Chapter 5

Implementation of Image Sensors with 1:4 Tree Structure

In this chapter, the implementation of the 1:4 tree structure discussed in chapter 3 will be discussed.

It will be discussed the implementation of the 1:4 tree structure in CMOS circuits, mainly aiming the low power operation by activating the target area selectively in section 5.1. In the following section 5.2 will describe the alternative implementation of 1:4 tree sensor by placing the pixel select circuits external of the pixel plain. It will be discussed the architecture and its circuits for the decoder of 1:4 tree code in section 5.3.

In the section 5.4, the functions and the circuits for each pixel will be discussed, aiming the intelligent image sensor. It will be discussed the concept of the image sensor driven by the received photo energy for the ultra low power operation in th following section 5.5.

5.1 Image Sensor with Tree Structure of Node Automata

5.1.1 Circuit of node automaton

In order to implement the function of nodes discussed in section 3.1, the node is described as the finite state machine whose state transition diagram is shown as

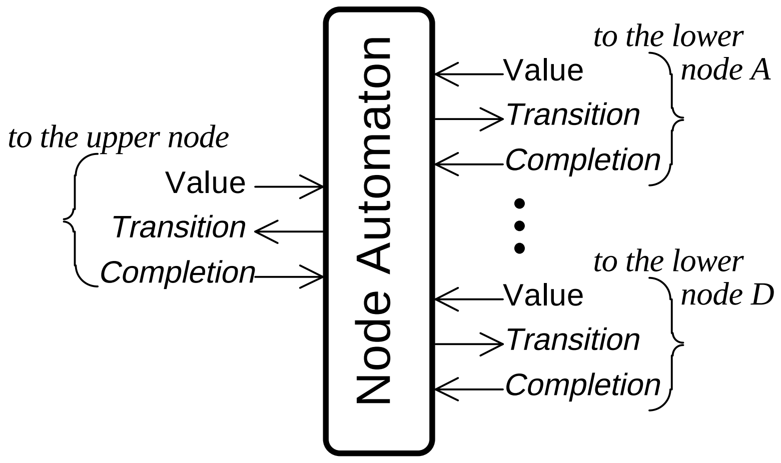

Figure 5.1: Signals of the nodes in the tree structure.

Table 5.1: State transition diagram for the node automata of 1:4 tree structure.

Inputs								Outputs					
T	V	CA	CB	CC	CD	S	S'	TA	TB	TC	TD	V	C
0	–	–	–	–	–	W	W	0	0	0	0	VO	0
↑	0	–	–	–	–	W	W	0	0	0	0	VO	1
↑	1	–	–	–	–	W	A	↑	0	0	0	VA	0
↑	–	0	–	–	–	A	A	↑	0	0	0	VA	0
↑	–	1	–	–	–	A	B	0	↑	0	0	VB	0
↑	–	–	0	–	–	B	B	0	↑	0	0	VB	0
↑	–	–	1	–	–	B	C	0	0	↑	0	VC	0
↑	–	–	–	0	–	C	C	0	0	↑	0	VC	0
↑	–	–	–	1	–	C	D	0	0	0	↑	VD	0
↑	–	–	–	–	0	D	D	0	0	0	↑	VD	0
↑	–	–	–	–	1	D	W	0	0	0	0	VO	1

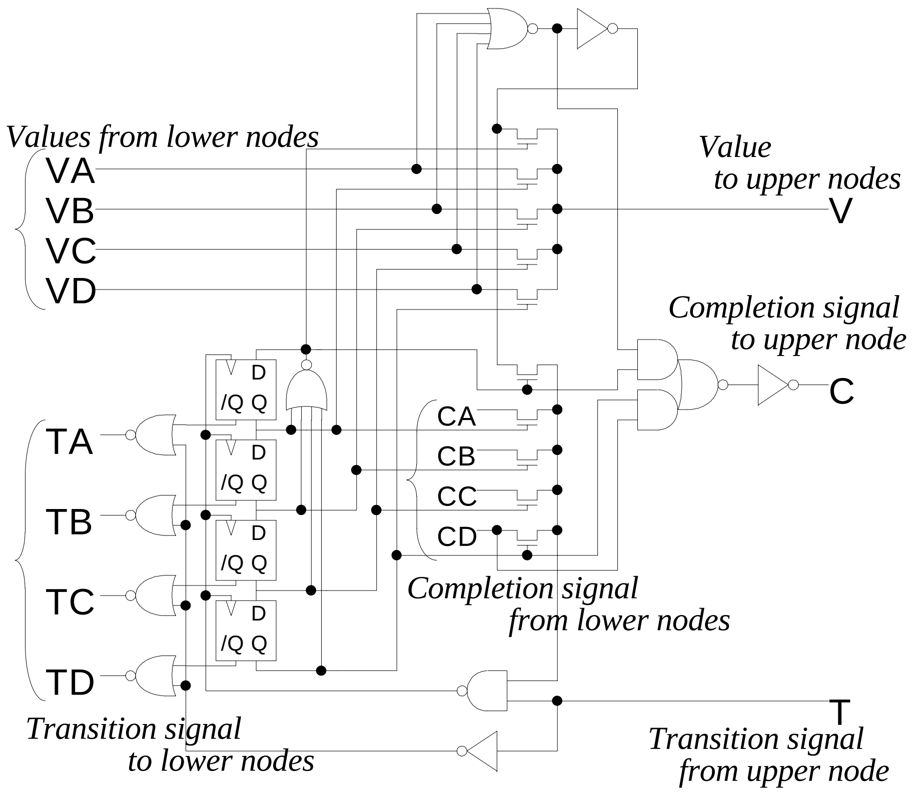

Figure 5.2: Circuit of node automaton.

Table 5.1.

Here the state `W` represents *waiting* states, and the state `SA`, ..., `SD` are the states of *scanning* lower nodes `A`, ..., `D`, respectively.

`T` is the *transition* signal to the automaton, and `TA`, ..., `TD` are the *transition* signals to its lower nodes `A`, ..., `D`, respectively. `CA`, ..., `CD` are the *completion* signals of the scan from the lower nodes `A`, ..., `D`, respectively. `VO` is the logical-OR of `VA`, ..., `VD`, and `V` and `E` is the output value and *completion* signal of automaton, respectively. These signals are summerized in Fig.5.1.

The designed circuit of the node automaton is shown in Figure 5.2. The tree structure containing a number of the conventional sequential circuits will consume much redundant power especially in clock line, since the most of the node automata have no transition except the nodes along the scan path. In the designed circuit in Figure 5.2, the clock signals for flip-flops are provided only in case of the flip-flops have made state transition, in order to reduce the redundant power consumption.

