Chapter 6

Design and Evaluation of 1:4 Tree Sensors

In this chapter, the designs and the evaluations for the manufactured 1:4 tree sensors are described. The characteristics of photo diodes fabricated the used CMOS 1.5μm process is discussed at first.

The prototype systems of the 1:4 tree image sensor using FPGAs will be described in section 6.2. It will be described the design and the evaluation of the 1:4 tree image sensor having the tree structure of node automata in the following section 6.3.

The design and the evaluation of the 1:4 tree sensor with the architecture of placing address selector externally will be described in section 6.4.

The design and the evaluation of the decoder of 1:4 tree code will be described in the section 6.5.

6.1 Design and Evaluation of Photo Diodes

We have designed the photo diode TEGs (test element groups) to estimate the photo current. The designed TEG chip in 2.3mm×2.3mm using CMOS 1.5μm technology is shown in Figure 6.1[1]. The designed photo diodes are shown in Figure 6.2.

[1]The VLSI chip in this study has been fabricated in the chip fabrication program of VLSI Design and Education Center(VDEC), the University of Tokyo with the collaboration by Nippon

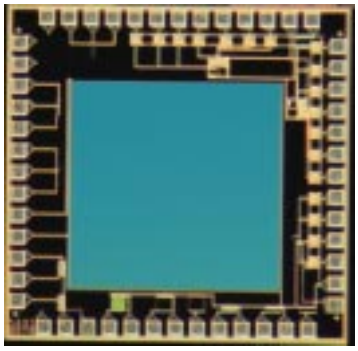


Figure 6.1: Chip photograph of the designed photo diode TEGs

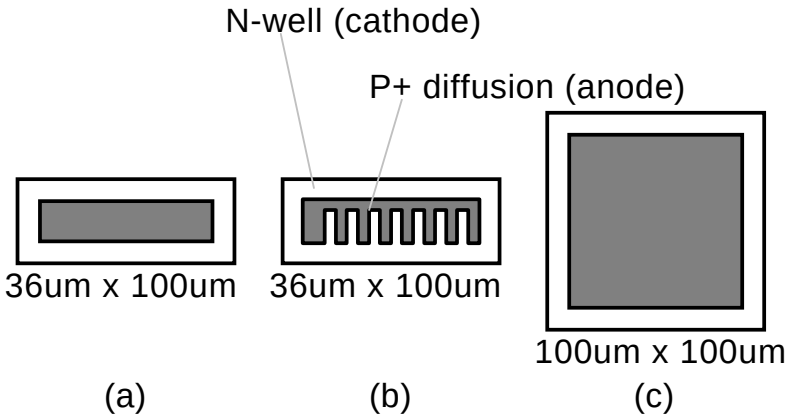


Figure 6.2: Designed photo diodes of tree types. (a)small diode, (b)small comb-shaped diffusion diode, and (c)large diode.

Three types of photo diodes are designed; the small one whose size is $36\mu\text{m} \times 100\mu\text{m}$ shown in Figure 6.2(a), the one of same size with the comb-shaped p+ diffusion in order to obtain the larger edge length of diffusion area shown in Figure 6.2(b), and the large one whose size is $100\mu\text{m} \times 100\mu\text{m}$ shown in Figure 6.2(c).

The characteristics of photo diodes are shown in Figure 6.3. Here the intensity of light is measured by exposure meter in exposure value (EV), defined as the follows,

$$E = 2^{\text{EV}} C/S. \tag{6.1}$$

where E is the illuminance in lux[2], C is the correction coefficient determined ex-

Motorola and Dai Nippon Printing Corporation.

[2]1[lux] is defined as the illuminance at the distance of 1m from the light source of 1[cd]. 1[cd]

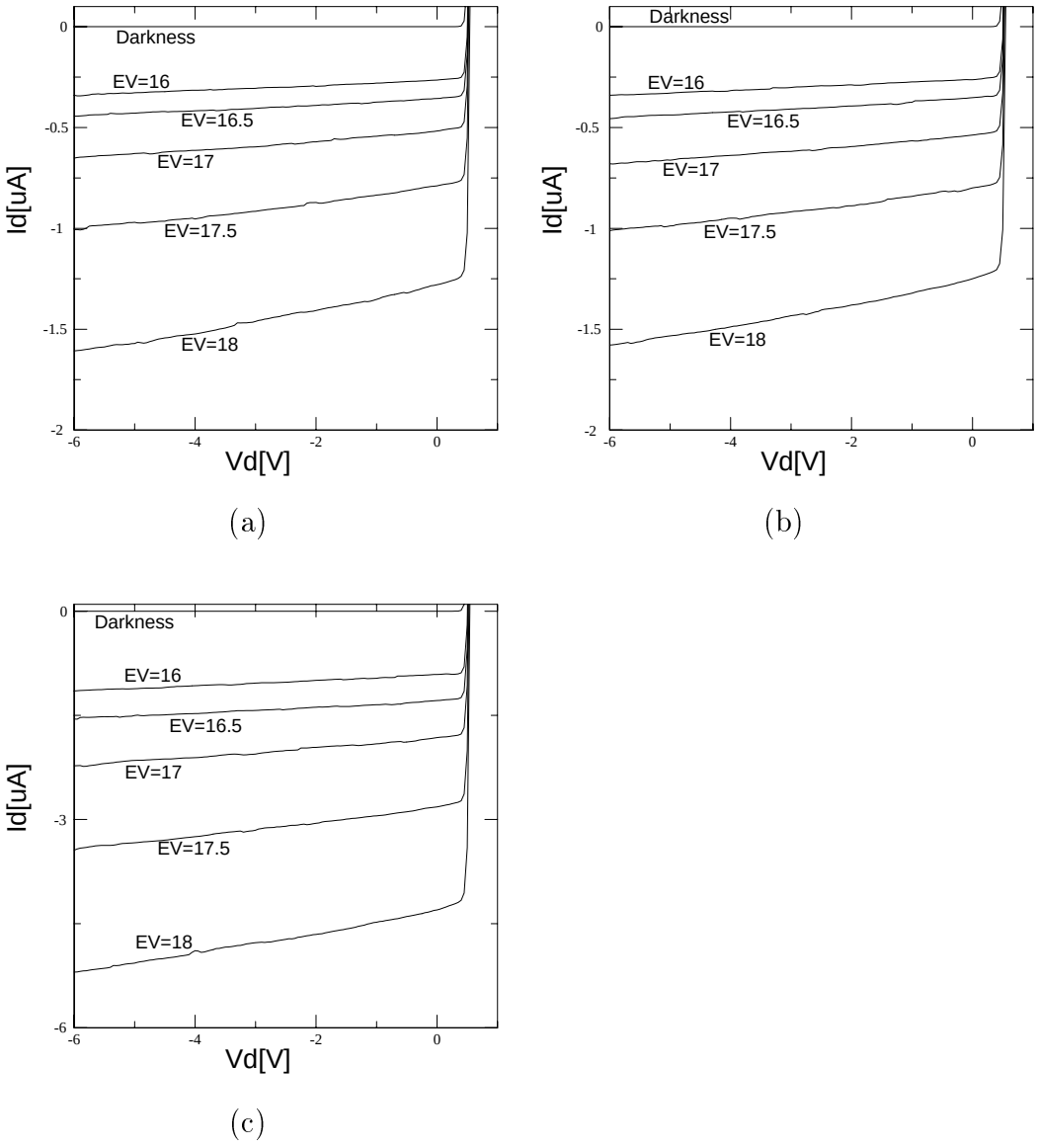


Figure 6.3: Measured characteristics of fabricated photo diodes. (a)small diode, (b)small comb-shaped diffusion diode, and (c)large diode.

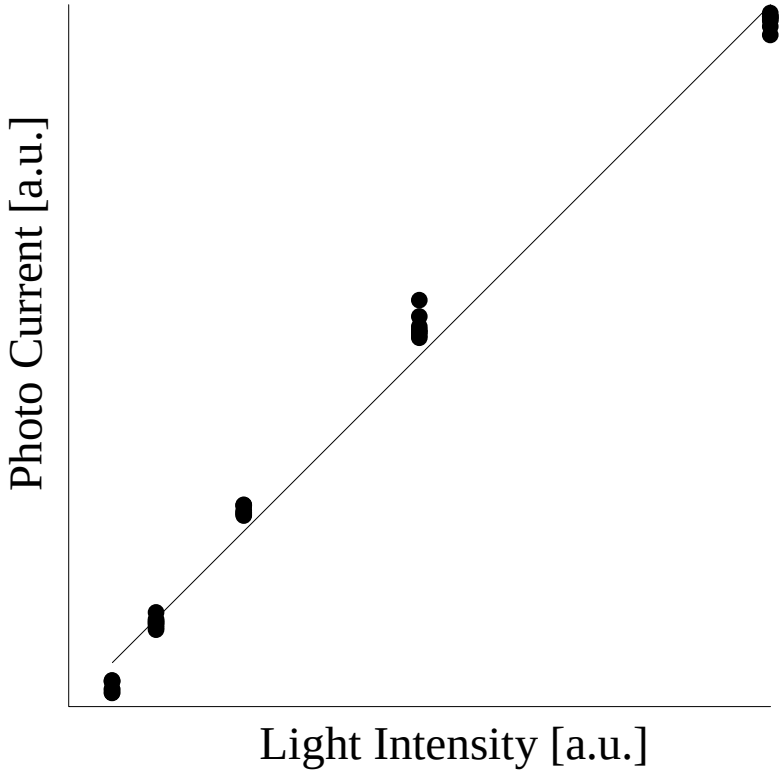


Figure 6.4: Relation of the normalized photo current per area and the normalized light intensity

perimentally which is 330, and S is the film sensitivity used in the exposure meter which is ISO400 in this case. The distance between exposure meter and light source is 3cm.

Seen in Figure 6.3, the photo currents for the photo diode(a) and (b) in Figure 6.2(a) and (b), respectively, are almost the same, and the reason is that the depletion layer between teeth in comb-shaped diffusion area in photo diode (b) is overlapped, and the distribution of depletion layer, which is the area photon generates the pair of electron and hole, is identical with that of photo diode (a).

It is also indicated that the intensity of light increases twice, the photo current is also increase about twice, and the photo current is almost proportional to the size

is defined as 1/60 of the light intensity of the black body whose temperature is the melting point of platinum, 2045K.

