I, Madhuri Mailavaram, hereby submit this work as part of the requirements for the degree of:
Master of Science (M.S)
in:
Electrical Engineering
It is entitled:
A Standard Cell Library Using CMOS Transconductance Amplifiers for Cellular Neural Networks

This work and its defense approved by:
Chair: Dr. Carla Purdy
       Dr. Wen-Ben Jone
       Dr. George B. Purdy
A Standard Cell Library Using CMOS Transconductance Amplifiers for Cellular Neural Networks

A thesis submitted to the
Division of Research and Advanced Studies
in partial fulfillment
of the requirements for the degree of

Master of Science
in the
Department of Electrical and Computer Engineering
and Computer Science
of the College of Engineering at
University of Cincinnati, Cincinnati
2006

By

Madhuri Mailavaram
B.Tech. (Electrical and Electronics Engineering),
Jawaharlal Nehru Technological University, Hyderabad, India
2003

Thesis Advisor and Committee Chair: Dr. Carla Purdy
Abstract

Cellular Neural Networks (CNNs) form a class of information-processing systems which like neural networks are large-scale nonlinear analog circuits performing real time parallel processing of signals. Their local connectivity and regular architecture, unlike neural networks, make very efficient VLSI layouts, yielding higher chip densities, at very high operating speeds. Due to their continuous time feature, CNN’s form an alternative to conventional computers with great potential for image processing and pattern recognition applications.

In the analog CMOS VLSI realization of CNN’s, inverter based CMOS transconductance amplifier forms a basic building block. By choosing the appropriate transconductance parameters, according to the predetermined coefficients, this approach can be adapted for various CNN applications, thus enabling a standard cell library realization for various image processing applications. Standard cell library is laid out in MAGIC layout editor with 0.5µ technology and simulations are carried out in HSPICE. The library includes LOGICNOT, LOGICOR, SHIFT, DILATION and FILBLACK applications. Additional cells can be designed in a similar manner. In addition, simulations were carried out in MATLAB for the above mentioned CNN applications.
Acknowledgements

I acknowledge the contributions of all those members who made this Masters thesis possible. I express my heartfelt gratitude and indebtedness for my thesis advisor Dr. Carla Purdy for her constant support and excellent guidance all through the research work. She had always been a great source of inspiration and I strongly look forward to future collaborations with her.

I thank Dr. Wen- Ben Jone and Dr. George B. Purdy for agreeing to being the advisory members and for their reviews and appreciations.

For the unforgettable times, pleasant experiences and their invaluable support, I thank my friends Sirisha, Lakshmi V, Soumya, Bijoy, Sandeep, Lakshmi S, Manjiri and Deepti.

I am grateful to my lab mates Vikram, Jothiram, Shruthi N and Shruthi B, Aravind, Rajesh, Kiran, Vaishali, and Sriniv at Circuits and Systems Design Laboratory for making it a great place to work.

A special thanks to my class mates for the time we spent in both academic and nonacademic issues. I thank you all for making my stay in Cincinnati a cherished one.

I am obliged to University of Cincinnati and ECECS dept, for its excellent faculty, staff, and research, lab facilities. I take pride in choosing this institution for pursuing my Masters.

I dedicate this work to my dear Mom, Dad and Sisters (Jyothi and Sravanthi). I am deeply grateful for all their loving care, unfailing support, and motivation. I am fortunate to be a part of a great family and I owe them all that I achieved.

Finally, I end this note in thanking the almighty without whom I would not be what I am today.
# Table of Contents

List of Figures 4

List of Tables 7

1. Introduction
   1.1 Introduction to Cellular Neural Networks (CNN’s) 8
   1.2 Motivation 9
   1.3 Thesis Outline 10

2. Background
   2.1 Introduction to Feedback Neural Networks 12
   2.2 Cellular Neural Networks (CNNs) 13
   2.3 Various CNN Cloning Templates 23
   2.4 Typical CNN Applications 29
   2.5 CNN Dynamics Implementation Methods 30
   2.6 Analog VLSI Implementation Approach 31
   2.7 CNN Universal Machine 33

3. CNN Design Procedure
   3.1 MATLAB Simulation Procedure 36
      3.1.1 A Step by Step Procedure 37
3.2 Analog VLSI Implementation of Cellular Neural Networks 41
   3.2.1 Standard Cell Neuron Component Library Approach 41
   3.2.2 CMOS Inverter Based Transconductance Amplifier Approach 44
3.3 CMOS Transconductance Amplifier 44
3.4 A CNN “Cell” Implementation Using CMOS Transconductance Amplifier 47
   3.4.1 Resistor Implementation 49
   3.4.2 Positive/Excitatory Coupling Coefficient Implementation 50
   3.4.3 Negative/Inhibitory Coupling Coefficient Implementation 53

4. Standard Cell Library of CNN Target Applications

   4.1 Why This Library? 55
   4.2 A Modified CNN Cell Circuit 57
   4.3 CNNs Requiring No Coupling from Neighboring Cell Inputs 59
      4.3.1 LOGIC OR CNN Cell Realization 60
      4.3.2 LOGIC NOT CNN Cell Realization 62
   4.4 CNNs Requiring Coupling from Neighboring Cell Inputs 65
      4.4.1 SHIFT WEST CNN Cell Realization 65
      4.4.2 SHIFT EAST CNN Cell Realization 67
      4.4.3 DILATION WEST CNN Cell Realization 69
   4.5 CNNs Requiring No Input Coupling 72
      4.5.1 FILBLACK CNN Cell Realization 72
   4.6 Summary of Standard CNN Cell Library Realizations 74
5. Results and Simulations

5.1 Chapter Outline 76

5.2 MATLAB Results 78

5.2.1 LOGIC OR CNN 78

5.2.2 LOGIC NOT CNN 79

5.2.3 SHIFT WEST CNN 80

5.2.4 DILATION WEST CNN 81

5.2.5 FILBLACK CNN 82

5.3 MAGIC Layouts and HSPICE Simulation Results 83

5.3.1 Basic CNN Cell Component Library 83

5.3.2 Standard Cell Library of CNNs 94

5.3.2.1 CNNs Requiring No Coupling from Neighboring Cell Inputs 94

5.3.2.2 CNNs Requiring Coupling from Neighboring Cell Inputs 101

5.3.2.3 CNNs Requiring No Input Coupling 107

6. Conclusions and Future Work

6.1 Summary and Conclusions 113

6.2 Future Work 114

Bibliography 116

Appendix A 119

Appendix B 128
List of Figures

2.1 A CNN with r=1 (3×3) neighborhood 14
2.2 Some nonlinear output functions for a CNN cell 16
2.3 System structure of a CNN cell C (i, j) [1. p.28] 17
2.4 Circuit realization of a standard CNN cell C (i, j) 20
2.5 Proposed cell circuit model of Chua and Yang 21
2.6 The paradigm of CNN with a 3×3 neighborhood 22
2.7 Typical CNN applications 29
2.8 Architecture of the CNN Universal Machine 35
3.1 MATLAB code for CNN DILATION WEST part (1/4) 37
3.2 MATLAB code for CNN DILATION WEST part (2/4) 38
3.3 MATLAB code for CNN DILATION WEST part (3/4) 39
3.4 MATLAB code for CNN DILATION WEST part (4/4) 40
3.5 An op amp implementation of a simple CNN cell 41
3.6 Two-stage operational amplifier with class AB output buffer stage 43
3.7 a) The transistor schematic b) Input-output characteristics of CMOS transconductance element 45
3.8 Realization of cell circuit with CMOS transconductance elements 48
3.9 Resistor implementation by CMOS VCT 50
3.10 Circuit diagram of fixed positive transconductance amplifier 51
3.11 Circuit diagram of programmable positive transconductance amplifier 52
3.12 Circuit diagram of fixed negative transconductance amplifier 53
3.13 Circuit diagram of programmable negative transconductance amplifier

4.1 Modified CNN cell circuit using CMOS transconductance amplifier

4.2 A LOGIC OR CNN cell circuit using CMOS transconductance elements

4.3 A LOGIC NOT CNN cell circuit using CMOS transconductance elements

4.4 A SHIFT WEST CNN cell circuit using CMOS transconductance elements

4.5 A SHIFT EAST CNN cell circuit using CMOS transconductance elements

4.6 A DILATION WEST CNN cell circuit using CMOS transconductance elements

4.7 A FILBLACK CNN cell circuit using CMOS transconductance elements

5.1 LOGIC OR/ Set union/ Disjunction CNN

5.2 LOGIC NOT/ Set complementation CNN

5.3 SHIFT WEST/ Translation by one pixel unit CNN

5.4 DILATION/ Grow-until-it-fits CNN

5.5 FILBLACK/ Gray-scale to black CNN

5.6 Transconductance amplifier Magic layout

5.7 HSPICE voltage transfer characteristics of the transconductance amplifier at \( V_{G1}=5, V_{G4}=-5 \)

5.8 Resistor Magic layout a) top level and b) expanded design

5.9 HSPICE VI characteristics of the resistor at \( V_{G1}=5, V_{G4}=-5 \)

5.10 Programmable positive transconductance amplifier Magic layout a) top level and b) expanded design

5.11 HSPICE voltage transfer characteristics of the programmable positive transconductance amplifier at a particular W/L ratio
5.12 Programmable negative transconductance amplifier Magic layout a) top level and b) expanded design

5.13 Transfer characteristics of the programmable negative transconductance amplifier at a particular W/L ratio

5.14 A Single cell LOGIC OR CNN Magic layout a) top level b) expanded design

5.15 Single cell LOGIC OR CNN HSPICE results verifying its functionality

5.16 A 1×3 LOGIC OR CNN Magic layout a) top level b) expanded design

5.17 1×3 LOGIC OR CNN HSPICE results verifying its functionality

5.18 A Single cell LOGIC NOT CNN Magic layout a) top level b) expanded design

5.19 A 1×3 SHIFT WEST CNN Magic layout a) top level b) expanded design

5.20 1×3 SHIFT WEST CNN HSPICE results verifying its functionality

5.21 A 1×3 DILATION WEST CNN Magic layout a) top level b) expanded design

5.22 A Single cell FILBLACK CNN Magic layout a) top level b) expanded design

5.23 Single cell FILBLACK CNN HSPICE results verifying its functionality

5.24 A 1×3 FILBLACK CNN Magic layout a) top level b) expanded design

5.25 1×3 FILBLACK CNN HSPICE results verifying its functionality
## List of Tables

3.1 Comparison of parameters for different CNN cell descriptions 49  
4.1 Summary of standard CNN cell library realizations 75  
5.1 Transconductance variation with gate voltages 85  
5.2 Resistance variation with gate voltages 87  
5.3 Positive transconductance variation with gate voltages 90  
5.4 Negative transconductance variation with gate voltages 93  
5.5 LOGIC OR CNN HSPICE results 99  
5.6 LOGIC NOT CNN HSPICE results 101  
5.7 1×3 SHIFTWEST CNN HSPICE results 104  
5.8 1×3 SHIFTEAST CNN HSPICE results 105  
5.9 1×3 DILATIONWEST CNN HSPICE results 107  
5.10 FILBLACK CNN HSPICE results 112
1. **Introduction**

1.1 **Introduction to Cellular Neural Networks (CNN’s)**

The revolutionary analogic cellular computer paradigm called CNN Universal Machine is a major computing paradigm for processing analog array signals. Its implementations have been shown to be far superior to any equivalent DSP implementation in terms of the speed, power and area (SPA) measures. Its many applications include real –time image and video processing along with implementing brain-like information processing schemes. [1, pp. 1-3].

Cellular Neural Network (CNN), proposed by Chua and Yang in 1988, is the core of this novel class of information-processing systems. Each processing cell interacts with its nearest neighbor according to a program or an algorithm. Global interactions between cells not directly connected together are possible because of the propagation effects of the continuous dynamics of CNNs [3]. There exist several properties that give CNNs more advantages than Artificial Neural Networks (ANN). As the cells are only connected within a certain neighborhood but not to the entire network, extension is easy without having to readjust the entire network. Due to their complex dynamic behavior, as seen with other neural networks, CNNs can be used in image processing applications such as noise removal, image thinning and connected component detector (CCD) or as an associative memory [2, pp. 10].
1.2 Motivation

As described by Chua and Yang, a cellular neural network, like an artificial neural network (ANN), is a large-scale nonlinear array circuit of analog dynamic processors or cells. CNNs have the nearest neighbor interactions found in cellular automata and accept and generate analog continuous real time signals [3]. The regularity, the parallelism and the local connectivity found in the CNN circuit architecture make it suitable for very high speed VLSI implementations with easy routing and increased cell density per silicon area [4]. Attributes like compactness, continuous real-time dynamics and parallel processing features make analog VLSI techniques the preferred technique for hardware realizations.

Motivated by the above facts, a CNN structure based on analog CMOS transconductance elements approach [5] has been implemented. Since the major goal is design simplicity in the approach, the design is reduced to the design of few types of transconductance elements. By choosing the appropriate transconductance parameters according to the predetermined coupling coefficients between the neighboring cells, various types of CNN applications can be realized. Programmability of the transconductance amplifier can be achieved by adjusting the external voltage sources. The goal of this thesis is to extend the architecture described in [5] to other image processing applications by exploring the use of standard analog cells, the basic building blocks of the CNN cells for the implementation of standard cell library of CNN applications such as LOGICNOT, LOGICOR, FILBLACK, SHIFT and DILATION.
1.3 Thesis Outline

This section outlines the organization of the thesis.

Chapter 2 gives background information about the ANNs and CNNs. CNN theory, its various templates and typical applications are discussed. The CNN dynamics implementation methods and reasons for choosing analog VLSI implementation approach for CNN realizations are presented. This is then followed by a brief discussion on CNN Universal Machines.

Chapter 3 includes the procedure for MATLAB simulations. We then describe two analog design approaches for a CNN cell, the standard cell neuron component library approach and the CMOS inverter based transconductance amplifier approach. A brief discussion on CMOS transconductance amplifier is given. The realization of the components of a cell and thus a cell using the above transconductance amplifier approach is described.

Chapter 4 presents the extension of the above approach to realizing a standard library of cells for CNN target applications of LOGICOR, LOGICNOT, SHIFT, DILATION and FILBLACK.

Chapter 5 shows the MATLAB simulation plots for the above mentioned applications, the MAGIC layouts and HSPICE simulation results for the basic components of a CNN cell and of various CNN cells in the library.
Chapter 6 includes the conclusions and future work of this thesis.

Bibliography and Appendices follow at the end.
2. Background

2.1 Introduction to Feed Back Neural Networks

Artificial Neural Network (ANN) is any computing architecture consisting of massively parallel interconnections of simple neural processors, which partially mimic the structure and functions of the brain and nervous system [5]. Here we will be focusing on Cellular Neural Networks (CNNs) which are derived from a particular case of ANN, the continuous Hopfield Neural Networks, which are large-scale dynamical feedback networks consisting of highly interconnected simple processors called neurons [2, p.2]. Dynamical systems are those systems which process the initial condition information over time while moving through a sequence of states. In analogy to biological structures, these systems take advantage of distributed information processing and their inherent potential for parallel computation [2, p.2]. These networks thus prove to be useful for pattern associations or classifications (pattern recognition), optimization problem solutions and pattern restoration. These networks are of two types: continuous-time (gradient type) and discrete-time (recursive type). Discrete-time networks are a limit case of continuous-time networks. For a very high gain of the neurons, continuous-time networks perform similarly to discrete-time networks. Continuous-time neural networks are examples of nonlinear, dynamical, and asymptotically stable systems [6]. One of the major applications of Hopfield networks is in implementing an associative memory [2, p.2] (i.e., a memory able to store certain memories or patterns in a manner rather similar to the brain; the full pattern can be recovered if the network is presented with only partial information).
In order to reduce the number of interconnections between neurons but to keep the advantages of parallel processing of ANNs, Chua and Yang proposed the CNN architecture. Their architecture is promising for modeling biological nervous systems and electronic neural networks [2]. As in cellular automata, here neurons or cells are only connected to each other within a certain neighborhood (i.e., they are governed by local interconnections between neurons within this neighborhood). This property would also enable easy network extensions without entire network readjustments. The major differences between a CNN and a general continuous Hopfield network are the local interconnections between neurons and piece-wise linear output activation functions, which affect the possibilities of implementation and CNN applications [2, p.1]. Whereas the general ANN often suffers from severe dynamic restrictions in the circuit implementation stage, a CNN has a practical dynamic range. [2, p.30]

2.2 Cellular Neural Networks (CNNs)

Cellular Neural Network (CNN) is an analog dynamic processor array with the processing elements interacting directly within a finite local neighborhood. Here the processors and interactions are analog in nature, unlike a cellular automaton or systolic array [7]. Due to dynamic propagation, a cell not only interacts with the near neighbors but it can also have some global properties because of the propagation effects of these local interactions during the transient regime [3].
Following [7], we define a CNN to be an n-dimensional array of mainly identical dynamical systems, called cells, which interact only with those cells within a fixed radius r called the sphere of influence, $N_r(i,j)$, also called a $(2r+1) \times (2r+1)$ neighborhood.

Considering a (2-D, n=2) grid, a class I $M \times N$ standard CNN architecture is defined by an $M \times N$ rectangular array of cells, $C(i,j)$ located at site $(i,j)$, $i=1,2,...M$, $j=1,2,...N$ [1, p.7]. The CNN cells can be classified as regular (internal) and boundary cells. Boundary cells include the edge and the corner cells [1, p.8]. Figure 2.1 below shows a CNN with $r=1$ (3 x 3 neighborhood) along with the various cells in the network.

![Fig. 2.1 A CNN with r=1 (3 x 3) neighborhood](image)
The CNN array dynamics [7] can be described by the state equation (2.1) and output equation (2.2):

\[
C_x \cdot x_{ij}(t) = \frac{1}{R_x} x_{ij}(t) + \sum_{C(k,l)\in N_{ij}} A(i,j;k,l) \cdot (y_{kl}(t), y_{ij}(t)) + \sum_{C(k,l)\in N_{ij}} B(i,j;k,l) \cdot (u_{kl}(t), u_{ij}(t)) + z_{ij}(t) \tag{2.1}
\]

\[
y_{ij}(t) = f(x_{ij}) \tag{2.2}
\]

\(\forall 1 \leq (k, i) \leq M \text{ and } 1 \leq (l, j) \leq N\)

Here \(x_{ij}, y_{ij}, u_{ij}, z_{ij}\) are called state, output, input and threshold respectively, of cell \(C(i, j)\). Here \(ij\) is a grid point associated with a cell on the 2-D grid and \(kl\) is a grid point in the neighborhood \((N_r)\) within a radius \(r\) of the cell \(C(i, j)\). \(C_x > 0\) is a linear capacitor, \(R_x > 0\) is linear resistor and \(A, B\) are nonlinear cloning templates.

For an example gray scale image, continuous input (output) signal values are represented by values in the range \([-1, 1]\) given below by (2.3):

\[
|x_{ij}(0)| \leq 1, \quad |u_{ij}| \leq 1 \tag{2.3}
\]

Complete stability of a CNN is proven by the symmetry of the feedback cloning template \(A\), given below by (2.4).

\[A(i, j; k, l) = A(k, l; i, j)\] and \(A(i, j; i, j) > 1/R_x\) \tag{2.4}
Output Functions

$f(x_y)$ is any of the output functions given in the Figure 2.2 below and the output equations for these figures follow.

![Image of some nonlinear output functions for a CNN cell]

**Fig. 2.2 Some nonlinear output functions for a CNN cell**

- Piecewise-linear sigmoid [3] function with a unity gain factor is given by:

$$f(x_y) = \frac{1}{2} \phi \left( [x_y + 1] - [x_y - 1] \right)$$

(2.5)

- Gaussian function is given by:

$$f(x_y) = \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{(x_y - \mu)^2}{2\sigma^2}}$$

(2.6)

- Threshold function is given by:

$$f(x) = \begin{cases} 0 & \text{if } 0 > x \\ 1 & \text{if } x \geq 0 \end{cases}$$

(2.7)

Note that other output functions are also possible, e.g. Inverse Gaussian

Dynamic Range of a CNN

In a CNN, all states $x_y$ are bounded for all time $t>0$ and the bound $V_{\text{max}}$ [2. p30] is given by

$$x_{\text{max}} = 1 + |I_y| + \max_{1 \leq i \leq M, 1 \leq j \leq N} \left[ \sum_{C(k,j) \in N_r(i,j)} (|A(i, j; k, l)| + |B(i, j; k, l)|) \right]$$

(2.8)
A system structure of a cell C (i, j) representing the equations (2.1) and (2.2) of Chua and Yang model is shown in Figure 2.3 below. Here arrows in bold indicate parallel data paths from input and the output of the surrounding cells $u_{kl}$ and $y_{kl}$ respectively, and arrows in thin indicate the threshold, input, state and output $z$, $u_{ij}$, $x_{ij}$ and $y_{ij}$ respectively and $\tau = R_x C_x$ is the time constant governing the CNN dynamics [1. p.28].

![Fig. 2.3 System structure of a CNN cell C (i, j) [1. p.28]](image)

For a linear space-invariant template each cell is described by simple identical cloning templates defined by 2 real matrices, A and B, called feedback and feed-forward/ input template matrices respectively, as shown below with $r=1$.

$$A = \begin{bmatrix}
    a_{1,-1} & a_{1,0} & a_{1,1} \\
    a_{0,-1} & a_{0,0} & a_{0,1} \\
    a_{1,-1} & a_{1,0} & a_{1,1}
\end{bmatrix}$$

$$B = \begin{bmatrix}
    b_{1,-1} & b_{1,0} & b_{1,1} \\
    b_{0,-1} & b_{0,0} & b_{0,1} \\
    b_{1,-1} & b_{1,0} & b_{1,1}
\end{bmatrix}$$
It is convenient to split the \( A \) matrix as \( A = A^0 + \mathbf{\tilde{A}} \), where \( A^0 = a_{0,0} \) and \( \mathbf{\tilde{A}} \), the rest of the \( A \) matrix are called the center and surround feedback component templates respectively and \( B \) matrix as \( B = B^0 + \mathbf{\tilde{B}} \), where \( B^0 = b_{0,0} \) and \( \mathbf{\tilde{B}} \), the rest of the \( B \) matrix are called the center and surround feed-forward component templates respectively [1. p.26].

This form of \( A \) and \( B \) enables us to write the state equation in the linear case in a more compact form by introducing the 2-dimensional convolution operator \( \otimes \).

The convolution operator \( \otimes \) for any cloning template \( T \), which defines the dynamic rule of the cell circuit, is given by (2.9) [1. p25].

\[
\begin{align*}
T \otimes S_g &= \begin{bmatrix}
= \sum_{C (k,j) \in N_r (i,j)} T (i, j; k, l) s_{kl} \\
= \sum_{C (k,j) \in N_r (i,j)} T (k - i, l - j) s_{kl} \\
= \sum_{|k - i| \leq 1} \sum_{|l - j| \leq 1} T (k - i, l - j) s_{kl}
\end{bmatrix} \\
&= t_{-1,-1} s_{i-1,j-1} + t_{-1,0} s_{i-1,j} + t_{-1,1} s_{i-1,j+1} \\
&\quad + t_{0,-1} s_{i,j-1} + t_{1,0} s_{i,j} + t_{0,1} s_{i,j+1} \\
&\quad + t_{1,-1} s_{i+1,j-1} + t_{1,0} s_{i+1,j} + t_{1,1} s_{i+1,j+1} \\
&= \sum_{k = -1}^{1} \sum_{l = -1}^{1} t_{k,l} s_{i+k,j+l}
\end{align*}
\]
Here $t_{mn} = T(m,n)$ denotes the $m^{th}$ row and $n^{th}$ column entry of the cloning template where $m=-1, 0, 1$ and $n=-1, 0, 1$ respectively and $s_{ij} = S(i,j)$ denotes the element in the $i^{th}$ row and $j^{th}$ column of $S$ matrix, where $S$ could be either the input matrix $U$ or the output matrix $Y$.

Using the above notation, a space invariant linear CNN is completely described by the following state equation (equation defining the state ‘$x$’ of the cell dynamics) after setting $C_x = 1$ and $R_x = 1$.

$$
\dot{x}_{ij} = -x_{ij} + A \otimes Y_{ij} + B \otimes U_{ij} + z_{ij} \quad (2.10)
$$

A possible electronic circuit model of a CNN cell is shown below in Figure 2.4 [1 p.31]. In this figure, the Voltage Controlled Current Sources (VCCS’s) are represented by diamond-shaped symbols. These linear VCCS’s are used to model various coupling coefficients, by injecting a current proportional to the indicated controlling voltage $u_{kl}$ or $y_{kl}$, weighted by $b_{kl}$ and $a_{kl}$ respectively. A nonlinear VCCS $f(x_{ij})$ models the output current.
Figure 2.4 is equivalent to the proposed cell circuit model by Chua and Yang [3] shown in Figure 2.5 below, where $V_{xij}, V_{ykl}(t), V_{ukl}$ represent state, output, and input voltages respectively of a cell $C(i,j)$. $C > 0$ models a linear capacitor and $R_x, R_y > 0$ model linear resistors and $I, E_{uij}$ model independent current and voltage sources respectively.

In Figure 2.5, for space invariant CNN with $r=1$, a total of 9 linear VCCS are modeled by $I_{xv}(i,j;k,l) = B(i,j;k,l) \cdot V_{ukl}$, each of which depend on the 8 input voltages $V_{ukl}$ of
neighbors plus its own input voltage $V_{uij}$. This is equivalent to $B \otimes U_{ij}$ of Figure 2.4. A total of 9 more linear VCCS are modeled by $I_{xy}(i, j; k, l) = A(i, j; k, l) \cdot V_{ykl}(t)$ each of which depend on the 8 output voltages $V_{ykl}$ of neighbors and its own output voltage $V_{yij}$.

This is equivalent to $A \otimes Y_{ij}$ of Figure 2.4. Each cell also has one nonlinear VCCS modeled by $I_{yx} = V_{yij} = \frac{f(V_{xy})}{R_y}$, where $f(V_{xy})$ could be any of the output functions given by equations (2.5)-(2.7). This is equivalent to $Y_{ij}$ of Figure 2.4.

![Fig. 2.5 Proposed cell circuit model of Chua and Yang [3]](image)

**A Typical CNN**

In the figure below, we see a CNN for a particular image processing application, where a cell has the weighted contributions ($A$, $B$, $z$) of the inputs from neighbors, outputs from the neighbors and itself, its external inputs and bias voltages specified for an application.
Several types of CNN’s can be generated, based on the preceding generic definition of CNN. These can be classified according to the grid type, the processor (cell), the interaction (cloning template), and the mode of operation. Here we concentrate on

- Grid type: Square 2-D planar grid, fixed grid-size, one layer.
- Processor type: Sigmoid, first order (one capacitor), no local analog memory and no local logic.
- Interaction types: Linear of one, two or more variables, memoryless, symmetric, dynamic, noiseless, continuous valued fixed template
- Modes of operation: Continuous- time, fully analog, transient/ dc steady state, deterministic

\[ A = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 0 \end{bmatrix} \]

\[ B = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} \]

\[ z = -0.5 \]

**CNN Taxonomy/ Glossary of CNN types [7]**

Fig. 2.6 The paradigm of CNN with a $3\times3$ neighborhood [8]
2.3 Various CNN Cloning Templates [1]

Space-invariant standard CNNs with a $3 \times 3$ neighborhood form the majority of CNN applications. Here the triple $\{A, B, z\}$ uniquely defines the CNN and thus, for $r=1$ neighborhood a total of 19 real numbers govern each CNN application. As there are uncountably many real numbers, there are infinitely many distinct CNN templates. Three mathematically simple subclasses are discussed below. Here $A$ is the feedback or output cloning template, $B$ is the feed forward or input cloning template and $z$ is the threshold/bias. $A$ and $B$ are split according to the equations $A = A^0 + \bar{A}$ and $B = B^0 + \bar{B}$ as discussed in the previous section.

- Zero-Feedback (Feed-forward) class $C\{0, B, z\}$, if and only if all feedback elements are zero, i.e. $A = 0$.

- Zero-input (Autonomous) class $C\{A, 0, z\}$, if and only if all feed-forward template elements are zero, i.e. $B = 0$.

- Uncoupled (scalar) class $C\{A^0, B, z\}$, if and only if $\bar{A} \equiv 0$ except $A^0$.

Already developed, simple CNN templates belonging to uncoupled scalar class $C\{A^0, B, z\}$ given in [1, pp.58-97], since $\bar{A} \equiv 0$ except $A^0$, governing some CNN applications are discussed below.
**LOGIC OR: Logic OR and set union template**

Global task: Is to transform an Input Image \( U(t) = P_1 \) and Initial State \( X(0) = P_2 \) into an Output Binary Image \( Y(t=\infty) \), which is the logic OR operation between \( P_1 \) and \( P_2 \), obeying the following

**Local rules:**

<table>
<thead>
<tr>
<th>( u_{ij} (0) )</th>
<th>( x_{ij} (0) )</th>
<th>( \rightarrow )</th>
<th>( y_{ij} (\infty) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>White Pixel</td>
<td>White Pixel</td>
<td>White, independent of neighbors</td>
<td></td>
</tr>
<tr>
<td>White Pixel</td>
<td>Black Pixel</td>
<td>Black, independent of neighbors</td>
<td></td>
</tr>
<tr>
<td>Black Pixel</td>
<td>White Pixel</td>
<td>Black, independent of neighbors</td>
<td></td>
</tr>
<tr>
<td>Black Pixel</td>
<td>Black Pixel</td>
<td>Black, independent of neighbors</td>
<td></td>
</tr>
</tbody>
</table>
Global task: Is to transform an input Image $U(t) = P$ and Zero Initial State $X(0) = 0$ into an Output Binary Image $Y(t=\infty)$, where black pixel turns white and vice versa obeying the following

Local rules:

<table>
<thead>
<tr>
<th>$u_{ij}(0)$</th>
<th>$\rightarrow$</th>
<th>$y_{ij}(\infty)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Black</td>
<td>White, independent of initial states</td>
<td></td>
</tr>
<tr>
<td>White</td>
<td>Black, independent of initial states</td>
<td></td>
</tr>
</tbody>
</table>

Note that if we have LOGICOR and LOGICNOT we can represent all Boolean Functions.
Global task: Is of transforming an Input Image \( U(t) = P \) and a Zero Initial State \( X(0) = 0 \) into an Output Binary Image \( Y(t) \rightarrow Y(\infty) = P(x-\alpha, y-\beta) \) i.e. \( P \) shifted by one pixel along the eight compass directions \((\alpha, \beta), \in \{-1, 0, 1\} \) and \((x, y)\) being the Cartesian coordinates. The Output Image obeys the following Local rules:

<table>
<thead>
<tr>
<th>( u_{ij}(0) )</th>
<th>( \rightarrow )</th>
<th>( y_{ij}(\infty) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arbitrary (-1 or 1)</td>
<td>1 (Black) if ( u(i+\alpha, j+\beta)=1 )</td>
<td></td>
</tr>
<tr>
<td>Arbitrary (-1 or 1)</td>
<td>-1 (White) if ( u(i+\alpha, j+\beta)=-1 )</td>
<td></td>
</tr>
</tbody>
</table>

Changing the values of \( B \), Input template according to the eight compass directions, shifts the output image by one-pixel in a particular compass direction.
DILATION: Grow-until-it-fits template

Global task: Is of transforming an Input Image \( U(t)=P \) with a given \( 3 \times 3 \), 0-1 representation of the structuring template \( S \) and a Zero Initial State \( X(0)=0 \) into an Output Binary Image \( Y(t) \to Y(\infty) \) which is a binary image generated by taking the set-union of \( S \) over all black pixels \( C_b(i, j) \) of \( P \), Where \( S \) is that set of black pixels obtained by translating the structuring template \( S \) and anchoring it to each black pixel \( C_b(i, j) \in P \).

Note: By coding the pixels of the set \( S \) by “black=1” and “White=0” and having the structuring template \( S \), the obtained \( B \) template is a reflection of \( S \) with respect to origin, i.e. by an 180° rotation.

\[
A= \begin{bmatrix}
0 & 0 & 0 \\
0 & 2 & 0 \\
0 & 0 & 0
\end{bmatrix}
\]

\[
B= \begin{bmatrix}
b_{-1,-1} & b_{-1,0} & b_{-1,1} \\
b_{0,-1} & b_{0,0} & b_{0,1} \\
b_{1,-1} & b_{1,0} & b_{1,1}
\end{bmatrix}
\]

\[
z=z_D
\]

Where \( b_{kl}=0 \) or 1, \( z_D=p_1-0.5 \), and \( p_1=\text{total number of 1s in the B template, } p_1 > 0 \)

The Output Image obeys the following Local rules:

<table>
<thead>
<tr>
<th>( u_{ij}(0) )</th>
<th>( -&gt; )</th>
<th>( y_{ij}(\infty) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Black</td>
<td>Black, if all eight nearest neighbors of ( C(i, j) ) are black</td>
<td></td>
</tr>
<tr>
<td>Black</td>
<td>Black, if ( b_{00}=1 ) (or if the central pixel of ( S ) is black)</td>
<td></td>
</tr>
<tr>
<td>Arbitrary (-1 or 1)</td>
<td>1 (Black), if there is at least one nearest neighbor ( C(i+\alpha, j+\beta) ) of (-1 or 1) ( C(i, j) ) whose color is black and if the color code of the corresponding pixel ( b_{\alpha\beta} ) of the ( B ) template (at the same relative position ( (\alpha, \beta) )) is also black, i.e., ( u_{i+\alpha, j+\beta}=1, b_{\alpha\beta}=1 )</td>
<td></td>
</tr>
<tr>
<td>Arbitrary (-1 or 1)</td>
<td>-1 (White) if there is no nearest neighborhood of (-1 or 1) ( C(i,j) ) with the above property</td>
<td></td>
</tr>
</tbody>
</table>
FILBLACK: Gray-scale to black template

Global task: Is to transform an input Image $U(t) = 0$ and Initial State $X(0) = P$ into an Output Binary Image $Y(t=\infty)$, which is a black image where all pixels are black and obeying the following

Local rules:

| Arbitrary | $x_{ij}(0) \in (-\infty, \infty)$ | $\rightarrow$ | $y_{ij}(\infty) = 1$ |

Note: 1 represents black pixel

This CNN belongs to the zero-input (Autonomous) class $C\{A, 0, z\}$ of CNNs as all the feed-forward template elements are zero i.e. $B \equiv 0$
2.4 Typical CNN Applications

Numerous CNN applications have already been found, and new applications are emerging rapidly. In Figure 2.7 below we list some of them [9].