The integration of photo detectors and signal processing circuits in computational sensors, will result in the reduction of the ratio of the photo detectors' area to the pixels' area, which is called "fill factor." This is a main reason why the functions on computational sensor are restricted to the simple signal processing, such as early vision problems. The number of transistors in many studies on computational sensors, will keep less than about 50, to maintain a large fill factor[5, 7, 8].

The number of transistors of the circuit in Figure 5.2 is 142. In case of the 1:4 tree structure, the node automata will be shared by the four sub-areas, and the effective number of transistors per pixel will be smaller than this number. Assuming the number of branches as b and the number of levels as N, the total number of pixels, n_{pixel} and the nodes, n_{node} are derived as follows.

$$n_{\text{pixel}} = b^N \tag{5.1}$$

$$n_{\text{node}} = \sum_{l=1}^{N} b^{l-1} = \frac{b^N - 1}{b-1} = \frac{n_{\text{pixel}} - 1}{b-1} \approx \frac{n_{\text{pixel}}}{b-1} \tag{5.2}$$

Thus the effective number of the transistors of node automata in 1:4 tree per pixel is derived as $142 \div (4-1) \approx 47$, which is expected to be reasonable to keep the large fill factor.

It is also notable that the signal pathes for values in the circuits of Figure 5.2 are implemented by the transfer gates, which will be easily extended for the analog signal pathes.

5.1.2 Layout of node automata

The node automata and pixels should be placed in the two dimensional focal plain, since the 1:4 tree image sensor are implemented using the conventional CMOS technology. A possible layout is shown in Figure 5.3. The pixels are placed with the equal intervals, and the nodes at the lowest level is placed at the center of the four neighbor pixels, and the nodes for the upper nodes are also placed at the center of the four sub-areas. This layout can be implemented for any number of levels, by creating the magnified self-similar cross-shaped node automata.

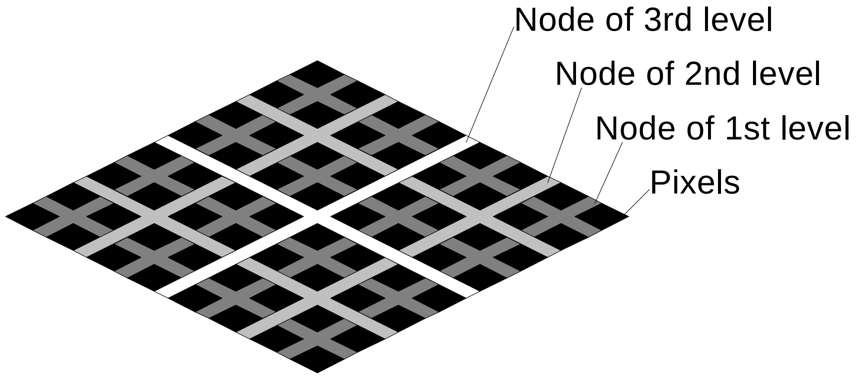

Figure 5.3: Possible two dimensional layout of the 1:4 tree structure.

As shown in Figure 5.3, since the upper nodes can occupy the larger areas, the upper nodes can have the larger signal drivers to drive the longer signal path, which is reasonable in order to minize the delays[27].

5.2 Alternative Implementation of Tree-Structured Image Sensor

5.2.1 Alternative architecture of tree sensor

The direct implementation of 1:4 tree sensor using the tree structure of node automata cannot avoid the problem of low fill factor intrinsically.

Here we propose the alternative architecture to implement the 1:4 tree scan, using the decoder architecture discussed in section 3.4. In each step of decode, the pixels are selected according to the selected sub-areas, for the values to be stored, while the scan will be executed for such selected sub-areas by reading the logical-OR of in the sub-areas alternatively. For example, the 4×4 binary image shown in Figure 5.4(a), the 1:4 tree scan proceeds by the following steps. (Here the number at the top side and the left side of pixel plain in Figure 5.4 is the column and row select lines, respectively, and the pixel whose column and row select lines are both 1, is selected. The logical-OR of selected pixels is shown at the upper-left corner of the pixel plain.)

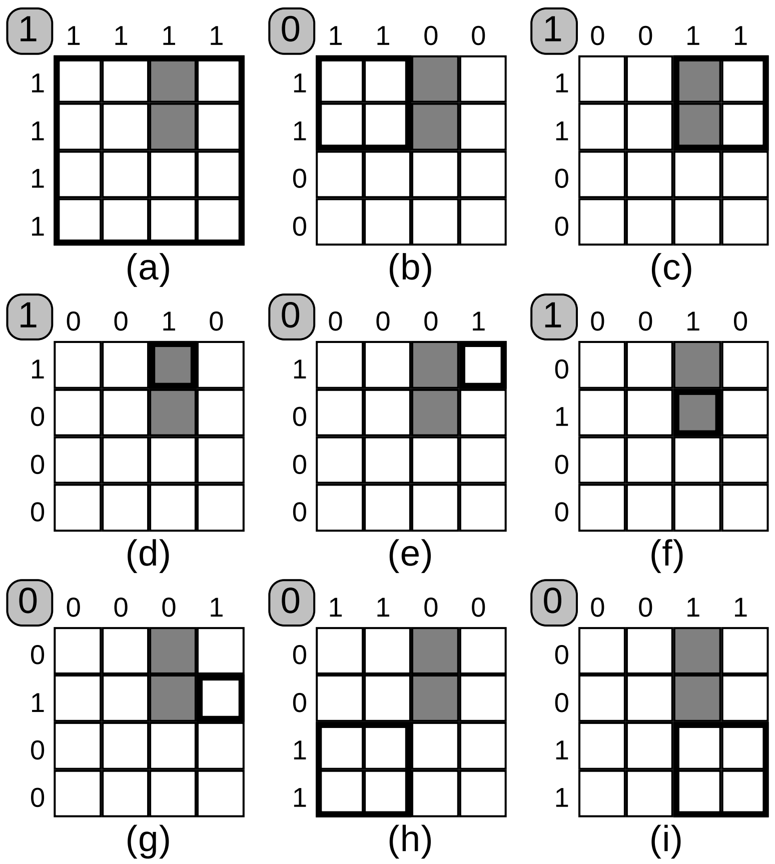

Figure 5.4: Example of 1:4 tree scan step using external address selectors

1. First, the logical-OR of whole pixels should be read out. The whole pixels are selected by the row and column address lines, and the logical-OR of the all selected pixels is read out, as shown in Figure 5.4(a).

2. Next, the scan proceeds to the upper-left 2×2 sub-area, and the pixels in this sub-area are selected, as shown in Figure 5.4(b). The logical-OR of these pixels is 0, and no more detail scan is needed for this sub-area.