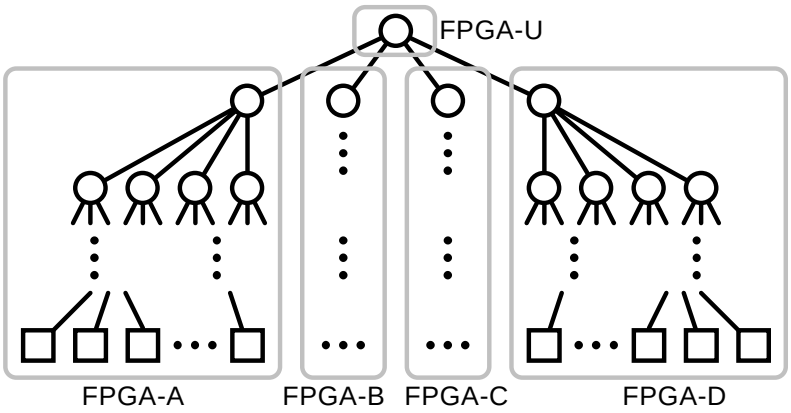


Figure 6.5: Configure of the 1:4 tree structure using five FPGAs.

of the diffusion area. The relation of the normalized photo current per area and the normalized intensity is shown in Figure 6.4, and it indicates that the photo current is proportional to both light intensity and size of diffusion area.

6.2 Prototype System for 1:4 Tree Sensor using Node Automata

6.2.1 Prototype implementation 1:4 tree sensor using FPGAs

We have developed the prototype system of the 1:4 tree sensor using FPGAs(Field Programmable Gate Array). We have employed the FPGA of XC4025[29] , which is the production of Xilinx Inc., which has logic blocks whose logical functions can be defined by the configure ROMs outside. The developed prototype systems consists of five FPGAs, four for the lower sub-trees, and one for the top node, as shown in Figure 6.5, where the number of pixels is $1,024 = 2^{10}$ with the 6 levels of 1:4 tree structure. Figure 6.6 shows the photograph of the developed prototype system.

The function of the pixel is emulated by having the flip-flops as the pixel value, which can be set serially from outside using shift registers, as shown in Figure 6.8. The image data captured by the video camera are transferred to the PC, then the PC transfers them to the prototype system serially.

The 1:4 tree scan for the prototype systems is executed, with the clock signal from

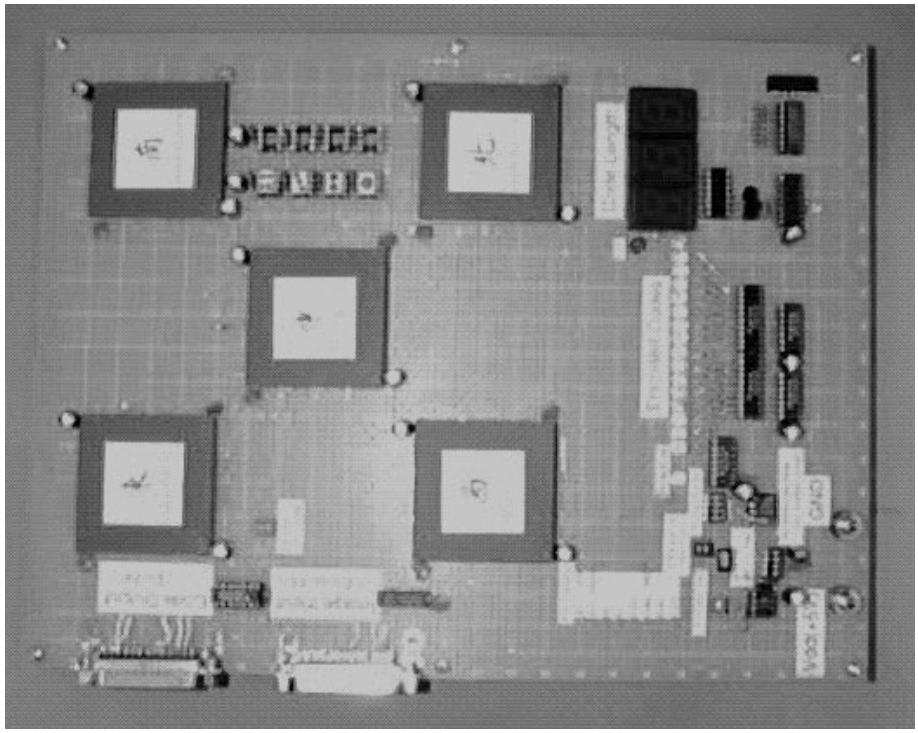


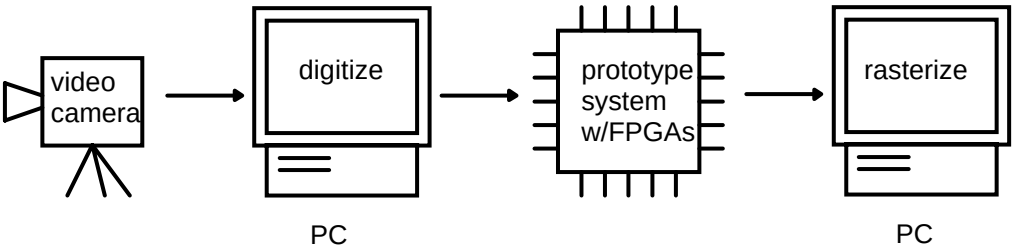
Figure 6.6: Photograph of the developed prototype system for 1:4 tree sensor.

the another PC; this PC is prepared to receive the 1:4 tree code and decode it to two dimensional images, too.

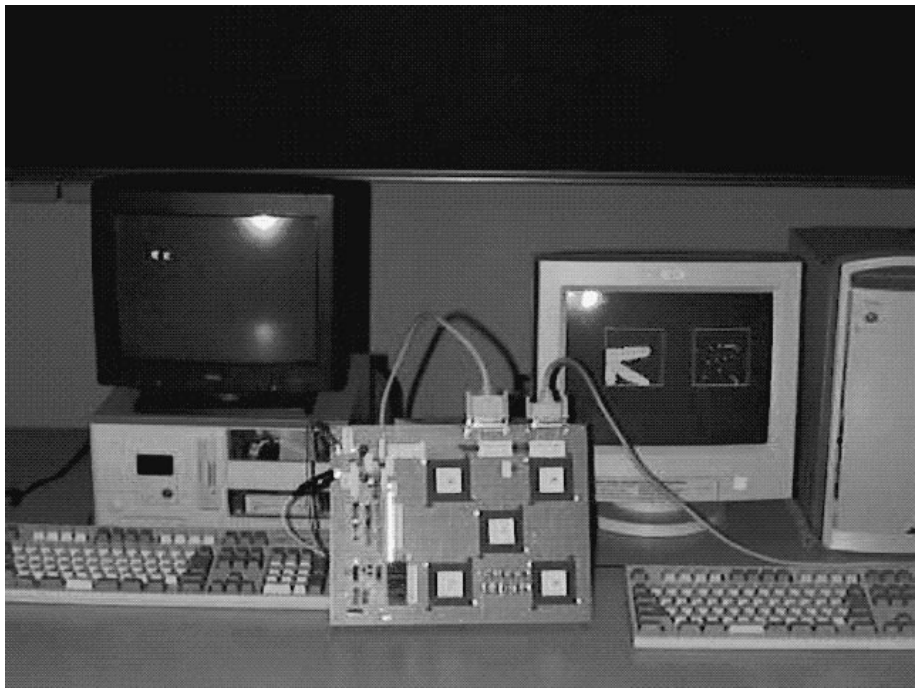
The configure of this system is described in Figure 6.7(a), with the photograph in Figure 6.7(b).

6.2.2 Evaluation of prototype system

Figure 6.9 shows the decoded images from the 1:4 tree code generated by the developed prototype system, from the image captured by video camera. The proper operations of the developed prototype system are observed, and it is also shown that 1:4 tree code length can be smaller than that of the raster scan, 1024. The operation of the developed prototype system with the clock frequency up to 1MHz is observed, which is restricted by the ability of the clock generator. The operation with the higher clock frequency is expected, since the number of inter-connection between FPGAs is very small.



(a)



(b)

Figure 6.7: Model of the developed prototype system of 1:4 tree sensor(a), and its photograph(b).

