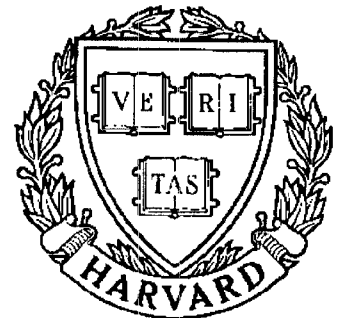


TECHNICAL RESEARCH REPORT



S Y S T E M S
R E S E A R C H
C E N T E R



*Supported by the
National Science Foundation
Engineering Research Center
Program (NSFD CD 8803012),
Industry and the University*

VLSI Implementation of Real-Time Parallel DCT/DST Lattice Structures for Video Communications

*by C.T. Chiu, R.K. Kolagotla,
K.J.R. Liu and J.F. Jájá*

VLSI Implementation of Real-Time Parallel DCT/DST Lattice Structures for Video Communications

C.T. Chiu, R. K. Kolagotla, K.J.R. Liu, and J. F. Jájá.

Electrical Engineering Department
Systems Research Center
University of Maryland
College Park, MD 20742
Phone: (301) 405-6619

ABSTRACT

The alternate use [1] of the discrete cosine transform (DCT) and the discrete sine transform (DST) can achieve a higher data compression rate and less block effect in image processing. A parallel lattice structure that can dually generate the 1-D DCT and DST is proposed. We also develop a fully-pipelined 2-D DCT lattice architecture that consists of two 1-D DCT/DST arrays without transposition. Both architectures are ideally suited for VLSI implementation because they are modular, regular, and have only local interconnections. The VLSI implementation of the lattice module using the distributed arithmetic approach is described. This realization of the lattice module using $2\mu m$ CMOS technology can achieve an 80Mb/s data rate.

This work is partially supported by the NSF grant ECD-8803012.

1 Introduction

Due to the advances in ISDN network and high definition television (HDTV) technology, high speed transmission of digital video signal becomes very desirable. Transform coding plays a significant role in reducing the bit rate for video, image, and digital signal transmission and processing. The discrete cosine transform (DCT), and the discrete sine transform (DST) are two of the most efficient transforms for video and image coding applications. The DCT is widely used for data compression. This is due to its better energy compaction property and to the fact that, for highly correlated signals, its performance is closest to that of the optimal Karhunen-Loeve Transform (KLT) compared to other discrete transforms. This is especially true for first order Markov processes [2, 3, 4]. It was shown by Jain that the performance of the DST approaches that of the KLT for a first-order Markov sequence with given boundary conditions, especially for signals with low correlation coefficients [5, 6]. The DCT approaches the optimal performance of the KLT transform for highly correlated signals, while the DST approaches the optimal performance of the KLT for signals with low correlation coefficients. If we are able to compute the DCT and DST simultaneously, this structure will be very useful, especially when we do not know the statistics of the incoming signal. Rose *et.al* showed in [1] that alternate use of the DCT and DST can achieve the removal of redundancies in the correlation between neighboring blocks, as well as the preservation of continuity across the block boundaries.

We propose a time-recursive lattice structure that can generate the DCT and DST simultaneously. The time-recursive approach is adopted because data arrives serially in digital signal transmission. The resulting architectures are regular, modular, and have only local communication. These structures can compute the transformed data

for sequential input time-recursively with a throughput rate of one sample per clock cycle. The total number of multipliers required is a linear function of the transform size N . Furthermore, there is no constraint on N . Therefore, our architectures are ideally suited for VLSI implementation.

In image and video processing, the two-dimensional discrete cosine transform (2-D DCT) is used for performing data compression. By using the time-recursive concept, we develop a new real-time fully-pipelined architecture to compute the 2-D DCT using the direct 2-D approach. These structures are fully-pipelined and have a throughput rate of N clock cycles for an $N \times N$ successive input data frame. Moreover, the resulting 2-D DCT architectures are modular, regular, and locally-connected and require only two 1-D DCT blocks which are extended directly from the 1-D DCT structure without transposition. It is an attractive direct 2-D DCT architecture for real-time applications of video communications.

The most important block in the 1-D and 2-D DCT lattice structures is the lattice module. The VLSI implementation of the lattice DCT/DST module is described in this paper. In order to achieve an efficient VLSI realization, we adopt the distributed-arithmetic method to implement the multipliers in the structure. The advantages of using this memory-oriented realization are the savings in chip area, better accuracy and higher speed [7]. This chip is currently being fabricated using $2\mu m$ CMOS technology, and will be capable of processing 80 Mb/s data in real-time.

The rest of this paper is organized as follows. In Section 2, the algorithm and structure to generate the parallel 1-D DCT and DST are described. The fully-pipelined 2-D DCT architecture which can be obtained from two 1-D DCT/DST lattice blocks without transposition is discussed in Section 3. In Section 4, the VLSI implementation

of the lattice module by employing the distributed-arithmetic method is discussed. The detailed architectures of the building blocks in the DCT/DST lattice module are described in Section 5. The VLSI realization of the chip and simulation results are given in Section 6. Finally, the conclusion is given in Section 7.

2 Parallel 1-D DCT/DST Lattice Structure

A new scheme employing the time-recursive approach to compute the DCT and DST for time series input data is presented. Since data arrives serially in digital signal transmission, we consider the orthogonal transforms from a time-recursive point of view instead of the entire block of data at a time. Denote $X_c(k, t)$ and $X_s(k, t)$ as the DCT and DST of a data sequence $[x(t), x(t+1), \dots, x(t+N-1)]$. The time-recursive relations for the new transforms $X_c(k, t+1)$ and $X_s(k, t+1)$ in terms of the previous transforms $X_c(k, t)$ and $X_s(k, t)$ are given by

$$\begin{aligned} X_c(k, t+1) = & \left\{ X_c(k, t) + [-x(t) + (-1)^k x(t+N)] \left(\frac{2}{N} \right) \cos \left(\frac{\pi k}{2N} \right) \right\} \cos \left(\frac{\pi k}{N} \right) \\ & + \left\{ X_s(k, t) + [-x(t) + (-1)^k x(t+N)] \left(\frac{2}{N} \right) \sin \left(\frac{\pi k}{2N} \right) \right\} \sin \left(\frac{\pi k}{N} \right), \end{aligned} \quad (1)$$

and

$$\begin{aligned} X_s(k, t+1) = & \left\{ X_s(k, t) + [-x(t) + (-1)^k x(t+N)] \left(\frac{2}{N} \right) \sin \left(\frac{\pi k}{2N} \right) \right\} \cos \left(\frac{\pi k}{N} \right) \\ & - \left\{ X_c(k, t) + [-x(t) + (-1)^k x(t+N)] \left(\frac{2}{N} \right) \cos \left(\frac{\pi k}{2N} \right) \right\} \sin \left(\frac{\pi k}{N} \right). \end{aligned} \quad (2)$$

The lattice module manifesting this approach is shown in Fig. 1(a), where $k = 1, 2, \dots, N-1$. The simplified module for $k = 0$ in the DCT and $k = N$ in the DST is given in Fig. 1(b). The resulting parallel lattice structure is shown in Fig. 2. Note that the transform domain data $X(k, t)$ has been decomposed into N disjoint

components that use similar lattice modules with different multipliers coefficients in them. This structure requires $6N - 4$ multipliers and $5N - 1$ adders; the total computational time is N clock cycles. It is important to note that the transformed data of subsequent input vectors can be generated every clock cycle.

3 Real-Time and Fully Pipelined 2-D DCT Lattice Architectures

The two-dimensional discrete cosine transform (2-D DCT) has been widely recognized as the most effective technique in image and video processing. Although the 2-D DCT can be computed from the 1-D DCT through the row-column decomposition methods, the direct 2-D DCT approach is more efficient since the transposition operation can be eliminated. We propose a parallel and fully pipelined direct 2-D DCT lattice structure based on the frame-recursive approach.

The parallel 2-D DCT lattice structures we propose can generate the 2-D DCT and discrete sine-cosine transform (DSCT) simultaneously. Let $\{X_c(k, l, t) : k, l = 0, 1, \dots, N - 1\}$ and $\{X_{sc}(k, l, t) : k, l = 0, 1, \dots, N - 1\}$ be the t 'th frame $N \times N$ 2-D DCT and DSCT (defined in [8] for the $N \times N$ 2-D data sequence $\{x(m, n) : m = t, t + 1, \dots, t + N - 1; n = 0, 1, \dots, N - 1\}$). The recursive relations for the $(t + 1)$ 'th frame transformed data $X_c(k, l, t + 1)$ and $X_{sc}(k, l, t + 1)$ in terms of the t 'th frame transformed data $X_c(k, l, t)$ and $X_{sc}(k, l, t)$ are given by

$$\begin{aligned} X_c(k, l, t + 1) = & \left\{ X_c(k, l, t) + \delta_c(k, l) \frac{2}{N} C(k) \cos\left(\frac{\pi k}{2N}\right) \right\} \cos\left(\frac{\pi k}{N}\right) \\ & + \left\{ X_{sc}(k, l, t) + \delta_c(k, l) \frac{2}{N} C(k) \sin\left(\frac{\pi k}{2N}\right) \right\} \sin\left(\frac{\pi k}{N}\right), \end{aligned} \quad (3)$$

and

$$X_{sc}(k, l, t+1) = \left\{ X_{sc}(k, l, t) + \delta_c(k, l) \frac{2}{N} C(k) \sin\left(\frac{\pi k}{2N}\right) \right\} \cos\left(\frac{\pi k}{N}\right) - \left\{ X_c(k, l, t) + \delta_c(k, l) \frac{2}{N} C(k) \cos\left(\frac{\pi k}{2N}\right) \right\} \sin\left(\frac{\pi k}{N}\right), \quad (4)$$

where

$$X_{sc}(k, l, t) = \frac{4}{N^2} C(k) C(l) \sum_{m=t}^{N+t-1} \sum_{n=0}^{N-1} x(m, n) \sin\left[\frac{\pi[2(m-t)+1]k}{2N}\right] \cos\left[\frac{\pi(2n+1)l}{2N}\right], \quad (5)$$

and

$$\delta_c(k, l) = \frac{2}{N} C(l) \sum_{n=0}^{N-1} [(-1)^k x(t+N, n) - x(t, n)] \cos\left[\frac{\pi(2n+1)l}{2N}\right]. \quad (6)$$