CNN APPLICATIONS

- **Image processing including gray scale image inputs**
  - Feature extraction
  - Motion detection and estimation
  - Path tracking
  - Collision avoidance
  - Half-toning including motion
  - Object counting and size estimation

- **Analyzing 3D surfaces**
  - Detecting minima and maxima
  - Detecting areas where the gradients exceed a given limit

- **Solving partial differential equations**

- **Reducing non-visual problems to geometric maps**
  - Thermographic images
  - Somatosensory maps (tactile sensing)
  - Antenna array images
  - Different Medical maps and images

- **Modeling biological vision and other sensory-motor organs**
  *In a neuromorphic way* (*neuromorphic* means that the structure of the templates follows the structure of the neuroanatomy of the given organ), many CNN models of various parts of the visual pathway have been found. For various form, motion, color, and depth calculations, CNN analogic algorithms have been developed

Fig. 2.7 Typical CNN applications [9]
2.5 CNN Dynamics Implementation Methods [1]

CNN dynamics may be simulated, analyzed and implemented in several different ways [1, pp.100]:

- **Mathematical analysis** of qualitative behavior and numerical methods to calculate the quantitative results.

- **Software implementation**: Software simulators (e.g. MATLAB) using one of the numerical methods for solving the set of ODE’s of CNN dynamics.

- **Hardware implementation**:
  1) Digital implementation: multi-processor (DSP) digital emulators, hardware accelerator boards to speed up the software simulators.
  2) Analog implementation: continuous-time or discrete-time physical implementation of the CNN dynamics in the form of programmable analog VLSI chips.

- **Living organs** which reflect the CNN dynamics (e.g. the retina or other parts of the retinotopic visual pathway.
2.6 Analog VLSI implementation Approach

Here we mention some reasons for choosing the hardware implementation approach and for choosing the analog VLSI approach for creating a standard library of CNN cells for image processing applications.

☐ Why Hardware Instead of Software?

CNN architectures easily lend themselves to hardware implementation for the following reasons.

- Local Connectivity: The CNN architecture enables very efficient VLSI layouts as the number and the distances of the physical connections are limited, thereby leading to the increased cell density per silicon area, and the ability to operate at very high speeds [7]. This is in contrast to the connectivity required for fully connected neural networks whose fabrication is difficult because of the large number and the distances needed for the connections.

- Regularity: This enables
  - Easier routing and layout than for traditional analog circuits.
  - Usage of same routing and layout tools as in digital circuit design.
  - Usage of the same standard cell design techniques which enable the digital VLSI circuits to be rapidly completed, for a majority of CNN applications (space invariant CNNs), as the same circuit is duplicated throughout the chip.

- Simple Cells: The very simple structured and space invariant processing units and interconnection elements are highly suitable for an implementation in a VLSI technology like CMOS.
In addition, hardware implementations of CNNs take advantage of the inherent parallelism to yield faster solutions than we can obtain in software, though software may be easier to implement.

Why Analog Implementation?

Some advantages of analog circuits for CNNs include:

- Simple and compact circuits: analog circuits are smaller and simpler, thereby enabling increased packing density. The simplicity of CNNs also supports rapid prototyping.
- High speed to hardware ratio and huge computing power: these are faster in terms of speed to the amount of hardware ratio as compared to the digital implementation, leading to enhanced performance. The Analogic (Analog + logic) implementation offers higher speeds and huge computing power than digital implementation.
- Accurate modeling of real time applications: the continuous analog computing and real time processing capability is used for modeling real time applications of CNNs.
- Less power dissipation: these circuits dissipate less power, as the analog conductance devices and processing elements operate at reduced voltage swings. This result in reduced heat transfer associated with the system.

However in these circuits the smallest practical signal in the system is limited because of the matching requirements between its components, temperature variations, and thermal noise, manufacturing tolerances, i.e., these circuits are noise and temperature sensitive.
Why Analog Standard Cell Library?

Similar to the standard cell library approach employed in digital design, in this approach a standard cell library of predefined circuits residing in a software database allows their use as basic building blocks to implement the desired circuitry. We mention the reasons for choosing this approach in realizing various CNN applications.

- Reduced design efforts: implementation efforts for circuits are reduced by the reuse of the same limited library of cells.
- Regularizing and rapid prototyping: the inherent advantage here is that the cells only need to be designed, verified and characterized once for a given technology and they can be reused many times, thus lowering the design cost and saving a lot of time[10], thus promoting rapid prototyping.
- Hierarchical analog design: this approach aids in hierarchical design
- Validated HDL models: HDL models can be developed and validated for the cell library.

In this work, a standard cell library for some CNN image processing applications is realized using a modular and regular analog CMOS transconductance amplifier approach. The details are covered in the following chapters.

2.7 CNN Universal Machine

CNN Universal Machine (CNN-UM) [2, pp 19-21] is an analogic stored program nonlinear array computer, which can efficiently combine analog array operations with local logic. It is fully programmable (executing stored programs) with combined local memory storage.
This CNN UM is an extension of the computing core of a simple CNN, a cell, where a Global Analogic Programming Unit [GAPU] is added to the array to ensure programmability. Its architecture is shown in Figure 2.8 below. This extended architecture consists of additional Local Logical Memory (LLM) and Local Analog Memory (LAM) units which are used to store logic and analog values in each cell, a Local Analog Output Unit (LAOU) and a Local Logic Unit (LLU) which are used to perform analog and logic operations on these stored values, an Optical sensor (OPT) which is used to acquire the sensory input, a Local Communications and Control Unit (LCCU) unit which provides communications between this extended cell and GAPU. [2]

As can be seen in the figure, GAPU has four functional units, the Analog Program Register (APR), the Logic Program Register (LPR), Switch Configuration Register (SCR), and Global Analogic Control Unit (GACU). APR is used to store the analog program instructions, the CNN templates. LPR and SCR local registers contain the control code sequences for the operation of the cell array and GACU is used to store the instruction sequence of the main (analogic) algorithm. [2]
This CNN UM architecture performs difficult image processing operations in ns timescales, much faster than the serial digital systems, with just µs of reprogramming speed [2]. It exhibits supercomputing power for those applications and universality comparable to the Turing machine [2, p.142]. The input-output accuracy is not constrained by the iterative digital processes as the computation here is a non-iterative wave-like process. These analogic CNN computers with the above characteristics even closely mimic the brain-like processing functions along with the anatomy and physiology of many sensory/processing organs [1, pp. 2-3].

Fig. 2.8 Architecture of the CNN Universal Machine [8]
3. CNN Design Procedure

3.1 MATLAB Simulation Procedure

The CNN dynamics equations (2.10) and (2.5) of Chapter 2, revised for MATLAB simulations are given below.

\[
\begin{align*}
\dot{x}_{ij} & = -x_{ij} + A \otimes Y + B \otimes U + z \\
Y(i,j) & = \frac{1}{2} \left( |x_{ij} + 1| - |x_{ij} - 1| \right), \text{ where } i = 1...M; j = 1...N
\end{align*}
\]  

(3.1) (3.2)

Where for \( r=1 \), \( A \) and \( B \) represent the 3×3 feedback and feed forward template matrices respectively. \( X, U \) and \( Y \) denote initial state (time \( t=0 \)), input and output matrix respectively. \( x_{ij} \) denotes the state variable of \( (i, j) \) th cell in the array for \( t>0 \) and \( z \) is a constant which represents the Input bias element. Note that the resistance and capacitance values here are all set equal to 1.

The boundary conditions of a CNN are represented here by matrix elements which are outside the \( M \times N \) array. A 15×15 array is considered and so \( X, U \) and \( Y \) denote 17×17 sized matrices to include the boundary elements. These boundary values are set to zeros. Here each matrix element is considered to correspond to a pixel of a real image with -1 matrix element value indicating the white pixel, +1 indicating the black pixel and intermediate values indicating the gray image pixels. We choose the size 15×15 to match the examples patterns given in [1, 35-97]. This way we can verify that our results are correct.
3.1.1. A Step by Step Procedure for MATLAB Simulations

Step 1. Initializations

M, N which indicate the array size and the template elements A, B and z are initialized according to the given application. U, X matrices are also initialized.

Step 1 is described by the MATLAB code given in Figure 3.1 for a CNN DILATION application.

```matlab
% For CNN DILATION Application
M=15; N=15;

% Initialization
M = zeros(1, 15);
0 = zeros(1, 15);
0 = [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0];
A = zeros(1, 15);
B = zeros(1, 15);
X = zeros(1, 15);

% MATLAB code

% M, N indicate the array size and A, B and z are initialized according to the given application.

% U, X matrices are also initialized.

% Step 1 is described by the MATLAB code given in Figure 3.1 for a CNN DILATION application.

% Fig. 3.1 MATLAB code for CNN DILATION WEST part (1/4)
```

37
Step 2. Calculation of the initial output matrix Y

Output matrix $Y$ is calculated from the given initial state matrix $X$ using the equation:

$$ Y(i, j) = \frac{1}{2} \cdot \text{abs} \left( X(i, j) + 1 \right) - \frac{1}{2} \cdot \text{abs} \left( X(i, j) - 1 \right); $$

(3.2)

Where $1 \leq (i, j) \leq 17$ and $X(i, j)$, $Y(i, j)$ denote the $i^{th}$ row and $j^{th}$ column element of the matrix $X$ and $Y$ respectively.

The MATLAB code in Figure 3.2 describes step 2.

```
for i=1:X+2
    for j=1:N+2
        Y(i,j)=1/2*abs(X(i,j)+1)-1/2*abs(X(i,j)-1);
    end
end
```

Fig. 3.2 MATLAB code for CNN DILATION WEST part (2/4)

Step 3. Calculation of $A \otimes Y + B \otimes U + z = (m)$

As $A \otimes Y$ and $B \otimes U$ represent the sum of the dot products of two vector matrices as described in Chapter 2, we use two for loops to iterate through all the rows and columns of the matrices to calculate the term $A \otimes Y + B \otimes U$. This sum is then added to the constant term $z$ to obtain the sum $A \otimes Y + B \otimes U + z$. Variable $m$ is then assigned to this sum reducing the above state equation to

$$ x_{ij} = - x_{ij} + m $$

(3.4)

Step 4. Solving the equation $x_{ij} = - x_{ij} + m$

This equation is solved by using MATLAB `dsolve` function and is then assigned to a temporary variable `temp` as given below.
temp = dsolve ('D x_y + x_y = m', 'x_y(0) = -X(i, j)');  \hspace{1cm} (3.5)

This gives the following output result

temp = - m + \exp (- t) * (- m - X(i, j));  \hspace{1cm} (3.6)

Where \( X(i, j) \) is the \( (i, j)^{th} \) element of the Initial state matrix \( X \).

**Step5. Calculation of the Output function matrix element \( Y(i, j) \)**

We use the above value of \( temp \) to calculate \( Y(i, j) \) by using the below equation,

\[
Y(i, j) = \frac{1}{2} * \text{abs} (temp + 1) - \frac{1}{2} * \text{abs} (temp - 1)  \hspace{1cm} (3.7)
\]

Figure 3.3 shows the code for steps 3-5.

```
p=0;
for i=2:N+1
    for j=2:N+1
        for k=i-1:i+1
            for l=j-1:j+1
                d=A(k-i+2,l-j+2)*Y(k,l);
                c=B(k-i+2,l-j+2)*U(k,l);
                S(i,j)=p+d+c;
                p=S(i,j);
            end
        end
    end
end
m=p+z;
t=2;
\%temp=dsolve('Dx=x+m','x(0)=-p'); => temp= m*exp(-t)*(m-p)
\%temp=m*(exp(-t)*(-m));  \% for p=0;
Y(i,j)=1/2*abs(temp+1)-1/2*abs(temp-1);
p=0;
end
```

Fig. 3.3 MATLAB code for CNN DILATION WEST part (3/4)
Step 6. Plot the output function matrix Y

Y matrix is plotted in a 3D plot by using the MATLAB function \texttt{surf}. By assigning the ith row and jth column indices of matrix \( Y \) to the matrices \( I \) (i, j) and \( J \) (i, j) respectively, we plot \( I \), \( J \) and \( Y \) in the x, y, z-axis respectively as shown in the function,

\[
surf(I, J, Y, 'EdgeColor', 'black')
\]

(3.8)

Where the 3 arguments of the function \texttt{surf} are matrices \( I \), \( J \), \( Y \) and ‘Edgecolor’ denotes the attribute of the function. The code for this step is given in Figure 3.4.

Similarly the initial state matrix \( X \) and input matrix \( U \) can be plotted.

```
for i=1:17
    for j=1:17
        I(i,j)=i;
        J(i,j)=j;
        surf(I, J, Y, 'EdgeColor', 'black');
    end
end
```

Fig. 3.4 MATLAB code for CNN DILATION WEST part (4/4)
3.2 Analog VLSI implementation of Cellular Neural Networks

As was already mentioned in Chapter 2, due to its advantages, analog VLSI implementation method is chosen here to realize CNNs. There have been many previous analog implementations ([4], [17], [18], [19], [20], [21]), but the implementations we have chosen here are most promising from the point of view of standard cells and programmability. Two different approaches are followed, which are outlined below.

3.2.1 Standard Cell Neuron Component Library Approach

Based on the cell architecture discussed in [3], the equivalent op amp implementation of a simple cell is shown in Figure 3.5 below. Here an op amp forms the basic circuit in realizing a VCCS and thus a CNN cell. The cell uses a sigmoid output function.

Fig. 3.5 An op amp implementation of a simple CNN cell [3]
Since a complete CNN cell with \( r=1 \) neighborhood is modeled by 19 coupling coefficients, we need 19 Voltage Controlled Current Source (VCCS) as discussed in Chapter 2. Each of these VCCS could be achieved by adding an additional resistor to the first op amp of the above cell circuit, which would act in a Differential Summing amplifier configuration with appropriate voltage scaling by resistors.

In [13] and [14], a standard cell library of artificial neuron activation functions was developed. In this approach of realizing a CNN cell, we discuss whether the sigmoid activation function of the library can be used. Since the operational amplifier forms the basis for realizing most activation functions in the library, its design assumes significant importance. The operational amplifier [12] considered in this case was a two-stage amplifier with a class AB push-pull amplifier in the output stage as shown in Figure 3.6. The op amp design is the modified design of op amp in [12], and it requires solving for the sizes of the transistors based on the bias current and gate voltages. The compensation circuitry consisting of compensation capacitor and null- resistor is designed to ensure a stable DC operating condition and to eliminate AC signals fed back from the output of the op amp.
Fig. 3.6 Two-stage operational amplifier with class AB output buffer stage [12]

- **Problems/difficulties of this approach:**
  - The cell architecture given above for implementing a single coupling coefficient has many resistors and as discussed it would require an additional resistor for every other CC. The main problem is in the layout of the resistors which take large chip area and so are not commonly used in the standard CMOS technology. The problem aggravates as we go for a larger network yielding decreased chip density.
  - In the op amp design requiring 12 transistors shown in Figure 3.6, taking all design specifications into consideration for a particular application would lead to a complex design procedure.
  - In addition, it turns out that the capacitance needed for the realization of the CNN is less than the effective value of capacitances used (compensation cap’s) in the op-amp design of standard cell library discussed above and the other parasitic capacitances
obtained through the layout extraction using SPICE. This makes it a more complex approach, as care should be taken about the compensation capacitance value for proper stability and convergence of the network.

Thus, this design approach of using the standard cell library of neuron activation function library (sigmoid, which mainly uses the op amp) in the cell implementation could be used, but would require a redesign of the standard cells for the op amp and the sigmoid activation function according to the CNN cell requirements.

3.2.2 CMOS Inverter Based Transconductance Amplifier Approach

An alternative method is based on a design in [5] where a CNN structure using an analog CMOS transconductance amplifier circuit has been implemented for connected component detector application. Since the major goal is design simplicity, in this approach the design is reduced to the design of few types of transconductance elements. By choosing the appropriate transconductance parameters according to the predetermined coupling coefficients between the neighboring cells, the approach can be easily adapted to various types of CNN applications. Programmability of the transconductance amplifier can be achieved by adjusting the external voltage sources. The following sections discuss the above concept.

3.3 CMOS Transconductance Amplifier

The transconductance amplifier, a Voltage to Current Transducer (VCT) used in [5] is based on a design given in [14]. It resembles a basic CMOS inverter [replacing each
transistor in the CMOS inverter by a p-channel-n-channel pair] but without the matching problems between PMOS and NMOS transistors. It also has the advantage of tunability. Figure 3.7a below shows the transistor schematic and Figure 3.7b shows the input-output characteristic of this four-transistor transconductance element.

Fig. 3.7 a) The transistor schematic b) input-output characteristics of CMOS transconductance element [5]
The equations describing the operation of the transconductance element as described in [14] are given below.

When all transistors operate in their saturation region, it can be easily proven that the output current \( I_{out} = I_a - I_b \) equals

\[
I_{out} = -g_m V_{in} + I_{off}
\]  

(3.9)

Here \( g_m \) is the transconductance parameter and \( I_{off} \) is the offset current given by

\[
g_m = 2k_{eff} \left( V_{G1} + V_{G4} - \sum V_T \right) \]

(3.10)

\[
I_{off} = \frac{g_m}{2} \left( (V_{Tn3} - V_{Tn1}) + (|V_{Tp2}| - |V_{Tp4}|) + (V_{G1} - V_{G4}) \right)
\]

(3.11)

And,

\[
k_{eff} = \frac{k_n k_p}{\left( \sqrt{k_n} + \sqrt{k_p} \right)^2} \]

(3.12)

\[
k_{n,p} = \frac{1}{2} \left[ \mu_{eff} C_{ox} (W/L) \right]_{n,p}, \text{ is the gain factor}
\]

(3.13)

Due to the body effect, the offset current \( I_{off} \) is not equal to zero, but it can be easily eliminated by setting appropriately the bulk voltages \( V_{B4} \) in an n-well process and \( V_{B1} \) in a p-well process.

This simple, fast, linear, tunable four transistor CMOS transconductance element is used in the realization of a variety of continuous-time signal processing components which are
compatible with digital CMOS technologies such as voltage amplifiers, integrators, complete filters, comparators, resistors, OTA’s etc. [14].

This transconductance element (Voltage to Current (VCT) transducer) is chosen as a basic building block for the analog CMOS VLSI implementation of a CNN cell circuit.

3.4 A CNN “Cell” Implementation Using CMOS Transconductance Amplifier

A CNN cell integrated circuit realization with CMOS transconductance element is shown in Figure 3.8. This circuit was reported in [5]. It consists of the bias, input and output voltage-controlled current sources, the state resistor $R_x$, the summation node $V_x$ (where all the input, output and the bias currents are summed) and a block realizing the sigmoid function. At the output of the cell, two transconductance elements are needed to perform the sigmoid type nonlinear transfer characteristic function where resistor $R_y$ acts as the second transconductance element. The individual current sources whose outputs are coupled to the neighbors are driven by the output voltage $V_y$. The input control and the output feedback voltage-controlled current sources are obtained by using the transconductance elements $g_{xm}(i,j;k,l)$ and $g_{xy}(i,j;k,l)$ in their linear region of operation, as the input voltages $V_u$ and the output voltages are bounded by $\pm 1$ V. The appropriate transconductance ($g_m$) parameters can be easily chosen to get the desired coupling coefficient.
Table 3.1 compares the parameters of the above cell circuit of Figure 3.8 with the parameters of the proposed cell circuit model of Figure 2.5, the standard CNN cell circuit of Figure 2.4, and of the MATLAB Simulations.

<table>
<thead>
<tr>
<th>Fig 3.8</th>
<th>Fig 2.5</th>
<th>Fig 2.4</th>
<th>MATLAB</th>
</tr>
</thead>
<tbody>
<tr>
<td>( g_{bias} ), ( V_{bias} )</td>
<td>( I )</td>
<td>( z_{ij} )</td>
<td>( z )</td>
</tr>
<tr>
<td>( g_{xu} ), ( V_{ukl} )</td>
<td>( I_{xu}(i,j;k,l) )</td>
<td>( b_{mn} \cdot u_{kl} )</td>
<td>( B(m,n)U(k,l) )</td>
</tr>
<tr>
<td>( g_{xy} ), ( V_{ykl} )</td>
<td>( I_{xy}(i,j;k,l) )</td>
<td>( a_{mn} \cdot y_{kl} )</td>
<td>( A(m,n)Y(k,l) )</td>
</tr>
<tr>
<td>( g_{yx} ), ( V_x )</td>
<td>( I_{yx} )</td>
<td>( f(x_{ij}) )</td>
<td>( Y(i,j) )</td>
</tr>
<tr>
<td>( V_x )</td>
<td>( V_{xij} )</td>
<td>( x_{ij} )</td>
<td>( x(i,j) )</td>
</tr>
<tr>
<td>( V_y )</td>
<td>( V_{yij} )</td>
<td>( y_{ij} )</td>
<td>( Y(i,j) )</td>
</tr>
<tr>
<td>( g_x )</td>
<td>( 1/R_x )</td>
<td>( 1 )</td>
<td>( 1 )</td>
</tr>
<tr>
<td>( g_y )</td>
<td>( 1/R_y )</td>
<td>( 1 )</td>
<td>( 1 )</td>
</tr>
<tr>
<td>( C_x )</td>
<td>( C_x )</td>
<td>( 1 )</td>
<td>( 1 )</td>
</tr>
</tbody>
</table>

Table 3.1 Comparison of parameters for different CNN cell descriptions
From equation (2.1) we have
\[ g_{xy} = A_{(i,j,k,l)}, \quad g_{zu} = B_{(i,j,k,l)}, \quad g_{bias} = z_{ij} \]  

(3.14)

The stability requirement of Chua [3] as given by equation (2.4) for the above circuit is achieved by equation (3.15).
\[ g_{xy}(i, j; i, j) > g_x \]  

(3.15)

We discuss below the realization of the various building blocks of a CNN cell by CMOS transconductance amplifier. These building blocks are made available in a standard cell library and could be used accordingly in realizing any CNN cell for an application.

### 3.4.1 Resistance Implementation

Resistor implementation is a main problem in VLSI circuits and it is difficult to implement networks with a huge number of resistors as they usually occupy a large chip area, and so their usage is uncommon in standard CMOS technology. To avoid the above problem, the transconductance amplifier is used to implement cell circuit resistors (R_x and R_y), by connecting its input and output terminals as shown in Figure 3.9 below. Desired R is obtained by fixing the gate voltages \( V_{G1} = V_{DD} = 5V \) and \( V_{G4} = V_{SS} = -5V \) and accordingly choosing the appropriate W/L ratios from the equations of \( g_m \) and \( k_{eff} \). Adjustable R is obtained by fixing the W/L ratio and changing the gate bias voltages \( V_{G1} \) and \( V_{G4} \) externally according to the equations of \( g_m \) and \( k_{eff} \) [5].
It can be observed from the equation (3.10) that the transconductance ($g_m$) is directly proportional to $V_{G1}$ and $V_{G4}$. Since the obtained resistance $R$ is inversely proportional to $g_m$ and it is thus inversely proportional to $V_{G1}$ and $V_{G4}$ (called here as $V_{GR1}$ and $V_{GR4}$).

![Resistor implementation by CMOS VCT](image)

**Fig. 3.9 Resistor implementation by CMOS VCT [5]**

### 3.4.2 Positive/Excitatory Coupling Coefficient Implementation

CNNs can be defined with fixed or programmable templates/ coupling coefficients [5]. Fixed coupling coefficients allows the CNN array to perform one fixed- function or a related set of processing functions, thus allowing higher fabrication densities and readily implementable circuit designs. Programmable coupling coefficient allows the CNN array to perform a wide range of applications. In most of the cellular neural network VLSI implementations, the requirement of programmability or adaptability of coupling coefficient is difficult to achieve. Here in the realization of the CNN structure with CMOS transconductance elements approach programmable implementation is achieved by varying the transconductance parameter $g_m$. 
Fixed Positive Coupling Coefficient Implementation

The desired positive coupling coefficients are achieved by choosing the appropriate transconductance \( g_m \) parameter. Fixed positive \( g_m \) is obtained from the fixed positive transconductance amplifier, whose circuit diagram is shown in the Figure 3.10, by choosing the desired gate area W/L ratio appropriately of the transistors (M1 through M4) after having fixed the gate voltages \( V_{G1} = V_{DD} = 5V \) and \( V_{G4} = V_{SS} = -5V \) [5].

![Fig. 3.10 Circuit diagram of fixed positive transconductance amplifier](image)

Programmable Positive Coupling Coefficient Implementation

A programmable positive transconductance amplifier (Figure 3.11) is used to realize a programmable positive coupling coefficient. This is implemented by cascading two transconductance amplifiers, with the first amplifier having gate bias voltages \( V_{G1} \) and \( V_{G4} \) and second amplifier acting as a resistor with its gate voltages of \( V_{GR1} \) and \( V_{GR4} \). As given by the equation (3.11), \( g_m \) can be varied by changing the gate bias voltages \( V_{G1}, V_{G4}, V_{GR1} \) and \( V_{GR4} \) with external voltage sources after fixing the W/L ratios of the transistors [5].
Analytical Expression for Positive Programmable Transconductance in Terms of Gate Voltages

From the above circuit we have

\[ V_{\text{out}} = \left( \frac{1}{g_m} \right) I_{\text{out}} \]

\[ = \frac{1}{g_m} \left( - g_m V_{\text{in}} \right) \]

\[ = \frac{2 k_{\text{eff}}}{2 k_{\text{eff}}} \left( V_{G1} + V_{G4} - \sum V_T \right) V_{\text{in}} \]

\[ \therefore V_{\text{out}} = \frac{V_{G1} + V_{G4} - \sum V_T}{V_{GR1} + V_{GR4} - \sum V_T} V_{\text{in}} \quad (3.11) \]

[As from equations \( I_{\text{out}} = -g_m V_{\text{in}} + I_{\text{off}} \) (\( I_{\text{off}} \) can be made zero)

and \( g_m = 2 k_{\text{eff}} (V_{G1} + V_{G4} - \sum V_T) \)]

From the above equation of \( V_{\text{out}} \), it can be seen that slope (considered here as transconductance) is directly proportional to \( V_{G1} \) and \( V_{G4} \) and inversely proportional to \( V_{GR1} \) and \( V_{GR4} \).
However, in the VLSI programmable implementation of CNN, one of the major design concerns is the wiring required for changing the template coefficients of each cell. For most of the CNN applications such as space invariant template applications, since all the cells have the same coupling coefficients $g_{xx}(i, j; k, l)$ and $g_{xy}(i, j; k, l)$, the wiring problem could be eliminated largely by controlling the transconductance parameter variation of all cells with the same set of external gate voltages [5].

3.4.3 Negative/Inhibitory Coupling Coefficient Implementation

- Fixed Negative Coupling Coefficient Implementation

A fixed negative coupling coefficient is obtained from a fixed negative transconductance amplifier circuit as in Figure 3.12. This is realized by inverting the positive (excitatory) input with a cascaded transconductance element pair (2 and 3 components) as shown in Figure 3.12. Here a fixed positive transconductance amplifier is used for building the cascaded transconductance element pair [5].

![Circuit diagram of fixed negative transconductance amplifier](image)

Fig. 3.12 Circuit diagram of fixed negative transconductance amplifier [5]
Programmable Negative Coupling Coefficient Implementation

A programmable negative coupling coefficient is obtained from a programmable negative transconductance amplifier circuit of Figure 3.13, which is realized by inverting the positive (excitatory) input with a cascaded transconductance element pair as shown in Figure 3.13. Here a programmable positive transconductance amplifier is used for building the above cascaded transconductance element pair.

Fig. 3.13 Circuit diagram of programmable negative transconductance amplifier
4. Standard Cell Library for CNN Target Applications

4.1 Why This Library?

A CNN for image processing, [5] basically maps an input image and an initial state image into a corresponding output image which are all restricted to ±1 V as pixel values. The above CMOS transconductance amplifier approach when used for image processing applications has the images presented as initial voltages to the state capacitor. By choosing the appropriate transconductance parameters according to the predetermined coupling coefficients between the neighboring cells, various types of CNN applications can be realized just by the design of a few types of CMOS transconductance elements. Either based on computer simulation or based on the image processing prominent kernels, these coupling coefficients may be chosen for a particular application.

In [1] a library of CNN templates for image processing is given. Using the transconductance amplifier approach, all of the cells can be implemented using the basic architecture shown in Figure 4.1, which is a slightly modified form of the CNN cell circuit of Figure 3.8 in Chapter 3. The basic idea is to build the appropriate coupling/summing module which is used to couple its own external input and bias, its own output, and the inputs and outputs of the neighboring cells based on transconductance values set by the predefined templates. The initialization module which is used to feed in the initial state voltage is fixed for all the applications, if the value of $g_x$ is set appropriately to meet the stability requirements of the circuit, for a particular output function block, which is a sigmoid here.
As a proof of concept we implement a subset of the functions / applications given in [1]. These are LOGIC NOT, LOGIC OR, SHIFT, DILATION and FILBLACK CNN’s. The remaining functions can be implemented similarly.

As already discussed in Chapter 2, all these applications belong to the uncoupled (scalar) class $C\{A^0, B, z\}$ of CNNs, as $\tilde{A} \equiv 0$ except $A^0$ where $A^0 = a_{0,0}$ and $\tilde{A}$, the rest of the $A$ matrix are called the center and surround feedback component templates respectively. Here we classify the above set of applications into three categories as given below.

- The first set of applications governed by template $\{A^0, B^0, z\}$ with $\tilde{B} = 0$, requires no coupling from its neighboring cell’s input voltages $V_{ukl}$, such as LOGIC OR, LOGIC NOT CNNs.
- The second set of applications requires coupling from neighboring cell input voltages such as SHIFT CNN, which belongs to the class $C\{A^0, \tilde{B}, z\}$ with $B^0 = 0$, and DILATION CNN which belongs to the class $C\{A^0, B, z\}$.
- The third set of applications belongs to the uncoupled autonomous class $C\{A^0, 0, z\}$ of CNNs, having no input coupling since $B = 0$, such as FILBLACK CNN.

The mentioned LOGIC NOT and LOGIC OR CNNs form a complete Boolean set.
4.2 A Modified CNN Cell Circuit

In this modified CNN cell circuit model of Figure 4.1, one can notice two switches (NMOS pass transistor switches) driven by two gate signals, the Start and the Select signals. The Start signal is used to charge the capacitor $C_x$ of the cell core with initial state voltage $V_x$, and the Select signal is used to feed the summed voltages from the summing module into the cell core once the Start signal is turned off. After the initial transient delay equal to the propagation delay of the CNN cell circuit, the CNN settles to a final steady state output value, which is obtained from the output module, thus implementing a particular application. It is to be noted that one has to feed inverted initial state voltage $\overline{V_x}$ instead of $V_x$, as it gets inverted in the circuit, whereas all the other voltages to be fed are not inverted.

![Fig. 4.1 Modified CNN cell circuit using CMOS transconductance amplifier](image)
In realizing the templates / coupling coefficients by the CMOS transconductance amplifier approach, for a CNN cell realization two methods can be followed. One is fixed coupling coefficient method and the other programmable coupling coefficient method. As is already discussed in Chapter 3, in the first method, to realize a desired fixed value of coupling coefficient suitable for a function or a related set of applications we use fixed transconductance (positive/ negative) amplifiers where the gate voltages of the amplifiers are fixed to the supply voltages and a particular W/L ratio of the transistors in the amplifier circuits is chosen. The second method uses programmable transconductance (positive/ negative) amplifiers to realize programmable coupling coefficients, where W/L ratios are fixed and the gate voltages are varied accordingly, by adjusting the external voltage sources.

In this work of realizing a standard cell library of CNNs, the programmable coupling coefficients method is chosen due to the following reasons:

- For each CNN application, in order to meet the stability requirements of the cell circuit one has to set the resistor \( g_x \) which feeds in the summed voltages, according to the condition \( g_{xy} (i, j; i, j) > g_x \), given by equation (3.15) of Chapter 3. In the programmable approach, the same transconductance amplifier used to implement \( g_x \) can easily be adapted to various applications, due to its programmability/ adaptability. But in the fixed coupling coefficient approach W/L ratios of the \( g_x \) amplifier are to be changed for each application, making the design cumbersome and non modular.
• The same cell architecture implemented by programmable transconductance amplifiers can be easily adapted to the set of applications which require the same number and type (positive/ negative) of coupling coefficients without having to redesign the cell circuit by varying the W/L ratios. LOGIC OR and SHIFT CNNs belong to this set as they have 3 positive coupling coefficients, but with different magnitudes even though they belong to different classes as discussed in the Section 4.1. This gives a simple and a modular approach in designing the standard cell library.