3. The scan proceeds to the upper-right 2×2 sub-area as shown in Figure 5.4(c). Since the logical-OR of these pixels is 1, the more detail scan for each 1×1 sub-area (or pixel) is proceeded in order, as shown in Figure 5.4(d), (e), (f), and (g).

4. The following two steps of scan return 0 as the logical-OR of lower-left and lower-right 2×2 sub-areas, respectively, and now the all scan step has finished.

The logical-OR of the selected pixels can be made by the circuit as shown in Figure 5.5, where `PRB` and `PRW` are the signal of precharging each bit line of one column and word line, respectively, and ϕ is the clock of output latch for `v`. `r0` and `r1` are row select lines which indicate the selected rows in pixel plain, and `c0` and `c1` are the column select lines.

The scanning procedures at each step are follows, as shown in Figure 5.6.

1. precharge the each bit line and word line by keeping `PRB` and `PRW` as low.

2. set `r0`,`r1`, `c0` and `c1` according to the state of selection.

3. make `PRB` as high, and the each bit line will be discharged if there are at least values of pixels, `v` of “1” at the selected row.

4. make `PRW` as high, and the word line will be discharged if there are at least one discharged bit line in the selected column.

5. latch the value of word line by the latch clock ϕ.

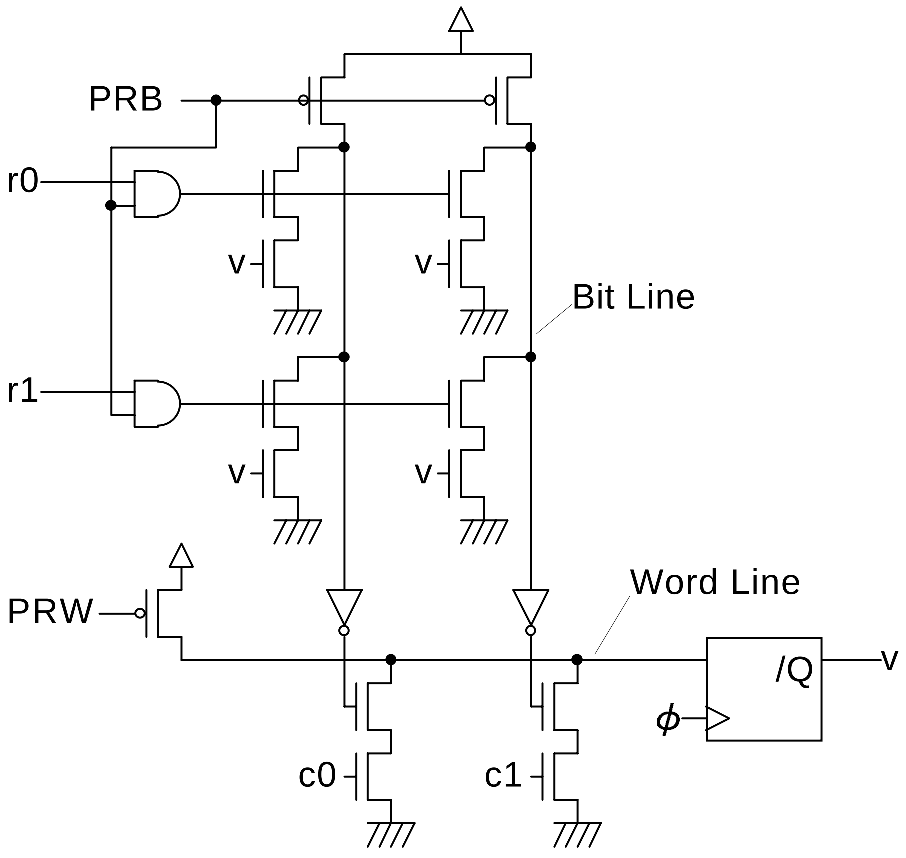

Figure 5.5: Architecture of pixel plains to make the logical-OR of the selected pixels.