The relation between $X_c(k, l, t+1)$ and $X_c(k, l, t)$ can be realized by a lattice array (LAII in Fig. 3) whose structure is similar to the lattice module described in the previous section. It should be noted that $\delta_c(k, l)$ in (6) is the 1-D DCT of the data vector which is the difference between the parity of the $(t+N)$ 'th and t 'th rows of the 2-D input sequence. From the previous section, it is shown that $\delta_c(k, l)$ can be generated time recursively by the lattice array I (as shown in Fig. 3) whose lattice module is described in Fig. 1. The fully-pipelined lattice structure for the frame-recursive 2-D DCT and DSCT is shown in Fig. 3. The circular shift matrix in Fig. 3 is used to store all the 1-D DCT data of the $(t+N)$ 'th and t 'th rows of the input data frame for the frame-recursive 2-D DCT.

In most image processing applications, the 2-D DCT are executed block by block instead of in successive frames. Let us consider the case of obtaining the 0'th frame 2-D DCT by the moving frame 2-D DCT architecture. Since the 0'th frame is the first input data frame, all the values in the circular shift matrix are set to zero. When the $(N-1)$ 'th row data vector (the last row of the 0'th frame) arrive, the 2-D DCT of the

0'th input data frame is obtained. During this initial stage, the 2-D DCT of the 0'th frame obtained by the moving frame approach is equal to the block 2-D DCT of the 0'th frame. Therefore, if we want to compute only the block 2-D DCT, the circular shift matrix can be removed. This means that we can use a simpler structure than that shown in Fig. 3 for the computation of the block 2-D DCT. There are many interesting results in this structure. First, the lattice array can be viewed as a filter bank. This is because every lattice module itself is an independent digital filter with different frequency components $l = 0, 1, \dots, N - 1$. Moreover, all the lattice modules in this architecture have the same structure which are regular, modular, and without global communication. Second, the system requires only two 1-D DCT arrays and operates in a fully pipelined fashion with a throughput rate of N clock cycles for the frame recursive approach.

4 Implementation of the Lattice Structure Using High-Speed Distributed-Arithmetic

Since the lattice module is the most important component in the DCT/DST lattice structures, we focus on the VLSI implementation of the lattice module. It has been shown in [9, 10] that the DCT and other sinusoidal transforms can be implemented in the form of filter banks. Every lattice module in the DCT/DST structure is a modified normal form digital filter [11], which has the least roundoff noise and is less sensitive to coefficient inaccuracy. Due to the fact that the block 2-D DCT operation will reset all the outputs of the LAI and $LAII$ every N and N^2 clock cycles respectively, the roundoff errors will be further minimized [12]. In the following we will focus our

discussion on the 8 point DCT with 8-bit input signals and 12-bit output signals using two's complement binary number system.

Suppose we want to construct the lattice module that is based on the Liu-Chiu2 module with 4 multipliers [13]. Then the total number of multipliers needed for the 8×8 2-D DCT is 64, which is quite excessive. In addition, the system throughput will be limited by the operational speed of the multipliers.

Sun *et. al.* [7, 11] proposed the first working 16×16 DCT chip which incorporates the distributed-arithmetic method. Using this memory-oriented structure, a high speed, high accuracy efficient hardware implementation of the 2-D DCT can be achieved. We adopt the distributed-arithmetic scheme in our VLSI implementations. The multiplication results stored in the ROM are computed using double precision numbers on a SUN workstation. The lattice module can be redrawn as shown in Fig. 4. The dashed box in Fig. 4 can be implemented by using a single ROM with three inputs and four outputs. Using this realization, the roundoff errors due to multiplication are minimized since distributed-arithmetic transforms explicit multiplication into implicit multiplication. Therefore, the errors of the system are all due to the quantization errors resulting from finite precision implementation and addition operations. Under the 12-bit two's complement realization, the RMS error values are approximately 40dB [7], which is satisfactory for most applications. Assuming that every input of the ROM is 2-bit long, the lattice module can be implemented using 6 ROMs and 22 adders as shown in Fig. 5. The ROM size for each lattice array is 18432 bits. By reducing the number of bits of every input of the ROM to one, the ROM size reduces to 4608 bits. This is one-fourth of the previous case, but the the number of adders needed is doubled.

Another way to implement the lattice module using ROMs is shown in Fig. 6. Each dashed box is realized using a ROM with one input and two outputs. Fig. 6 illustrates the realization of each ROM when the number of bits of the input signal is 6 bits. Using this method, the ROM size of each lattice array is 13824 bits and the number of adders needed is 10. When the number of bits of the input signal is reduced, the ROM size is reduced but the number of adders is increased. We implement our system based on the schematic diagram for each lattice module as shown in Fig. 6. The adder is a 12-bit carry lookahead full adder/subtractor which is constructed using three 4-bit carry-look-ahead adders. Since, ROMs need less area than general purpose multipliers and can achieve a higher speed, circuit implementations using this approach can be used for very high speed video signal processing.

5 Design of the Building Blocks

The main building blocks of the lattice structure are ROMs, adders, shift registers, and latches. These components will be described individually in this section. The critical path in our lattice modules is the feedback loop which includes one ROM and three adders. A traditional two-phase clocking scheme would use one phase to perform these computations and a second phase to latch the results. In order to make the two phases of the clock more symmetric, we perform computations on both phases of the clock. As shown in Fig. 7, we perform one addition and the ROM lookup during the first phase and two additions during the second phase. Intermediate results are stored in half-latches as described below.

5.1 ROM Implementation

As described in Section 4, the main component of the distributed-arithmetic based lattice structure is the ROM. Most existing ROMs are implemented based on the precharge concept, that is, the bit lines are precharged high during the precharge phase, and then the selected word lines discharge some of the bit lines according to the coefficients stored during the evaluate phase. In order to reduce the ROM access time, we use a novel ROM design [14]. Fig. 8 shows the detail of each cell in the ROM. A simple inverter with a feedback transistor and a transmission gate controlled by phase ϕ_1 is used as a sense amplifier at the output of the bit-lines. We precharge the bit lines to an intermediate voltage between GND and Vdd, and use n-channel transistors to either charge the bit line from this intermediate voltage to Vdd-Vt or discharge it to GND, during the evaluate phase. In this scheme, the array is fully populated; *i.e.* the number of n-channel transistors in the array is MN, where M is the number of word lines and N is the number of bit lines. A ‘zero’ is stored at a particular location, by connecting the n-channel transistor at that location to Vdd; a ‘one’ is stored by connecting the transistor to GND. The layout of the storage cells in the ROM array is shown in Fig. 9. The cell size is only $13\lambda \times 16\lambda$.

In our distributed-arithmetic scheme, the multiplication of the 12-bit input number with a 12-bit sine or cosine coefficient is performed by two ROMs each with 6-bit inputs and two adders. This reduces the chip area needed to implement multiplication with fixed coefficients. The ROM includes two 6-bit decoders and two small ROMs as shown in Fig. 7. The 12-bit input is divided into two parts; the most significant 6-bits of the input are used to generate the coefficients for small ROM1 and the least significant 6-bits are used for small ROM2. The final result of the multiplication is

obtained by adding the outputs of small ROM1 with a shifted version of the outputs of small ROM2. We only store the most significant 7-bit result of the multiplication at ROM2. The sizes of small ROM1 and ROM2 are 64 words by 24 bits and 64 words by 14 bits respectively.

In order to improve the ROM access time, each 6-bit decoder is implemented as a tree consisting of two 3-bit decoders and a linear array of 64 AND gates. The delay time for this 6-bit decoder is 8.55ns, while a straightforward implementation would have a delay of 20.40ns. The outputs of the 64 AND gates form the word lines of the ROM array. The logical layout of the 6-bit decoder is shown in Fig. 10. The physical size of the final ROM is $1292\lambda \times 1646\lambda$ which is much smaller than the area needed by a general purpose multiplier. The total ROM access time is 20ns.

5.2 Adders

Since the lattice modules are implemented based on a word-serial bit-parallel approach, high-speed bit-parallel adders are necessary. We build a 12-bit carry lookahead adder using three 4-bit carry lookahead adders [15]. Two large inverters are placed at the outputs of the adders to supply sufficient drive capability. The physical size of the 12-bit adder is $1022\lambda \times 256\lambda$ and the propagation delay through the adder is 18ns.

5.3 Shift Registers and Latches

There are two kinds of latches and one shift registers in the circuit. One of the latches is the half-latch which is controlled by phase ϕ_2 and is used to store the intermediate results obtained from the adders. The logical representation of the 12-bit half-latch

is depicted in Fig. 11. The other latch is the reset controlled half-latch located in the feedback loop. Its logical circuit is shown in Fig. 11. When the reset signal goes low, the outputs from ROM2 and ROM3 are set to zero. The shift register located at the input stage of the system is a regular two phase shift register which delays the input sequence by eight clock cycles.

