4. VQ SYSTEM

In this chapter, I will present the “DCELL matrix Layout Automation Tool” (LATool) developed to automate the design of the DCELL matrix layout. In Chapter 2, I described that the Vector Quantizer System divides the input vector space to the hyper-planes in such way that every template-vector in the template-book represents one of them. I will demonstrate that property with 3 simple examples. Finally, I will present realized VQ system.

4.1 DCELL matrix Layout Automation Tool (LATool)

LATool consists of a SKILL code and a Parameterized DCELL layout (PDCELL). I will first present the PDCELL that is a special layout drawn using Cadence® Parameterized Cell (PCELL) Tool and then explain tool’s SKILL code.

I analyzed in detail all the building blocks (DCELL, ROW_WTA and IN_AMP) of the DCELL matrix in Chapter 3. PDCELL is a special parametric layout cell consisting of blocks DCELL and ROW_WTA. LATool uses PDCELL to reproduce layout view of the matrix. In order to reduce silicon cost, layout view of PDCELL is designed as modular as possible. If we examine closely the structure of the matrix, we can note the following conclusions:

1. Number of row of the matrix is equal to the size of the template-book.
2. Each row is terminated by a ROW_WTA block
3. Number of columns of the matrix is equal to the size of the template-vector
4. An IN_AMP block must drive the common input nodes of each column. Thus we must utilize 1 IN_AMP block for each column.

The connection complexities and the device count of the blocks DCELL and ROW_WTA allow the design of a modular layout. Unlike these 2 blocks, due to its relatively large silicon area consumption, the block IN_AMP is not suitable for a modular placement. The layout view of PDCELL can be seen in Figure 4.1

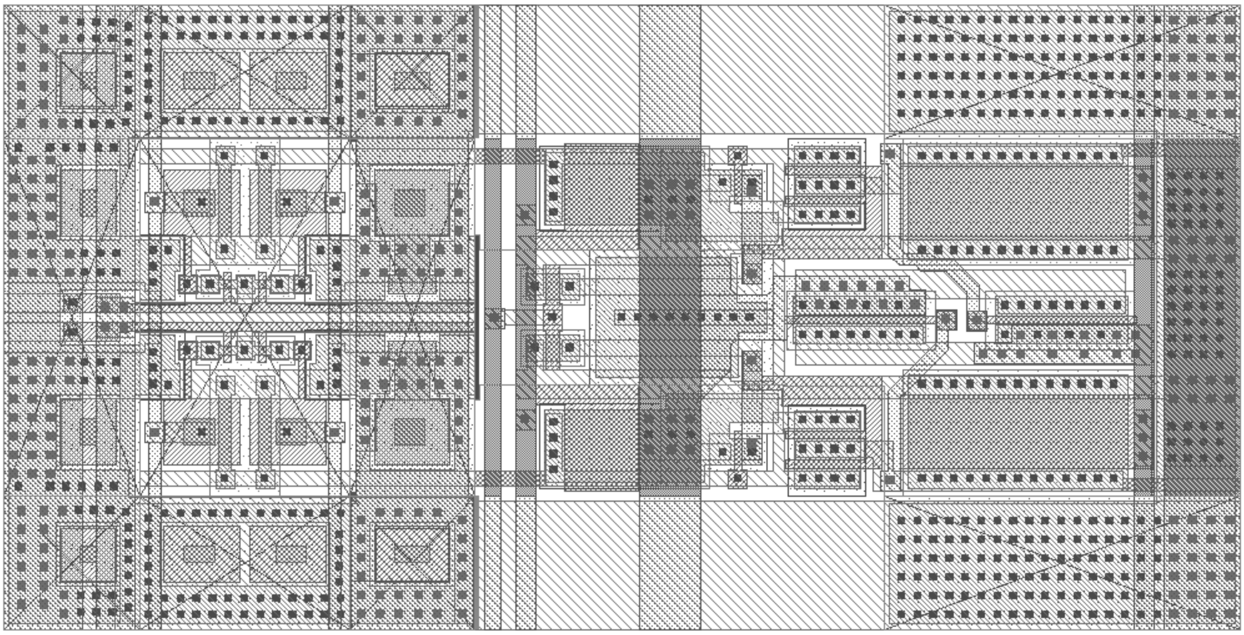

Figure 4.1 : The layout view of PDCELL

We can easily note that the block PDCELL is constituted by 2 rows containing only 1 DCELL and ended by the block ROW_WTA. The layout in Figure 4.1 is built of 12 sub-block's layout. The decompositions of PDCELL's layout view can be seen in Figure 4.2.

1	2	3	4
5	6	7	8
9	10	11	12

Figure 4.2 : The decompositions of PDCELL's layout view

Sub-blocks 3,7 and 11 are utilized to create Power Grid in the DCELL matrix. Sub-block 6 contains the DCELL blocks and sub-block 8 contains the ROW_WTA blocks. Sub-blocks 1,2,3,4,5,9,10,11 and 12 are used to create a guard ring around the DCELL matrix. The cell has 5 Boolean parameters named "SOL", "SAG", "UST", "ALT" and "ARA". These parameters indicating the position of the cell in the matrix, control the conditional inclusion of the sub-block layouts into the

PDCELL layout. For example, if all of the parameters are false, the layout of PDCELL contains only the layout view of the sub-block 6. The table showing the relation between truth table of the Boolean parameter and the inclusion of the sub-block's layout into the PDCELL's layout can be seen in Table 4.1.

Table 4.1 : Relation between truth table of the PDCELL's Boolean parameterand the inclusion of thesub-block's layout

Block No	UST	ALT	SOL	SAG	ARA	Boolean Expression
1	True		True			UST & SOL
2	True					UST
3	True			True	True	UST & (SAG | ARA)
4	True			True		UST & SAG
5			True			SOL
6						
7				True	True	SAG | ARA
8				True		SAG
9		True	True			ALT & SOL
10		True				ALT
11		True		True	True	ALT & (SAG | ARA)
12		True		True		ALT & SAG

To utilize LATool, we also have to load the SKILL code describing the procedures of the tool from the CIW window of the Cadence DFII. Loading operation can be done basically with the command **load("*skill_code_file_name*")**. As a result of loading the skill code, a new menu item "DCELL Layout Automation Tool" appears in the menu named "Tools" of CIW window. By choosing this menu item, we call the procedure **DCELLmenu()** that creates an application form containing the input fields of the design parameters. After the form's fields are filled, a procedure named **paramkontrol()** controls if the specified design parameters are within the allowed range. If not, the procedure **DCELLmenu()** is recalled and we refill the application form. Otherwise, the procedure **paramkontrol()** call in turn the procedure named **DCELLlayout()** for creating physical views such as layout, extracted, abstract and the procedure **DCELLsymbol()** for creating the symbolic view of the design. The procedure **DCELLlayout()** can run only with the layout view of PDCELL.

Let us now examine the input fields of the application form appearing when ‘DCELL Layout Automation Tool’ menu item in the ‘Tools’ menu is chosen.

The library name where the design will be created must be written in the input field “Design Library”. The skill code creates the library if necessary.

The name of the design that will be created in design library must be written in the input field “Design Name”.

The number of the template-vector in the template-book must be written in the input field “Template-Book Size”. This number is equal to the number of row of the matrix. As mentioned previously, PDCELL has 2 rows, thus if the number of template-vector is not multiple of 2, we add one extra row to the design. The block ROW_WTA works slower with the increase of the row number. Thus, although there is no limitation for the layout generation, the maximum allowed row number is determined as 100.

The size of the template-vector must be written in the input field “Template-Vector Size”. This number is equal to the number of column of the matrix. The upper bound of the vector size is determined as 100.

As mentioned previously, with the aid of the parameter “ARA” of PDCELL, we can easily create power grid columns in the matrix. The number of the block DCELL between two adjacent power grid column must be written in the input field “Power Grid Step”. Number of power grid column in a matrix can be determined with respect to the parameters: the power consumption of a DCELL and the number of DCELL in the row.

Finally, we indicate if we want the extracted view used in the post-layout simulation of the matrix. Obviously, the extraction time increases with the increase of the matrix dimensions.

Once the form is filled up, the skill code creates the specified design in the desired library. It creates the layout view of the design by configuring PDCELL with respect to the determined parameters and placing it as a matrix manner. Afterwards, it adds the input/output pins to the layout. It creates extracted view if wanted. Finally, it creates the symbolic view used in the schematic views and the abstract view used by the Place&Route tools.

