國 立 交 通 大 學

電機與控制工程學系

碩士論文

使用細胞神經網路架構實現影像穩定
處理之震動向量估測晶片

Chip Design of CNN-Based Local Motion Estimation for Image Stabilization Processing

研究生：程盈彰

指導教授：林進燈　　博士

中華民國 九十四 年 七 月

使用細胞神經網路架構實現影像穩定處理之震動向量估測晶片

Chip Design of CNN-Based Local Motion Estimation for Image Stabilization Processing

研 究 生：程盈彰　　Student：Ying-chang Cheng

指導教授：林進燈 博士　　Advisor：Dr. Chin-Teng Lin

國立交通大學

電機與控制工程學系

碩士論文

A Thesis

Submitted to Department of Electrical and Control Engineering

College of Engineering and Computer Science

National Chiao Tung University

in Partial Fulfillment of the Requirements

for the Degree of Master

in

Electrical and Control Engineering

June 2005

Hsinchu, Taiwan, Republic of China

中華民國 九十四年 七 月

使用細胞神經網路架構實現影像穩定處理之震動向量估測晶片

學生：程盈彰　　指導教授：林進燈 博士

國立交通大學電機與控制工程研究所

中文摘要

利用硬體估測攝影裝置於拍攝時，因手晃動或是機體支架震動所產生之不預期背景震動向量，以提供影像穩定系統(Image Stabilization)補償為本論文的主要貢獻。影像穩定系統包含背景震動向量估測單元及震動向量之補償單元兩部分，然而計算背景震動向量佔去整個影像穩定系統大部分的處理時間，因此本論文提出一特殊應用導向之晶片(ASIC)，處理拍攝影像背景震動向量的估測。晶片設計採用仿細胞神經網路(Cellular Neural Network, CNN) 架構用以偵測背景震動向量的特徵；仿細胞神經網路之基本架構為一個和週圍細胞(cell)相連並規則排列之二維陣列，具有即時平行運算處理的能力。本論文提出使用 CNN 可調節偏壓的設計，快速計算影像穩定系統中所需要的背景震動向量，並配合全區域輸出連結鍊(global output connect chains)之設計，將區域背景震動向量位置定址出來。此外，區域類比記憶體(Local Analog Memory)陣列的設計來儲存影像差值資訊。本論文所設計之背景震動向量估測晶片採用 TSMC 0.35um 混合訊號製程，其大小為 8.1mm^2 共包含19×25個像素處理單元。電路模擬及 CNNUM 分析結果證實採用以 CNN 實現 IS 系統中估測背景震動向量具有遠高於 DSP 處理器的運算速度，讓整體 IS 系統具有即時運算之能力。

Chip Design of CNN-Based Local Motion Estimation for Image Stabilization Processing

Student：Ying-chang, Cheng Advisor：Dr. Chin-Teng Lin

Department of Electrical and Control Engineering

National Chiao Tung University

Abstract

The objective of this thesis is to investigate the hardware design in image stabilization (IS) technique for local motion vectors (LMVs) in the image sequences. The IS technique is used to remove unwanted shaking phenomena in the image sequences captured by hand-held camcorders without affecting moving objects in image sequences and the intentional motion of panning condition, etc. It consists of motion estimation and motion compensation. Most of the complex and time consuming computations occur in motion estimation, an application-specific IC is designed to solve this problem. Cellular Neural Network (CNN) technology is used to implement the local motion estimation chip. CNN is a regular two-dimensional array and connects with its neighborhood locally. Real-time and parallel analog computing elements are contained in the architecture. CNN adaptive threshold template is proposed to extract reliable motion vectors from a given region. The design of global output connected chains can easily decode the LMV address. The local analog memory (LAM) is designed to store image difference information. The size of CNN array is 19×25 pixels. The chip has integrated in the total area of $8.1mm^2$ by using TSMC 0.35um mixed-signal process. Results with HSPICE simulation and CNNUM analysis prove that the performance of the proposed CNN-based local motion estimation is better than that of a digital signal processor so that the IS system has the capability of real-time operations.

誌謝

在電控所兩年的求學生涯中，首先要感謝指導教授林進燈老師這兩年來不僅在學業方面的悉心指導，讓我學習到許多寶貴的知識，在學業及研究方法上也受益良多；更在爲人處世及求學態度上給予啟蒙及悉心指導，使得本論文能順利完成。另外也要感謝口試委員們的的建議與指教，使得本論文更為完整。

其次，感謝曾提供我相當多協助的學長鶴章、仁峰、朝暉、世安、立偉、匈牙利來的 Guiri、資訊媒體實驗室的同學宗恆、俊永、弘昕、Linda、Martin、育緯、晴慧，資訊媒體實驗室的學弟家昇、經翔、谷谷、Jack，618 的重甫、307 的文芩，以及實驗高中好友們孫民、吉哥、鈞哥、剛哥、以信、小 P 在研究過程中所給我的鼓勵與協助；其實該感謝的人很多，無法一一道盡，只有將感激收藏在心裡。

在交大的這兩年中曾與我在校園中度過的師友們，因有你們的參與使得我在研究所的求學階段更加多采多姿，留下許多美好的回憶。

最後，更要感謝我的父母親對我的教育與栽培，給予我一切鼓勵與協助，使我能安心無負擔的完成碩士學業。僅將本論文獻給我的家人所有關心我的師長與朋友們，願他們共享這份榮耀。

Contents

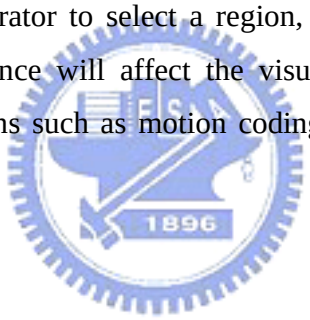
List of Tables

List of Figures

CHAPTER 1

INTRODUCTION

Video image stabilizers that compensate for camcorder shaking have already become indispensable for consumer video camcorders [1]. Image stabilization (IS) is also known as vibration reduction, which is a digital camcorder technology that helps preventing images from blurring. It reduces vibration caused by camcorder shake, slow shutter speed or when using a long telephoto lens without a tripod. For developed video camcorders, image stabilization is finding a way into more consumer and professional digital camcorders. Various image-stabilizing systems have been developed for camcorders to free from degradation in picture quality by hand-movement. In the IS processing, scene points are motionless in spite of camcorder motion. This makes it easier for an operator to select a region, for example. The unwanted positional fluctuations of the video sequence will affect the visual quality and impede the subsequent processes for various applications such as motion coding, video compression, feature tracking, etc.



1.1 Motivation

Basically, the IS technique can be classified into two processing methods. One is the optical image stabilization (OIS), and the other is the electronic image stabilization (EIS). Optical image stabilization (see APPENDIX B) uses mechanical motion compensation to physically move the lens, and hence the image that falls on the image sensor, in the opposite direction from the camcorder shake. Camcorder makers offering optical stabilizers include Sony, Panasonic and Canon corp. [2], but this feature is generally reserved for high-end models. Optical image stabilization for consumer's video cameras has been proposed by Holder [4] and Oshima [5]. Both systems are similar in the sense that they produce angular velocity by using gyro control sensors, but they differ in methods for compensating the angular velocity [6]. One common disadvantage of Holder's method and Oshima's system is that they are using mechanical parts such as gyros and they control deflection coils (Holder) or a gimbal mechanism (Oshima) for

motion compensation. The mechanical parts of the IS system result in higher cost, larger space, and heavier.

Electronic image stabilizer, so-called digital image stabilizers (DIS), takes the property of image sensors with more pixels than the video image required and does the digital image processing. The video image is like a “window” that moves around within the larger frame of the image sensor. When camcorder shake moves the image up, EIS moves this “video window” down to compensate. Many DIS algorithms have been proposed. Chang et al. [7] use optical flow to remove the translational and rotational motion disturbance. The optical flow technique is used to estimate the local motion vector field of the image and yield the velocity of each pixel in the current image frame. Ko [8] propose a gray-coded bit-plan DIS algorithm to estimate the irregular condition motion vector due to moving objects and intentional panning. ITRI [9] has developed a DIS prototype system with FPGA and DSP implementation. The system composes of software and hardware blocks to utilize the gray-coded bit-plan matching algorithm for the video sequences. The DIS technique has been widely used for the computation of ego-motion [10], and video compression [11].

DIS consists of the motion estimating system and the motion compensation system. The motion estimation based on block matching algorithm (BMA) plays an important role in DIS [13]-[16]. The full-search (FS) BMA under the mean absolute difference (MAD) and the mean square error (MSE) criteria can be considered as an optimal solution for block motion estimation [8]. For the motion compensation, the accumulated motion vector estimation [18] and frame position smoothing (FPS) [28]-[31] are two of the most popular approaches. The accumulated motion vector estimation needs to compromise stabilization and intentional panning preservation since the panning condition causes a steady-state lag in the motion trajectory[28]-[31]. FPS accomplishes the smooth reconstruction of an actual long-term camera motion by filtering out jitter components based on the concept of designing the filter with appropriated cut-off frequency [28] or adaptive fuzzy filter to continuously improve stabilization performance [31].

The full-search block matching algorithm requires complicated computation which is time consuming, and hardware implementation. Several computationally efficient DIS algorithms, such as representative point matching (RPM) [17], edge pattern matching (EPM) [18], and bit-plane matching (BPM) [19] have devoted for portable camcorders [13], [14], [16]. The major

objective of these algorithms is to reduce the computational complexity, in comparison with full-search block-matching method, without losing too much accuracy and reliability. However, RPM still costs many computation cycles in comparing with the global minimum position, and EPM requires large amount of computations due to preprocessing for generating edge maps [8].

After analyzing the DIS algorithm, the motion estimation part has accounted for 80% of the computational complexity. The last 20% computation load belongs to motion compensation [32]. In contrast to the heavy arithmetic computational load from RPM (which is used in the proposed motion estimation), the motion compensation simply does some decisions on LMVs and then the calculation of the compensated motion vector. Hence we focus on a cost effective hardware design for the motion estimation.

1.2 Objective

Most of the DIS systems are processed by PC or FPGA implementation. The chip implantation for consumer camcorders of the IS system is only for gyro control sensor in optical stabilization system. We aim at designing a novel architecture for local motion estimation of the DIS system with VLSI approach. Due to the complex computation found in motion estimation, the DIS system is hard to perform in real time. In order to further reduce the computational complexity in finding LMVs, the mixed-signal cellular neural network (CNN) architecture is considered [20]. Comparing with conventional digital technology, cellular neural/nonlinear network (CNN)-based computing is capable of realizing the trillions of operations per second (TeraOPS)-range image processing tasks in a cost-effective implementation [21]. CNNs are an analog nonlinear dynamic processor arrays in which direct inter-connections among the basic processing units are restricted to a finite local neighborhood [22]. By changing the weight of local interconnections between neighborhood CNN cells, many image processing tasks can be realized with CNN framework. Because of their inherently parallel processing architecture, CNN can achieve a high speed computation while realizing the image processing tasks. In spite of that, their uniformity and local connectivity make them especially suited for VLSI implementation [23][24][25].

In this thesis, we propose an application-specific CNN (ASCNN) chip which could highly reduce the heavy computation problem for motion estimation. The RPM method is used to find the local motion vectors. The pre-processing image information will first pass a D/A converter and store in the memories. Then CNN will search the global minimum position and decode the x and y coordinates by using global output connected chains. Local analog memory (LAM) is designed to store the image difference information which is passed from an 8-bit D/A converter. With the aid of CNN technique [20], [27] , the global minimum position according to the RPM method could be easily generated and stored in comparing with the traditional DSP computation. The method of global output connected chains is used to connect CNN output and decode the LMV address. A reliable local motion vector extraction method based on CNN architecture is designed for the determination of global motion vector and image compensation processing in practical applications.

1.3 Thesis Organization

This thesis is organized as follows. Chapter 2 describes the models of the proposed image stabilization algorithm and gives a computational analysis to the DIS system. Chapter 3 introduces CNN algorithm, the hardware design of the CNN core and templates design for the CNN-based local motion estimator. Chapter 4 describes the circuit design of application specific CNN (ASCNN) chip. The simulation results with HSPICE, CNN universal machine (CNNUM), and ModelSim respectively are shown in Chapter 5. Conclusions and future works are made in Chapter 6.

CHAPTER 2

DIGITAL IMAGE STABILIZATION

The architecture of the proposed image stabilizer technique shown in Fig. 1 is divided into two processing blocks as motion estimation and motion compensation. The motion estimation block consists of three estimators: the local motion vectors (LMVs), the ill-conditioned motion vector (IMV), and the global motion vector (GMV) estimators. The motion compensation unit consists of the compensating motion vector (CMV) estimation and image compensation. The two incoming consecutive images (at time (*t-1*) and time (*t*)) will be firstly divided into four regions. A LMV will be derived in each region by the representative point matching (RPM) algorithm [12], [19]. The motion estimation block also contains a reliability detection function that will generate an ill-conditioned motion vector for the irregular image conditions such as the lack of features or containing large low-contrast area, etc. The GMV estimation determines a global motion vector among LMVs, the IMV, and other pre-selected motion vectors through background-based evaluation function. Finally, the compensating CMV is generated according to the resultant GMV and the image sequences will be compensated based on the CMV in the motion compensation unit.

The proposed digital image stabilizer system contains motion estimation step and motion compensation step. The design blocks are described as follows.

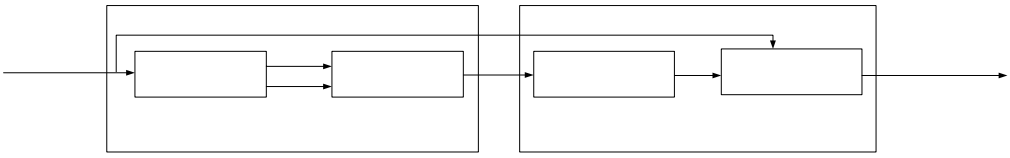


Fig. 1 The architecture of the proposed image stabilization technique.

2.1 Motion Estimation

The motion estimation unit shown in Fig. 1 contains the LMVs, IMV, and GMV estimators. As shown in Fig. 2, the LMVs and IMV estimation is to generate the LMVs and IMV for global

motion vector estimation. The LMVs can be obtained from the correlation between two consecutive images by the representative point matching (RPM) algorithm [12], [19]. The IMV can be obtained from LMVs by evaluating the corresponding confidence indices through the irregular condition detection and the proposed IMV generation algorithm.

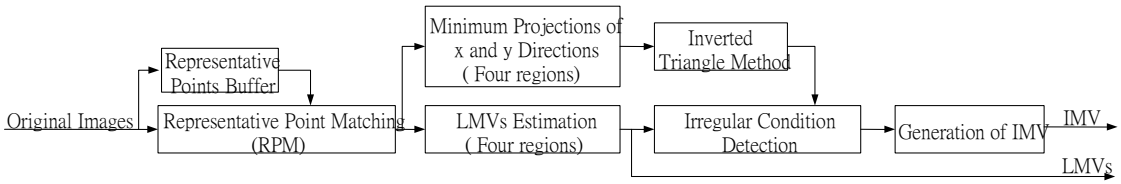


Fig. 2 The block diagram of LMVs and IMV estimation.

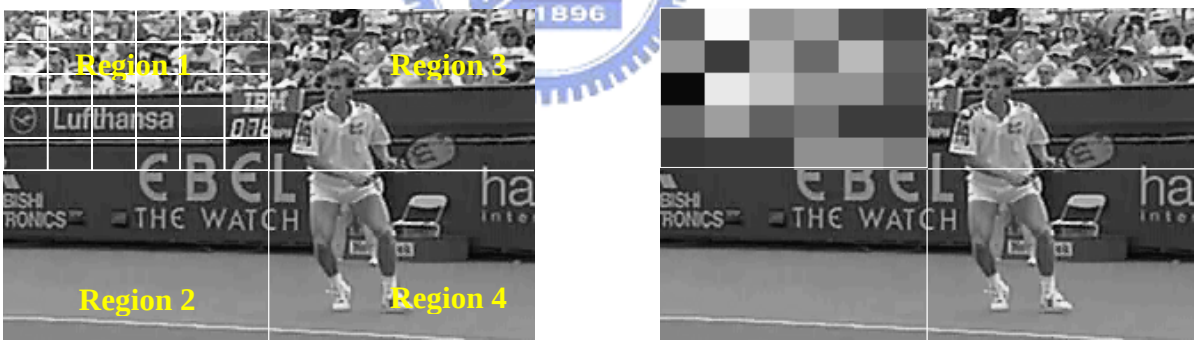
2.1.1 Local Motion Vector (LMV) and Irregular Condition Detection

First, we obtain location motion vectors by using the representative point matching method. A 19×25 pixels' macro-block is the basic processing unit of the algorithm. For a given sequence of video image, the specific boundary region of the incoming frame will be discarded first which saves as the compensating area. Then the pre-processing frame will be separated into four regions which will generate a local motion vector after the later steps. Then each region will further cut into many 19×25 pixels sub-regions. We take the center point color of the sub-region image as our representative point value. Each sub-region image has its own representative point value. The previous representing point value is subtracted from the present sub-region image and the absolute value is taken. The minimum value position in the 19×25 absolute differences matrix is considered as the previous representative moving point due to vibration noise. Summing all of the absolute differences (SAD) matrices and give a statistical analysis on them. The vector which calculates form the center point to the minimum SAD value position is considered as the local motion vector of each region.

Our testing image sequence is a tennis player video clip which is 312×200 pixels for each incoming frame. The steps and the results of the algorithm are listed below.

- Each incoming pre-processing image (300×190 pixels) is divided into four equal regions and each region is 150×95 pixels.

- Segment the prescribed sequence region (*t*-1) and (*t*) into sub-images which each of them is 19×25 pixels as shown in Fig. 3(a).
- Map all the pixels with the central point in each sub-image (*t*-1). The mapping array is called representative point macro-block (RPM) as shown in Fig. 3 (b).
- *Subtraction*: The operation is defined by |Msub(t)| := Sub-image(*t*)-Representative_point image (*t*-1) to provide absolute difference for the Msub(*t*) matrix (19×25 pixels).
- *Addition*: Add all the |Msub(t)| in the prescribed region to form an 19×25 difference value matrix as shown in Fig. 4. Fig. 4 is the SAD matrix that map into a 3D view. The z-coordinate is the absolute difference value. The lower of the SAD value, the closer the LMV is.
- *Minimum*: Find the minimum absolute difference position from the prescribed region and calculate the vector from the center point. The vector is called the local motion vector of the prescribed region. After analyzing the Fig. 4, we can find that the minimum position lies in the left and top place in the array and we will discuss it more in Chapter 5.



(a) (b)

Fig. 3 (a) The original image is divided into 4 regions, and then each region is cut into an array of 19×25 pixels again. (b) All the pixels are mapped with the central point in each sub-image (*t*-1).

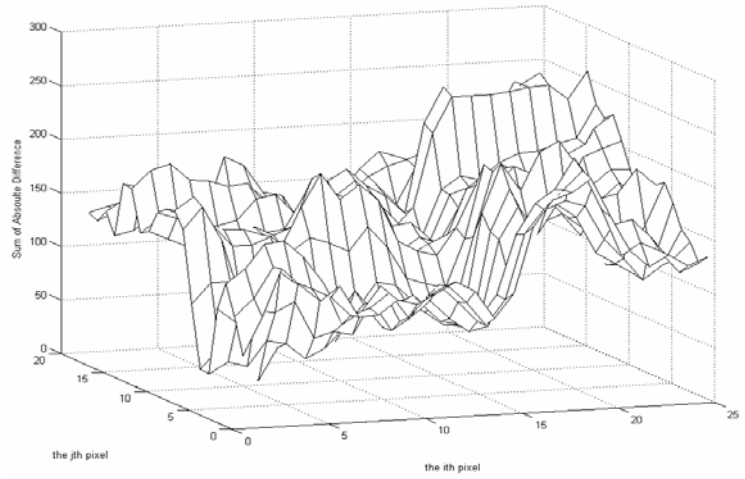
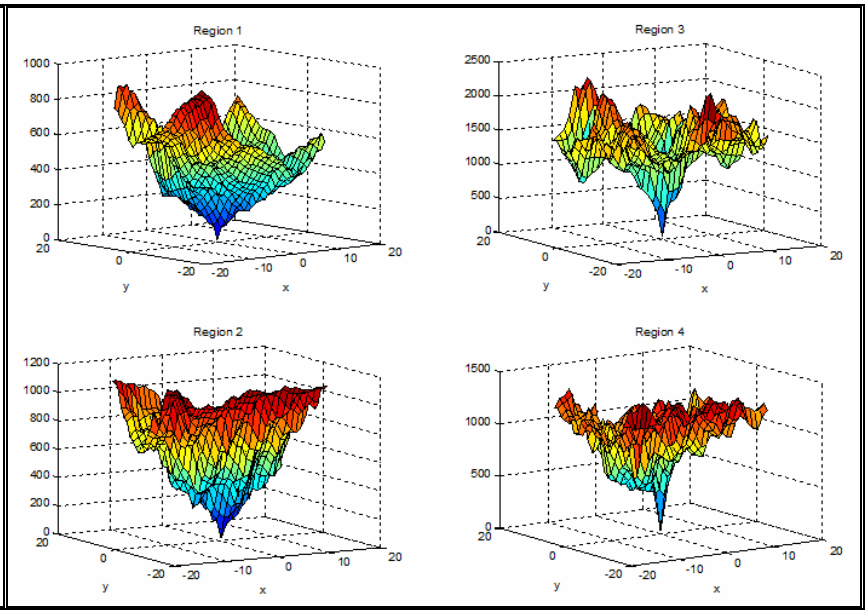


Fig. 4 The accumulated results of all | Msub(*t*) | in the prescribed region form an 19×25 difference value array.

After analyzing the curves of correlation values corresponding to image sequences with various conditions, it is found that the curve of correlation values is related to the reliability of motion detection. Fig.5 shows various correlation curves corresponding to image sequences with different conditions. Fig. 5(a) shows a normal case that each region of the incoming frame has its own obvious minimum position, and we can distinguish them from x and y coordinates. Fig. 5(b) shows a valley shape distribution seeing from y coordinate and each region has its own obvious minimum position. But every x coordinate is not reliable due to lack of clear critical point.

(a)



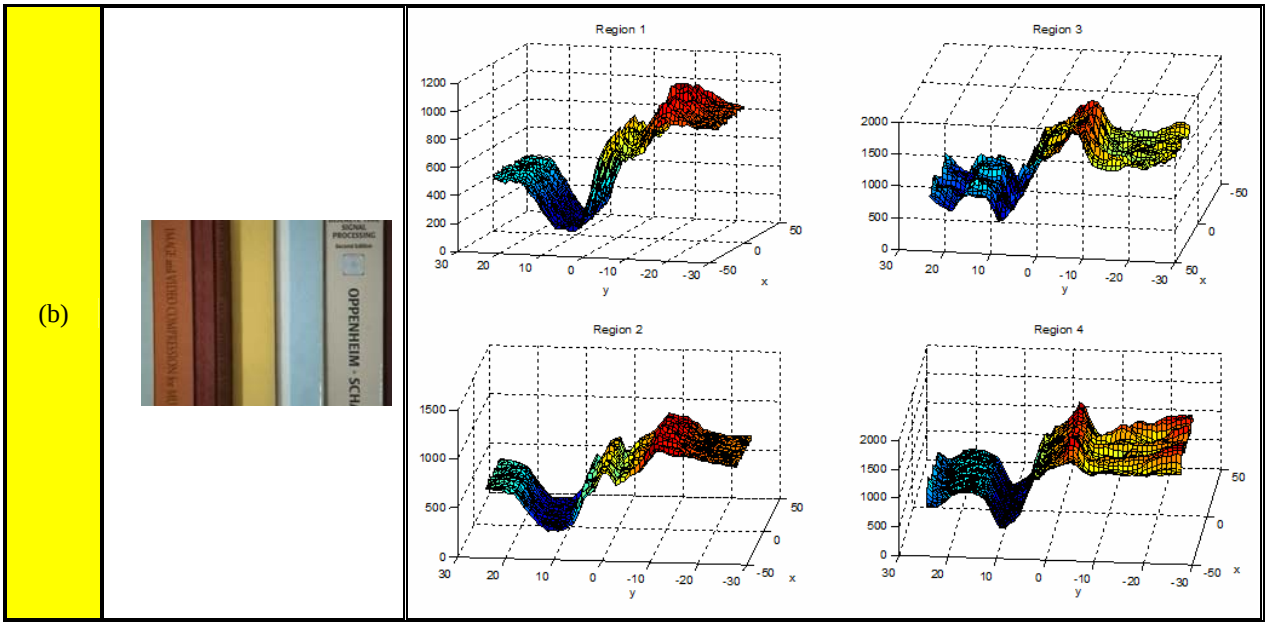


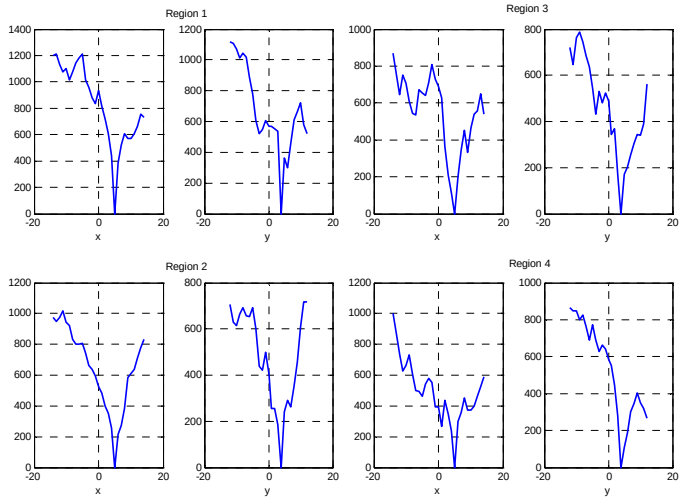
Fig. 5 Various correlation curves corresponding to image sequences with different conditions. (a) Normal case that each region of the incoming frame has its own obvious minimum position. (b) Case of only y coordinate has valley shape distribution.