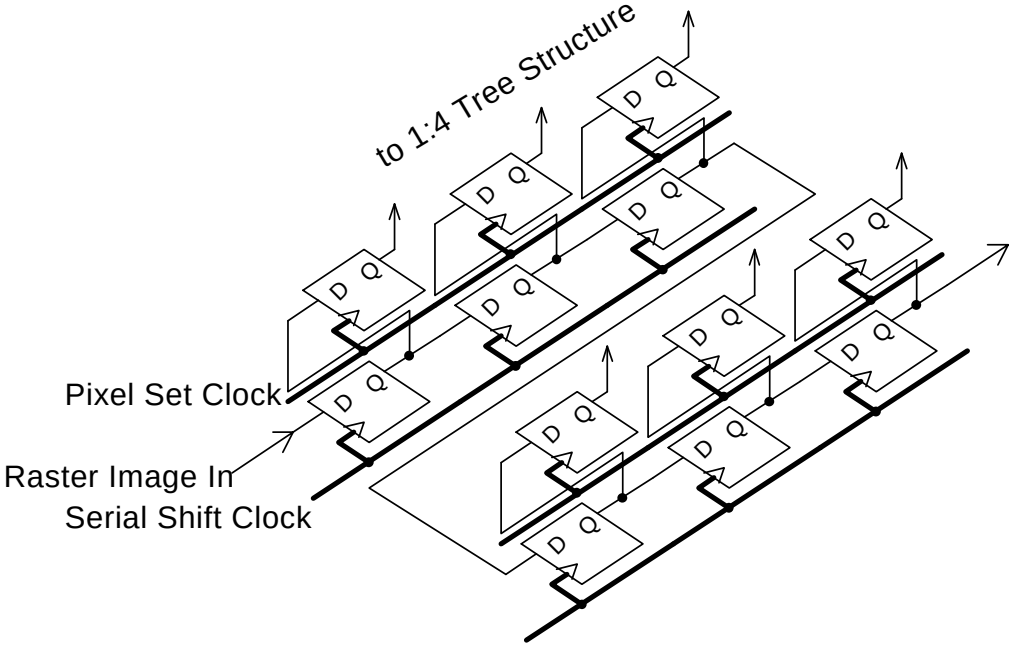
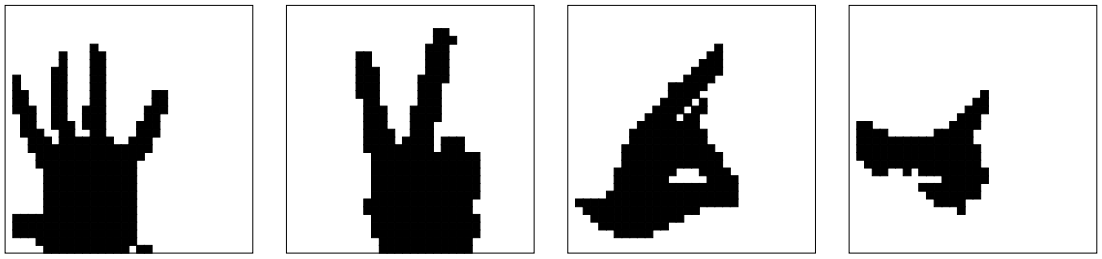
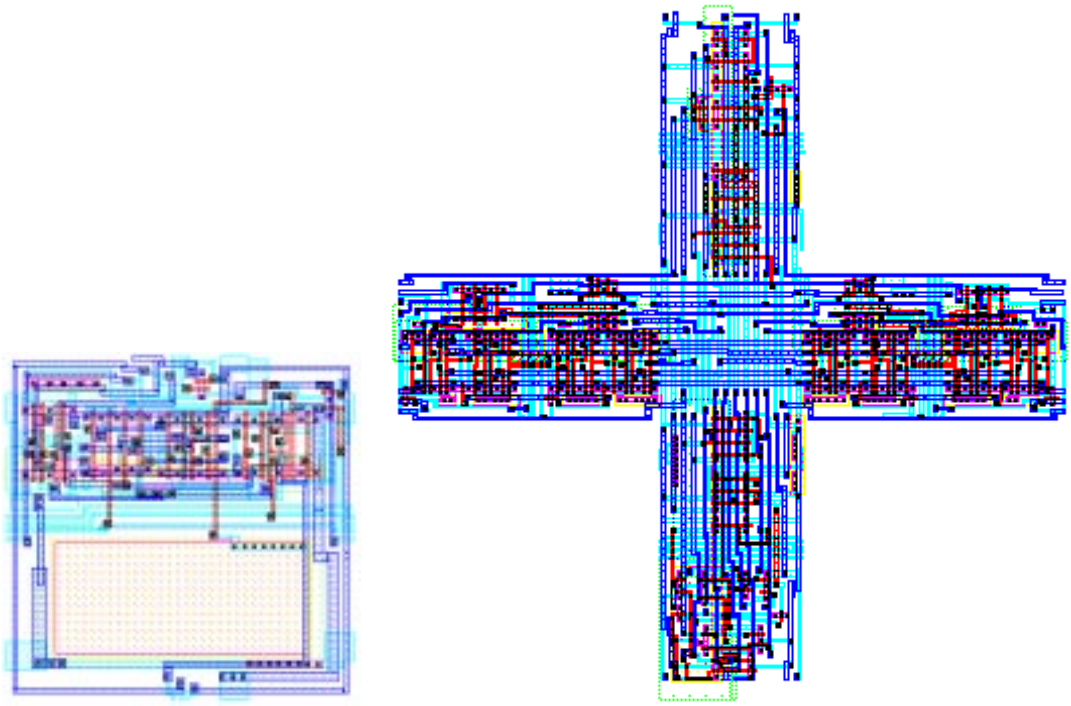


Figure 6.8: Architecture of serial shift registers to emulate the pixels



"Par"	"Peace"	"Fox"	"Good"
(573bit)	(529it)	(433it)	(293bit)
(a)	(b)	(c)	(d)

Figure 6.9: Four sample images decoded from 1:4 tree code generated by the developed prototype system. The code lengthes are also shown below.



(a) (b)

Figure 6.10: Layout of the pixel(a) and the node automata(b) in the designed 1:4 tree sensor.

6.3 Design and Evaluation of 1:4 Tree Sensor using Node Automata

6.3.1 Design of 1:4 tree sensor using node automata

The 1:4 tree image sensor with the tree structure of node automata has been designed, by using the circuits of node automata designed in Figure 5.2, with the pixel which can take inter-frame difference, as designed in Figure 5.15. The layouts of the each pixel and the each node are shown in Figure 6.10, using CMOS 1.5μm technology, and the whole layout of the chip is shown in Figure 6.11. The chip photo graph of the fabricated chip is also shown in Figure 6.12[3]. The chip size is

[3]The VLSI chip in this study has been fabricated in the chip fabrication program of VLSI Design and Education Center(VDEC), the University of Tokyo with the collaboration by Nippon Motorola and Dai Nippon Printing Corporation.

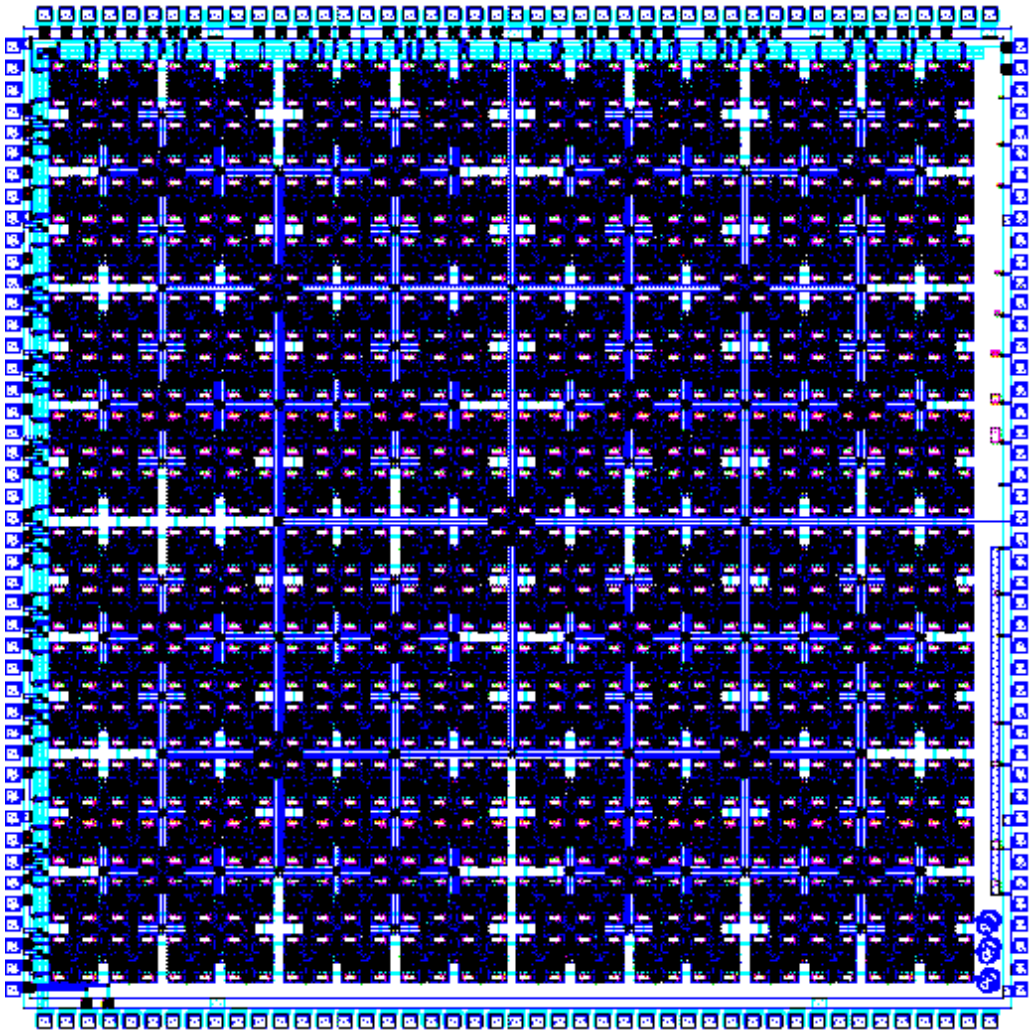


Figure 6.11: Whole layout of the designed 32×32 pixels 1:4 tree sensor.