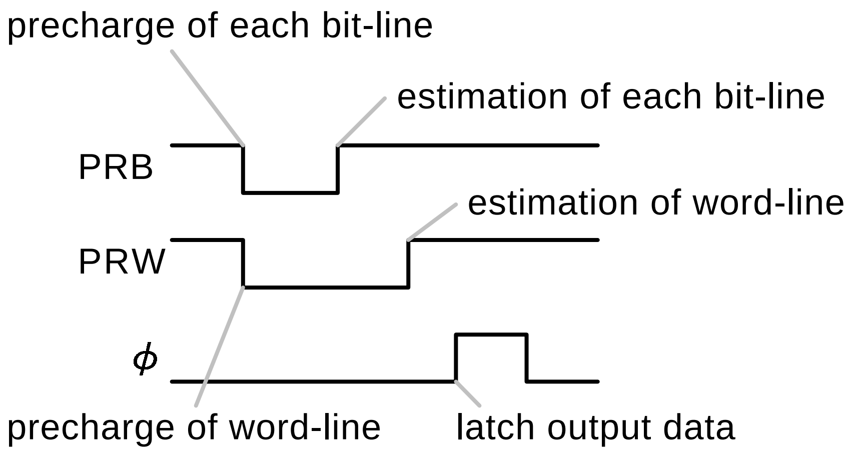

Figure 5.6: Timing chart of scan procedure of each step

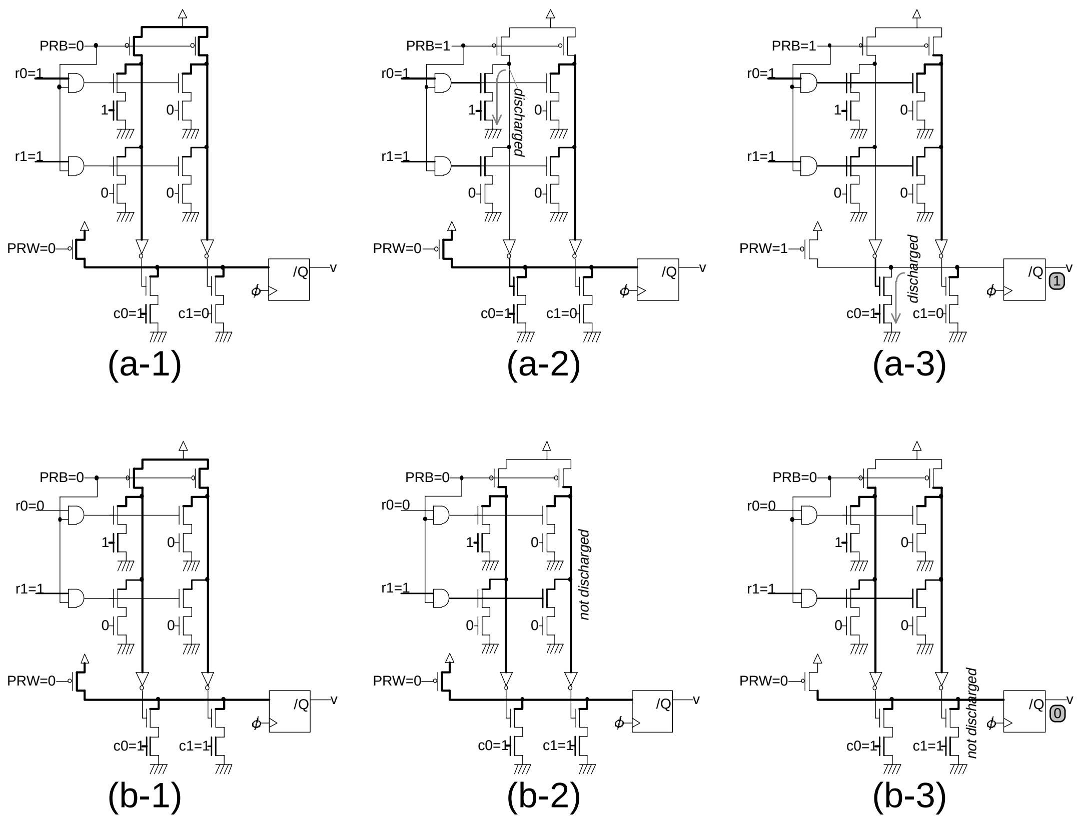

Figure 5.7: An example of scan steps using the pixel plain circuit in Figure 5.5. (The bold lines are charged lines)

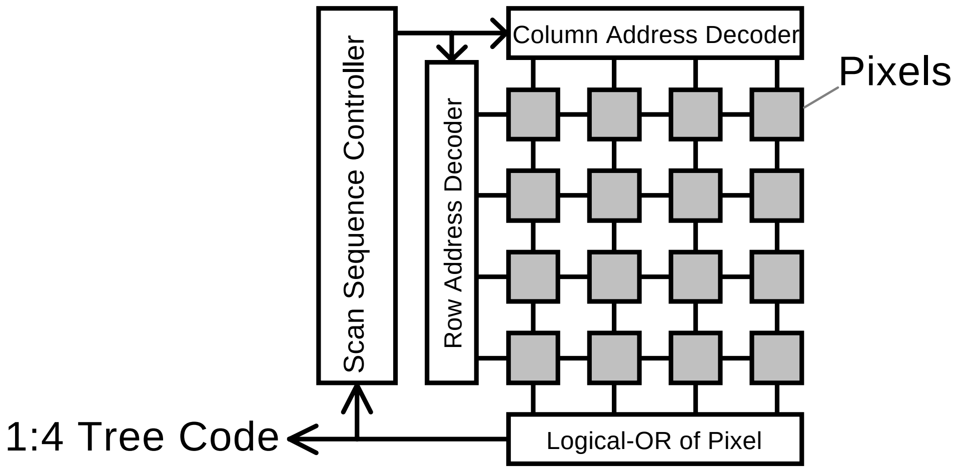

Figure 5.8: Architecture of 1:4 tree sensor using external address decoders

An example of scan steps with the state of each signal line are shown in Figure 5.7. In case of Figure 5.7(a-1), ...(a-3), the left half of four pixels are selected by (`r0`,`r1`)=(1,1) and (`c0`,`c1`)=(1,0). The bit lines are precharged in Figure 5.7(a-1), but the left bit line is discharged since upper-left pixel whose value is “1” is selected in Figure 5.7(a-2). Since the left bit line is selected by `c0`, the word line is discharged in Figure 5.7(a-3), and logical-OR of selected pixels is read out as “1.”

In the other case in Figure 5.7(b-1), ...(b-3), the lower half of four pixels are selected by (`r0`,`r1`)=(0,1) and (`c0`,`c1`)=(1,1). The bit lines are precharged in Figure 5.7(b-1), and they are not discharged, since there are no discharge pathes from each bit line to ground, as shown in Figure 5.7(b-2). The word line is not also discharged since there are no discharged bit lines in selection, and the logical-OR of selected pixels is read out as “0.”

The whole architecture of implementing these procedures is shown in Figure 5.8. The “scan sequence controller” makes selection signals of pixels based on the logical-OR of all the selected pixels, and “row and column address decoders” make address selection signals for pixels along to the currently selected sub-areas.

The problem of low fill factor because of the placed node automata in pixel plain will be solved by this architecture, which we call “external address decoder” architecture afterward, with implementing completely the same functions of 1:4 tree scan

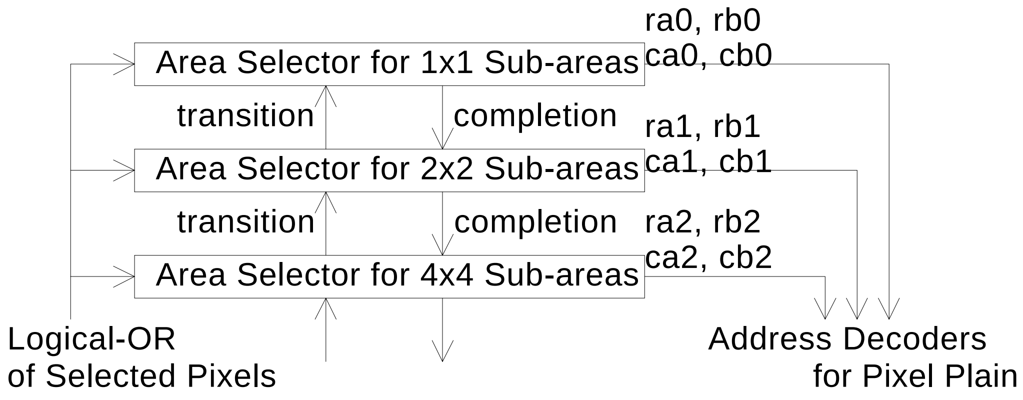

Figure 5.9: Architecture of selection controllers for pixel plain.

Table 5.2: States of automata in the decoding process shown in Figure 5.4.

upper(`s2`)	`W`	`A`	`B`	`B`	`B`	`B`	`B`	`C`	`D`
lower(`s1`)	`W`	`W`	`W`	`A`	`B`	`C`	`D`	`W`	`W`
1:4 tree code	1	0	1	1	0	1	0	0	0

discussed in the previous section.

It is also notable that the external controller of scan procedure and address decoders are completely identical with those for the decoders of 1:4 tree code discussed in section 3.4.

5.2.2 Circuit of controller

The selection procedure of pixels discussed above is implemented by the hierarchical structure of controllers which correspond to each size of sub-area, as shown in Figure 5.9. Each controller has output of `ra` and `rb` for the selection of the upper half and the lower half of its sub-areas, respectively, and `ca` and `cb` for the selection of the left half and the right half of its sub-areas, respectively.