As mentioned above, the developed SKILL code, especially the procedure **DCELLlayout()**, may run with only a special parameterized cell named PDCELL designed suitably to the code and a special parameter set indicating the coordinate of the input/output pins of the PDCELL. Under the condition that we obey the Boolean parameter structure and modifying pins coordinate set, this code can be used also with different PDCELL's layout optimized for different circuit parameters. The code does not contain any information about the structure of the circuit. The SKILL code can be found in Appendix K.

A DCELL matrix generated using LATool for the following parameter can be seen in Figure 4.3.

Template-book Size= 4   Template-Vector Size= 10   Power Grid Step=6

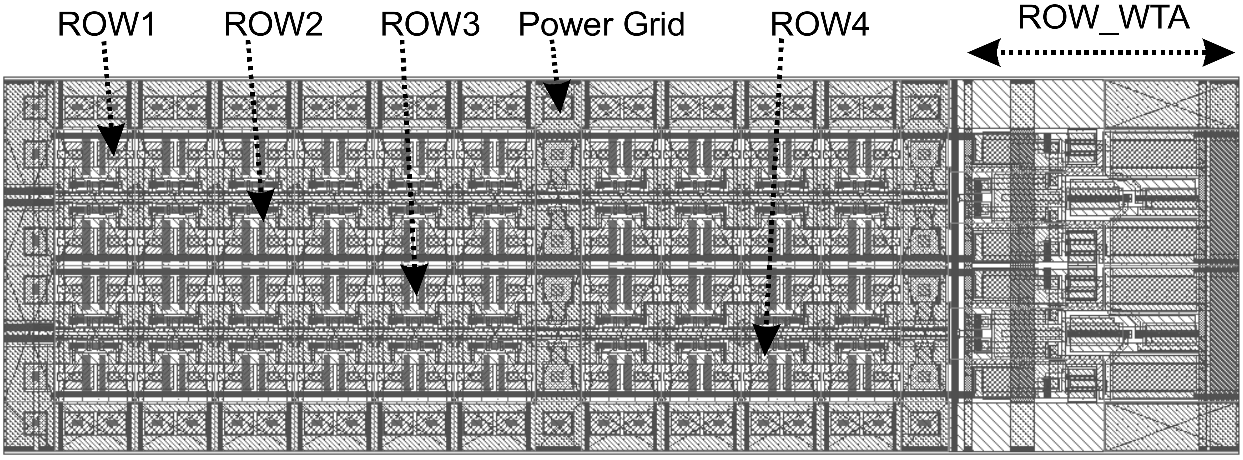

Figure 4.3 : DCELL matrix generated using LATool.

4.2 Examples

In Chapter 2, I mentioned that N being the number of the template-vector in the template-book, the Vector Quantizer system divides the input vector space to N hyper-planes. In order to show this partition, I will analyze three simple examples. For the simplicity, let us assume that the input vector space dimension and the template-book size, both are equal to 2. Thus, we want to divide a plane (more generally a surface) to 2 separate region. To simulate this system, we need a DCELL matrix of size 2x2. We have designed 2 matrix blocks. The first one is designed using LATool. In order to show the response of an ideal system, the second one is designed using AHDL code of the building blocks.

The input signal range of the matrix is 0-3V. Thus, we can express the template-vectors as:

$T_I = \{(t_{i1}, t_{i2}) | t_{i1}, t_{i2} \in [0\ 3]\}$ where I=1,2

For the first example, let us chose the template-vectors as follows:

$T_1 = (1, 1.5) \quad T_2 = (2, 1.5)$

The partition of the input vector space with respect to these two vectors can be seen in Figure 4.4.

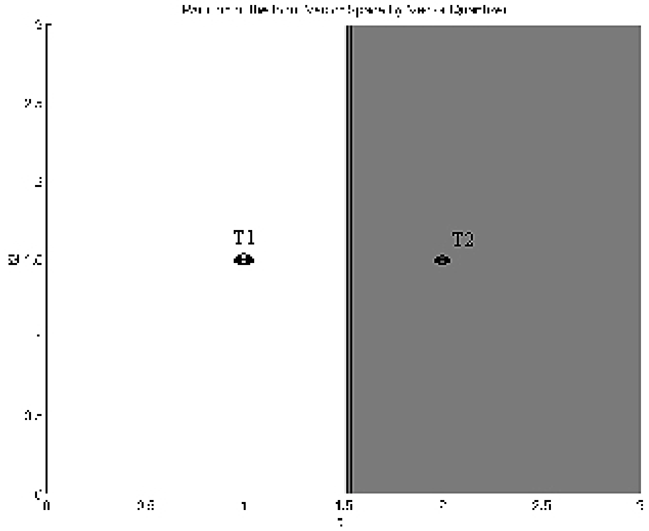

Figure 4.4 : Input Vector space partition for $T_1 = (1, 1.5)$ and $T_2 = (2, 1.5)$

To scan the whole input vector space, I connected two triangle wave sources to the inputs t_l and t_2. By choosing the frequency of the t_2's source sufficiently small with respect to the frequency of t_l's source, we can scan input space row by row. The period of t_2's source is chosen 16ms and the period of t_l's source is chosen 800µs. The variation of the input signals can be seen in Figure 4.5.

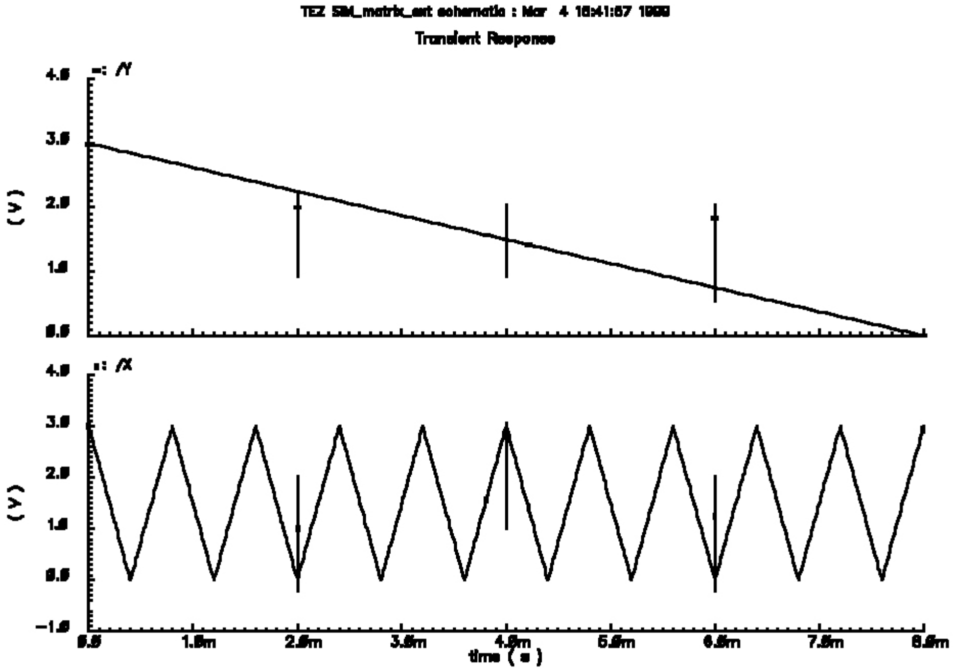

Figure 4.5 : The variation of the input signals t_l and t_2.

Refresh time is 2ms. Refresh period can be seen in Figure 4.5 as sharp changes at every 2ms. In Figure 4.5, the signal /X represents $\mathbf{t_l}$ and the signal /Y represents $\mathbf{t_2}$. If we scan the input space in Figure 4.4 with the input signals in Figure 4.5, we can easily state that the template-vectors T1 and T2 will sequentially be the winner vectors for equal time periods. This behavior can be seen in Figure 4.6 where the outputs of the ideal and the real system are shown. The output of the winner template-vector is Logic 0. The outputs of the real system change smoothly with respect to the ideal system's output. The reason for that is the resolution of the Winner-Takes-All network. As mentioned in Section 3.3, to resolve this problem, we can either increase the output current level of the block DCELL in order to guarantee the difference current is always greater than the resolution or increase the feedback loop gain of the WTA by using for example cascode structures.