The curve of correlation values is related to the reliability of motion detection, so we propose a strategy that combines the minimum projections of correlation curve in x and y directions (minimum projections) and the inverse triangle method to detect the irregular conditions from each region to reduce computation complexity. The mathematical expression of minimum projections can be written as

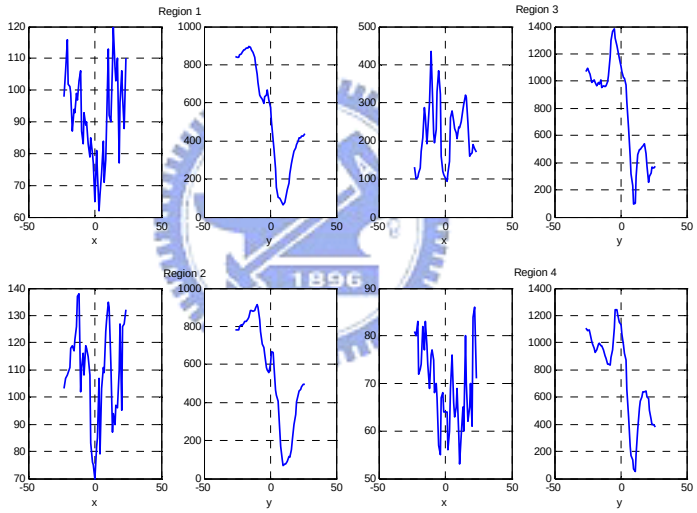
$$x_i\_\min(p) = \min_q R_i(p,q)$$
$$y_i\_\min(q) = \min_p R_i(p,q), \qquad (2.1)$$

where $x_i\_\min(p)$ and $y_i\_\min(p)$ are the minimum projections of correlation curve in x and y directions in region i, respectively. The concept derives from the intuitional sense that the high reliable curve for determining the LMV has a sharp and obvious peak, and no other equivalent peaks appeared in the same curve.

Figure 6 is the projection of x and y correlation curve of each region from Fig. 5. We can see that in Fig. 6(a) each region has obvious minimum x and y position. There are many local minimum points in the x projection in Fig. 6 (b) and only y coordinate can distinguish the minimum position.



(a)



(b)

Fig. 6 Examples of minimum projections of correlation curve from *x* and *y* directions in four regions: (a) regular image sequence and (b) ill-conditioned image sequence.

In order to judge the reliability of the motion vector from Fig. 6 , we combine the inverse triangle method with the minimum projections of correlation curve to find the reliability indices. While the local minimum difference distance is larger than a specific value, the minimum position is not reliable.

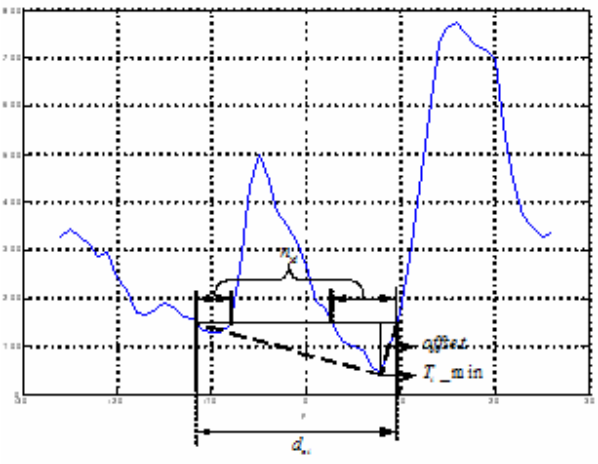


Fig. 7 Illustration of the proposed inverse triangle method.

Base on this criterion, the algorithm is designed as follows. In the first step, we find T_i _min that represents the global minimum of the minimum projection curve in region i and can be calculated by Eq. (2.2). In the second step, we calculate S_{xi} and S_{yi} by Eq. (2.3), where *offset* is the altitude of the inverse triangle, n_{xi} and n_{yi} are defined as the numbers of S_{xi} and S_{yi}, respectively Eq. (2.4), d_{xi} and d_{yi} are defined as the distances of two vertexes of the base of inverse triangle obtained by Eq. (2.5). The confidence level of *x* and *y* directions are calculated by Eq. (2.6). Since the condition of multiple peaks seriously degrades and affects the determination of reliability, the penalty of multiple peaks is taken into account by Eq. (2.6) to improve the discrimination of reliability. The example shown in Fig. 7 is a curve with twin peaks which will get the penalty of $d_{xi} - n_{xi}$. In the third step, we determine the confidence indices of x_i and y_i in region i through a threshold denoted as TH. The smaller value of confidence level represents the higher reliability. In the final step, summing up the counts of reliable motion components of *x* and *y* in four regions as Eq. (2.7), we get $Num(x_i)$ and $Num(y_i)$, $i = 1 \sim 4$.

The follows describe the procedure:

Step 1.

Find global minimum T_i _min from x_i _min(p) or y_i _min(q).

$$T_i\_\min = \min_p(x_i\_\min(p)) \text{ or } T_i\_\min = \min_q(y_i\_\min(q)). \tag{2.2}$$

Step 2.

Calculate the confidence level, x_i_conf and y_i_conf .

$$\begin{cases} S_{xi} = \{p \mid x_i\_\min(p) < T_i\_\min + offset\} \\ S_{yi} = \{q \mid y_i\_\min(q) < T_i\_\min + offset\} \end{cases}, \tag{2.3}$$

$$\begin{cases} n_{xi} = \text{number of } S_{xi} \\ n_{yi} = \text{number of } S_{yi} \end{cases}, \tag{2.4}$$

$$\begin{cases} d_{xi} = \max_P S_{xi} - \min_P S_{xi} \\ d_{yi} = \max_q S_{yi} - \min_q S_{yi} \end{cases}, \tag{2.5}$$

$$\begin{cases} x_i\_conf = 2d_{xi} - n_{xi} \\ y_i\_conf = 2d_{yi} - n_{yi} \end{cases}. \tag{2.6}$$

Step 3.

Set the threshold, TH , for determining the reliability indices.

If $x_i\_conf < TH$ Then x_i is reliable,

Else x_i is unreliable,

End if.

If $y_i\_conf < TH$ Then y_i is reliable,

Else y_i is unreliable,

End if.

Step 4.

Calculate the numbers of x_i and y_i in four regions.

$$\begin{cases} Num(x_i) = \text{sum of } (x_i \text{ is reliable}) \\ Num(y_i) = \text{sum of } (y_i \text{ is reliable}) \end{cases}, \qquad i = 1 \sim 4 \tag{2.7}$$

.

2.1.2 Irregular Motion Vector (IMV)

Irregular motion vectors can be detected and excluded by using minimum projection and inverse triangle method; however, image sequence with ill-condition such as lack of feature, large low-contrast area, moving object or repeated pattern, may contain fewer available MVs (most of the MVs are irregular) in four regions. Therefore, recombination of these available regular MVs is necessary to form an ill-conditioned motion vector (IMV). To solve this problem, a median function is used to extract a motion vector with respect to each direction for ill condition.

The calculation to determine the IMV is described as follows in details.

Case 1. If $Num(x_i(t)) = 4$ then

$$V_{ill\_x}(t) = Med(V_{a\_x}(t), V_{b\_x}(t), V_{c\_x}(t), V_{d\_x}(t), GMV_x(t-1)),$$

Case 2. If $Num(x_i(t)) = 3$ then

$$V_{ill\_x}(t) = Med(V_{a\_x}(t), V_{b\_x}(t), V_{c\_x}(t)),$$

Case 3. If $Num(x_i(t)) = 2$ then

$$V_{ill\_x}(t) = Med(V_{a\_x}(t), V_{b\_x}(t), GMV_x(t-1)), \tag{2.8}$$

Case 4. If $Num(x_i(t)) = 1$ then

$$V_{ill\_x}(t) = V_{a\_x}(t),$$

Case 5. If $Num(x_i(t)) = 0$ then

$$V_{ill\_x}(t) = \gamma \times GMV_{avgx}(t-1),$$

where $Num(x_i(t))$ is the number of *x* component of reliable LMVs, $V_{ill\_x}(t)$ is the *x* component of IMV, $V_{a\_x}(t)$, $V_{b\_x}(t)$, $V_{c\_x}(t)$, and $V_{d\_x}(t)$ represent *x* components of reliable LMVs in different region, respectively, $Med(\,)$ in Eq. (2.8) is the function of median operation, $GMV_x(t-1)$ is the *x* component of last previous *GMV*, t is frame number, γ is attenuation coefficient, $0 < \gamma < 1$.

The $GMV_{avgx}(t)$ can be calculated by

$$GMV_{avgx} = \varsigma \times GMV_{avgx}(t-1) + (1-\varsigma)GMV_x(t) \quad 0 < \varsigma < 1 \tag{2.9}$$

Then we apply the similar process to obtain $V_{ill\_y}(t)$. The resultant IMV is represented by

$$V_{ill}(t) = \begin{bmatrix} V_{ill\_x}(t) \\ V_{ill\_y}(t) \end{bmatrix} \tag{2.10}$$

2.1.3 Global Motion Vector

The LMV in each region may represent global motion vector, moving object motion vector, or even error vector. The error vector may cause by the ill condition or the mixture of global motion and moving object motion. Although the reliable global motion vector is essentially selected from LMVs and IMV, however, in the worst case, i.e. estimations of LMVs and IMV are all fault due to high noise image sequence, it will induce artificial shaking result due to adopt an error GMV. Therefore, if the evaluation includes the zero motion vector (ZMV), it can prevent the occurrence of this case. Similarly, for a high noise image sequence with panning, the last previous GMV will be the best choice if the estimations of LMVs and IMV are all fault. In the proposed IS technique, the seven motion vectors including four LMVs, the IMV, the ZMV, and the last previous GMV, referred as pre-selected motion vectors (*pre_MV*), are employed to estimate the GMV of the current frame. In general, one of LMVs is the highly probable GMV for the regular image; the IMV is the highly probable GMV for ill-conditioned image; the ZMV can prevent worse compensation result caused by the fault MVs; and the last previous GMV is useful for panning condition. In this paper, a background-based evaluation function is proposed to overcome this problem. Fig. 8 shows the areas for background-based evaluation. Five regions are selected to evaluate the result, which are located on the surroundings of the image. The reason is that, in most cases, the foreground object is located on the center of the image, so the surroundings of the image are the best candidates for background detection.

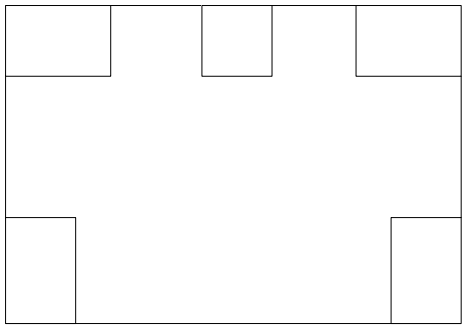


Fig. 8 Areas for background detection and evaluation

The estimation of the GMV is calculated by the summation of absolute difference (SAD) [33],

$$SAD_{B_i,c} = \sum_{X,Y \in B_i} \left| I(t-1, X, Y) - I(t, X + X_c, Y + Y_c) \right|, \quad 1 \le i \le 5, \quad 1 \le c \le 7, \tag{2.11}$$

where $I(t-1, X, Y)$ is the intensity of the point (X, Y) at frame *t-1*, B_i is the *i*-th background region in the image, X_c, Y_c are the components of the seven pre-select motion vectors (pre_MV_c) in *x* and *y* directions. Each pre_MV_c can obtain it's $SAD_{B_i,c}$ in each region. The smaller $SAD_{B_i,c}$ represents the higher probability of the desired motion vector among theses pre-selected motion vectors. Five-region peer-to-peer evaluation can prevent the situation that some partial high-contrast image regions dominate the evaluation result. In this algorithm, each region has an equal priority to determine the result. The pre_MV_c with the smallest S_c is the desired GMV and it can be expressed as

$$GMV = pre\_MV_i, \text{ for } i = \arg(\min_c S_c), \tag{2.12}$$

where $S_c = \sum_{i=1}^{5} S_{i,c}$. The score for each pre_MV_c in region i is denoted as $S_{i,c}$. Hence, S_c is the important index to determine the GMV.

2.2 Motion Compensation

It is necessary to generate the compensating motion vectors (CMVs) for removing the undesired shaking motion while keeping the steady motion of the image sequence. The conventional compensating motion vector estimation was given by [18]

$$CMV(t) = k(CMV(t-1)) + (\alpha GMV(t) + (1-\alpha)GMV(t-1)), \tag{2.13}$$

where t represents the frame number, $0 < k < 1$ and $0 \le \alpha \le 1$. In the case, there is the tremendous lag condition due to the steady panning effect. It will reduce the available effective image area. The CMVs are generated by Eq. (2.13) with the clipper function [34] as

$$CMV\,(t) = clipper\,(CMV\,(t)) = \frac{1}{2}\left(\left|CMV\,(t) + l\right| - \left|CMV\,(t) - l\right|\right), \tag{2.14}$$

where l is boundary limitation, i.e. maximum window shift allowance. The lag can be reduced to a certain range; however it will also decrease the performance of shaking compensation due to the picking window operating near the boundary area. To attack this drawback, we combine the inner feedback-loop integrator with clipper function to reduce the steady-state lag for steady motion as well as to keep the CMV to operate in the appropriated range. Fig. 9 shows the block diagram of the proposed CMV generation method. There is an integrator in the inner feedback loop, which can eliminate the steady-state lag of the CMV in panning condition. That means, by employing the integrator, shaking components of images with constant panning as well as those in regular images can be stabilized. The proposed CMV computation procedure is presented by

$$CMV(t) = \mathrm{K} \cdot CMV(t-1) + [\alpha \cdot GMV(t) + (1-\alpha) \cdot GMV(t-1)] - \beta \cdot CMV\_I(t-1), \qquad (2.15)$$

$$CMV\_I(t) = CMV\_I(t\text{-}1) + CMV(t) \text{ and } \mathrm{CMV(t)} = clipper(CMV(t)),$$

where $[0\ 0]^{\mathrm{T}} \le \mathrm{K},\alpha,\beta \le [1\ 1]^{\mathrm{T}}$ and *clipper*() is defined as Eq. (2.14).

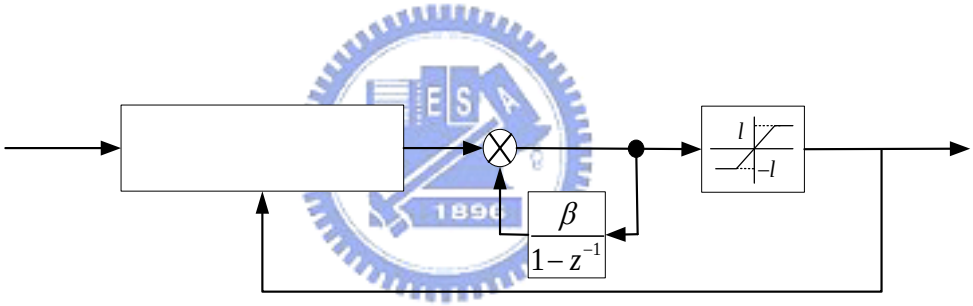


Fig. 9 The block diagram of the proposed CMV generation method.

2.3 Implementation of Local Motion Estimation

After programming the DIS system, we find that it can do an excellent off-line job comparing with the RPM fuzzy set theory [32]. But it is hard to implement the algorithm into practical consumer camcorders. Although there are still several DIS systems which could do the on-line job by using PC [45], it is still a long way to go into the DIS hardware design.

Therefore it is necessary to analyze the DIS computation load before the system implement on practical camcorders. We record the computation time from each DIS step and compare them with whole processing time of each incoming test video frame. Then we take average of the computation percentage of all incoming frames and list them in Fig. 10. The percentage data

might lack of accuracy due to different test videos, but we still can recognize that most of the computation loading belongs to the motion estimation step. Fig. 10 shows the percentage of the computation loading of each block in DIS system. The motion estimation step contains 80 % processing time and the motion compensation contains the rest 20%. And more than halve computation time belongs to local motion estimation step. This is because the RPM method in LMV step need to load the image SAD values into memories first and find the minimum value pixel by pixel. The tasks of finding global minimum position by using a DSP processor is comparing and storing the minimum value of neighbor pixel and jump to next pixel. The minimum position information should also store in the memory. The larger the processing unit is, the longer processing time it takes. Our strategy is to design an application specific chip which could highly reduce the computation time in LMV estimation. The less the processing time in LMV estimation step the higher capability of real-time operations for image stabilization processing is.

The LMV estimation chip is designed with CNN technology to solve the heavy computation time problem. Compared with conventional digital technology, CNN-based computing is capable of realizing these TeraOPS-range image processing tasks in a cost-effective implementation. The design concept of ASCNN chip is shown in Fig. 11. By using an 8-bit D/A converter, the absolute image difference which ranges from 0~256 could store into the CNN local analog memories (LAM). And with the aid of CNN technology, the LMV position could be easily found compared with the DSP processor. The CNN theory and chip design are introduced in Chapter 3 and Chapter 4.

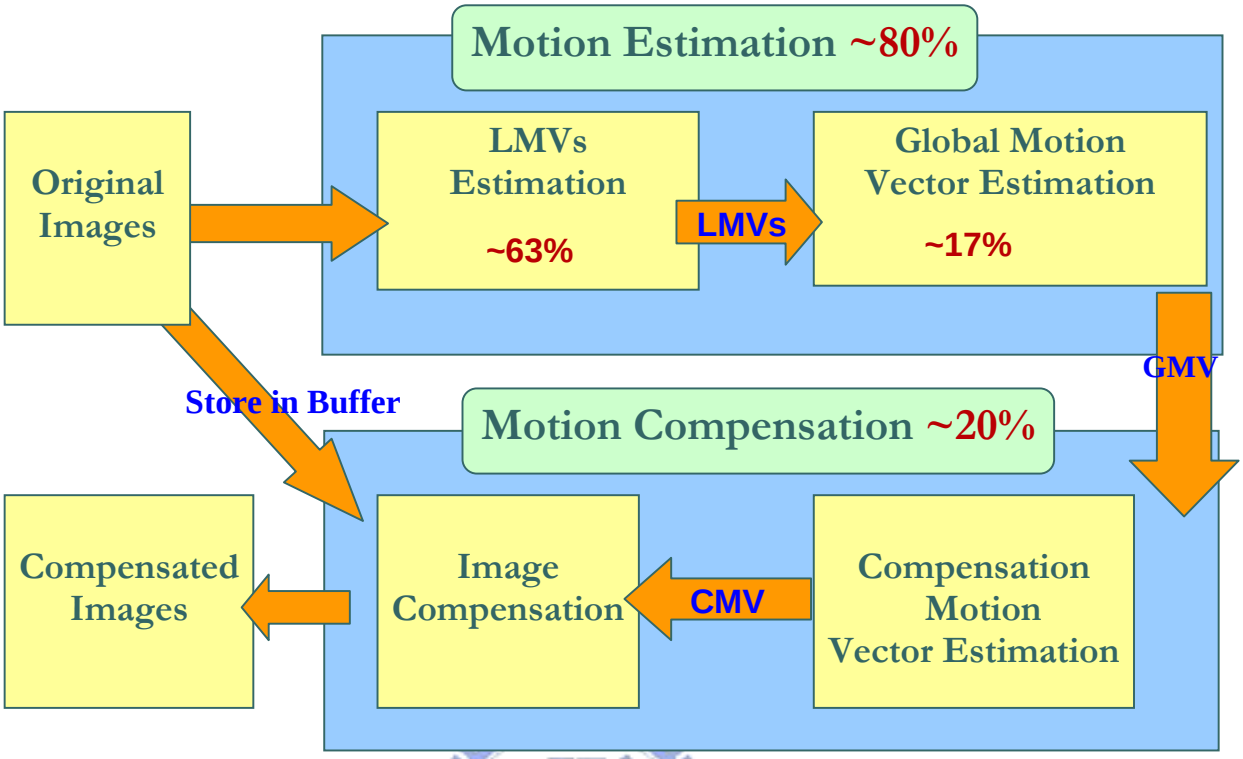


Fig. 10 Computational complexity of the DIS flow.

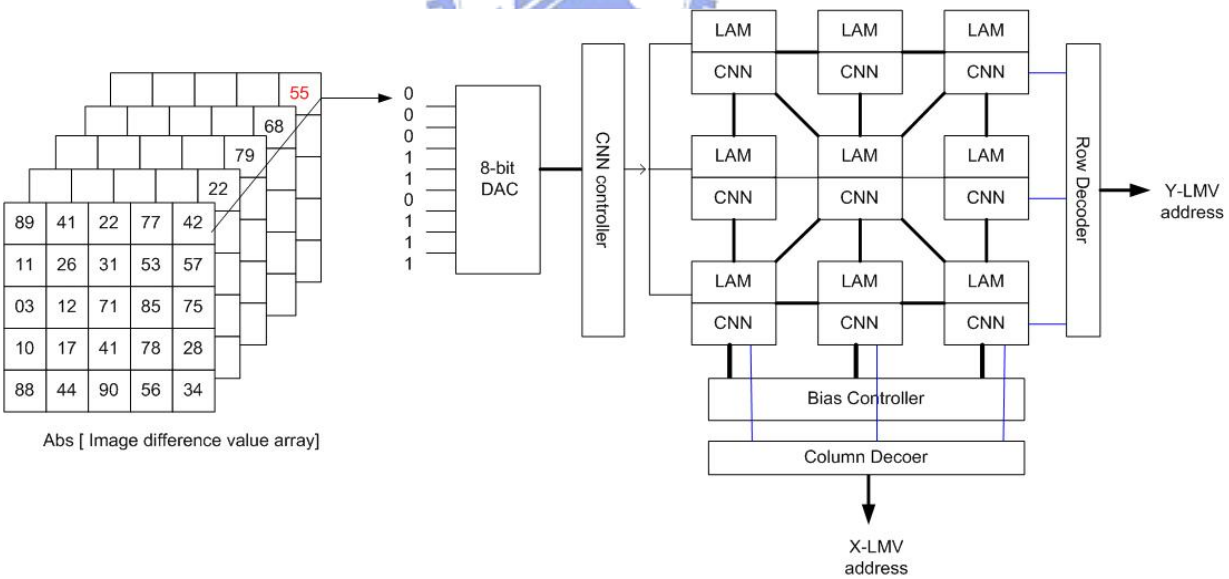


Fig. 11 The design concept of ASCNN chip architecture.

CHAPTER 3

Cellular Neural Network

The original Cellular Neural/Nonlinear Networks (CNN) paradigm was first introduced by Chua and Yang [20]. CNN technology is both a revolutionary and experimentally proven new computing paradigm. The two of most fundamental ingredients of the CNN are: the use of analog processing cells with continuous signal values, and local interaction within a finite radius. CNN possesses some of the key features of neural network, which has important potential applications in such areas as image processing and pattern recognition. The CNN theory and architecture will be introduced first and the next is CNN circuit. The last will include the inversion and adaptive threshold properties of CNN which are used to calculate the LMV.

3.1 CNN Theory

CNN can be considered an implementable alternative to fully connected neural networks and a remarkable improvement in hardware implementation of artificial Neural Networks. Local interconnection and simple synaptic operators are the most attractive features of the CNN for VLSI implementation in high-speed, real-time applications [35] and the CNN are widely used in several application fields, such as image processing and pattern recognition. Several hardware implementations of the CNN have been reported in the literatures [37], [38], [39].

The state equation of CNN can be represented by

$$\dot{x}_{i,j}(t) = -x_{i,j}(t) + \sum_{k,l \in Nr(i,j)} A_{i,j;k,l} y_{k,l}(t) + \sum_{k,l \in Nr(i,j)} B_{i,j;k,l} u_{k,l}(t) + I_{i,j}, \tag{3.1}$$

$$y(t) = f(x(t)) = \frac{1}{2}\left(|x(t)+1| - |x(t)-1|\right), \tag{3.2}$$

where i, j refers to a grid point associated with a cell on the 2-D grid, and $k, l \in Nr(i,j)$ is a grid point in the neighborhood within a radius r of the cell i, j. A and B are the nonlinear cloning templates [40]. Fig. 12 shows the dynamic route of state in CNN. The feature of the Eq. (3.2) has been plotted at Fig. 13.

In many applications, the templates (A,B) and the threshold *I* are translation invariant. In the case of single variable *A* and *B* functions, the linear (space-invariant) template is represented by the additive terms as Eq. (3.1). When the template is space invariant, each cell is described by a simple identical cloning template defined by two (2r + 1) × (2r + 1) real matrices *A* and *B*, as well as the constant term *I*. In addition, as a very special case, if the input and the initial state values are sufficiently small and *f* is piecewise linear, then the dynamics of the CNN array is linear.

Unlike other standard analog processing arrays, or neural networks, the one-to-one geometric (topographic) correspondence between the processing elements and the processed signal-array elements (e.g., pixels) is of crucial importance. Moreover, the template has geometrical meanings which can be exploited to provide with geometric insights and simpler design methods.

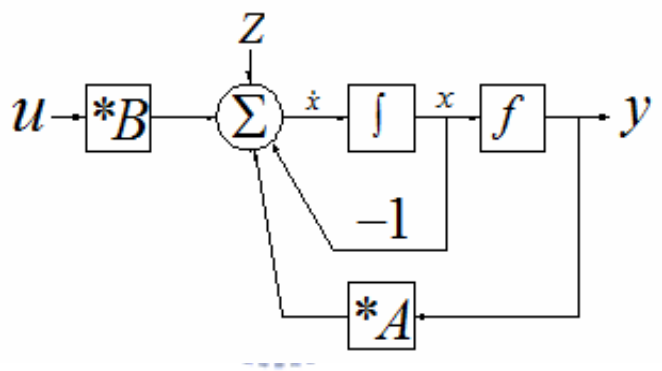


Fig. 12 The dynamic route of state in CNN.

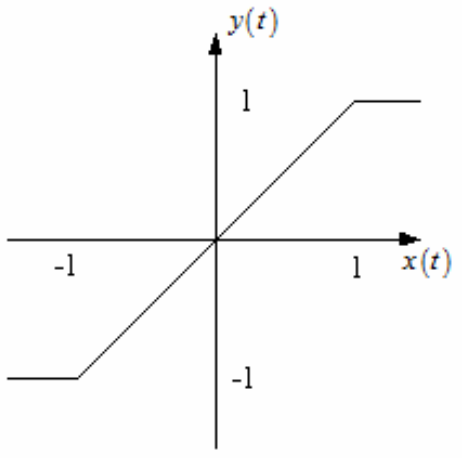


Fig. 13 The output of sigmoid function.