For example, the scan procedure shown in Figure 5.4 can be implemented by two level automata, and their states should be transient as shown in Table 5.2. Here `s1` and `s2` are the automata indicating 1×1 and 2×2 sub-areas, respectively, and state `W` represents selecting all of its sub-area. The state `A`, ...`D` represent the selecting upper-left, upper-right, lower-left, and lower-right of its sub-area, respectively. For

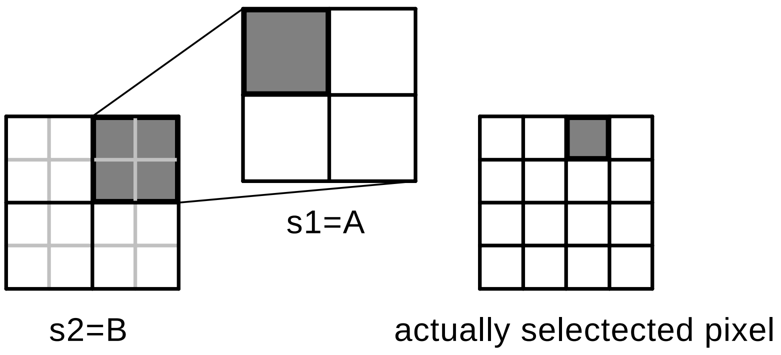

Figure 5.10: Selected pixel by the states of (`s2`,`s1`)=(`B`,`A`)

example, the states of (`s2`,`s1`)=(`B`,`A`) in the fourth step in Table 5.2, the upper-left 1×1 area (or pixel) which is indicated by `s1`=`A`, in the upper-right 2×2 sub-area which is indicated by `s2`=`B`, as shown in Figure 5.10.

The logical-OR of the selected pixels' value is given back to each controller, and the controllers make transition according to its current state and the value, as shown in Table 5.3. Here the state of `W` represents the state of selecting the whole of its sub-area, and `A`, `B`, `C`, and `D` represents the state of selecting the upper-left, upper-right, lower-left, and lower-right of its sub-area, respectively. `Ti` and `Co` is the transition and the completion signals as inputs, respectively, and `To` and `Co` is the transition and the completion signals as outputs, respectively, and `V` is the logical-OR of the selected pixels' value in the pixel plain.

The designed circuit of the controller is shown in Figure 5.11. It is notable that the new signal `SC` is added, to control the scan process in order to implement the spatial adaptive resolution scan, discussed in section 4.2. If `SC` is low, the scan is proceeded as usual 1:4 tree scan, while the scan step for this controller no longer proceeds to its lower level if `SC` is high; it returns the *completion* signal immediately.

5.2.3 Circuit of address decoder

The select lines for each pixel are decoded according to `Ra`, `Rb`, `Ca`, and `Cb` of each controller. For example, the row select line for the top row, `r0`, is activated when `Ra0`, ...`Ran` are all 1, since `Ra` represents the selection of upper-half in each subarea.

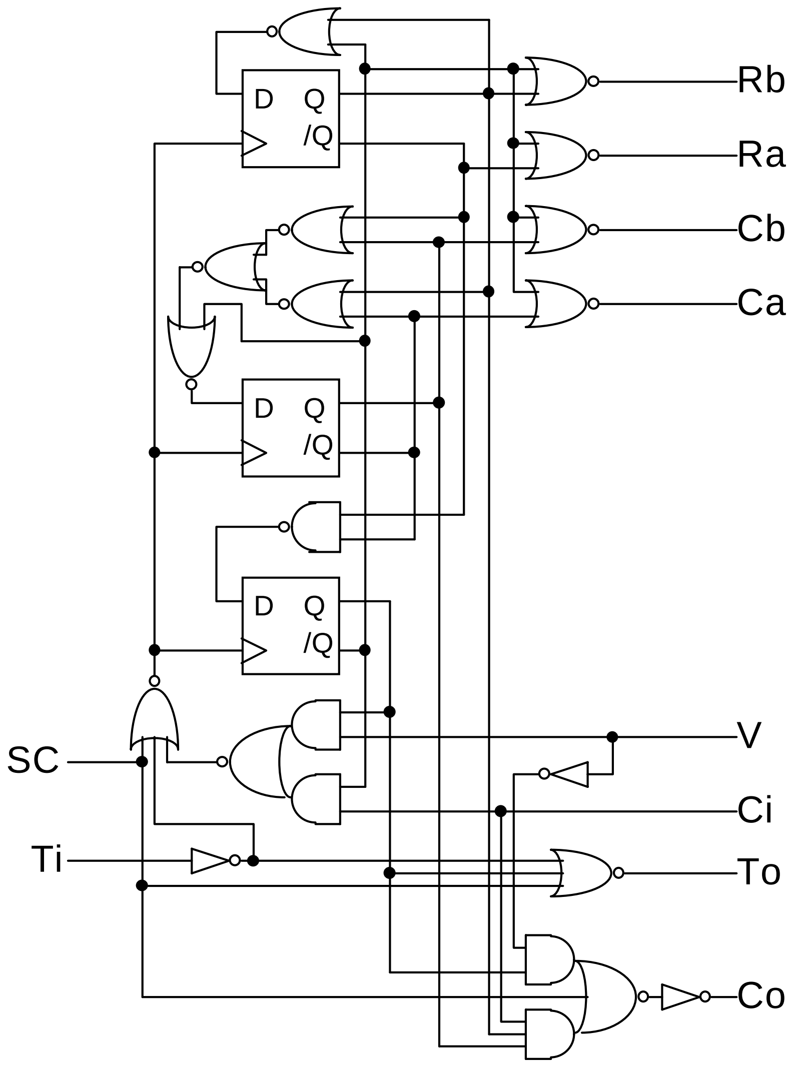

Figure 5.11: Circuit of the controller for the external address decoders.

Table 5.3: State transition diagram of the controller in each level for the external address decoder of 1:4 tree sensor.