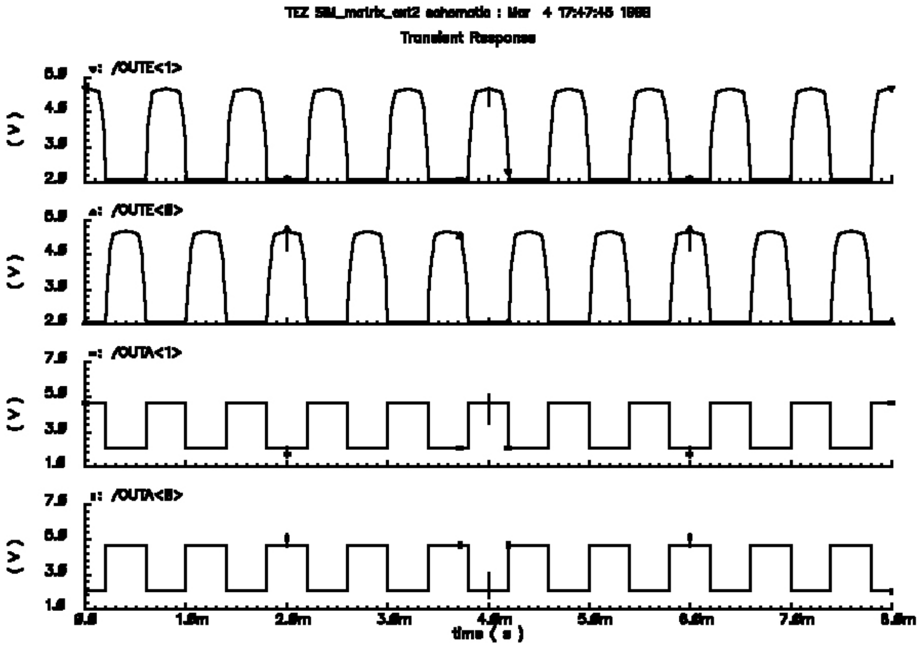

Figure 4.6 : The outputs of the ideal and the "real" system for $T_1 = (1, 1.5)$ and $T_2 = (2, 1.5)$

In Figure 4.6, signals named OUTE<x> are the outputs of the real system. Signals named OUTA<x> are the output of the ideal system. The template-vector T1 is represented by the signals OUTx<0> and the template-vector T2 is represented by the signals OUTx<1>.

As a second example, let us choose the template-vectors as follows:

$T_1 = (1.5, 1)$ $T_2 = (1.5, 2)$

The partition of the input vector space for the second template-book can be seen in Figure 4.7.

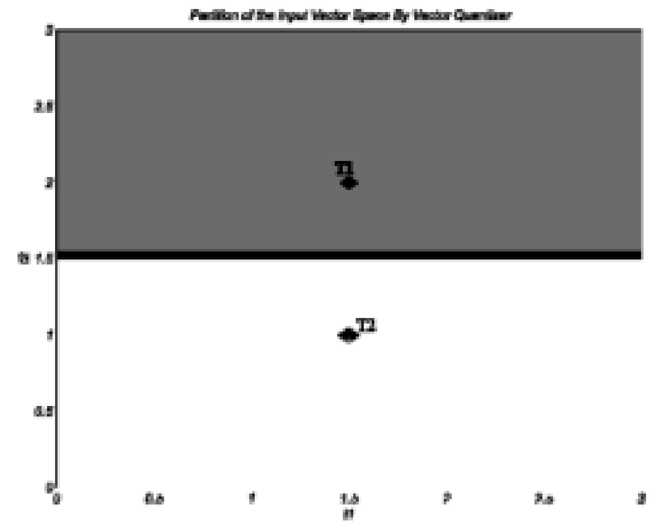

Figure 4.7 : Input Vector space partition for $T_1 = (1.5, 1)$ and $T_2 = (1.5, 1)$

If we scan the input vector space in Figure 4.7 with the input signals in Figure 4.5, we can state that the template-vector T1 will be the winner during the scan of the upper part of the input space and then T2 will be the winner during the rest. This behavior can be seen in Figure 4.8 where the outputs of the ideal and the real system are drawn. The output signal level of the winner is equal to Logic 0. The signals in the figure are described in the previous example.

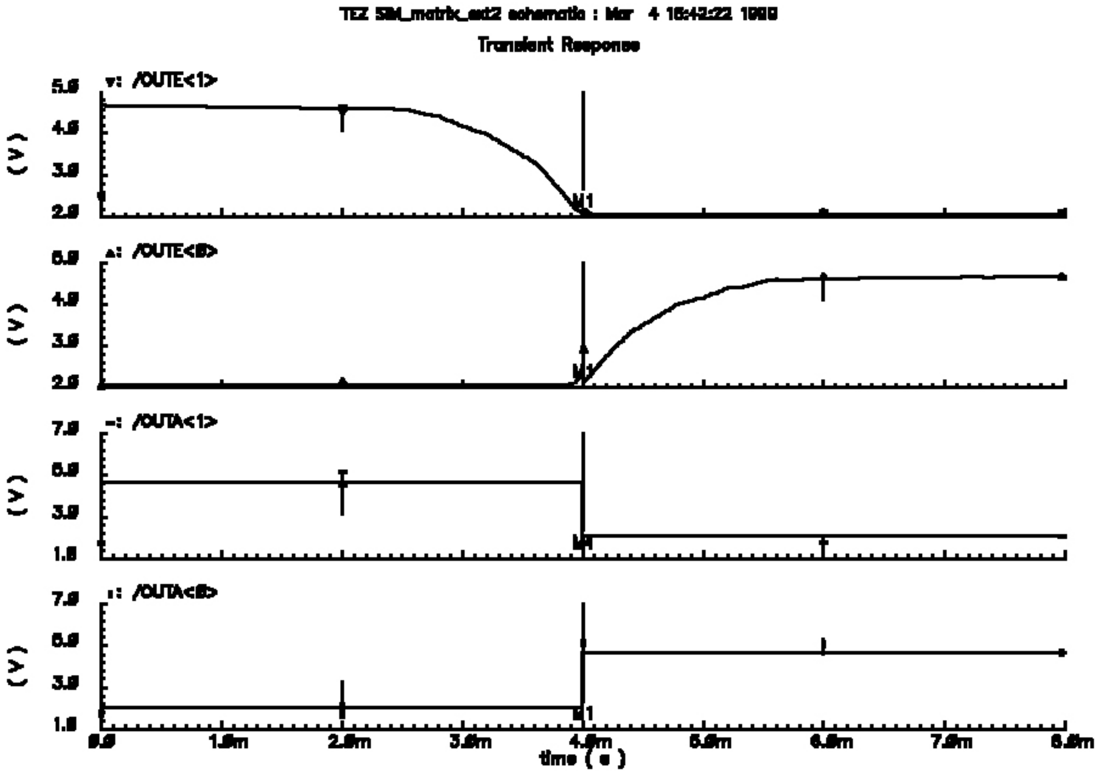

Figure 4.8 : The outputs of the ideal and the real system for $T_1 = (1.5, 1)$ and $T_2 = (1.5, 1)$

As a last example, the template-vectors are chosen as follows:

$T_1 = (1, 2)$ $T_2 = (2, 1)$

This is a more complex example with respect to the previous ones. The partition of the input vector space can be seen in Figure 4.9.

