol. E79-A No.8 pp1210, 1996.

[5] 江藤剛治, "4500枚/秒の高速度ビデオカメラ", テレビ誌, Vol.46, No.5, pp.543–549, 1992.

[6] T. Poggio, V. Torre C. Koch, "Computational vision and regularization theory," Nature, Vol. 317, No. 6035, pp. 314-319, 1985.

[7] C.Koch, "Implementing early vision algorithms in analog hardware", *SPIE*, Vol.1473, pp. 2–16, 1991.

[8] C.Koch and H.Li, "Vision Chips: Implementing Vision Algorithms with Analog VLSI", IEEE Computer Society Press, 1995.

[9] C.Mead: "Analog VLSI and Neural System", *Addison-Wesley*,1993

[10] Alireza Moini, "Vision Chips or Seeing Silicon", http://www.eleceng.adelaide.edu.au/Groups/GAAS/Bugeye/visionchips/index.html,

[11] H.Kobayashi, L.White and A.A.Abidi, "An Active Resistor Network for Gaussian Filtering of Images", *IEEE J.Solid State Cirsuits*, Vol.26, No.5, pp.738–748, 1991.

[12] L.T.Bruton. " RC active Circuits Theory and Design", by Prentice-Hall, Inc. 1980

[13] Y. Li, F. Lin, Y. Hu, and Y. Lou, " A novel investigation on NIC", IEEE Proceedings of ISCAS, pp.642-645, May, 1984

[14] J. Tanner and C. Mead, "A Correlating Optical Motion Detector", in MIT Advanced Research in VLSI, pp.57–64, 1984.

[15] T. Horiuchi and J. Lazzaro and A. Moore and C. Koch, "A correlation-based motion detection chip", *Advances in Neural Information Processing 3*, 1991.

[16] J. Tanner C. Mead, "An integrated analog optical motion sensor", In R.W. Brodersen H.S. Moscovitz, editor, VLSI Signal Processing II, pp. 59-87. IEEE, New York, 1988.

[17] Lisa Dron: "Computing 3D motion in custom analog and digital VLSI", *PhD thesis*, Dept. Elec. Eng. Comp. Sci., Massachuesetts Institute of Technology, Jan. 1991

[18] K. Laker, W. Sansen, "Design of Analog Integrated Circuits and Systems", McGraw-Hill, Inc, 1994

[19] F. Ferrari, J. Nielsen, P. Questa G. Sandini, "Space variant imaging," Sensor Review, Vol. 15, No. 2, pp. 17-20, 1995.

[20] 石嵜 透, "イメージセンサと統合した高速動きベクトル検出'、東京大学大学院、電子情報工学専攻、修士論文、1996年2月

[21] Lisa Dron McIlrath, "A CCD/CMOS Focal-Plane Array Edge Detection Processor Implementing the Multiscale Veto Algorithm", IEEE J. Solid State Circuits, Vol. 31. No. 9, Sept. 1996

[22] Marr, D. and Hilderth, E. " Theory of Edge Detection", Proc. R. Lond., Vol. B207, pp.187-217, 1980.

[23] PAUL I. SUCIU, DAVAD A. HODGES, "Image Contour Extraction with Analog MOS Circuit Techniques", IEEE Journal of Solid-State Circuits, Vol. SC-12, No. 1, Feb. 1977

[24] W.Bair and C.Koch,"Real-time Motion Detection Using an Analog VLSI Zerocrossing Chip", *Proc. SPIE*, Vol.1473, pp.59–65, 1991.

[25] R. Wodnicki and G.W. Roberts and M.D. Levine, "A Foveated Image Sensor in Standard CMOS Technology", *Custom Integrated Circuits Conf.*, 1995.

[26] S.Mallat and W. L. Hwang: "Singularity detection and processing with wavelets", IEEE Trans. on Information Theory, Vol. 38, No. 2, pp. 617-643, March 1992

[27] 沢田 和明 久保田 節ら, " 情報センシング", 映像情報 Media 学会誌, 1998 年 Vol.52, No.8, pp1066-1072.

[28] 川人 詳二 ら" 低消費電力動きベクトル検出"、1997 年重点領域研究、"Ultimate Integration of Intelligence on Silicon Electronic Systems" 報告論文集 pp78-84.