5.4 Control Unit

Only one control signal (reset) and two clock phases (ϕ_1 and ϕ_2) are required in this system. Phase ϕ_1 is used to latch the computational results from one adder and the ROM, while phase ϕ_2 is used to latch the results from the remaining adders. Since the propagation delay time for the ROM and the adder is approximately the same, we can make both clock phases symmetric to each other. The signal diagram of these two phases is depicted in Fig. 12. The reset signal is active low. One of the attractive features of this chip is the very simple control signals used. No additional logical control circuitry is needed in the design.

6 Chip Realization and Simulations

Having realized the symbolic layout of the individual blocks, the next issue is to integrate all these components efficiently. The floor plan of the lattice module is shown in Fig. 13. This includes three ROMs, eleven adders, four half-latches, two reset controlled half-latches and one shift register. ROM2 and ROM3 are rotated by 90 and 270 degree respectively to simplify inter-component routing. This chip accepts 8-bit input signals and produces 12-bit DCT and DST coefficients every 100ns. The

physical layout of the lattice module chip is depicted in Fig. 14. There are 18000 transistors in the chip, most of which are used in the three fully-populated ROMs. The total size of the active circuitry is $5400\lambda \times 3400\lambda$. This is fabricated in a die of size $6800\lambda \times 4600\lambda$ and packaged in a 40-pin chip.

Both logical and timing simulations were performed on the this chip. From the simulation results due to Sun *et. al* [7], the word length of the ROM must be at least 9 bits to ensure that the SNR is greater than 40dB. To reduce the error due to recursive computations, we increase the word length of the ROM to 12 bits. We used IRSIM to perform logic simulations on the layout of the chip. The results from IRSIM were compared with the DCT and DST of the same input sequence obtained from a C program written on a SUN workstation. The results are accurate up to the least significant bit of the 12-bit representation. The SNR of this computation from simulations is about 41dB, which is satisfactory for image and video processing applications. SPICE simulations indicate that the worst case rise and fall time for the ROM bit-lines are 8ns and 9ns respectively. Fig. 15 shows the timing simulation for the entire ROM which includes the decoder, cell array, and sense amplifier. The sense amplifier, together with a feedback transistor is used to precharge the bit lines to an intermediate voltage between GND and Vdd. This decreases the voltage swing on the bit lines during the evaluate phase, and hence reduces the ROM access time. Fig. 15(a) shows the case when the output is charged to high. The total delay time from input to output is 20ns. Fig. 15(b) shows the case when the output discharges from this intermediate voltage to GND. It should be noted that although the bit line does not charge up to Vdd, the sense amplifier can still discharge the output to GND in a relatively short time. The delay time in this case is 15ns. The delay time for the

three stage 12-bit carry lookahead adder is 20ns. The critical path in the structure is the feedback loop, which contains one ROM and three adders. Timing simulations indicate that the chip will work at a frequency of 10MHz and hence input data at a rate of 80Mb/s can be processed in real-time.

7 Conclusion

The algorithm and architecture of the first chip that can generate the 1-D DCT and DST simultaneously were described. The fully-pipelined 2-D DCT lattice structure that can be generated from two 1-D DCT/DST lattice arrays without transposition was proposed. We implemented the lattice module using the distributed arithmetic method with a data rate of 80Mb/s under $2\mu\text{m}$ CMOS technology. Simulation results clearly show that our VLSI implementation is a good candidate for real-time video and image processing. We are currently exploring further refinements and a full-implementation of a system that can handle 16×16 block sizes.

References

- [1] K. Rose, A. Heiman, and I. Dinstein, "DCT/DST Alternate-Transform Image Coding," IEEE Trans. on Communication, vol.38, No. 1, pp. 94-101, Jan. 1990.
- [2] A. Rosenfeld, and A. C. Kak, *Digital Picture Processing*, 2nd edition, Academic Press, 1982.
- [3] N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete cosine transform," IEEE Trans. Comput., vol. C-23, pp. 90-93, Jan. 1974.
- [4] R. J. Clark, "Relation between the Karhunen-Loeve and cosine transform," IEE Proc., vol. 128, pt. F, no. 6, pp. 359-360, Nov. 1981.
- [5] A. K. Jain, "A fast Karhunen-Loeve transform for a class of stochastic process," IEEE Trans., Commun., vol. COM-24, pp.1023-1029, 1976.

- [6] R. Yip and K. R. Rao, "On the computation and effectiveness of discrete sine transform," *Comput. Elec. Eng.*, vol. 7, pp. 45-55, 1980.
- [7] M. T. Sun, T. C. Chen, and A. M. Gottlieb, "VLSI implementation of a 16×16 Discrete Cosine Transform," *IEEE Trans. on Circuits and Systems*, vol. 36, No. 4, pp. 610-617, Apr. 1989.
- [8] P. Duhamel and C. Guillemot, "Polynomial Transform computation of the 2-D DCT," *IEEE ICASSP Proc.*, pp. 1515-1518, March. 1990.
- [9] R. D. Koilpillai, and P. P. Vaidyanathan, "New Results on Cosine-Modulated FIR filter banks satisfying perfect reconstruction," *IEEE ICASSP*, 1991, pp. 1793-1796.
- [10] R. Gluth, "Regular FFT-related transform kernels for DCT/DST based polyphase filter banks," *IEEE ICASSP*, 1991, pp. 2205-2208.
- [11] S. A. White, "Applications of Distributed-Arithmetic to Digital Signal Processing: A Tutorial Review," *IEEE ASSP Magazine*, pp. 4-19, July. 1989.
- [12] C. T. Chiu, and K. J. R. Liu, "Real-Time Parallel and Fully-Pipelined Two-Dimensional DCT Lattice Structures with Application to HDTV Systems," to appear on *IEEE Trans. on Circuits and Systems for Video Technology*, March 1992.
- [13] K. J. R. Liu, and C. T. Chiu, "Unified Parallel Lattice Structures for Time-Recursive Discrete Cosine/Sine/Hartley Transforms," accepted by the *IEEE Trans. on Signal processing*.
- [14] L. A. Glasser, and D. W. Dobberpuhl, "The Design and Analysis of VLSI Circuits," Addison Wesley, 1985.
- [15] Neil H. E. Weste, and Kamran Eshraghian, "Principles of CMOS VLSI Design, A Systems Perspective," Addison WEsley, 1985.

Fig.1(a) The lattice module for the DCT and DST with $k = 1, 2, \dots, N - 1$.

Fig.1(b) The lattice module of $k = 0$ and $k = N$ for the DCT and DST.

Fig.2 The lattice array.

Fig.3 The moving frame 2-D DCT architecture.

Fig.4 The reorganized lattice module.

Fig.5 The realization of the lattice module using one ROM.

Fig.6 The realization of the lattice module using three ROMs.

Fig.7 The building blocks of the lattice module with clocks.

Fig.8 The logical diagram of ROM.

Fig.9 The layout of cells in the ROM array.

Fig.10 The logical diagram of the 6-bit decoder.

Fig.11 The logical diagram of half latch and reset controlled half latch.

Fig.12 The signal diagram of the control signals.

Fig.13 The floorplan diagram of the lattice module.

Fig.14 The layout diagram of the lattice module.

Fig.15(a) The Spice timing simulation of the ROM for output charged to Vdd.

Fig.15(b) The Spice timing simulation of the ROM for output discharged to GND.

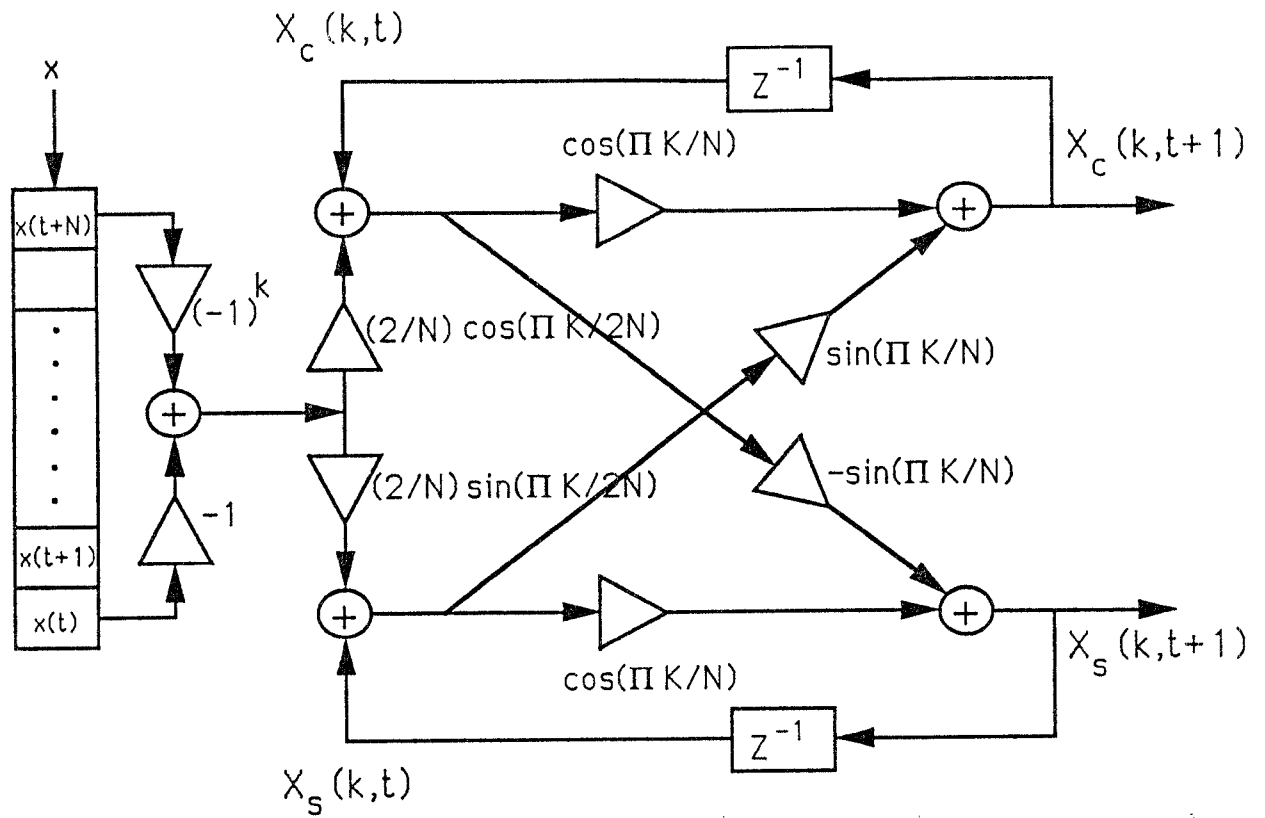


Fig.1(a) The lattice module for the DCT and DST with $k = 1, 2, \dots, N-1$.