3.2 CNN Architecture

The basic circuit unit of CNN is called a cell. It contains linear and nonlinear circuit elements, which typically are linear capacitors, linear resistors, linear and nonlinear controlled sources, and independent sources. The structure of CNN is similar to that found in cellular automata, and each cell in a CNN is connected only to its neighbor cells. Adjacent cells can interact directly with each other. Cells not directly connected together may affect each other indirectly because of the propagation effects of the continuous-time dynamics of the network. A typical example of a cell *C*(i, j) is shown in Fig. 14, where the suffixes *u*, *x*, and *y* denote the input, state, and output, respectively.

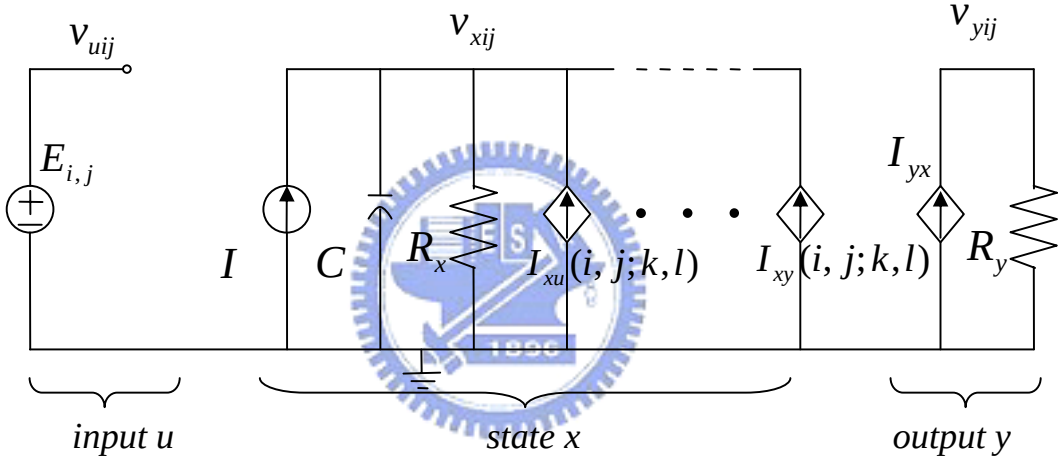


Fig. 14 The circuit of a CNN cell.

The differential equation governing a CNN in Eq. (3.1) is rewritten as follow :

$$C\frac{dv_{xij}(t)}{dt} = -\frac{1}{R_x}v_{xij}(t) + \sum_{C(k,l)\in N_r(i,j)} A(i,j;k,l)v_{ykl}(t) + \sum_{C(k,l)\in N_r(i,j)} B(i,j;k,l)v_{ukl}(t) + I,$$
$$1 \le i \le M; 1 \le j \le N. \qquad (3.3)$$

$$where\ y \in R^N : -1 \le y \le 1$$

And A is an M-by-N (M=N) real symmetric matrix defined as

$$A = \begin{bmatrix} A_0 & A_1 & 0 & \dots \\ A_1 & A_0 & A_1 & \dots \\ \dots & \dots & A_0 & A_1 \\ \dots & \dots & A_1 & A_0 \end{bmatrix} \qquad (3.4)$$

Here $I = [I_1\ I_2\ \dots..\ I_N]^T$ is an N-by-1 constant vector and the input vector v_u can be defined in a similar way. A_0 and A_1 are two $m \times m$ Toeplitz matrix with elements determined by a given cloning template. The synapse weights of the shift-invariant CNN can be described by the feedback and feed forward cloning templates:

$$T_A = \begin{bmatrix} a_2 & a_1 & a_2 \\ a_1 & a_0 & a_1 \\ a_2 & a_1 & a_2 \end{bmatrix} \qquad T_B = \begin{bmatrix} b_2 & b_1 & b_2 \\ b_1 & b_0 & b_1 \\ b_2 & b_1 & b_2 \end{bmatrix} \qquad (3.5)$$

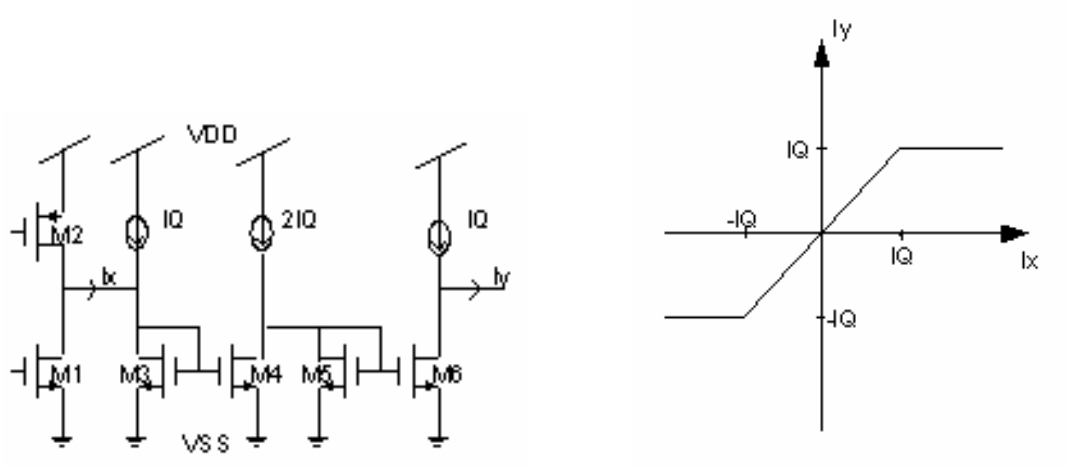
Where all elements respect the normalized numbers to *Tx=1/Rx*, and $a_i = A(i, j; i, j)/Tx > 1$. The matrix B can be defined in the similar way. Then, the maximum value of x in the steady state is the sum of absolute values of all inputs from the neighborhood cells,

$$|x_{\max}| = \sum_{i,j=1}^{3} |T_A(i, j)| + \sum_{i,j=1}^{3} |T_B(i, j)| + |x_0| \qquad (3.6)$$

where $|u| \le 1, |y| \le 1,$ and $x_0 = I/T_x$. The neuron cell should be able to handle the state voltage of the range $|x| \le |x_{\max}|$.

3.3 CNN Circuit Design

The current-mode approach [40], [42] is used in CNN circuit design because it has superior mathematical addition properties. The summation of weighted currents is simply done by appropriate transistor sizing. The piecewise-linear function is achieved by cascading two current limiters as shown in Fig. 15.



(a) (b)

Fig. 15 Piecewise linear function. (a) Schematic view and (b) Transfer characteristics of two current limiters in the cascade [36].

The limiting operation of the input current denoted by Ix first takes place at a negative value $Ix = -IQ$ and at a positive value $Ix = IQ$. For $Ix \leq -IQ$, there is no currents that flow through the transistors M_3 and M_4. Therefore, $I_{DS5} = I_{DS6} = 2IQ$ and $Iy = -IQ$, where I_{DS5} represents the drain-to-source current of M_5, and so on. For $Ix > -IQ$, $I_{DS3} = I_{DS4} = IQ + Ix$ and $I_{DS5} = I_{DS6} = (IQ - Ix)$ produce the output current $Iy = IQ - I_{DS6} = Ix$. However, if $Ix > IQ$, then $I_{DS5} = I_{DS6} = 0$ and $Iy = IQ$.

Figure 16 shows a detailed schematic diagram of a neuron cell [36]. The synaptic weight is realized by M_{11} - M_{14} for a_0 and $M_{15} - M_{16}$ for a_1. Four copies of a current mirror are used to provide the weight for fore neighboring cells. The external input current, bias current, feedback current, and those from the neighboring cells are summed at the drain terminal of M_1. The offset circuit provides a bias current which is set by bias voltage VBI. The output voltage generator is made of a simple current comparator using a cascade of two inverters. The input currents from neuron circuit, weighting circuit and reference voltage set by VBB are compared to produce an output Vy which represents the sign of the neuron output.

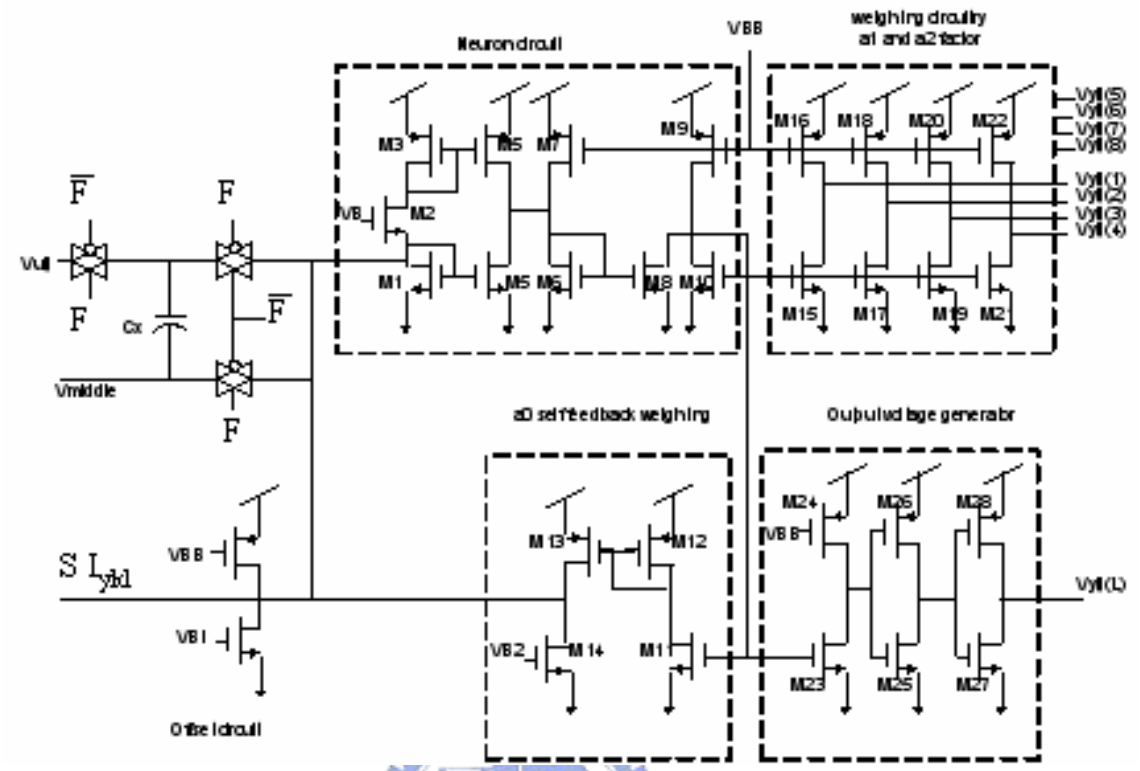


Fig. 16 The CNN cell with fixed weights (templates).

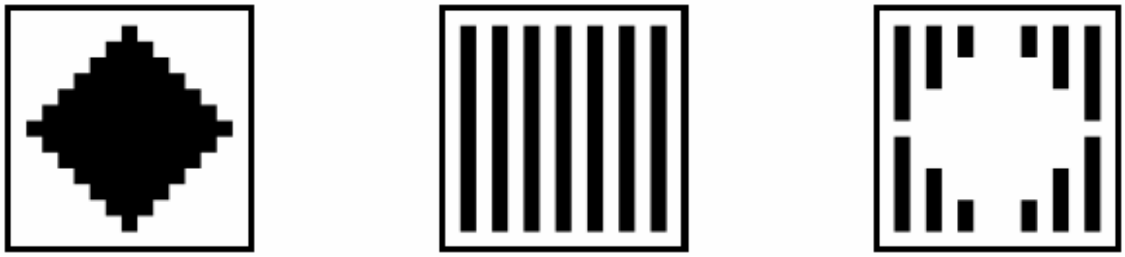
3.4 CNN Template Consideration

3.4.1 Image Difference

The first step of RPM method will subtract the present sub-region pixels with past representative point pixel color. We can implement subtraction step with image inversion and current addition. The inversion template [43] lists below. The input of CNN is grayscale representative sub-region.

$$T_A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad T_B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{3.7}$$

Figure 17 shows the simulation result of CNN inversion template [44]. Fig. 17 (a) shows the input of CNN. Fig. 17(b) shows the initial state of CNN, and the state will subtract from input. The Fig. 17(c) shows the output after processing.



(a) (b) (c)

Fig. 17 Simulation of CNN inverse template. (a) Input of gray-scale image. (b)The initial state of CNN. (c) Output after difference processing.

Because we take current mode CNN as processing core, the addition step can set initial state of CNN as zero and directly combine the input current between the inversion representative sub-region and present sub-region.

3.4.2 Global Minimum

To search the minimum position in a specify area not only takes time but also consumes lots power. Comparing previous value and storing the minimum value is the basic processing step. The larger area need to be determined, the more clock cycle, ie., power, it takes. Egusa [17] proposed to use analog circuit to find the global minimum value. But the circuit only suit for few input application. Therefore, we propose to use CNN adaptive threshold template with capability of finding the global minimum position in larger array and can process with less clock period. The adaptive threshold template lists below.

$$T_A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad T_B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{3.8}$$

The adaptive threshold template not only simplifies the CNN state equation (3.1), but also makes the template easier to implement in VLSI. We can write an equation to represent Eq.(3.8) as:

$$\dot{X} = -X + AY + Bu + I = -X + 2Y + u + I \tag{3.9}$$

We then set the set the initial state of CNN as zero which is reasonable for circuit design and also don't need to implement any initial circuit for CNN core. And we set the sigmoid function will saturate at $\pm 20uA$. This is because we give every current source of CNN core as 20uA.

The CNN adaptive threshold template analysis is shown in
Table 1.

Table 1： CNN Adaptive Threshold Template Analysis

	Case 1	**Case 2**
Ibias	*Iu = -10uA and x (initial) = 0* $\dot{x} = -x + 2x + (-10) + Ibias = x - 10 + Ibias$	*Iu = 10uA and x (initial) = 0 ;* $\dot{x} = -x + 2x + (-10) + Ibias = x + 10 + Ibias$
-20	$\dot{x} = -10 + -20 = -30$ *,saturate* *y still strict in -20*	$\dot{x} = 10 + -20 = -10$ *now x = 0 – 10 = -10 ,unstable* $x = -10 + 10 + -20 = -20$ *now x = -10 -20 = -30 ,* *and y strict in -20 ,stable*
-10	$\dot{x} = -10 + -10 = -20$ *now x = 0 -20 = -20 = y , stable*	$\dot{x} = 10 + -10 = 0$ *now x = 0 +0 = 0 , critical point !*
0	$\dot{x} = -10 + 0 = -10$ *now x = 0 -10 = -10 , unstable* $\dot{x} = -10 - 10 = -20$ *now x = -30 and y strict in -20*	$\dot{x} = 10 + 0 = 10$ *now x = 0 +10 = 10 , unstable* $\dot{x} = 10 + 10 + 0 = 20$ *now x = 20 = y , stable*

10	$\dot{x} = -10 + 10 = 0$ *now x = 0 + 0 = 0 , critical point !*	$\dot{x} = 10 + 10 = 20$ *now x = 20 = y ,stable*
20	$\dot{x} = -10 + 20 = 10$ now x = 0 + 10 =10 = y , unstable $\dot{x} = 10 - 10 + 20 = 20$ now x = 20 + 10 = 30, and y strict in 20 ,stable	$\dot{x} = 10 + 20 = 30$ *now x = 0 + 30 = 30* *and y strict in 20 ,stable*

We use the property of 【case 2】 to implement adaptive threshold template on searching global minimum position in RPM method. With the aid of CNN array, a brand-new searching method has developed for DIS algorithm. As shown in Fig. 18, the SAD values are plotted in 3D view, and the CNN output will change from logic 1 to logic 0 while any of the difference value is below the threshold level and others remain logic 1. If none of the position in the processing array flip its output logic, CNN bias control circuit will tune the threshold to a higher level until the minimum input in the array is lower than bias current.

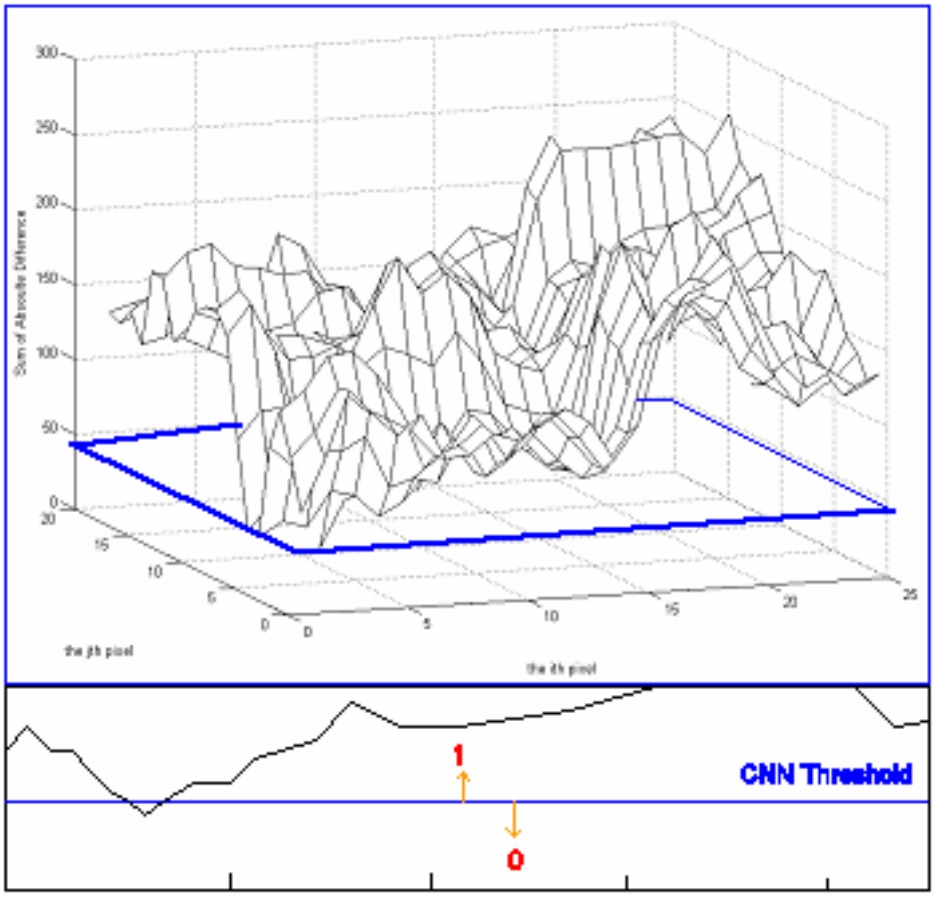


Fig. 18 Searching global minimum by using CNN adaptive threshold template.

CHAPTER 4
CIRCUIT DESIGN OF CNN-BASED LMV ESTIMATION

The process of finding LMV is very computationally intensive, requiring billions of operations for each image. The most complexity operations occur in 1) computing the motion vector and the difference value and 2) storing the difference value with the position information if it is smaller than any previous value. Since the operation slows down the computation, the CNN architecture is suitable for motion computation and is done by CNN with a fixed template and the tunable bias current circuit for each cell.

The tested image is a tennis player video of 312×200 pixels. Since image sensors can get more pixels than the video image requires, each image captured by sensors will first cut out a specific boundary pixels which saves as compensation area. Removing boundary area is called pre-processing step, and each image size now becomes 300×190 pixels. The motion estimation block diagrams which finds LMVs with CNN processing is shown in Fig. 19. Before entering CNN processing, the captured image has to be cut into four regions, and each of region will be found their own LMV. After this, each region again is divided into 30 sub-regions as mentioned in Chapter 2. Each sub-region is the size of 19×25 pixels processing block. We first store the center point image color value (0~255) of every sub-region in the prescribed region which is captured by $(t\text{-}1)^{th}$ image sequence. According to the RPM method described in Chapter 2, the absolute difference information should be computed by t^{th} and $(t\text{-}1)^{th}$ sub-images. Through the digital-to-analog converter, the difference information could be stored in Local Analog Memories (LAM). The 30 sub-regions' absolute difference value array will stack into CNN LAM from each processing region and the memories voltage information will vary from the difference values stored in the LAM. CNN does not begin to compute motion vectors until LAM accumulates all the difference information of 30 sub-regions. The CNN processor will check the global minimum position by using a 32-level threshold bias. The minimal difference value and the position information would be found within 32 clock cycles and then be latched in the location registers.

The processing time is not effected by the size of CNN. Therefore the larger the difference array is, the faster processing time compared with DSP processor will be.

The system shown in Fig. 19 includes windowing (RPM and SAD), the 19×25 CNN array and LAM, the bias control circuit, and the addressing decoder.

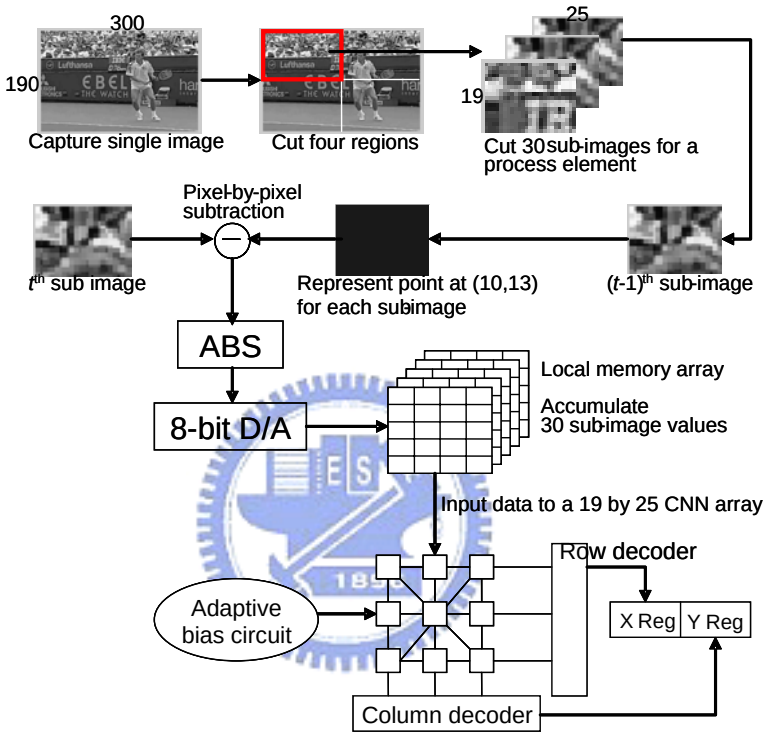


Fig. 19 The flow of CNN-based local motion estimation.

Figure 20 shows the architecture of the application-specific CNN (ASCNN) design. By the windowing component of Fig. 19, the sequential images are segmented into many 19×25 pixels sub-regions for each region, and the absolute difference [17] of two images is calculated. Through an 8-bit D/A converter, the difference value is loaded and accumulated into LAM which consists of switch MOS capacitors. The bias current circuit will adjust the CNN array's threshold according to the values of global output connected chains. If all difference values are higher than the given bias, the higher current fed into the CNN input from bias circuit will be. The process will be finished if the smallest input difference value is lower than the bias current. With the global connection of each row and column's output, the digital row and column address decoder

will detect the X and Y position information in no time and store in the registers.

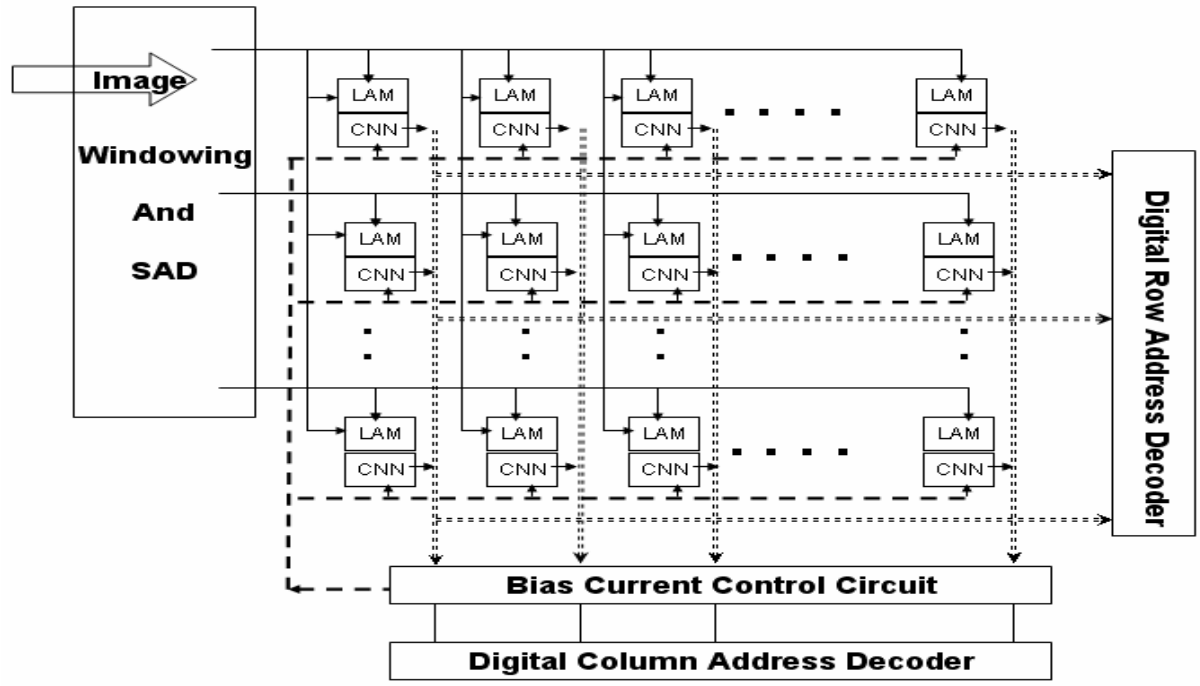


Fig. 20 AS-CNN chip architecture.