4.3 CNNs Requiring No Coupling from Neighboring Cell Inputs

This library of uncoupled CNN applications is governed by the template \( \{A^0, B^0, z\} \) since \( B = 0 \) and requires no coupling from its neighboring cell’s input voltages \( V_{ukl} \).

Applications which are considered here are LOGIC OR and LOGIC NOT CNNs. This section discusses the cell realizations of these applications by the programmable CMOS transconductance amplifier approach.
4.3.1 LOGIC OR CNN Cell Realization

LOGIC OR CNN used in image processing kernel, transforms an input and an initial state image into an output image which is a set union/ logic OR of these two images.

A LOGIC OR CNN template which was previously defined in chapter 2 is given below. It can be seen that this template has 3 coupling coefficients as given by

\[ A_c = g_{xy} = 3g_{eff}, B_c = g_{xu} = 3g_{eff} \text{ and } z = g_{bias} = 2g_{eff}. \]

In order to realize these positive coupling coefficients, we use a programmable positive transconductance amplifier of Chapter 3.

A LOGIC OR CNN cell circuit using CMOS transconductance elements is shown in the Figure 4.3 below. Here a programmable positive transconductance amplifier is used to realize, \( g_{xu} \) (which couples the cells external input voltage \( V_{uc} \) to the cell core), \( g_{xy} \) (which couples the cell output voltage \( V_{yc} \)), \( g_{bias} \) (which couples the external bias voltage \( V_{bias} \)), a resistor \( g_x < 3g_{eff}(A_c) \) and \( g_{yx}, g_y = g_{eff} \) coefficients (which realize a sigmoid activation function). The implementation details are discussed in what follows.
In order to realize the output coupling coefficient $A_c = 3g_{eff}$, after having fixed the widths of the four transistors in CMOS VCT’s and in CMOS resistors according to $W_n = \frac{1}{3}W_p$ in the Figure 3.11 of programmable positive transconductance amplifier, we set the gate bias voltages $V_{G1} = V_{DD} = 5V$, $V_{G4} = V_{SS} = -5V$, $V_{GR1} = +3.2V$ and $V_{GR4} = -3.2V$.

In order to realize the input coupling coefficient $B_c = 3g_{eff}$, also called center feed forward template, same as above $A_c$ value after fixing the widths of the transistors in CMOS VCT’s and in CMOS resistors according to $W_n = \frac{1}{3}W_p$, in the Figure 3.13 of programmable positive transconductance amplifier, we set the gate bias voltages $V_{G1} = V_{DD} = +5V$, $V_{G4} = V_{SS} = -5V$, $V_{GR1} = +3.2V$ and $V_{GR4} = -3.2V$.

Fig. 4.2 A LOGIC OR CNN cell circuit using CMOS transconductance elements
• **z implementation**

For realizing the bias \( z = 2g_{\text{eff}} \) the gate bias voltages of the transistors which are to be set in the positive transconductance amplifier circuit are \( V_{G1} = V_{DD} = +5\text{V} \), \( V_{G4} = V_{SS} = -5\text{V} \), \( V_{GR1} = +3.67\text{V} \) and \( V_{GR4} = -3.67\text{V} \) after fixing the W/L ratios.

• **\( g_s \) implementation**

The cell resistor \( g_s < 3g_{\text{eff}} (A_c) = 0.6g_{\text{eff}} \) (for example) whose value is set to meet cell circuit stability requirements is realized using a programmable positive transconductance amplifier of Figure 3.11, where we set the gate bias voltages \( V_{G1} = 4\text{V} \), \( V_{G4} = -4\text{V} \) and \( V_{GR1} = V_{DD} = 5\text{V} \) and \( V_{GR4} = V_{SS} = -5\text{V} \) after fixing the W/L ratios of the transistors.

These gate bias voltage values which are to be set according to a particular transconductance \( g_m \) value, are obtained from the HSPICE results of positive and negative programmable transconductance amplifiers. Please refer Chapter 5 for details.

**4.3.2 LOGIC NOT CNN Cell Realization**

LOGIC NOT CNN is used to transform an input image and a zero initial state image into an output image which is an inversion or Logic Not of the input image.
Given below is the template for LOGIC NOT CNN which was previously defined in chapter 2. It can be seen that this template has 2 coupling coefficients given by $A_c = g_{xy} = g_{eff}$ and $B_c = g_{xu} - 2 g_{eff}$. In order to realize these positive and negative coupling coefficients, we use programmable positive and negative transconductance amplifiers respectively as discussed in Chapter 3.

A LOGIC NOT CNN cell circuit using CMOS transconductance elements is shown in the Figure 4.2 below. Here a programmable positive transconductance amplifier is used to realize $g_{xy}$, which couples the cells output voltage $V_{yc}$, a resistor $g_x < g_{eff} (A_c)$ and $g_{yx}g_y = g_{eff}$ coefficients which realizes a sigmoid activation function. A programmable negative transconductance amplifier is used to realize $g_{xu}$, which couples the cells external input voltage $V_{uc}$. The details are discussed next.

![Fig. 4.3 A LOGIC NOT CNN cell circuit using CMOS transconductance elements](image)
• **A<sub>c</sub> implementation**

In order to realize the output coupling coefficient \( A_c = g_{eff} \), after having fixed the widths of the four transistors in CMOS VCT’s and in CMOS resistor according to \( W_n = \frac{1}{3} W_p \) in the Figure 3.11 of programmable positive transconductance amplifier, we set the gate bias voltages \( V_{G1} = V_{GR1} = V_{DD} = 5V \) and \( V_{G4} = V_{GR4} = V_{SS} = -5V \).

• **B<sub>c</sub> implementation**

In order to realize the input coupling coefficient \( B_c = -2g_{eff} \), after having fixed the widths of the four transistors in CMOS VCT’s and in CMOS resistors according to \( W_n = \frac{1}{3} W_p \), in the Figure 3.13 of programmable negative transconductance amplifier, we set the gate bias voltages \( V_{G1} = V_{DD} = 5V \), \( V_{G4} = V_{SS} = -5V \), \( V_{GR1} = 3.3V \) and \( V_{GR4} = -3.3V \).

• **g<sub>x</sub> implementation**

The cell resistor \( g_x < g_{eff} (A_c) = 0.5g_{eff} \) whose value is set to meet cell circuit stability requirements is realized using a programmable positive transconductance amplifier of Figure 3.11, where we set the gate bias voltages \( V_{G1} = 3.7V \), \( V_{G4} = -3.7V \) and \( V_{GR1} = V_{DD} = 5V \) and \( V_{GR4} = V_{SS} = -5V \) after fixing the widths of the transistors.

These gate bias voltage values which are to be set according to a particular transconductance \( g_m \) value, are obtained from the HSPICE results of positive and negative programmable transconductance amplifiers. Please refer Chapter 5 for details.
4.4 CNNs Requiring Coupling from Neighboring Cells Inputs

Realization of the uncoupled set of applications requiring coupling from its neighboring cell’s input voltages such as a SHIFT CNN belonging to class $C\{\mathbf{A}^0, \mathbf{B}, z\}$ with $\mathbf{B}^0 = \mathbf{0}$ and DILATION CNN belonging to class $C\{\mathbf{A}^0, \mathbf{B}, z\}$ by CMOS transconductance amplifier approach is discussed here.

4.4.1 SHIFT WEST CNN Cell Realization

This CNN transforms the input image $V_B$ and a zero initial state image $V_x$ into an output image $V_y$, which is shifted version of input image by one unit west.

| A=  
<table>
<thead>
<tr>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>0</td>
</tr>
</tbody>
</table>

| B=  
<table>
<thead>
<tr>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>1</td>
</tr>
</tbody>
</table>

| Z=0 |

The table shows a SHIFT WEST CNN template as was previously defined in Chapter 2. It can be seen that this SHIFT WEST template has 2 positive coupling coefficients as given by $A_c = g_{xy} = g_{\text{eff}}$, $B_w = g_{xuw} = g_{\text{eff}}$ which are realized using programmable positive transconductance amplifier of Chapter 3, where as $B_c = g_{xuc} = 0$.

A SHIFT WEST CNN cell circuit using CMOS transconductance elements is shown in the Figure 4.4 below. Here a programmable positive transconductance amplifier is used to realize, $g_{xuc} = 0$ (which couples the cells external input voltage $V_{uc}$ to the cell core), $g_{xuw} = g_{\text{eff}}$ (which is used to couple the
input voltage $V_{ue}$ of its east neighbor to the cell core), $g_{xy}$ (which couples the cell output voltage $V_{yc}$), a resistor $g_x < g_{eff}(A_c)$ and $g_{yx}, g_y = g_{eff}$ coefficients (which realize a sigmoid activation function). It is to be noted in the figure that the cell performs shift west operation, by taking the input voltage $V_{ue}$ of its east neighbor, where as the coupling coefficient is termed as $g_{xue}$, in order to suggest its functioning as a SHIFT WEST. The implementation details are discussed in what follows.

![Fig. 4.4 A SHIFT WEST CNN cell circuit using CMOS transconductance elements](image)

- **$A_c$ implementation (same as LOGIC NOT CNN $A_c$)**

In order to realize the positive output feedback coupling coefficients $A_c$, after having fixed the widths of four transistors in CMOS VCT’s and in CMOS resistor according to $W_n = \frac{1}{3} W_p$ in the Figure 3.11 of programmable positive transconductance amplifier we set the gate bias voltages $V_{G1} = V_{GR1} = V_{DD} = 5V$ and $V_{G4} = V_{GR4} = V_{SS} = -5V$. 

66
• **B\textsubscript{w} implementation**

The input west feed forward coupling coefficient \( B\textsubscript{w} \) is realized in the same way as the above \( A\textsubscript{c} \).

• **\( g\textsubscript{s} \) implementation** (Same as LOGIC NOT CNN \( g\textsubscript{s} \))

The cell resistor \( g\textsubscript{s} < g\textsubscript{eff} (A\textsubscript{c}) = 0.5g\textsubscript{eff} \) whose value is set to meet cell circuit stability requirements is realized using a programmable positive transconductance amplifier of Figure 3.11, where we set the gate bias voltages \( V\textsubscript{G1} = 3.7V, V\textsubscript{G4} = -3.7V \) and \( V\textsubscript{GR1} = V\textsubscript{DD} = 5V \) and \( V\textsubscript{GR4} = V\textsubscript{SS} = -5V \) after fixing the widths of the transistors.

### 4.4.2 SHIFT EAST CNN Cell Realization

This CNN transforms the input image \( V\textsubscript{u} \) and a zero initial state image \( V\textsubscript{x} \) into an output image \( V\textsubscript{y} \), which is shifted version of input image by one unit east.

The table shows a SHIFT EAST CNN template as was previously defined in Chapter 2. It can be seen that this SHIFT EAST template has 2 positive coupling coefficients as given by \( A\textsubscript{c} = g\textsubscript{xy} = g\textsubscript{eff} \), \( B\textsubscript{c} = g\textsubscript{xuc} = g\textsubscript{eff} \) which are realized using programmable positive transconductance amplifier of Chapter 3, where as \( B\textsubscript{c} = g\textsubscript{xuc} = 0 \).

A SHIFT EAST CNN cell circuit using CMOS transconductance elements is shown in the Figure 4.5 below. Here a programmable positive
transconductance amplifier is used to realize, \( g_{xue} = 0 \) (which couples the cells external input voltage \( V_{uc} \) to the cell core), \( g_{xue} = g_{eff} \) (which is used to couple the input voltage \( V_{uw} \) of its west neighbor to the cell core), \( g_{xy} \) (which couples the cell output voltage \( V_{yc} \)), a resistor \( g_s < g_{eff} \) (\( A_c \)) and \( g_{xy}, g_y = g_{eff} \) coefficients (which realize a sigmoid activation function). It is to be noted in the figure that the cell performs shift east operation, by taking the input voltage \( V_{uw} \) of its west neighbor, where as the coupling coefficient is termed as \( g_{xue} \), in order to suggest its functioning as a SHIFT EAST CNN. Its implementation details are discussed below. Thus, a SHIFT EAST CNN and a SHIFT WEST CNN differ in what direction the neighbor’s input voltage is fed.

![Fig. 4.5 A SHIFT EAST CNN cell circuit using CMOS transconductance elements](image)

- **\( A_c \) implementation (same as SHIFT WEST CNN \( A_c \))**

In Figure 3.11 of programmable positive transconductance amplifier we set the gate bias voltages \( V_{G1} = V_{GR1} = V_{DD} = 5V \) and \( V_{G4} = V_{GR4} = V_{SS} = -5V \) after fixing the W/L ratios.
- **$B_e$ implementation**

The input east feed forward coupling coefficient $B_e$ is realized in the same way as the above $A_e$.

- **$g_x$ implementation (same as SHIFT WEST CNN $g_x$)**

The cell resistor $g_x < g_{eff} (A_e) = 0.5g_{eff}$ is realized using a programmable positive transconductance amplifier of Figure 3.11, where we set the gate bias voltages $V_{G1} = 3.7V$, $V_{G4} = -3.7V$ and $V_{GR1} = V_{DD} = 5V$ and $V_{GR4} = V_{SS} = -5V$ after fixing the widths of the transistors.

### 4.4.3 DILATION WEST CNN Cell Realization

This CNN transforms the input image $V_u$ and a zero initial state image $V_x$ into an output image $V_y$, which is dilated (expanded) version of input image by one unit west.

![Table]

The table shows a DILATION WEST CNN template as was previously defined in Chapter 2. It can be seen that this template has 4 positive coupling coefficients as given by, $A_e = g_{xy} = 2g_{eff}$, $B_w = g_{xwn} = g_{eff}$ and $z = g_{bias} = 1.5g_{eff}$ which are realized using programmable positive transconductance amplifier of Chapter 3.

A DILATION WEST CNN cell circuit using CMOS transconductance elements is shown in the Figure 4.6 below. Here a programmable positive transconductance amplifier is used to realize $g_{xuc} = g_{eff}$ (which couples the cells external
input voltage $V_{ue}$ to the cell core), $g_{xuw} = g_{eff}$ (which is used to couple the input voltage $V_{ue}$ of its east neighbor to the cell core), $g_{xy}$ (which couples the cell output voltage $V_{yc}$), $g_{bias}$ (which couples the external bias voltage $V_{bias}$), a resistor $g_x < 2g_{eff} (A_c)$ and $g_{yx}, g_y = g_{eff}$ coefficients (which realize a sigmoid activation function). It is to be noted in the figure that the cell performs dilation west operation, by taking the input voltage $V_{ue}$ of its east neighbor, where as the coupling coefficient is termed as $g_{xuw}$, in order to suggest its functioning as a DILATION WEST CNN. Its implementation details are discussed below.

• **$A_c$ implementation**

Inorder to realize the positive output feedback coupling coefficients $A_c$, after having fixed the widths of four transistors in CMOS VCT’s and in CMOS resistor according to $W_n = \frac{1}{3} W_p$ in the Figure 3.11 of programmable positive transconductance amplifier we set the gate bias voltages $V_{G1} = V_{DD} = +5V$, $V_{G4} = V_{SS} = -5V$, $V_{GR1} = +3.67V$ and $V_{GR4} = -3.67V$.

• **$B_c$ and $B_w$ implementation (same as SHIFT WEST CNN $A_c$)**

The feed forward coupling coefficients $B_c$ which feeds external input voltage $V_{ue}$ and $B_w$ which feed east neighbors input $V_{ue}$ are realized in the same way as the SHIFT WEST CNN $A_c$ by setting $V_{G1} = V_{GR1} = V_{DD} = 5V$ and $V_{G4} = V_{GR4} = V_{SS} = -5V$ in Figure 3.11, after fixing the W/L ratios.
• \textit{z implementation}

For realizing the bias $z = 1.5g_{\text{eff}}$ the gate bias voltages of the transistors which are to be set in the positive transconductance amplifier circuit are $V_{G1} = V_{DD} = +5V$, $V_{G4} = V_{SS} = -5V$, $V_{GR1} = +4V$ and $V_{GR4} = -4V$ after fixing the W/L ratios.

• \textit{g_s implementation (same as SHIFT WEST CNN $g_s$)}

The cell resistor $g_s < g_{\text{eff}} (A_c) = 0.5g_{\text{eff}}$ whose value is set to meet cell circuit stability requirements is realized using a programmable positive transconductance amplifier of Figure 3.11, where we set the gate bias voltages $V_{G1} = 3.7V$, $V_{G4} = -3.7V$ and $V_{GR1} = V_{DD} = 5V$ and $V_{GR4} = V_{SS} = -5V$ after fixing the widths of the transistors in the circuit.

![Fig. 4.6 A DILATION WEST CNN cell circuit using CMOS transconductance elements](image-url)
4.5 CNNs Requiring No Input Coupling

This library of applications belongs to the uncoupled autonomous class of CNNs governed by template $\{A^0, 0, z\}$. It has no input coupling since $B = 0$, such as a FILBLACK CNN. Its realization is discussed in what follows.

4.5.1 FILBLACK CNN Cell Realization

A FILBLACK CNN is used to transform an initial state image and a zero input image into an output image which is a completely black version of initial state image.

\[
A = \begin{bmatrix}
0 & 0 & 0 \\
0 & 2 & 0 \\
0 & 0 & 0 \\
\end{bmatrix}
\]

\[
B = \begin{bmatrix}
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
\end{bmatrix}
\]

\[z = 4\]

A FILBLACK CNN template which was previously defined in chapter 2 is given below. It can be seen that this template has 2 coupling coefficients as given by $A^c = g_{xy} = 2g_{eff}$ and $z = g_{bias} = 4g_{eff}$. In order to realize these positive coupling coefficients, we use a programmable positive transconductance amplifier of Chapter 3.

A FILBLACK CNN cell circuit using CMOS transconductance elements is shown in the Figure 4.7 below. Here a programmable positive transconductance amplifier is used to realize $g_{xy}$ (which couples the cell output voltage $V_{yc}$), $g_{bias}$ (which couples the external bias voltage $V_{bias}$), a resistor $g_x < 2g_{eff} (A_z)$ and $g_{xs} g_y = g_{eff}$ coefficients (which realize a sigmoid activation function). The implementation details are discussed in what follows.
• **A<sub>c</sub> implementation**

In order to realize the output coupling coefficient \( A_c \), after having fixed the widths of the four transistors in CMOS VCT’s and in CMOS resistors according to \( W_a = \frac{1}{3} W_p \) in the Figure 3.11 of programmable positive transconductance amplifier, we set the gate bias voltages \( V_{G1} = V_{DD} = 5V \), \( V_{G4} = V_{SS} = -5V \), \( V_{GR1} = 3.67V \) and \( V_{GR4} = -3.67V \).

• **z implementation**

For realizing the bias \( z = 4g_{eff} \) the gate bias voltages of the transistors which are to be set in the positive transconductance amplifier circuit are \( V_{G1} = V_{DD} = +5V \), \( V_{G4} = V_{SS} = -5V \), \( V_{GR1} = +3V \) and \( V_{GR4} = -3V \) after fixing the W/L ratios.

• **g<sub>x</sub> implementation**

The cell resistor \( g_x < 2g_{eff} (A_c) = 0.6g_{eff} \) (for example), whose value is set to meet cell circuit stability requirements is realized using a programmable positive transconductance amplifier of Figure 3.11, where we set the gate bias voltages \( V_{G1} = 4V \), \( V_{G4} = -4V \) and \( V_{GR1} = V_{DD} = 5V \) and \( V_{GR4} = V_{SS} = -5V \) after fixing the W/L ratios of the transistors.
Fig. 4.7 A FILBLACK CNN cell circuit using CMOS transconductance elements

4.6 Summary of Standard CNN Cell Library Realizations

Table 4.1 below shows the summary of realizations of the Standard CNN Cell Library, including its templates and gate voltages required to be set in the transconductance amplifiers for the templates realization.
<table>
<thead>
<tr>
<th>CNN Applications</th>
<th>Templates and Resistor</th>
<th>Programmable CMOS Transconductance Amplifier Realization</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOGIC OR</td>
<td>$A_c = B_c = 3 \ g_{eff}$ with $g_c = 0.6 \ g_{eff}$</td>
<td>$V_{G1}= V_{DD}=5V, V_{G4}=V_{SS}=-5V, V_{GR4}=-3.2V$</td>
</tr>
<tr>
<td></td>
<td>$z_c = 2 \ g_{eff}$</td>
<td>$V_{G1}=V_{DD}=5V, V_{G4}=V_{SS}=-5V, V_{GR1}=3.67V, V_{GR4}=-3.67V$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$V_{G1}=4V, V_{G4}=-4V, V_{GR1}=V_{DD}=5V$ and $V_{GR4}=V_{SS}=-5V$</td>
</tr>
<tr>
<td>LOGIC NOT</td>
<td>$A_c = g_{eff}$ with $g_c = 0.5 \ g_{eff}$</td>
<td>$V_{G1}= V_{GR1}=V_{DD}=5V$ and $V_{G4}= V_{GR4}=V_{SS}=-5V$</td>
</tr>
<tr>
<td></td>
<td>$B_c = -2 \ g_{eff}$</td>
<td>$V_{G1}=V_{DD}=5V, V_{G4}=V_{SS}=-5V, V_{GR1}=3.3V, V_{GR4}=-3.3V$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(A programmable negative transconductance amplifier is used here, where as all other realizations require a programmable positive transconductance amplifier)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$V_{G1}=3.7V, V_{G4}=-3.7V, V_{GR1}=V_{DD}=5V, V_{GR4}=V_{SS}=-5V$</td>
</tr>
<tr>
<td>SHIFT WEST</td>
<td>$A_c = B_a = g_{eff}$ with $g_c = 0.5 \ g_{eff}$</td>
<td>$V_{G1}= V_{GR1}=V_{DD}=5V$ and $V_{G4}= V_{GR4}=V_{SS}=-5V$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$V_{G1}=3.7V, V_{G4}=-3.7V, V_{GR1}=V_{DD}=5V, V_{GR4}=V_{SS}=-5V$</td>
</tr>
<tr>
<td>SHIFT EAST</td>
<td>$A_c = B_e = g_{eff}$ with $g_c = 0.5 \ g_{eff}$</td>
<td>$V_{G1}= V_{GR1}=V_{DD}=5V$ and $V_{G4}= V_{GR4}=V_{SS}=-5V$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$V_{G1}=3.7V, V_{G4}=-3.7V, V_{GR1}=V_{DD}=5V, V_{GR4}=V_{SS}=-5V$</td>
</tr>
<tr>
<td>DILATION</td>
<td>$A_c = 2 \ g_{eff}$</td>
<td>$V_{G1}= V_{DD}=5V, V_{G4}= V_{SS}=-5V, V_{GR1}=3.67V, V_{GR4}=-3.67V$</td>
</tr>
<tr>
<td></td>
<td>$B_c = B_a = g_{eff}$</td>
<td>$V_{G1}=V_{GR1}=V_{DD}=5V$ and $V_{G4}= V_{GR4}=V_{SS}=-5V$</td>
</tr>
<tr>
<td></td>
<td>$z_c = 1.5 \ g_{eff}$</td>
<td>$V_{G1}=V_{DD}=+5V, V_{G4}= V_{SS}=-5V, V_{GR1}=+4V, V_{GR4}=-4V$</td>
</tr>
<tr>
<td></td>
<td>with $g_c = 0.6 \ g_{eff}$</td>
<td>$V_{G1}=4V, V_{G4}=-4V, V_{GR1}=V_{DD}=5V$ and $V_{GR4}=V_{SS}=-5V$</td>
</tr>
<tr>
<td>FILBLACK</td>
<td>$A_c = 2 \ g_{eff}$ with $g_c = 0.6 \ g_{eff}$</td>
<td>$V_{G1}= V_{DD}=5V, V_{G4}= V_{SS}=-5V, V_{GR1}=3.67V, V_{GR4}=-3.67V$</td>
</tr>
<tr>
<td></td>
<td>$z_c = 4 \ g_{eff}$</td>
<td>$V_{G1}=V_{DD}=+5V, V_{G4}= V_{SS}=-5V, V_{GR1}=+3V, V_{GR4}=-3V$</td>
</tr>
<tr>
<td></td>
<td>with $g_c = 0.6 \ g_{eff}$</td>
<td>$V_{G1}=4V, V_{G4}=-4V, V_{GR1}=V_{DD}=5V$ and $V_{GR4}=V_{SS}=-5V$</td>
</tr>
</tbody>
</table>

Table 4.1 Summary of standard CNN cell library realizations
5. Results and Simulations

5.1 Chapter Outline

Here is the brief outline of topics of this chapter. These are discussed in detail in the following sections.

- MATLAB Simulations (5.2)

Various CNN applications have been realized in MATLAB (7.0.1) based on different A and B CNN templates. Applications simulated include LOGICOR, LOGICNOT, SHIFT DILATION and FILBLACK CNNs. These results are shown in section 5.2.

- MAGIC Layouts and HSPICE Simulation Results (5.3)

Here we present the MAGIC layouts and HSPICE simulation results of the basic CNN cell components and of the various CNNs of the standard cell library, which are obtained from these basic components. These are described in detail in sections 5.3.1 and 5.3.2.

  - Basic CNN Cell Components Library (5.3.1)

A component library of the various blocks of a CNN cell, based on analog CMOS transconductance amplifier approach, is laid out in MAGIC (7.1) layout editor with –T SCN3ME_SUBM.30 (AMI 0.5µ) technology. These layouts include a transconductance amplifier, a resistor, and programmable positive and negative transconductance amplifiers. HSPICE (X-2005.09-SP1) results of the above components are also discussed. The HSPICE model file used is Level 49.
- Standard Cell Library of CNNs (5.3.2)

This section includes the Magic layouts and HSPICE simulations of the standard cell library of CNNs for a single cell and a 1×3 array of cells for the above mentioned image processing applications. Sections 5.3.2.1, 5.3.2.2 and 5.3.3.3 cover these topics in detail.
5.2 MATLAB Simulations

The set of revised equations (3.1), (3.2) governing the CNN dynamics described in Chapter 3 has been employed to realize a CNN in MATLAB (Version 7.0.1). For the CNN implementation templates described in Chapter 2, MATLAB simulation results are shown below.

5.2.1 LOGIC OR CNN

LOGIC OR CNN is used to transform an input and an initial state image into an output image which is a set union/ logic OR of these two images.

In the MATLAB simulations the image sizes chosen are 15 ×15 array of +1 or -1 magnitude. Figure 5.1 shows a) input image U (P1) and b) initial state X (P2), and c) the obtained output is (P1 U P2), a set union of these two images, thus verifying its functionality.

![Fig. 5.1 LOGIC OR / Set union/ Disjunction CNN](image-url)
5.2.2 LOGIC NOT CNN

A LOGIC NOT CNN is used to transform an input image and a zero initial state image into an output image which is an inversion or a Logic Not of the input image, where black pixel turns white and vice versa obeying the local rules mentioned in Chapter 2.

In the MATLAB simulations the image sizes chosen are $15 \times 15$ array of +1 or -1 magnitude. Figure 5.2 shows a) input image $U (P_1)$ and b) initial state $X (P_2 = 0)$, and c) the obtained output is $\bar{P_1}$, a complement of $P_1$, thus verifying its functionality.

![Fig. 5.2 LOGIC NOT/ Set complementation CNN](image)
5.2.3 SHIFT WEST CNN

This CNN transforms the input image and a zero initial state image into an output image $V_y$, which is shifted version of input image by one unit west.

In the MATLAB simulations the image sizes chosen are $15 \times 15$ array of +1 or -1 magnitude. Figure 5.3 shows a) input image, b) initial state image and c) output image transient at $t=1$ which is in accordance with the global task of transforming an input image $U(t) = P$ and a Zero Initial State $X(0) = 0$ into an Output Binary Image $Y(t) \rightarrow Y(\infty) = P(x-\alpha, y-\beta)$ i.e. $P$ shifted by one pixel along the eight compass directions $(\alpha, \beta) \in \{-1,0,1\}$ and $(x, y)$ being the Cartesian coordinates, according to the local rules mentioned in Chapter 2. Here the compass direction considered is Shift west. One could thus verify the functionality.

![Fig. 5.3 SHIFT WEST/ Translation by one pixel unit CNN](image-url)
5.2.4 DILATION WEST CNN

This CNN transforms the input image \( U \) and a zero initial state image \( X \) into an output image \( V \), which is dilated (expanded) version of input image by one unit west, according to the global and local rules mentioned in Chapter 2. The Output settles at \( t=2 \) time units. The images of \( 15 \times 15 \) size with +1 or -1 magnitudes are considered. Figure 5.4 verifies this function.

![Fig. 5.4 DILATION/ Grow-until-it-fits CNN](image-url)
5.2.5 FILBLACK CNN

A FILBLACK CNN is used to transform an initial state image (X) and a zero input image (U) into an output image (Y) which is a completely black version of initial state image.

Figure 5.5 shows a) input image b) initial state image c) output image transients at t=1.7, obtained in accordance to the global and local rules mentioned in Chapter 2. The images of 15 ×15 size with +1 or -1 magnitudes are considered, thus verifying this function.

Fig. 5.5 FILBLACK/ Gray-scale to black CNN
5.3 MAGIC Layouts and HSPICE Simulation Results

Standard cell library approach of CNN cells is followed to regularize the design inline with the widely used standard cell technique in digital design. Here we present the MAGIC Layouts and HSPICE Simulation Results of the basic CNN cell components and of the various CNNs of the Standard Cell library, which are obtained from these basic components. HSPICE (X-2005.09-SP1) simulations are carried out from HSPICE file which is extracted from the layouts for the above mentioned CNN cell components and cell library. The model file used is HSPICE Level 49. These results are shown in this section.

5.3.1 Basic CNN Cell Component Library

In this section we discuss the MAGIC layouts and HSPICE results of the basic components of a CNN cell realized using the CMOS transconductance amplifier approach as discussed in Chapter 3.

- Transconductance Amplifier

Transconductance amplifier forms the basic component of the cell library. The other cells in the library discussed in this section are derived from this basic structure. Its Magic layouts and HSPICE plots are shown next.

In the Figure 5.6 we give the MAGIC Layout for the CMOS transconductance amplifier, whose circuit details can be found in Chapter 3. The device sizes mentioned below are NMOS: $33\lambda \times 34\lambda$, PMOS: $100\lambda \times 34\lambda$. The technology is 0.5u AMIc5
SCN3ME_SUBM.30 and so the $\lambda = 0.3\mu$. It has been laid out for optimal area. Here the gate voltages of the NMOS and PMOS transistors, $V_{G1}$ and $V_{G4}$ respectively, are used to vary the transconductance ($g$) of the amplifier. W/L ratios of the transistors could also be varied to obtain different $g$ values.

![Fig. 5.6 Transconductance amplifier Magic layout](image)

Shown in Figure 5.7 are the HSPICE simulations of the voltage transfer characteristics for the amplifier, at $V_{G1}=5$, $V_{G4}=-5$. The slope, given by the value $\left(\frac{\Delta V_y}{\Delta V_x}\right)$ is equal to -28.5. It could be seen in the figure that the transconductance of the amplifier, given by the value $\left(\frac{\Delta I_y}{\Delta V_x}\right)$ is equal to the slope of the graph, since the magnitudes of the y-axis currents ($I_y$) are equal to that of the voltages ($V_x$) as can be seen in the Figure 5.7 below.
Fig. 5.7 HSPICE voltage transfer characteristics of the transconductance amplifier

at $V_{G1}=5$, $V_{G4}=-5$

Table 5.1 shows the transconductances for various gate voltages.

<table>
<thead>
<tr>
<th>$V_{G1}$</th>
<th>$V_{G4}$</th>
<th>Transconductance</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>-5</td>
<td>-28.5</td>
</tr>
<tr>
<td>4</td>
<td>-4</td>
<td>-25.9</td>
</tr>
<tr>
<td>3</td>
<td>-3</td>
<td>-24.4</td>
</tr>
<tr>
<td>2</td>
<td>-2</td>
<td>-18.0</td>
</tr>
<tr>
<td>1</td>
<td>-1</td>
<td>-5.24</td>
</tr>
</tbody>
</table>

Table 5.1 Transconductance variation with gate voltages
Resistor

In Figure 5.8 we show the resistor circuit obtained from the transconductance amplifier of chapter 3. This layout is obtained from the above transconductance cell by shorting the $V_{in}$ terminal to the $V_{out}$ terminal.

Figure 5.8 Resistor Magic layout a) top level and b) expanded design

Figure 5.9 shows the HSPICE Simulations of the voltage vs. current characteristics for the resistor at $V_{GR1}=5$, $V_{GR4}=-5$. As can be seen, this resistor is linear and its value is given by the slope of the curve ($\Delta V_y / \Delta I_x$), which is equal to -12k, indicating that it is negative resistance.
Fig. 5.9 HSPICE VI characteristics of the resistor at VG1=5, VG4=-5

Table 5.2 shows the resistances for various gate voltages.

<table>
<thead>
<tr>
<th>$V_{GR1}$</th>
<th>$V_{GR4}$</th>
<th>Resistances (Ω)</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>-5</td>
<td>12k</td>
</tr>
<tr>
<td>4</td>
<td>-4</td>
<td>19k</td>
</tr>
<tr>
<td>3</td>
<td>-3</td>
<td>53k</td>
</tr>
<tr>
<td>2</td>
<td>-2</td>
<td>252k</td>
</tr>
<tr>
<td>1</td>
<td>-1</td>
<td>641k</td>
</tr>
</tbody>
</table>

Table 5.2 Resistance variation with gate voltages
Programmable Positive Transconductance Amplifier

Figure 5.10 shows the layout of programmable positive transconductance amplifier obtained from the transconductance amplifier as described in Chapter 3. Here by choosing a particular W/L ratio and varying the gate voltages \( V_{G1} \), \( V_{GR1} \), \( V_{G4} \) and \( V_{GR4} \), one can obtain different positive transconductances (\( g \)) values needed to realize various positive coupling coefficients. Thus, it forms a basic building block in the realizations of the various CNNs of the library discussed in the next section.