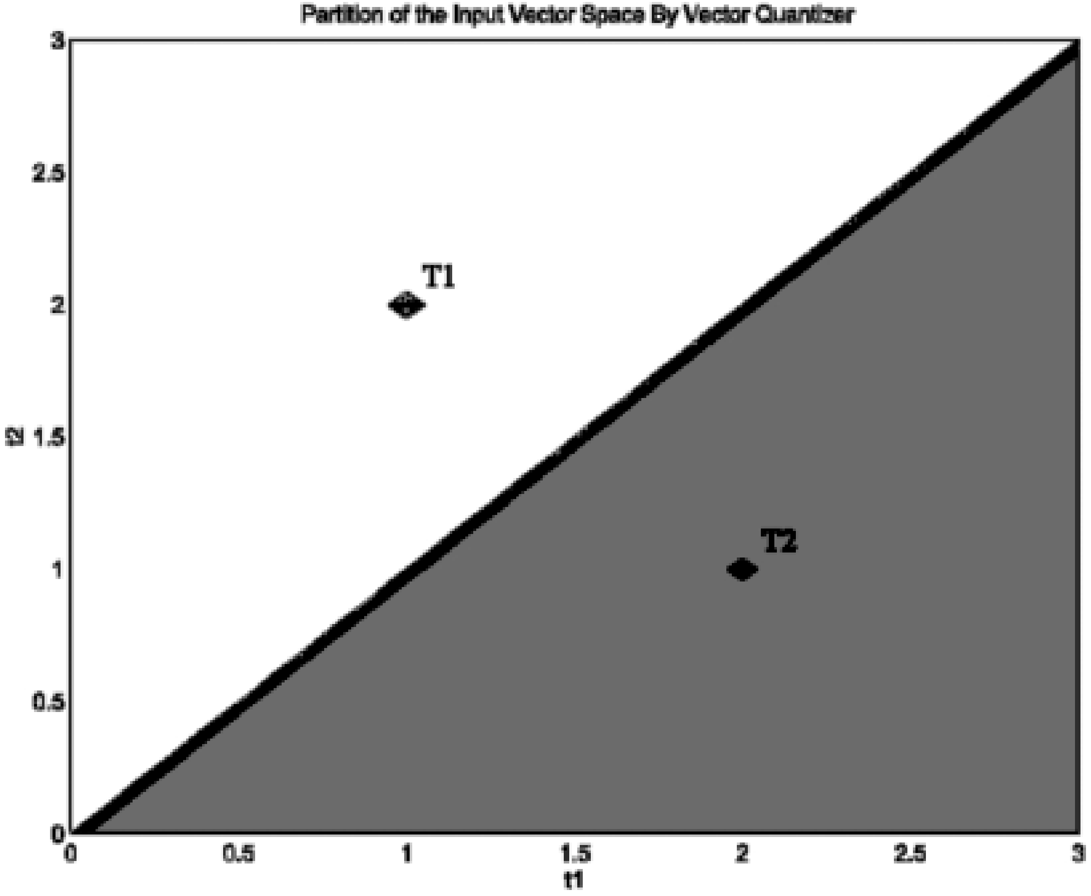

Figure 4.9 : The partition of the input vector space for $T_1 = (1, 2)$ and $T_2 = (2, 1)$

If we scan the input space in Figure 4.9 with the input signals in Figure 4.5, we can state that during the scan of the first row, the template-vector T1 will be the winner for a long time and then T2 will be the winner for a short time. This will be repeated sequentially during the scan of the each row, but each time T1's winning time shortens while T2's winning time grows longer. This behavior can be seen in Figure 4.10. where the outputs of the ideal and the real system are drawn. Signals' description is same as in the previous examples.

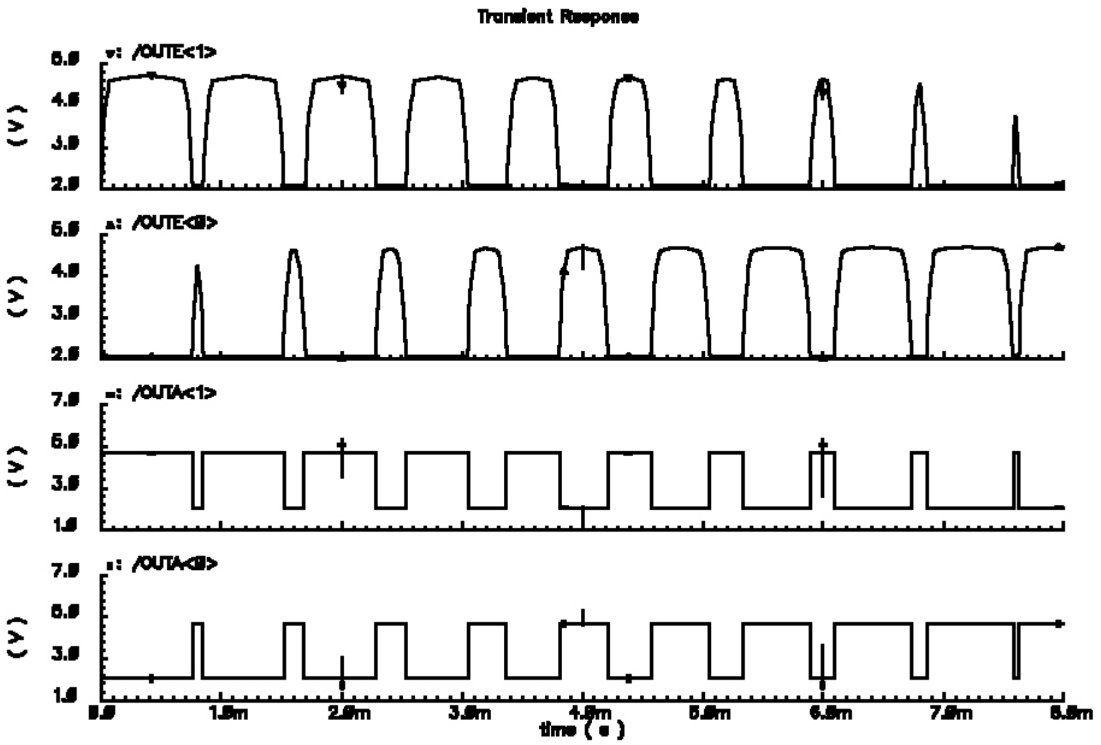

Figure 4.10 : The outputs of the ideal and the real system for T_1 =(1, 2) and T_2 =(2, 1)

4.3 Analysis of the implemented VQ system

In Chapter 3, we have analyzed in detail all the building block of the VQ system. In this chapter, I will concentrate on the Vector Quantizer system built by the analyzed blocks. I will begin with analyze of the global system parameter, System Resolution, System Power Consumption, System Silicon Area Consumption, System Speed and we will terminate by showing the schematic and the layout views of the system.

a. System Resolution

The system resolution is influenced by the blocks IN_AMP, DCELL and ROW_WTA. The resolution of the block IN_AMP is very high for the input signals changing slowly. The reason is the high DC open loop gain of the OPAMP in the block. Obviously, the decrease of this gain due to the increase of the input signal frequency worsens the system resolution. Under the conditions that the input signal frequency is chosen properly, we can state that this block does not determine the resolution limit of the system. The resolution of the block DCELL is determined by the mismatch of the output transistors and the input capacitors. Yet as the system classifies the input vector with respect to the output current of each row in the DCELL matrix, we must take into account the resolution of the row current which is equal to the sum of resolutions of each DCELL block on the row. By increasing the size of the components and/or increasing the number of DCELL block in the row, the required resolution can be achieved. Thus, the resolution limit is not determined neither by the block DCELL. The upper bound of the resolution is determined by precision of the block ROW_WTA analyzed in Chapter 3.3.

b. System Power Consumption

In the VQ system, the biasing circuitry of the block IN_AMP and the voltage dividers of the block DAC6 consume the major part of the power. The digital blocks are designed using CMOS standard cell library. Thus, there is not a static power consumption for these blocks. The power consumption of the CMOS standard cells and the block RAM is a function of their operating frequency. Finally, the power consumption of the DCELL matrix can be easily calculated using the design parameters I_B, I_K and I_{TAIL}. The total power consumption of the Vector Quantizer system can be expressed as follows:

$$P_{SYSTEM} = M\left(P_{DAC6} + P_{INAMP}\right) + \underbrace{V_{DD}\left[N\left(I_B + I_K\right) + I_{TAIL}\right]}_{\text{DCELL Matrix}} + \frac{N}{T_{REF}} P_{F(RAM)} + P_{DIG} \qquad 4.1$$

Where N is the number of row in the DCELL matrix; M is the number of column in the DCELL matrix; P_{DAC6} is the power consumption of one DAC6 block; P_{INAMP} is the power consumption of one IN_AMP block; $P_{F(RAM)}$ is the dynamic power consumption of the RAM block; T_{REF} (Refresh Time) is the time period between two adjacent refresh period; Finally P_{DIG} is the power consumption of the digital blocks. The value of these parameters in the implemented system can be seen in Table 4.2.