4.1 D/A Converter

Analyzing the minimum image difference value of the input video sequence is necessary for the D/A converter. Fig. 21 shows the image sequences verses the minimum pixel difference values of each frame. We can calculate that the mean of minimum difference value is located in 529 difference values, but the data would vary from each video. Therefore the upper bound of the minimum difference has 1024 pixels. The upper bound takes maximum charging input for four times, ie., 256×4, before LAM reach the 3.3 volts. Note that there is an exception in the sequence No.48. The minimum value is over 1024. This is because there's a great movement for the whole region which is caused by intentionally moving the camcorder or the object is too large in this region so that it is considered as the background. LMV located in this region is not dependable and should not be stored. This kind of situation can be detected and discarded by the CNN controller.

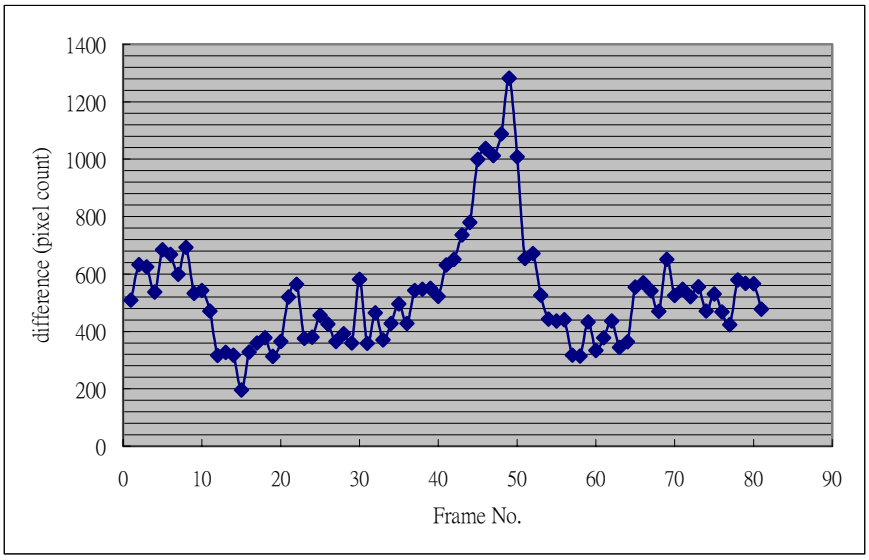


Fig. 21 Analysis of minimum difference value for tennis player video frames.

An 8-bit D/A converter is used to translate the image absolute difference code into analog current and load into local analog memory. In the first, the DAC input will pass registers in order to synchronize with digital control circuit, which is trigger by 20 MHz clock.

The DAC is made of eight sets of current mirrors shown in Fig. 22 and the output stage is the 475 sets of LAM. Table 2 lists the DAC specification to keep the function accuracy of the input stage design.

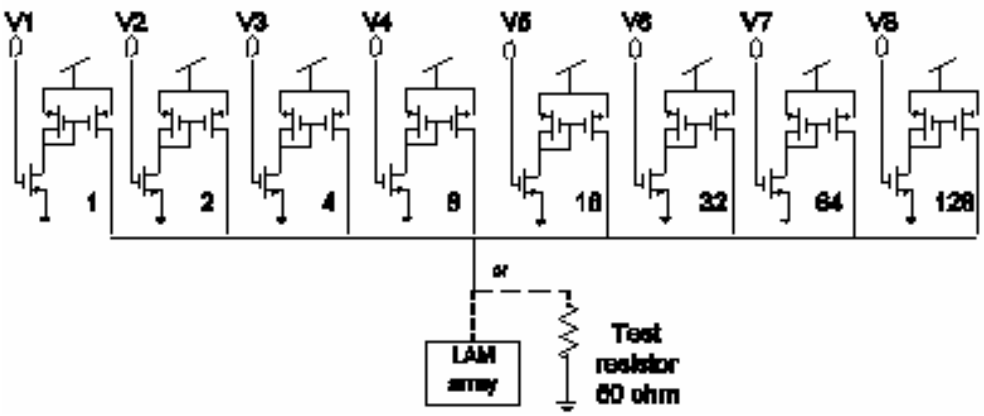


Fig. 22 8-bit current mode D/A converter.

Table 2： 8-bit D/A Converter Specification Table.

Model	Application Specific CNN 8-bit DAC
Output Loading	50Ω
Resolution	8-bit
Relative Accuracy **0~70°C**	| 0.5 LSB |
Output range *0~255* **Full output for LAM**	0~5.5mV 0~3.3V
DNL	Max DNL < 0.25 LSB
INL	0.1615 LSB
SNR	55.8148 dB
Digital Input Data Input , Voltage : Logic 1 Logic 0 Control Input, Voltage : Logic 1 Logic 0 Input capacitance	 3.3V 0V 3.3V 0V 4pF
Power Supply Operating Voltage Range (VDD=3.3v) MAX @ 11111111 current (Icc)	 6.9513 mW 2.1064 mA
Operating Temperature	0 ~ 70 °C

1. Output Swing

Figure 23 shows the output voltage variation of 256 levels with 50Ω loading. The charging voltage of the analog memory in the output stage is proportional to the input current and the loading time. The larger the image difference value is, the higher the memory voltage will be.

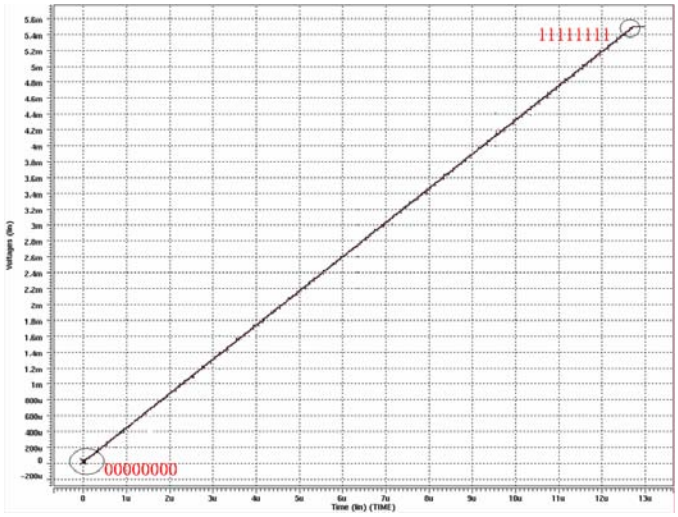


Fig. 23 The output voltage variation due to input change of (8-bit) 256 steps.

2. DNL

Figure 24 is the DNL analysis between the output voltage in Fig. 23 and ideal voltage curve ie. V_ideal=50Ω×0.43091/1000×t. The value 0.43091e-3 is the slope of output voltage due to the input change from 0 to 1. The coordinate Y represents the difference percentage of two curves versus LSB. The largest DNL in Fig. 24 is about 0.25 LSB. The coordinate X represents the time and the time step is 50 ns while input change data from 0 to 255.

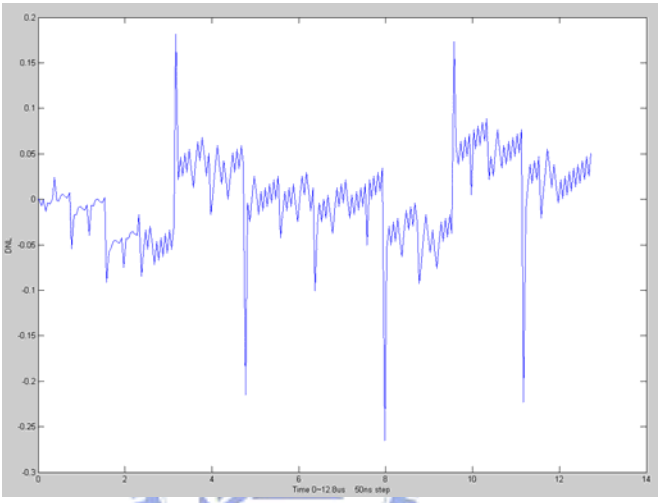


Fig. 24 DNL analysis.

3. INL

The INL is accumulated form all DNL data in Fig. 24. The result shows that,

INL = 0.1615 LSB < 0.5 LSB

4. SNR

Because the input working frequency is set in 20MHz ie, fclk=20MHz, we give a sine wave input with 20MHz/28 frequency for DAC and record the output wave form as shown in (a). Then we sample the output stable voltage at 25 ns and get 140 sampling points. DFT position analysis is shown in (b). We take logarithm for the x coordinate and find the harmonic position in (c). The calculation results list below.

Power DC= 0.1479

Power Sine = 0.0369

Power Homonic = 1.6947e-007

Power Noise = 9.6831e-008

SNR_dB = 55.8148dB > 49 dB

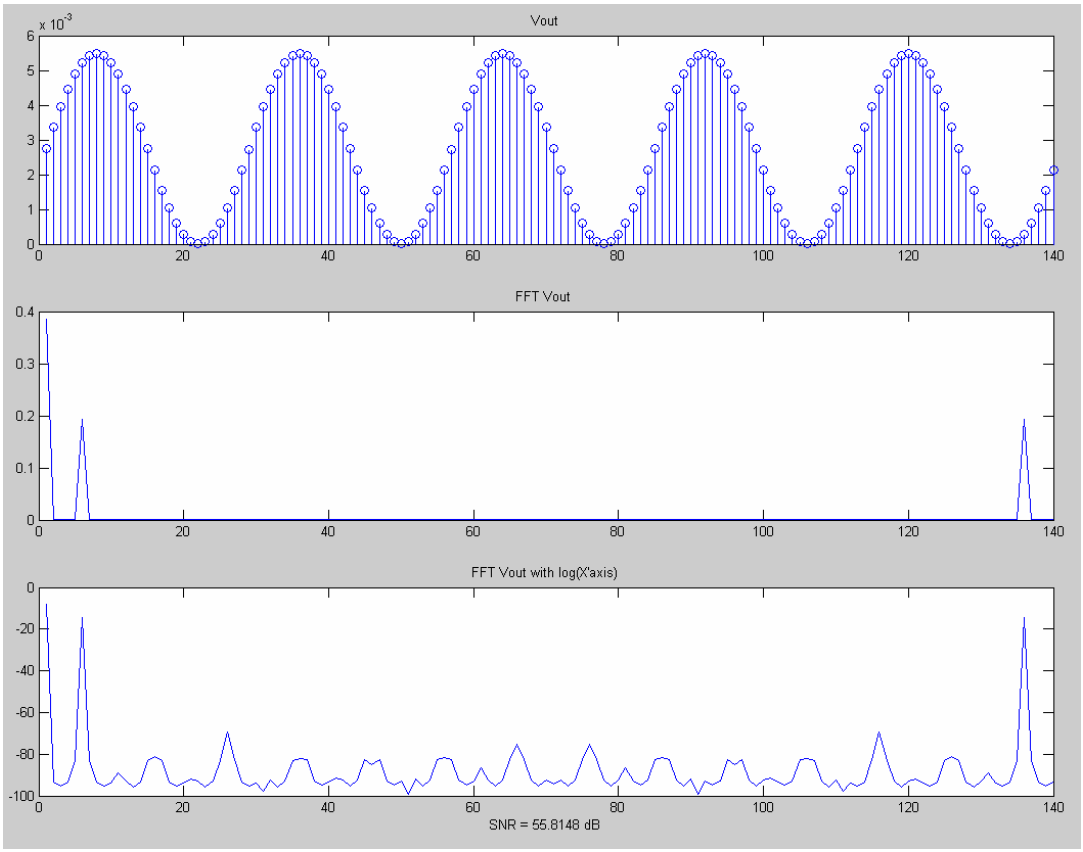


Fig. 25 SNR analysis of 8-bit DAC. (a) sine wave output wave form of DAC; (b) DFT analysis of DAC output form with 140 sampling points; (c) The logarithm value for the x-coordinate in (b).

Layout of the DAC is shown in Fig. 26. The 8-bit input is from the output of synchronize registers and the maximum current output is 2.1064mA which is designed to make each LAM cell has five times charging capability before reaching the 3.3 volt upper bound.

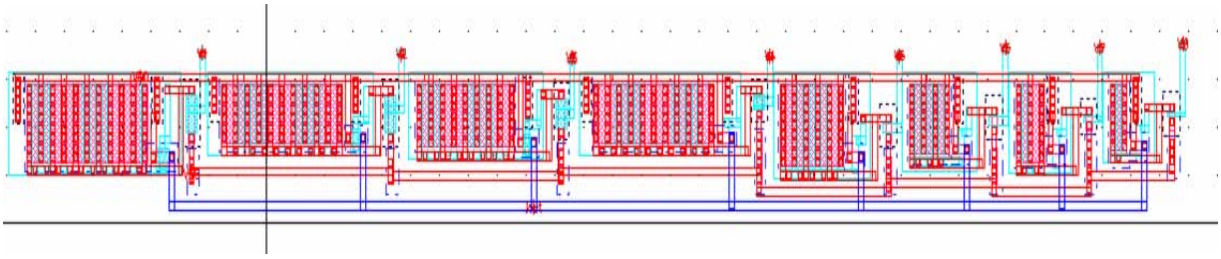


Fig. 26 Layout of 8-bit current mode D/A converter

4.2 Local Analog Memory

Local analog memory (LAM) is designed to store the image difference value. The basic structure of the LAM is shown in Fig. 27 is a LAM cell. It consists of a transmission gate controlled by the CNN controller and a 2p MOS capacitor. defined in input stage. Vctrl_P and Vctrl_N are switch signals which are from the position decoder. Although there exists non-linear problems for the MOS capacitor design, for area consideration we still choose it rather than the poly capacitor. For a 2pf poly capacitor design, the nonlinear problems is much easier to solve, but the area take 2314um^2 while the MOS capacitance only cost 1800um^2 [51]. Therefore using of MOS capacitor could save much on-chip area. The characteristic curve of MOS capacitor is shown in Fig. 28. Since MOS capacitor has nonlinear transformative property, the capacitor value will have 600f variation while the gate voltage Vgs change from 0 to 0.65 volts. Pre-charging a specific voltage, i.e., 0.65 should be done before loading information into each capacitor. Layout view is show in Fig. 29. Symmetric layout style is used for every two LAM cells to reduce the mismatch during fabrication.

However, accuracy of MOS capacitor value is not important. As long as the global LAM array has uniformly capacitor properties, CNN processor will be able to find the correct global minimum position.

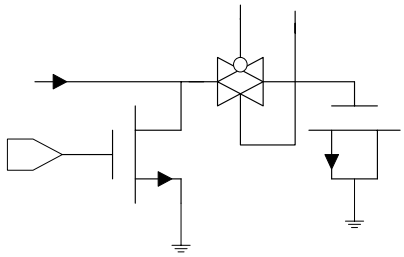


Fig. 27 The single structure cell LAM.

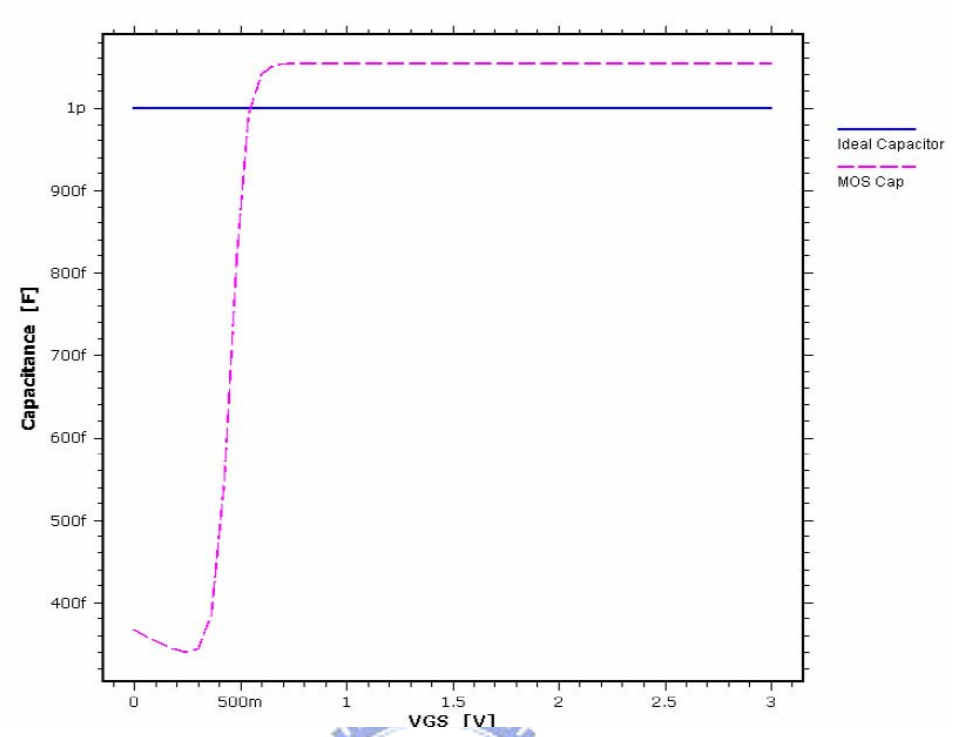


Fig. 28 The characteristic curve of MOS capacitor.

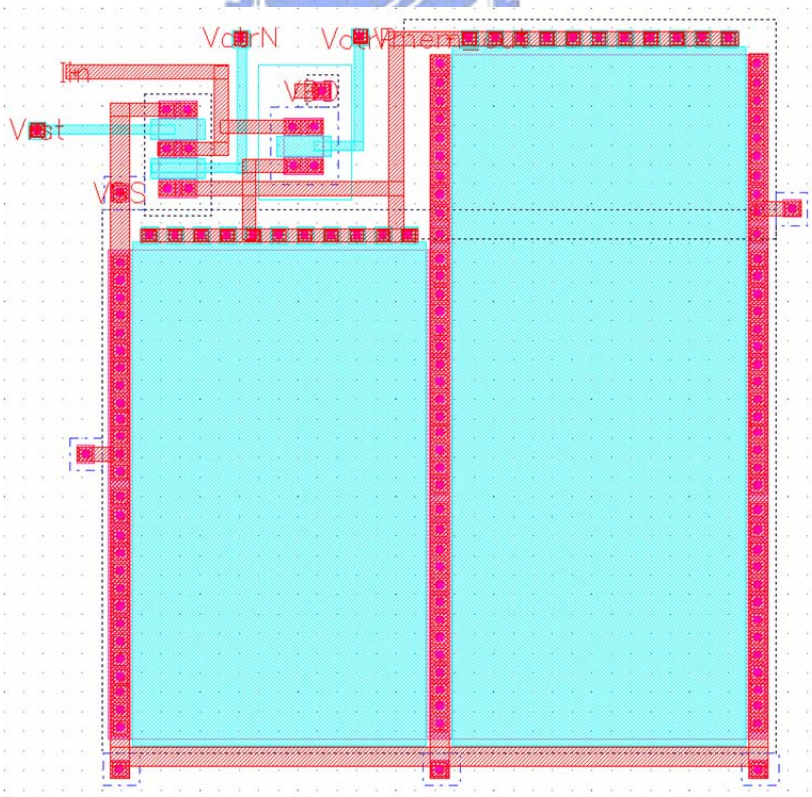


Fig. 29 Layout of MOS capacitor.

4.3 Voltage to Current Converter

A voltage to current converter (VCC) shown in Fig. 30 is required to transform the LAM voltage into CNN current input. With properly design the W/L ratio, the output of VCC to CNN is limited to 20uA. VCC maximum power consumption is 104.6uW when the input signal is 3.3 volts. Fig. 31 shows the layout view of VCC. A common-centroid arrangement is used for the current mirror device M1 and M2.

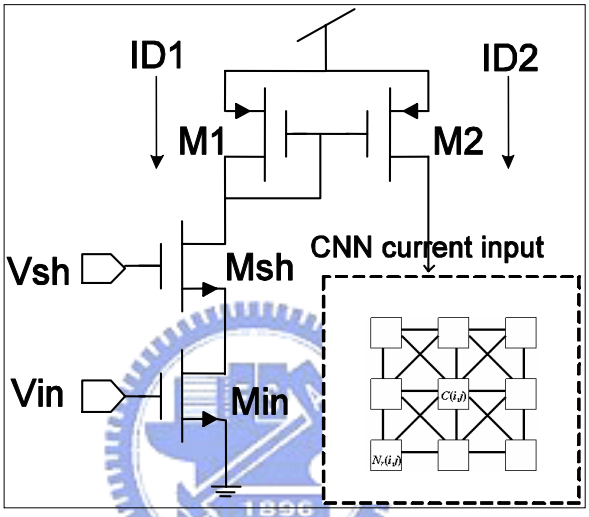


Fig. 30 Schematic of voltage to current converter.

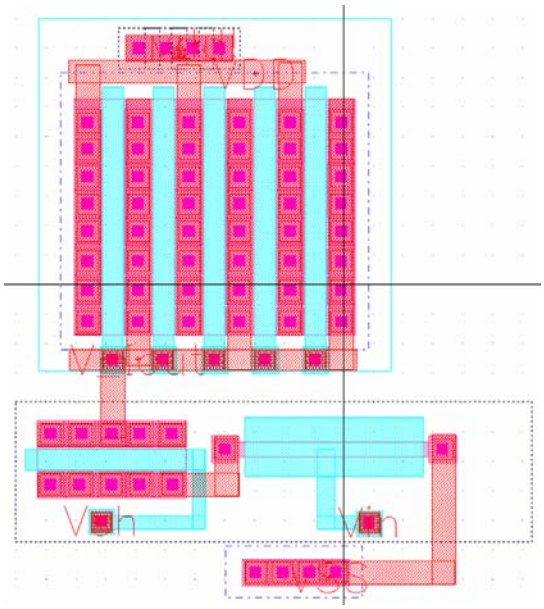


Fig. 31 Layout of voltage current converter.

Transistor M2 with Vsh (set to 2.85 volts) is used to shield switch noise from output stage and prevent LAM voltage from leakage. Analysis the current mirror of Fig. 30, we can write

$$I_{D1} = \frac{1}{2}\mu_p C_{ox}(\frac{W}{L})_1(V_{GS} - V_{TH})^2(1+\lambda V_{DS1}) \tag{4.1}$$

$$I_{D2} = \frac{1}{2}\mu_p C_{ox}(\frac{W}{L})_2(V_{GS} - V_{TH})^2(1+\lambda V_{DS2}) \tag{4.2}$$

$$\frac{I_{D2}}{I_{D1}} = \frac{(W/L)_2}{(W/L)_1}\cdot\frac{1+\lambda V_{DS2}}{1+\lambda V_{DS1}} \tag{4.3}$$

where λ is the channel length modulation coefficient.

Figure 32 is the VCC voltage to current transfer curve. With 50Ω output loading, the current will change from 0 to 140uA while Vin changes from 0 to 3.3 volts. Quite low sub-threshold current arises as Vin changes from 0 to 0.5 volts. This is because Vin is lower than the threshold voltage (V_{TH}) of Min transistor.

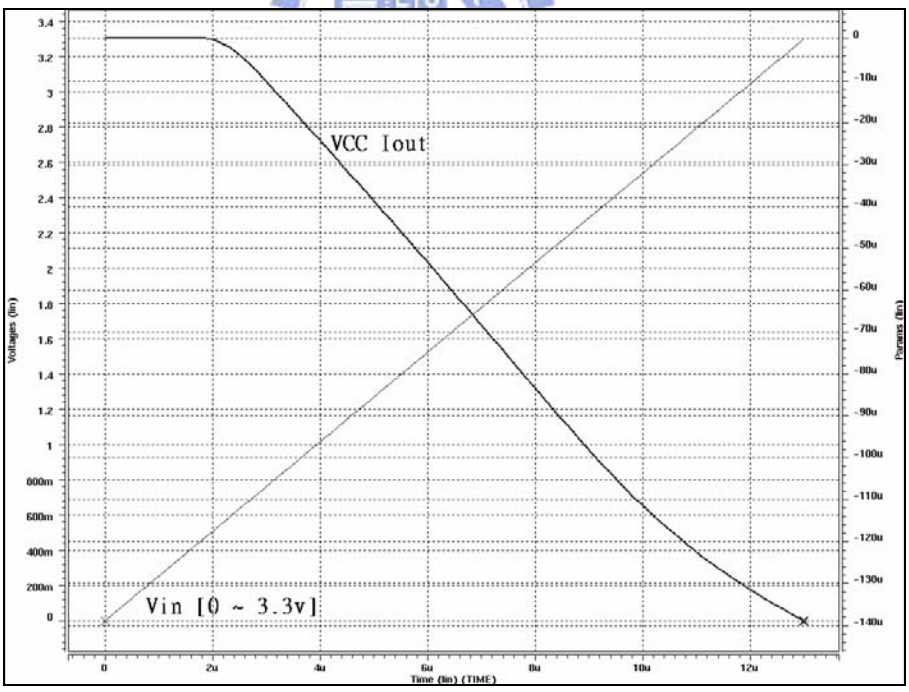


Fig. 32 VCC voltage to current transfer curve.

4.4 CNN Core and Template (A,B) Design

With the template defined in Eq. (3-7) and the simplify state equation list in Eq. (3-8), the ASCNN core with adaptive threshold template is designed as follows. Since $\dot{X} = -X + 2Y + u + I$, template A is set to $(W/L)_4 / (W/L)_3 = 2$ and other neighborhood connections are zero. Fig. 33 shows the schematic of CNN cell. The input current Iu from VCC and Bias current I are fed into the drain of M1. M5 and M6 form the output stage which is the diode connect common source amplify. The amplifier output is a robust binary value. A well-designed current mirror can generates 20uA for M1, M2 and M3 current source. M4 current source takes 40uA since the W/L is twice of M3. A switch signal Vac_CNN is used to control CNN operations. While the switch is turned on, the ASCNN works as a comparator that compares the current between Iu and Ibias. If the input current is higher than the bias current, the output will get logic '1' value and trigger the 5-bit bias circuit to raise the threshold, ie, bias current. The bias current is hold until the current is higher than input and logic '0' is obtained. The coordinate in the CNN array is considered as global minimum position.