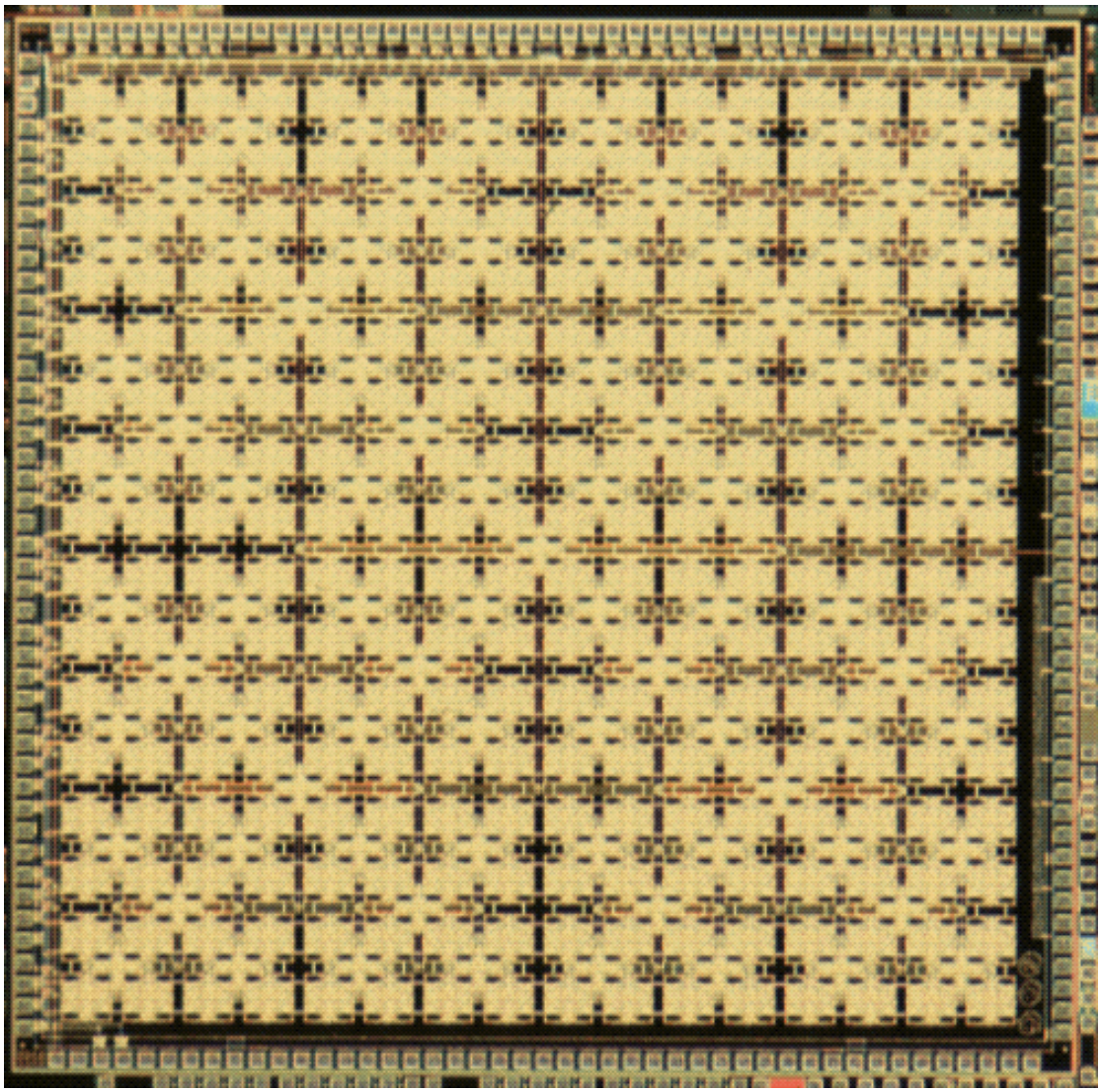


Figure 6.12: Chip photograph of the designed 32×32 pixels 1:4 tree sensor.

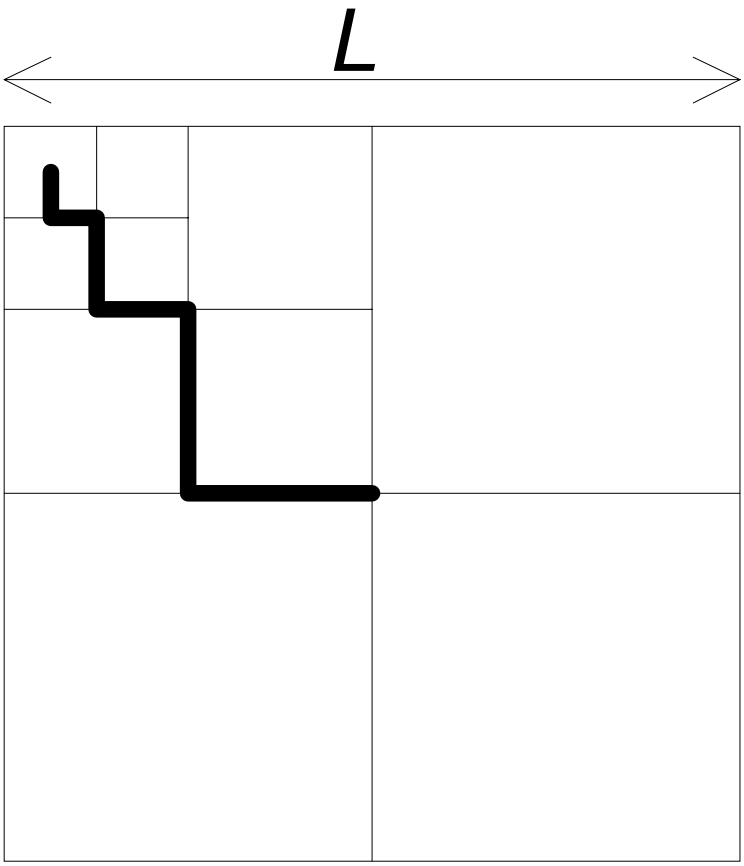


Figure 6.13: Model of scan path to from the top node to the pixel. L is the length of the edge in whole chip.

7.2mm×7.2mm, and the number of transistors 104,784, with the fill factor of about 20%.

6.3.2 Evaluation of 1:4 tree sensor using node automata

The power consumption of designed 1:4 tree sensor is estimated by the model and the simulations. In the designed node circuit, the clock signal is provided just along the scanned signal path, which is expected to be suitable to reduce power consumption.

Assuming the number of pixels is $n \times n$ and the length of edge in whole chip is L, the activated signal scan path from the top node to the pixel is shown in Figure

6.13. The length of this scan path, L_t, is derived as follows.

$$L_t = \sum_{l=1}^{N} \frac{L}{2} \times 2^{-(l-1)} \approx L \propto n \tag{6.2}$$

This scan path is activated at every step of 1:4 tree scan, and the power consumption per step by charging and discharging this scan path is expected to be proportional to n.

Assuming the number of branches as b, the number of transitions of each node through all the scan steps until finish, is equal to $(b+1)$ for the nodes whose values are "1", while no transitions occur in the nodes whose values are "0". Total number of transition of nodes in whole tree structure, N_{node} is expressed as follows,

$$N_{\text{node}} = (b+1) \sum_{l=1}^{N} b^{N-l} p_l, \tag{6.3}$$

where p_l is the probability that the node at the level l has the value of "1".

The expectation of power consumption per scan step, $\overline{N_{\text{node}}}$, is derived by dividing N_{node} by the mean code length $\overline{L}$, as follows.

$$\begin{aligned} \overline{N_{\text{node}}} &= N_{\text{node}} / \overline{L} \\ &= \frac{(b+1) \sum_{l=1}^{N} b^{N-l} p_l}{1 + \sum_{l=1}^{N} b^{N-l+1} p_l} \approx \frac{b+1}{b} \end{aligned} \tag{6.4}$$

In case of 1:4 tree, assuming $b = 4$, $\overline{N_{\text{node}}}$ is derived as $5/4$. In the designed circuit of node automaton shown in Figure 5.2, the expectation of the number of flip-flops per each transition is derived as follows.

$$\frac{1+1+2+2+2}{5} = \frac{8}{5} \tag{6.5}$$

Thus the total expectation of the number of flip-flops which make transitions in the whole 1:4 tree structure is derived as $5/4 \times 8/5 = 2$, which is constant through all the scan steps.

The total power consumption per scan step, U_{tree} is expressed the sum of the constant part, U_0^t and the part proportional to n, $U_1^t \cdot n$, as follows.

Table 6.1: Power consumption of 1:4 tree structure per scan step

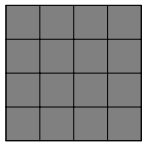
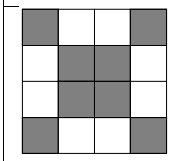
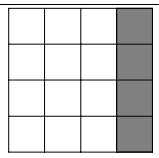
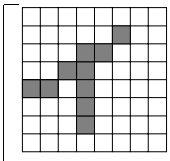
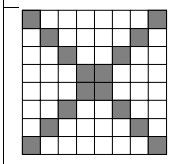
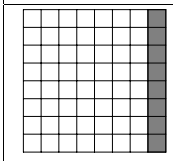
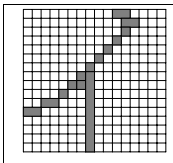
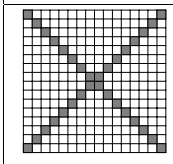
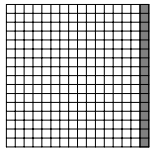
Image	# of pixels	Code length	U[pJ/bit]
	4×4	21bit	19.5
	4×4	21bit	19.8
	4×4	13bit	19.1
	8×8	41bit	24.4
	8×8	53bit	24.6
	8×8	29bit	24.2
	16×16	93bit	32.5
	16×16	117bit	34.7
	16×16	61bit	32.5

Figure 6.14: Layout of the designed address select controller.

$$U_{\text{tree}} = U_0^t + U_1^t n \tag{6.6}$$

Table 6.1 shows the calculated power consumption of 1:4 tree structure per scan step, using the simulation results of `spice` for the circuits of 1:4 tree structure consists of node automata and voltage sources as the value of pixels. The load capacitance of each node is determined to be proportional to 2^l, where l is the level with assuming the standard 1.5μm CMOS process, and the total number of levels, N is $N = 2$ for 4×4 pixels, $N = 3$ for 8×8 pixels, and $N = 4$ for 16×16 pixels, respectively.

The two part of power consumption in Equation (6.6) is derived as $U_0^t = 14.4$pJ and $U_1^t = 1.26$pJ, respectively. Seen above, it is shown that the power consumption in 1:4 tree structure is proportional to $O(n)$, while $O(n^2)$ for CCD raster scan image sensor, which is expected to be small enough for larger n.

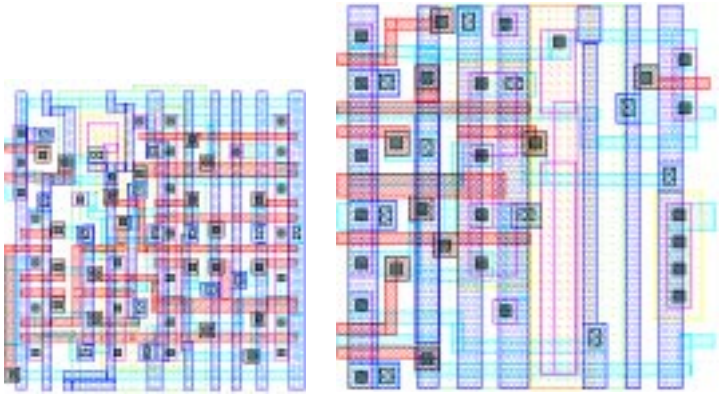
Because of the design mistake in pad assignment suitable for LSI tester, the measurement of the fabricated chip has not been carried out yet.