Fig. 5.10 Programmable positive transconductance amplifier Magic layout a) top level and b) expanded design
Figure 5.11 gives the HSPICE Simulations of the voltage transfer characteristics for the variable positive amplifier at gate voltages of $V_{G1}=5$, $V_{GR1}=5$ for a particular W/L ratio of the above mentioned NMOS and PMOS transistors. The slope is given by the value $(\Delta V_y / \Delta V_x) = -0.987$. 

![HSPICE voltage transfer characteristics of the programmable positive transconductance amplifier at a particular W/L ratio](image)

Fig. 5.11 HSPICE voltage transfer characteristics of the programmable positive transconductance amplifier at a particular W/L ratio
Table 5.3 shows the positive transconductances obtained from this amplifier for various gate voltages. Note that $V_{G4} = -V_{G1}$, $V_{GR4} = -V_{GR1}$ and so these are not mentioned in the table.

<table>
<thead>
<tr>
<th>$V_{G1}$</th>
<th>$V_{GR1}$</th>
<th>Positive Transconductance</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.7</td>
<td>5</td>
<td>0.5</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>0.62</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>0.987</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>1.56</td>
</tr>
<tr>
<td>5</td>
<td>3.7</td>
<td>1.92</td>
</tr>
<tr>
<td>5</td>
<td>3.65</td>
<td>2.05</td>
</tr>
<tr>
<td>5</td>
<td>3.2</td>
<td>2.98</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>3.95</td>
</tr>
<tr>
<td>5</td>
<td>2.97</td>
<td>4.53</td>
</tr>
</tbody>
</table>

Table 5.3 Positive transconductance variation with gate voltages

- **Programmable Negative Transconductance Amplifier**

Figure 5.12 shows the layout of programmable negative transconductance amplifier obtained from the transconductance amplifier as described in Chapter 3. Here by choosing a particular W/L ratio and varying the gate voltages $V_{G1}$, $V_{GR1}$, $V_{G4}$ and $V_{GR4}$, one can obtain different negative transconductances ($g$) values needed to realize various
negative coupling coefficients. Thus, it forms a basic building block in the realizations of the various CNNs of the library discussed in the next section.

Figure 5.13 gives the HSPICE Simulations of the voltage transfer characteristics for the programmable negative amplifier at gate voltages of $V_{G1}=5$, $V_{GR1}=5$ for a particular W/L ratio of the above mentioned NMOS and PMOS transistors. The slope is given by the value $(\Delta V_y/\Delta V_x) = 0.945$. 

Fig. 5.12 Programmable negative transconductance amplifier Magic layout a) top level and b) expanded design

---

91
Fig. 5.13 Transfer characteristics of the programmable negative transconductance amplifier at a particular W/L ratio
Table 5.4 shows the negative transconductances for various gate voltages

<table>
<thead>
<tr>
<th>$V_{G1}$</th>
<th>$V_{GR1}$</th>
<th>Negative Transconductance(-)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>5</td>
<td>0.196</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>0.487</td>
</tr>
<tr>
<td>5</td>
<td>4.2</td>
<td>0.806</td>
</tr>
<tr>
<td>5</td>
<td>4.0</td>
<td>0.945</td>
</tr>
<tr>
<td>5</td>
<td>3.9</td>
<td>1.02</td>
</tr>
<tr>
<td>5</td>
<td>3.7</td>
<td>1.23</td>
</tr>
<tr>
<td>5</td>
<td>3.5</td>
<td>1.54</td>
</tr>
<tr>
<td>5</td>
<td>3.3</td>
<td>2.05</td>
</tr>
<tr>
<td>5</td>
<td>3.0</td>
<td>3.58</td>
</tr>
<tr>
<td>5</td>
<td>2.9</td>
<td>4.74</td>
</tr>
<tr>
<td>5</td>
<td>2.8</td>
<td>5.9</td>
</tr>
<tr>
<td>5</td>
<td>2.7</td>
<td>8</td>
</tr>
<tr>
<td>5</td>
<td>2.69</td>
<td>8.38</td>
</tr>
<tr>
<td>5</td>
<td>2.68</td>
<td>9.75</td>
</tr>
<tr>
<td>5</td>
<td>2.6</td>
<td>13</td>
</tr>
<tr>
<td>5</td>
<td>2.5</td>
<td>16</td>
</tr>
</tbody>
</table>

Table 5.4 Negative transconductance variation with gate voltages
5.3.2 Standard Cell Library of CNNs

We now describe the functional cells in the standard cell library of CNNs, obtained from the basic components of the cell library. This library is classified into three sets of uncoupled CNN applications as discussed in Chapter 4.

1) CNNs requiring no coupling from neighboring cell inputs.

2) CNNs requiring coupling from neighboring cell inputs.

3) CNNs requiring no input coupling.

The input and initial state images presented to the state capacitor are restricted to $\pm 1$ V as their pixel values; therefore the transconductance elements which are used to realize the VCCS’s feeding the above images are set to operate is in their linear mode. The obtained output voltage is always in the (-1, +1) range.

5.3.2.1 CNNs Requiring No Coupling from Neighboring Cell Inputs

As mentioned in chapter 4, LOGIC OR and LOGIC NOT CNNs belong to this class of CNNs. Here we present the details of LOGIC OR CNN with its MAGIC layouts and HSPICE plots and for the LOGIC NOT CNN, we present the MAGIC layouts of a single cell and the tabulated HSPICE results, for any further details on it extended 1×3 CNN and the HSPICE plots please refer Appendix B.

LOGIC OR CNN

LOGIC OR CNN is used to transform an input and an initial state image into an output image which is a set union/ logic OR of these two images. Here we have realized both a
single cell and a CNN of 1×3 array size. Their MAGIC layouts and HSPICE results follow.

- **A Single Cell LOGIC OR MAGIC Layout**

Based on its template values, 3 positive programmable transconductance amplifiers (posvartrans.mag cells whose gate values are set accordingly) are used to realize the 3 positive coupling coefficients which are shown below in the Figure 5.14. As mentioned in Chapter 4, the same positive amplifiers are used to realize the other components of the cell.

![A Single Cell LOGIC OR MAGIC Layout](image)

**Fig. 5.14 A Single cell LOGIC OR CNN Magic layout a) top level b) expanded design**
• Single Cell LOGIC OR HSPICE Simulation Plots

Figure 5.15 below shows all the signal voltages of a simple LOGIC OR cell, where a) shows initial state voltage \( (V_x) \) of -1V b) shows external input voltage \( (V_u) \) of 1V c) shows the obtained output voltage \( (V_y) \) which is a Logic OR of \( V_x \) and \( V_u \) of +0.233V, thus verifying its functionality d) a Start signal which is used to set the \( V_x \) to the state capacitor, e) a Select signal which is used to feed in the summed voltages from the coupling module into the cell core. It is to be noted that as was already mentioned in Chapter 4 (refer Figure 4.1), that the actual initial state voltage externally applied to the cell is \( \bar{V}_x \) (inverted \( V_x \)). Therefore one has to consider that the initial state voltage would be the inverse of \( V_x \).

![Fig. 5.15 Single cell LOGIC OR CNN HSPICE results verifying its functionality](image-url)
• **1×3 LOGIC OR CNN MAGIC Layout**

As an extension to the cell single architecture, a simple 1×3 LOGIC OR CNN cell array is realized. Its Magic layout is given below in Fig. 5.16.

![Fig. 5.16 A 1×3 LOGIC OR CNN Magic layout a) top level b) expanded design](image-url)
• 1×3 LOGIC OR CNN HSPICE Simulation Plots

Figure 5.17 below shows HSPICE results of a 1×3 LOGIC OR CNN array, indicating various initial state $V_x$, input voltage $V_u$ combinations applied to the 3 cells and their obtained respective output voltages $V_y$ for an input bias $V_z$ of 1 V. Thus, verifying its functionality.

![Diagram of HSPICE simulation plots showing various voltage levels and waveforms for different input and output combinations.](image)

Fig. 5.17 1×3 LOGIC OR CNN HSPICE results verifying its functionality
• LOGIC OR CNN HSPICE Results

Table 5.5 shows the summary of obtained HSPICE results for a LOGIC OR CNN, for all combinations of initial state Vx and input Vu voltages with an input bias Vz voltage of +1V magnitude. These results coincide for a single cell as well as for a 1×3 array.

<table>
<thead>
<tr>
<th>Initial State Vx (V)</th>
<th>Input Vu (V)</th>
<th>Expected Output Vy</th>
<th>HSPICE Output Vy (V)</th>
</tr>
</thead>
<tbody>
<tr>
<td>+1</td>
<td>+1</td>
<td>Positive</td>
<td>0.253</td>
</tr>
<tr>
<td>-1</td>
<td>+1</td>
<td>Positive</td>
<td>0.233</td>
</tr>
<tr>
<td>+1</td>
<td>-1</td>
<td>Positive</td>
<td>0.0469</td>
</tr>
<tr>
<td>-1</td>
<td>-1</td>
<td>Negative</td>
<td>-0.0142</td>
</tr>
</tbody>
</table>

Table 5.5 LOGIC OR CNN HSPICE Results

□ LOGIC NOT CNN

A LOGIC NOT CNN is used to transform an input image and a zero initial state image into an output image which is an inversion or Logic Not of the input image. As an extension to the cell single architecture a simple 1×3 LOGIC NOT CNN cell array is realized. MAGIC layout of a single cell and a table of the HSPICE results are shown in what follows.

• A Single Cell LOGIC NOT MAGIC Layout

Based on its template values, both a negative and a positive programmable transconductance amplifiers (negvartrans.mag cells and posvartrans.mag cells whose gate values are set accordingly) are used to realize the 2 coupling coefficients which are
shown below in the Figure 5.18. As mentioned in Chapter 4, the same positive amplifiers are used to realize the other components of the cell. Please refer Appendix B for Magic layout of 1×3 LOGIC NOT CNN cell array.

Fig. 5.18 A Single cell LOGIC NOT CNN Magic layout a) top level b) expanded design

- **LOGIC NOT CNN HSPICE Results**

Table 5.6 shows the summary of obtained HSPICE results for a LOGIC NOT CNN, for both positive and negative input voltages $V_u$ with an input bias $V_z$ voltage of 0V magnitude. These results coincide for a single cell as well as for a 1×3 CNN array.
<table>
<thead>
<tr>
<th>Input $V_u$ (V)</th>
<th>Expected Output $V_y$</th>
<th>HSPICE Output $V_y$ (V)</th>
</tr>
</thead>
<tbody>
<tr>
<td>+1</td>
<td>Negative</td>
<td>-0.0763</td>
</tr>
<tr>
<td>-1</td>
<td>Positive</td>
<td>0.0886</td>
</tr>
</tbody>
</table>

Table 5.6 LOGIC NOT CNN HSPICE Results

### 5.3.2.2 CNNs Requiring Coupling from Neighboring Cell Inputs

SHIFT (WEST, EAST) and DILATION CNNs belong to this class of CNNs as was already discussed in Chapter 4. Here we present in detail the SHIFT WEST CNN with its MAGIC layouts and HSPICE plots and for the other CNNs we present tabulated HSPICE results. For further details please refer Appendix B.

#### SHIFT WEST CNN

This CNN transforms the input image $V_u$ and a zero initial state image $V_x$ into an output image $V_y$, which is shifted version of input image by one unit west. Here we have realized a CNN of 1×3 array size. Its MAGIC layouts and HSPICE results follow.

- **1×3 SHIFT WEST MAGIC Layout**

A simple 1×3 SHIFT WEST CNN cell array, requiring the coupling from the east neighboring cells inputs is realized. Its Magic layout is shown in Figure 5.19. Based on its template values, positive programmable transconductance amplifiers (posvartrans.mag cells whose gate values are set accordingly) are used to realize the positive coupling coefficients. As mentioned in Chapter 4, the same positive amplifiers are used to realize even the other components of the cell.
Fig. 5.19 A 1×3 SHIFT WEST CNN Magic layout a) top level b) expanded design

- **1×3 SHIFT WEST CNN HSPICE Simulation Plots**

Figure 5.20 below shows the HSPICE results of a 1×3 SHIFT WEST CNN array, indicating various input voltages \(V_a\) applied to the 3 cells, input voltage to the third cell from its east neighbor \(V_{ue3}\) and their respective obtained output voltages \(V_y\) which are a shift west version of input voltages and an input state voltage \(V_x\) of 0V. Here initial state voltage \(V_x\) of 0V is set.
Table 5.7 shows the summary of obtained HSPICE results for a 1×3 SHIFT WEST CNN, for various combinations of input voltage vectors \((V_{uc11}, V_{uc12}, V_{uc13}, V_{uc13})\) with both input bias voltage \((V_z)\) and initial state voltage \((V_x)\) of 0V magnitude for all cells.
This CNN transforms the input image $V_u$ and a zero initial state image $V_x$ into an output image $V_y$, which is shifted version of input image by one unit east. Here we have realized a CNN of $1\times3$ array size. For MAGIC layouts and HSPICE results please refer Appendix B.

- **1×3 SHIFT EAST MAGIC Layout**

A simple $1\times3$ SHIFT EAST CNN cell array, requiring the coupling from the west neighboring cells inputs is realized. Its layout is similar to that of a SHIFT WEST CNN, but for its input voltages coupling connections.

- **SHIFT EAST CNN HSPICE Results**

Table 5.8 shows the summary of obtained HSPICE results for a $1\times3$ SHIFT EAST CNN, for various combinations of input voltage vectors $(V_{uw11}, V_{uc11}, V_{uc12}, V_{uc13})$ with both input bias voltage $(V_z)$ and initial state voltage $(V_x)$ of 0V magnitude for all cells.
<table>
<thead>
<tr>
<th>Input vectors Vu (V)</th>
<th>Expected Output (Y)</th>
<th>HSPICE Output (Y)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(V_{uw1}, V_{uc11}, V_{uc12}, V_{uc13})</td>
<td>((V_{y11out}, V_{y12out}, V_{y13out}))</td>
<td>((V_{y11out}, V_{y12out}, V_{y13out}))</td>
</tr>
<tr>
<td>-1, -1, +1, -1</td>
<td>Negative, Negative, Positive</td>
<td>-0.0994, -0.0994, +0.0945</td>
</tr>
<tr>
<td>+1, -1, -1, +1</td>
<td>Positive, Negative, Negative</td>
<td>+0.0945, -0.0994, -0.0994</td>
</tr>
<tr>
<td>-1, +1, -1, +1</td>
<td>Negative, Positive, Negative</td>
<td>-0.0994, +0.0945, -0.0994</td>
</tr>
</tbody>
</table>

Table 5.8 1×3 SHIFT EAST CNN HSPICE Results

It could be inferred that the SHIFT WEST architecture could easily be extended to other
to even NORTH and SOUTH, SHIFT CNNs by arranging the cells vertically as in a 3×1
array. SHIFT CNNs in other compass directions are obtained with an extension to this
architecture to minimum of 3×3.

DILATION WEST CNN

This CNN transforms the input image \(V_u\) and a zero initial state image \(V_x\) into an output
image \(V_y\), which is dilated (expanded) version of input image by one unit west. Here we
have realized a CNN of 1×3 array size. For MAGIC layouts and HSPICE results please
refer Appendix B.

- 1×3 DILATION WEST MAGIC Layout

A simple 1×3 DILATION WEST CNN cell array, requiring the coupling from the east
neighboring cells inputs is realized. Its Magic layout is shown in Figure 5.21. Based on
its template values, 4 positive programmable transconductance amplifiers (posvartrans.mag cells whose gate values are set accordingly) are used to realize the 4 positive coupling coefficients. As mentioned in Chapter 4, the same positive amplifiers are used to realize even the other components of the cell.

Fig. 5.21 A 1×3 DILATION WEST CNN Magic layout a) top level b) expanded design
**DILATION WEST CNN HSPICE Results**

Table 5.7 shows the summary of obtained HSPICE results of a 1×3 DILATION WEST CNN, for various combinations of input voltage vectors ($V_{uc11}$, $V_{uc12}$, $V_{uc13}$, $V_{uc13}$) for an input bias voltage ($V_z$) of +1V and initial state voltage ($V_x$) of 0V magnitude for all cells.

<table>
<thead>
<tr>
<th>Input (U) ($V_{uc11}$, $V_{uc12}$, $V_{uc13}$, $V_{uc13}$)</th>
<th>Expected Output (Y) ($V_{y11out}$, $V_{y12out}$, $V_{y13out}$)</th>
<th>Actual Output (Y) ($V_{y11out}$, $V_{y12out}$, $V_{y13out}$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1, -1, +1, -1</td>
<td>Negative, Positive, Positive</td>
<td>-0.066, +0.0986, +0.0768</td>
</tr>
<tr>
<td>+1, -1, -1, +1</td>
<td>Positive, Negative, Positive</td>
<td>+0.0767, -0.083, +0.0767</td>
</tr>
<tr>
<td>-1, +1, -1, +1</td>
<td>Positive, Positive, Positive</td>
<td>+0.0768, +0.0968, +0.0768</td>
</tr>
<tr>
<td>-1, +1, -1, -1</td>
<td>Positive, Positive, Negative</td>
<td>+0.0768, +0.0968, -0.066</td>
</tr>
<tr>
<td>-1, -1, -1, +1</td>
<td>Negative, Positive, Positive</td>
<td>-0.066, -0.083, +0.0768</td>
</tr>
</tbody>
</table>

Table 5.9 1×3 DILATION WEST CNN HSPICE Results

**5.3.2.3 CNNs Requiring No Input Coupling**

FILBLACK CNN belongs to this class of CNNs. Here we present its details including the MAGIC layouts and HSPICE results plots.

**FILBLACK CNN**

A FILBLACK CNN is used to transform an initial state image and a zero input image into an output image which is a completely black version of initial state image. Here we
have realized both a single cell and a 1×3 array CNN. Their MAGIC layouts and HSPICE results follow.

- **A Single Cell FILBLACK MAGIC Layout**

Based on its template values, 2 positive programmable transconductance amplifiers (posvartrans.mag cells whose gate values are set accordingly) are used to realize the 2 positive coupling coefficients as shown below in the Figure 5.22. As mentioned in Chapter 4, the same positive amplifiers are used to realize the other components of the cell.

![A Single cell FILBLACK CNN Magic layout](image)

**Fig. 5.22 A Single cell FILBLACK CNN Magic layout a) top level b) expanded design**
Single Cell FILBLACK HSPICE Simulation Plots

Figure 5.15 below shows all the signal voltages of a simple FILBLACK cell, where a) shows initial state voltage ($V_x$) of -1V b) shows the obtained output voltage ($V_y$) of +0.16V which is a FILBLACK of $V_x$, thus verifying its functionality c) external input bias voltage ($V_z$) of +1V d) a Start signal which is used to set the $V_x$ to the state capacitor and e) a Select signal which is used to feed in the summed voltages from the coupling module into the cell core. As mentioned before in LOGIC OR section, the actual initial state voltage externally applied to the cell is $\bar{V}_x$ (inverted $V_x$). Therefore one has to consider that the initial state voltage would be the inverse of $\bar{V}_x$.

Fig. 5.23 Single cell FILBLACK CNN HSPICE results verifying its functionality
• **A 1×3 FILBLACK CNN MAGIC Layout**

As an extension to the cell single architecture, a simple 1×3 FILBLACK CNN cell array is realized. Its Magic layout is given below in Figure 5.24.

![Diagram](image)

**Fig. 5.24 A 1×3 FILBLACK CNN Magic layout a) top level b) expanded design**
• 1×3 FILBLACK CNN HSPICE Simulation Plots

Figure 5.25 below shows HSPICE results of a 1×3 FILBLACK CNN array, indicating various initial state (V_x) voltages applied to the 3 cells and their respective obtained output voltages (V_y) for an input bias (V_z) of +1 V. Thus, verifying its functionality.

![Fig. 5.25 1×3 FILBLACK CNN HSPICE results verifying its functionality](image-url)
Table 5.10 shows the summary of obtained HSPICE results for a FILBLACK CNN, two possible initial state Vx voltages (+1 and -1), with an input bias voltage $V_z$ of +1V magnitude. These results coincide for a single cell as well as for a $1 \times 3$ array.

<table>
<thead>
<tr>
<th>Initial State (Vx)</th>
<th>Expected Output (Y)</th>
<th>HSPICE Output (Y)</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1</td>
<td>Positive</td>
<td>+0.186</td>
</tr>
<tr>
<td>1</td>
<td>Positive</td>
<td>+0.161</td>
</tr>
</tbody>
</table>

Table 5.10 FILBLACK CNN HSPICE Results
6. Conclusions and Future Work

6.1 Summary and Conclusions

This thesis presents the analog CMOS implementation of cellular neural networks (CNNs), which form a class of information processing systems highly suitable for VLSI implementations. The reasons for choosing the analog implementations and for realizing a standard cell library are presented.

In this work, a standard cell library of CNNs is realized using analog CMOS inverter based transconductance amplifier approach. In this approach, the analog CMOS transconductance amplifier forms a basic building block of the CMOS CNN cell circuit and all the other cell components are derived from this amplifier, thus making it a simple and a modular approach. It is clearly demonstrated that this approach is a programmable one, as appropriate transconductance parameters could be chosen according to the predetermined coupling coefficients, by just varying the gate voltages of the amplifiers, thus enabling us to adapt the same cell architecture for realizing various image processing applications.

A standard cell library of some uncoupled CNNs, where all the feedback coupling coefficients other than the center one are zero, is laid out in the MAGIC layout editor (Version 7.1) with -T SCN3ME_SUBM.30 0.5μ technology. Simulations are carried out in HSPICE (Version X-2005.09-SP1) using a Level-49 MOS model description, verifying the functionality of these applications. A 1×3 array of CNNs for those
applications is successfully implemented by extending the single cell architectures. To understand their working principle, MATLAB (7.0.1) simulations were also carried out for this set of CNN applications.

6.2 Future Work

- **Library Extension**

  - **To a larger network**

    One can extend this library to larger CNN array chips which could be fabricated and tested.

  - **To other image processing CNN applications**

    The library belonging to the uncoupled class of CNNs consists of some simple image processing applications which require few coupling coefficients. This could be extended to include other applications requiring more coupling coefficients such as edge detection, line detection and corner detection, etc.

  - **To other classes of CNNs**

    This library of cells belonging to the class of uncoupled CNNs could be extended to include other coupled classes of CNNs. One can extend this library to even include multiple layers and multiple dimensions. These could be used for applications such as object/ feature extraction, color image processing, wavelet transformation, noise removal, etc.

  - **To include other application paradigms**

    One could extend this library to other application paradigms, such as modeling biological vision or sensory organs, solving of partial differential equations, etc.
HDL Modeling

Hardware description language models in VHDL-AMS or Verilog A/AMS can be developed and validated for this cell library.
Bibliography


Appendix A

MATLAB Codes

This appendix includes MATLAB codes for the standard cell library of CNNs discussed in Chapters 4 and 5. These are given below in the following sections.

A.1 LOGIC OR CNN
A.2 LOGIC NOT CNN
A.3 SHIFT WEST CNN
A.4 DILATION OR CNN
A.5 FILBLACK CNN

A.1 LOGIC OR CNN

```matlab
% LOGOR page 62 of [1]
%clc;

M=15;N=15;

A = [0 0 0;
     0 3 0;
     0 0 0];

B = [0 0 0;
     0 3 0;
     0 0 0];

z=2;

U=[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0;
   0,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,0;
   0,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,0;
   0,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,0;
   0,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,0;
   0,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,0;
   0,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,0;
   0,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,0;
   0,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,0;
   0,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,0;
   0,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,0;
   0,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,0;
   0,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,0;
   0,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,0;
   0,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,0];

X=[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0];
```
0,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,0;
0,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,0;
0,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,0;
0,-1,1,1,1,1,1,1,1,1,1,1,1,1,1,-1,0;
0,-1,1,1,1,1,1,1,1,1,1,1,1,1,1,-1,0;
0,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,0;
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0;
p=0;
for i=1:M+2
    for j=1:N+2
        Y(i,j)=1/2*abs(X(i,j)+1)-1/2*abs(X(i,j)-1);
    end
end
for i=2:M+1
    for j=2:N+1
        for k=i-1:i+1
            for l=j-1:j+1
                d=A(k-i+2,l-j+2)*Y(k,l);
                c=B(k-i+2,l-j+2)*U(k,l);
                S(i,j)=p+d+c;
                p=S(i,j);
            end
        end
        m=p+z;
        t=1.5;
        %temp1=dsolve('Dx+x=m','x(0)=-X(i,j)'); => temp1=
m+exp(-t)*(-m-X(i,j))% Initial State value -
        %temp1=m+(exp(-t)*(-m-X(i,j))); % Initial State value -
        temp1=m+(exp(-t)*(-m));
        Y(i,j)=1/2*abs(temp1+1)-1/2*abs(temp1-1);
        p=0;
    end
end
for i=1:17
    for j=1:17
        I(i,j)=i;
        J(i,j)=j;
        surf(I,J,Y,'EdgeColor','black');
    end
end
A.2 LOGIC NOT CNN

% LOGIC NOT page 60 of book
cic;
M=15;N=15;

A = [0 0 0;
     0 1 0;
     0 0 0];

B = [0 0 0;
     0 -2 0;
     0 0 0];

z = 0;

U = [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0;
     0,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,0;
     0,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,0;
     0,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,0;
     0,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,0;
     0,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,0;
     0,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,0;
     0,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,0;
     0,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,0;
     0,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,0;
     0,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,0;
     0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0];

X = [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0;
     0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0;
     0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0;
     0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0;
     0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0;
     0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0;
     0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0;
     0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0;
     0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0;
     0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0;
     0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0;
     0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0;
     0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0;
     0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0];

p = 0;

for i=1:M+2
for j=1:N+2

121
\[ Y(i,j) = \frac{1}{2} \text{abs}(X(i,j)+1) - \frac{1}{2} \text{abs}(X(i,j)-1); \]

end
end

for i=2:M+1
    for j=2:N+1
        for k=i-1:i+1
            for l=j-1:j+1
                d = A(k-i+2,l-j+2) \times Y(k,l);
                c = B(k-i+2,l-j+2) \times U(k,l);
                S(i,j) = p + d + c;
                p = S(i,j);
            end
        end
        m = p + z;
        %temp1 = dsolve('Dx+x=m', 'x(0)=-X(i,j)'); => temp1 =
        m + \exp(-t) \times (-m - X(i,j))
        t = 1.5;
        %temp1 = m + (\exp(-t) \times (-m - X(i,j))); % Initial State value
        Y(i,j) = \frac{1}{2} \text{abs}(temp1+1) - \frac{1}{2} \text{abs}(temp1-1);
        p = 0;
    end
end

for i=1:17
    for j=1:17
        I(i,j) = i;
        J(i,j) = j;
        surf(I,J,Y,'EdgeColor','black');
    end
end

A.3 SHIFT WEST CNN

% Shift West Translation (by 1 pixel-unit) template

M=15; N=15;
A = [0 0 0;
    0 1 0;
    0 0 0];
B = [0 0 0;
    0 0 1;
    0 0 0];
z = 0;
U = [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0; 0,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,0; 0,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,0];
\[
\begin{align*}
X &= \begin{bmatrix}
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0; \\
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0; \\
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0; \\
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0; \\
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0; \\
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0; \\
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0; \\
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0; \\
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0; \\
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0; \\
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0; \\
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0; \\
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0; \\
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0; \\
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0;
\end{bmatrix};
\end{align*}
\]

\[
p=0;
\]

\[
\text{for } i=1:M+2 \\
\text{for } j=1:N+2 \\
\quad Y(i,j)=1/2*ab\cdot(X(i,j)+1)-1/2*abs(X(i,j)-1); \\
\text{end} \\
\text{end} \\
\text{for } i=2:M+1 \\
\text{for } j=2:N+1 \\
\quad \text{for } k=i-1:i+1 \\
\quad \quad \text{for } l=j-1:j+1 \\
\quad \quad \quad d=A(k-i+2,l-j+2)\cdot Y(k,l); \\
\quad \quad \quad c=B(k-i+2,l-j+2)\cdot U(k,l); \\
\quad \quad \quad S(i,j)=p+d+c; \\
\quad \quad \quad p=S(i,j); \\
\quad \text{end} \\
\quad \text{end} \\
\text{end}
\]

\[
m=p+z; \\
t=3; \\
\%temp1=dsolve('Dx+x=m','x(0)=-X(i,j)'); \Rightarrow \text{temp1=}
m+\exp(-t)*(-m-X(i,j)) \% \text{Initial State value - X(i,j)} \\
\text{temp1}=m+\exp(-t)*(-m); \\
\text{Y(i,j)=1/2*abs(temp1+1)-1/2*abs(temp1-1);}
for i=1:17
  for j=1:17
    I(i,j)=i;
    J(i,j)=j;
    surf(I,J,Y,'EdgeColor','black');
  end
end

A.4 DILATION CNN

% DILATION page 87 of [1]
clc;
M=15; N=15;
A = [0 0 0;
     0 2 0;
     0 0 0];
B = [0 0 0;
     0 1 1;
     0 0 0];
z=1.5;
U=[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0;
   0,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,0;
   0,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,0;
   0,-1,-1,1,1,1,1,1,1,1,1,1,1,1,1,1,0;
   0,-1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0;
   0,-1,1,1,1,1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,0;
   0,-1,1,1,1,1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,0;
   0,-1,1,1,1,1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,0;
   0,-1,1,1,1,1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,0;
   0,-1,1,1,1,1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,0;
   0,-1,1,1,1,1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,0;
   0,-1,1,1,1,1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,0;
   0,-1,1,1,1,1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,0;
   0,-1,1,1,1,1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,0;
   0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0];
X=[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0;
   0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0;
   0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0;
   0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0;
   0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0;
   0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0;
   0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0;
   0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0;
   0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0;
   0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0;
   0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0;
   0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0;
   0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0;
   0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0;
   0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0];
for i=1:M+2
    for j=1:N+2
        Y(i,j)=1/2*abs(X(i,j)+1)-1/2*abs(X(i,j)-1);
    end
end
p=0;
for i=2:M+1
    for j=2:N+1
        for k=i-1:i+1
            for l=j-1:j+1
                d=A(k-i+2,l-j+2)*Y(k,l);
                c=B(k-i+2,l-j+2)*U(k,l);
                S(i,j)=p+d+c;
                p=S(i,j);
            end
        end
        m=p+z;
        t=2;
        %temp1=dsolve('Dx+x=m','x(0)=-X(i,j)'); => temp1 = m+exp(-t)*(-m-X(i,j))
        %temp1=m+(exp(-t)*(-m)); x = m+exp(-t)*(-m-p);
        temp1=m+(exp(-t)*(-m));
        Y(i,j)=1/2*abs(temp1+1)-1/2*abs(temp1-1);
        p=0;
    end
end
for i=1:17
    for j=1:17
        I(i,j)=i;
        J(i,j)=j;
        surf(I,J,Y,'EdgeColor','black');
    end
A.5 FILBLACK CNN

% FILBLACK of page 62 of [1]
clc;
M=15;N=15;
A = [0 0 0;
     0 2 0;
     0 0 0];
B = [0 0 0;
     0 0 0;
     0 0 0];
z=4;
X=[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0;
   0,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,0;
   0,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,0;
   0,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,0;
   0,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,0;
   0,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,0;
   0,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,0;
   0,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,0;
   0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0];
U=[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0;
   0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0;
   0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0;
   0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0;
   0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0;
   0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0;
   0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0;
   0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0;
   0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0;
   0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0;
   0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0;
   0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0;
   0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0;
   0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0;
   0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0];
p=0;
for i=1:M+2
    for j=1:N+2

126
Y(i,j)=1/2*abs(X(i,j)+1)-1/2*abs(X(i,j)-1);
end
end

for i=2:M+1
    for j=2:N+1
        for k=i-1:i+1
            for l=j-1:j+1
                d=A(k-i+2,l-j+2)*Y(k,l);
                c=B(k-i+2,l-j+2)*U(k,l);
                S(i,j)=p+d+c;
                p=S(i,j);
            end
        end
        m=p+z;
        t=1;
        %temp1=dsolve('Dx+x=m','x(0)=-X(i,j)'); => temp1=
        m+exp(-t)*(-m-X(i,j)); % Initial State value - X(i,j)
        temp1=m+(exp(-t)*(-m-X(i,j))); % Initial State value - X(i,j)
        Y(i,j)=1/2*abs(temp1+1)-1/2*abs(temp1-1);
        p=0;
    end
end

for i=1:17
    for j=1:17
        I(i,j)=i;
        J(i,j)=j;
        surf(I,J,Y,'EdgeColor','black');
    end
end
Appendix B

MAGIC Layouts, HSPICE Files and Simulations

The appendix is organized into these sections

HSPICE Model File
B.1 SPICE 3f5 Level 8, Star-HSPICE Level 49, UTMOST Level 8.