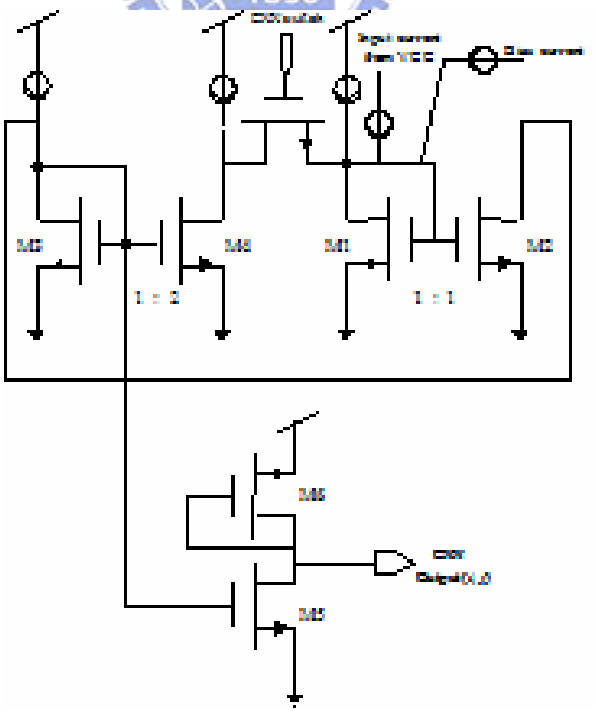


Fig. 33 Schematic of the CNN cell.

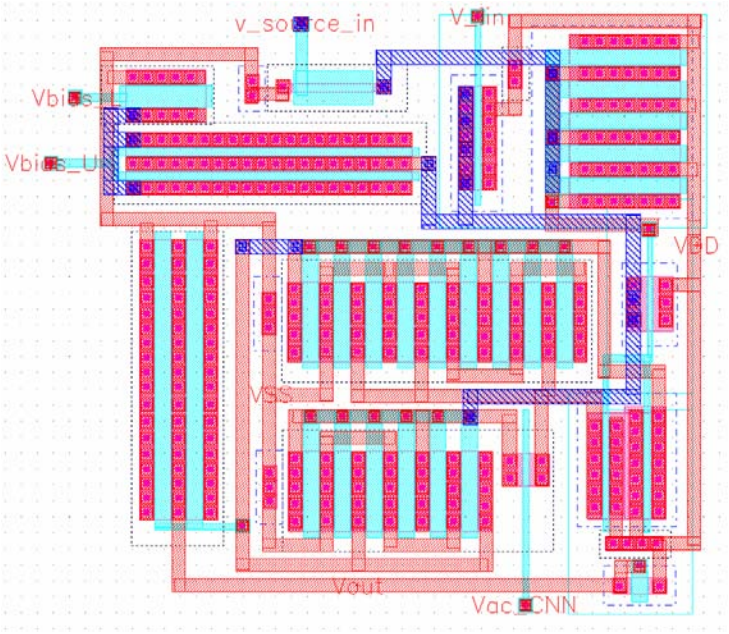


Fig. 34 Layout view of the CNN cell.

Figure 34 shows the layout of CNN core. Splitting the devices into parallel devices and interdigitizing them can distribute process gradients across both devices, thus improving matching.

4.5 Adaptive Bias Circuit

The structure of 5-bit current decision circuit is the same as input stage shown in Fig. 22. By using a 5-bit counter from digital control circuit, 32-level bias current is fed into CNN. Due to a low drain voltage in the M_3 transistor of the CNN cell, a cascade current source for lower minimum voltage across the current source is used and shown in Fig. 35 to provide an adaptive bias current.

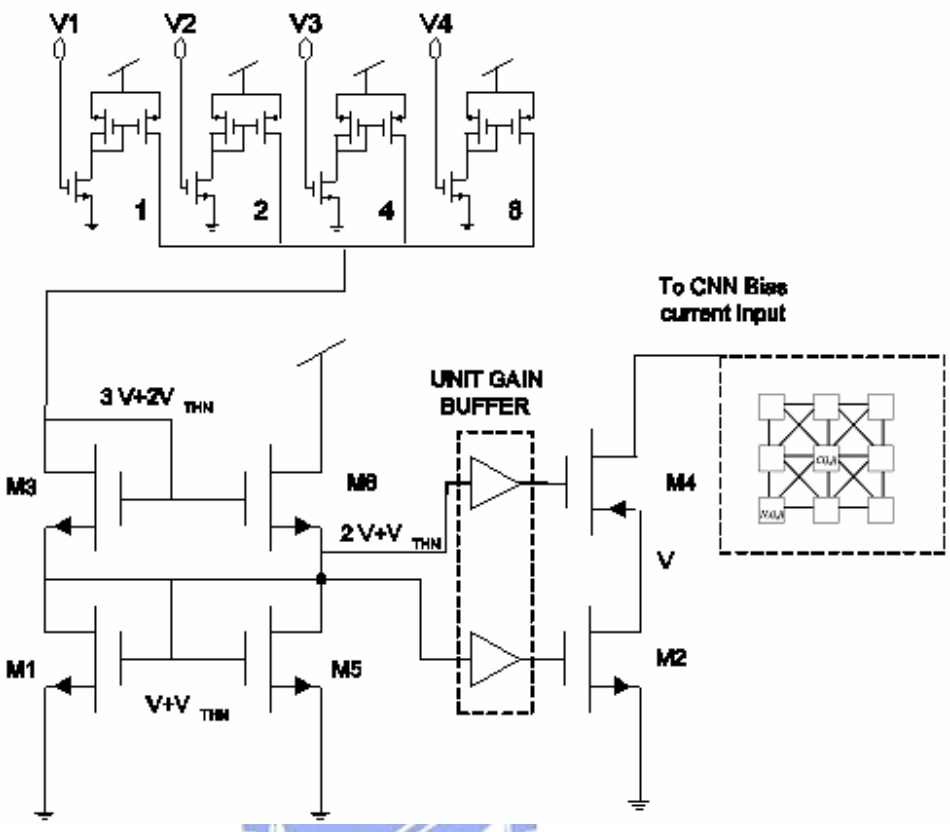


Fig. 35 5-bit cascade current mirror using a source follower level shifter.

We define the gate-source biasing voltage in terms of the excess gate-source voltage △V as $V_{GS}=\triangle V+V_{THN}$. The voltage on M_6 is use to drop the potential at the gate of M_4 down to $2\triangle V+V_{THN}$. This reduces the voltage on drain of M_4 to $2\triangle V$ before M_2 and M_4 enter the triode region. The MOSFET M_3 is resized to generate $3\triangle V+2V_{THN}$ to provide the gate voltage of M_4, i.e., $V_{GS3}=2\triangle V+V_{THN}$. The MOSFET with its gate and drain tied together being fed by a constant current (M_1 and M_3) behaves as a constant DC potential. While all the output of CNN array is high, the 5-bit bias decision circuit will change the bias current to a higher step and iterate until one of the CNN cell comes out with a low signal. For the enhanced ability of signal fanouts in Fig. 35, a unit gain buffer [33] is added to the output of the bias circuit as shown in Fig. 36. A common mode feedback (CMFB) network is added to sense the common mode (CM) level of the two outputs and accordingly adjust one of the bias currents in the amplifier. We can divide the task of CMFB into three operations: sensing the output CM level, comparison with reference, and returning the error to the amplifier's bias network. Fig. 37 shows the layout of the unit gain

buffer and Fig. 38 shows the layout of 5-bit bias control circuit. The output of the 5-bit control circuit should connect to the unit gain buffer before connecting to 475 CNN cells. The specification of the unit gain buffer is limited as Table 2.

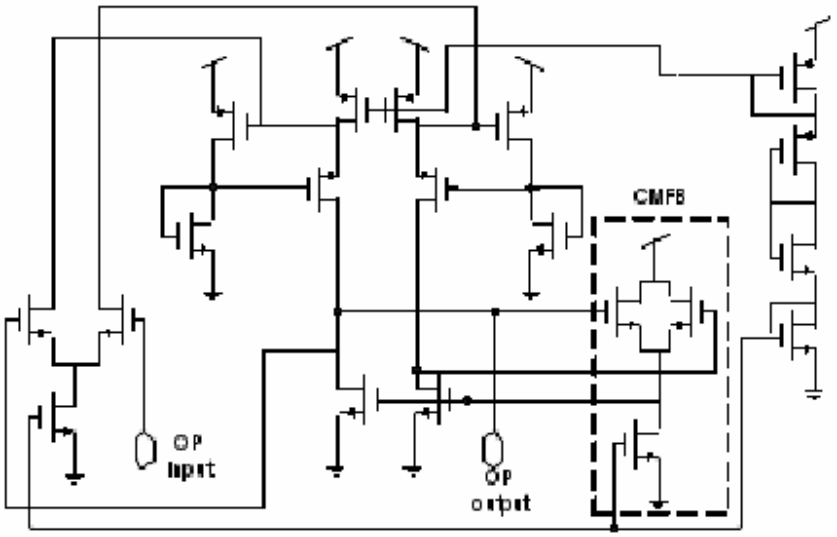


Fig. 36 Schematic of the unit gain buffer

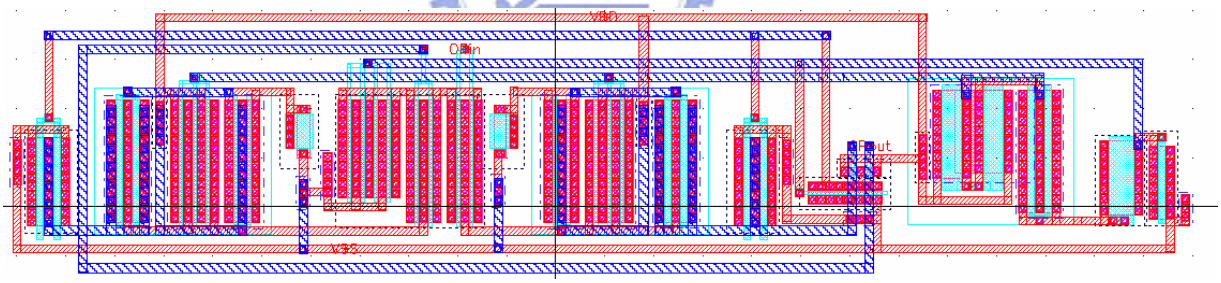


Fig. 37 Layout of the unit gain buffer.

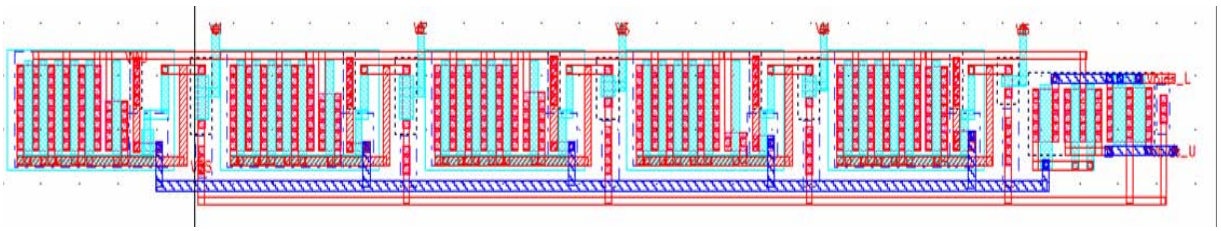


Fig. 38 Layout of 5-bit cascade current mirror using a source follower level shifter.

Table 3： Specification of Unit Gain Buffer.

Parameters	Performance
Open-loop gain	72.4dB
Common mode input range	2.45V
Common mode output level	1.5~1.7
Output swing	2.306V
Power supply	13.04mW (@ 3.3 volts)
Total current consumption	3.95mA
Low frequency PSRR	-3.66dB
High frequency PSRR	-3.35dB
DC CMRR	-30.4dB
High frequency CMRR	-16.5dB
Unit-gain bandwidth	197MHz
Phase margin	64
Slew rate	Rising 272V/us Falling 327/V/us

4.6 ASCNN Processing Unit

The ASCNN processing unit which consists of LAM, VCC, adaptive threshold template CNN and output connect AND gate is shown in Fig. 39. Each LAM switch in the processing array is turned on according to the Vx and Vy signal. The input current will feed into the designate LAM. Then VCC circuit will transform the SAD voltage into current and send into CNN core as input. The AND gates will take CNN output to the right and down cells and pass to the global output connect chain register. A digital decision circuit will raise the bias for all CNN cells if none of the global minimum position has been found.

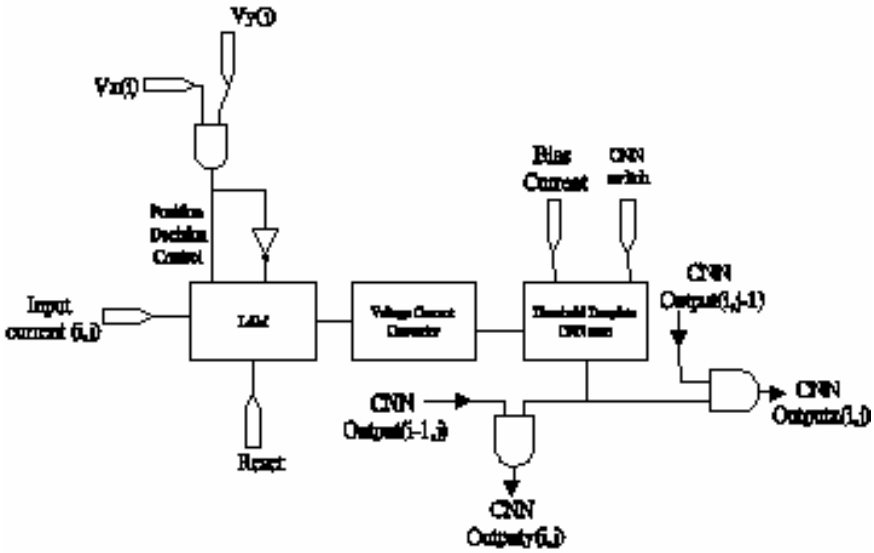


Fig. 39 Components of ASCNN processing unit.

The ASCNN processing unit including the CNN bias circuit is shown in Fig. 40. The circuit inside the rectangle is an ASCNN processing unit. The 5-bit bias circuit is connected to processing unit array by passing unit gain buffers. The CNN cell will compare the input current (1) and bias current (2) and give a binary output to the row and column global connected chains. Layout of processing unit is shown in Fig. 41. In order to overcome the mismatch during fabrication, symmetric layout skill is used. Fig. 41 (a) shows the odd number columns processing unit layout, and Fig. 41 (b) shows the even number columns processing unit layout. Therefore every column layout is mirrored from neighborhood. Power lines are drawing as nets, which are similar to cell based design to give every processing cells stable power supply. The signal lines are drawing in the same way, which result in convenient connections to neighbor cells. Fig. 42 shows the layout of ASCNN cells. The T shape main power line is used to provide power streams in each processing unit.

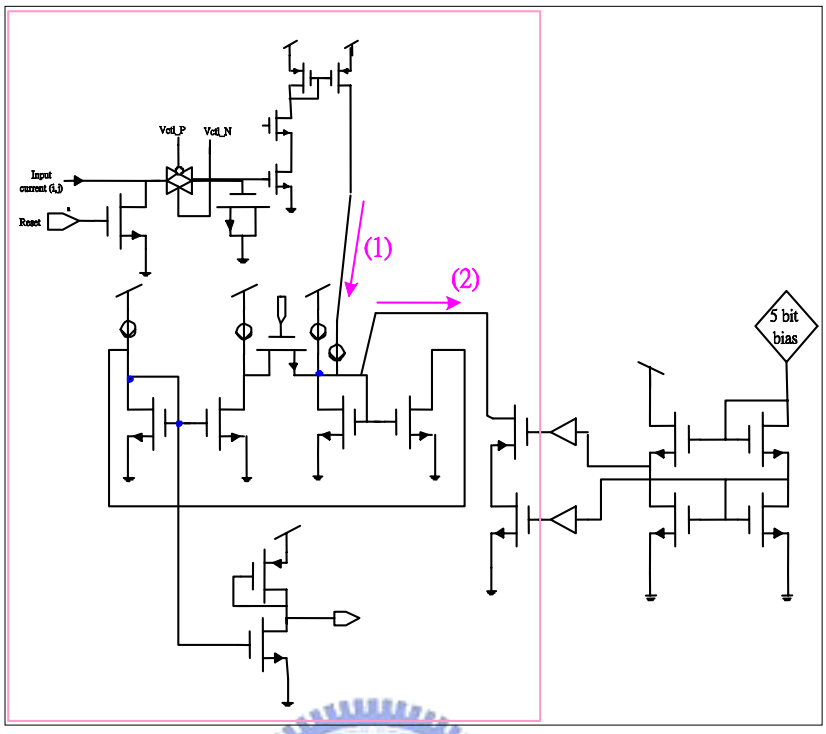


Fig. 40 Schematic of an ASCNN processing unit (inside the rectangle) with CNN bias circuit.

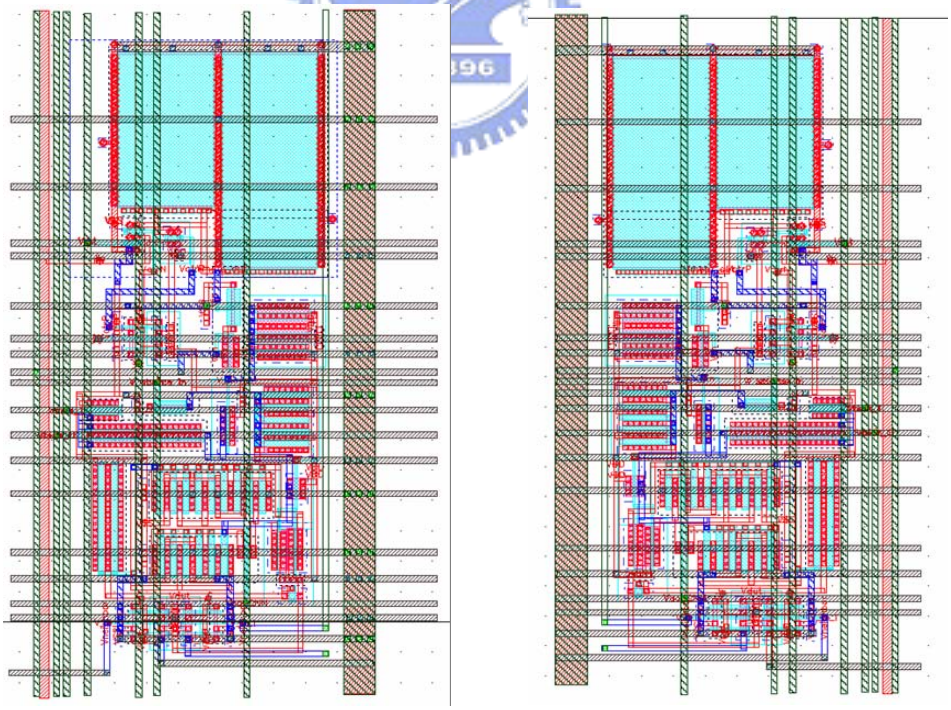
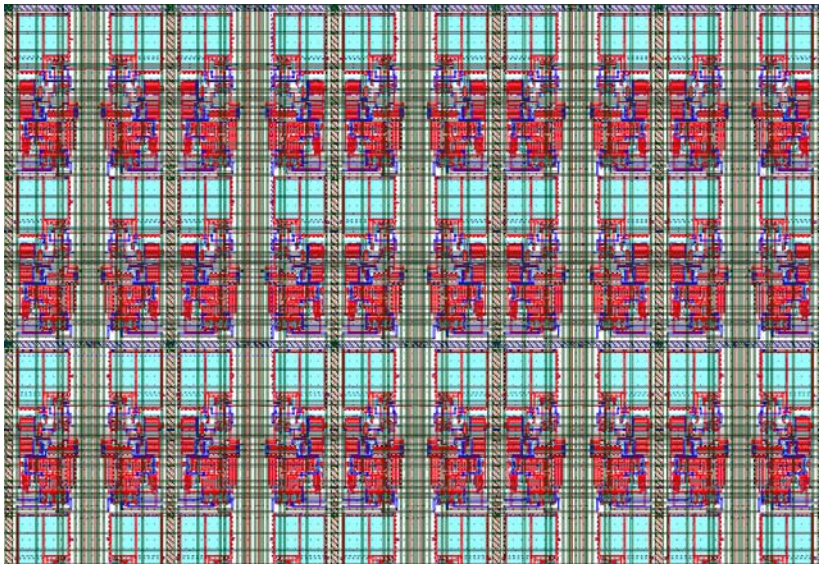
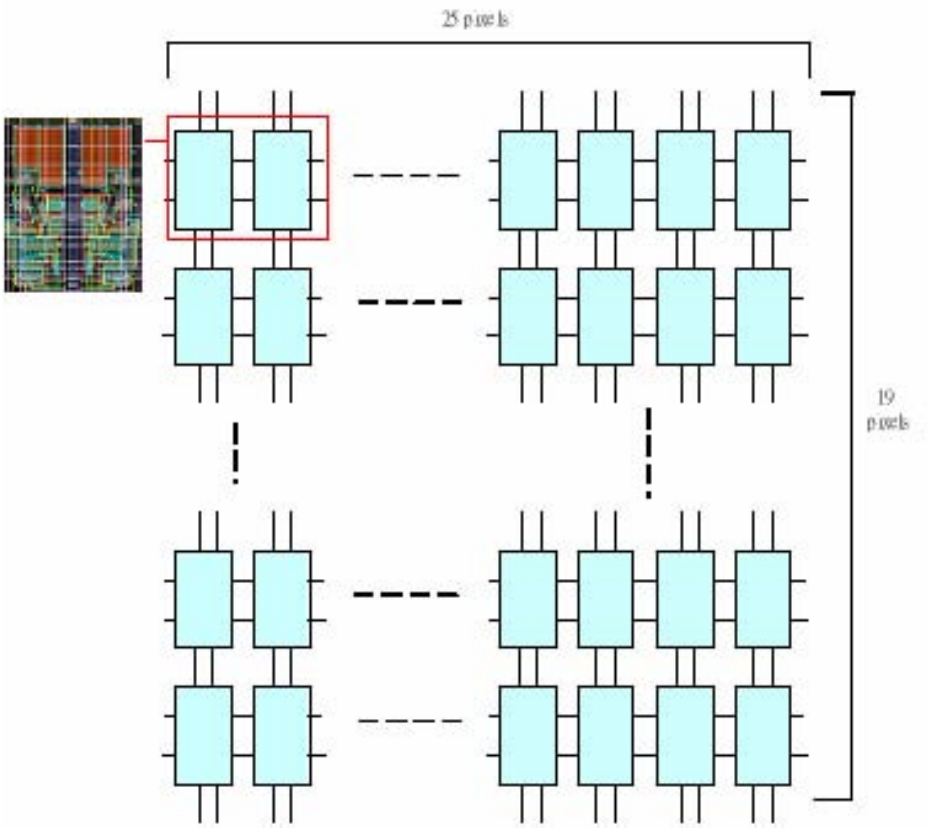


Fig. 41 Symmetric layout of ASCNN processing unit. (a) Odd column processing unit layout. (b) Even column processing layout.



(a)



(b)

Fig. 42 (a) Layout of ASCNN array (part view). (b) Connections of CNN cells.

4.7 Global Output Connected Chain

Global output connected chains as shown in Fig. 43 are designed as address decoder. There are 19 row chains and 25 column chains in the processing array (19 x 25 pixels). Since the CNN cell can translate the analog signal input into binary output after processing, AND gate chains work as the output stage for each processing unit. If one of the input current is lower than bias current, the flip output signal (logic 0) will directly pass to the output registers with the connection of neighbor cells (right and down) for every processing unit.

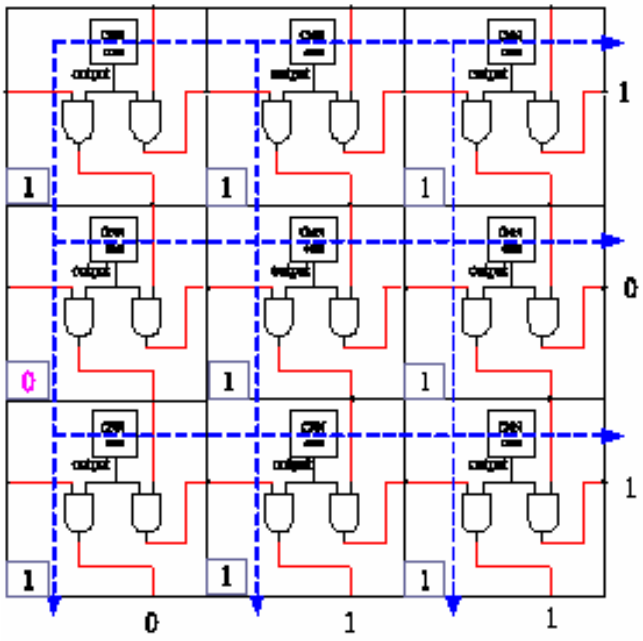


Fig. 43 Structure of global output connect chain (3 x 3 array).

4.8 ASCNN Controller

For the CNN active circuit, there are two operations such as loading and searching minimum. A digital control circuit in the mixed-signal ASCNN chip is used as a controller to operate whole ASCNN function. The finite stage machine (FSM) of the control circuit in Fig. 45 contains eight states to control the motion estimation processor. The state machine performs loading image into LAM, switching bias steps, and decoding LMV position.

- ***Loading*:** After coming of the start signal, initial signal will reset the LAM voltage and

load the SAD information sequentially into LAM. Since the parasitic capacitance of the processing array will also share the input charge current, a reset signal is needed before changing to next storage LAM cell. A pre-charge voltage (about 0.6 volts) for each MOS capacitor is also required to pass the nonlinear region as mentioned in *section 4-2*. Due to pixel-by-pixel input, excepting pre-charge all capacitors of LAM in the first time, the total of loading operations is 30 times for each location. The procedure of loading SAD information into each LAM cell location is shown in Fig. 44. After the 475 (19 x 25) LAM have loaded the SAD information, a finish signal will trigger the CNN on/off switches and CNN current source switches turn on and jump to searching global minimum state. During the loading step, the switches are turned off for saving power.