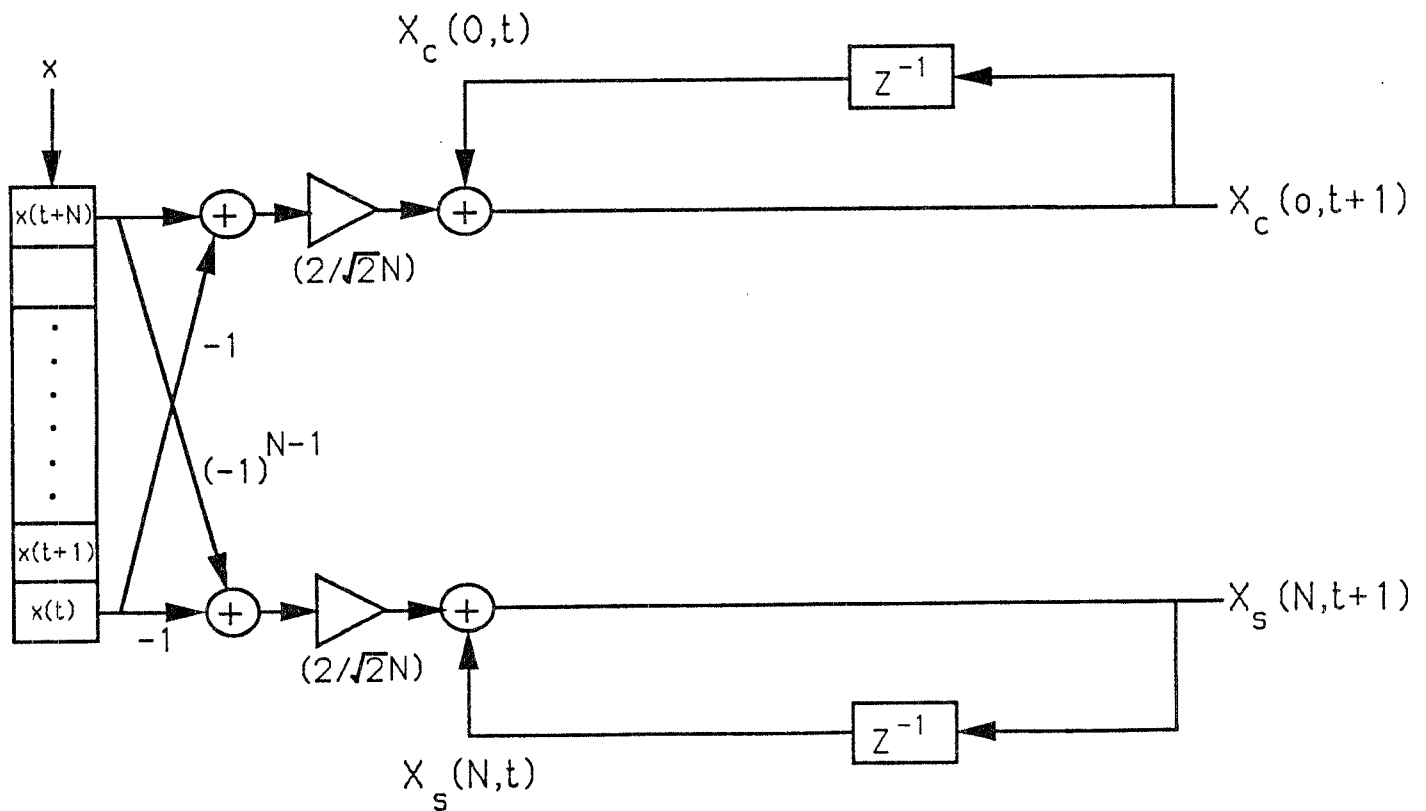


Fig.1(b) The lattice module of $k = 0$ and $k = N$ for the DCT and DST.

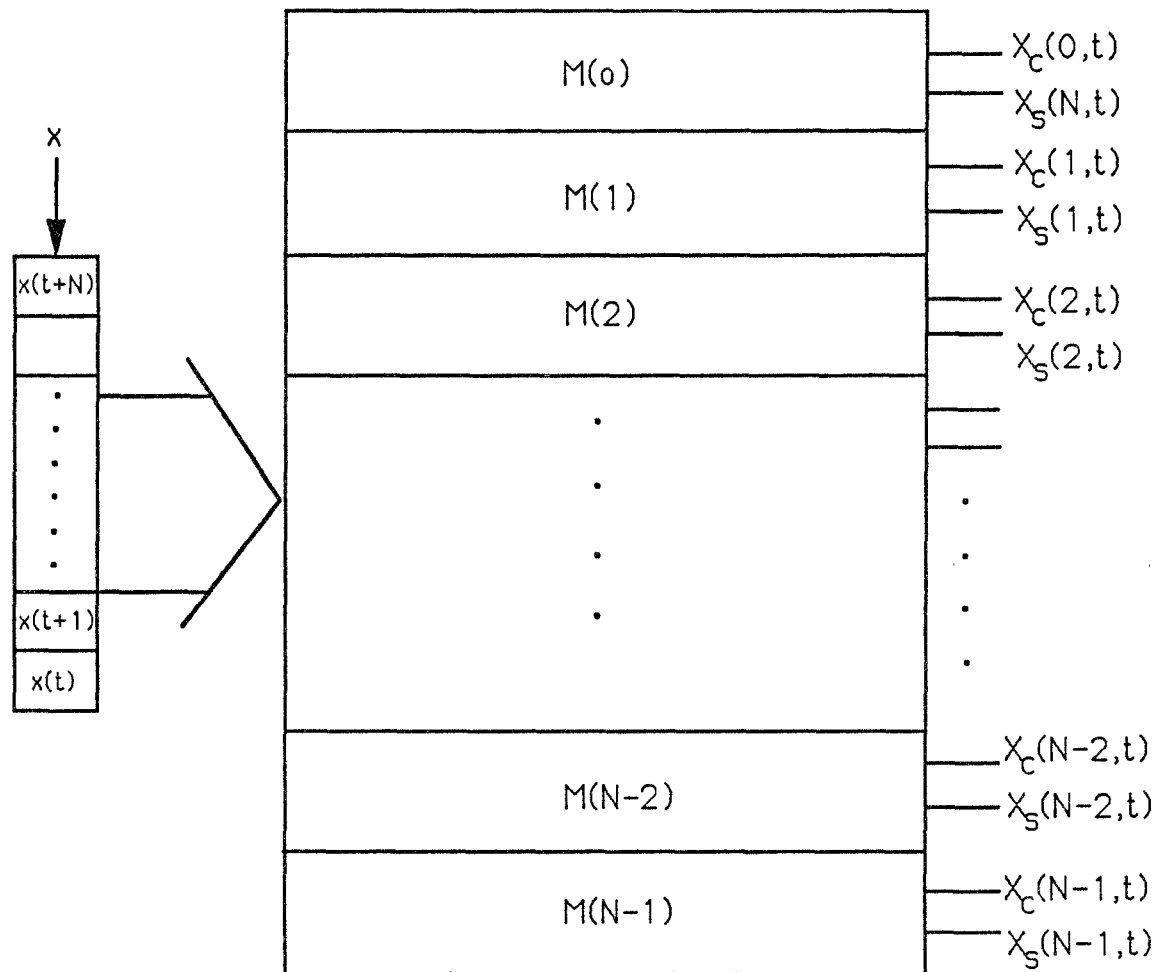


Fig.2 The lattice array.