Table 4.2 : The parameters related to the power consumption of the system

Parameter	Value	Unit
Number of Row (N)	64	
Number of column (M)	16	
I_B	30	µA
I_K	1	µA
I_{TAIL}	50	µA
V_{DD}	5	V
P_{DAC6} (For $V_P - V_N = 3\,V$)	2.4	mW
P_{INAMP}	3.1	mW
$P_{F(RAM)}$	116	µW/MHz
T_{REF}	2	ms
P_{DIG}	> 5	mW

The power consumption of the Vector Quantizer system is approximately equal to P_{SYSTEM}=105mW. The blocks DAC6 and IN_AMP consume more than 80% of the total system power.

c. System Silicon Area Consumption

In the implemented system, the building blocks are ranged with respect to their silicon area consumption as follows: RAM, Digital blocks, IN_AMP, DAC6, ROW_WTA, DCELL. RAM block occupies the most of the system silicon area. As mentioned in Chapter 3, the required number of RAM's word is equal to the template-book size and the RAM's word size is equal to the product of the template-vector size and DCELL resolution. If we assume that the silicon area consumption (SAC) of the block DCELL is approximately equal to 1 SRAM cell's, the SAC of the RAM block can be calculated as a product of DCELL resolution and the SAC of the DCELL matrix. We will see later that the DCELL matrix, ROW_WTA column and ENCODER block occupy only 2.7% of the system SAC while the RAM block occupies 70% of it. In fact, one of the reasons of this huge difference is that the RAM block is organized using 12 small capacity RAM block. We can state that the cell count of the digital blocks is a weak function of only the template-book size. The Vector Quantizer system contains one DAC6, IN_AMP blocks couple for each column of the matrix. Thus, the SAC of the system core can be expressed as follows:

$$\mathrm{Area}_{\mathrm{CORE}} = \left(\underbrace{r}_{\mathrm{RAM}} + \underbrace{1}_{\mathrm{DCELLmatrix}} \right) N.M.A_{\mathrm{DCELL}} + M.\left(A_{\mathrm{DAC6}} + A_{\mathrm{INAMP}}\right) + N.A_{\mathrm{ROW\_WTA}} + A_{\mathrm{DIG}} \quad 4.2$$

Where N is the number of row of the DCELL matrix, M is the number of column of the DCELL matrix, **r** is the resolution of the block DCELL, A_X is the SAC of the block X. The parameter set of the implemented system can be seen in Table 4.3.

Table 4.3 : The parameter related to the silicon area consumption of the VQ system

Parameter	Value	Unit
DCELL resolution (r)	6	bit
Number of row (N)	64	
Number of column (M)	16	
A_{DCELL}	2.1172e-4	mm^2
A_{DAC6}	27.87e-3	mm^2
A_{INAMP}	55.491e-3	mm^2
A_{ROW_WTA}	7.705e-4	mm^2
A_{DIG}	651.69e-3	mm^2

When we substitute above listed parameters in (4.2), the estimated core area is found approximately equal to 3.6 mm^2. However, the area occupied by the implemented system core is 7.9 mm^2. As mentioned previously, the difference is due to the fragmented structure of the RAM block. We have assumed 1.3 mm^2, in fact it occupies 5.6 mm^2.

d. System Speed

The maximum input signal frequency is determined with respect to two parameters. The first one is the settling time of the block IN_AMP and the second one is the classification speed of the Winner-Takes-All network. This last one is highly affected by the input signal difference between the winner and the other cell. Obviously, for large input difference, network response is faster.

The block ROW_WTA was designed to respond within the 100ns for 1μA row current difference. The settling time of the block IN_AMP is 360ns.

The clock frequency of digital blocks is determined with respect to 4 parameters. The first one, as before, is the settling time of the IN_AMP block, the second one is the conversion time of the DAC6 block, the third one is the access time of the RAM block, and the last one is the DCELL row activation signal delay. The conversion time of the DAC6 is less than 20ns. The access time of the RAM is 7.73ns. The control signal delay is 6ns. The clock frequency is chosen 2MHz.

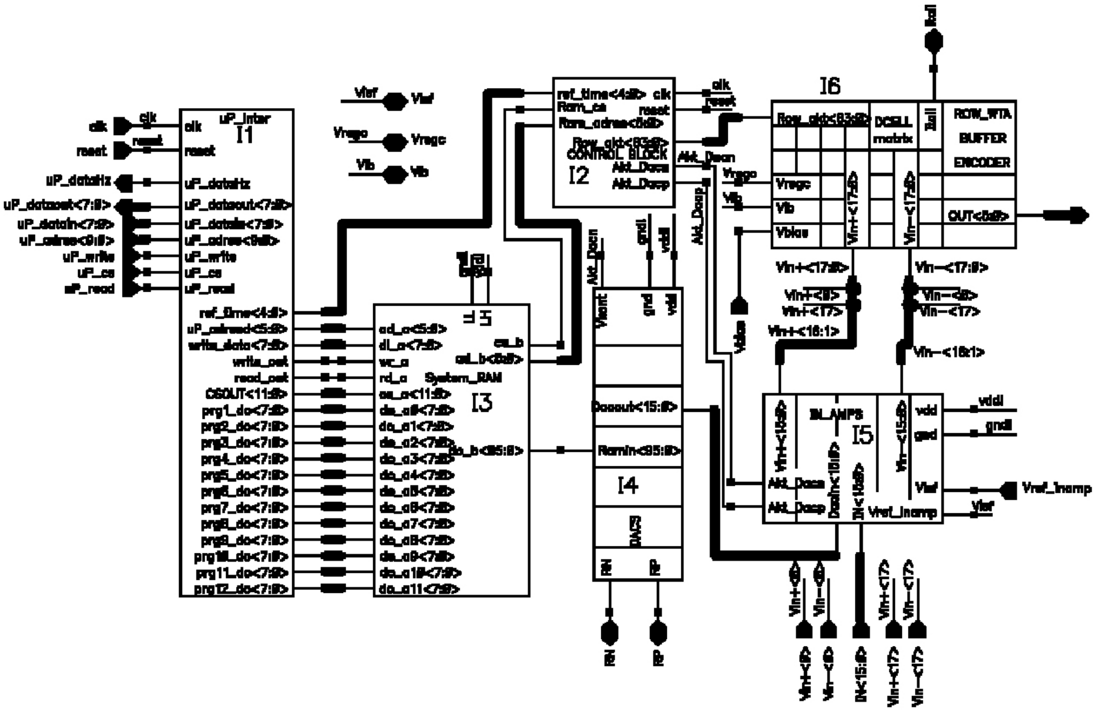

Figure 4.11 : The Schematic view of the Vector Quantizer System

The schematic view of the Vector Quantizer system can be seen in Figure 4.11. During the implementation phase, we have added 2 extra DCELL column in the matrix for testing. The input nodes of these two columns are directly connected to the input pads named $V_{IN+<0>}$, $V_{IN-<0>}$, $V_{IN+<17>}$ and $V_{IN-<17>}$.

In Figure 4.11, the block number I1 (uP_inter) contains the standard cells of the block Interface. The block number I2 (Control Block) contains the standard cells of the block Control. The block number I3 (System_RAM) contains the dual port RAM blocks. The schematic view of this block can be seen in Figure 3.82. The block number I4 (DACS) contains 16 independent DAC6 blocks. The block number I5 (IN_AMPS) contains 16 independent IN_AMP blocks. The schematic view of the block number I6 can be seen in Figure 4.12.

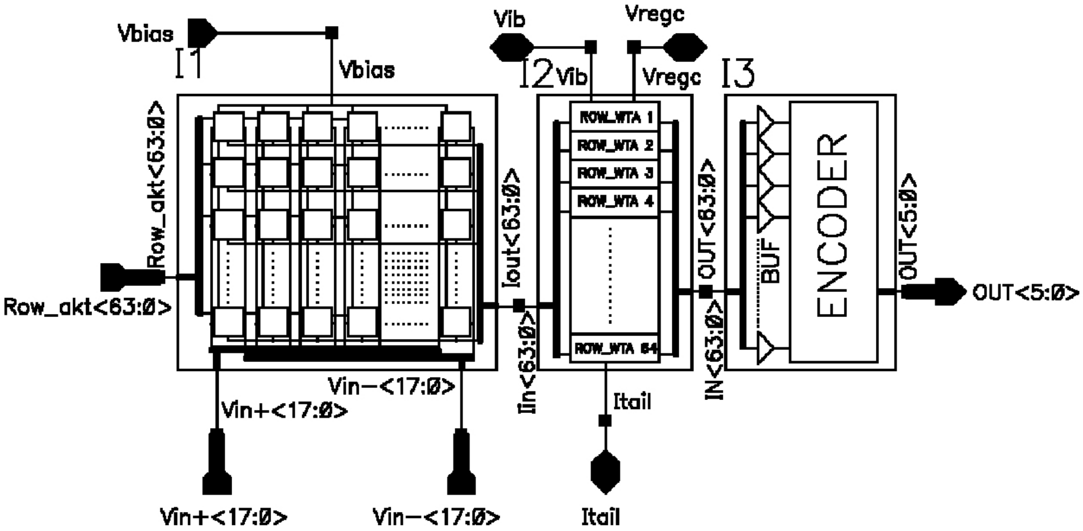

Figure 4.12 : The Schematic View of the block number I6

In Figure 4.12, the block number I1 contains the DCELL matrix. The block number I2 contains 64 ROW_WTA block. The block number I3 contains input buffers and the standard cells of the block Encoder.