HSPICE files for CNN cell components
B.2 Transconductance amplifier
B.3 Resistor using a CMOS transconductance amplifier
B.4 Programmable positive transconductance amplifier
B.5 Programmable negative transconductance amplifier

HSPICE models for the CNN library of applications
B.6 LOGIC OR Single cell CNN
B.7 LOGIC OR 1×3 CNN
B.8 LOGIC NOT Single cell
B.9 LOGIC NOT 1×3 CNN
B.10 SHIFT WEST 1×3 CNN
B.11 SHIFT EAST 1×3 CNN
B.12 DILATION WEST 1×3 CNN
B.13 FILBLACK single cell CNN
B.14 FILBLACK 1×3 CNN

HSPICE simulation results
B.15 LOGIC NOT single cell
B.16 LOGIC NOT 1×3 CNN
B.17 SHIFT EAST 1×3 CNN
B.18 DILATION WEST 1×3 CNN

Magic Layouts for CNN library of applications
B.19 LOGIC NOT 1×3 CNN
B.20 SHIFT EAST 1×3 CNN
B.21 DILATION WEST 1×3 CNN
**HSPICE Transistor model File**

B.1 SPICE 3f5 Level 8, Star-HSPICE Level 49, UTMOST Level 8. [16]

This is included as model.txt file in all the HSPICE models given below.

```plaintext
* DATE: Aug 13/03
* LOT: T36S                WAF: 1101
* Temperature_parameters=Default

.MODEL nfet NMOS (LEVEL   = 49
+VERSION = 3.1           TNOM    = 27             TOX     = 1.41E-8
+XJ     = 1.5E-7           NCH     = 1.7E17         VTH0    = 0.6461285
+K1     = 0.9152455         K2      = -0.105898       K3      = 26.1007653
+K3B    = -8.3258109        W0      = 1E-8           NLX     = 1E-9
+DVTOW  = 0                DVT1W   = 0                DVT2W   = 0
+DVT0   = 3.7861788        DVT1    = 0.3717718       DVT2    = -0.0830195
+U0     = 442.2002499      UA      = 1E-13          UB      = 1.185519E-18
+UC     = 6.470959E-12      VSAT    = 1.727533E5     A0      = 0.6072333
+AGS    = 0.1377586        VOFF    = 0                NFACTOR = 0.472999
+CDSCB  = 0                CDSC    = 2.4E-4         CDSCD   = 0
+DSUB   = 0.0689754        PCLM    = 2.4881989      PDIBLC1 = 0.9941032
+PDIBLC2 = 2.298113E-3      PDIBLCB = -0.0180618     DROUT   = 0.0838803
+PSCBE1 = 6.445443E8        PSCBE2 = 2.598335E-4     PVAG    = 0
+DELTA  = 0.01             RSH     = 81.8             MOBMOD  = 1
+PRT    = 0                UTE     = -1.5           KT1     = -0.11
+KTIL   = 0                KT2     = 0.022          UA1     = 4.31E-9
+UB1    = -7.61E-18        UC1     = -5.6E-11       AT      = 3.3E4
+WL     = 0                WLN     = 1              WW      = 0
+WNN    = 1                WWL     = 0              WWL     = 0
+LLN    = 1                LW      = 0              LLN     = 1
+LWL    = 0                CAPMOD  = 2              XPART   = 0.5
+CGDO   = 2.01E-10         CGSO    = 2.01E-10       CGBO    = 1E-9
+CL     = 4.227962E-4      PB      = 0.9113851       MJ      = 0.4296861
+CJ     = 2.925306E-10      PBSW    = 0.8            MJSW    = 0.170165
+CJSW   = 1.64E-10         PBSWG   = 0.8            MJSWG   = 0.170165
+CF     = 0                PVTH0   = 0.1375778      PRDSW   = -174.8924404
+PK2    = -0.0194887       WKETAG = -0.0237035     LKETAG  = 0.205439
)

.MODEL pfet PMOS (LEVEL   = 49
+VERSION = 3.1           TNOM    = 27             TOX     = 1.41E-8
+XJ     = 1.5E-7           NCH     = 1.7E17         VTH0    = -0.9213369
```
HSPICE files for CNN cell components

B.2 Transconductance amplifier

* HSPICE file created from transconductance.ext - technology: scmos
.option scale=0.3u
.include model.txt
m1000 Va VG1 VDD Gnd nfet w=33 l=34
+ ad=107 pd=96 as=169 ps=94
m1001 Vb VG4 VSS Vdd pfet w=100 l=34
+ ad=404 pd=210 as=500 ps=210
m1002 Vout Vin Va Vdd pfet w=100 l=34
+ ad=398 pd=210 as=404 ps=210
m1003 Vout Vin Vb Gnd nfet w=33 l=34
+ ad=107 pd=96 as=107 ps=90
C0 Vout GND 2.8fF
B.3 Resistor using a CMOS transconductance amplifier

* HSPICE file created from Resistortrans.ext - technology: scmos
.option scale=0.3u
.include model.txt
m1000 x0/Va VGR1 VDD Gnd nfet w=33 l=34
+ ad=107 pd=96 as=169 ps=94
m1001 x0/Vb VGR4 VSS Vdd pfet w=100 l=34
+ ad=404 pd=210 as=500 ps=210
m1002 VR VR x0/Va Vdd pfet w=100 l=34
+ ad=398 pd=210 as=404 ps=210
m1003 VR VR x0/Vb Gnd nfet w=33 l=34
+ ad=107 pd=96 as=107 ps=90
C0 VR GND 9.1fF
C1 VSS GND 5.7fF
C2 VDD GND 5.4fF

** hspice subcircuit dictionary
* x0 transconductance_0

VDD VDD 0 5
VSS VSS 0 -5
Vgr1 VGR1 0 4
Vgr4 VGR4 0 -4

IR VR 0 DC 0
*.DC IR -0.1u 0.1u 0.05u
*.DC IR -3u 3u 0.5u
*.DC IR -30u 30u 5u
.options post
.end
B.4 Programmable positive transconductance amplifier

* HSPICE file created from posvartrans.ext - technology: scmos

    .option scale=0.3u
    .include model.txt
m1000 x0/Va VGR1 VDD Gnd nfet w=33 l=34
    + ad=107 pd=96 as=338 ps=188
m1001 x0/Vb VGR4 VSS Vdd pfet w=100 l=34
    + ad=404 pd=210 as=1000 ps=420
m1002 VOUT VOUT x0/Va Vdd pfet w=100 l=34
    + ad=796 pd=420 as=404 ps=210
m1003 VOUT VOUT x0/Vb Gnd nfet w=33 l=34
    + ad=214 pd=192 as=107 ps=90
m1004 x1/Va VG1 VDD Gnd nfet w=33 l=34
    + ad=107 pd=96 as=0 ps=0
m1005 x1/Vb VG4 VSS Vdd pfet w=100 l=34
    + ad=404 pd=210 as=0 ps=0
m1006 VOUT VIN x1/Va Vdd pfet w=100 l=34
    + ad=0 pd=0 as=404 ps=210
m1007 VOUT VIN x1/Vb Gnd nfet w=33 l=34
    + ad=0 pd=0 as=107 ps=90
C0 VIN GND 6.4fF
C1 VOUT GND 6.0fF
C2 VSS GND 9.9fF
C3 VDD GND 10.4fF

** hspice subcircuit dictionary
* x0  transconductance_1
* x1  transconductance_0

VDD VDD 0 5
VSS VSS 0 -5
Vg1 VG1 0 5
Vg4 VG4 0 -5
Vgr1 VGR1 0 4
Vgr4 VGR4 0 -4
Vinput VIN 0 DC 0
.DC Vinput 2 -2 -0.01
.options post
.end

B.5 Programmable negative transconductance amplifier

* HSPICE file created from negvartrans.ext - technology: scmos

    .option scale=0.3u
    .include model.txt
m1000 x0/Va VGR1 VDD Gnd nfet w=33 l=34
    + ad=107 pd=96 as=845 ps=470
m1001 x0/Vb VGR4 VSS Vdd pfet w=100 l=34
** hspice subcircuit dictionary
* x0 posvartrans_1/transconductance_1
* x1 posvartrans_1/transconductance_0
* x2 posvartrans_1
* x3 transconductance_3
* x4 posvartrans_0/transconductance_1
* x5 posvartrans_0/transconductance_0

**************************************
HSPICE models for the CNN library of applications

As HSPICE file is extracted from the corresponding Magic layout, the part of the model which is to be set for these applications is only shown for the below set of applications except for a LOGIC OR single cell CNN.

**B.6 LOGIC OR Single cell CNN**

* HSPICE file created from LOGICOR_1CELL.ext - technology: scmos

```plaintext
* option scale=0.3u
.include model.txt
m1000 x0/Va VY11GR1 VDD Gnd nfet w=33 l=34 + ad=107 pd=96 as=2028 ps=1128
m1001 x0/Vb VY11GR4 VSS Vdd pfet w=100 l=34 + ad=404 pd=210 as=6000 ps=2520
m1002 VY11OUT VY11OUT x0/Va Vdd pfet w=100 l=34 + ad=1592 pd=840 as=404 ps=210
m1003 VY11OUT VY11OUT x0/Vb Gnd nfet w=33 l=34 + ad=428 pd=384 as=107 ps=90
m1004 x1/Va VY11G1 VDD Gnd nfet w=33 l=34 + ad=107 pd=96 as=0 ps=0
m1005 x1/Vb VY11G4 VSS Vdd pfet w=100 l=34 + ad=404 pd=210 as=0 ps=0
m1006 VY11OUT VY11OUT x1/Va Vdd pfet w=100 l=34 + ad=0 pd=0 as=404 ps=210
m1007 VY11OUT VY11OUT x1/Vb Gnd nfet w=33 l=34 + ad=0 pd=0 as=107 ps=90
m1008 x2/Va VYX11GR1 VDD Gnd nfet w=33 l=34 + ad=107 pd=96 as=0 ps=0
m1009 x2/Vb VYX11GR4 VSS Vdd pfet w=100 l=34 + ad=404 pd=210 as=0 ps=0
m1010 VY11OUT VY11OUT x2/Va Vdd pfet w=100 l=34 + ad=0 pd=0 as=404 ps=210
m1011 VY11OUT VY11OUT x2/Vb Gnd nfet w=33 l=34 + ad=0 pd=0 as=107 ps=90
m1012 x3/Va VYX11G1 VDD Gnd nfet w=33 l=34 + ad=107 pd=96 as=0 ps=0
```

134
** hspice subcircuit dictionary
  * x0 posvartrans_17/transconductance_1
  * x1 posvartrans_17/transconductance_0
  * x2 posvartrans_16/transconductance_1
  * x3 posvartrans_16/transconductance_0
  * x4 posvartrans_15/transconductance_1
  * x5 posvartrans_15/transconductance_0
  * x6 posvartrans_14/transconductance_1
  * x7 posvartrans_14/transconductance_0
  * x8 posvartrans_13/transconductance_1
  * x9 posvartrans_13/transconductance_0
  * x10 posvartrans_12/transconductance_1
  * x11 posvartrans_12/transconductance_0

***************
Vdd VDD 0 5
Vss VSS 0 -5

Vstart start 0 PWL(0 0 100e-9 0 10le-9 5 400e-9 5 401e-9 0 4000e-9 0)
Vselect select 0 PWL(0 0 400e-9 0 401e-9 5 400e-6 5)

Vz11 VZ11 0 1
Cx11 VX11INITIAL 0 4pF

******* INPUTS *---- Initializing the State "VXINITIAL" and Input Voltage "U"

Vx11initial in_i11 0 DC -1
Vuc11 VUC11 0 DC -1

*************** Setting the transconductance values
B.7 LOGIC OR 1×3 CNN

* HSPICE file created from LOGICOR_3CELLS.ext - technology: scmos

.option scale=0.3u
.include model.txt
m1000 x0/Va VY13GR1 VDD Gnd nfet w=33 l=34
+ ad=107 pd=96 as=6084 ps=3384
m1001 x0/Vb VY13GR4 VSS Vdd pfet w=100 l=34
+ ad=404 pd=210 as=18000 ps=7560
m1002 VY13OUT VY13OUT x0/Va Vdd pfet w=100 l=34
+ ad=1592 pd=840 as=404 ps=210
m1003 VY13OUT VY13OUT x0/Vb Gnd nfet w=33 l=34
+ ad=428 pd=384 as=107 ps=90
m1004 x1/Va VY13G1 VDD Gnd nfet w=33 l=34
+ ad=107 pd=96 as=0 ps=0
m1005 x1/Vb VY13G4 VSS Vdd pfet w=100 l=34
m1063 VSUM12 VSUM12 x15/Vb Gnd nfet w=33 l=34
m1064 x16/Va VZ12GR1 VDD Gnd nfet w=33 l=34
m1065 x16/Vb VZ12GR4 VSS Vdd pfet w=100 l=34
m1066 VSUM12 VSUM12 x16/Va Vdd pfet w=100 l=34
m1067 VSUM12 VSUM12 x16/Vb Gnd nfet w=33 l=34
m1068 x17/Va VZ12G1 VDD Gnd nfet w=33 l=34
m1069 x17/Vb VZ12G4 VSS Vdd pfet w=100 l=34
m1070 VSUM12 VZ12 x17/Va Vdd pfet w=100 l=34
m1071 VSUM12 VZ12 x17/Vb Gnd nfet w=33 l=34
m1072 x18/Va VYC12GR1 VDD Gnd nfet w=33 l=34
m1073 x18/Vb VYC12GR4 VSS Vdd pfet w=100 l=34
m1074 VSUM12 VSUM12 x18/Va Vdd pfet w=100 l=34
m1075 VSUM12 VSUM12 x18/Vb Gnd nfet w=33 l=34
m1076 x19/Va VYC12G1 VDD Gnd nfet w=33 l=34
m1077 x19/Vb VYC12G4 VSS Vdd pfet w=100 l=34
m1078 VSUM12 VY12OUT x19/Va Vdd pfet w=100 l=34
m1079 VSUM12 VY12OUT x19/Vb Gnd nfet w=33 l=34
m1080 x20/Va VUC13GR1 VDD Gnd nfet w=33 l=34
m1081 x20/Vb VUC13GR4 VSS Vdd pfet w=100 l=34
m1082 VSUM13 VSUM13 x20/Va Vdd pfet w=100 l=34
m1083 VSUM13 VSUM13 x20/Vb Gnd nfet w=33 l=34
m1084 x21/Va VUC13G1 VDD Gnd nfet w=33 l=34
m1085 x21/Vb VUC13G4 VSS Vdd pfet w=100 l=34
m1086 VSUM13 VUC13 x21/Va Vdd pfet w=100 l=34
m1087 VSUM13 VUC13 x21/Vb Gnd nfet w=33 l=34
m1088 x22/Va VY11GR1 VDD Gnd nfet w=33 l=34
m1089 x22/Vb VY11GR4 VSS Vdd pfet w=100 l=34
m1090 VY11OUT VY11OUT x22/Va Vdd pfet w=100 l=34
+ ad=0 pd=0 as=404 ps=210
+ ad=107 pd=96 as=0 ps=0
+ ad=404 pd=210 as=0 ps=0
+ ad=0 pd=0 as=404 ps=210
+ ad=0 pd=0 as=107 ps=90
+ ad=107 pd=96 as=0 ps=0
+ ad=404 pd=210 as=0 ps=0
+ ad=0 pd=0 as=404 ps=210
+ ad=0 pd=0 as=107 ps=90
+ ad=107 pd=96 as=0 ps=0
+ ad=404 pd=210 as=0 ps=0
+ ad=0 pd=0 as=404 ps=210
+ ad=0 pd=0 as=107 ps=90
+ ad=107 pd=96 as=0 ps=0
+ ad=404 pd=210 as=0 ps=0
+ ad=0 pd=0 as=404 ps=210
+ ad=107 pd=96 as=0 ps=0
+ ad=404 pd=210 as=0 ps=0
+ ad=0 pd=0 as=404 ps=210
+ ad=107 pd=96 as=0 ps=0
+ ad=404 pd=210 as=0 ps=0
+ ad=0 pd=0 as=404 ps=210
+ ad=1592 pd=836 as=404 ps=210
<table>
<thead>
<tr>
<th>Device</th>
<th>Function</th>
<th>Pins</th>
<th>Width</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>m1091</td>
<td>VY11OUT</td>
<td>x22/Vb</td>
<td>33</td>
<td>34</td>
</tr>
<tr>
<td>m1092</td>
<td>x23/Va</td>
<td>VY11G1</td>
<td>33</td>
<td>34</td>
</tr>
<tr>
<td>m1093</td>
<td>x23/Vb</td>
<td>VY11G4</td>
<td>100</td>
<td>34</td>
</tr>
<tr>
<td>m1094</td>
<td>VY11OUT</td>
<td>x23/Va</td>
<td>100</td>
<td>34</td>
</tr>
<tr>
<td>m1095</td>
<td>VY11OUT</td>
<td>x23/Vb</td>
<td>33</td>
<td>34</td>
</tr>
<tr>
<td>m1096</td>
<td>x24/Va</td>
<td>VYX11G1</td>
<td>33</td>
<td>34</td>
</tr>
<tr>
<td>m1097</td>
<td>x24/Vb</td>
<td>VYX11G4</td>
<td>100</td>
<td>34</td>
</tr>
<tr>
<td>m1098</td>
<td>VY11OUT</td>
<td>x24/Va</td>
<td>100</td>
<td>34</td>
</tr>
<tr>
<td>m1099</td>
<td>VY11OUT</td>
<td>x24/Vb</td>
<td>33</td>
<td>34</td>
</tr>
<tr>
<td>m1100</td>
<td>x25/Va</td>
<td>VYX11G1</td>
<td>33</td>
<td>34</td>
</tr>
<tr>
<td>m1101</td>
<td>x25/Vb</td>
<td>VYX11G4</td>
<td>100</td>
<td>34</td>
</tr>
<tr>
<td>m1102</td>
<td>VY11OUT</td>
<td>x25/Va</td>
<td>100</td>
<td>34</td>
</tr>
<tr>
<td>m1103</td>
<td>VY11OUT</td>
<td>x25/Vb</td>
<td>33</td>
<td>34</td>
</tr>
<tr>
<td>m1104</td>
<td>x26/Va</td>
<td>VYX11G1</td>
<td>33</td>
<td>34</td>
</tr>
<tr>
<td>m1105</td>
<td>x26/Vb</td>
<td>VYX11G4</td>
<td>100</td>
<td>34</td>
</tr>
<tr>
<td>m1106</td>
<td>VSUM11</td>
<td>VSUM11</td>
<td>100</td>
<td>34</td>
</tr>
<tr>
<td>m1107</td>
<td>VSUM11</td>
<td>VSUM11</td>
<td>33</td>
<td>34</td>
</tr>
<tr>
<td>m1108</td>
<td>x27/Va</td>
<td>VX11G1</td>
<td>33</td>
<td>34</td>
</tr>
<tr>
<td>m1109</td>
<td>x27/Vb</td>
<td>VX11G4</td>
<td>100</td>
<td>34</td>
</tr>
<tr>
<td>m1110</td>
<td>VSUM11</td>
<td>VSUM11</td>
<td>100</td>
<td>34</td>
</tr>
<tr>
<td>m1111</td>
<td>VSUM11</td>
<td>VSUM11</td>
<td>33</td>
<td>34</td>
</tr>
<tr>
<td>m1112</td>
<td>x28/Va</td>
<td>VX11G1</td>
<td>33</td>
<td>34</td>
</tr>
<tr>
<td>m1113</td>
<td>x28/Vb</td>
<td>VX11G4</td>
<td>100</td>
<td>34</td>
</tr>
<tr>
<td>m1114</td>
<td>VSUM11</td>
<td>VSUM11</td>
<td>100</td>
<td>34</td>
</tr>
<tr>
<td>m1115</td>
<td>VSUM11</td>
<td>VSUM11</td>
<td>33</td>
<td>34</td>
</tr>
<tr>
<td>m1116</td>
<td>x29/Va</td>
<td>VZ11G1</td>
<td>33</td>
<td>34</td>
</tr>
<tr>
<td>m1117</td>
<td>x29/Vb</td>
<td>VZ11G4</td>
<td>100</td>
<td>34</td>
</tr>
<tr>
<td>m1118</td>
<td>VSUM11</td>
<td>VZ11</td>
<td>100</td>
<td>34</td>
</tr>
<tr>
<td>m1119</td>
<td>VSUM11</td>
<td>VZ11</td>
<td>33</td>
<td>34</td>
</tr>
</tbody>
</table>
m1148 VX13INITIAL select VSUM13 Gnd nfet w=33 l=34  
+ ad=218 pd=180 as=0 ps=0
m1149 in i13 start VX13INITIAL Gnd nfet w=33 l=34  
+ ad=136 pd=100 as=0 ps=0
C0 start GND 4.7fF
C1 select GND 4.8fF
C2 VUC11 GND 6.4fF
C3 VUC12 GND 6.4fF
C4 VZ11 GND 6.4fF
C5 VX11INITIAL GND 6.7fF
C6 VUC13 GND 6.4fF
C7 VSUM13 GND 11.6fF
C8 VZ12 GND 6.4fF
C9 VSUM12 GND 11.7fF
C10 VX12INITIAL GND 7.2fF
C11 VY12OUT GND 19.9fF
C12 VZ13 GND 6.4fF
C13 VX13INITIAL GND 7.0fF
C14 VYX13GR4 GND 2.2fF
C15 VSS GND 137.2fF
C16 VDD GND 47.3fF

** hspice subcircuit dictionary
* x0 posvartrans_11/transconductance_1
* x1 posvartrans_11/transconductance_0
* x2 posvartrans_10/transconductance_1
* x3 posvartrans_10/transconductance_0
* x4 posvartrans_9/transconductance_1
* x5 posvartrans_9/transconductance_0
* x6 posvartrans_8/transconductance_1
* x7 posvartrans_8/transconductance_0
* x8 posvartrans_7/transconductance_1
* x9 posvartrans_7/transconductance_0
* x10 posvartrans_5/transconductance_1
* x11 posvartrans_5/transconductance_0
* x12 posvartrans_4/transconductance_1
* x13 posvartrans_4/transconductance_0
* x14 posvartrans_3/transconductance_1
* x15 posvartrans_3/transconductance_0
* x16 posvartrans_2/transconductance_1
* x17 posvartrans_2/transconductance_0
* x18 posvartrans_1/transconductance_1
* x19 posvartrans_1/transconductance_0
* x20 posvartrans_6/transconductance_1
* x21 posvartrans_6/transconductance_0
* x22 posvartrans_17/transconductance_1
* x23 posvartrans_17/transconductance_0
* x24 posvartrans_16/transconductance_1
* x25 posvartrans_16/transconductance_0
* x26 posvartrans_15/transconductance_1
* x27 posvartrans_15/transconductance_0
* x28 posvartrans_14/transconductance_1
* x29 posvartrans_14/transconductance_0
* x30 posvartrans_13/transconductance_1
* x31 posvartrans_13/transconductance_0
* x32 posvartrans_0/transconductance_1
* x33 posvartrans_0/transconductance_0
* x34 posvartrans_12/transconductance_1
* x35 posvartrans_12/transconductance_0

**********************
Vdd VDD 0 5
Vss VSS 0 -5

Vstart start 0 PWL(0 0 100e-9 0 101e-9 5 400e-9 5 401e-9 0 4000e-9 0)
Vselect select 0 PWL(0 0 400e-9 0 401e-9 5 400e-6 5)

Vz11 VZ11 0 1
Vz12 VZ12 0 1
Vz13 VZ13 0 1

Cx11 VX11INITIAL 0 4pF
Cx12 VX12INITIAL 0 4pF
Cx13 VX13INITIAL 0 4pF

******** INPUTS *---- Initializing the State "VXINITIAL" and Input Voltage "U"

Vx11initial in_i11 0 DC 1
Vuc11 VUC11 0 DC 1

Vx12initial in_i12 0 DC -1
Vuc12 VUC12 0 DC -1

Vx13initial in_i13 0 DC -1
Vuc13 VUC13 0 DC 1

*********************** Setting the transconductance values

Vuc11g1 VUC11G1 0 5
Vuc11g4 VUC11G4 0 -5
Vuc11gr1 VUC11GR1 0 3.2
Vuc11gr4 VUC11GR4 0 -3.2

Vyc11g1 VYC11G1 0 5
Vyc11g4 VYC11G4 0 -5
Vyc11gr1 VYC11GR1 0 3.2
Vyc11gr4 VYC11GR4 0 -3.2

Vz11g1 VZ11G1 0 5
Vz11g4 VZ11G4 0 -5
Vz11gr1 VZ11GR1 0 3.6
Vz11gr4 VZ11GR4 0 -3.6

***************
Vuc12g1 VUC12G1 0 5
Vuc12g4 VUC12G4 0 -5
Vuc12gr1 VUC12GR1 0 3.2
Vuc12gr4 VUC12GR4 0 -3.2

Vyc12g1 VYC12G1 0 5
Vyc12g4 VYC12G4 0 -5
Vyc12gr1  VYC12GR1  0  3.2
Vyc12gr4  VYC12GR4  0  -3.2

Vz12g1  V212G1  0  5
Vz12g4  V212G4  0  -5
Vz12gr1  V212GR1  0  3.6
Vz12gr4  V212GR4  0  -3.6

************************************

Vuc13g1  VUC13G1  0  5
Vuc13g4  VUC13G4  0  -5
Vuc13gr1  VUC13GR1  0  3.2
Vuc13gr4  VUC13GR4  0  -3.2

Vyc13g1  VYC13G1  0  5
Vyc13g4  VYC13G4  0  -5
Vyc13gr1  VYC13GR1  0  3.2
Vyc13gr4  VYC13GR4  0  -3.2

Vz13g1  V213G1  0  5
Vz13g4  V213G4  0  -5
Vz13gr1  V213GR1  0  3.6
Vz13gr4  V213GR4  0  -3.6

************************************

Vx11g1  VX11G1  0  4
Vx11g4  VX11G4  0  -4
Vx11gr1  VX11GR1  0  5
Vx11gr4  VX11GR4  0  -5

Vy11g1  VY11G1  0  5
Vy11g4  VY11G4  0  -5
Vy11gr1  VY11GR1  0  5
Vy11gr4  VY11GR4  0  -5

Vy12g1  VY12G1  0  5
Vy12g4  VY12G4  0  -5
Vy12gr1  VY12GR1  0  5
Vy12gr4  VY12GR4  0  -5

****

Vx12g1  VX12G1  0  4
Vx12g4  VX12G4  0  -4
Vx12gr1  VX12GR1  0  5
Vx12gr4  VX12GR4  0  -5

Vy12g1  VY12G1  0  5
Vy12g4  VY12G4  0  -5
Vy12gr1  VY12GR1  0  5
Vy12gr4  VY12GR4  0  -5

Vy12g1  VY12G1  0  5
Vy12g4  VY12G4  0  -5
Vy12gr1  VY12GR1  0  5
Vy12gr4  VY12GR4  0  -5
B.8 LOGIC NOT Single cell

* HSPICE file created from LOGICNOT_1CELL.ext - technology: scmos

   .options post
   .TRAN 1ns 4us
   .end

   .option scale=0.3u
   .include model.txt
   m1000 x0/Va VY11GR1 VDD Gnd nfet w=33 l=34 + ad=107 pd=96 as=2217 ps=1222
   m1001 x0/Vb VY11GR4 VSS Vdd pfet w=100 l=34 + ad=404 pd=210 as=6500 ps=2730
   m1002 VY11OUT VY11OUT x0/Va Vdd pfet w=100 l=34 + ad=1592 pd=840 as=404 ps=210
   m1003 VY11OUT VY11OUT x0/Vb Gnd nfet w=33 l=34 + ad=428 pd=384 as=107 ps=90
   m1004 x1/Va VY11G1 VDD Gnd nfet w=33 l=34 + ad=107 pd=96 as=0 ps=0
   m1005 x1/Vb VY11G4 VSS Vdd pfet w=100 l=34 + ad=404 pd=210 as=0 ps=0
   m1006 VY11OUT VY11OUT x1/Va Vdd pfet w=100 l=34 + ad=0 pd=0 as=404 ps=210
   m1007 VY11OUT VY11OUT x1/Vb Gnd nfet w=33 l=34 + ad=0 pd=0 as=107 ps=90
   m1008 x2/Va VYX11GR1 VDD Gnd nfet w=33 l=34 + ad=107 pd=96 as=0 ps=0
   m1009 x2/Vb VYX11GR4 VSS Vdd pfet w=100 l=34 + ad=404 pd=210 as=0 ps=0
   m1010 VY11OUT VY11OUT x2/Va Vdd pfet w=100 l=34 + ad=0 pd=0 as=404 ps=210
   m1011 VY11OUT VY11OUT x2/Vb Gnd nfet w=33 l=34 + ad=0 pd=0 as=107 ps=90
   m1012 x3/Va VYX11G1 VDD Gnd nfet w=33 l=34
m1041 x11/Vb VSS VSS Vdd pfet w=100 l=34  
+ ad=404 pd=210 as=0 ps=0
m1042 x10/VIN x10/VIN x11/Va Vdd pfet w=100 l=34  
+ ad=1194 pd=1076 as=404 ps=210
m1043 x10/VIN x10/VIN x11/Vb Gnd nfet w=33 l=34  
+ ad=321 pd=288 as=107 ps=90
m1044 x12/Va VUC11GR1 VDD Gnd nfet w=33 l=34  
+ ad=107 pd=96 as=0 ps=0
m1045 x12/Vb VUC11GR4 VSS Vdd pfet w=100 l=34  
+ ad=404 pd=210 as=0 ps=0
m1046 x10/VIN x10/VIN x12/Va Vdd pfet w=100 l=34  
+ ad=0 pd=630 as=107 ps=90
m1047 x10/VIN x10/VIN x12/Vb Gnd nfet w=33 l=34  
+ ad=0 pd=0 as=107 ps=90
m1048 x13/Va VUC11G1 VDD Gnd nfet w=33 l=34  
+ ad=107 pd=96 as=0 ps=0
m1049 x13/Vb VUC11G4 VSS Vdd pfet w=100 l=34  
+ ad=404 pd=210 as=0 ps=0
m1050 x10/VIN VUC11 x13/Va Vdd pfet w=100 l=34  
+ ad=0 pd=0 as=404 ps=210
m1051 x10/VIN VUC11 x13/Vb Gnd nfet w=33 l=34  
+ ad=0 pd=0 as=107 ps=90
m1052 VX11INITIAL select VYC11OUT Gnd nfet w=33 l=34  
+ ad=218 pd=180 as=0 ps=0
m1053 in_i11 start VX11INITIAL Gnd nfet w=33 l=34  
+ ad=136 pd=100 as=0 ps=0
C0 VSS x10/VIN 2.7fF  
C1 VUC11 GND 6.4fF
C2 VUC11G4 GND 3.9fF
C3 x10/VIN GND 15.6fF
C4 VUC11G1 GND 2.1fF
C5 VUC11GR4 GND 3.9fF
C6 VUC11GR1 GND 2.1fF
C7 VSS GND 43.7fF
C8 VYC11OUT GND 41.4fF
C9 VX11G4 GND 2.1fF
C10 VX11INITIAL GND 6.1fF
C11 VY11OUT GND 8.5fF
C12 VDD GND 31.4fF

** hspice subcircuit dictionary
* x0 posvartrans_1/transconductance_1
* x1 posvartrans_1/transconductance_0
* x2 posvartrans_16/transconductance_1
* x3 posvartrans_16/transconductance_0
* x4 posvartrans_15/transconductance_1
* x5 posvartrans_15/transconductance_0
* x6 posvartrans_0/transconductance_1
* x7 posvartrans_0/transconductance_0
* x8 negvartrans_0/posvartrans_1/transconductance_1
* x9 negvartrans_0/posvartrans_1/transconductance_0
* x10 negvartrans_0/posvartrans_1
* x11 negvartrans_0/transconductance_3
* x12 negvartrans_0/posvartrans_0/transconductance_1
* x13 negvartrans_0/posvartrans_0/transconductance_0
Vdd VDD 0 5
Vss VSS 0 -5

Vstart start 0 PWL(0 0 100e-9 0 101e-9 5 400e-9 5 401e-9 0 4000e-9 0)
Vselect select 0 PWL(0 0 400e-9 0 401e-9 5 400e-6 5)

Cx11 VX11INITIAL 0 4pF
Vx11initial in_i11 0 DC 0

******** INPUTS *---- Initializing the Input Voltage "U"

Vuc11 VUC11 0 DC 1

**********************************************************************
* Setting the transconductance values

**********************************************************************

Vuc11g1 VUC11G1 0 5
Vuc11g4 VUC11G4 0 -5
Vuc11gr1 VUC11GR1 0 3.3
Vuc11gr4 VUC11GR4 0 -3.3

Vyc11g1 VYC11G1 0 5
Vyc11g4 VYC11G4 0 -5
Vyc11gr1 VYC11GR1 0 5
Vyc11gr4 VYC11GR4 0 -5