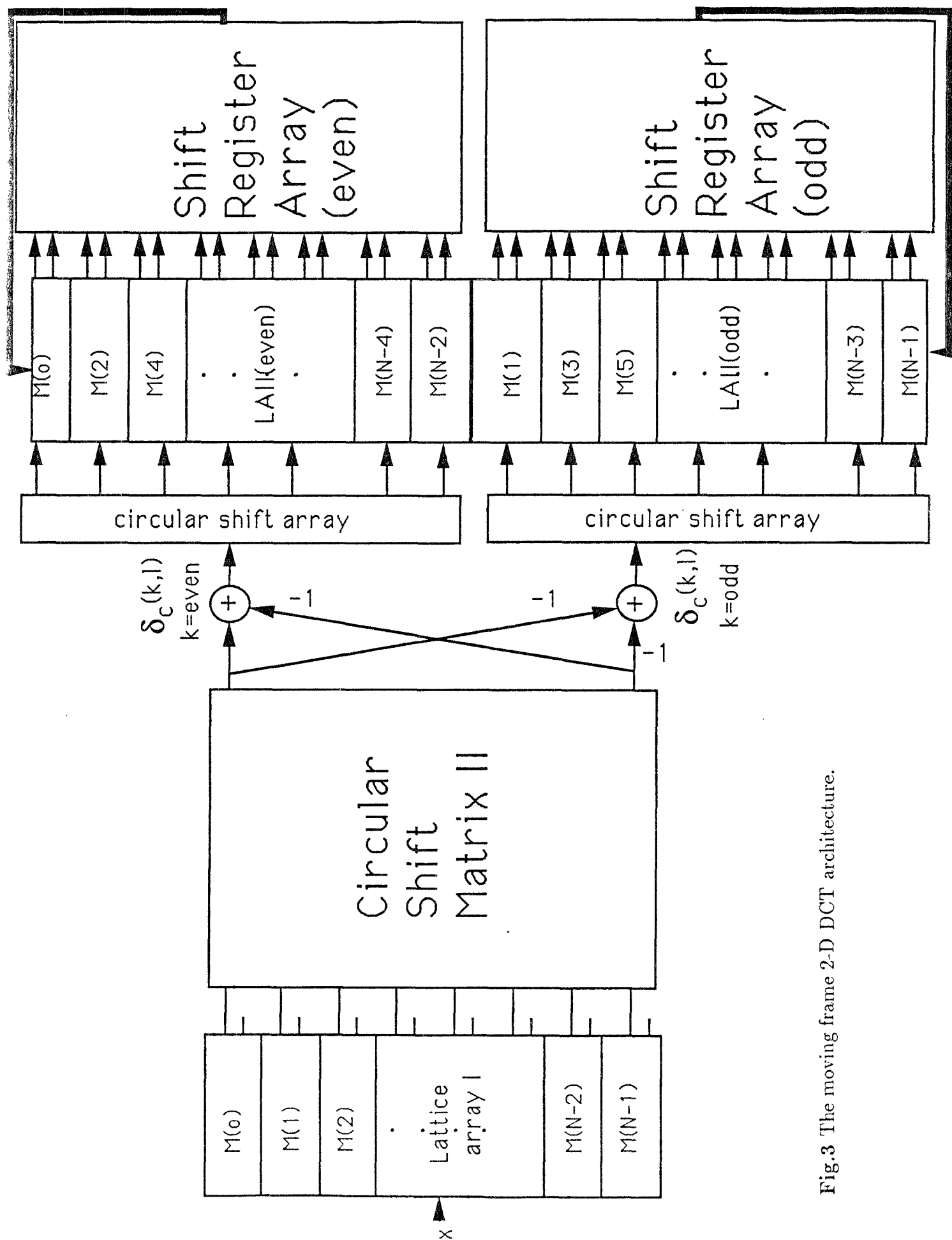


Fig.3 The moving frame 2-D DCT architecture.

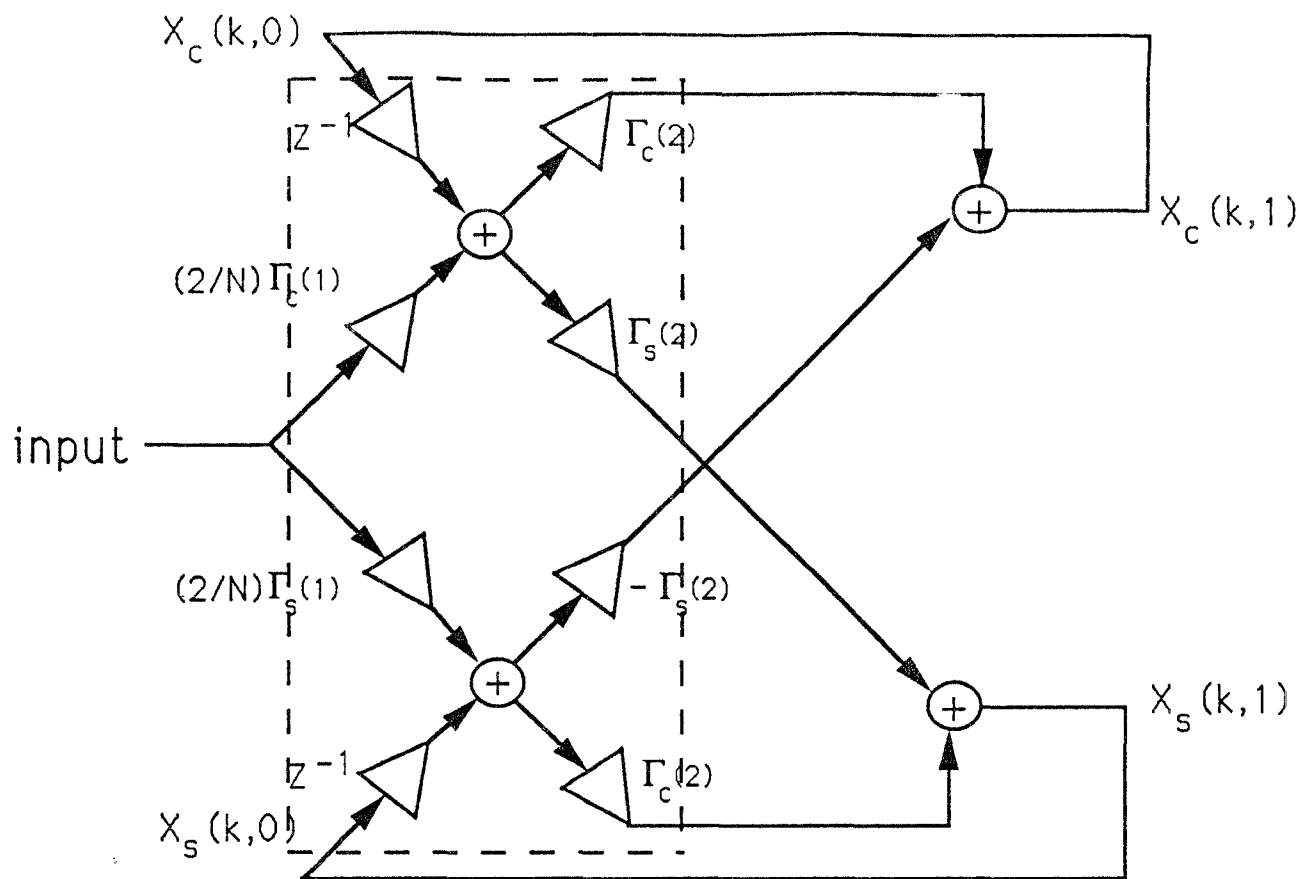


Fig.4 The reorganized lattice module.

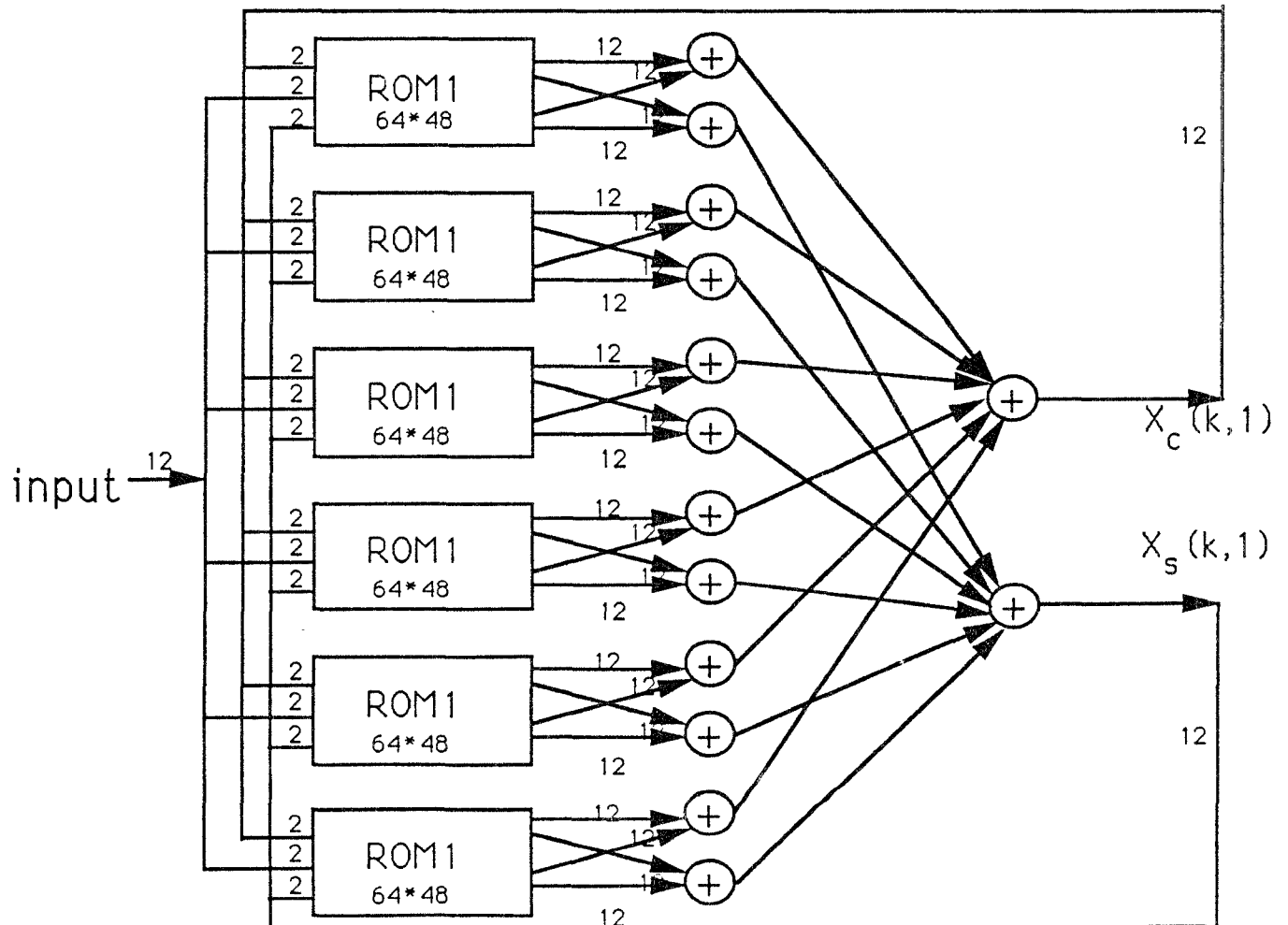


Fig.5 The realization of the lattice module using one ROM.