- ***Searching minimum*:** An adaptive-minimized threshold template is used for searching the minimum value in the CNN array as mentioned in *Section 3-5 and Section 4-7*. The digital control circuit will check the global output connect chains of each rows and columns during every clock cycle. Two sets of 5-bit registers are used to decode and store the outputs of rows and columns global connect chains. If all the outputs remain logic 1, registers sent all zeros and command the bias counter rising to upper bit. The process will stop until one row and column flip their outputs to logic 0 and register will decode the address to output pad. In order to skip the worst case that all of the current input is higher than 32 levels bias current, the registers will sent all one (11111) to output pad and suggest the motion vector in the prescribed region is not dependable. A restart signal will make the searching minimum state jumping to initial state and again reset all the LAM before loading new SAD information.
- ***Working frequency:*** By taking advantage of the computing power of the CNN array, a single chip can be used to process a complete video frame by operating on a fraction of the image at a time. A frame rate of 40 Hz—adequate for high-quality video applications [47] —represents each frame processing time should take less than 25ms. Using working frequency at 20MHz on 19 times 25 ASCNN chip should be capable of fining 4 LMV at each image frame less than 6.2 ms and will discuss this in later chapters.

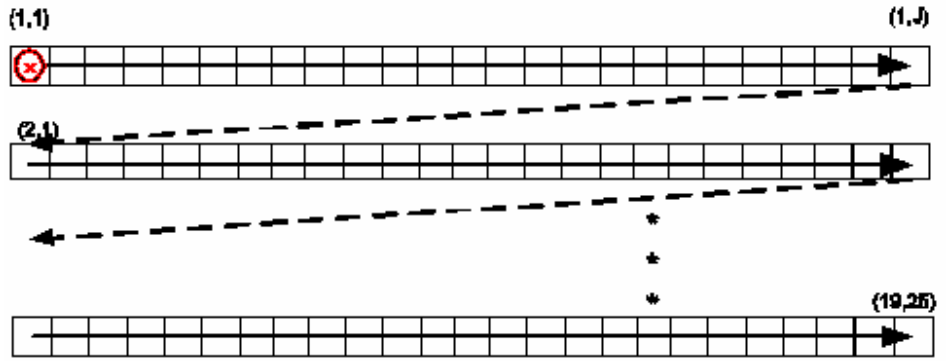


Fig. 44 Loading SAD information into each LAM location.

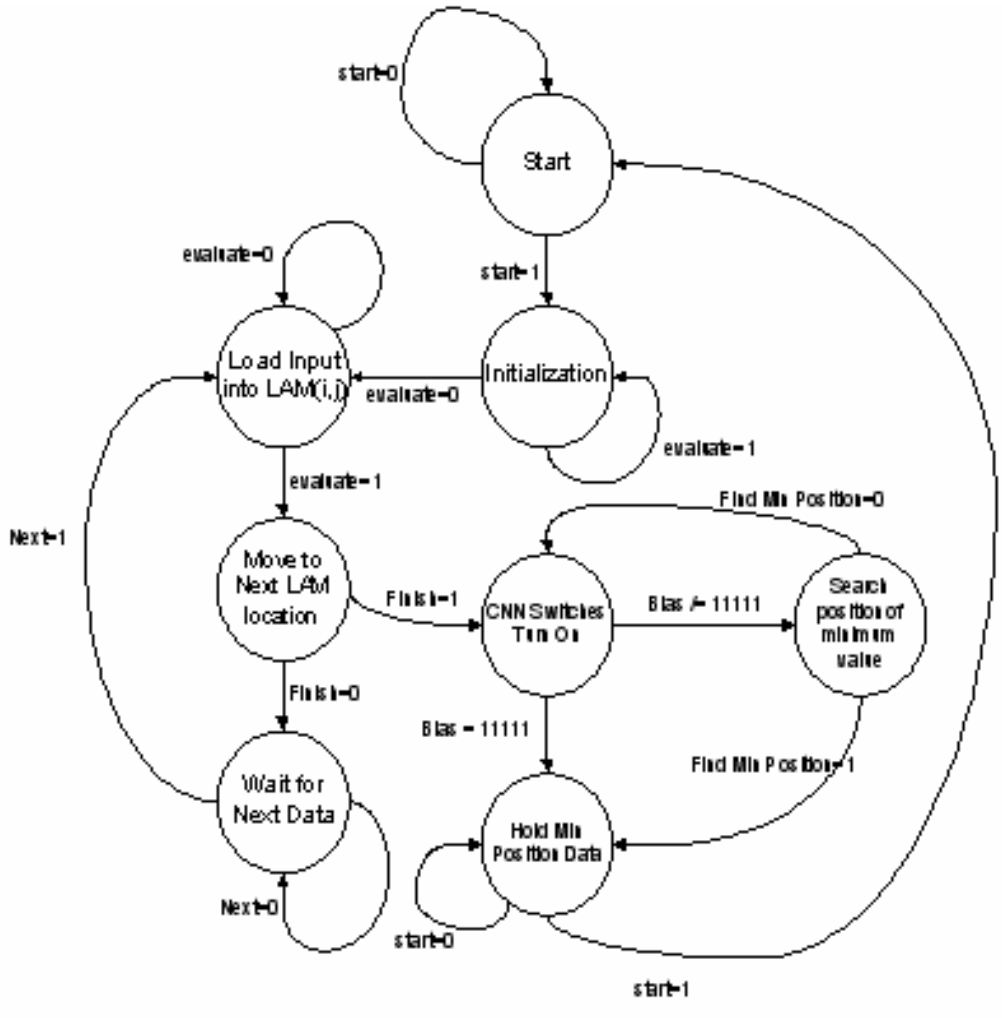


Fig. 45 Complete FSM for the ASCNN controller design.

CHAPTER 5

SIMULATION RESULTS

Based on the CNN-based local motion estimation algorithm, we can both verify this method using CNNUM and the transistor-level simulation. Fluctuated video sequences with various irregular conditions are used for testing. Each video sequence contains 200 frames with the resolution of 312×200.

5.1 Simulation Results of CNNUM

The CNN Universal Machine (CNNUM) [48] provides the framework for the definition of an algorithmically programmable analog array computer with supercomputer power on a chip. Its character is using a mixed mode on-chip analog and logic (analogic) stored program [21] to realize highly complex image processing tasks. CNNUM is advantageous in terms of power consumption and computation speed as compared to these digital counterparts [49]. A 64×64 array CNN processor [50] shown in Fig. 49 has been used to simulate the proposed motion estimation algorithm. A practical method of using 64×64 pixels processor is to make the input image operate onto fractions and process each fraction at a time. With the size of 19×25 pixels processing frames defined in chapter 2 have stored in CNNUM memories, the sub-image subtraction and addition can be done by using inversion, first and second addition templates.

0	0	0
0	-1	0
0	0	0

Template A

0	0	0
0	-2	0
0	0	0

Template B

$$\begin{pmatrix} STATE = 0, \quad INPUT = \textit{Static grayscale image } \boldsymbol{P}, \\ \text{bias=0}, \\ OUTPUT = -\boldsymbol{P} \end{pmatrix}$$

Fig. 46 Inversion template.

0	0	0
0	-1	0
0	0	0

Template A

0	0	0
0	1	0
0	0	0

Template B

$$STATE = 0,\ INPUT = \text{Static grayscale image } \boldsymbol{P},$$
$$bias = \text{Static grayscale image } \boldsymbol{Q},$$
$$OUTPUT = 0.5\times(\boldsymbol{P}+\boldsymbol{Q})$$

Fig. 47 First addition template.

0	0	0
0	-0.25	0
0	0	0

0	0	0
0	2.5	0
0	0	0

Template B

$$STATE = 0,\ INPUT = 0.5\ (\boldsymbol{P}+\boldsymbol{Q}),$$
$$bias=0,$$
$$OUTPUT = \boldsymbol{P}+\boldsymbol{Q}$$

Fig. 48 Second addition template.

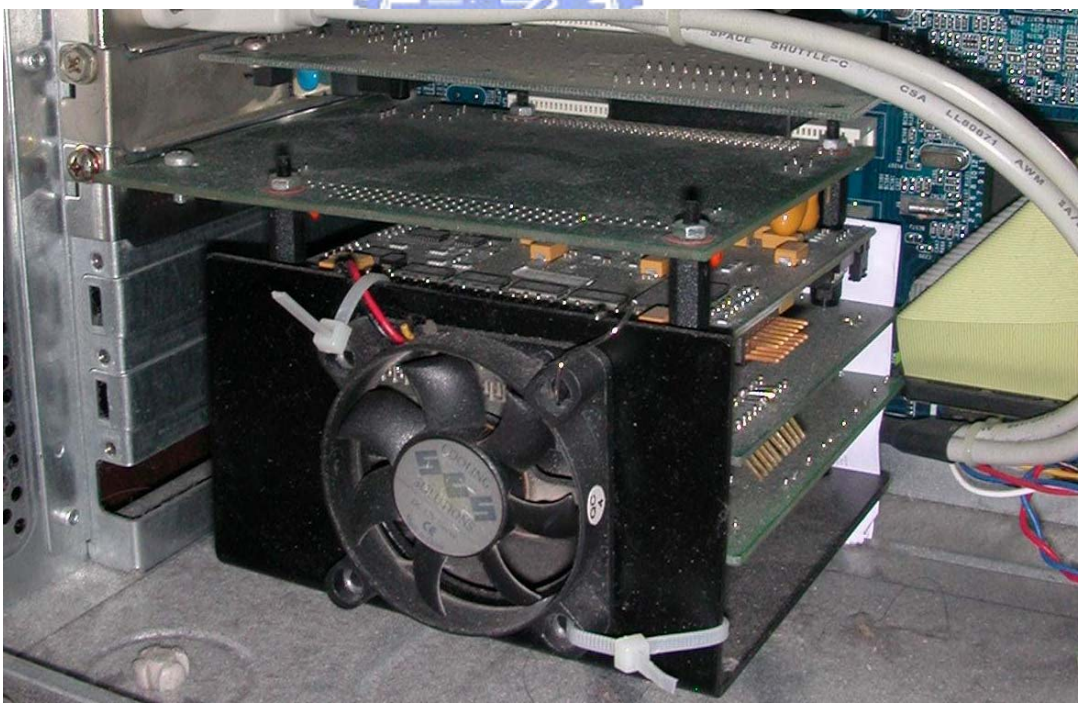


Fig. 49 Chip sets of CNNUM

The result of the SAD array (19×25 pixels) after CNNUM processing is shown in Fig. 51 (a). After analysis these SAD values, the minimum difference value is 85 pixel counts and locates in (6,4) while the top left point is defined as (1,1).

Since the location of the designate motion vector (LMV) is located in the minimum SAD value position, the adaptive threshold as shown in Fig. 50 is applied to CNNUM. With proper choice of threshold bias current Ibias, only the minimum position appear with white color and others remain black color. The final result is shown in Fig. 51 (b). Since the threshold value is artificial, the result is time consuming.

0	0	0
0	2	0
0	0	0

Template A

0	0	0
0	1	0
0	0	0

Template B

$STATE = 0$, $INPUT =$ *Static grayscale image* $\boldsymbol{P}$,

$I_{bias} = -Z$, where $-1 < Z < 1$

$OUTPUT =$ Binary image where black pixels correspond to pixels in with grayscale intensity $P_{ij} > Z$

Fig. 50 Adaptive threshold template.

There are two ways to decode the global minimum address. One make use of the DSP processor of CNNUM to search the white color position. The other make use of shifting template. To compare the position information with the simulation of circuit, the vertical and horizontal shifting templates as follows are used, and their output is shown in Fig. 51 (c) and (d).

The vertical shifting template:

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 2 & 0 \\ 0 & -1 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \text{Ibias=0} \tag{5-1}$$

The horizontal shifting template:

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 2 & -1 \\ 0 & 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \text{Ibias=0} \tag{5-2}$$

Figure 51 (c) is the result after applying vertical shifting template to Fig. 51 (b). Vertical template will propagate the white color position until it saturate at boundary. Fig. 51 (d) is the result after applying horizontal shifting template. With the definition of the top left point in Fig. 11(b) as (1,1), the correct coordinate (X, Y) should be (6,4) and the position is the same as shown in Fig. 51 (a).

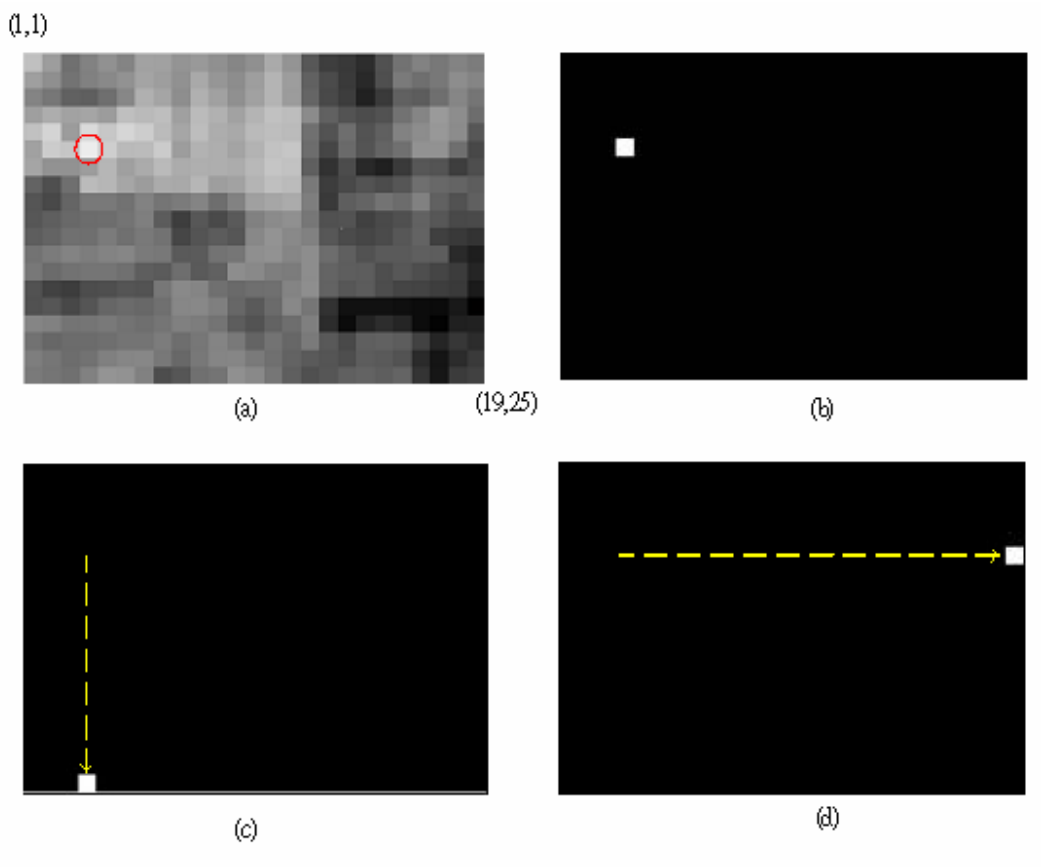


Fig. 51 (a) The output of 19×25 difference value array image. (b) The output of (a) is fed into CNNUM using adaptive minimized threshold template with Eq. (7). (c) Vertical output of (b) with eq. (5-1). (d) Horizontal output of (b) with eq. (5-2).

5.2 Chip Implementation

The section contains the description of mixed-signal IC design flow, explanation of post simulation results by using HSPICE and ModelSim, whole chip layout and testing consideration.

5.2.1 Design flow

The idea of the proposed DIS algorithm has been simulated by Matlab. Then, we study the feasibility on combining the CNN technology into DIS motion estimation to reduce the computation load. The next step is to implement the motion estimation chip with ASCNN design.

ASCNN design follows the flow chart of Fig. 52. The chip design is divided into analog and digital part. Full-custom and cell-based design indicate the CNN processing units and the CNN controller, respectively.

Fully custom design flow as follows:

(1) Transistor level circuit design and pre-simulation by using HSPICE and SPECTRE

(2) Drawing layout by using Cadence Virtuso and post-layout verification with Calibre..

(3) Post- Simulation with cell based circuit design.

Cell-based design flow as follows:

(1) CNN control behavior model code design.

(2) CNN controller RTL code design.

(3) Logic Synthesis by using design complier and mixed-signal layout integrated with SOC Encounter.

(4) Post-layout verification with Calibre .

(5) Post- Simulation with fully custom circuit design.

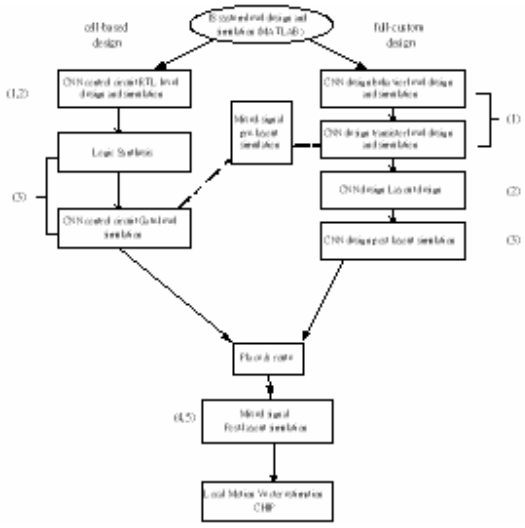


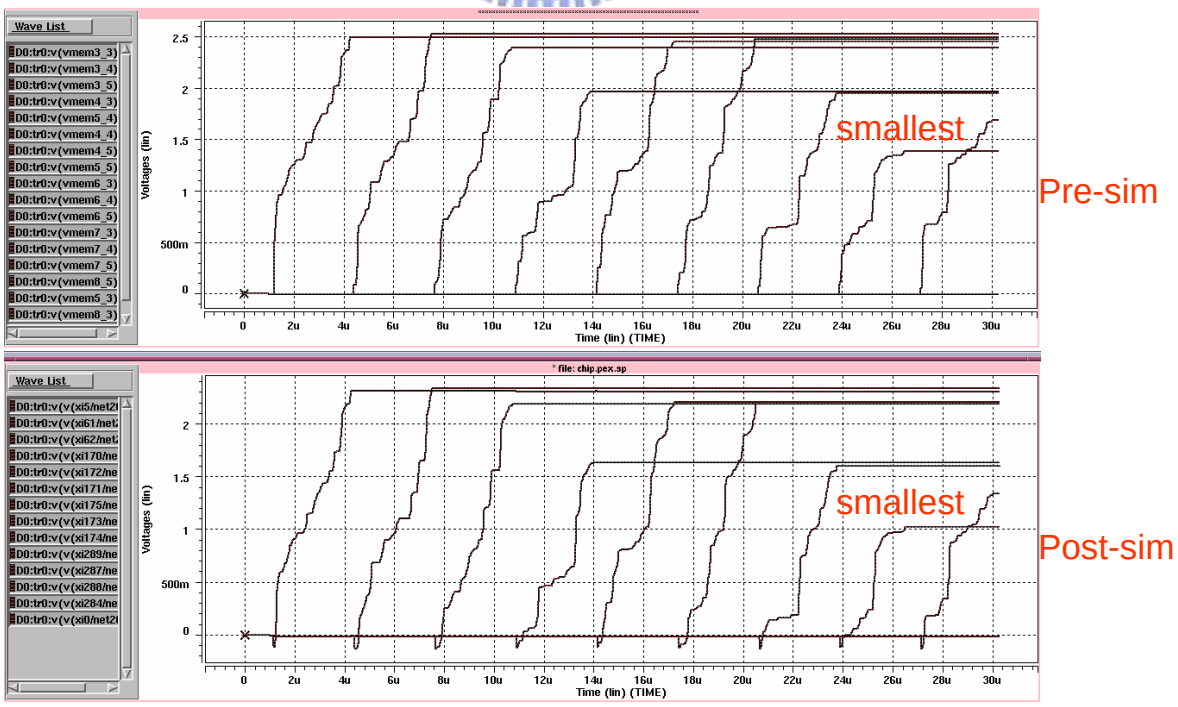
Fig. 52 ASCNN IC design flow.

5.2.2 Simulation Results

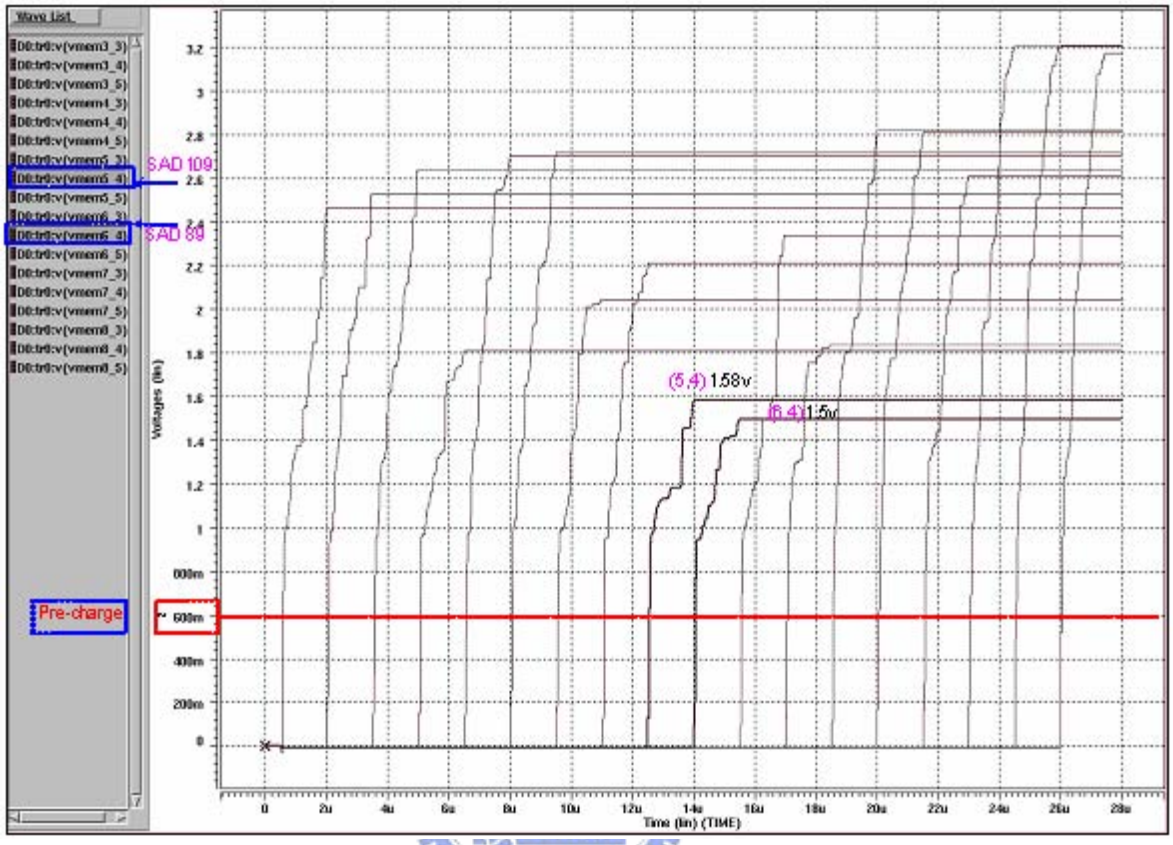
The SAD result shown in Fig. 51(a) is loaded into LAM by 8-bit DAC. We mark the rectangular region (Fig. 55(a)) which contains the minimum SAD position to compare the LMV result with CNNUM. The rectangular contains 18 cells of LAM, and each output shown in Fig. 53 is labeled as Vmem(x,y)**.** As can be seen, the smallest voltage is 1.5 volts in the position (6,4) and the second is 1.58 volts in position (5,4) which is exactly the same as analyzing from Matlab. The 1.5 volts is the result of accumulating 89 pixel differences form a specific position of 30 sub-regions and 1.58 volt is the result of accumulating 109 pixel differences. The operating frequency of the chip is 20MHz. We use the frequency to load the pixel differences information in sequence. The total loading time (including pre-charge operation) needs 65 clock cycles. Since each pixel takes 2 clock cycles to load into LAM, 30 sub-regions cost 60 clock cycles. Each cell has to spend 100ns (2 clock cycles) pre-charge time to reach 0.6 volts. The rest of 3 clock cycles are the rest time for parasitic capacitance in the circuits. Thus, loading a set of LAM will take 3.25 us and for 19 times 25 LAM array totally cost 1543.75us.

For a complete searching global minimum position as discussed in Fig. 45, each threshold level needs 2 clock cycles before rising to higher level. This is because CNN switches should turn on after bias signal VB heaving its value. For a worst case that none of the input current is lower than bias current, total processing time include loading LAM information cost 1.55ms ($1543.75us + 100ns \times 32 = 1546.95us$). Each image sequence processing time is less than 6.2ms. The number of searching global minimum clock cycles in using CNN design has superior computational effect than using DSP processor. For a 19 times 25 array, DSP processor takes $19 \times 25 \times 30 = 14250$ loading clock cycles for a 312 times 200 pixels video frame and the proposed ASCNN chip takes $19 \times 25 \times 31 = 14725$ loading clock cycles which include pre-charge clock cycles for LAM. But a DSP at least takes $19 \times 25 \times 2 = 950$ clock cycles, comparing and storing, to find the global minimum position while the CNN with adaptive threshold template only takes 32 clock cycles in worst case. With the aid of global output connect chains, ASCNN chip does not need extra clock cycles to decode the LMV coordinates, but DSP processor does. As mentioned above, the result saves at least $19 \times 25 \times (32 - 31) - 32 = 443$ clock cycles compared with other DSPs. The larger the incoming frame size, the more clock cycles it saves in using ASCNN chip to find LMV.

Figure 53(a) shows the pre-simulation and post-simulation results of difference values voltage stored in LAM. Due to the existence of parasitic capacitors, the charging capacitance value in LAM is higher than that without parasitic effect. Compared with these two results in Fig. 53(a), we can find that the smallest voltage position is the same. Fig. 53(b) is the LAM cells post simulation results which locate in the region that marked in Fig. 55(a). The region contains 18 pixels and the global minimum point is included in. Because CNN bias circuit provides 32 levels threshold and the reliable LAM voltage output swing is from 0.65 volt to 3.15 volt. Therefore the change rate of the 5-bit bias circuit is 0.08 volts for each step which is equal to 24 pixels [0.08/3.3/(256×4)] resolution. Thus the global minimum position can be found by using 32 levels bias circuit in Fig. 53 (b). Due to the bias circuit, the changeable bias current from high to low can be found in the CNN array. The variation of bias current is shown in Fig. 54 which ranges form 0 to 20.15uA. When the minimum value is found, the output voltage of the related cell is low. Through the location decoder, the LMV position can be recognized and latched to the output register. The minimum SAD position is exactly the same as running by CNNUM. Moreover, the lower SAD value usually happens near the lowest SAD position and can also be verified in Fig. 53(b). Thus, if the difference pixel values are less than 24, the output decoder will choose the closer to original position's address. But the global minimum position is still near the correct position if the address captured in registers is not correct.