Inputs					Outputs					
V	Ci	Ti	S	S'	Ra	Rb	Ca	Cb	Co	To
0	-	↑	W	W	1	1	1	1	1	0
1	-	↑	W	A	1	1	1	1	0	↑
-	0	↑	A	A	1	0	1	0	0	↑
-	1	↑	A	B	1	0	1	0	0	↑
-	0	↑	B	B	1	0	0	1	0	↑
-	1	↑	B	C	1	0	0	1	0	↑
-	0	↑	C	C	0	1	1	0	0	↑
-	1	↑	C	D	0	1	1	0	0	↑
-	0	↑	D	D	0	1	0	1	0	↑
-	1	↑	D	W	0	1	0	1	1	↑

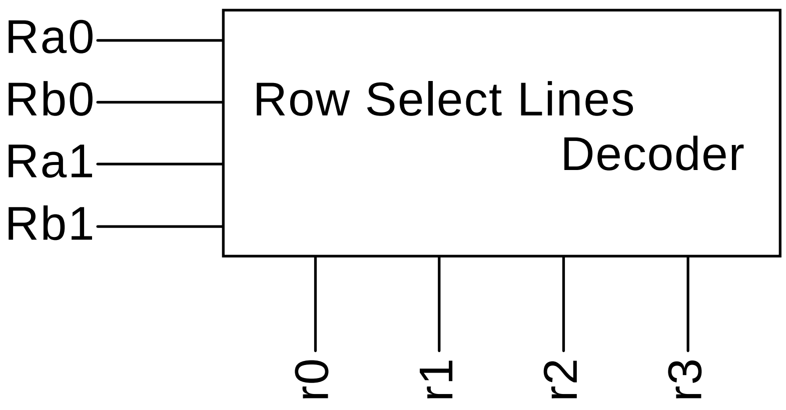

Figure 5.12: Functional block of row decoder for four rows of pixel plain.

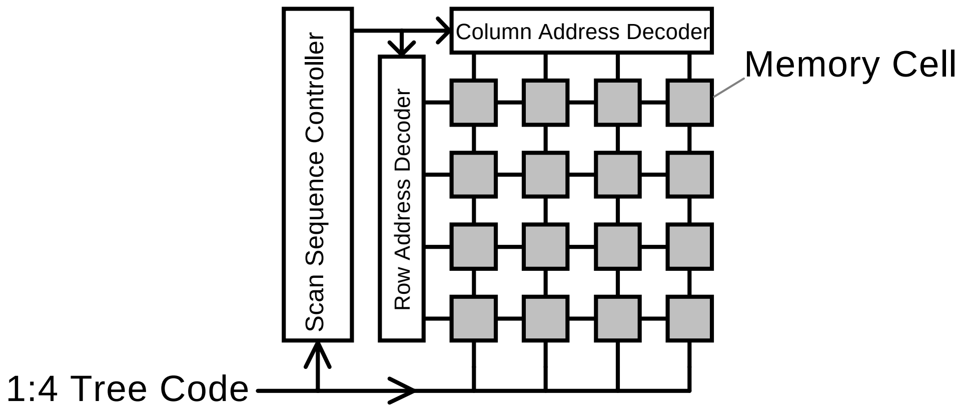

Figure 5.13: Architecture of 1:4 tree code decoder

The four row select lines of pixels, `r0`, ...
r3, as shown in Figure 5.12 are decoded as the following equations, which is similar to those discussed in section 3.4.

$$\texttt{r0} = \texttt{Ra1} \ \&\& \ \texttt{Ra0}$$
$$\texttt{r1} = \texttt{Ra1} \ \&\& \ \texttt{Rb0}$$
$$\texttt{r2} = \texttt{Rb1} \ \&\& \ \texttt{Ra0}$$
$$\texttt{r3} = \texttt{Rb1} \ \&\& \ \texttt{Rb0}$$

Note that && represents the logical-AND. The equations in case of N levels, which can select 2^N rows of pixels, are easily derived by extending the above equations for using N-input AND gates, and the column select lines are also implemented by the same way.

5.3 Circuits of Memory Cell Array for Decoding 1:4 Tree Code

The controllers and the address line decoders designed in section 5.2 are also used to implement the decoder of 1:4 tree code. The architecture of 1:4 tree code decoder is shown in Figure 5.13, and the "scan sequence controller" and "row and column address decoders" are completely identical to those in 1:4 tree sensor using external

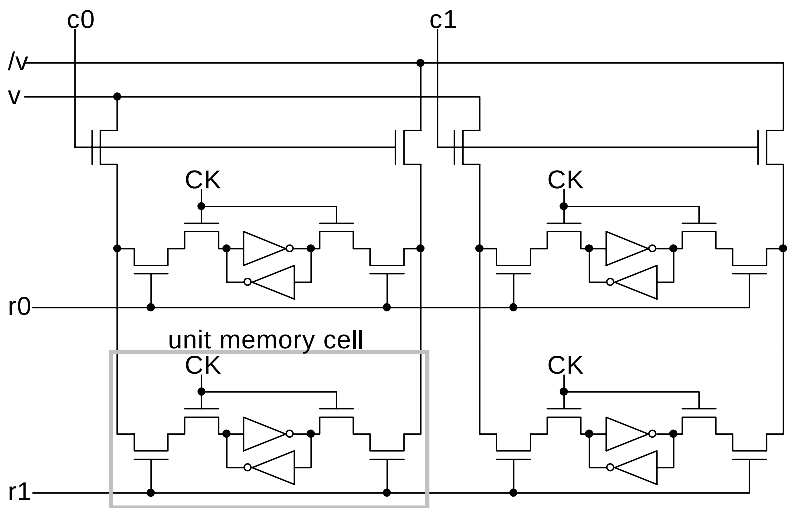

Figure 5.14: Architecture of memory cell plains to implement the decoders of 1:4 tree code

address decoders described in section 3.4.

The architecture of memory cell plain for the decoder, alternatively the pixel plain in Figure 5.5 is shown in Figure 5.14. Here `c0`, `c1` and `r0`, `r1` are column and row select lines generated by the controllers and the address decoders designed in the section 5.2, respectively. `v` and `/v` is the value to be stored for the selected areas, and its inverse, respectively, and `CK` is the clock signal to store the values for the memory cells.

This architecture is similar to that of the conventional SRAM, while more than two memory cells can be selected in this decoder architecture.

5.4 Functions and Circuits of Pixels for On-Sensor Image Processing

5.4.1 Pixel circuit for the inter-frame difference

Figure 5.15 shows the designed pixel circuit with the functions of making inter-frame difference for movie compression, as discussed in section 4.4.

The photo signal is received by the photo diode, and the photo current is converted

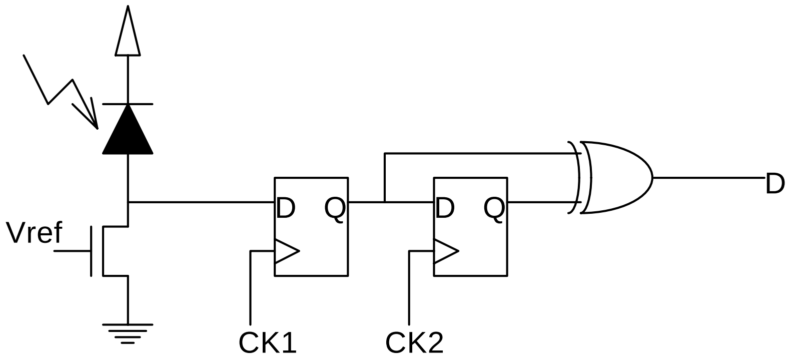

Figure 5.15: Circuit of pixels with the functions of making inter-frame difference.