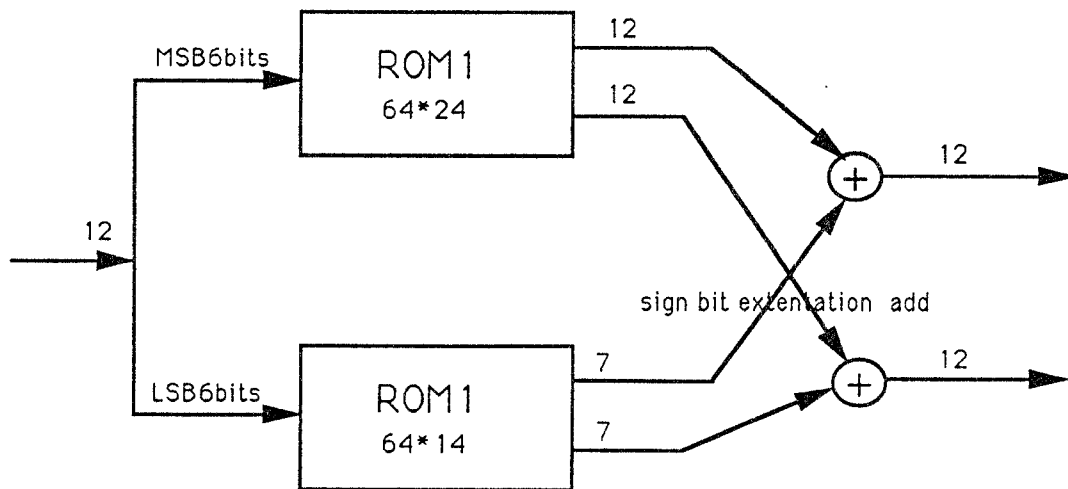
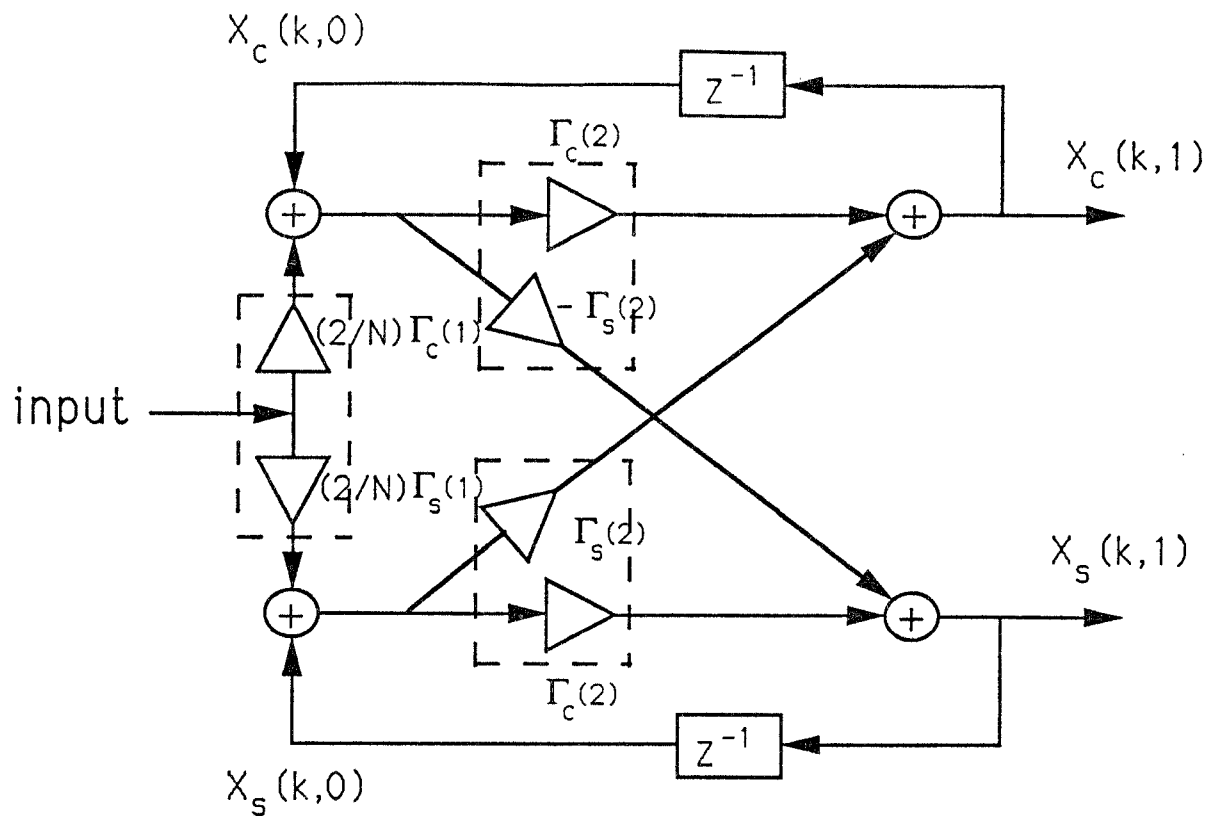


Fig.6 The realization of the lattice module using three ROMs.

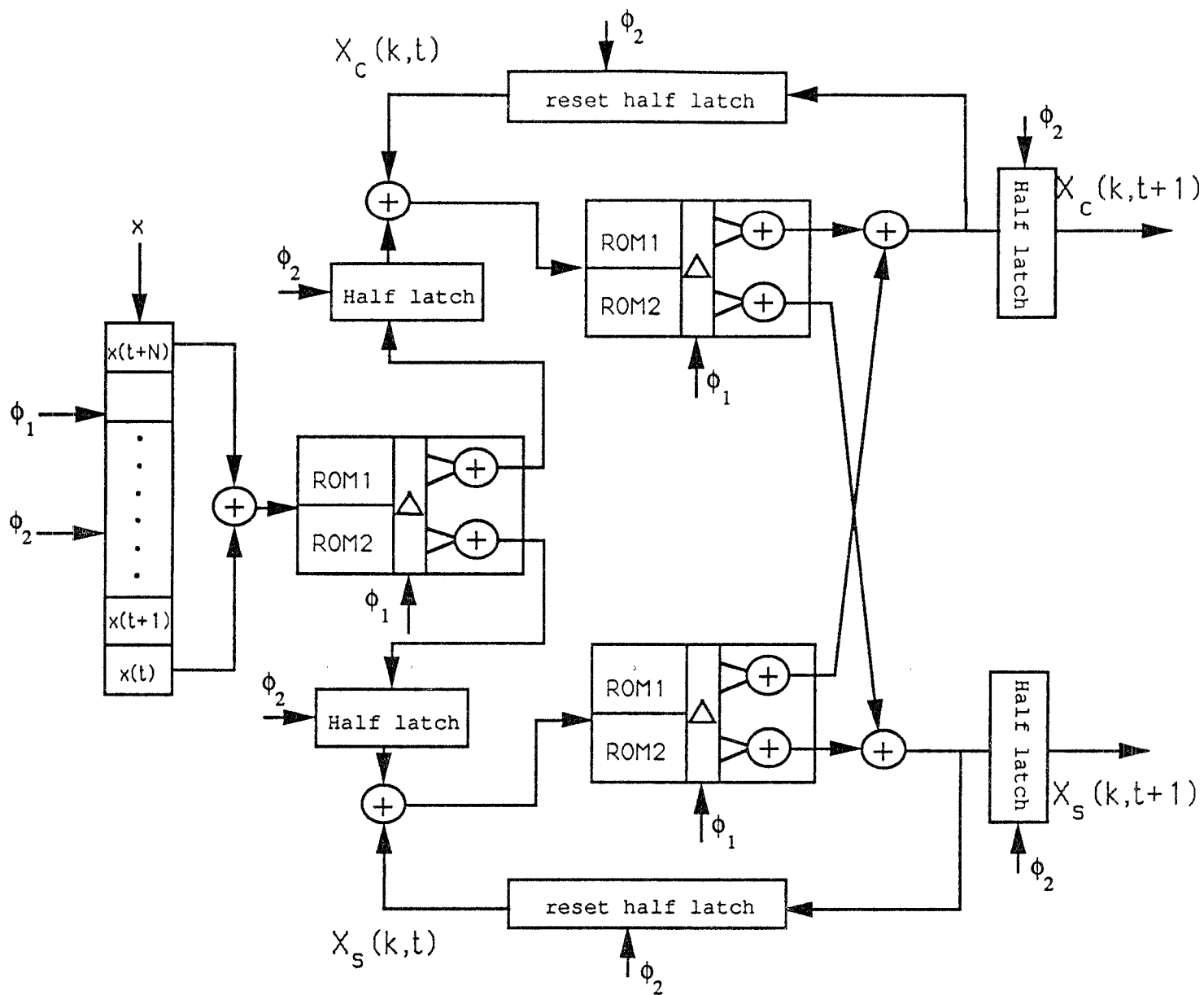


Fig.7 The bulding blocks of the lattice module with clocks.