******
Vx11g1 VX11G1 0 4
Vx11g4 VX11G4 0 -4
Vx11gr1 VX11GR1 0 5
Vx11gr4 VX11GR4 0 -5

Vy11g1 VY11G1 0 5
Vy11g4 VY11G4 0 -5
Vy11gr1 VY11GR1 0 5
Vy11gr4 VY11GR4 0 -5

******
.options post
.TRAN 1ns 4us
.end
B.9 LOGIC NOT 1×3 CNN

* HSPICE file created from LOGICNOT_3CELLS.ext - technology: scmos

```
.option scale=0.3u
.include model.txt
m1000 x0/Va VY11GR1 VDD Gnd nfet w=33 l=34
  + ad=107 pd=96 as=6615 ps=3666
m1001 x0/Vb VY11GR4 VSS Vdd pfet w=100 l=34
  + ad=404 pd=210 as=19500 ps=8190
m1002 VY13OUT VY13OUT x0/Va Vdd pfet w=100 l=34
  + ad=1592 pd=840 as=404 ps=210
m1003 VY13OUT VY13OUT x0/Vb Gnd nfet w=33 l=34
  + ad=428 pd=384 as=107 ps=90
m1004 x1/Va VY11G1 VDD Gnd nfet w=33 l=34
  + ad=107 pd=96 as=0 ps=0
m1005 x1/Vb VY11G4 VSS Vdd pfet w=100 l=34
  + ad=404 pd=210 as=0 ps=0
m1006 VY13OUT VY13OUT x1/Va Vdd pfet w=100 l=34
  + ad=0 pd=0 as=404 ps=210
m1007 VY13OUT VY13OUT x1/Vb Gnd nfet w=33 l=34
  + ad=0 pd=0 as=107 ps=90
m1008 x2/Va VYX11GR1 VDD Gnd nfet w=33 l=34
  + ad=107 pd=96 as=0 ps=0
m1009 x2/Vb VYX11GR4 VSS Vdd pfet w=100 l=34
  + ad=404 pd=210 as=0 ps=0
m1010 VY13OUT VY13OUT x2/Va Vdd pfet w=100 l=34
  + ad=0 pd=0 as=404 ps=210
m1011 VY13OUT VY13OUT x2/Vb Gnd nfet w=33 l=34
  + ad=0 pd=0 as=107 ps=90
m1012 x3/Va VYX11G1 VDD Gnd nfet w=33 l=34
  + ad=107 pd=96 as=0 ps=0
m1013 x3/Vb VYX11G4 VSS Vdd pfet w=100 l=34
  + ad=404 pd=210 as=0 ps=0
m1014 VY13OUT VX13INITIAL x3/Va Vdd pfet w=100 l=34
  + ad=0 pd=0 as=404 ps=210
m1015 VY13OUT VX13INITIAL x3/Vb Gnd nfet w=33 l=34
  + ad=0 pd=0 as=107 ps=90
m1016 x4/Va VYX11GR1 VDD Gnd nfet w=33 l=34
  + ad=107 pd=96 as=0 ps=0
m1017 x4/Vb VYX11GR4 VSS Vdd pfet w=100 l=34
  + ad=404 pd=210 as=0 ps=0
m1018 VYC13OUT VYC13OUT x4/Va Vdd pfet w=100 l=34
  + ad=2388 pd=1260 as=404 ps=210
m1019 VYC13OUT VYC13OUT x4/Vb Gnd nfet w=33 l=34
  + ad=749 pd=672 as=107 ps=90
m1020 x5/Va VX11G1 VDD Gnd nfet w=33 l=34
  + ad=107 pd=96 as=0 ps=0
m1021 x5/Vb VX11G4 VSS Vdd pfet w=100 l=34
  + ad=404 pd=210 as=0 ps=0
m1022 VYC13OUT VYC13OUT x5/Va Vdd pfet w=100 l=34
  + ad=0 pd=0 as=404 ps=210
m1023 VYC13OUT VYC13OUT x5/Vb Gnd nfet w=33 l=34
  + ad=0 pd=0 as=107 ps=90
m1024 x6/Va VYX11GR1 VDD Gnd nfet w=33 l=34
  + ad=107 pd=96 as=0 ps=0
```

150
<table>
<thead>
<tr>
<th>Device</th>
<th>Description</th>
<th>Width</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>m1054</td>
<td>VY12OUT</td>
<td>100</td>
<td>34</td>
</tr>
<tr>
<td>m1055</td>
<td>VY12OUT</td>
<td>33</td>
<td>34</td>
</tr>
<tr>
<td>m1056</td>
<td>x15/Va VY11G1 VDD Gnd nFET</td>
<td>33</td>
<td>34</td>
</tr>
<tr>
<td>m1057</td>
<td>x15/Vb VY11G4 VSS Vdd pFET</td>
<td>100</td>
<td>34</td>
</tr>
<tr>
<td>m1058</td>
<td>VY12OUT VY12OUT x14/Va Vdd pFET</td>
<td>100</td>
<td>34</td>
</tr>
<tr>
<td>m1059</td>
<td>VY12OUT VY12OUT x15/Vb Gnd nFET</td>
<td>33</td>
<td>34</td>
</tr>
<tr>
<td>m1060</td>
<td>x16/Va VYX11G1 VDD Gnd nFET</td>
<td>33</td>
<td>34</td>
</tr>
<tr>
<td>m1061</td>
<td>x16/Vb VYX11G4 VSS Vdd pFET</td>
<td>100</td>
<td>34</td>
</tr>
<tr>
<td>m1062</td>
<td>VY12OUT VY12OUT x16/Va Vdd pFET</td>
<td>100</td>
<td>34</td>
</tr>
<tr>
<td>m1063</td>
<td>VY12OUT VY12OUT x16/Vb Gnd nFET</td>
<td>33</td>
<td>34</td>
</tr>
<tr>
<td>m1064</td>
<td>x17/Va VYX11G1 VDD Gnd nFET</td>
<td>33</td>
<td>34</td>
</tr>
<tr>
<td>m1065</td>
<td>x17/Vb VYX11G4 VSS Vdd pFET</td>
<td>100</td>
<td>34</td>
</tr>
<tr>
<td>m1066</td>
<td>VY12OUT VY12OUT x17/Va Vdd pFET</td>
<td>100</td>
<td>34</td>
</tr>
<tr>
<td>m1067</td>
<td>VY12OUT VY12OUT x17/Vb Gnd nFET</td>
<td>33</td>
<td>34</td>
</tr>
<tr>
<td>m1068</td>
<td>x18/Va VYX11G1 VDD Gnd nFET</td>
<td>33</td>
<td>34</td>
</tr>
<tr>
<td>m1069</td>
<td>x18/Vb VYX11G4 VSS Vdd pFET</td>
<td>100</td>
<td>34</td>
</tr>
<tr>
<td>m1070</td>
<td>VY12OUT VY12OUT x18/Va Vdd pFET</td>
<td>100</td>
<td>34</td>
</tr>
<tr>
<td>m1071</td>
<td>VY12OUT VY12OUT x18/Vb Gnd nFET</td>
<td>33</td>
<td>34</td>
</tr>
<tr>
<td>m1072</td>
<td>x19/Va VYX11G1 VDD Gnd nFET</td>
<td>33</td>
<td>34</td>
</tr>
<tr>
<td>m1073</td>
<td>x19/Vb VYX11G4 VSS Vdd pFET</td>
<td>100</td>
<td>34</td>
</tr>
<tr>
<td>m1074</td>
<td>VY12OUT VY12OUT x19/Va Vdd pFET</td>
<td>100</td>
<td>34</td>
</tr>
<tr>
<td>m1075</td>
<td>VY12OUT VY12OUT x19/Vb Gnd nFET</td>
<td>33</td>
<td>34</td>
</tr>
<tr>
<td>m1076</td>
<td>x20/Va VYX11G1 VDD Gnd nFET</td>
<td>33</td>
<td>34</td>
</tr>
<tr>
<td>m1077</td>
<td>x20/Vb VYX11G4 VSS Vdd pFET</td>
<td>100</td>
<td>34</td>
</tr>
<tr>
<td>m1078</td>
<td>VY12OUT VY12OUT x20/Va Vdd pFET</td>
<td>100</td>
<td>34</td>
</tr>
<tr>
<td>m1079</td>
<td>VY12OUT VY12OUT x20/Vb Gnd nFET</td>
<td>33</td>
<td>34</td>
</tr>
<tr>
<td>m1080</td>
<td>x21/Va VYX11G1 VDD Gnd nFET</td>
<td>33</td>
<td>34</td>
</tr>
<tr>
<td>m1081</td>
<td>x21/Vb VYX11G4 VSS Vdd pFET</td>
<td>100</td>
<td>34</td>
</tr>
</tbody>
</table>
m1082 VYC12OUT VY11OUT x21/Va Vdd nfet w=100 l=34
  + ad=0 pd=0 as=404 ps=210
m1083 VYC12OUT VY12OUT x21/Vb Gnd nfet w=33 l=34
  + ad=0 pd=0 as=107 ps=90
m1084 x22/Va VUC11GR1 VDD Gnd nfet w=33 l=34
  + ad=107 pd=96 as=0 ps=0
m1085 x22/Vb VUC11GR4 VSS Vdd nfet w=100 l=34
  + ad=404 pd=210 as=0 ps=0
m1086 VYC12OUT VY12OUT x22/Va Vdd nfet w=100 l=34
  + ad=0 pd=0 as=404 ps=210
m1087 VYC12OUT VY12OUT x22/Vb Gnd nfet w=33 l=34
  + ad=0 pd=0 as=107 ps=90
m1088 x23/Va VUC11G1 VDD Gnd nfet w=33 l=34
  + ad=107 pd=96 as=0 ps=0
m1089 x23/Vb VUC11G4 VSS Vdd nfet w=100 l=34
  + ad=404 pd=210 as=0 ps=0
m1090 VYC12OUT x24/VIN x23/Va Vdd nfet w=100 l=34
  + ad=0 pd=0 as=404 ps=210
m1091 VYC12OUT x24/VIN x23/Vb Gnd nfet w=33 l=34
  + ad=0 pd=0 as=107 ps=90
m1092 x25/Va VDD VDD Gnd nfet w=33 l=34
  + ad=107 pd=96 as=0 ps=0
m1093 x25/Vb VSS VSS Vdd nfet w=100 l=34
  + ad=404 pd=210 as=0 ps=0
m1094 x24/VIN x24/VIN x25/Va Vdd nfet w=100 l=34
  + ad=1194 pd=630 as=404 ps=210
m1095 x24/VIN x24/VIN x25/Vb Gnd nfet w=33 l=34
  + ad=321 pd=288 as=107 ps=90
m1096 x26/Va VUC11GR1 VDD Gnd nfet w=33 l=34
  + ad=107 pd=96 as=0 ps=0
m1097 x26/Vb VUC11GR4 VSS Vdd nfet w=100 l=34
  + ad=404 pd=210 as=0 ps=0
m1098 x24/VIN x24/VIN x26/Va Vdd nfet w=100 l=34
  + ad=0 pd=0 as=404 ps=210
m1099 x24/VIN x24/VIN x26/Vb Gnd nfet w=33 l=34
  + ad=0 pd=0 as=107 ps=90
m1100 x27/Va VUC11G1 VDD Gnd nfet w=33 l=34
  + ad=107 pd=96 as=0 ps=0
m1101 x27/Vb VUC11G4 VSS Vdd nfet w=100 l=34
  + ad=404 pd=210 as=0 ps=0
m1102 x24/VIN VUC12 x27/Va Vdd nfet w=100 l=34
  + ad=0 pd=0 as=404 ps=210
m1103 x24/VIN VUC12 x27/Vb Gnd nfet w=33 l=34
  + ad=0 pd=0 as=107 ps=90
m1104 x28/Va VY11GR1 VDD Gnd nfet w=33 l=34
  + ad=107 pd=96 as=0 ps=0
m1105 x28/Vb VY11GR4 VSS Vdd nfet w=100 l=34
  + ad=404 pd=210 as=0 ps=0
m1106 VY11OUT VY11OUT x28/Va Vdd nfet w=100 l=34
  + ad=1592 pd=840 as=404 ps=210
m1107 VY11OUT VY11OUT x28/Vb Gnd nfet w=33 l=34
  + ad=428 pd=384 as=107 ps=90
m1108 x29/Va VY11G1 VDD Gnd nfet w=33 l=34
  + ad=107 pd=96 as=0 ps=0
m1109 x29/Vb VY11G4 VSS Vdd nfet w=100 l=34
  + ad=404 pd=210 as=0 ps=0
m1110 VY11OUT VY11OUT x29/Va Vdd nfet w=100 l=34
m1111 VY11OUT VY11OUT x29/Vb Gnd nfet w=33 l=34
m1112 x30/Va VYX11GR1 VDD Gnd nfet w=33 l=34
m1113 x30/Vb VYX11GR4 VSS Vdd pfet w=100 l=34
m1114 VY11OUT VY11OUT x30/Va Vdd pfet w=100 l=34
m1115 VY11OUT VY11OUT x30/Vb Gnd nfet w=33 l=34
m1116 x31/Va VYX11G1 VDD Gnd nfet w=33 l=34
m1117 x31/Vb VYX11G4 VSS Vdd pfet w=100 l=34
m1118 VY11OUT VX11INITIAL x31/Va Vdd pfet w=100 l=34
m1119 VY11OUT VX11INITIAL x31/Vb Gnd nfet w=33 l=34
m1120 x32/Va VX11GR1 VDD Gnd nfet w=33 l=34
m1121 x32/Vb VX11GR4 VSS Vdd pfet w=100 l=34
m1122 VYC11OUT VYC11OUT x32/Va Vdd pfet w=100 l=34
m1123 VYC11OUT VYC11OUT x32/Vb Gnd nfet w=33 l=34
m1124 x33/Va VX11G1 VDD Gnd nfet w=33 l=34
m1125 x33/Vb VX11G4 VSS Vdd pfet w=100 l=34
m1126 VYC11OUT VYC11OUT x33/Va Vdd pfet w=100 l=34
m1127 VYC11OUT VYC11OUT x33/Vb Gnd nfet w=33 l=34
m1128 x34/Va VYC11GR1 VDD Gnd nfet w=33 l=34
m1129 x34/Vb VYC11GR4 VSS Vdd pfet w=100 l=34
m1130 VYC11OUT VYC11OUT x34/Va Vdd pfet w=100 l=34
m1131 VYC11OUT VYC11OUT x34/Vb Gnd nfet w=33 l=34
m1132 x35/Va VYC11G1 VDD Gnd nfet w=33 l=34
m1133 x35/Vb VYC11G4 VSS Vdd pfet w=100 l=34
m1134 VYC11OUT VY11OUT x35/Va Vdd pfet w=100 l=34
m1135 VYC11OUT VY11OUT x35/Vb Gnd nfet w=33 l=34
m1136 x36/Va VUC11GR1 VDD Gnd nfet w=33 l=34
m1137 x36/Vb VUC11GR4 VSS Vdd pfet w=100 l=34
m1138 VYC11OUT VYC11OUT x36/Va Vdd pfet w=100 l=34
m1139 VYC11OUT VYC11OUT x36/Vb Gnd nfet w=33 l=34
  + ad=0 pd=0 as=107 ps=90
m1140 x37/Va VUC11G1 VDD Gnd nfet w=33 l=34
  + ad=107 pd=96 as=0 ps=0
m1141 x37/Vb VUC11G4 VSS Vdd pfet w=100 l=34
  + ad=404 pd=210 as=0 ps=0
m1142 VYC11OUT x38/VIN x37/Va Vdd pfet w=100 l=34
  + ad=0 pd=0 as=404 ps=210
m1143 VYC11OUT x38/VIN x37/Vb Gnd nfet w=33 l=34
  + ad=0 pd=0 as=107 ps=90
m1144 x39/Va VDD VDD Gnd nfet w=33 l=34
  + ad=107 pd=96 as=0 ps=0
m1145 x39/Vb VSS VSS Vdd pfet w=100 l=34
  + ad=404 pd=210 as=0 ps=0
m1146 x38/VIN x38/VIN x39/Va Vdd pfet w=100 l=34
  + ad=1194 pd=630 as=404 ps=210
m1147 x38/VIN x38/VIN x39/Vb Gnd nfet w=33 l=34
  + ad=321 pd=288 as=107 ps=90
m1148 x40/Va VUC11GR1 VDD Gnd nfet w=33 l=34
  + ad=107 pd=96 as=0 ps=0
m1149 x40/Vb VUC11GR4 VSS Vdd pfet w=100 l=34
  + ad=404 pd=210 as=0 ps=0
m1150 x38/VIN x38/VIN x40/Va Vdd pfet w=100 l=34
  + ad=0 pd=0 as=404 ps=210
m1151 x38/VIN x38/VIN x40/Vb Gnd nfet w=33 l=34
  + ad=0 pd=0 as=107 ps=90
m1152 x41/Va VUC11G1 VDD Gnd nfet w=33 l=34
  + ad=107 pd=96 as=0 ps=0
m1153 x41/Vb VUC11G4 VSS Vdd pfet w=100 l=34
  + ad=404 pd=210 as=0 ps=0
m1154 x38/VIN VUC11 x41/Va Vdd pfet w=100 l=34
  + ad=0 pd=0 as=404 ps=210
m1155 x38/VIN VUC11 x41/Vb Gnd nfet w=33 l=34
  + ad=0 pd=0 as=107 ps=90
m1156 VX11INITIAL select VYC11OUT Gnd nfet w=33 l=34
  + ad=218 pd=180 as=0 ps=0
m1157 in_i11 start VX11INITIAL Gnd nfet w=33 l=34
  + ad=136 pd=100 as=0 ps=0
m1158 VX12INITIAL select VYC12OUT Gnd nfet w=33 l=34
  + ad=218 pd=180 as=0 ps=0
m1159 in_i12 start VX12INITIAL Gnd nfet w=33 l=34
  + ad=136 pd=100 as=0 ps=0
m1160 VX13INITIAL select VYC13OUT Gnd nfet w=33 l=34
  + ad=218 pd=180 as=0 ps=0
m1161 in_i13 start VX13INITIAL Gnd nfet w=33 l=34
  + ad=136 pd=100 as=0 ps=0
C0 VSS x24/VIN 2.7fF
C1 VSS x38/VIN 2.7fF
C2 VSS x10/VIN 2.7fF
C3 start GND 4.7fF
C4 select GND 4.8fF
C5 VY12OUT GND 38.6fF
C6 VUC11 GND 6.5fF
C7 x38/VIN GND 15.6fF
C8 VYC11OUT GND 41.3fF
C9 VX11INITIAL GND 5.9fF
C10 VY11OUT GND 8.2fF
C11 VUC12 GND 6.5fF
C12 x24/VIN GND 15.6fF
C13 VDD GND 80.8fF
C14 VY12OUT GND 32.3fF
C15 VX12INITIAL GND 7.0fF
C16 VUC13 GND 6.6fF
C17 VUC11G4 GND 5.4fF
C18 x10/VIN GND 15.6fF
C19 VUC11GR4 GND 5.9fF
C20 VYC11G4 GND 2.8fF
C21 VSS GND 72.3fF
C22 VYC11GR4 GND 3.1fF
C23 VYC11GR1 GND 2.3fF
C24 VX11G4 GND 2.2fF
C25 VX11G1 GND 2.6fF
C26 VY13OUT GND 41.2fF
C27 VX11GR4 GND 2.4fF
C28 VX11GR1 GND 2.6fF
C29 VY11G4 GND 3.1fF
C30 VX13INITIAL GND 10.9fF
C31 VY11G1 GND 2.7fF
C32 VY11GR4 GND 2.2fF
C33 VY11GR1 GND 2.7fF
C34 VY11G4 GND 2.2fF
C35 VY11G1 GND 2.6fF
C36 VY13OUT GND 46.9fF
C37 VY11GR4 GND 2.2fF
C38 VY11GR1 GND 2.6fF

** hspice subcircuit dictionary
  * x0  posvartrans_9/transconductance_1
  * x1  posvartrans_9/transconductance_0
  * x2  posvartrans_8/transconductance_1
  * x3  posvartrans_8/transconductance_0
  * x4  posvartrans_7/transconductance_1
  * x5  posvartrans_7/transconductance_0
  * x6  posvartrans_6/transconductance_1
  * x7  posvartrans_6/transconductance_0
  * x8  negvartrans_2/posvartrans_1/transconductance_1
  * x9  negvartrans_2/posvartrans_1/transconductance_0
  * x10 negvartrans_2/posvartrans_1
  * x11 negvartrans_2/transconductance_3
  * x12 negvartrans_2/posvartrans_0/transconductance_1
  * x13 negvartrans_2/posvartrans_0/transconductance_0
  * x14 posvartrans_5/transconductance_1
  * x15 posvartrans_5/transconductance_0
  * x16 posvartrans_4/transconductance_1
  * x17 posvartrans_4/transconductance_0
  * x18 posvartrans_3/transconductance_1
  * x19 posvartrans_3/transconductance_0
  * x20 posvartrans_2/transconductance_1
  * x21 posvartrans_2/transconductance_0
  * x22 negvartrans_1/posvartrans_1/transconductance_1
  * x23 negvartrans_1/posvartrans_1/transconductance_0
  * x24 negvartrans_1/posvartrans_1
  * x25 negvartrans_1/transconductance_3
  * x26 negvartrans_1/posvartrans_0/transconductance_1
* x27 negvartrans_1/posvartrans_0/transconductance_0
* x28 posvartrans_1/transconductance_1
* x29 posvartrans_1/transconductance_0
* x30 posvartrans_16/transconductance_1
* x31 posvartrans_16/transconductance_0
* x32 posvartrans_15/transconductance_1
* x33 posvartrans_15/transconductance_0
* x34 posvartrans_0/transconductance_1
* x35 posvartrans_0/transconductance_0
* x36 negvartrans_0/posvartrans_1/transconductance_1
* x37 negvartrans_0/posvartrans_1/transconductance_0
* x38 negvartrans_0/posvartrans_1
* x39 negvartrans_0/transconductance_3
* x40 negvartrans_0/posvartrans_0/transconductance_1
* x41 negvartrans_0/posvartrans_0/transconductance_0

**********************************************************************

Vdd VDD 0 5
Vss VSS 0 -5

Vstart start 0 PWL(0 0 100e-9 0 101e-9 5 400e-9 5 401e-9 0 4000e-9 0)
Vselect select 0 PWL(0 0 400e-9 0 401e-9 5 400e-6 5)

Cx11 VX11INITIAL 0 4pF
Vx11initial in_i11 0 DC 0

Cx12 VX12INITIAL 0 4pF
Vx12initial in_i12 0 DC 0

Cx13 VX13INITIAL 0 4pF
Vx13initial in_i13 0 DC 0

*************INITIALIZING INPUT VECTOR*************

Vuc11 VUC11 0 DC -1
Vuc12 VUC12 0 DC 1
Vuc13 VUC13 0 DC -1

*************INITIALIZING INPUT VECTOR*************

* Setting the transconductance values

***************

Vuc11g1 VUC11G1 0 5
Vuc11g4 VUC11G4 0 -5
Vuc11gr1 VUC11GR1 0 3.3
Vuc11gr4 VUC11GR4 0 -3.3

Vyc11g1 VYC11G1 0 5
Vyc11g4 VYC11G4 0 -5
Vyc11gr1 VYC11GR1 0 5
Vyc11gr4 VYC11GR4 0 -5
Vx11g1 VX11G1 0 4
Vx11g4 VX11G4 0 -4
Vx11gr1 VX11GR1 0 5
Vx11gr4 VX11GR4 0 -5
Vyx11g1 VYX11G1 0 5
Vyx11g4 VYX11G4 0 -5
Vyx11gr1 VYX11GR1 0 5
Vyx11gr4 VYX11GR4 0 -5
Vy11g1 VY11G1 0 5
Vy11g4 VY11G4 0 -5
Vy11gr1 VY11GR1 0 5
Vy11gr4 VY11GR4 0 -5

.options post
.TRAN 1ns 4us
.end

B.10 SHIFT WEST 1x3 CNN

* HSPICE file created from SHIFTWEST_3CELLS.ext - technology: scmos

.include model.txt
m1000 x0/Va VY13G1 VDD Gnd nfet w=33 l=34 + ad=107 pd=96 as=6084 ps=3384
m1001 x0/Vb VY13G4 VDD VSS pfet w=100 l=34 + ad=404 pd=210 as=18000 ps=7560
m1002 VY13OUT VY13OUT x0/Va Vdd pfet w=100 l=34 + ad=1592 pd=840 as=404 ps=210
m1003 VY13OUT VY13OUT x0/Vb Gnd nfet w=33 l=34 + ad=428 pd=384 as=107 ps=90
m1004 x1/Va VY13GR1 VDD Gnd nfet w=33 l=34 + ad=107 pd=96 as=0 ps=0
m1005 x1/Vb VY13GR4 VDD VSS pfet w=100 l=34 + ad=404 pd=210 as=0 ps=0
m1006 VY13OUT VY13OUT x1/Va Vdd pfet w=100 l=34 + ad=0 pd=0 as=404 ps=210
m1007 VY13OUT VY13OUT x1/Vb Gnd nfet w=33 l=34 + ad=0 pd=0 as=107 ps=90
m1008 x2/Va VYX13G1 VDD Gnd nfet w=33 l=34 + ad=107 pd=96 as=0 ps=0
m1009 x2/Vb VYX13G4 VDD VSS pfet w=100 l=34 + ad=404 pd=210 as=0 ps=0
m1010 VY13OUT VY13OUT x2/Va Vdd pfet w=100 l=34 + ad=0 pd=0 as=404 ps=210
m1011 VY13OUT VY13OUT x2/Vb Gnd nfet w=33 l=34 + ad=0 pd=0 as=107 ps=90
m1012 x3/Va VYX13G1 VDD Gnd nfet w=33 l=34 + ad=107 pd=96 as=0 ps=0
m1042 VY12OUT VY12OUT x10/Va Vdd pFET w=100 l=34
m1043 VY12OUT VY12OUT x10/Vb Gnd nFET w=33 l=34
m1044 x11/Va VY12G1 VDD Gnd nFET w=33 l=34
m1045 x11/Vb VY12G4 VSS Vdd pFET w=100 l=34
m1046 VY12OUT VY12OUT x11/Va Vdd pFET w=100 l=34
m1047 VY12OUT VY12OUT x11/Vb Gnd nFET w=33 l=34
m1048 x12/Va VYX12GR1 VDD Gnd nFET w=33 l=34
m1049 x12/Vb VYX12GR4 VSS Vdd pFET w=100 l=34
m1050 VY12OUT VY12OUT x12/Va Vdd pFET w=100 l=34
m1051 VY12OUT VY12OUT x12/Vb Gnd nFET w=33 l=34
m1052 x13/Va VYX12G1 VDD Gnd nFET w=33 l=34
m1053 x13/Vb VYX12G4 VSS Vdd pFET w=100 l=34
m1054 VY12OUT VX12INITIAL x13/Va Vdd pFET w=100 l=34
m1055 VY12OUT VX12INITIAL x13/Vb Gnd nFET w=33 l=34
m1056 x14/Va VYX12GR1 VDD Gnd nFET w=33 l=34
m1057 x14/Vb VYX12GR4 VSS Vdd pFET w=100 l=34
m1058 VYC12OUT VYC12OUT x14/Va Vdd pFET w=100 l=34
m1059 VYC12OUT VYC12OUT x14/Vb Gnd nFET w=33 l=34
m1060 x15/Va VYX12G1 VDD Gnd nFET w=33 l=34
m1061 x15/Vb VYX12G4 VSS Vdd pFET w=100 l=34
m1062 VYC12OUT VYC12OUT x15/Va Vdd pFET w=100 l=34
m1063 VYC12OUT VYC12OUT x15/Vb Gnd nFET w=33 l=34
m1064 x16/Va VUW12GR1 VDD Gnd nFET w=33 l=34
m1065 x16/Vb VUW12GR4 VSS Vdd pFET w=100 l=34
m1066 VYC12OUT VYC12OUT x16/Va Vdd pFET w=100 l=34
m1067 VYC12OUT VYC12OUT x16/Vb Gnd nFET w=33 l=34
m1068 x17/Va VUW12G1 VDD Gnd nFET w=33 l=34
m1069 x17/Vb VUW12G4 VSS Vdd pFET w=100 l=34
m1070 VYC12OUT VUC13 x17/Va Vdd pfet w=100 l=34
+ ad=0 pd=0 as=404 ps=210
m1071 VYC12OUT VUC13 x17/Vb Gnd nfet w=33 l=34
+ ad=0 pd=0 as=107 ps=90
m1072 x18/Va VYC12GR1 VDD Gnd nfet w=33 l=34
+ ad=107 pd=96 as=0 ps=0
m1073 x18/Vb VYC12GR4 VSS Vdd pfet w=100 l=34
+ ad=404 pd=210 as=0 ps=0
m1074 VYC12OUT VYC12OUT x18/Va Vdd pfet w=100 l=34
+ ad=0 pd=0 as=404 ps=210
m1075 VYC12OUT VYC12OUT x18/Vb Gnd nfet w=33 l=34
+ ad=0 pd=0 as=107 ps=90
m1076 x19/Va VYC12G1 VDD Gnd nfet w=33 l=34
+ ad=107 pd=96 as=0 ps=0
m1077 x19/Vb VYC12G4 VSS Vdd pfet w=100 l=34
+ ad=404 pd=210 as=0 ps=0
m1078 VYC12OUT VY12OUT x19/Va Vdd pfet w=100 l=34
+ ad=0 pd=0 as=404 ps=210
m1079 VYC12OUT VY12OUT x19/Vb Gnd nfet w=33 l=34
+ ad=0 pd=0 as=107 ps=90
m1080 x20/Va VUC13GR1 VDD Gnd nfet w=33 l=34
+ ad=107 pd=96 as=0 ps=0
m1081 x20/Vb VUC13GR4 VSS Vdd pfet w=100 l=34
+ ad=404 pd=210 as=0 ps=0
m1082 VYC13OUT VYC13OUT x20/Va Vdd pfet w=100 l=34
+ ad=0 pd=0 as=404 ps=210
m1083 VYC13OUT VYC13OUT x20/Vb Gnd nfet w=33 l=34
+ ad=0 pd=0 as=107 ps=90
m1084 x21/Va VUC13G1 VDD Gnd nfet w=33 l=34
+ ad=107 pd=96 as=0 ps=0
m1085 x21/Vb VUC13G4 VSS Vdd pfet w=100 l=34
+ ad=404 pd=210 as=0 ps=0
m1086 VYC13OUT VUC13 x21/Va Vdd pfet w=100 l=34
+ ad=0 pd=0 as=404 ps=210
m1087 VYC13OUT VUC13 x21/Vb Gnd nfet w=33 l=34
+ ad=0 pd=0 as=107 ps=90
m1088 x22/Va VY11GR1 VDD Gnd nfet w=33 l=34
+ ad=107 pd=96 as=0 ps=0
m1089 x22/Vb VY11GR4 VSS Vdd pfet w=100 l=34
+ ad=404 pd=210 as=0 ps=0
m1090 VY11OUT VY11OUT x22/Va Vdd pfet w=100 l=34
+ ad=1592 pd=836 as=404 ps=210
m1091 VY11OUT VY11OUT x22/Vb Gnd nfet w=33 l=34
+ ad=428 pd=384 as=107 ps=90
m1092 x23/Va VY11G1 VDD Gnd nfet w=33 l=34
+ ad=107 pd=96 as=0 ps=0
m1093 x23/Vb VY11G4 VSS Vdd pfet w=100 l=34
+ ad=404 pd=210 as=0 ps=0
m1094 VY11OUT VY11OUT x23/Va Vdd pfet w=100 l=34
+ ad=0 pd=0 as=404 ps=210
m1095 VY11OUT VY11OUT x23/Vb Gnd nfet w=33 l=34
+ ad=0 pd=0 as=107 ps=90
m1096 x24/Va VYX11GR1 VDD Gnd nfet w=33 l=34
+ ad=107 pd=96 as=0 ps=0
m1097 x24/Vb VYX11GR4 VSS Vdd pfet w=100 l=34
+ ad=404 pd=210 as=0 ps=0
m1098 VY11OUT VY11OUT x24/Va Vdd pfet w=100 l=34
m1127 VYC11OUT VY11OUT x31/Vb Gnd nfet w=33 l=34
+ ad=0 pd=0 as=107 ps=90
m1128 x32/Va VUC12GR1 VDD Gnd nfet w=33 l=34
+ ad=107 pd=96 as=0 ps=0
m1129 x32/Vb VUC12GR4 VSS Vdd pfet w=100 l=34
+ ad=404 pd=210 as=0 ps=0
m1130 VYC12OUT VYC12OUT x32/Va Vdd pfet w=100 l=34
+ ad=0 pd=0 as=404 ps=210
m1131 VYC12OUT VYC12OUT x32/Vb Gnd nfet w=33 l=34
+ ad=0 pd=0 as=107 ps=90
m1132 x33/Va VUC12G1 VDD Gnd nfet w=33 l=34
+ ad=107 pd=96 as=0 ps=0
m1133 x33/Vb VUC12G4 VSS Vdd pfet w=100 l=34
+ ad=404 pd=210 as=0 ps=0
m1134 VYC12OUT VUC12 x33/Va Vdd pfet w=100 l=34
+ ad=0 pd=0 as=404 ps=210
m1135 VYC12OUT VUC12 x33/Vb Gnd nfet w=33 l=34
+ ad=0 pd=0 as=107 ps=90
m1136 x34/Va VUC11GR1 VDD Gnd nfet w=33 l=34
+ ad=107 pd=96 as=0 ps=0
m1137 x34/Vb VUC11GR4 VSS Vdd pfet w=100 l=34
+ ad=404 pd=210 as=0 ps=0
m1138 VYC11OUT VYC11OUT x34/Va Vdd pfet w=100 l=34
+ ad=0 pd=0 as=404 ps=210
m1139 VYC11OUT VYC11OUT x34/Vb Gnd nfet w=33 l=34
+ ad=0 pd=0 as=107 ps=90
m1140 x35/Va VUC11G1 VDD Gnd nfet w=33 l=34
+ ad=107 pd=96 as=0 ps=0
m1141 x35/Vb VUC11G4 VSS Vdd pfet w=100 l=34
+ ad=404 pd=210 as=0 ps=0
m1142 VYC11OUT VUC11 x35/Va Vdd pfet w=100 l=34
+ ad=0 pd=0 as=404 ps=210
m1143 VYC11OUT VUC11 x35/Vb Gnd nfet w=33 l=34
+ ad=0 pd=0 as=107 ps=90
m1144 VX11INITIAL select VYC11OUT Gnd nfet w=33 l=34
+ ad=218 pd=180 as=0 ps=0
m1145 in_i11 start VX11INITIAL Gnd nfet w=33 l=34
+ ad=136 pd=100 as=0 ps=0
m1146 VX12INITIAL select VYC12OUT Gnd nfet w=33 l=34
+ ad=218 pd=180 as=0 ps=0
m1147 in_i12 start VX12INITIAL Gnd nfet w=33 l=34
+ ad=136 pd=100 as=0 ps=0
m1148 VX13INITIAL select VYC13OUT Gnd nfet w=33 l=34
+ ad=218 pd=180 as=0 ps=0
m1149 in_i13 start VX13INITIAL Gnd nfet w=33 l=34
+ ad=136 pd=100 as=0 ps=0
C0 start GND 4.7fF
C1 select GND 4.8fF
C2 VUC11 GND 34.2fF
C3 VUC12 GND 38.5fF
C4 VX11INITIAL GND 7.5fF
C5 VY11OUT GND 29.9fF
C6 VUC13 GND 38.8fF
C7 VYC13OUT GND 12.5fF
C8 VYC12OUT GND 12.5fF
C9 VX12INITIAL GND 8.0fF
C10 VY12OUT GND 32.2fF
C11 VUU13 GND 37.8fF
C12 VX13INITIAL GND 8.0fF
C13 VYX13GR4 GND 2.2fF