6.4 Design and Evaluation of 1:4 Tree Sensor using External Address Decoders

6.4.1 Design of the 1:4 tree sensor using external address decoders

The 1:4 tree image sensors using the external address decoders have been designed, by using the circuits of the controller and the address decoders designed in Figure 5.11, which is shown in Figure 6.14.

The two types of pixels are designed; the pixels with the function of taking inter-frame difference, as shown in Figure 6.15(a), and the compact pixels without taking inter-frame difference, which has only one latch to hold data, as shown in Figure



(a) (b)

Figure 6.15: Layouts of the designed pixels for the 1:4 tree sensor using external address decoders. (a)pixel with taking inter-frame difference, (b)simple, compact pixel without taking inter-frame difference.

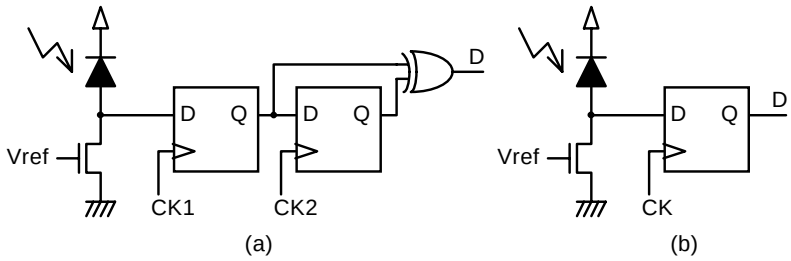


Figure 6.16: Circuits of pixels used for the 1:4 tree sensor using external address decoders. (a)pixel with taking inter-frame difference, (b)simple, compact pixel without taking inter-frame difference.

6.15(b). The circuit of each pixel is shown in Figure 6.16.

Two chips are designed using CMOS 1.5μm technology; 64×64 pixels chip with the function of taking inter-frame difference using the pixels in Figure 6.15(a), and 128×128 pixels chip without the function of taking inter-frame difference using the pixels in Figure 6.15(b). Whole layouts of the designed chip are shown in Figure 6.17 and Figure 6.18, respectively.

The chip photographs of both chips are shown in Figure 6.19 and Figure 6.20, respectively[4].

The chip size of the 64×64 chip is 4.8mm$\times$4.8mm, and the number of transistors is 120,045, with $64 \times 64 = 4,096$ pixels which have the functions of taking inter-frame differences.

The chip size of the 128×128 chip is 7.2mm$\times$7.2mm, and the number of transistors is 207,661, with $128 \times 128 = 16,384$ pixels which have the simple photo-electron conversion function without taking inter-frame differences.

6.4.2 Evaluation of the 1:4 tree sensor using external address decoders

The power consumption in the 1:4 tree architecture using external address decoders should be larger than that of the 1:4 tree sensor designed in section 6.3, since the long pixel select lines across the pixel plain is activated at scan step, while the only one signal scan path is activated in the 1:4 tree sensor as discussed in section 6.3.2.

The power consumption in the 1:4 tree sensor using external address decoders is composed of the following parts.

- power consumption of charging and discharging each bit line, B_i.
- power consumption of charging and discharging each row select line, R_j.
- power consumption of charging and discharging the word line, W.

[4]The VLSI chip in this study has been fabricated in the chip fabrication program of VLSI Design and Education Center(VDEC), the University of Tokyo with the collaboration by Nippon Motorola and Dai Nippon Printing Corporation.

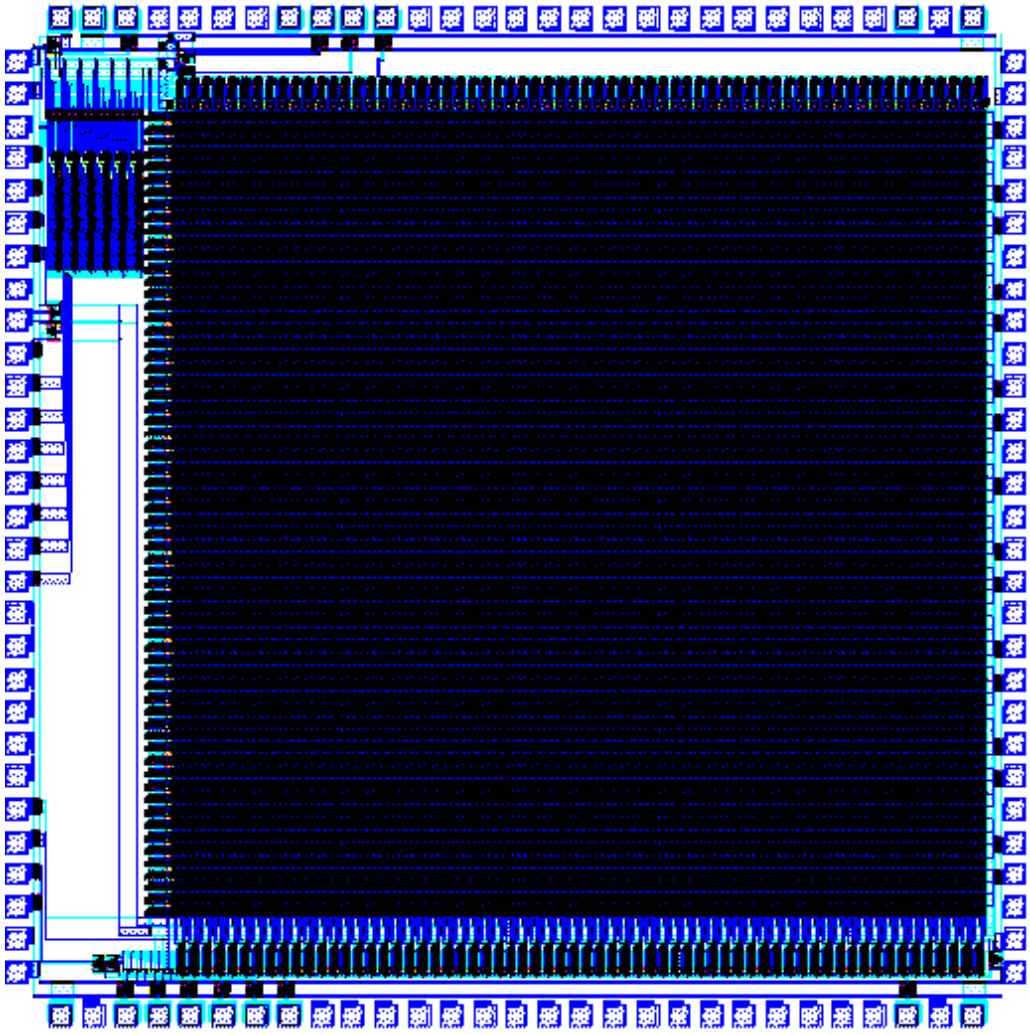


Figure 6.17: Whole layout of the designed 64×64 1:4 tree sensor using external address decoders, with taking inter-frame difference.

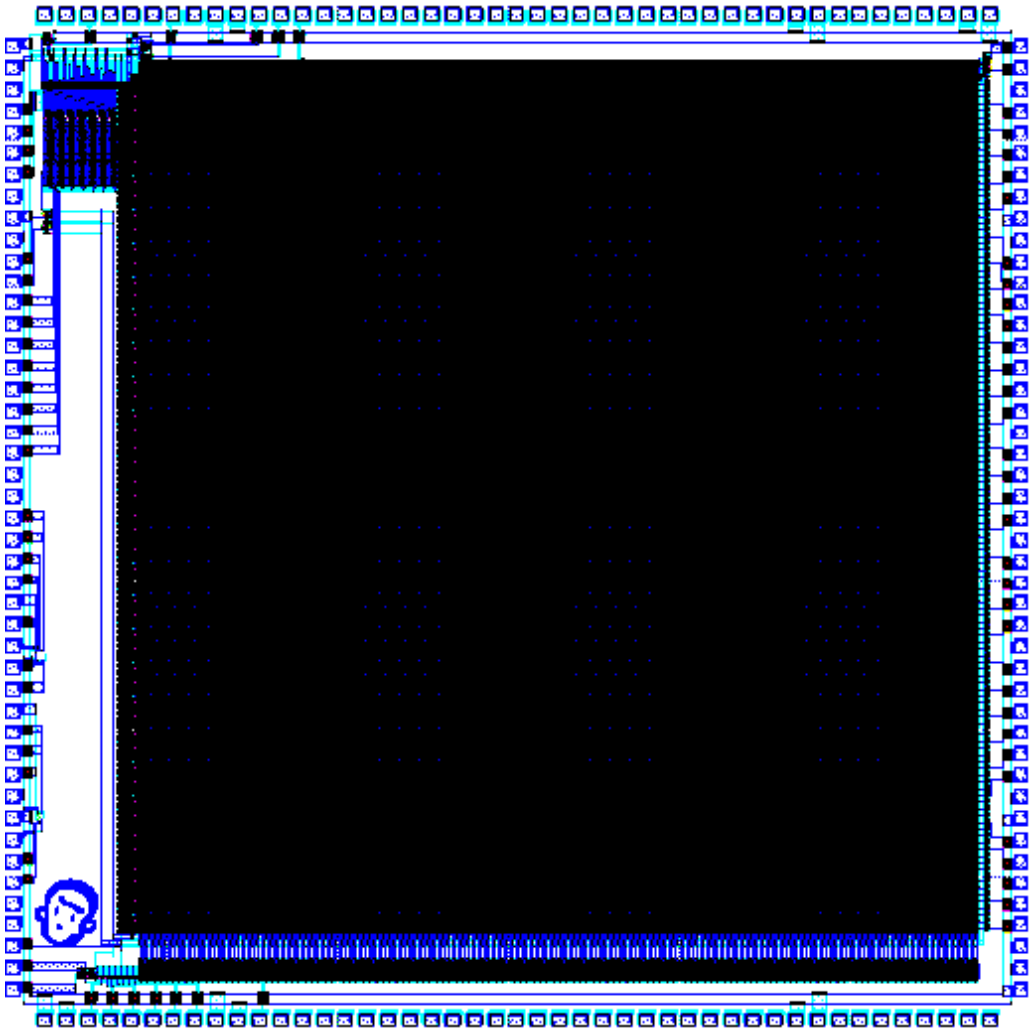


Figure 6.18: Whole layout of the designed 128 × 128 1:4 tree sensor using external address decoders.