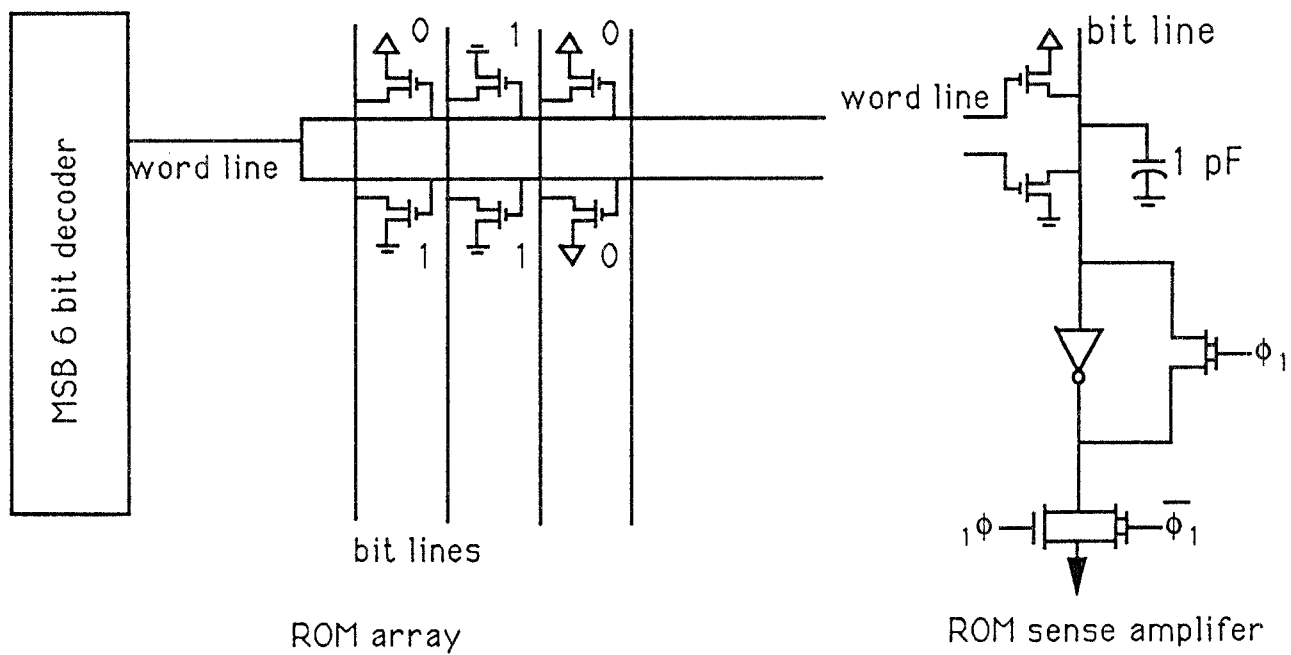
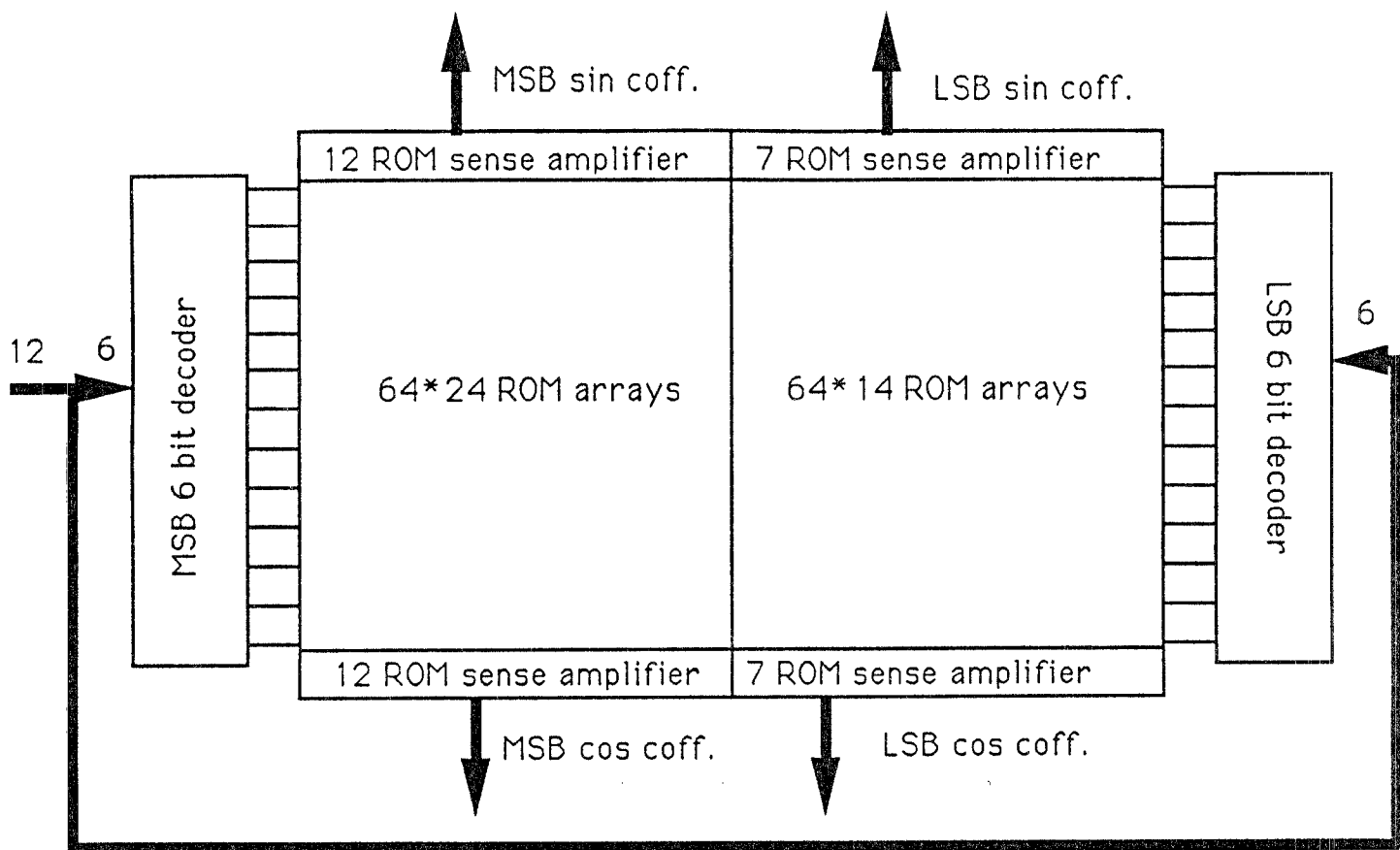


Fig.8 The logical diagram of ROM.