[29] Dwi Handoko ら,"Frame Interpolation method for Low power motion vector detection", 電子情報通信学会, 信学技報 (ICD98-113),pp.15-20, 1998 年 8 月

○ 大学院論文輪講資料:

[30] 李 正, "Computational Circuits in the Vision Chips", 大学院論文輪講, 1996 年 11 月 22 日

[31] 李 正, "動きベクトル推定: アルゴリズム からチップ まで ", 大学院論文輪講, 1997 年 10 月 17 日

[32] 李 正, "Video Transmision over Wireless Channels ", 大学院論文輪講, 1998 年 6 月 26 日

Publications

○ 学会誌:

(1) 李 正, 石嵜 透, 相澤 清晴, 羽鳥 光俊, "センサ面上での二次元動きベクトル検出の提案と設計", 映像情報メディア学会誌,1998年12月 Vol.52, No.12, pp. 1880-1883
"2D Motion Vector Detection on Image Sensor Plane", The Journal of the Institute of Image Information and Television Engineers, Vol.52, No.12, pp. 1880-1883, 1998

(2) Zheng Li, Kiyoharu Aizawa, Mitsutoshi Hatori, "Vision Chip for Very Fast Detection of Motion Vectors: Design and Implementation", IEICE Transactions on Electronics: Special issue on "Integrated Electronics and New System Paradigms", to be published in September 1999. (受付)

○ 国際会議:

(3) Zheng Li, Kiyoharu Aizawa, Mitsutoshi Hatori, "Motion Vector Estimation on Focal Sensor Plane by Block Matching", 2nd EUROPTO International Conference on Advanced Focal Plane Arrays and Electronic Cameras (AFPAEC'98), Proceedings of SPIE Reprint Vol. 3410, pp.215-221, Zurich, Switzerland, May 18-19, 1998

(4) Zheng Li, Kiyoharu Aizawa, Mitsutoshi Hatori, "Design and implementation of a smart image sensor with 2D motion vector estimation", IS&T・SPIE's 11th Annual Symposium on Electronic Imaging'99 Science and Technology, Vol. 3649-23, San Jose, California USA, January 23-28, 1999

(5) Zheng Li, Kiyoharu Aizawa, Mitsutoshi Hatori,"Implementation of a 2D motion vector detection on image sensor focal plane", 1999 IEEE International Symposium on Circuits and Systems, Orlando, Florida, USA, May 30-June 2, 1999 (Accepted)

(6) Zheng Li, Kiyoharu Aizawa, Mitsutoshi Hatori," Focal plane processing for a fast 2D motion vector detection ", 1999 IEEE Workshop on Charge-Coupled Devices and Advanced Image Sensors, at Karuizawa Prince Hotel, Japan, June 10-12, 1999. (submitted)

(7) Zheng Li, Kiyoharu Aizawa,“A Specific Processor for Motion Vector Estimation Integrated on Focal Plane”, International Symposium on Future of Intellectual Integrated Electronics, ISFIIE’99, March, 1999, Sendai, Japan (Accepted)

○ 研究会:

(8) 石嵜 透, 李 正, 相澤 清晴, 羽鳥 光俊: “センサ上での即時動きベクトル検出 - 方式と設計 -”, 電子情報通信学会技術報告, EID96-47, pp.37–42 (1996-10)
TECHNICAL REPORT OF IEICE, EID96-47, pp.37–42, Oct. 1996

(9) 李 正, 相澤 清晴, 羽鳥 光俊,“イメージセンサ面上での高速ブロックマッチングによる動き検出”, 映像情報メディア学技報, Vol. 22, No. 13, pp.7-12, 1998 年 2 月
ITE TECHNICAL REPORT Vol.22, No.13, pp.7-12, Feb. 1998

(10) 李 正, 相澤 清晴, 羽鳥 光俊, ”撮像面上での動きベクトル検出 ”, 信学技報 (集積回路研究会),ICD98-113, pp1-6, 1998 年 8 月
TECHNICAL REPORT OF IEICE, ICD98-113, pp.1-6, Aug. 1998

○ シンポジウム と ワークショップ:

(11) 李 正, 相澤 清晴, 羽鳥 光俊, ”イメージセンサ上でのウェーブレット変換 – A wavelet transform on image sensor ”, 1996 年 10 月, 映像メディア処理 シンポジウム (IMPS96), pp41-42, 横浜

(12) 李 正, 相澤 清晴, 羽鳥 光俊, “CMOS 焦点面上での二次元動きベクトル検出の設計”, 映像メディア処理 シンポジウム (IMPS97), pp27-28, 1997 年 10 月, 長野県軽井沢

(13) 李 正, 相澤 清晴, 羽鳥 光俊,“イメージセンサ面上での動き推定 ASIC ”, 第 4 回　画像センシングシンポジウム (SII’98), pp315-320, 1998 年 6 月、横浜

(14) 李 正, 相澤 清晴, 羽鳥 光俊,“シフティングウィンドウによる撮像面上で動きベクトル検出の設計と試作”, 映像メディア処理 シンポジウム (IMPS98), pp.59-60, 1998 年 10 月 26 日, 長野県軽井沢

(15) 李 正, 相澤 清晴, 羽鳥 光俊, ” 高速フレームレートでの動きベクトル検出処理を統合したイメージセンサの試作と評価”, 第 2 回システムＬＳＩワークショップ, pp.281-283, 1998 年 11 月 26 日, 守山市琵琶湖