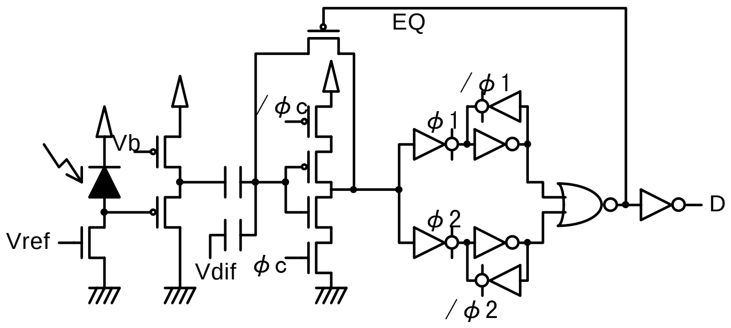

Figure 5.16: Pixel circuit for taking analog inter-frame differences

by the load transistor controlled by `Vref`, as the dark and bright light generates the logical low and high, respectively. The signal is latched by the former D-latch, controlled by `CK1`. The output of the former D-latch is transferred to the later D-latch after reading the output of `D`, which represents the difference of the two D-latchs' outputs. Keeping `CK2` as low after reset sequence, the output `D` is equal to the output of the former D-latch, which is just equal to the output of photo signal.

Figure 5.16 shows the pixel circuit for taking analog inter-frame differences for movie compression. Here `Vref` and `Vb` are bias voltage for load transistor and source follower, respectively. The clocked inverter at the center of figure is equalized by `EQ`, and then ϕ_c and $/\phi_c$ are provided for the clocked inverter to compare the input with the inverter threshold voltage. If the input is higher than inverter threshold voltage V_{inv}, the output goes to low, while it goes to high if the input is lower than V_{inv}. The output of source follower at the previous cycle, V_o^0 is stored at the upper

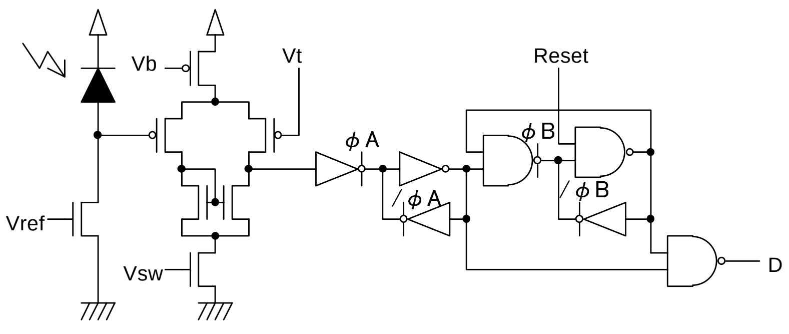

Figure 5.17: Pixel circuit for the scan of intensity by referring the charge-up time

capacitor, and `Vdif` $= V_{\text{inv}} - V_\theta$ is given, where V_θ is the effective difference to be detected between the current output of source follower, V_o and V_o^0. If V_o is larger than $V_o^0 + V_\theta$, the output of clocked inverter goes to 1, and this output is latched to the upper D-latch using ϕ_1 and $/\phi_1$. Next the `Vdif` of $V_{\text{inv}} + V_{\text{theta}}$ is given, and the output of clocked inverter goed to 1 when V_o is smaller than $V_o^0 - V_\theta$. This output is stored the lower D-latch using ϕ_2 and $/\phi_2$. Thus the output `D` is 1 if the absolute value of difference, $|V_o - V_o^0|$ is larger than V_θ, which indicates the effective difference in the successive two frames.

5.4.2 Pixel circuit for the gray-scale converting

In this section, we consider the pixel circuit for the scan of intensity by referring the charge-up time using 1:4 tree structure, discussed in section 4.3.2. Figure 5.17 shows this pixel circuit. The photo current is charged to the junction capacitance of photo diode, and the output voltage of the junction capacitance is compared by the following differential amplifier with the reference voltage `Vt`. `Vb` is the bias voltage for the differential amplifier, and `Vsw` is used to shutdown the amplifier when it is not used. At the first step of scan, the later D-latch is cleared by `Reset` signal, and the output of the differential amplifier is transferred to the former D-latch using ϕ_A and $/\phi_A$. The output of pixel `D` goes high if the output of photo-electronic converter is higher than reference voltage `Vt`, and the time from the reset sequence up to `D` goes high represents the intensity of light. The output `D` is scanned by 1:4 tree structure, and it is cleared after the scan, by transferring "1" of the former D-latch

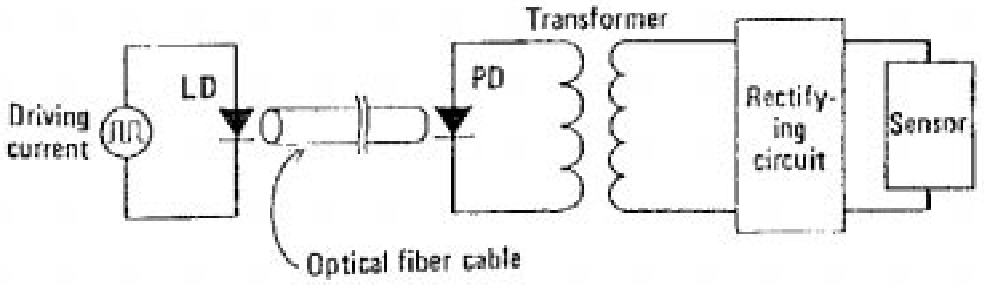

Fig. 1 Basic configuration of power supply of the pulsed-light-power-supply-type sensor

Figure 5.18: Block diagram of the power supply system using the energy of laser diode and optical fiber. (From [28])

to the later D-latch using ϕ_B and $/\phi_B$.

5.5 Light-Powered Image Sensor for Ultra-Low Power Operation

The conventional image sensors are operated by the external power supply. This section describes a concept of the image sensor operated by the power of received light.

Some previous studies of the methodologies to supply the power for image sensor using the laser diode and optical fiber are discussed, as shown in Figure 5.18[28].

The current of photo diode is expressed as follows,

$$I = I_s(e^{qV/nk_BT-1} - 1) - I_p, \tag{5.3}$$

where I_s and I_p is the saturation current and the photo current, respectively, and V is the given voltage of the photo diode. q, n, k_B, and T is the elementary charge, n-value, the Boltzmann's constant, and the operation temperature, respectively.