(a)



(b)

Fig. 53 (a) Pre-simulation and post-simulation results of the difference values which were stored in the LAM. (b) Post-simulation of the difference values which are stored in the prescribed rectangular LAM region. The 0.6 volts pre-charge voltage for each LAM cell is needed to pass the nonlinear region of the MOS capacitor.

Figure 55(a) is the output of 19×25 difference value array image. Fig. 55 (b) and (c) shows the row and column global connect chains output corresponding to the minimum SAD position. The simulation only checks the 18 LAM (contained in the dot line) value and CNN switches turns on at 78us. With the help of automatic tuning bias current circuit, the output flips from high to low with a proper threshold of CNN. Fig. 55 (d) shows other global connect chains for the rest of 42 outputs.

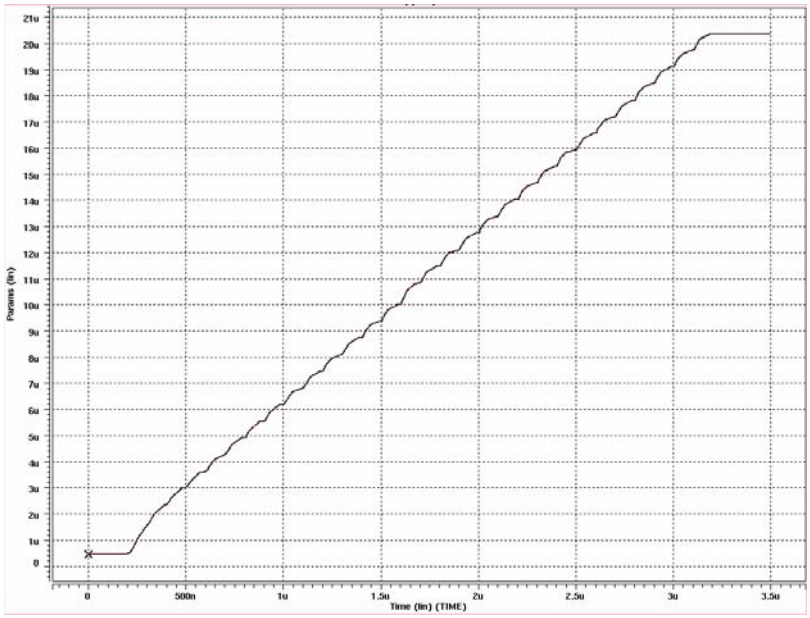


Fig. 54　32 levels CNN bias current which ranges from 0uA to 20.15uA.

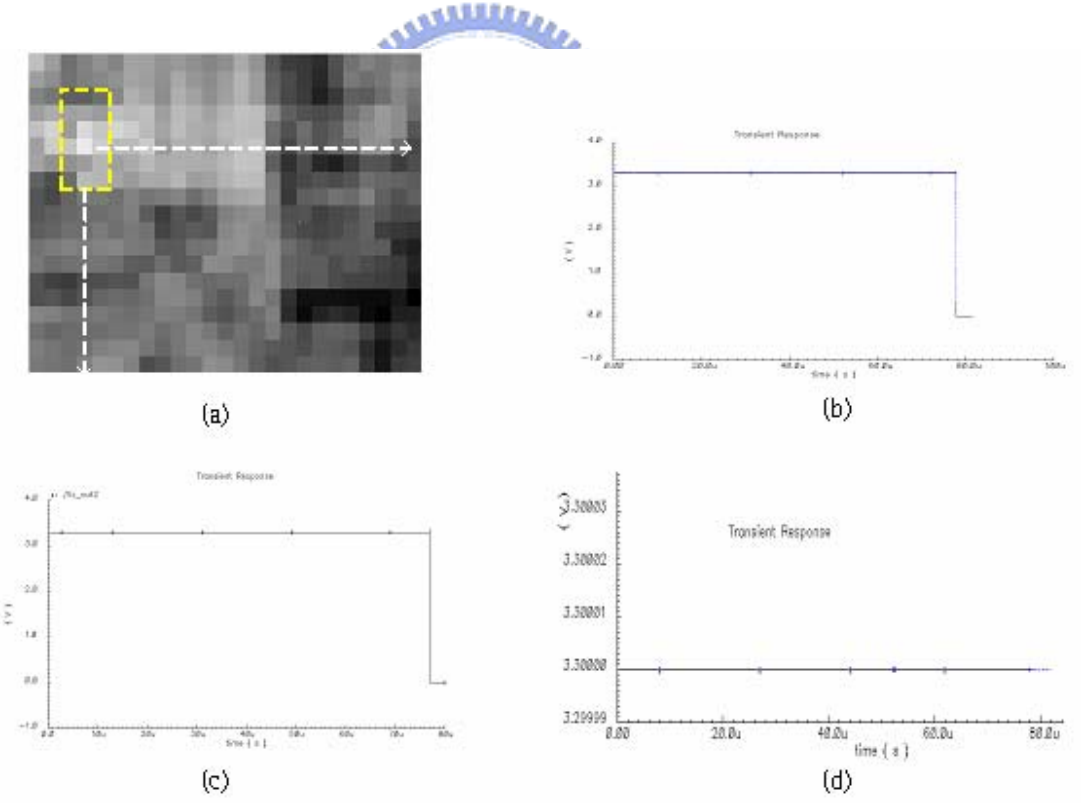
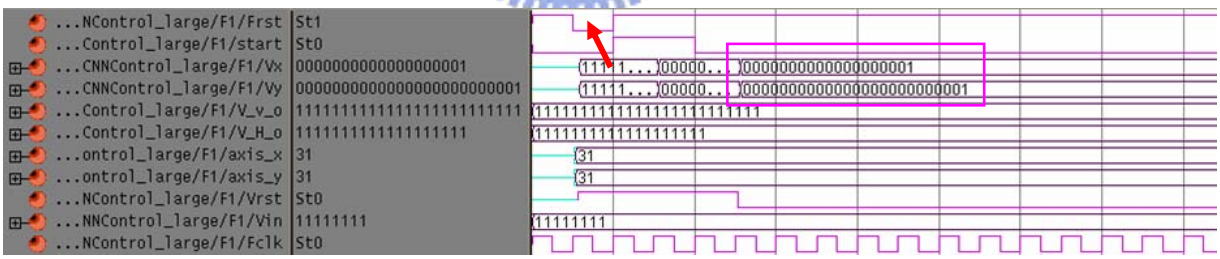
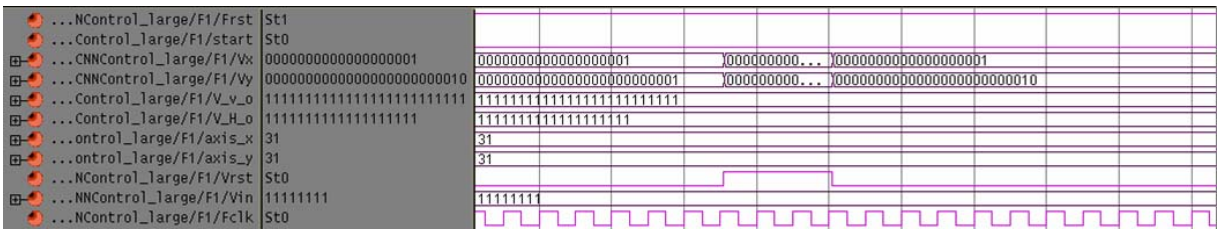


Fig. 55　Simulation of CNN output inside the dot line. (a) The output of 19×25 difference value array image.(b) (c) row and column global connect chains output corresponding to minimum SAD position. (d) global connect chains for the rest of 42 outputs.

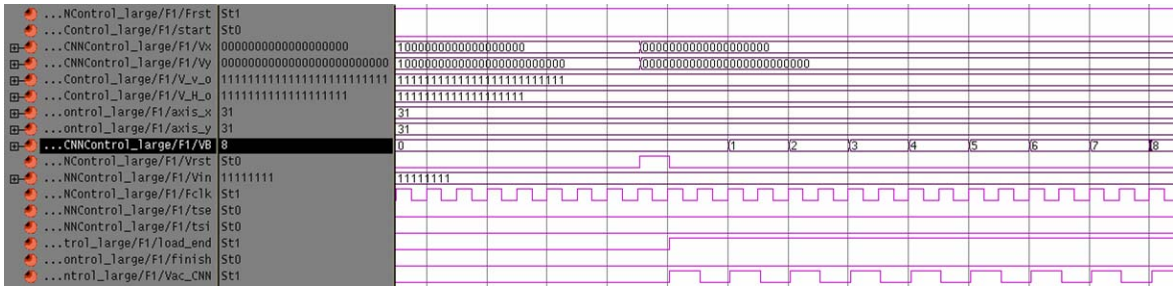
Figure 56 shows the simulation results of CNN control circuit from Modelsim. Fig. 56 (a) shows the control signals in initial state. The clock (Fclk) frequency is 20MHz. With a given reset signal (Frst), a Vrst signal will discharge all voltage stored in LAM and parasitic capacitance. The loading procedure as discussed in Fig. 44 is implemented with Vx (19-bit) and Vy (25-bits). The circuit use the Vx (19-bits) and Vy (25-bits) signal to address a position and turn on the input switch of the designate LAM cell. SAD information from 30 sub-regions of the specific position will then store in the memory. Fig. 56 (b) shows the control signals in the loading state. After loading a set SAD information into LAM, new position will be assigned by Vx and Vy according to the loading procedure. Fig. 56 (c) shows the control signals of the CNN active state. After 475 (19×25) LAM have stored SAD information from the 8-bit D/A converter, the CNN switches are turned on to check the global minimum position of the array. VB is the bias current control counter and V_H_o (19-bit) and V_v_o (25-bit) are the output from 19 rows and 25 columns global connected chains. If none of the CNN input is lower than the threshold level, VB will raise to the higher level. Fig. 56 (d) shows that the minimum position (6,4) has been found at VB equals to 23 and store the information in the registers axis_x (5) and axis_y (3). After the LMV position has accepted outside the chip, signal “start” will be changed to high and new SAD information is again sent into LAM as Fig. 56 (a).



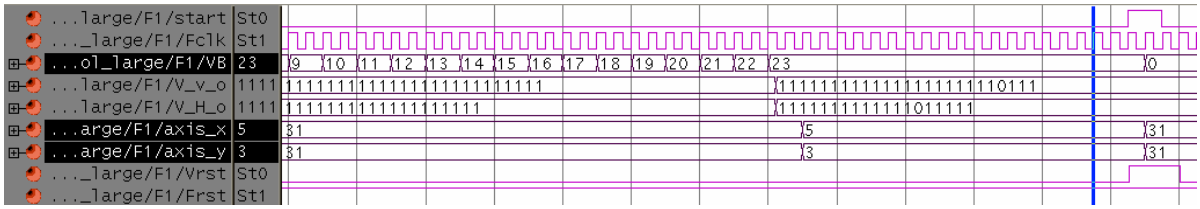
(a)



(b)



(c)



(d)

Fig. 56 Simulation results of the CNN control circuit.(a) the initial state. (b) the loading state. (c) the CNN active state. (d) the global minimum position by raising VB.

Figure 57 shows the overall CNN output simulation from 19 rows 25 columns global output connected chains. The storage time takes 1543.75us after all SAD value have loaded into LAM. Note that the row No.6 and column No.4 flip from logic 1 to logic 0 while the CNN bias counter (VB) raise up to 23 at 1565us as shown in Fig. 56 (d). Only the minimum value location recorded in the positional registers axis_x and axis_y.

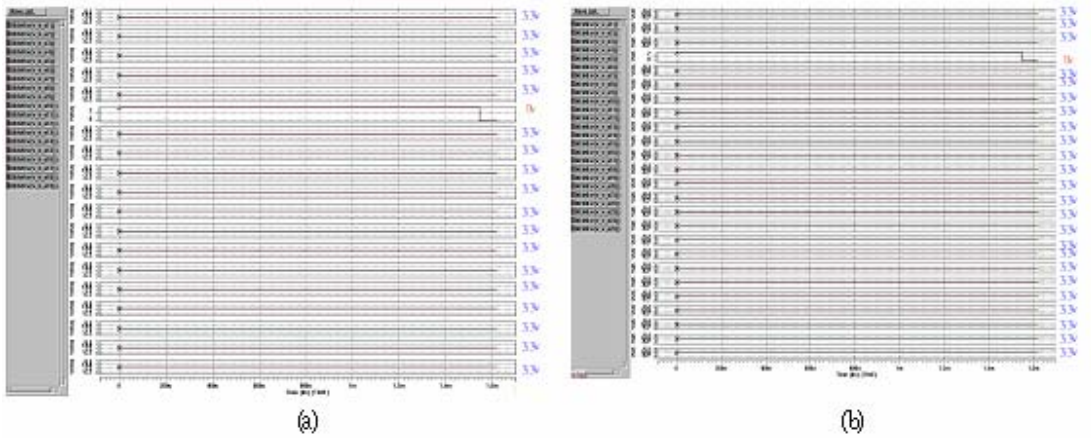
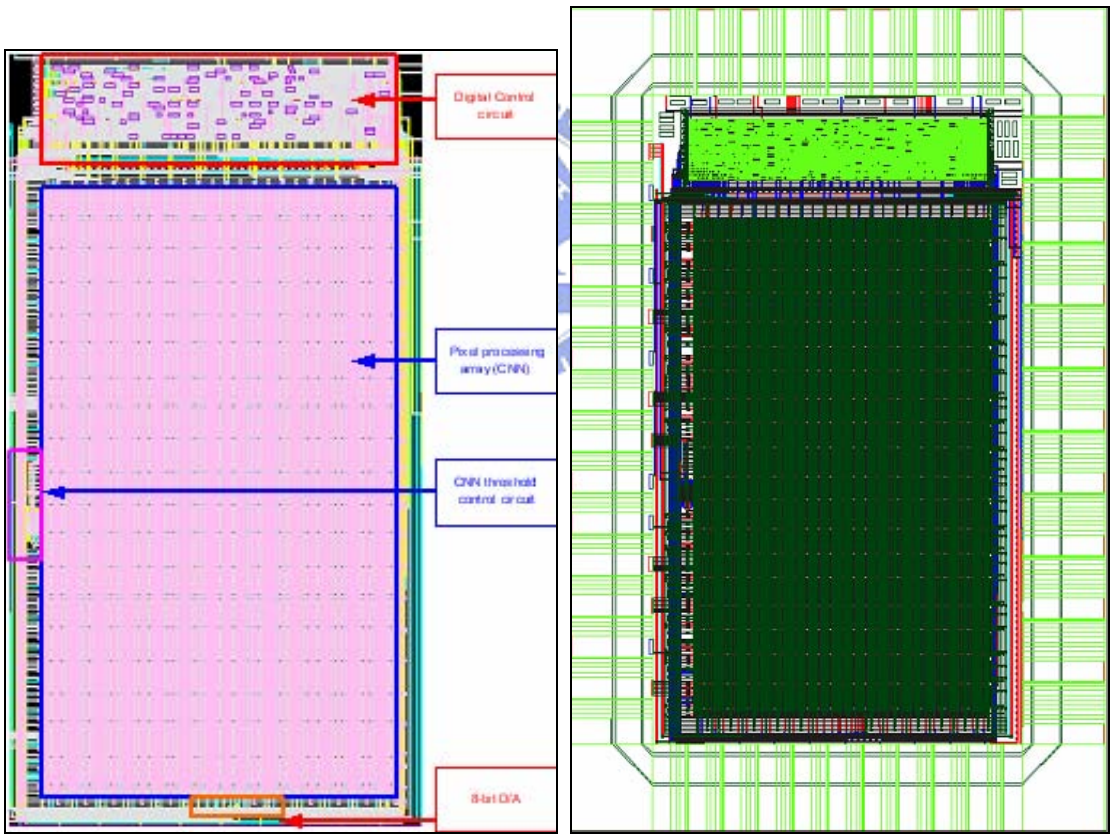


Fig. 57 The overall simulation for the 19 times 25 array. (a) 25 columns of global output connect chains. (b) 19 rows of global output connect chains.

5.2.3 Chip Specification and Layout

Figure 58 is the whole chip layout of mixed-signal ASCNN chip and its specification is listed in Table 3. The upper part is the CNN control circuit which is designed by cell-based design flow and the central 19 times 25 pixels array is pixel processing units as discussed in *Section 4-6*. The 8-bit D/A converter and CNN threshold control circuit are indicated in Fig. 58(a). Fig. 58(b) shows chip layout with 48 bonding pads. A guard ring is drawing around the fully custom design in order to shield analog signals from digital signals. There are 3 sets of analog power for the processing array and one set of digital power for the digital control circuit. Chip information is shown in APPENDIX A which includes the specification of chip, I/O pad information, bounding wire layout, and Caliber DRC LVS verification results.



(a) (b)

Fig. 58 Layout of mixed-signal ASCNN chip. (a) Topology of ASCNN chip. (b) ASCNN chip layout with 48 bonding pads.

Table 3 Specification of ASCNN chip

Item	Specification (unit)	measure
VDD	3.3 V	N/A
System Clock	20 MHz	Yes
Local Analog Memory	0.5 v ~ 3.3 v	Yes
CNN bias	550 mV ~ 900 mV	Yes
Power Consumption	170mW (average) 335mW (worst case)	Yes
Die size	2.375×3.442 um^2	N/A
Package	SB 48	N/A
Max Freq.	95.476 MHz (cell-Based design)	PostSim

5.2.4 Testing consideration

The testing methods could be separated in two parts: digital CNN control circuit testing and fully custom ASCNN circuit testing. The package will first test by IMS testing machine to check the function of CNN control circuit. With the design of scan chain in digital circuit, the errors could simply feed the ATPG test pattern which generated by Tetra-max into chip testing pad (itsi) and analysis the output pattern for output pad (oY_tso). The testing pad information is discussed in Appendix A.

Then, testing image generated by Logic Analyzer could feed into LAM by passing on chip 8-bit DAC pad (iV8~1). Wait until all the LAM have loaded the image SAD information, testing pads (oY_mem1_1、oY_mem19_1、oY_mem25_1、oY_mem19_25) which represent the 4 corner LAM storage voltage could be analyzed and check if the memories voltage could raise as prediction.

The CNN threshold control circuit could check the testing pad (oY_biasL) voltage and see if the bias voltage can active after all LAM have loaded information.

The pixel processing array could be check by feeding simple test pattern which only contain a extremely small pixel value and see if the CNN global connect output chain could mark the minimum position.

◎ Digital Part： Add a scan chain by using DFT compiler. The test patterns have 119 sets in total and the fault and test coverage are 98.72 % and 100 %, respectively.

```
//      Uncollapsed Stuck Fault Summary Report
// -----------------------------------------------
// fault class                     code   #faults
// ------------------------------  ----  ---------
// Detected                         DT       4095
// Possibly detected                PT          0
// Undetectable                     UD         53
// ATPG untestable                  AU          0
// Not detected                     ND          0
// -----------------------------------------------
// total faults                              4148
// test coverage                           100.00%
// fault coverage                           98.72%
// -----------------------------------------------
```

Fig. 59 ATPG reports fromTetra-max.

◎ Analog Part：

1. LAM test: with a flat image pattern loaded by passing 8-bit DAC into LAM as shown in Fig. 60, the testing pad could help to analysis the properties of LAM functions to see if they can charge successfully and check the mismatch due to fabrication.
2. CNN function test: feed a simple testing pattern that makes only one position’s CNN input is smaller than others. The bias current should quickly distinguish the global

minimum coordinate.

3. Bias circuit test: feed the input with a flat but large SAD iamge and let all the pixel prcessing units cannot catch the minimuim position. Then test the bias testing point voltage and see if the volgate has reach the predic value. If the bias is out of the predic range due to differnet fabrication environment, the pad iVsh can be tunned to change the VCC input current to the CNN to feed the need of threshold level.

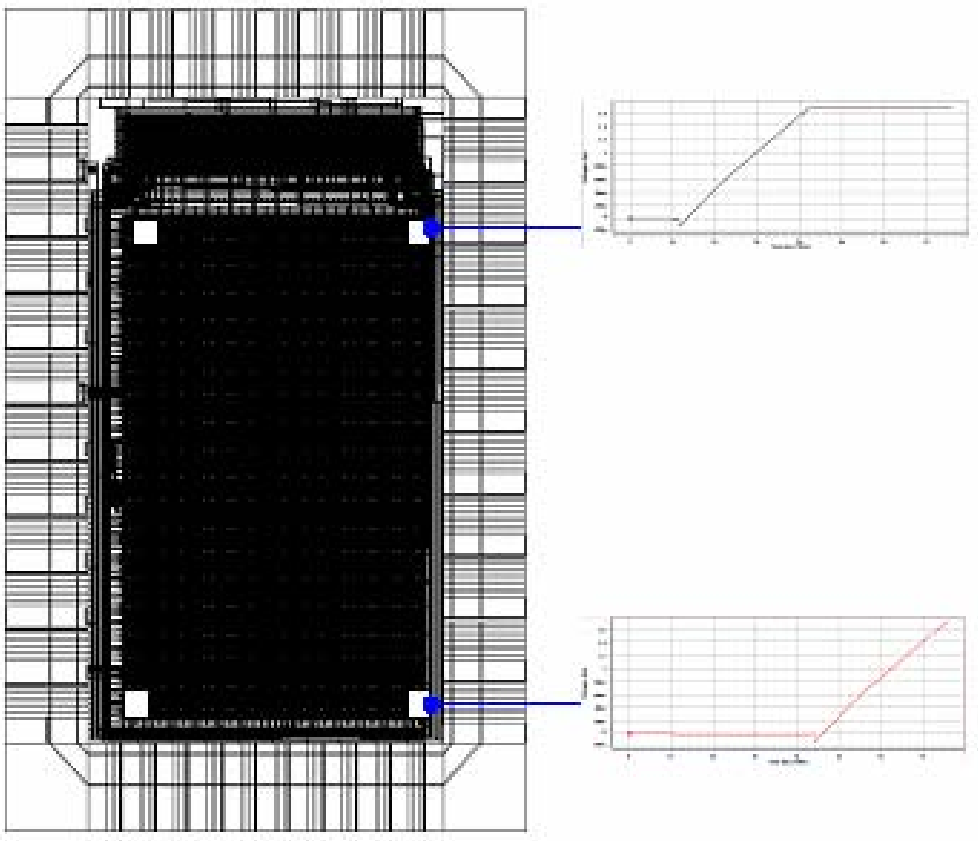


Fig. 60 Digital and analog part testing consideration of ASCNN chip.

CHAPTER 6

CONCLUSIONS AND FUTURE WORKS

In this thesis we propose the local motion estimation chip based on CNN architecture for image stabilization and combine the motion compensation method to construct the image stabilization system. To obtain reliable LMVs from the video sequence captured in various conditions and to reduce the computational complexity in finding candidate of motion vectors, these are two challenges for an image stabilization system. According to the simulation results, the proposed technique demonstrates the remarkable performance in both quantitative and qualitative (human vision) evaluations compared with the existence of approaches. The ASCNN chip is implemented for real-time video stabilization applications.

In particular, CNN is used to reduce computational complexity occurring in motion estimation and the ASCNN chip has superior performance compared with DSP processors. An 8-bit D/A converter is designed to transform digital input into analog current and stored in LAM. LAM is designed to store SAD values and the memory cell is made of MOS capacitance. Various SAD voltage stored in LAM can produce input current of CNN by the VCC circuit. An adaptive and adjustable bias current circuit is combined with CNN to automatically increase the threshold level and detect the global minimal SAD position. The global output connected chains which take the CNN connection properties are used to decode LMV position without wasting more processing cycles in searching address. The proposed CNN-based method can be implemented on VLSI, and CNNUM is able to perform the idea of the proposed design. Results with mixed-signal simulation based on TSMC 0.35μm 2P4M process have demonstrated the superior functionality of the designed circuit.

For the chip design or DIS implementation in the future, several possible improved directions will be considered as follows.

1. The power consumption of CNN array should be reduced to fit the most consumer electronics. Reduce the supply current of CNN seems to be the best way to solve this problem since adaptive threshold template only takes the stable value from CNN output instead of transfer curve. Power management design in the chip can also save more power during LAM switches

turned on.

2. A programmable CNN template is used instead of a fixed CNN template. The programmable method can reuse CNN architecture and perform more CNN functions. Dual clock design can also be used to speed up processing efficiency. The faster clock is used for CNN array in searching global minimum position. The other clock is used for new LAM which has shorter loading time.

3. To further distinguish the SAD value into fewer pixels, more digits should be used to control bias signals, and a judgment control circuit should also be used to skip the raising clock cycle if the bias current is much lower than input current.

REFERENCES

[1] M. Sekine, T. Kondou, and H. Hirose, "Motion Vector Detecting System for Video Image Stabilizers," *Proc. of IEEE International Conference on Consumer Electronics*, pp. 268-269, Jun. 1994.

[2] http://www.personaltechpipeline.com/news/59100088

[3] http://www.canon.com/technology/dv/02.html

[4] D. W. Holder and W. R. Philips, "Electronic Image Stabilization," U.S. Patent no.4637571, Jan. 20, 1987.

[5] M. Oshima, "VHS camcorder with electronic image stabilizer," *IEEE Trans. on Consumer Electronics*, vol. 35, pp. 749-758, Nov. 1989.

[6] Joon ki Paik, Yong Chul Park, and Dong Wook Kim, "An adaptive motion decision system for digital image stabilizer based on edge pattern matching," *IEEE Transactions on Consumer Electronics*, vol. 38, pp. 607-616, Aug. 1992.

[7] Jyh-Yeong Chang, Wen-Feng Hu, Mu-Huo Cheng, and Bo-Sen Chang, "Digital image translational and rotational motion stabilization using optical flow technique," *IEEE Transactions on Consumer Electronics*, vol. 48, pp. 108-115, Feb. 2002.

[8] Sung-Jea Ko, Sung-Hee Lee, Seung-Won Jeon, and Eui-Sung Kang, "Fast digital image stabilizer based on Gray-coded bit-plane matching," *IEEE Transactions on Consumer Electronics*, vol. 45, pp. 598-603, Aug. 1999.