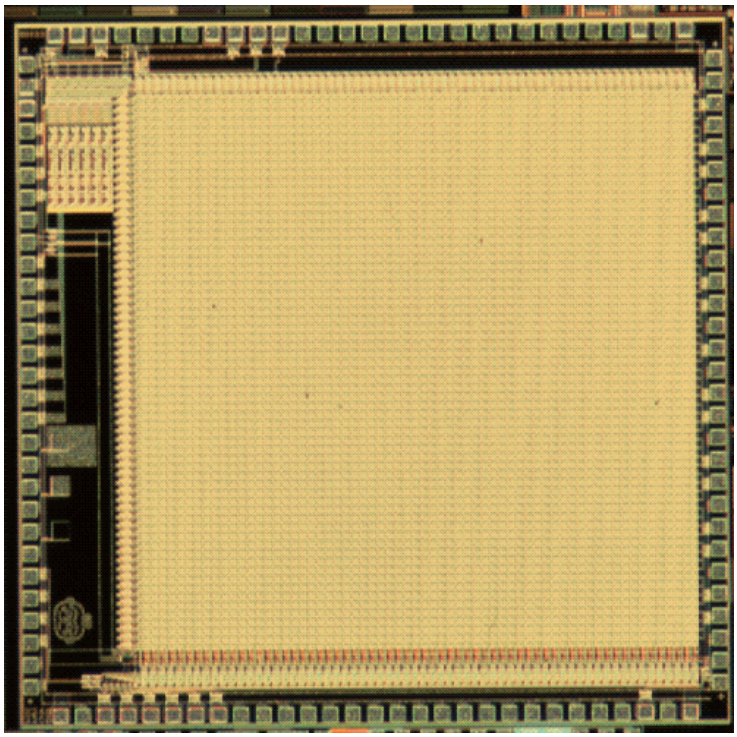


Figure 6.19: Chip photograph of the designed 64×64 1:4 tree sensor using external address decoders, with taking inter-frame difference.

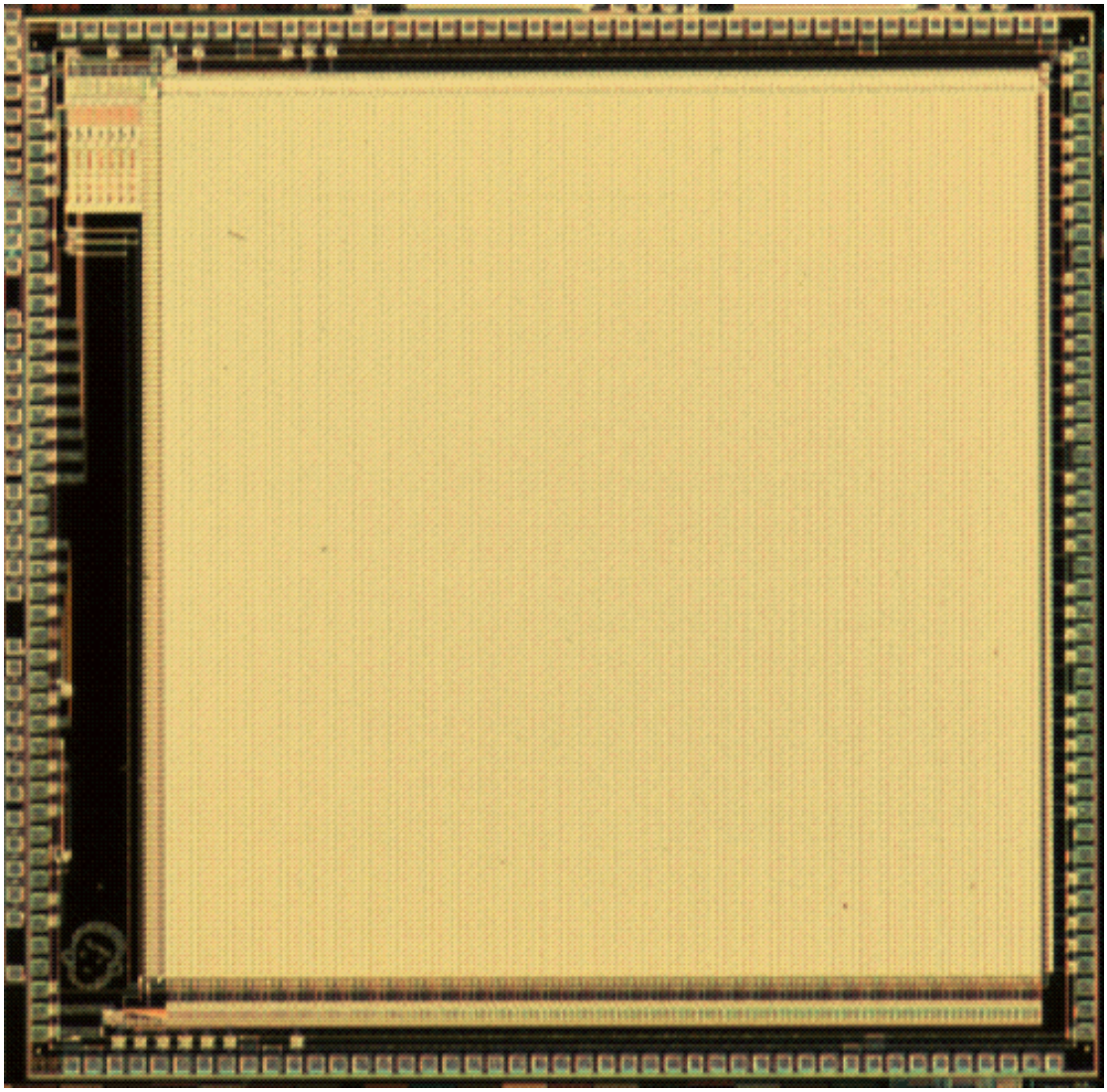


Figure 6.20: Chip photograph of the designed 128×128 1:4 tree sensor using external address decoders, without taking inter-frame difference.

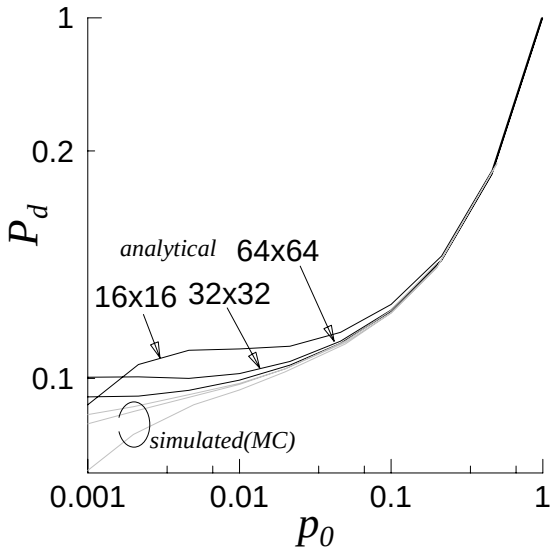


Figure 6.21: Relation of the selection probability of the bit lines or the row select lines, P_s, and the probability of the pixels being "1", p_0

- power consumption if the scan sequence controller, U_{FSM}.

The probability of each bit line B_i or row select line R_j being selected, P_s is derived as dividing the frequency of selection through all the scan steps by the number of scan steps as follows,

$$P_s = \frac{\sum_{l=1}^{N} 2^{N+1-l} p_l}{1 + \sum_{l=1}^{N-1} 4^{N+1-l} p_l}, \tag{6.7}$$

where p_l is the probability of the node at the level l being "1" derived in Equation (3.2) as $p_l = 1 - (1 - p_0)^{4^l}$, where p_0 is the probability of the pixel being "1". The relation of P_s and p_0 is shown in Figure 6.21, with the results of the Monte-Carlo switch level simulation, for the various size of pixel plain, 16×16, 32×32, and 64×64. The relative frequency of the bit lines or row select lines being selected is lower for the larger p_0, since the number of scan steps increases rapidly.

It is also derived the probability of the bit in 1:4 tree code being "1", P_d as dividing the total number of the nodes whose value is "1" by the number of scan steps as follows.

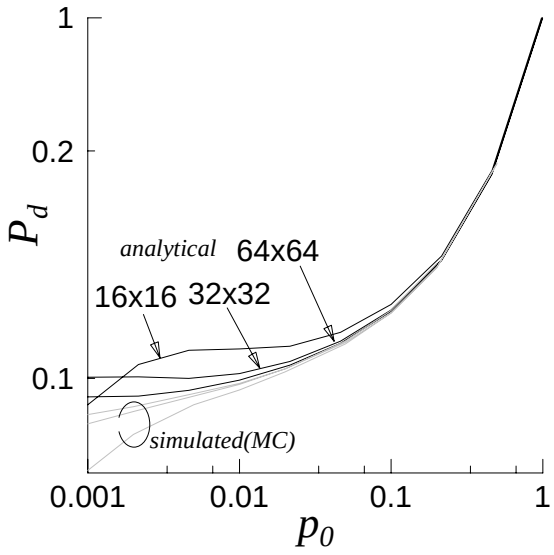


Figure 6.22: Relation of the probability of the bit in 1:4 tree code being "1" and the probability of the pixels being "1", p_0

$$P_d = \frac{\sum_{l=0}^{N} b^{N-l} p_l}{1 + \sum_{l=1}^{N-1} b^{N+1-l} p_l} \tag{6.8}$$

The relation of P_d and p_0 is shown in Figure 6.22, with the results of the Monte-Carlo switch level simulation, for the various size of pixel plain, 16×16, 32×32, and 64×64. The probability of the bit in 1:4 tree code being "1" is smaller for the smaller p_0.

Using these probabilities, the probabilities of charging or discharging signal lines are derived as follows.

- The bit line B_i is discharged when it is selected and at least one pixel at the selected rows is "1", and the probability is expressed as $P_s\{1 - (1 - p_0)^{nP_s}\}$, since nP_s is the expectation of selected rows.

- The bit line B_i is charged when it is discharged at the previous scan step, and it is also selected at the current scan step, and the probability is expresses as $P_s \cdot P_s\{1 - (1 - p_0)^{nP_s}\}$.