Assuming $I_p = 1\mu$A, $I_s = 20$pA, $n = 1$, and $T = 300$K, the supplied power by the photo diode and the operation voltage, and the results shows that one photo diode has a ability to supply about 0.1μW. Assuming that the consumed energy per each step of 1:4 tree scan as 100pJ, which will be estimated in section 6.1, the

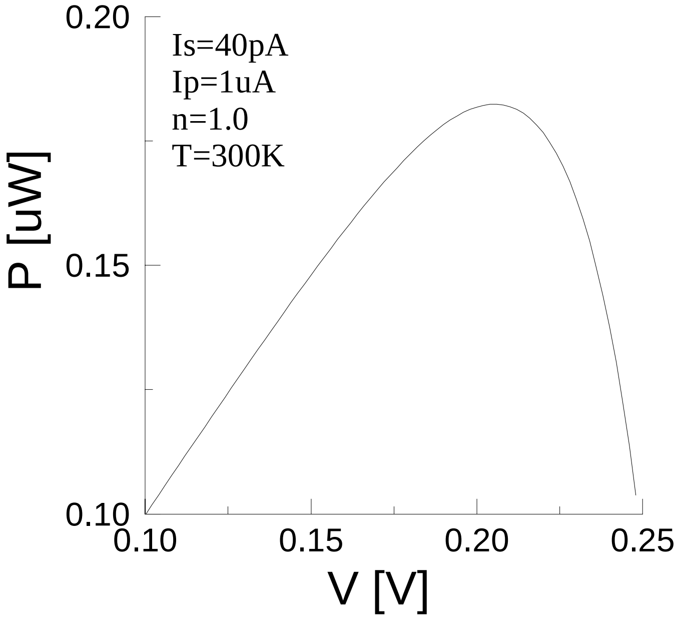

Figure 5.19: Calculated supplied power by the photo diode and the operation voltage.

operation frequency is estimated as 0.1μW÷100pJ=1kHz, which is expected to be enough operation frequency for the ultra-low power applications.

Figure 5.20 shows a concept of the architecture of 1:4 tree structure using the current mode operation. Each node selects the pixels, by closing the current paths to the upper, and the sum of the photo currents for the selected areas is generated.

One of the problems to implement such a circuit in Figure 5.20 is the power supply for the selection controllers. A possible solution is to place two photo diodes for each pixel; one for signals, and the other for power supply for the selection controllers, as shown in Figure 5.21.

5.6 Summary and Conclusion

In this chapter, the circuit implementations of 1:4 tree structure are discussed.

The circuit of node automata for the 1:4 tree structure are described, which has the advantage of lower operation by the gated clock. The reasonable way of two dimensional layout of nodes is also described, which is also reasonable to minize the

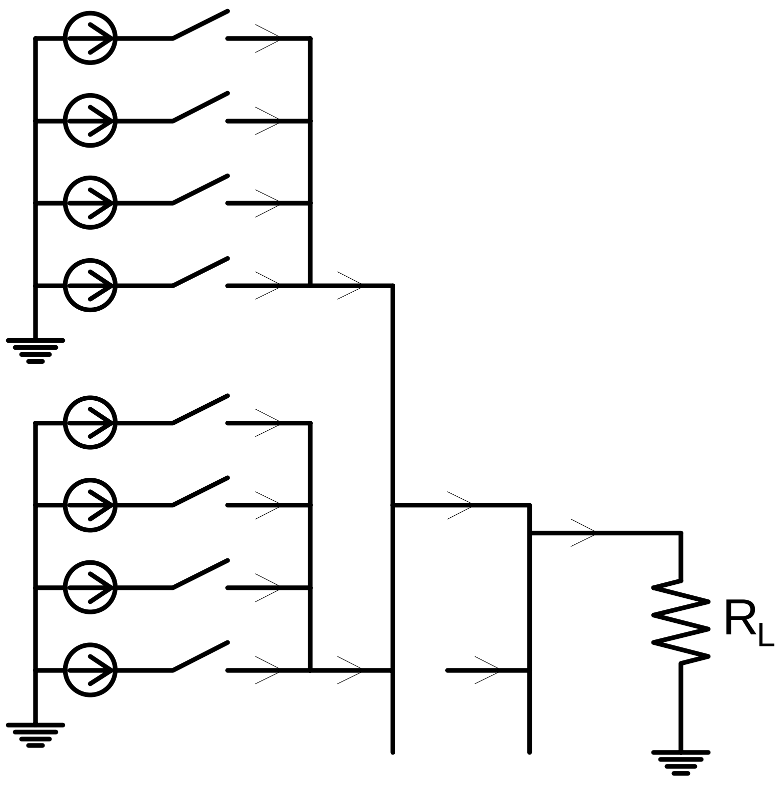

Figure 5.20: Architecture of 1:4 tree structure using current mode operation.

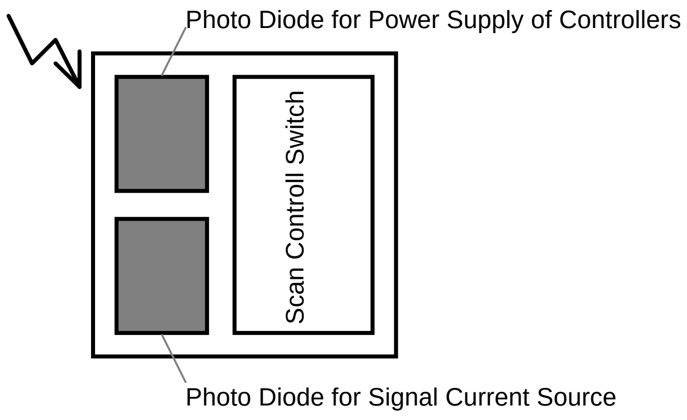
Photo Diode for Power Supply of Controllers
Scan Controll Switch
Photo Diode for Signal Current Source

Figure 5.21: Pixel layout for light-powered 1:4 tree sensor

signal delay, since the upper node automaton can occupy the larger area for the larger signal driver.

It is also described the alternative architecture of the 1:4 tree sensor, the architecture of placing the pixel selection controllers outside the pixel plain, which has the advantage of keeping the high fill factor. The architecture of decoders of the 1:4 tree codes to two dimensional raster image is also described, which can easily implemented by using the address select controllers identical with the external-addressed 1:4 tree sensor.

It is also discussed the circuits and the functions of pixels, in order to have the functions more than simple photo-electric conversion. The idea and implemenatation of the pixels for inter-frame difference and gray-scale conversion are discussed.

The concept of the 1:4 tree sensor driven by the power of received light is also discussed.