[9] http://www.itri.org.tw/index.jsp

[10] M. Irani, B. Rousso, and S. Peleg, "Recovery of ego-motion using image stabilization," *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, Seattle, WA, pp. 454-460, Jun. 1994.

[11] Kwon, R. Chellappa, and C. Morimoto, "Motion compensated sub-band coding of video acquired from a moving platform," *Proc. of IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, pp. 2185-2188, Jan. 1995.

[12] C. Morimoto, D. DeMenthon, L. Davis, R. Chellappa, and R. Nelson, "Detection of independently moving objects in passive video," *Proc. of Intelligent Vehicles Workshop*, pp. 270-275, Sep. 1995.

[13] K. Uornori, A. Morimura, H. Ishii, T. Sakaguchi, and Y. Kitamura, "Automatic image stabilizing system by full-digital signal processing," *IEEE Trans. on Consumer Electronics*, vol. 36, pp. 510-519, Aug. 1990.

[14] J. K. Paik, Y. C. Park, and D. W. Kim, "An adaptive motion decision system for digital image stabilizer based on edge pattern matching," *IEEE Trans*. on C*onsumer Electronics*, vol. 38, pp. 607-615, Aug. 1992.

[15] T. Kinugasa, N. Yamamoto, H. Komatsu, S. Takase, and T. Imaide, "Electronic image stabilizer for video camera use," *IEEE Trans. on Consumer Electronics*, vol. 36, pp. 520-525, Aug. 1990.

[16] S. J. KO, S. H. Lee, and K. H. Lee, "Digital image stabilizing algorithms based on bit-plane matching," *IEEE Trans. on Consumer Electronics*, vol. 44, pp. 617-622, Aug. 1998.

[17] Y. Egusa, “An application of fuzzy set theory for an electronic video camera image stabilizer,” *IEEE Trans. on Fuzzy Systems*, vol. 3, pp. 351-356, Aug. 1995.

[18] J. K. Paik, Y. C. Park, and D. W. Kim, “An adaptive motion decision system for digital image stabilizer based on edge pattern matching,” *IEEE Trans. Consumer Electronics*, vol. 38, pp. 607-616, Aug. 1992.

[19] S. W. Jeon, “Fast digital image stabilizer based on Gray-coded bit-plane matching,” *IEEE Trans. Consumer Electronics*, vol. 45, pp. 598-603, Aug. 1999.

[20] L. O. Chua and L. Yang, “Cellular neural networks: theory and application,” *IEEE Trans. on Circuits and Systems*, vol. 35, pp. 1257-1290, Oct. 1988.

[21] R. Carmona Galan, A. Rodriguez Vazquez, S. Espejo Meana, R. Dominguez-Castro, T. Roska, T, Kozek, and L. O. Chua, “An 0.5-μm CMOS analog random access memory chip for TeraOPS speed multimedia video processing,” *IEEE Transactions on Multimedia* , vol.1, pp. 121-135, Jun. 1999.

[22] L. O. Chua and T. Roska, “The CNN paradigm,” *IEEE Transaction on Circuits and System, I*, vol. 40, pp. 147-156, Mar. 1993.

[23] S. Espejo, A. Rodr´ıguez-V´azquez, and R. Dom´ınguez-Castro, “Smartpixel cellular neural networks in analog current-mode CMOS technology,” *IEEE J. Solid-State Circuits*, vol. 29, pp. 895-905, Aug. 1994.

[24] R. Dom´ınguez-Castro, S. Espejo, A. Rodr´ıguez-V´azquez, R. Carmona, P. Foldesy, A. Zar´andy, P. Szolgay, T. Sziranyi, and T. Roska, “A 0.8 um CMOS programmable mixed-signal focal-plane array processor with on-chip binary imaging and instructions storage,” *IEEE J. Solid-State Circuits*, vol. 32, pp. 1013-1026, Jul. 1997.

[25] S. Espejo, R. Dom´ınguez-Castro, G. Li˜n´an, and A. Rodr´ıguez-V´azquez, “64 X 64 CNN universal chip with analog and digital I/O,” in *IEEE1998 Inter. Conference Electronic Circuits and Systems*, pp. 203-206, Sep. 1998.

[26] T. Roska and L. O. Chua, “The CNN universal machine: An analogic array computer,” *IEEE Trans. Circuits Syst. II*, vol. 40, pp. 163-173, Mar. 1993.

[27] J. E. Varrients, “A current-mode cellular neural network implementation,” *IEEE Trans. on Circuits and Systems-II*, vol. 40, pp. 147-155, Mar. 1993.

[28] S. Erturk, “Image sequence stabilization: motion vector integration (MVI) versus frame position smoothing (FPS),” *Proc. of the 2nd International Symposium on Image and Signal Processing and Analysis*, pp. 266-271, 2001.

[29] M. K. Gullu and S. Erturk, “Fuzzy image sequence stabilization,” *Electronics Letters*, vol. 39, no. 16, pp. 1170-1172, Aug. 2003.

[30] M. K. Gullu, E. Yaman, and S. Erturk, “Image sequence stabilization using fuzzy adaptive Kalman filtering,” *Electronics Letters*, vol. 39, pp. 429-431, Mar. 2003.

[31] M. K. Gullu and S. Erturk, “Membership function adaptive fuzzy filter for image sequence stabilization,” *IEEE Trans. Consumer Electronics*, vol. 50, pp. 1-7, Feb. 2004.

[32] C. T. Lin, S. C. Hsu, and S. F. Liang, “A robust digital image stabilization technique based on inverse triangle method and background detection,” *IEEE Trans. on Circuits and Systems-I*, submitted paper for publication, 2004.

[33] Ealgoo Kim, Daeyun Shim, Seongik Jang, Gyudong Kim, and Wonchan Kim, “A global motion characterization method using one-dimensional characteristic curves for image

processing applications," *IEEE International Symposium on Circuits and Systems, ISCAS '97*, vol. 2, pp. 1393-1396, Jun. 1997.

[34] M. M. Sondhi, "New methods of pitch extraction," *IEEE Trans. Audio Electroacoustics*, vol. AU-16, pp. 262-266, Jun. 1968.

[35] Bing J. Sheu and Joongho Choi, Neural Information Processing and VLSI, KLUWER ACADEMIC PUBLISHERS, Ch.5, pp. 97-149, 1995.

[36] Bing J. Sheu and Joongho Choi, Neural Information Processing and VLSI, KLUWER ACADEMIC PUBLISHERS, Ch. 14, pp. 397-452, 1995.

[37] N. H. Farhat, "Optoelectronic neural networks and learning machines," *IEEE Circuits and Device Magazine*, pp. 32-41, Sep. 1989.

[38] G. A. Carpenter, S. Grossberg, and D. B. Rosen, "ART 2-A: an adaptive resonance algorithm for rapid category learning and recognition," *Neural Networks*, vol.4, 1991.

[39] J. S. Kane and M. J. Paquin, "POPART: partial optical implementation of adaptive resonance theory 2," *IEEE Trans. on Neural Networks*, vol. 4, pp. 695-702, Jul. 1993.

[40] T. Roska and L. O. Chua, "Cellular neural networks with nonlinear and delay-type template elements," *Proc. IEEE int. Workshop on Cellular Neural Networks and Their Applications*, pp. 12-25, 1990.

[41] A. Rodriguez-Vazquez, S. Espejo, R. Dominguez-Castron, Huertas, L. J., and E. Sanchez-Sinencio, "Current-mode techniques for the implementation of continuous- and discrete-time cellular neural networks," *IEEE Trans. on Circuits and Systems*, Part II, vol.40, pp. 132-146, Mar. 1993.

[42] J. E. Varrientos, E. Sanchez-Sinencio, and J. Ramirez-Angulo, "A current-mode cellular neural network implementation," *IEEE Trans. on Circuits and Systems*, part II, vol.40, pp. 147-155, Mar.1993.

[43] Hyongsuk Kim, T. Roska, H. Son, and I. Petras, "Analog addition/subtraction on the CNN-UM chip with short-time superimposition of input signals," *IEEE Trans. on Circuits and systems- I*: Fundamental Theory and Applications, vol.50, Mar. 2003.

[44] "CNN software library-version 1.1," Analogic Computing LTD.

[45] http://www.mxeduc.org.tw/SiliconAwards/2004/5thSiliconAward.htm

[46] C. Y. Wu, P. H. Lu, C. H. Lee, and M. K. Tsai**,** "New fully differential HF CMOS op amps with efficient common mode feedback," *IEEE International Symposium on Circuits and Systems*, vol. 3, pp. 2076-2079, May, 1989.

[47] C. A. Poynton, A Technical Introduction to Digital Video. New York: Wiley, 1996.

[48] T. Roska and L. O. Chua, "The CNN universal machine: An analogic array computer," *IEEE Trans. Circuits Syst. II*, vol. 40, pp. 163-173, Mar. 1993.

[49] L. O. Chua, T. Roska, T. Kozek, and A. Zarandy, "CNN universal chips crank up the computing power," *IEEE Circuits Devices Mag.*, vol. 12, pp. 18-28, July 1996.

[50] S. Espejo, R. Dom´ınguez-Castro, G. Li˜n´an, and A. Rodr´ıguez-V´azquez, "64 _ 64 CNN universal chip with analog and digital I/O," *IEEE 1998 Int. Conf. Electronic Circuits and Systems*, pp. 203-206, Sep. 1998.

[51] TSMC 0.35um Mixed-signal 2p4m Polycide 3.3V/5V SPICE Model.

APPENDIX

A. Chip Information

Chip Specification Table

Spec.	Result
Technology	TSMC 0.35μm 2P4M CMOS Mixed Signal Process
Power supply	3.3 Volts
Power dissipation	274mW(fully custom) + 8.3mW(cell based design)
Input level	Logical low: 0V, high: 3.3V
Output level	Logical low: 0V, high: 3.3V
Operational Frequency (Cell-based)	20MHz
Cell-based design Area	63315
Bias step time	0.1us
LAM accessing time	3.1us
Bias resolution	32-tap (5-bit)
No. of output	18 (fully=5, cell=13)
Finding LMV for each frame	Less than 6.2ms
Package	48-pin

Pad Names and Description

Pin No.	Name	I/O direction	Description
47	oY_mem19_25	O	Local Analog Memory testing point 19_25
2	oY_mem19_1	O	Local Analog Memory testing point 19_1
3	oY_mem1_25	O	Local Analog Memory testing point 1_25
6	oY_mem1_1	O	Local Analog Memory testing point 1_1
7	VDD_1	Power	Analog power 3.3 V
46	VSS_1	Power	Analog power 0 V
44	VDD_2	Power	Guard ring power 3.3 V
42	VSS_2	Power	Guard ring power 0 V
5	VDD_3	Power	Analog power 3.3 V
4	VSS_3	Power	Analog power 0 V
1	VDD_4	Power	Guard ring power 3.3 V
48	VSS_4	Power	Guard ring power 0 V
40	oY_biasL	O	CNN threshold bias output voltage test point
47	oY_biasU	I	Unity gain buffer voltage test point
8	iV8	I	Input Vin[8]
9	iV7	I	Input Vin[7]

10	iV6	I	Input Vin[6]
11	iV5	I	Input Vin[5]
12	iV4	I	Input Vin[4]
13	iV3	I	Input Vin[3]
14	iV2	I	Input Vin[2]
15	iV1	I	Input Vin[1]
16	itse	I	Scan chain enable
17	iVsh	I	Current voltage converter bias (2.85V)
18	iSTART	I	Start signal for CNN control circuit
20	itsi	I	Scan chain input
21	VSS_0	Power	Digital power 3.3V
22	iFRST	I	Reset
23	iFCLK	I	Clock
24	oY_Fsh	O	Finish signal
25	oY_LE	I	Load enable
26	oY_aY0	O	axis_Y[0]
27	oY_aY1	O	axis_Y[1]
28	oY_aY2	O	axis_Y[2]
29	oY_aY3	O	axis_Y[3]
30	oY_aY4	O	axis_Y[4]
33	VDD_0	power	I/O PAD power 3.3V
34	oY_ax0	O	axis_X[0]
35	INT_VSS_0	power	Digital power 0V
36	oY_ax1	O	axis_X[1]
37	oY_ax2	O	axis_X[2]
38	INT_VDD_0	power	Digital power 3.3V
39	oY_ax3	O	axis_X[3]
40	oY_ax4	O	axis_X[4]
41	oY_tso	O	Scan chain output

佈局驗證結果說明

Fully custom design LVS check

```
                ##################################################
                ##                                              ##
                ##        C A L I B R E   S Y S T E M           ##
                ##                                              ##
                ##            L V S   R E P O R T               ##
                ##                                              ##
                ##################################################


REPORT FILE NAME:         CHIP_temp.lvs.report
LAYOUT NAME:              CHIP_temp.lay.net ('CHIP_temp')
SOURCE NAME:              CHIP_temp.sp ('CHIP_temp')
RULE FILE:                /user/sunnycat/CNN_lvs/CHIP_temp/_cali035pMM5V_2P4M.lvs_
RULE FILE TITLE:          Calibre LVS Version V2.4a for TSMC 0.35um MIXED SINGAL POLYCIDE
CREATION TIME:            Thu Jun  2 02:19:43 2005
CURRENT DIRECTORY:        /user/sunnycat/CNN_lvs/CHIP_temp
USER NAME:                sunnycat
CALIBRE VERSION:          v2004.2_5.19    Tue Jun 29 19:44:37 PDT 2004



                              OVERALL COMPARISON RESULTS



                         #       ####################       _   _
                       #         #                  #       *   *
                     #   #       #     CORRECT      #         |
                      # #        #                  #       \___/
                       #         ####################
```

Whole CHIP DRC check

```
------------------------------------------------------------------------------------
--- RULECHECK RESULTS STATISTICS (BY CELL)
---
CELL PRDIODE ................... TOTAL Result Count = 7    (14)
    RULECHECK VERTEX_OFFGRID ... TOTAL Result Count = 7    (14)
CELL CHIP_TOP .................. TOTAL Result Count = 1001 (1001)
    RULECHECK VERTEX_OFFGRID ... TOTAL Result Count = 993  (993)
    RULECHECK M3.W.1 ........... TOTAL Result Count = 2    (2)
    RULECHECK M4.W.1 ........... TOTAL Result Count = 2    (2)
    RULECHECK M1.R.1 ........... TOTAL Result Count = 1    (1)
    RULECHECK M2.R.1 ........... TOTAL Result Count = 1    (1)
    RULECHECK M3.R.1 ........... TOTAL Result Count = 1    (1)
    RULECHECK M4.R.1 ........... TOTAL Result Count = 1    (1)
------------------------------------------------------------------------------------
--- SUMMARY
---
TOTAL CPU Time:                  19
TOTAL REAL Time:                 20
TOTAL Original Layer Geometries: 7880 (1495057)
TOTAL DRC RuleChecks Executed:   149
TOTAL DRC Results Generated:     1008 (1015)
```

DRC error : VERTEX OFFGRID 假錯! 1000

M3.W.1 為 corner layout 假錯 ! 2

M4.W.1 為 corner layout 假錯 ! 2

M1.R.1 為 metal1 density 不足 30% warning，為 CIC 可忽略假錯 1

M2.R.1 為 metal2 density 不足 30% warning，為 CIC 可忽略假錯 1
M3.R.1 為 metal3 density 不足 30% warning，為 CIC 可忽略假錯 1
M4.R.1 為 metal4 density 不足 30% warning，為 CIC 可忽略假錯 1
共 1008 個可忽略假錯

Whole CHIP LVS check

```
                ######################################################
                ##                                                  ##
                ##         C A L I B R E    S Y S T E M             ##
                ##                                                  ##
                ##             L V S   R E P O R T                  ##
                ##                                                  ##
                ######################################################



REPORT FILE NAME:          lvs.rep
LAYOUT NAME:               layout.spi ('CHIP_TOP')
SOURCE NAME:               ./CHIP_TOP.spi ('CHIP_TOP')
RULE FILE:                 Calibre-lvs-cur
RULE FILE TITLE:           Calibre LVS Version V2.4a for TSMC 0.35um MIXED SINGAL POLYCIDE
HCELL FILE:                (-automatch)
CREATION TIME:             Sun Jun  5 00:58:23 2005
CURRENT DIRECTORY:         /user/sunnycat/Finish/AMS1/Cclibre2/LVS/run
USER NAME:                 sunnycat
CALIBRE VERSION:           v2004.2_5.19    Tue Jun 29 19:44:37 PDT 2004



                              OVERALL COMPARISON RESULTS



                         #       ####################       _   _
                        #        #                  #       *   *
                     #   #       #     CORRECT      #         |
                      # #        #                  #       \___/
                       #         ####################
```

*** Chip Features CAD Tools ***

CKT name : CHIP_TOP (設計名稱) HSPICE

Technoloy : TSMC 0.35um 2P4M CMOS (使用製程) OPUS

Package : 48 S/B (包裝種類)

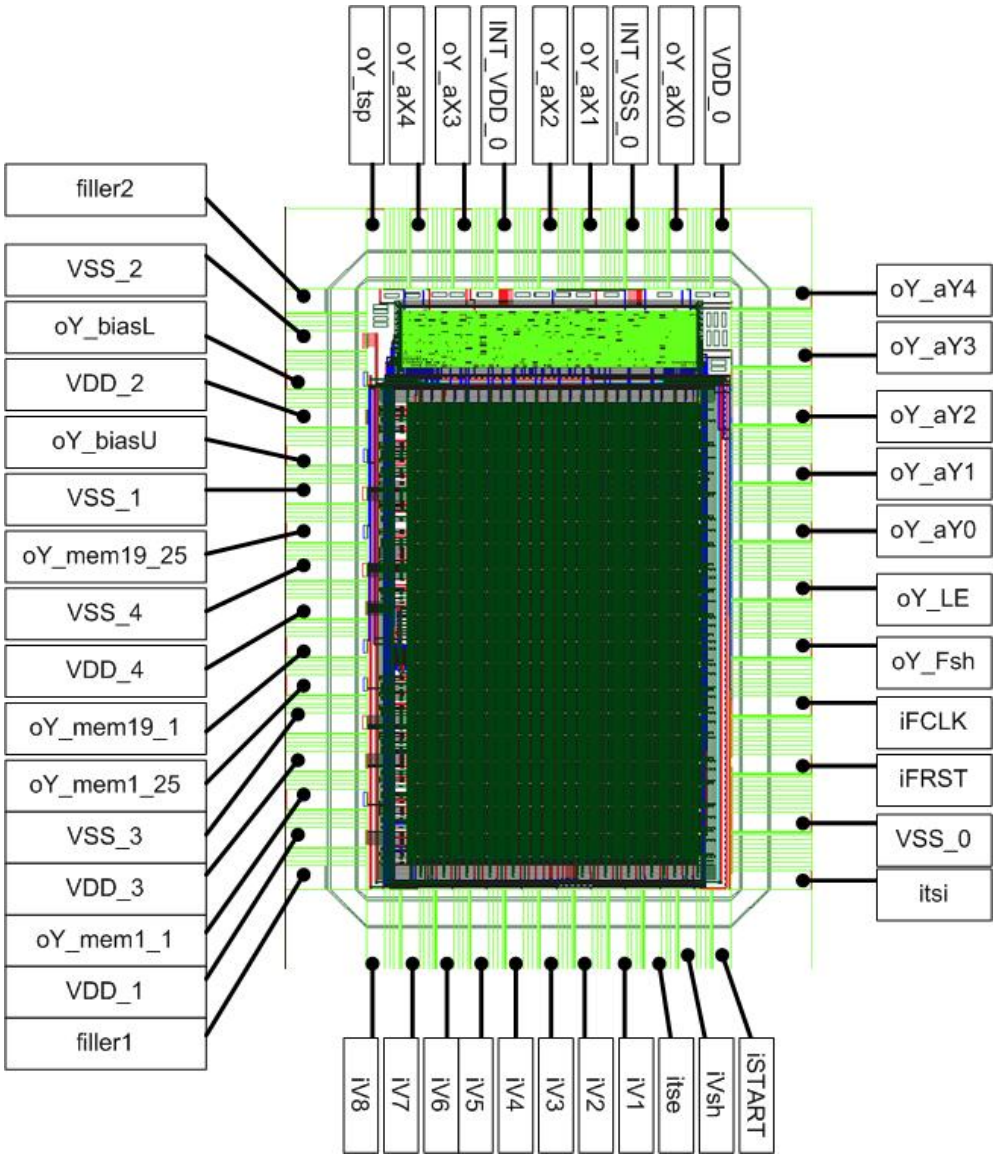
Chip Size : 2.375× 3.442 mm^2 (晶片面積)

Transistor/Gate Count : 4776 MOS / 6.33K gate count (電晶體/邏輯閘數)

Power Dissipation : ~355mW(最高)/~170mW(平均) (功率消耗；mW)

Max. Frequency : < 20 MHz (最高工作頻率，MHz) (最高工作頻率)

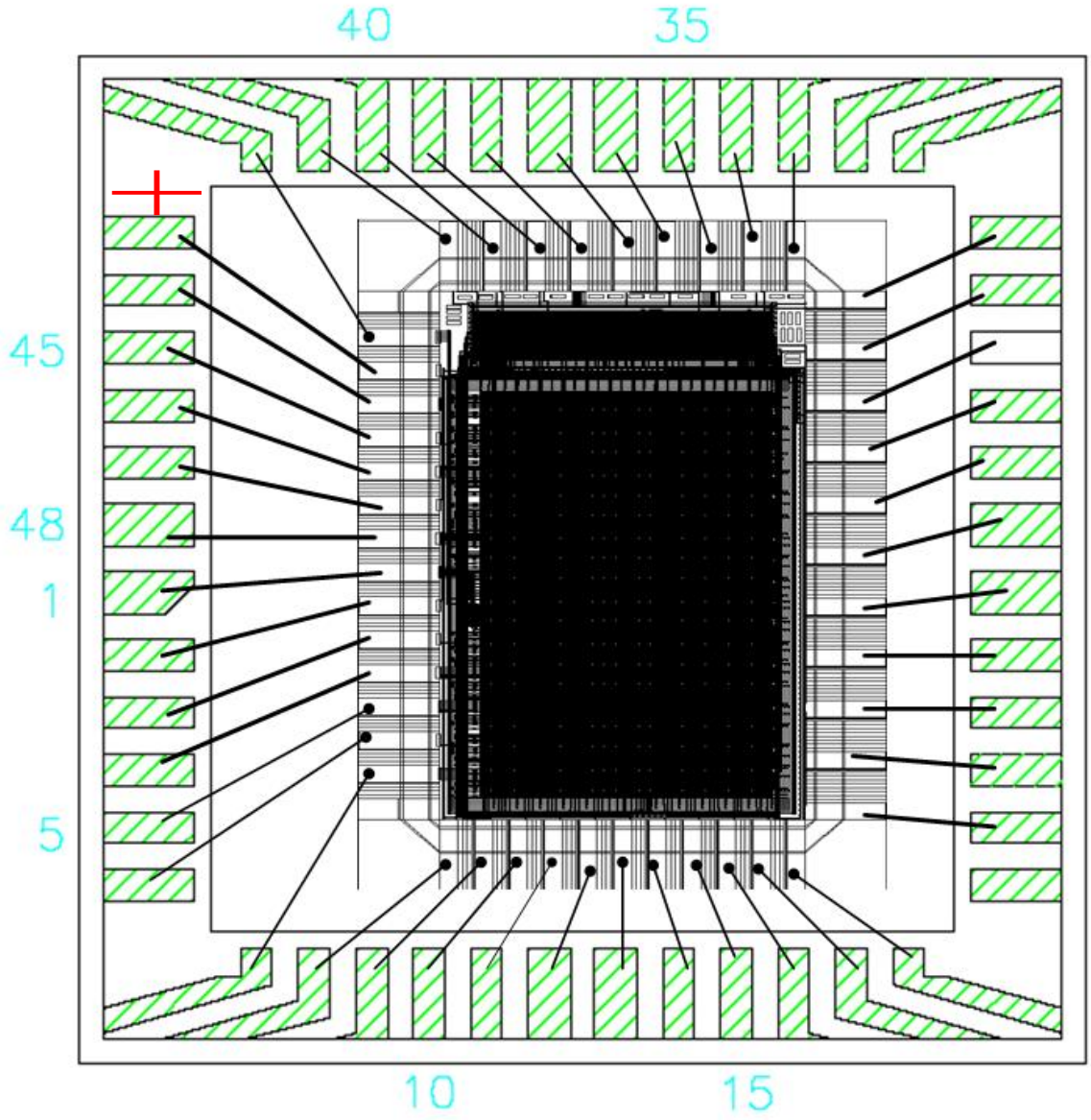
Testing Results : ☐Function work ☐Partial work ☐Fail



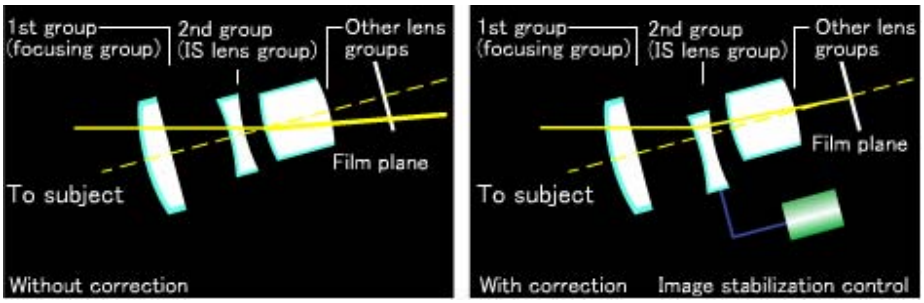
佈 局 平 面 圖

Side Braze 48-pin lead

Bonding Diagram (打線圖)



B. Optical IS Architecture



Source: Cannon website: http://www.canon.com/technology/dv/02.html [3]

If the lens front moves downward due to camera shake, the light from the subject shifts from the optical axis of the lens, and the image on the film plane is moved downwards. When this occurs, the corrective optical system is moved downwards to refract the light and the image is returned to the center of the film plane. In reality, both vertical and horizontal vibration occurs and the corrective optical system is moved in four directions along a plane perpendicular to the optical axis.