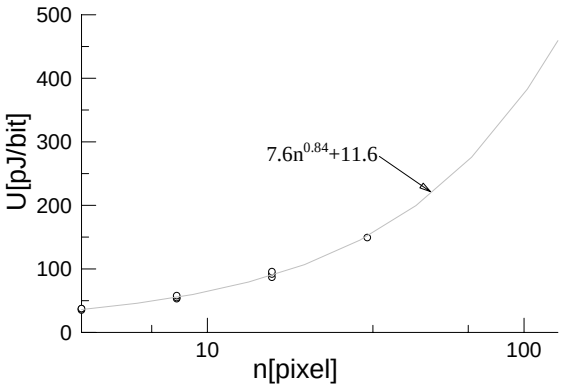


Figure 6.23: Relation of power consumption per scan step and the number of pixels at one edge, for various images

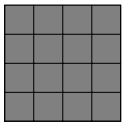
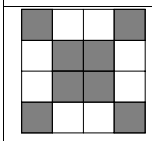
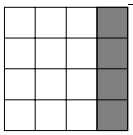
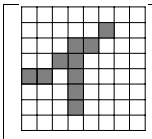
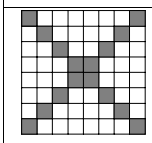
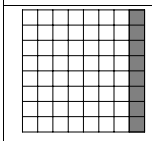
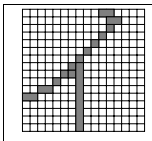
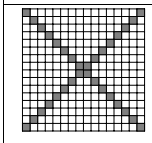
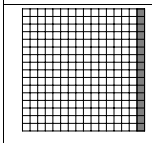
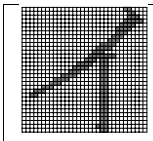
- The row select line R_j is charged and discharged by the probability of selection, P_s at the every scan step.
- The word line W is discharged when the bit of 1:4 tree code is "1", and the probability is expressed as P_d.
- The word line W is charged at the pre-charge cycle if it is discharged at the previous scan step, and the probability is expressed as P_d.

The load capacitance of the bit lines and the row select lines and the word line are proportional to their lengthes, and they are proportional to the number of pixels at one edge, n. Thus the expectation of power consumption at each scan step, P is expressed as the sum of each power consumption as follows.

$$P = n \times \frac{P_s(1+P_s)\{1-(1-p_0)nP_s\}}{2} nC_{B0}V_{dd}^2 + nP_s \times nC_{R0}V_{dd}^2 + P_d \times nC_{W0}V_{dd}^2 + U_{\mathrm{FSM}} \tag{6.9}$$

The power consumption for some sample images is calculated by `spice` simulation, since the active probability of pixels have spatial correlation in these cases, which can not be derived by the Equation (6.9), that is derived by assuming no correlation among pixels. The load capacitance of bit lines, row select lines, and word line is assumed to be proportional to the number of pixels at one edge using the standard

Table 6.2: Power consumption of 1:4 tree structure using external address decoders per scan step

Image	# of pixels	Code length	U[pJ/bit]
	4×4	21bit	38.5
	4×4	21bit	40.5
	4×4	13bit	43.0
	8×8	41bit	55.1
	8×8	53bit	57.5
	8×8	29bit	62.6
	16×16	93bit	89.5
	16×16	117bit	95.5
	16×16	61bit	101.1
	32×32	317bit	150.3

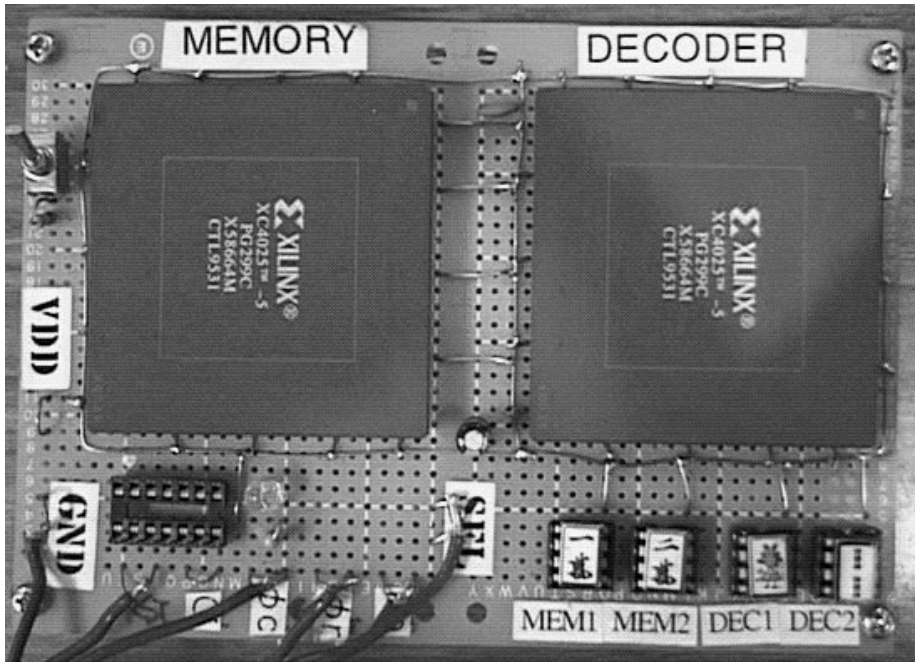


Figure 6.24: Photograph of the developed prototype system for 1:4 tree code decoder.

1.5μm CMOS process. The simulation results are shown in Table 6.2. The relation of power consumption per scan step and the number of pixels at one edge is shown in Figure 6.23, with the curve-fitting results using least mean square (LMS) method. It is expected to be proportional to $O(n^{0.84})$, which is expected to be small enough than that of CCD image sensors for the larger n.

Because of the design mistake in pad assignment suitable for LSI tester, the measurement of the fabricated chip has not been carried out yet.

6.5 Design and Evaluation of Tree-Code Decoder

6.5.1 Prototype implementation of tree-code decoder

We have developed the prototype system of the 1:4 tree code decoder using FPGAs(Field Programmable Gate Array). We have employed the FPGA of XC4025, which is used in section 6.2.1, and one for memory cell arrays, and the other for address decoders and controllers. The number of memory cells is $32 \times 32 = 1,024$, and the decoded image is serially read out.

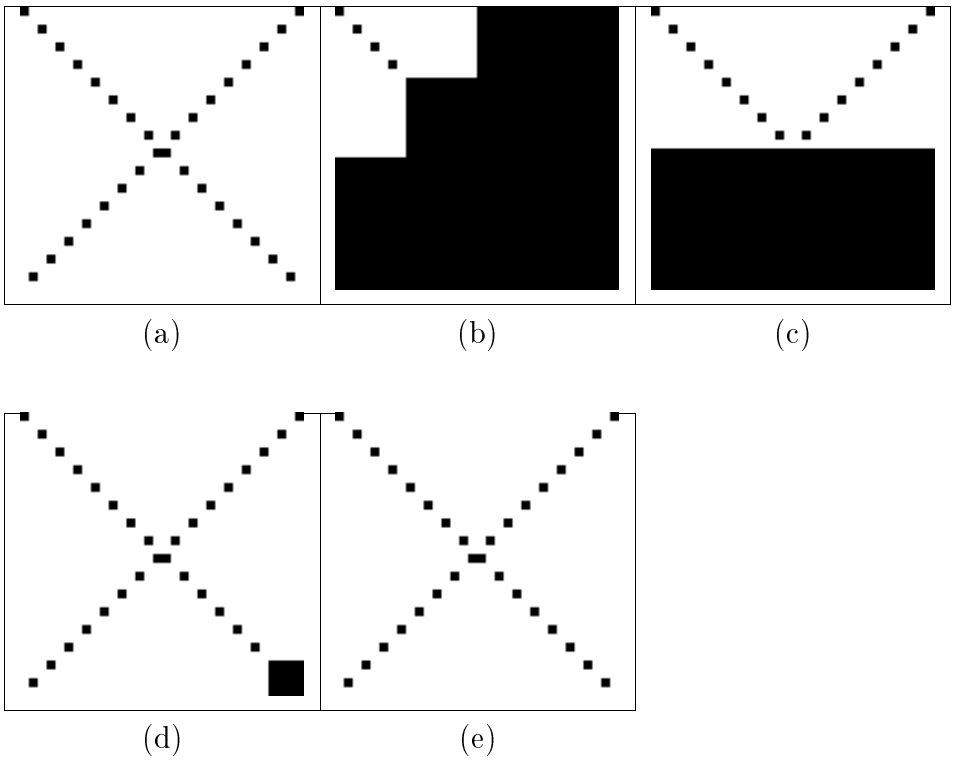


Figure 6.25: Original image(a) and the images under decode(b), (c), (d), and the completely decoded image(e).

The photograph of the prototype 1:4 tree code decoder is shown in Figure 6.24.

The 1:4 tree code to be decoded is given by PC, with the system clocks, and it is decoded to binary image by the developed prototype system. The binary image is again transferred to PC serially. The image under decode procedure can be also read out serially, since each memory cell is set its value according to the decoding step, which is more detail at the decode step for lower levels. Figure 6.25 shows the original image and the images under decode, and the completely decoded image.

The operation frequency of the developed prototype system is up to 1MHz, which is restricted by the speed of PC.