** hspice subcircuit dictionary
* x0 posvartrans_11/transconductance_1
* x1 posvartrans_11/transconductance_0
* x2 posvartrans_10/transconductance_1
* x3 posvartrans_10/transconductance_0
* x4 posvartrans_9/transconductance_1
* x5 posvartrans_9/transconductance_0
* x6 posvartrans_8/transconductance_1
* x7 posvartrans_8/transconductance_0
* x8 posvartrans_7/transconductance_1
* x9 posvartrans_7/transconductance_0
* x10 posvartrans_5/transconductance_1
* x11 posvartrans_5/transconductance_0
* x12 posvartrans_4/transconductance_1
* x13 posvartrans_4/transconductance_0
* x14 posvartrans_3/transconductance_1
* x15 posvartrans_3/transconductance_0
* x16 posvartrans_2/transconductance_1
* x17 posvartrans_2/transconductance_0
* x18 posvartrans_1/transconductance_1
* x19 posvartrans_1/transconductance_0
* x20 posvartrans_6/transconductance_1
* x21 posvartrans_6/transconductance_0
* x22 posvartrans_17/transconductance_1
* x23 posvartrans_17/transconductance_0
* x24 posvartrans_16/transconductance_1
* x25 posvartrans_16/transconductance_0
* x26 posvartrans_15/transconductance_1
* x27 posvartrans_15/transconductance_0
* x28 posvartrans_14/transconductance_1
* x29 posvartrans_14/transconductance_0
* x30 posvartrans_13/transconductance_1
* x31 posvartrans_13/transconductance_0
* x32 posvartrans_0/transconductance_1
* x33 posvartrans_0/transconductance_0
* x34 posvartrans_12/transconductance_1
* x35 posvartrans_12/transconductance_0

******************************
Vdd VDD 0 5
Vss VSS 0 -5
Vstart start 0 PWL(0 0 100e-9 0 101e-9 5 400e-9 5 401e-9 0 4000e-9 0)
Vselect select 0 PWL(0 0 400e-9 0 401e-9 5 400e-6 5)

Cx11 VX11INITIAL 0 4pF
Cx12 VX12INITIAL 0 4pF
Cx13 VX13INITIAL 0 4pF

******* INPUTS *---- Initializing the State "VXINITIAL" and Input Voltage "U"
Vx11 initial in i11 0 DC 0 * make VXIN =0 in SHIFTWEST
Vuc11 VUC11 0 DC -1

Vx12 initial in i12 0 DC 0
Vuc12 VUC12 0 DC 1

Vx13 initial in i13 0 DC 0
Vuc13 VUC13 0 DC -1
Vuw13 VUW13 0 1

******************** Setting the transconductance values
Vuc11g1 VUC11G1 0 0  *guc=0 for shift west cell
Vuc11g4 VUC11G4 0 0
Vuc11gr1 VUC11GR1 0 0
Vuc11gr4 VUC11GR4 0 0
Vyc11g1 VYC11G1 0 5  *gyc=1
Vyc11g4 VYC11G4 0 -5
Vyc11gr1 VYC11GR1 0 5
Vyc11gr4 VYC11GR4 0 -5

Vuw11g1 VUW11G1 0 5  *guw=1 for shift west cell
Vuw11g4 VUW11G4 0 -5
Vuw11gr1 VUW11GR1 0 5
Vuw11gr4 VUW11GR4 0 -5

*************
Vuc12g1 VUC12G1 0 0
Vuc12g4 VUC12G4 0 0
Vuc12gr1 VUC12GR1 0 0
Vuc12gr4 VUC12GR4 0 0
Vyc12g1 VYC12G1 0 5
Vyc12g4 VYC12G4 0 -5
Vyc12gr1 VYC12GR1 0 5
Vyc12gr4 VYC12GR4 0 -5

Vuw12g1 VUW12G1 0 5
Vuw12g4 VUW12G4 0 -5
Vuw12gr1 VUW12GR1 0 5
Vuw12gr4 VUW12GR4 0 -5

*************
Vuc13g1 VUC13G1 0 0
Vuc13g4 VUC13G4 0 0
Vuc13gr1 VUC13GR1 0 0
Vuc13gr4 VUC13GR4 0 0
Vyc13g1 VYC13G1 0 5
Vyc13g4 VYC13G4 0 -5
Vyc13gr1 VYC13GR1 0 5
Vyc13gr4 VYC13GR4 0 -5

Vuw13g1 VUW13G1 0 5
.options post
.TRAN 1ns 4us
.end
B.11 SHIFT EAST 1×3 CNN

* HSPICE file created from SHIFTEAST_3CELLS.ext - technology: scmos

.option scale=0.3u
.include model.txt
m1000 x0/Va VY13GR1 VDD Gnd nfet w=33 l=34
+ ad=107 pd=96 as=6084 ps=3384
m1001 x0/Vb VY13GR4 VSS Vdd pfet w=100 l=34
+ ad=404 pd=210 as=18000 ps=7560
m1002 VY13OUT VY13OUT x0/Va Vdd pfet w=100 l=34
+ ad=1592 pd=840 as=404 ps=210
m1003 VY13OUT VY13OUT x0/Vb Gnd nfet w=33 l=34
+ ad=428 pd=384 as=107 ps=90
m1004 x1/Va VY13G1 VDD Gnd nfet w=33 l=34
+ ad=107 pd=96 as=0 ps=0
m1005 x1/Vb VY13G4 VSS Vdd pfet w=100 l=34
+ ad=404 pd=210 as=0 ps=0
m1006 VY13OUT VY13OUT x1/Va Vdd pfet w=100 l=34
+ ad=0 pd=0 as=404 ps=210
m1007 VY13OUT VY13OUT x1/Vb Gnd nfet w=33 l=34
+ ad=0 pd=0 as=107 ps=90
m1008 x2/Va VYX13GR1 VDD Gnd nfet w=33 l=34
+ ad=107 pd=96 as=0 ps=0
m1009 x2/Vb VYX13GR4 VSS Vdd pfet w=100 l=34
+ ad=404 pd=210 as=0 ps=0
m1010 VY13OUT VY13OUT x2/Va Vdd pfet w=100 l=34
+ ad=0 pd=0 as=404 ps=210
m1011 VY13OUT VY13OUT x2/Vb Gnd nfet w=33 l=34
+ ad=0 pd=0 as=107 ps=90
m1012 x3/Va VYX13G1 VDD Gnd nfet w=33 l=34
+ ad=107 pd=96 as=0 ps=0
m1013 x3/Vb VYX13G4 VSS Vdd pfet w=100 l=34
+ ad=404 pd=210 as=0 ps=0
m1014 VY13OUT VX13INITIAL x3/Va Vdd pfet w=100 l=34
+ ad=0 pd=0 as=404 ps=210
m1015 VY13OUT VX13INITIAL x3/Vb Gnd nfet w=33 l=34
+ ad=0 pd=0 as=107 ps=90
m1016 x4/Va VY13GR1 VDD Gnd nfet w=33 l=34
+ ad=107 pd=96 as=0 ps=0
m1017 x4/Vb VY13GR4 VSS Vdd pfet w=100 l=34
+ ad=404 pd=210 as=0 ps=0
m1018 VYC13OUT VYC13OUT x4/Va Vdd pfet w=100 l=34
+ ad=3184 pd=1680 as=404 ps=210
m1019 VYC13OUT VYC13OUT x4/Vb Gnd nfet w=33 l=34
+ ad=963 pd=864 as=107 ps=90
m1020 x5/Va VY13G1 VDD Gnd nfet w=33 l=34
+ ad=107 pd=96 as=0 ps=0
m1021 x5/Vb VY13G4 VSS Vdd pfet w=100 l=34
+ ad=404 pd=210 as=0 ps=0
m1022 VYC13OUT VYC13OUT x5/Va Vdd pfet w=100 l=34
+ ad=0 pd=0 as=404 ps=210
m1023 VYC13OUT VYC13OUT x5/Vb Gnd nfet w=33 l=34
+ ad=0 pd=0 as=107 ps=90
m1081 x20/Vb VUC13GR4 VSS Vdd pfet w=100 l=34
  + ad=404 pd=210 as=0 ps=0
m1082 VYC13OUT VYC13OUT x20/Va Vdd pfet w=100 l=34
  + ad=0 pd=0 as=404 ps=210
m1083 VYC13OUT VYC13OUT x20/Vb Gnd nfet w=33 l=34
  + ad=0 pd=0 as=107 ps=90
m1084 x21/Va VUC13G1 VDD Gnd nfet w=33 l=34
  + ad=107 pd=96 as=0 ps=0
m1085 x21/Vb VUC13G4 VSS Vdd pfet w=100 l=34
  + ad=404 pd=210 as=0 ps=0
m1086 VYC13OUT VUC13 x21/Va Vdd pfet w=100 l=34
  + ad=0 pd=0 as=404 ps=210
m1087 VYC13OUT VUC13 x21/Vb Gnd nfet w=33 l=34
  + ad=0 pd=0 as=107 ps=90
m1088 x22/Va VY11GR1 VDD Gnd nfet w=33 l=34
  + ad=107 pd=96 as=0 ps=0
m1089 x22/Vb VY11GR4 VSS Vdd pfet w=100 l=34
  + ad=404 pd=210 as=0 ps=0
m1090 VY11OUT VY11OUT x22/Va Vdd pfet w=100 l=34
  + ad=1592 pd=836 as=404 ps=210
m1091 VY11OUT VY11OUT x22/Vb Gnd nfet w=33 l=34
  + ad=428 pd=384 as=107 ps=90
m1092 x23/Va VY11G1 VDD Gnd nfet w=33 l=34
  + ad=107 pd=96 as=0 ps=0
m1093 x23/Vb VY11G4 VSS Vdd pfet w=100 l=34
  + ad=404 pd=210 as=0 ps=0
m1094 VY11OUT VY11OUT x23/Va Vdd pfet w=100 l=34
  + ad=0 pd=0 as=404 ps=210
m1095 VY11OUT VY11OUT x23/Vb Gnd nfet w=33 l=34
  + ad=0 pd=0 as=107 ps=90
m1096 x24/Va VYX11GR1 VDD Gnd nfet w=33 l=34
  + ad=107 pd=96 as=0 ps=0
m1097 x24/Vb VYX11GR4 VSS Vdd pfet w=100 l=34
  + ad=404 pd=210 as=0 ps=0
m1098 VY11OUT VY11OUT x24/Va Vdd pfet w=100 l=34
  + ad=0 pd=0 as=404 ps=210
m1099 VY11OUT VY11OUT x24/Vb Gnd nfet w=33 l=34
  + ad=0 pd=0 as=107 ps=90
m1100 x25/Va VYX11G1 VDD Gnd nfet w=33 l=34
  + ad=107 pd=96 as=0 ps=0
m1101 x25/Vb VYX11G4 VSS Vdd pfet w=100 l=34
  + ad=404 pd=210 as=0 ps=0
m1102 VY11OUT VX11INITIAL x25/Va Vdd pfet w=100 l=34
  + ad=0 pd=0 as=404 ps=210
m1103 VY11OUT VX11INITIAL x25/Vb Gnd nfet w=33 l=34
  + ad=0 pd=0 as=107 ps=90
m1104 x26/Va VX11GR1 VDD Gnd nfet w=33 l=34
  + ad=107 pd=96 as=0 ps=0
m1105 x26/Vb VX11GR4 VSS Vdd pfet w=100 l=34
  + ad=404 pd=210 as=0 ps=0
m1106 VYCL1OUT VYCL1OUT x26/Va Vdd pfet w=100 l=34
  + ad=3184 pd=1680 as=404 ps=210
m1107 VYCL1OUT VYCL1OUT x26/Vb Gnd nfet w=33 l=34
  + ad=963 pd=864 as=107 ps=90
m1108 x27/Va VX11G1 VDD Gnd nfet w=33 l=34
  + ad=107 pd=96 as=0 ps=0
m1109 x27/Vb VX11G4 VSS Vdd pfet w=100 l=34
** hspice subcircuit dictionary
* x0 posvartrans_11/transconductance_1
* x1 posvartrans_11/transconductance_0
* x2 posvartrans_10/transconductance_1
* x3 posvartrans_10/transconductance_0
* x4 posvartrans_9/transconductance_1
* x5 posvartrans_9/transconductance_0
* x6 posvartrans_8/transconductance_1
* x7 posvartrans_8/transconductance_0
* x8 posvartrans_7/transconductance_1
* x9 posvartrans_7/transconductance_0
* x10 posvartrans_6/transconductance_1
* x11 posvartrans_6/transconductance_0
* x12 posvartrans_5/transconductance_1
* x13 posvartrans_5/transconductance_0
* x14 posvartrans_4/transconductance_1
* x15 posvartrans_4/transconductance_0
* x16 posvartrans_3/transconductance_1
* x17 posvartrans_2/transconductance_1

C0 start GND 4.7fF
C1 select GND 4.8fF
C2 VUC11 GND 60.2fF
C3 VUC12 GND 31.3fF
C4 VUC13 GND 6.4fF
C5 VX11INITIAL GND 7.5fF
C6 VY11OUT GND 29.9fF
C7 VUC13 GND 52.2fF
C8 VYC13OUT GND 12.7fF
C9 VYC12OUT GND 12.6fF
C10 VX12INITIAL GND 8.0fF
C11 VY12OUT GND 32.2fF
C12 VX13INITIAL GND 8.0fF
C13 VX13GR4 GND 2.2fF
* x17 posvartrans_2/transconductance_0
* x18 posvartrans_1/transconductance_1
* x19 posvartrans_1/transconductance_0
* x20 posvartrans_6/transconductance_1
* x21 posvartrans_6/transconductance_0
* x22 posvartrans_17/transconductance_1
* x23 posvartrans_17/transconductance_0
* x24 posvartrans_16/transconductance_1
* x25 posvartrans_16/transconductance_0
* x26 posvartrans_15/transconductance_1
* x27 posvartrans_15/transconductance_0
* x28 posvartrans_14/transconductance_1
* x29 posvartrans_14/transconductance_0
* x30 posvartrans_13/transconductance_1
* x31 posvartrans_13/transconductance_0
* x32 posvartrans_0/transconductance_1
* x33 posvartrans_0/transconductance_0
* x34 posvartrans_12/transconductance_1
* x35 posvartrans_12/transconductance_0

*******************
Vdd VDD 0 5
Vss VSS 0 -5

Vstart start 0 PWL(0 0 100e-9 0 101e-9 5 400e-9 5 401e-9 0 4000e-9 0)
Vselect select 0 PWL(0 0 400e-9 0 401e-9 5 400e6 5)

Cx11 VX11INITIAL 0 4pF
Cx12 VX12INITIAL 0 4pF
Cx13 VX13INITIAL 0 4pF

****** INPUTS *---- Initializing the State "VXINITIAL" and Input
Voltage "U"
Vuw11 VUW11 0 -1
Vx11initial in i11 0 DC 0 * make VXIN =0 in SHIFTWEST
Vuc11 VUC11 0 DC 1

Vx12initial in i12 0 DC 0
Vuc12 VUC12 0 DC 1

Vx13initial in i13 0 DC 0
Vuc13 VUC13 0 DC -1

******************* Setting the transconductance values

Vuc11g1 VUC11G1 0 0 *guc=0 for shift west cell
Vuc11g4 VUC11G4 0 0
Vuc11gr1 VUC11GR1 0 0
Vuc11gr4 VUC11GR4 0 0

Vyc11g1 VYC11G1 0 5 *gyc=1
Vyc11g4 VYC11G4 0 -5
Vyc11gr1 VYC11GR1 0 5
Vyc11gr4 VYC11GR4 0 -5
Vuw11g1 VUW11G1 0 5 *guw=1 for shift west cell
Vuw11g4 VUW11G4 0 -5
Vuw11gr1 VUW11GR1 0 5
Vuw11gr4 VUW11GR4 0 -5

****************
Vuc12g1 VUC12G1 0 0
Vuc12g4 VUC12G4 0 0
Vuc12gr1 VUC12GR1 0 0
Vuc12gr4 VUC12GR4 0 0

Vyc12g1 VYC12G1 0 5
Vyc12g4 VYC12G4 0 -5
Vyc12gr1 VYC12GR1 0 5
Vyc12gr4 VYC12GR4 0 -5

Vuw12g1 VUW12G1 0 5
Vuw12g4 VUW12G4 0 -5
Vuw12gr1 VUW12GR1 0 5
Vuw12gr4 VUW12GR4 0 -5

****************
Vuc13g1 VUC13G1 0 0
Vuc13g4 VUC13G4 0 0
Vuc13gr1 VUC13GR1 0 0
Vuc13gr4 VUC13GR4 0 0

Vyc13g1 VYC13G1 0 5
Vyc13g4 VYC13G4 0 -5
Vyc13gr1 VYC13GR1 0 5
Vyc13gr4 VYC13GR4 0 -5

Vuw13g1 VUW13G1 0 5
Vuw13g4 VUW13G4 0 -5
Vuw13gr1 VUW13GR1 0 5
Vuw13gr4 VUW13GR4 0 -5

****************
Vx11g1 VX11G1 0 4
Vx11g4 VX11G4 0 -4
Vx11gr1 VX11GR1 0 5
Vx11gr4 VX11GR4 0 -5

Vyx11g1 VYX11G1 0 5
Vyx11g4 VYX11G4 0 -5
Vyx11gr1 VYX11GR1 0 5
Vyx11gr4 VYX11GR4 0 -5

Vy11g1 VY11G1 0 5
Vy11g4 VY11G4 0 -5
Vy11gr1 VY11GR1 0 5
Vy11gr4 VY11GR4 0 -5

****
Vx12g1 VX12G1 0 4
Vx12g4 VX12G4 0 -4
Vx12gr1 VX12GR1 0 5
Vx12gr4 VX12GR4 0 -5

Vy12g1 VY12G1 0 5
Vy12g4 VY12G4 0 -5
Vy12gr1 VY12GR1 0 5
Vy12gr4 VY12GR4 0 -5

Vy13g1 VY13G1 0 5
Vy13g4 VY13G4 0 -5
Vy13gr1 VY13GR1 0 5
Vy13gr4 VY13GR4 0 -5

***************
.options post
.TRAN 1ns 4us
.end

B.12 DILATION WEST 1×3 CNN

* HSPICE file created from DILATIONWEST_3CELLS.ext - technology: scmos

.option scale=0.3u
.include model.txt
m1000 x0/Va VZ11GR1 VDD Gnd nfet w=33 l=34
+ ad=107 pd=96 as=7098 ps=3948
m1001 x0/Vb VZ11GR4 VSS Vdd pfet w=100 l=34
+ ad=404 pd=210 as=21000 ps=8820
m1002 x1/VIN x1/VIN x0/Va Vdd pfet w=100 l=34
+ ad=3980 pd=2100 as=404 ps=210
m1003 x1/VIN x1/VIN x0/Vb Gnd nfet w=33 l=34
+ ad=1177 pd=1056 as=107 ps=90
m1004 x2/Va VZ11G1 VDD Gnd nfet w=33 l=34
+ ad=107 pd=96 as=0 ps=0
<table>
<thead>
<tr>
<th>Pin</th>
<th>Source 1</th>
<th>Source 2</th>
<th>Source 3</th>
<th>Source 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>m1034</td>
<td>x1/VIN</td>
<td>x1/VIN</td>
<td>x9/Va</td>
<td>Vdd</td>
</tr>
<tr>
<td></td>
<td>w=100</td>
<td>l=34</td>
<td></td>
<td></td>
</tr>
<tr>
<td>m1035</td>
<td>x1/VIN</td>
<td>x1/VIN</td>
<td>x9/Vb</td>
<td>Gnd</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>w=33</td>
<td>l=34</td>
</tr>
<tr>
<td>m1036</td>
<td>x10/Va</td>
<td>VUW13G1</td>
<td>VDD</td>
<td>Gnd</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>w=33</td>
<td>l=34</td>
</tr>
<tr>
<td>m1037</td>
<td>x10/Vb</td>
<td>VUW13G4</td>
<td>VSS</td>
<td>pfet</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>w=100</td>
<td>l=34</td>
</tr>
<tr>
<td>m1038</td>
<td>x1/VIN</td>
<td>VUW13</td>
<td>x10/Va</td>
<td>Gnd</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>w=33</td>
</tr>
<tr>
<td>m1039</td>
<td>x1/VIN</td>
<td>VUW13</td>
<td>x10/Vb</td>
<td>Gnd</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>w=33</td>
</tr>
<tr>
<td>m1040</td>
<td>x11/Va</td>
<td>VZ11G1</td>
<td>VDD</td>
<td>Gnd</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>w=33</td>
<td>l=34</td>
</tr>
<tr>
<td>m1041</td>
<td>x11/Vb</td>
<td>VZ11G4</td>
<td>VSS</td>
<td>Vdd</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>w=100</td>
<td>l=34</td>
</tr>
<tr>
<td>m1042</td>
<td>x12/VIN</td>
<td>x12/VIN</td>
<td>x11/Va</td>
<td>Vdd</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>pfet</td>
</tr>
<tr>
<td>m1043</td>
<td>x12/VIN</td>
<td>x12/VIN</td>
<td>x11/Vb</td>
<td>Gnd</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>w=33</td>
</tr>
<tr>
<td>m1044</td>
<td>x13/Va</td>
<td>VZ11G1</td>
<td>VDD</td>
<td>Gnd</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>w=33</td>
<td>l=34</td>
</tr>
<tr>
<td>m1045</td>
<td>x13/Vb</td>
<td>VZ11G4</td>
<td>VSS</td>
<td>Vdd</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>w=100</td>
<td>l=34</td>
</tr>
<tr>
<td>m1046</td>
<td>x12/VIN</td>
<td>VZ12</td>
<td>x13/Va</td>
<td>Gnd</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>w=33</td>
</tr>
<tr>
<td>m1047</td>
<td>x12/VIN</td>
<td>VZ12</td>
<td>x13/Vb</td>
<td>Gnd</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>w=33</td>
</tr>
<tr>
<td>m1048</td>
<td>x14/Va</td>
<td>VZ11GR1</td>
<td>VDD</td>
<td>Gnd</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>w=33</td>
<td>l=34</td>
</tr>
<tr>
<td>m1049</td>
<td>x14/Vb</td>
<td>VZ11GR4</td>
<td>VSS</td>
<td>Vdd</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>w=100</td>
<td>l=34</td>
</tr>
<tr>
<td>m1050</td>
<td>x15/VIN</td>
<td>x15/VIN</td>
<td>x14/Va</td>
<td>Gnd</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>w=33</td>
</tr>
<tr>
<td>m1051</td>
<td>x15/VIN</td>
<td>x15/VIN</td>
<td>x14/Vb</td>
<td>Gnd</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>w=33</td>
</tr>
<tr>
<td>m1052</td>
<td>x16/Va</td>
<td>VZ11G1</td>
<td>VDD</td>
<td>Gnd</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>w=33</td>
<td>l=34</td>
</tr>
<tr>
<td>m1053</td>
<td>x16/Vb</td>
<td>VZ11G4</td>
<td>VSS</td>
<td>Vdd</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>w=100</td>
<td>l=34</td>
</tr>
<tr>
<td>m1054</td>
<td>x15/VIN</td>
<td>VZ11</td>
<td>x16/Va</td>
<td>Gnd</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>w=33</td>
</tr>
<tr>
<td>m1055</td>
<td>x15/VIN</td>
<td>VZ11</td>
<td>x16/Vb</td>
<td>Gnd</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>w=33</td>
</tr>
<tr>
<td>m1056</td>
<td>x17/Va</td>
<td>VYC13GR1</td>
<td>VDD</td>
<td>Gnd</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>w=33</td>
<td>l=34</td>
</tr>
<tr>
<td>m1057</td>
<td>x17/Vb</td>
<td>VYC13GR4</td>
<td>VSS</td>
<td>Vdd</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>w=100</td>
<td>l=34</td>
</tr>
<tr>
<td>m1058</td>
<td>x1/VIN</td>
<td>x1/VIN</td>
<td>x17/Va</td>
<td>Gnd</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>w=33</td>
</tr>
<tr>
<td>m1059</td>
<td>x1/VIN</td>
<td>x1/VIN</td>
<td>x17/Vb</td>
<td>Gnd</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>w=33</td>
</tr>
<tr>
<td>m1060</td>
<td>x18/Va</td>
<td>VYC13G1</td>
<td>VDD</td>
<td>Gnd</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>w=33</td>
<td>l=34</td>
</tr>
<tr>
<td>m1061</td>
<td>x18/Vb</td>
<td>VYC13G4</td>
<td>VSS</td>
<td>Vdd</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>w=100</td>
<td>l=34</td>
</tr>
</tbody>
</table>
m1062 x1/VIN VY13OUT x18/Va Vdd pfet w=100 l=34
+ ad=0 pd=0 as=404 ps=210
m1063 x1/VIN VY13OUT x18/Vb Gnd nfet w=33 l=34
+ ad=0 pd=0 as=107 ps=90
m1064 x19/Va VY12GR1 VDD Gnd nfet w=33 l=34
+ ad=107 pd=96 as=0 ps=0
m1065 x19/Vb VY12GR4 VSS Vdd pfet w=100 l=34
+ ad=404 pd=210 as=0 ps=0
m1066 VY12OUT VY12OUT x19/Va Vdd pfet w=100 l=34
+ ad=1616 pd=840 as=404 ps=210
m1067 VY12OUT VY12OUT x19/Vb Gnd nfet w=33 l=34
+ ad=428 pd=384 as=107 ps=90
m1068 x20/Va VY12G1 VDD Gnd nfet w=33 l=34
+ ad=107 pd=96 as=0 ps=0
m1069 x20/Vb VY12G4 VSS Vdd pfet w=100 l=34
+ ad=404 pd=210 as=0 ps=0
m1070 VY12OUT VY12OUT x20/Va Vdd pfet w=100 l=34
+ ad=0 pd=0 as=404 ps=210
m1071 VY12OUT VY12OUT x20/Vb Gnd nfet w=33 l=34
+ ad=0 pd=0 as=107 ps=90
m1072 x21/Va VYX12GR1 VDD Gnd nfet w=33 l=34
+ ad=107 pd=96 as=0 ps=0
m1073 x21/Vb VYX12GR4 VSS Vdd pfet w=100 l=34
+ ad=404 pd=210 as=0 ps=0
m1074 VY12OUT VY12OUT x21/Va Vdd pfet w=100 l=34
+ ad=0 pd=0 as=404 ps=210
m1075 VY12OUT VY12OUT x21/Vb Gnd nfet w=33 l=34
+ ad=0 pd=0 as=107 ps=90
m1076 x22/Va VYX12G1 VDD Gnd nfet w=33 l=34
+ ad=107 pd=96 as=0 ps=0
m1077 x22/Vb VYX12G4 VSS Vdd pfet w=100 l=34
+ ad=404 pd=210 as=0 ps=0
m1078 VY12OUT VX12INITIAL x22/Va Vdd pfet w=100 l=34
+ ad=0 pd=0 as=404 ps=210
m1079 VY12OUT VX12INITIAL x22/Vb Gnd nfet w=33 l=34
+ ad=0 pd=0 as=107 ps=90
m1080 x23/Va VX12GR1 VDD Gnd nfet w=33 l=34
+ ad=107 pd=96 as=0 ps=0
m1081 x23/Vb VX12GR4 VSS Vdd pfet w=100 l=34
+ ad=404 pd=210 as=0 ps=0
m1082 x12/VIN x12/VIN x23/Va Vdd pfet w=100 l=34
+ ad=0 pd=0 as=404 ps=210
m1083 x12/VIN x12/VIN x23/Vb Gnd nfet w=33 l=34
+ ad=0 pd=0 as=107 ps=90
m1084 x24/Va VX12G1 VDD Gnd nfet w=33 l=34
+ ad=107 pd=96 as=0 ps=0
m1085 x24/Vb VX12G4 VSS Vdd pfet w=100 l=34
+ ad=404 pd=210 as=0 ps=0
m1086 x12/VIN x12/VIN x24/Va Vdd pfet w=100 l=34
+ ad=0 pd=0 as=404 ps=210
m1087 x12/VIN x12/VIN x24/Vb Gnd nfet w=33 l=34
+ ad=0 pd=0 as=107 ps=90
m1088 x25/Va VUW12GR1 VDD Gnd nfet w=33 l=34
+ ad=107 pd=96 as=0 ps=0
m1089 x25/Vb VUW12GR4 VSS Vdd pfet w=100 l=34
+ ad=404 pd=210 as=0 ps=0
m1090 x12/VIN x12/VIN x25/Va Vdd pfet w=100 l=34
<table>
<thead>
<tr>
<th>ID</th>
<th>Xref</th>
<th>Vref</th>
<th>Vout</th>
<th>Vdd</th>
<th>Vss</th>
<th>W</th>
<th>L</th>
</tr>
</thead>
<tbody>
<tr>
<td>m1148</td>
<td>x40/Va</td>
<td>VYC11G1</td>
<td>VDD</td>
<td>Gnd</td>
<td>nfet</td>
<td>w=33</td>
<td>l=34</td>
</tr>
<tr>
<td>m1149</td>
<td>x40/Vb</td>
<td>VYC11G4</td>
<td>VSS</td>
<td>Vdd</td>
<td>pfet</td>
<td>w=100</td>
<td>l=34</td>
</tr>
<tr>
<td>m1150</td>
<td>x15/VIN</td>
<td>x40/Va</td>
<td>Vdd</td>
<td>pfet</td>
<td>w=100</td>
<td>l=34</td>
<td></td>
</tr>
<tr>
<td>m1151</td>
<td>x15/VIN</td>
<td>x40/Vb</td>
<td>Gnd</td>
<td>nfet</td>
<td>w=33</td>
<td>l=34</td>
<td></td>
</tr>
<tr>
<td>m1152</td>
<td>x41/Va</td>
<td>VUC12GR1</td>
<td>VDD</td>
<td>Gnd</td>
<td>nfet</td>
<td>w=33</td>
<td>l=34</td>
</tr>
<tr>
<td>m1153</td>
<td>x41/Vb</td>
<td>VUC12GR4</td>
<td>VSS</td>
<td>Vdd</td>
<td>pfet</td>
<td>w=100</td>
<td>l=34</td>
</tr>
<tr>
<td>m1154</td>
<td>x12/VIN</td>
<td>x12/VIN</td>
<td>x41/Va</td>
<td>Vdd</td>
<td>pfet</td>
<td>w=100</td>
<td>l=34</td>
</tr>
<tr>
<td>m1155</td>
<td>x12/VIN</td>
<td>x12/VIN</td>
<td>x41/Vb</td>
<td>Gnd</td>
<td>nfet</td>
<td>w=33</td>
<td>l=34</td>
</tr>
<tr>
<td>m1156</td>
<td>x42/Va</td>
<td>VUC12G1</td>
<td>VDD</td>
<td>Gnd</td>
<td>nfet</td>
<td>w=33</td>
<td>l=34</td>
</tr>
<tr>
<td>m1157</td>
<td>x42/Vb</td>
<td>VUC12G4</td>
<td>VSS</td>
<td>Vdd</td>
<td>pfet</td>
<td>w=100</td>
<td>l=34</td>
</tr>
<tr>
<td>m1158</td>
<td>x12/VIN</td>
<td>x12/VIN</td>
<td>x42/Va</td>
<td>Vdd</td>
<td>pfet</td>
<td>w=100</td>
<td>l=34</td>
</tr>
<tr>
<td>m1159</td>
<td>x12/VIN</td>
<td>x12/VIN</td>
<td>x42/Vb</td>
<td>Gnd</td>
<td>nfet</td>
<td>w=33</td>
<td>l=34</td>
</tr>
<tr>
<td>m1160</td>
<td>x43/Va</td>
<td>VUC11GR1</td>
<td>VDD</td>
<td>Gnd</td>
<td>nfet</td>
<td>w=33</td>
<td>l=34</td>
</tr>
<tr>
<td>m1161</td>
<td>x43/Vb</td>
<td>VUC11GR4</td>
<td>VSS</td>
<td>Vdd</td>
<td>pfet</td>
<td>w=100</td>
<td>l=34</td>
</tr>
<tr>
<td>m1162</td>
<td>x15/VIN</td>
<td>x15/VIN</td>
<td>x43/Va</td>
<td>Vdd</td>
<td>pfet</td>
<td>w=100</td>
<td>l=34</td>
</tr>
<tr>
<td>m1163</td>
<td>x15/VIN</td>
<td>x15/VIN</td>
<td>x43/Vb</td>
<td>Gnd</td>
<td>nfet</td>
<td>w=33</td>
<td>l=34</td>
</tr>
<tr>
<td>m1164</td>
<td>x15/VIN</td>
<td>x15/VIN</td>
<td>x44/Va</td>
<td>Vdd</td>
<td>pfet</td>
<td>w=100</td>
<td>l=34</td>
</tr>
<tr>
<td>m1165</td>
<td>x15/VIN</td>
<td>x15/VIN</td>
<td>x44/Vb</td>
<td>Gnd</td>
<td>nfet</td>
<td>w=33</td>
<td>l=34</td>
</tr>
<tr>
<td>m1166</td>
<td>VX11INITIAL</td>
<td>select</td>
<td>x15/VIN</td>
<td>Gnd</td>
<td>nfet</td>
<td>w=33</td>
<td>l=34</td>
</tr>
<tr>
<td>m1167</td>
<td>VX11INITIAL</td>
<td>select</td>
<td>x12/VIN</td>
<td>Gnd</td>
<td>nfet</td>
<td>w=33</td>
<td>l=34</td>
</tr>
<tr>
<td>m1168</td>
<td>VX11INITIAL</td>
<td>select</td>
<td>x1/VIN</td>
<td>Gnd</td>
<td>nfet</td>
<td>w=33</td>
<td>l=34</td>
</tr>
<tr>
<td>m1169</td>
<td>VX11INITIAL</td>
<td>select</td>
<td>x1/VIN</td>
<td>Gnd</td>
<td>nfet</td>
<td>w=33</td>
<td>l=34</td>
</tr>
<tr>
<td>m1170</td>
<td>VX12INITIAL</td>
<td>select</td>
<td>x12/VIN</td>
<td>Gnd</td>
<td>nfet</td>
<td>w=33</td>
<td>l=34</td>
</tr>
<tr>
<td>m1171</td>
<td>VX12INITIAL</td>
<td>select</td>
<td>x1/VIN</td>
<td>Gnd</td>
<td>nfet</td>
<td>w=33</td>
<td>l=34</td>
</tr>
<tr>
<td>m1172</td>
<td>VX13INITIAL</td>
<td>select</td>
<td>x1/VIN</td>
<td>Gnd</td>
<td>nfet</td>
<td>w=33</td>
<td>l=34</td>
</tr>
<tr>
<td>m1173</td>
<td>VX13INITIAL</td>
<td>select</td>
<td>x1/VIN</td>
<td>Gnd</td>
<td>nfet</td>
<td>w=33</td>
<td>l=34</td>
</tr>
</tbody>
</table>