(16) 李 正, 相澤 清晴, 羽鳥 光俊,“即時動きベクトル検出を行なうビジョンチップの設計と試作”,
Design and Implementation of a Vision Chip for Fast Detection of Motion Vectors

1999年ロボティクス・メカトロニクス講演会: 知能ロボットシステム用高性能プロセッサ, pp. , 1999年6月11日-13日(発表予定), 東京工業大学大岡山キャンパス

(17) 李 正, 相澤 清晴, 羽鳥 光俊,“CMOSセンサ面上へ統合した動きベクトル検出処理回路”, 第38回計測自動制御学会学術講演会(SICE’99) , 1999年7月28日-30日, 岩手県盛岡市(投稿中)

○ 全国大会:

(18) 李 正, 石嵜 透, 相澤 清晴, 羽鳥 光俊, “2次元動きベクトル検出センサ”,1997年, 映像メディア学会年次大会、(ITE’97: 1997 ITE Annual Convention), pp33-34, 東京

(19) 李 正, 相澤 清晴, 羽鳥 光俊, “CMOS焦点面上での処理のための安定エッジ検出器の比較検討”、 電子情報通信学会エレクトロニクスソサイエティ大会, pp.138, 1997、東京

(20) 石嵜 透, 李 正、相澤 清晴、羽鳥 光俊 “瞬時動きベクトル検出センサの設計・試作”、電子情報通信学会総合大会, C12-58, pp.194, 1997年3月, 関西、吹田市

(21) 李 正, 相澤 清晴, 羽鳥 光俊,“イメージセンサ面上での高速ブロックマッチング”, 電子情報通信学会総合大会・情報システム、 pp.184, 平塚市、 1998年3月

(22) 李 正, 相澤 清晴, 羽鳥 光俊, “ローカル並列グローバル列並列構成による動きベクトル検出イメージセンサ ”,1998年, 映像メディア学会年次大会、(ITE’98: ITE Annual Convention), pp.146-147, 1998年7月, 千里、新大阪

(23) 李 正, 相澤 清晴, 羽鳥 光俊,“シフティングウィンドウによる高速動きベクトル検出センサの設計 ”, 1998年電子情報通信学会ソサイエティ大会, pp.186, 1998年10月1日, 山梨県甲府市

(24) 李 正, 相澤 清晴, 羽鳥 光俊,“CMOSセンサ面上へ統合した動きベクトル検出処理回路 ”, 1999年電子情報通信学会総合大会(発表予定), pp. , 1999年3月25-28日

Appendix A

Speed limitation of the image sensor

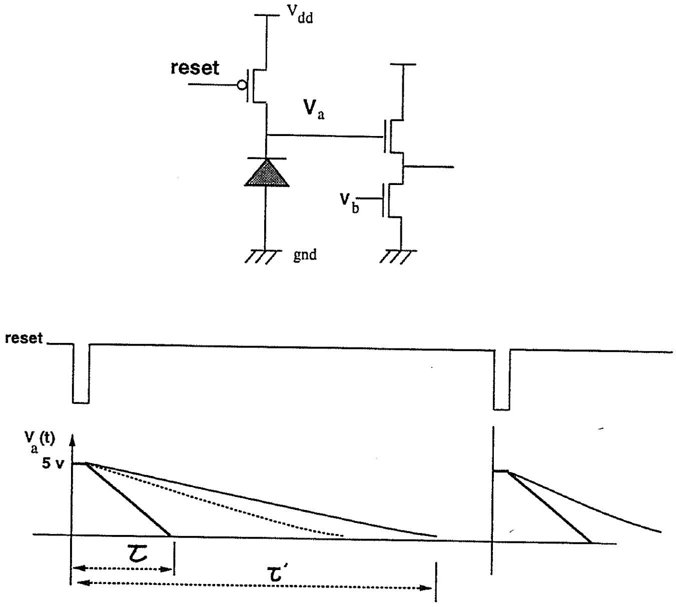


Figure A.1: Integral time limitation of the sensor.

As shown in Fig. A.1, the image sensor has a sampling limitation because of the charging and discharging ability of the photodiode. In some range, the time constant τ is approximately inpropotional to the strength of light, i.e. the stronger the light the smaller the τ. So that if we want a much higher speed of imaging, we should need a stronger light illumination. If we only need a low speed of imaging, we can use a week light illumination. But it is difficult to give out the concrete numeral figures for the highest speed of the CMOS imagers because of the complex time-variant character.

Appendix B

Wavelet transform on focal plane:

Wavelet transform is a good tool for space analysis. It can be designed on the focal plane according to the following analysis, but simlifications are neccessary.