The Layout view of the block number I6 in Figure 4.11 can be seen in Figure 4.13.
The Layout view of the block number I5 in Figure 4.11 can be seen in Figure 4.14.
The standard cells of the block number I1 (uP_inter) and the block number I2 (Control Block) in Figure 4.11 are placed in the same placement region. The Layout view of these blocks can be seen in Figure 4.15.
The layout view of the block number I3 (System_RAM) and the block number I4 (DACS) can be seen in Figure 4.16.
The schematic view of the Vector Quantizer System showing the bonding pads can be seen in Figure 4.17.
The Layout view of the Vector Quantizer System can be seen in Figure 4.18.
The bonding diagram of the system for the package type PGA84 and the pinout information can be found in Appendix H.

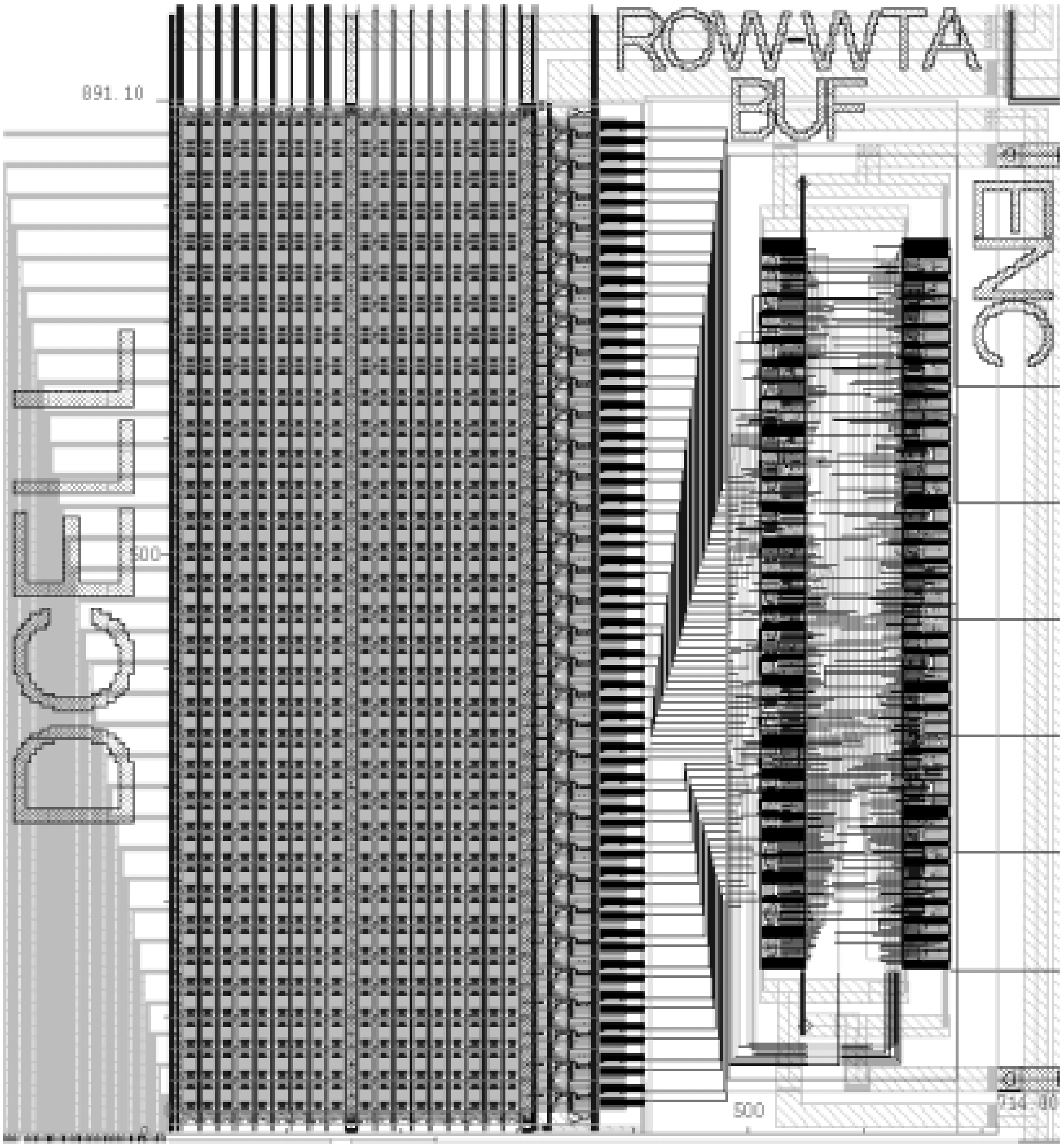

Figure 4.13 : The Layout view of the block number I6

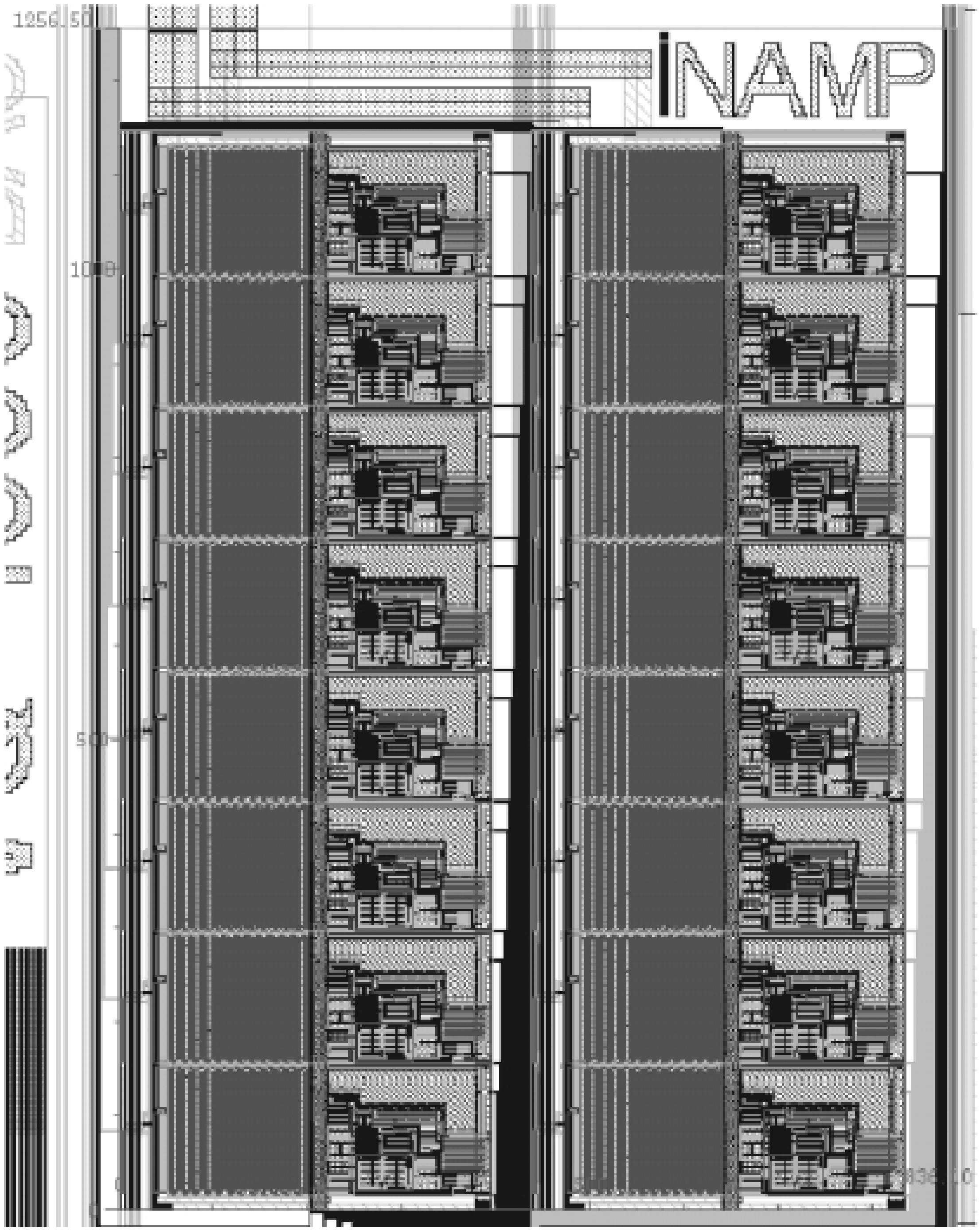

Figure 4.14 : The Layout view of the block number I5

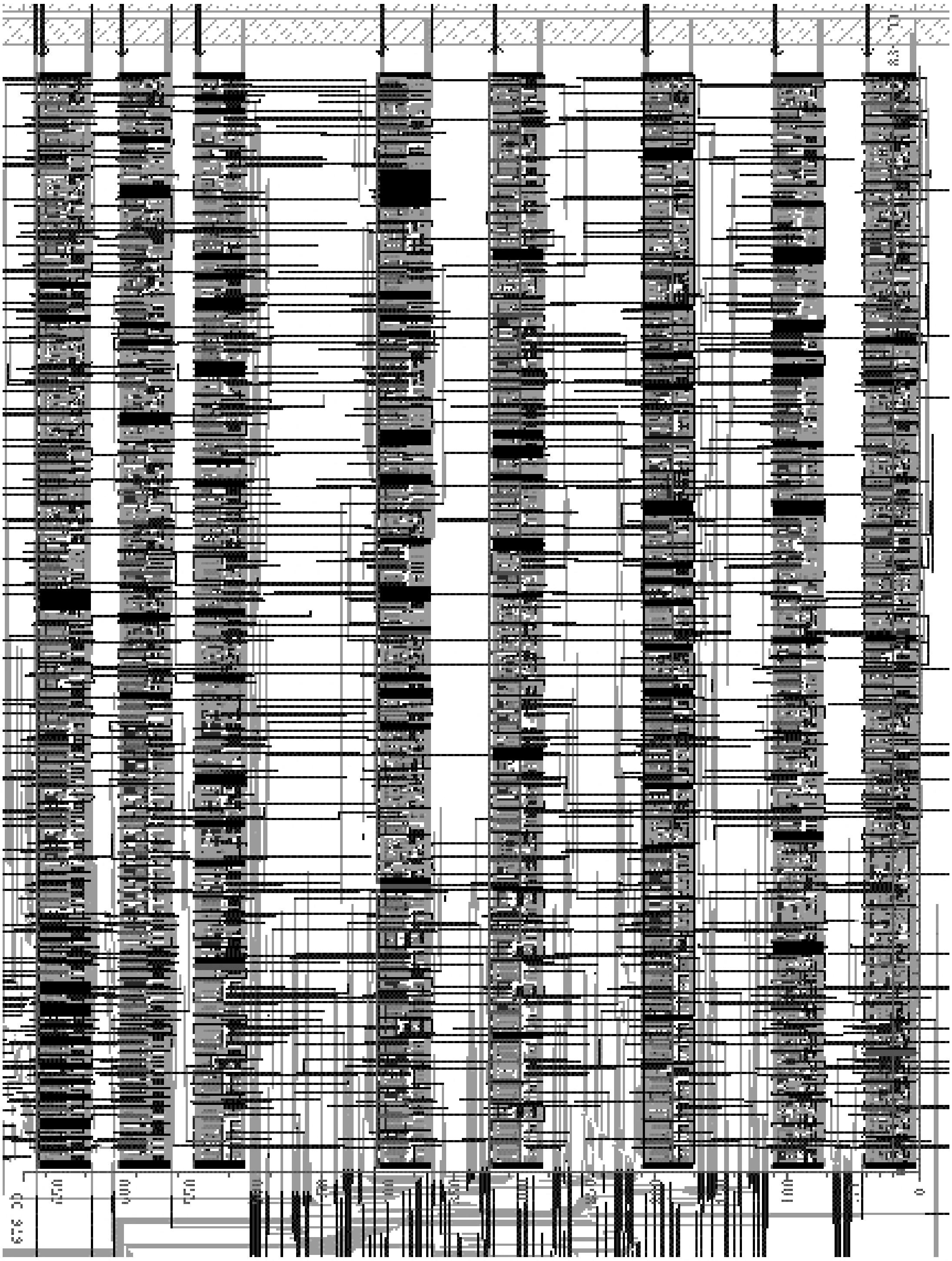

Figure 4.15 : The Layout view of the placed and routed standard cells of the blocks uP_inter and Control.