Capacitors:
- C0: VZ11GR4 x15/VIN 2.3fF
- C1: x12/VIN VSS 2.0fF
- C2: VSS x1/VIN 2.0fF
- C3: V211GR4 x12/VIN 2.3fF
C4 VZ11GR4 x1/VIN 2.3fF
C5 VSS x15/VIN 2.0fF
C6 start GND 4.7fF
C7 select GND 4.8fF
C8 VY12OUT GND 28.3fF
C9 VY11OUT GND 31.7fF
C10 VUC11 GND 34.2fF
C11 VUC12 GND 14.3fF
C12 VUW11G4 GND 2.1fF
C13 VUW11GR4 GND 2.1fF
C14 VX11INITIAL GND 8.7fF
C15 VUC13 GND 12.2fF
C16 VUW12G4 GND 2.1fF
C17 VUW12GR4 GND 2.1fF
C18 VX12INITIAL GND 8.4fF
C19 VZ11 GND 6.4fF
C20 VZ12 GND 6.4fF
C21 VUW13G4 GND 2.1fF
C22 VUW13 GND 37.8fF
C23 VUW13GR4 GND 2.1fF
C24 VX13INITIAL GND 8.4fF
C25 VYX13GR4 GND 2.2fF
C26 VDD GND 270.0fF
C27 VZ11G4 GND 5.4fF
C28 VZ13 GND 6.4fF
C29 VZ11GR4 GND 5.4fF

** hspice subcircuit dictionary
* x0 posvartrans_20/transconductance_1
* x1 posvartrans_9
* x2 posvartrans_20/transconductance_0
* x3 posvartrans_11/transconductance_1
* x4 posvartrans_11/transconductance_0
* x5 posvartrans_10/transconductance_1
* x6 posvartrans_10/transconductance_0
* x7 posvartrans_9/transconductance_1
* x8 posvartrans_9/transconductance_0
* x9 posvartrans_8/transconductance_1
* x10 posvartrans_8/transconductance_0
* x11 posvartrans_19/transconductance_1
* x12 posvartrans_3
* x13 posvartrans_19/transconductance_0
* x14 posvartrans_18/transconductance_1
* x15 posvartrans_15
* x16 posvartrans_18/transconductance_0
* x17 posvartrans_7/transconductance_1
* x18 posvartrans_7/transconductance_0
* x19 posvartrans_5/transconductance_1
* x20 posvartrans_5/transconductance_0
* x21 posvartrans_4/transconductance_1
* x22 posvartrans_4/transconductance_0
* x23 posvartrans_3/transconductance_1
* x24 posvartrans_3/transconductance_0
* x25 posvartrans_2/transconductance_1
* x26 posvartrans_2/transconductance_0
* x27 posvartrans_1/transconductance_1
* x28 posvartrans_1/transconductance_0
* x29 posvartrans_6/transconductance_1
* x30 posvartrans_6/transconductance_0
* x31 posvartrans_17/transconductance_1
* x32 posvartrans_17/transconductance_0
* x33 posvartrans_16/transconductance_1
* x34 posvartrans_16/transconductance_0
* x35 posvartrans_15/transconductance_1
* x36 posvartrans_15/transconductance_0
* x37 posvartrans_14/transconductance_1
* x38 posvartrans_14/transconductance_0
* x39 posvartrans_13/transconductance_1
* x40 posvartrans_13/transconductance_0
* x41 posvartrans_0/transconductance_1
* x42 posvartrans_0/transconductance_0
* x43 posvartrans_12/transconductance_1
* x44 posvartrans_12/transconductance_0

****************************************************************************** Initializing
Vdd VDD 0 5
Vss VSS 0 -5

Vstart start 0 PWL(0 0 100e-9 0 101e-9 5 400e-9 5 401e-9 0 4000e-9 0)
Vselect select 0 PWL(0 0 400e-9 0 401e-9 5 400e-6 5)

Cx11 VX11INITIAL 0 4pF
Cx12 VX12INITIAL 0 4pF
Cx13 VX13INITIAL 0 4pF

Vx11initial in_i11 0 DC 0 * VX=0
Vx12initial in_i12 0 DC 0
Vx13initial in_i13 0 DC 0

Vz11 VZ11 0 DC 1 * Fixed Bias
Vz12 VZ12 0 DC 1
Vz13 VZ13 0 DC 1

***** Input Pattern
Vuc11 VUC11 0 DC -1
Vuc12 VUC12 0 DC 1
Vuc13 VUC13 0 DC -1
Vuw13 VUW13 0 DC 1

****************************************************************************** Setting the transconductance values

****************************************************************************** gYC (2)
Vyc11g1 VYC11G1 0 5
Vyc11g4 VYC11G4 0 -5
Vyc11gr1 VYC11GR1 0 3.67
Vyc11gr4 VYC11GR4 0 -3.67

Vyc12g1 VYC12G1 0 5
Vyc12g4 VYC12G4 0 -5
Vyc12gr1 VYC12GR1 0 3.67
Vyc12gr4 VYC12GR4 0 -3.67
Vyc13g1 VYC13G1 0 5  
Vyc13g4 VYC13G4 0 -5  
Vyc13gr1 VYC13GR1 0 3.67  
Vyc13gr4 VYC13GR4 0 -3.67  

**********************************   gUC=1,gUW (1)  
Vuc11g1 VUC11G1 0 5  
Vuc11g4 VUC11G4 0 -5  
Vuc11gr1 VUC11GR1 0 5  
Vuc11gr4 VUC11GR4 0 -5  

Vuw11g1 VUW11G1 0 5  
Vuw11g4 VUW11G4 0 -5  
Vuw11gr1 VUW11GR1 0 5  
Vuw11gr4 VUW11GR4 0 -5  

******  
Vuc12g1 VUC12G1 0 5  
Vuc12g4 VUC12G4 0 -5  
Vuc12gr1 VUC12GR1 0 -5  
Vuc12gr4 VUC12GR4 0 5  

Vuw12g1 VUW12G1 0 5  
Vuw12g4 VUW12G4 0 -5  
Vuw12gr1 VUW12GR1 0 5  
Vuw12gr4 VUW12GR4 0 -5  

******  
Vuc13g1 VUC13G1 0 5  
Vuc13g4 VUC13G4 0 -5  
Vuc13gr1 VUC13GR1 0 5  
Vuc13gr4 VUC13GR4 0 -5  

Vuw13g1 VUW13G1 0 5  
Vuw13g4 VUW13G4 0 -5  
Vuw13gr1 VUW13GR1 0 5  
Vuw13gr4 VUW13GR4 0 -5  

********************************   gZ (1.5)  
Vz11g1 VZ11G1 0 5  
Vz11g4 VZ11G4 0 -5  
Vz11gr1 VZ11GR1 0 4  
Vz11gr4 VZ11GR4 0 -4  

******************************** gX(<1.5 = 0.5), gYX(1), gY(1)  
Vx11g1 VX11G1 0 4  
Vx11g4 VX11G4 0 -4  
Vx11gr1 VX11GR1 0 5  
Vx11gr4 VX11GR4 0 -5  

Vy11g1 VY11G1 0 5  

Vy11g4 VY11G4 0 -5  
Vy11gr1 VY11GR1 0 5  
Vy11gr4 VY11GR4 0 -5  

Vy11g1 VY11G1 0 5  

184
**B.13 FILBLACK single cell CNN**

* HSPICE file created from FILBLACK_1CELL.ext - technology: scmos

```
.options post
.TRAN 1ns 4us
.end
```

```
Vy11g4 VY11G4 0 -5
Vy11gr1 VY11GR1 0 5
Vy11gr4 VY11GR4 0 -5

************
Vx12g1 VX12G1 0 4
Vx12g4 VX12G4 0 -4
Vx12gr1 VX12GR1 0 5
Vx12gr4 VX12GR4 0 -5

Vy12g1 VY12G1 0 5
Vy12g4 VY12G4 0 -5
Vy12gr1 VY12GR1 0 5
Vy12gr4 VY12GR4 0 -5

Vy12g1 VY12G1 0 5
Vy12g4 VY12G4 0 -5
Vy12gr1 VY12GR1 0 5
Vy12gr4 VY12GR4 0 -5

Vy13g1 VY13G1 0 5
Vy13g4 VY13G4 0 -5
Vy13gr1 VY13GR1 0 5
Vy13gr4 VY13GR4 0 -5

Vy13g1 VY13G1 0 5
Vy13g4 VY13G4 0 -5
Vy13gr1 VY13GR1 0 5
Vy13gr4 VY13GR4 0 -5

Vy13g1 VY13G1 0 5
Vy13g4 VY13G4 0 -5
Vy13gr1 VY13GR1 0 5
Vy13gr4 VY13GR4 0 -5

.options post
.TRAN 1ns 4us
.end
```
<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
<th>Dimensions</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>m1003</td>
<td>VY11OUT VY11OUT x0/Vb Gnd nfet w=33 l=34</td>
<td>+ ad=428 pd=384 as=107 ps=90</td>
<td></td>
</tr>
<tr>
<td>m1004</td>
<td>x1/Va VY11G1 VDD Gnd nfet w=33 l=34</td>
<td>+ ad=107 pd=96 as=0 ps=0</td>
<td></td>
</tr>
<tr>
<td>m1005</td>
<td>x1/Vb VY11G4 VSS Vdd pfet w=100 l=34</td>
<td>+ ad=404 pd=210 as=0 ps=0</td>
<td></td>
</tr>
<tr>
<td>m1006</td>
<td>VY11OUT VY11OUT x1/Va Vdd pfet w=100 l=34</td>
<td>+ ad=0 pd=0 as=404 ps=210</td>
<td></td>
</tr>
<tr>
<td>m1007</td>
<td>VY11OUT VY11OUT x1/Vb Gnd nfet w=33 l=34</td>
<td>+ ad=0 pd=0 as=107 ps=90</td>
<td></td>
</tr>
<tr>
<td>m1008</td>
<td>x2/Va VYX11GR1 VDD Gnd nfet w=33 l=34</td>
<td>+ ad=107 pd=96 as=0 ps=0</td>
<td></td>
</tr>
<tr>
<td>m1009</td>
<td>x2/Vb VYX11GR4 VSS Vdd pfet w=100 l=34</td>
<td>+ ad=404 pd=210 as=0 ps=0</td>
<td></td>
</tr>
<tr>
<td>m1010</td>
<td>VY11OUT VY11OUT x2/Va Vdd pfet w=100 l=34</td>
<td>+ ad=0 pd=0 as=404 ps=210</td>
<td></td>
</tr>
<tr>
<td>m1011</td>
<td>VY11OUT VY11OUT x2/Vb Gnd nfet w=33 l=34</td>
<td>+ ad=0 pd=0 as=107 ps=90</td>
<td></td>
</tr>
<tr>
<td>m1012</td>
<td>x3/Va VYX11G1 VDD Gnd nfet w=33 l=34</td>
<td>+ ad=107 pd=96 as=0 ps=0</td>
<td></td>
</tr>
<tr>
<td>m1013</td>
<td>x3/Vb VYX11G4 VSS Vdd pfet w=100 l=34</td>
<td>+ ad=404 pd=210 as=0 ps=0</td>
<td></td>
</tr>
<tr>
<td>m1014</td>
<td>VY11OUT VX11INITIAL x3/Va Vdd pfet w=100 l=34</td>
<td>+ ad=0 pd=0 as=404 ps=210</td>
<td></td>
</tr>
<tr>
<td>m1015</td>
<td>VY11OUT VX11INITIAL x3/Vb Gnd nfet w=33 l=34</td>
<td>+ ad=0 pd=0 as=107 ps=90</td>
<td></td>
</tr>
<tr>
<td>m1016</td>
<td>x4/Va VYX11GR1 VDD Gnd nfet w=33 l=34</td>
<td>+ ad=107 pd=96 as=0 ps=0</td>
<td></td>
</tr>
<tr>
<td>m1017</td>
<td>x4/Vb VYX11GR4 VSS Vdd pfet w=100 l=34</td>
<td>+ ad=404 pd=210 as=0 ps=0</td>
<td></td>
</tr>
<tr>
<td>m1018</td>
<td>VSUM11 VSUM11 x4/Va Vdd pfet w=100 l=34</td>
<td>+ ad=2388 pd=1260 as=404 ps=210</td>
<td></td>
</tr>
<tr>
<td>m1019</td>
<td>VSUM11 VSUM11 x4/Vb Gnd nfet w=33 l=34</td>
<td>+ ad=749 pd=672 as=107 ps=90</td>
<td></td>
</tr>
<tr>
<td>m1020</td>
<td>x5/Va VYX11G1 VDD Gnd nfet w=33 l=34</td>
<td>+ ad=107 pd=96 as=0 ps=0</td>
<td></td>
</tr>
<tr>
<td>m1021</td>
<td>x5/Vb VYX11G4 VSS Vdd pfet w=100 l=34</td>
<td>+ ad=404 pd=210 as=0 ps=0</td>
<td></td>
</tr>
<tr>
<td>m1022</td>
<td>VSUM11 VSUM11 x5/Va Vdd pfet w=100 l=34</td>
<td>+ ad=0 pd=0 as=404 ps=210</td>
<td></td>
</tr>
<tr>
<td>m1023</td>
<td>VSUM11 VSUM11 x5/Vb Gnd nfet w=33 l=34</td>
<td>+ ad=0 pd=0 as=107 ps=90</td>
<td></td>
</tr>
<tr>
<td>m1024</td>
<td>x6/Va VZ11GR1 VDD Gnd nfet w=33 l=34</td>
<td>+ ad=107 pd=96 as=0 ps=0</td>
<td></td>
</tr>
<tr>
<td>m1025</td>
<td>x6/Vb VZ11GR4 VSS Vdd pfet w=100 l=34</td>
<td>+ ad=404 pd=210 as=0 ps=0</td>
<td></td>
</tr>
<tr>
<td>m1026</td>
<td>VSUM11 VSUM11 x6/Va Vdd pfet w=100 l=34</td>
<td>+ ad=0 pd=0 as=404 ps=210</td>
<td></td>
</tr>
<tr>
<td>m1027</td>
<td>VSUM11 VSUM11 x6/Vb Gnd nfet w=33 l=34</td>
<td>+ ad=0 pd=0 as=107 ps=90</td>
<td></td>
</tr>
<tr>
<td>m1028</td>
<td>x7/Va VZ11G1 VDD Gnd nfet w=33 l=34</td>
<td>+ ad=107 pd=96 as=0 ps=0</td>
<td></td>
</tr>
<tr>
<td>m1029</td>
<td>x7/Vb VZ11G4 VSS Vdd pfet w=100 l=34</td>
<td>+ ad=404 pd=210 as=0 ps=0</td>
<td></td>
</tr>
<tr>
<td>m1030</td>
<td>VSUM11 VS11 x7/Va Vdd pfet w=100 l=34</td>
<td>+ ad=0 pd=0 as=404 ps=210</td>
<td></td>
</tr>
<tr>
<td>m1031</td>
<td>VSUM11 VS11 x7/Vb Gnd nfet w=33 l=34</td>
<td>+ ad=0 pd=0 as=404 ps=210</td>
<td></td>
</tr>
</tbody>
</table>
** hspice subcircuit dictionary
* x0 posvartrans_17/transconductance_1
* x1 posvartrans_17/transconductance_0
* x2 posvartrans_16/transconductance_1
* x3 posvartrans_16/transconductance_0
* x4 posvartrans_15/transconductance_1
* x5 posvartrans_15/transconductance_0
* x6 posvartrans_14/transconductance_1
* x7 posvartrans_14/transconductance_0
* x8 posvartrans_13/transconductance_1
* x9 posvartrans_13/transconductance_0

***************
Vdd VDD 0 5
Vss VSS 0 -5

Vstart start 0 PWL(0 0 100e-9 0 101e-9 5 400e-9 5 401e-9 0 4000e-9 0)
Vselect select 0 PWL(0 0 400e-9 0 401e-9 5 400e-6 5)
**Inputs**

Initializing the State "VXINITIAL"

Vx11initial in_i11 0 DC 1

Setting the transconductance values

Vyc11g1 VYCl1G1 0 5 *gyc=1
Vyc11g4 VYCl1G4 0 -5
Vyc11gr1 VYCl1GR1 0 3.67
Vyc11gr4 VYCl1GR4 0 -3.67

Vz11g1 VZ11G1 0 5 *gue=1 for shift east cell
Vz11g4 VZ11G4 0 -5
Vz11gr1 VZ11GR1 0 3
Vz11gr4 VZ11GR4 0 -3

Vx11g1 VX11G1 0 4
Vx11g4 VX11G4 0 -4
Vx11gr1 VX11GR1 0 5
Vx11gr4 VX11GR4 0 -5

Vyx11g1 VYX11G1 0 5
Vyx11g4 VYX11G4 0 -5
Vyx11gr1 VYX11GR1 0 5
Vyx11gr4 VYX11GR4 0 -5

Vy11g1 VY11G1 0 5
Vy11g4 VY11G4 0 -5
Vy11gr1 VY11GR1 0 5
Vy11gr4 VY11GR4 0 -5

.options post
.TRAN 1ns 4us
.end

**B.14 FILBLACK 1×3 CNN**

* HSPICE file created from FILBLACK_3CELLS.ext - technology: scmos

.option scale=0.3u
.include model.txt
<table>
<thead>
<tr>
<th>Device ID</th>
<th>Clock</th>
<th>Clock pin</th>
<th>Data pin</th>
<th>Data pin size</th>
<th>Data pin orientation</th>
</tr>
</thead>
<tbody>
<tr>
<td>m1000</td>
<td>x0/Va</td>
<td>Vy13Gr1</td>
<td>Vdd</td>
<td>Gnd</td>
<td>nfet</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>w=33 l=34</td>
</tr>
<tr>
<td></td>
<td></td>
<td>x0/Vb</td>
<td>Vy13Gr4</td>
<td>Vss</td>
<td>pfet</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>w=100 l=34</td>
</tr>
<tr>
<td>m1001</td>
<td>x0/Vb</td>
<td>Vy13Gr4</td>
<td>Vss</td>
<td>Vdd</td>
<td>pfet</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>w=33 l=34</td>
</tr>
<tr>
<td>m1002</td>
<td>Vy13ouT</td>
<td>Vy13ouT</td>
<td>x0/Va</td>
<td>Vdd</td>
<td>pfet</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>w=100 l=34</td>
</tr>
<tr>
<td>m1003</td>
<td>Vy13ouT</td>
<td>Vy13ouT</td>
<td>x0/Vb</td>
<td>Gnd</td>
<td>nfet</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>w=33 l=34</td>
</tr>
<tr>
<td>m1004</td>
<td>x1/Va</td>
<td>Vy13G1</td>
<td>Vdd</td>
<td>Gnd</td>
<td>nfet</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>w=33 l=34</td>
</tr>
<tr>
<td>m1005</td>
<td>x1/Vb</td>
<td>Vy13G4</td>
<td>Vss</td>
<td>Vdd</td>
<td>pfet</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>w=100 l=34</td>
</tr>
<tr>
<td>m1006</td>
<td>Vy13ouT</td>
<td>Vy13ouT</td>
<td>x1/Va</td>
<td>Vdd</td>
<td>pfet</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>w=100 l=34</td>
</tr>
<tr>
<td>m1007</td>
<td>Vy13ouT</td>
<td>Vy13ouT</td>
<td>x1/Vb</td>
<td>Gnd</td>
<td>nfet</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>w=33 l=34</td>
</tr>
<tr>
<td>m1008</td>
<td>x2/Va</td>
<td>Vy1X3Gr1</td>
<td>Vdd</td>
<td>Gnd</td>
<td>nfet</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>w=33 l=34</td>
</tr>
<tr>
<td>m1009</td>
<td>x2/Vb</td>
<td>Vy1X3Gr4</td>
<td>Vss</td>
<td>Vdd</td>
<td>pfet</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>w=100 l=34</td>
</tr>
<tr>
<td>m1010</td>
<td>Vy13ouT</td>
<td>Vy13ouT</td>
<td>x2/Va</td>
<td>Vdd</td>
<td>pfet</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>w=100 l=34</td>
</tr>
<tr>
<td>m1011</td>
<td>Vy13ouT</td>
<td>Vy13ouT</td>
<td>x2/Vb</td>
<td>Gnd</td>
<td>nfet</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>w=33 l=34</td>
</tr>
<tr>
<td>m1012</td>
<td>x3/Va</td>
<td>Vy1X3G1</td>
<td>Vdd</td>
<td>Gnd</td>
<td>nfet</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>w=33 l=34</td>
</tr>
<tr>
<td>m1013</td>
<td>x3/Vb</td>
<td>Vy1X3G4</td>
<td>Vss</td>
<td>Vdd</td>
<td>pfet</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>w=100 l=34</td>
</tr>
<tr>
<td>m1014</td>
<td>Vy13ouT</td>
<td>Vy13Initial</td>
<td>x3/Va</td>
<td>Vdd</td>
<td>pfet</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>w=100 l=34</td>
</tr>
<tr>
<td>m1015</td>
<td>Vy13ouT</td>
<td>Vy13Initial</td>
<td>x3/Vb</td>
<td>Gnd</td>
<td>nfet</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>w=33 l=34</td>
</tr>
<tr>
<td>m1016</td>
<td>x4/Va</td>
<td>Vy1X3Gr1</td>
<td>Vdd</td>
<td>Gnd</td>
<td>nfet</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>w=33 l=34</td>
</tr>
<tr>
<td>m1017</td>
<td>x4/Vb</td>
<td>Vy1X3Gr4</td>
<td>Vss</td>
<td>Vdd</td>
<td>pfet</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>w=100 l=34</td>
</tr>
<tr>
<td>m1018</td>
<td>Vsum13</td>
<td>Vsum13</td>
<td>x4/Va</td>
<td>Vdd</td>
<td>pfet</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>w=100 l=34</td>
</tr>
<tr>
<td>m1019</td>
<td>Vsum13</td>
<td>Vsum13</td>
<td>x4/Vb</td>
<td>Gnd</td>
<td>nfet</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>w=33 l=34</td>
</tr>
<tr>
<td>m1020</td>
<td>x5/Va</td>
<td>Vy13G1</td>
<td>Vdd</td>
<td>Gnd</td>
<td>nfet</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>w=33 l=34</td>
</tr>
<tr>
<td>m1021</td>
<td>x5/Vb</td>
<td>Vy13G4</td>
<td>Vss</td>
<td>Vdd</td>
<td>pfet</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>w=100 l=34</td>
</tr>
<tr>
<td>m1022</td>
<td>Vsum13</td>
<td>Vsum13</td>
<td>x5/Va</td>
<td>Vdd</td>
<td>pfet</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>w=100 l=34</td>
</tr>
<tr>
<td>m1023</td>
<td>Vsum13</td>
<td>Vsum13</td>
<td>x5/Vb</td>
<td>Gnd</td>
<td>nfet</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>w=33 l=34</td>
</tr>
<tr>
<td>m1024</td>
<td>x6/Va</td>
<td>Vy13Gr1</td>
<td>Vdd</td>
<td>Gnd</td>
<td>nfet</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>w=33 l=34</td>
</tr>
<tr>
<td>m1025</td>
<td>x6/Vb</td>
<td>Vy13Gr4</td>
<td>Vss</td>
<td>Vdd</td>
<td>pfet</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>w=100 l=34</td>
</tr>
<tr>
<td>m1026</td>
<td>Vsum13</td>
<td>Vsum13</td>
<td>x6/Va</td>
<td>Vdd</td>
<td>pfet</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>w=100 l=34</td>
</tr>
<tr>
<td>m1027</td>
<td>Vsum13</td>
<td>Vsum13</td>
<td>x6/Vb</td>
<td>Gnd</td>
<td>nfet</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>w=33 l=34</td>
</tr>
<tr>
<td>m1028</td>
<td>x7/Va</td>
<td>Vy13G1</td>
<td>Vdd</td>
<td>Gnd</td>
<td>nfet</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>w=33 l=34</td>
</tr>
</tbody>
</table>
m1114 VSUM11 VSUM11 x28/Va Vdd pfet w=100 l=34
  + ad=0 pd=0 as=404 ps=210
m1115 VSUM11 VSUM11 x28/Vb Gnd nfet w=33 l=34
  + ad=0 pd=0 as=107 ps=90
m1116 x29/Va VYC11G1 VDD Gnd nfet w=33 l=34
  + ad=107 pd=96 as=0 ps=0
m1117 x29/Vb VYC11G4 VSS Vdd pfet w=100 l=34
  + ad=404 pd=210 as=0 ps=0
m1118 VSUM11 VY11OUT x29/Va Vdd pfet w=100 l=34
  + ad=0 pd=0 as=404 ps=210
m1119 VSUM11 VY11OUT x29/Vb Gnd nfet w=33 l=34
  + ad=0 pd=0 as=107 ps=90
m1120 VX11INITIAL select VSUM11 Gnd nfet w=33 l=34
  + ad=218 pd=180 as=0 ps=0
m1121 in_i11 start VX11INITIAL Gnd nfet w=33 l=34
  + ad=136 pd=100 as=0 ps=0
m1122 VX12INITIAL select VSUM12 Gnd nfet w=33 l=34
  + ad=218 pd=180 as=0 ps=0
m1123 in_i12 start VX12INITIAL Gnd nfet w=33 l=34
  + ad=136 pd=100 as=0 ps=0
m1124 VX13INITIAL select VSUM13 Gnd nfet w=33 l=34
  + ad=218 pd=180 as=0 ps=0
m1125 in_i13 start VX13INITIAL Gnd nfet w=33 l=34
  + ad=136 pd=100 as=0 ps=0
C0 start GND 4.7fF
C1 select GND 4.8fF
C2 VZ11G4 GND 2.0fF
C3 VZ11 GND 6.4fF
C4 VZ11GR4 GND 2.1fF
C5 VSUM11 GND 32.0fF
C6 VX11INITIAL GND 6.4fF
C7 VY11G4 GND 2.1fF
C8 VY11OUT GND 14.8fF
C9 VY11GR4 GND 2.0fF
C10 V212 GND 6.4fF
C11 VSUM12 GND 20.8fF
C12 VX12INITIAL GND 6.8fF
C13 VYX12GR4 GND 2.2fF
C14 VY12OUT GND 15.8fF
C15 V213 GND 6.4fF
C16 VSUM13 GND 26.7fF
C17 VYX13G4 GND 2.2fF
C18 VX13INITIAL GND 6.6fF
C19 VY13OUT GND 12.0fF
C20 VSS GND 111.9fF
C21 VDD GND 178.5fF

** hspice subcircuit dictionary
* x0 posvartrans_11/transconductance_1
* x1 posvartrans_11/transconductance_0
* x2 posvartrans_10/transconductance_1
* x3 posvartrans_10/transconductance_0
* x4 posvartrans_9/transconductance_1
* x5 posvartrans_9/transconductance_0
* x6 posvartrans_8/transconductance_1
* x7 posvartrans_8/transconductance_0
* x8 posvartrans_7/transconductance_1
* x9 posvartrans_7/transconductance_0
* x10 posvartrans_5/transconductance_1
* x11 posvartrans_5/transconductance_0
* x12 posvartrans_4/transconductance_1
* x13 posvartrans_4/transconductance_0
* x14 posvartrans_3/transconductance_1
* x15 posvartrans_3/transconductance_0
* x16 posvartrans_2/transconductance_1
* x17 posvartrans_2/transconductance_0
* x18 posvartrans_1/transconductance_1
* x19 posvartrans_1/transconductance_0
* x20 posvartrans_17/transconductance_1
* x21 posvartrans_17/transconductance_0
* x22 posvartrans_16/transconductance_1
* x23 posvartrans_16/transconductance_0
* x24 posvartrans_15/transconductance_1
* x25 posvartrans_15/transconductance_0
* x26 posvartrans_14/transconductance_1
* x27 posvartrans_14/transconductance_0
* x28 posvartrans_13/transconductance_1
* x29 posvartrans_13/transconductance_0

***************
Vdd VDD 0 5
Vss VSS 0 -5

Vstart start 0 PWL(0 0 100e-9 0 101e-9 5 400e-9 5 401e-9 0 4000e-9 0)
Vselect select 0 PWL(0 0 400e-9 0 401e-9 5 400e-6 5)

Vz11 VZ11 0 1
Vz12 VZ12 0 1
Vz13 VZ13 0 1

Cx11 VX11INITIAL 0 4pF
Cx12 VX12INITIAL 0 4pF
Cx13 VX13INITIAL 0 4pF

******* INPUTS *---- Initializing the State "VXINITIAL"
Vx11 initial in_i11 0 DC 1
Vx12 initial in_i12 0 DC -1
Vx13 initial in_i13 0 DC -1

*************** Setting the transconductance values
Vyc11g1 VYC11G1 0 5 *gyc=1
Vyc11g4 VYC11G4 0 -5
Vyc11gr1 VYC11GR1 0 3.67
Vyc11gr4 VYC11GR4 0 -3.67
Vz11g1 V211G1 0 5 *gz=1 for FILBLACK cell
Vz11g4 V211G4 0 -5
Vz11gr1 VZ11GR1 0 3
Vz11gr4 VZ11GR4 0 -3

************
Vyc12g1 VYC12G1 0 5
Vyc12g4 VYC12G4 0 -5
Vyc12gr1 VYC12GR1 0 3.67
Vyc12gr4 VYC12GR4 0 -3.67

Vz12g1 VZ12G1 0 5
Vz12g4 VZ12G4 0 -5
Vz12gr1 VZ12GR1 0 3
Vz12gr4 VZ12GR4 0 -3

**************
Vyc13g1 VYC13G1 0 5
Vyc13g4 VYC13G4 0 -5
Vyc13gr1 VYC13GR1 0 3.67
Vyc13gr4 VYC13GR4 0 -3.67

Vz13g1 VZ13G1 0 5
Vz13g4 VZ13G4 0 -5
Vz13gr1 VZ13GR1 0 3
Vz13gr4 VZ13GR4 0 -3

***************
Vx11g1 VX11G1 0 4
Vx11g4 VX11G4 0 -4
Vx11gr1 VX11GR1 0 5
Vx11gr4 VX11GR4 0 -5

Vy11g1 VY11G1 0 5
Vy11g4 VY11G4 0 -5
Vy11gr1 VY11GR1 0 5
Vy11gr4 VY11GR4 0 -5

****
Vx12g1 VX12G1 0 4
Vx12g4 VX12G4 0 -4
Vx12gr1 VX12GR1 0 5
Vx12gr4 VX12GR4 0 -5

Vy12g1 VY12G1 0 5
Vy12g4 VY12G4 0 -5
Vy12gr1 VY12GR1 0 5
Vy12gr4 VY12GR4 0 -5

Vy12g1 VY12G1 0 5
Vy12g4 VY12G4 0 -5
Vy12gr1 VY12GR1 0 5
• HSPICE simulation results

B.15 LOGIC NOT single cell

![Graph showing HSPICE simulation results for LOGIC NOT single cell]
B.16 LOGIC NOT 1×3 CNN

B.17 SHIFT EAST 1×3 CNN
B.18 DILATION WEST 1×3 CNN

- Magic Layouts for CNN library of applications

B.19 LOGIC NOT 1×3 CNN
B.20 SHIFT EAST 1×3 CNN
B.21 DILATION WEST 1×3 CNN

a)

b)