6.5.2 Design of full-custom tree-code decoder

The decoder of 1:4 tree code has been designed, by using the circuits of the controller and the address decoders, which are identical to those used in the 1:4

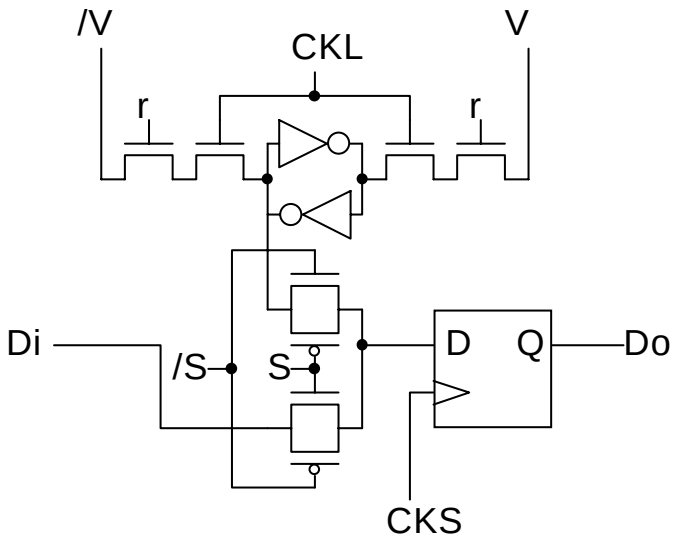


Figure 6.26: Circuit of memory cell for the 1:4 tree code decoder

tree sensor using external address decoder, used in section 6.4. The circuit of the memory cell is shown in Figure 6.26. The memory cell has coupled inverters to store the value of raster image, and one D-flip-flop for the shift register in order to shift out outside the decoded image. The decoded image is given by `V` and its complementary as `/V`, and they are stored to the coupled inverters using clock `CKL`, if the row of this pixel is selected by `r`. The selection of columns is done by the signal generation circuit placed at each column, as shown in Figure 5.14. The stored image is transferred to D-flip-flop at the positive edge of `CKS` by keeping `S` as low and `/S` as high. The transferred image is shifted out outside by the clock of `CKS` with keeping `S` as high and `/S` as low. Data input `Di` is connected to the data output of the previous stage in shift register, and data output `Do` is connected to the data input of the next stage in shift register, respectively. Data output `Do` of the final state in shift register is the serial output of the decoded image.

The layout of the memory cell used in this decoder is shown in Figure 6.27, which is implemented by the circuit described in section 5.3.

The whole layout of the designed 64×64 decoder of 1:4 tree code is shown in Figure 6.28, which has the functions of read out the rasterized image serially, and can be cascaded for the decoding of the larger raster images, as shown in Figure

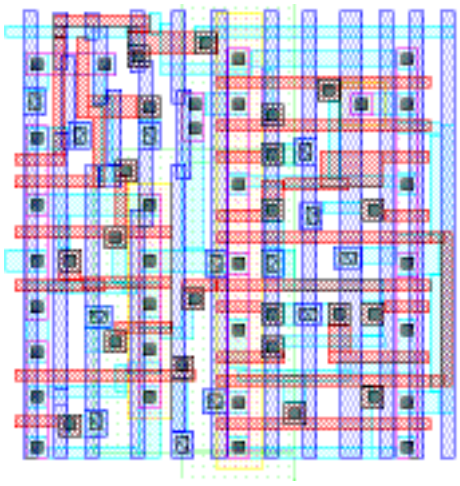


Figure 6.27: Layout of the memory cell for the decoder of 1:4 tree code.

6.29. In this case, one decoder chip is used as master, and the address select signals are send to the other slave chips. The other tree decoder chips are used as slave, where only the address decoders are used to select memory cells. One level of the controller in one of the slave decode chips is also used to generate the address select signals for the extended number of levels. The serial inputs and serial outputs of the decoded images in each chip is connected in order, so as to form a large shift register.

The chip photograph of the designed decoder of 1:4 tree code is shown in Figure 6.30[5]. The number of transistors is 110,024.

There was a fatal design mistake in the fabricated chip, one connection in serial shift registers is lost, and the measure and evaluation of this chip has not been done yet.

6.6 Summary and Conclusion

In this chapter, the designed chips of the 1:4 tree sensors and the decoder of 1:4 tree code and their prototype systems using FPGAs, are described. It is shown

[5]The VLSI chip in this study has been fabricated in the chip fabrication program of VLSI Design and Education Center(VDEC), the University of Tokyo with the collaboration by Nippon Motorola and Dai Nippon Printing Corporation.

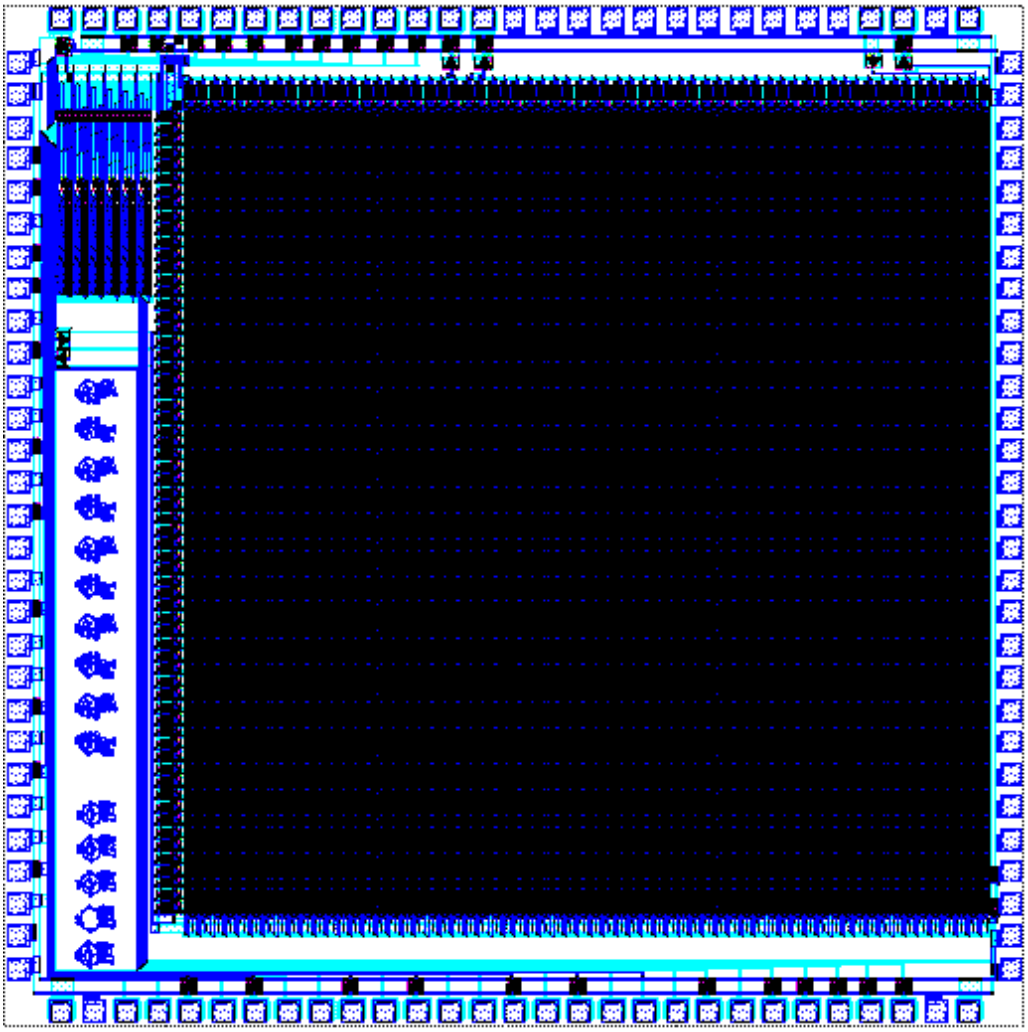


Figure 6.28: Whole layout of the designed 64×64 decoder of 1:4 tree code.

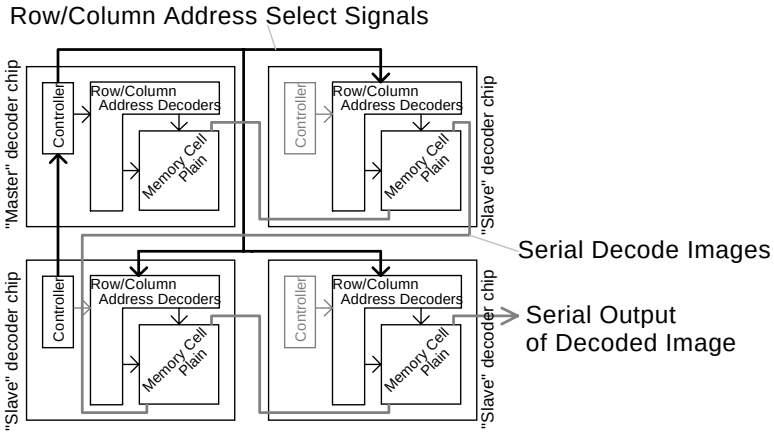


Figure 6.29: Extend connection of designed decoder chips for the larger image

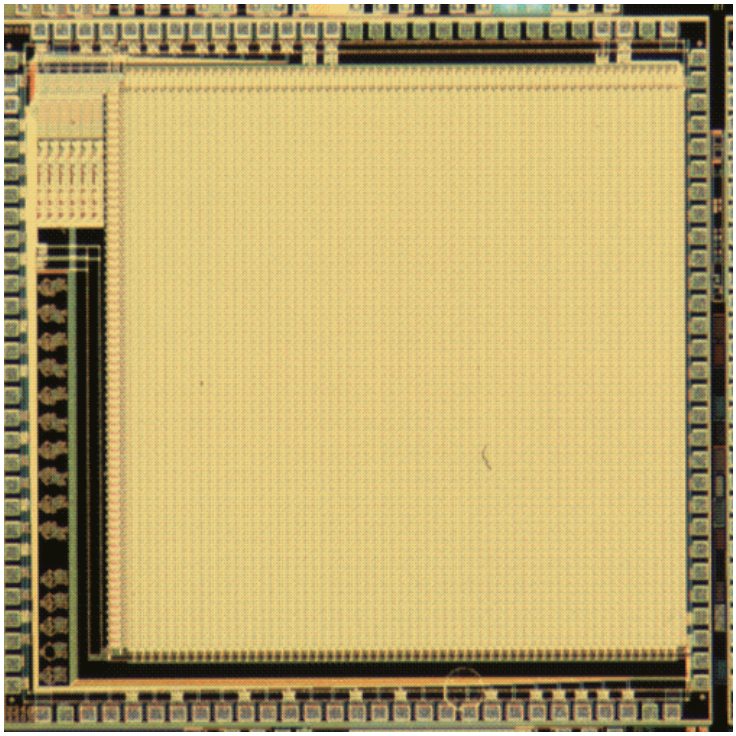


Figure 6.30: Chip photograph of the designed 64×64 decoder of 1:4 tree code.

that the 1:4 tree sensor using the external address decoder can integrate about 16 times pixels in the same size of chip, compared with that using tree structure of automata. It is also shown that the decoder of 1:4 tree code can be implemented by using the external address decoder designed for the 1:4 tree sensor with external address decoders.

The evaluations of them, mainly in terms of the power estimation are also described.