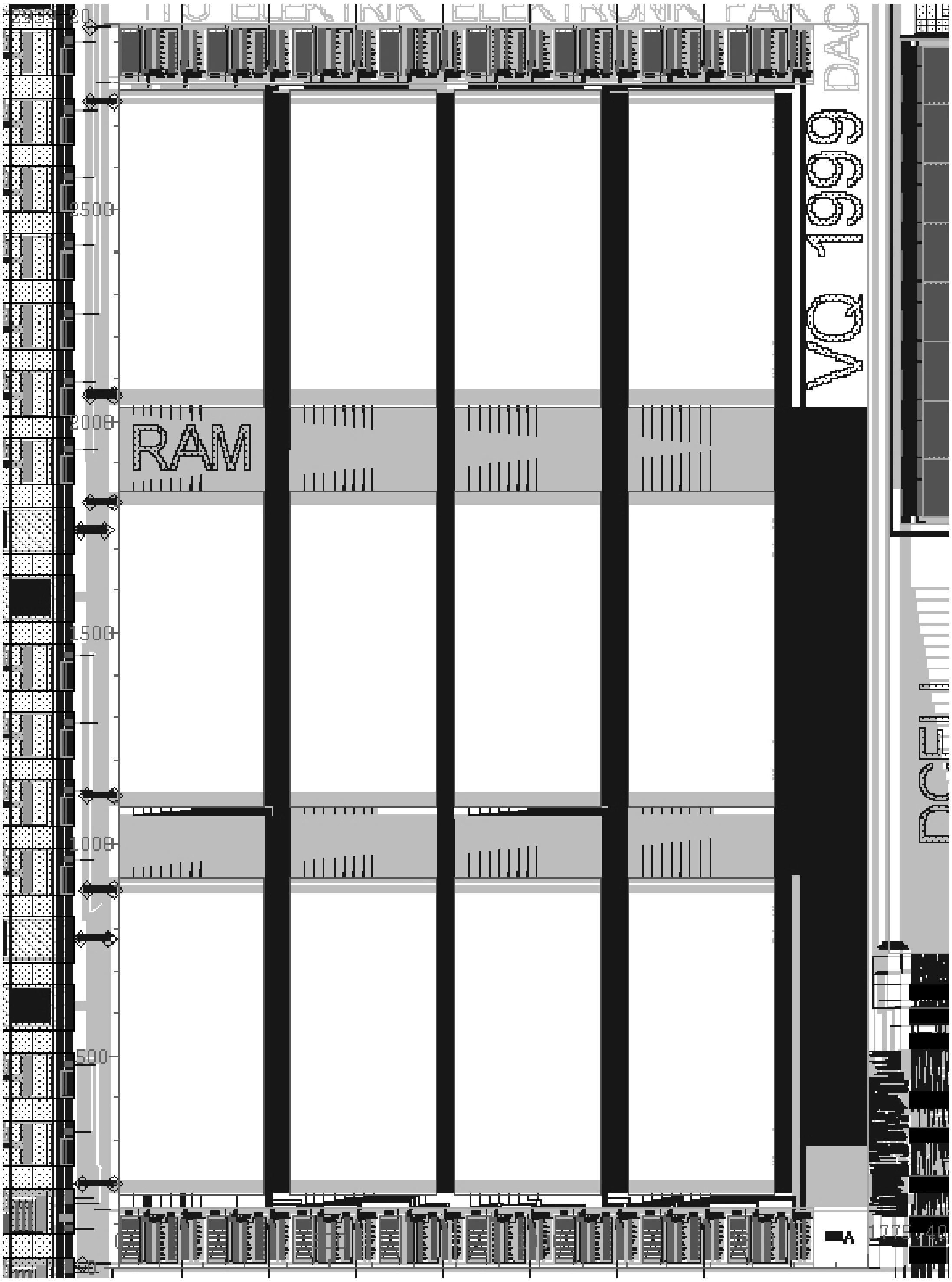

Figure 4.16 : The Layout view of the blocks number I3 and I4

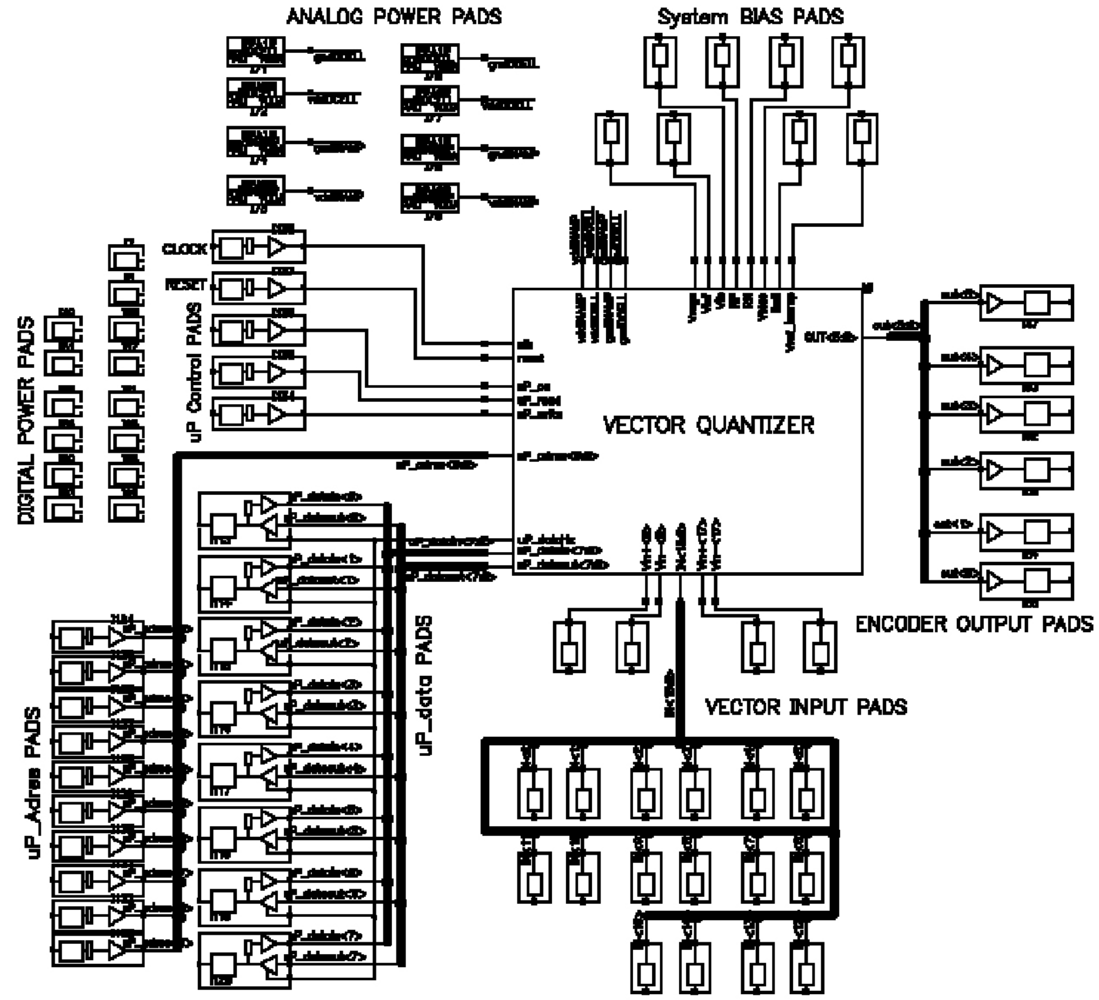

Figure 4.17 : The schematic view of the Vector Quantizer System showing the bonding pads

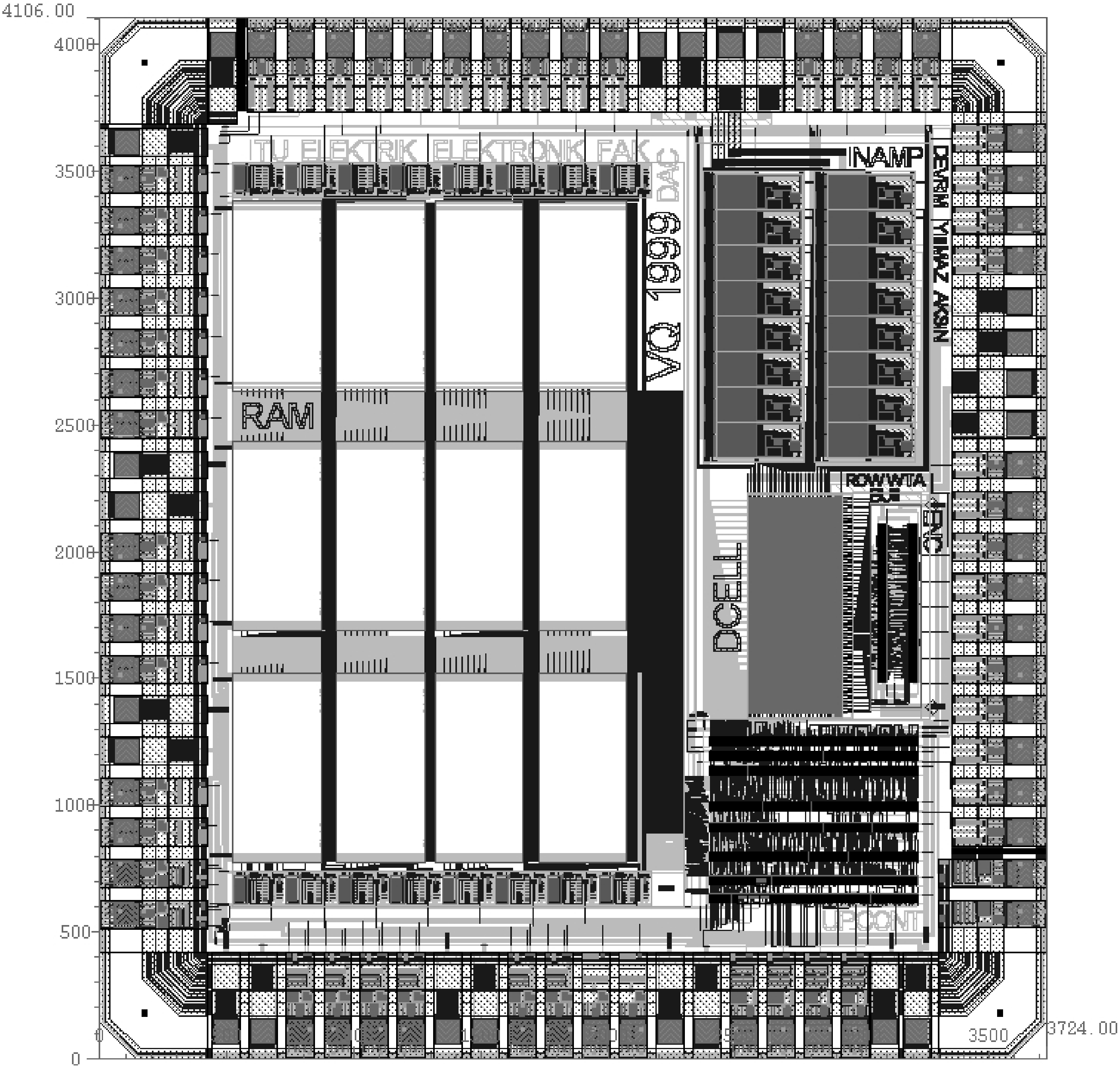

Figure 4.18 : The Layout view of the Vector Quantizer System