General wavelet transform:

As we know that the Mallat's Fast Algorithm can be written as:

$$\begin{aligned}
F^{l+1}_{n,m} &= \frac{1}{2}\sum_{i,j\in z} F^{l}_{i,j}h_{i-2n}h_{j-2m} && (B-1)\\
D1^{l+1}_{n,m} &= \frac{1}{2}\sum_{i,j\in z} F^{l}_{i,j}h_{i-2n}g_{j-2m} && (B-2)\\
D2^{l+1}_{n,m} &= \frac{1}{2}\sum_{i,j\in z} F^{l}_{i,j}g_{i-2n}h_{j-2m} && (B-3)\\
D3^{l+1}_{n,m} &= \frac{1}{2}\sum_{i,j\in z} F^{l}_{i,j}g_{i-2n}g_{j-2m} && (B-4)
\end{aligned}$$

It consists of the general Wavelet analysis for images. This complex equations are difficult to be designed on the focal plane.

Modified Haar wavelet transform(MH-WLT)

Since the Haar wavelet coefficients(QMF) can be represented as

$h_k = \{\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}, 0, 0, ...\}$,

$g_k = \{-\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}, 0, 0, ...\}$

then the Haar wavelet transform can be writen as:

$$\begin{aligned}
F^{l+1}_{n,m} &= \frac{1}{4}(F^{l}_{2n,2m} + F^{l}_{2n+1,2m} + F^{l}_{2n,2m+1} + F^{l}_{2n+1,2m+1}) && (B-5)\\
D1^{l+1}_{n,m} &= \frac{1}{4}(-F^{l}_{2n,2m} - F^{l}_{2n+1,2m} + F^{l}_{2n,2m+1} + F^{l}_{2n+1,2m+1}) && (B-6)\\
D2^{l+1}_{n,m} &= \frac{1}{4}(-F^{l}_{2n,2m} + F^{l}_{2n+1,2m} - F^{l}_{2n,2m+1} + F^{l}_{2n+1,2m+1}) && (B-7)\\
D3^{l+1}_{n,m} &= \frac{1}{4}(F^{l}_{2n,2m} - F^{l}_{2n+1,2m} - F^{l}_{2n,2m+1} + F^{l}_{2n+1,2m+1}) && (B-8)
\end{aligned}$$

By a proper coordinates shifting, we can get a modified Haar wavwlet transform (MH-WLT)$^{(11)}$ as the form of:

$$\begin{aligned}
F^{l+1}_{n,m} &= F^{l}_{2n,2m} && (B-9)\\
D1^{l+1}_{n,m} &= \frac{1}{2}(F^{l}_{2n,2m+1} - F^{l}_{2n,2m}) && (B-10)\\
D2^{l+1}_{n,m} &= \frac{1}{2}(F^{l}_{2n+1,2m} - F^{l}_{2n,2m}) && (B-11)\\
D3^{l+1}_{n,m} &= \frac{1}{4}(F^{l}_{2n,2m} - F^{l}_{2n+1,2m} - F^{l}_{2n,2m+1} + F^{l}_{2n+1,2m+1}) && (B-12)
\end{aligned}$$

The last second and third equation above is just the edge detection algoritm used for horizontal and vertical edge detection.

Acknoledgement

Thanks to all of those who helped me a lot during my Ph.D. program.

相澤清晴助教授には、博士課程の3年間にわたって日頃より懇切丁寧な御指導を賜わりまし、深く感謝致します。高機能ビジョンチップの研究の機会を上えて下さり、様々な御助言を頂きました。

羽鳥光俊教授には、大変ご多忙のなか研究全般にわたって御指導いただきました。深く感謝致します。

研究室において研究環境の整備に尽力してくださった塚本憲男技術官、田中崇助手にお礼申し上げます。秘書の和田秀子さん、斉藤淳子には、いろいろとお世話になりました。ありがとうございました。

本研究に関して、研究室においては博士課程の浜本隆之さん、大塚康弘さん、修士課程の石嵜透さん、久保田彰さん、杉浦和英 さん、卒論の篠政義さん から御討論、御助言を頂きました。ありがとうございました。

最後に、この3年間、共に学生生活を送った，同じ博士課程の児玉 和也さんに感謝したいと思います。研究室においていろいろとお世話になった國頭吾郎さん、崔潤基 さん、Ajith PASQUAL さん、 Vincent van de Laar さん、 石川浩二さん、 吉川典史さん、神英樹 さん、 SUPATANA さん、 Michael TAN さん、研究員の 川崎淳司さん、永田晃司さん、中村大二さん、大学院生、卒論生、 研究生の皆様に感謝致します。

研究室外においては、 VDEC の浅田邦博教授、柴田 直教授、 CMP の Jean-Francois Paillotin さんに感謝致します。

どうもありがとうございました。

1998年12月18日

李正