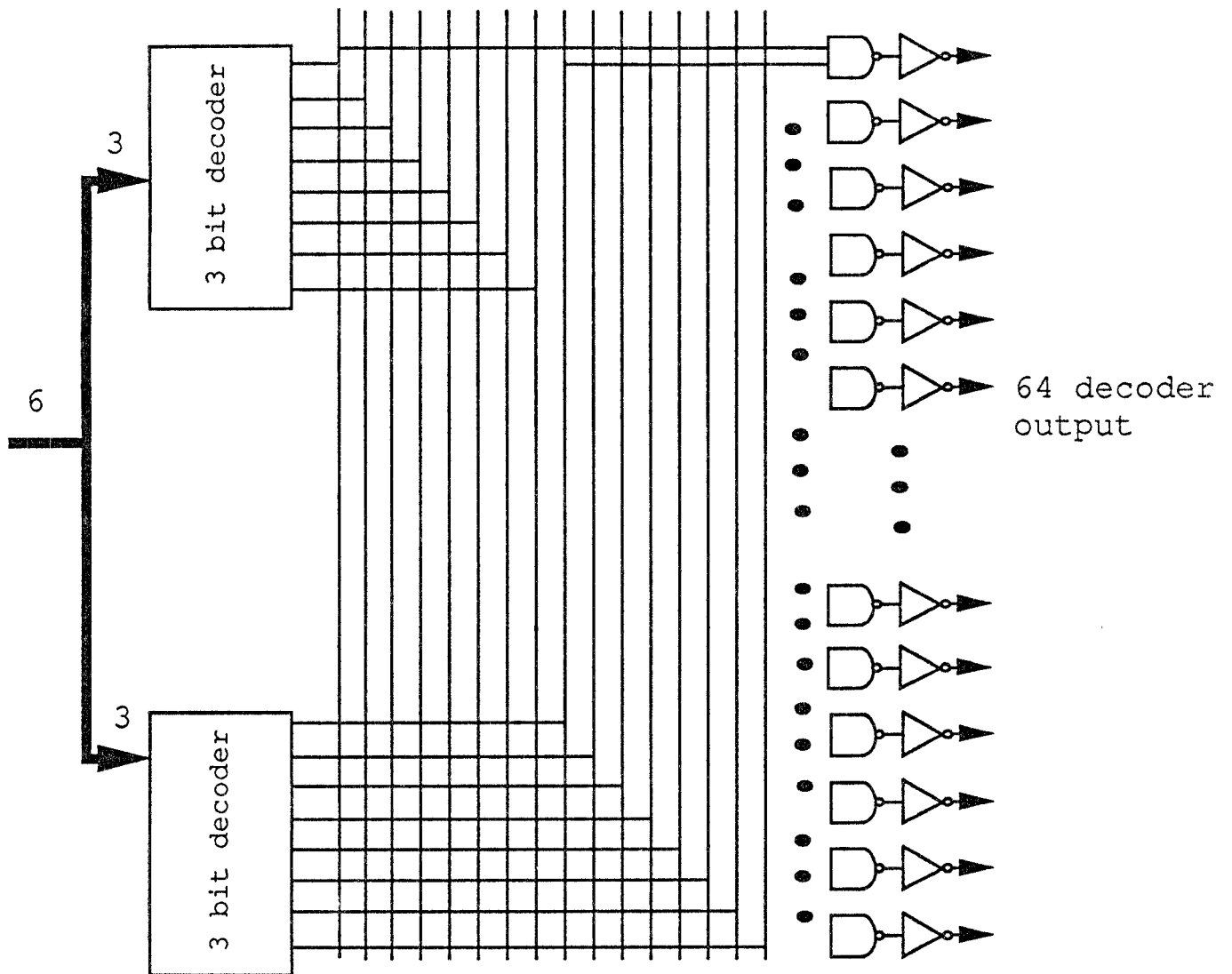
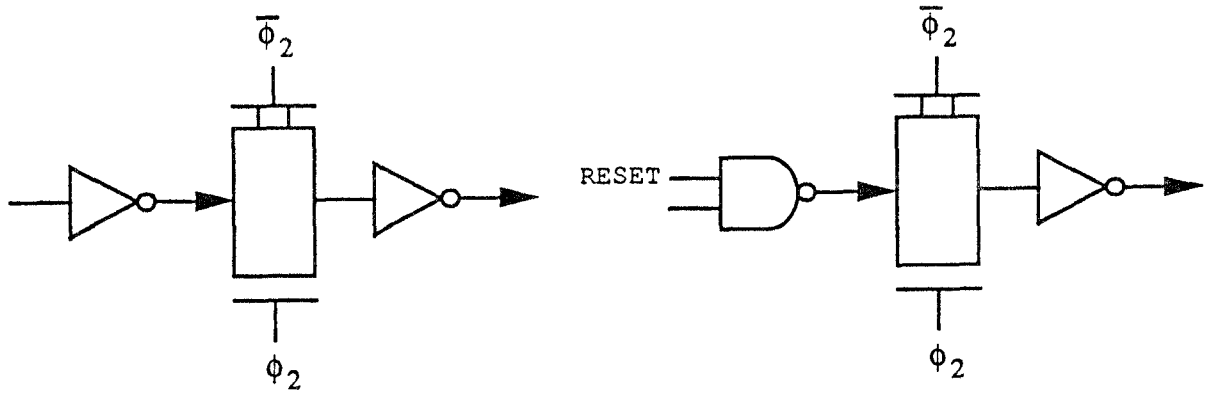


Fig.10 The logical diagram of the 6-bit decoder.



Half latch

Reset control half latch

Fig.11 The logical diagram of half latch and reset controlled half latch.

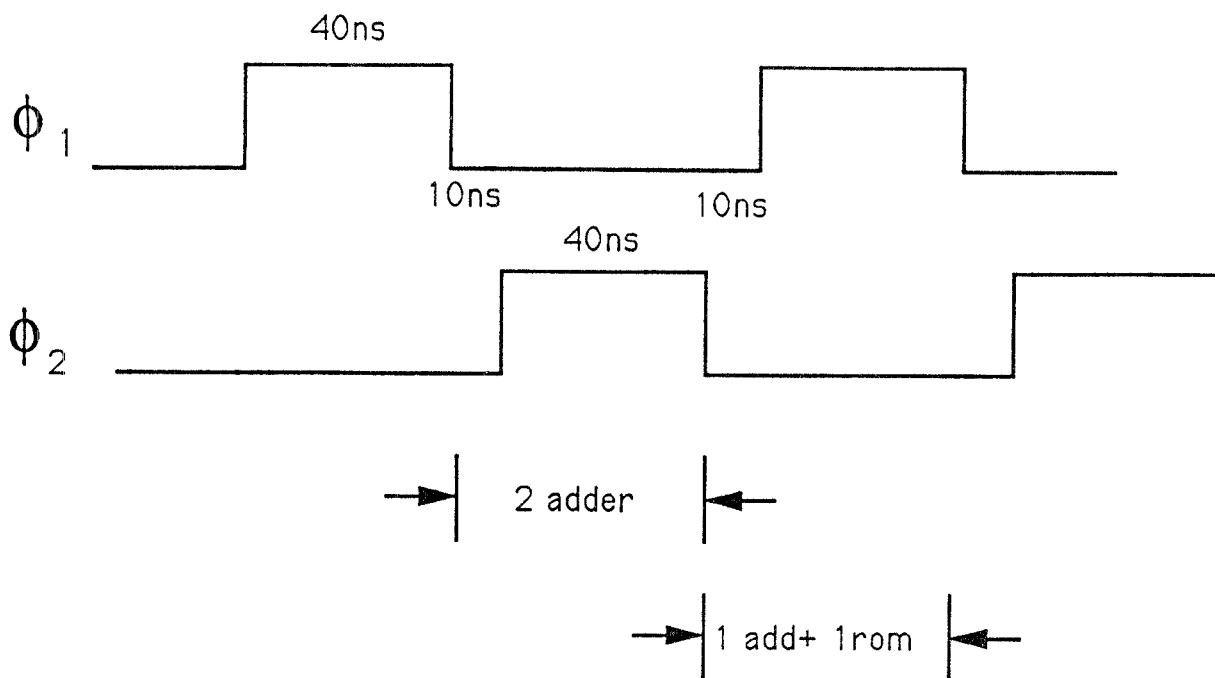


Fig.12 The signal diagram of the control signals.

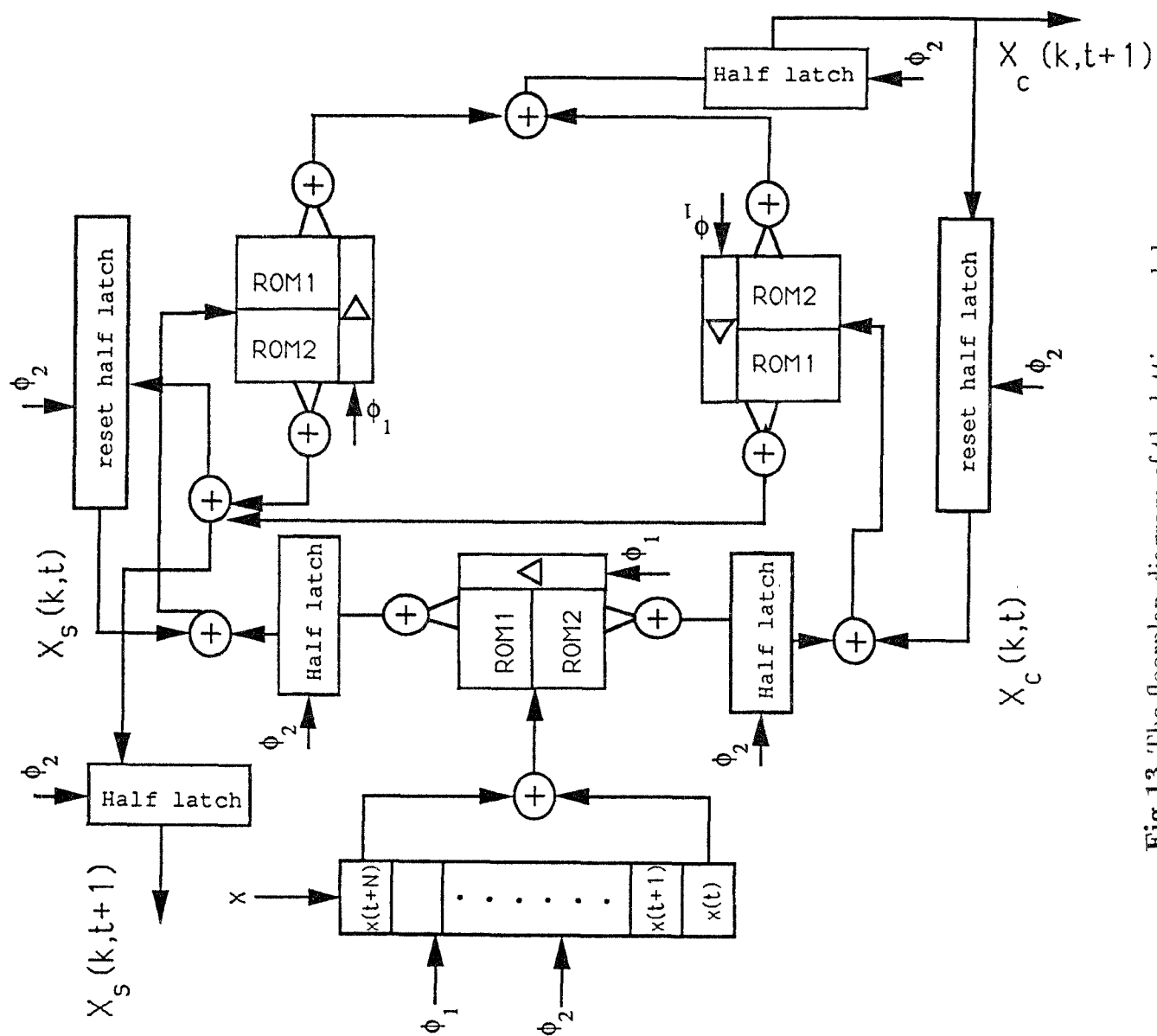


Fig.13 The floorplan diagram of the lattice module.

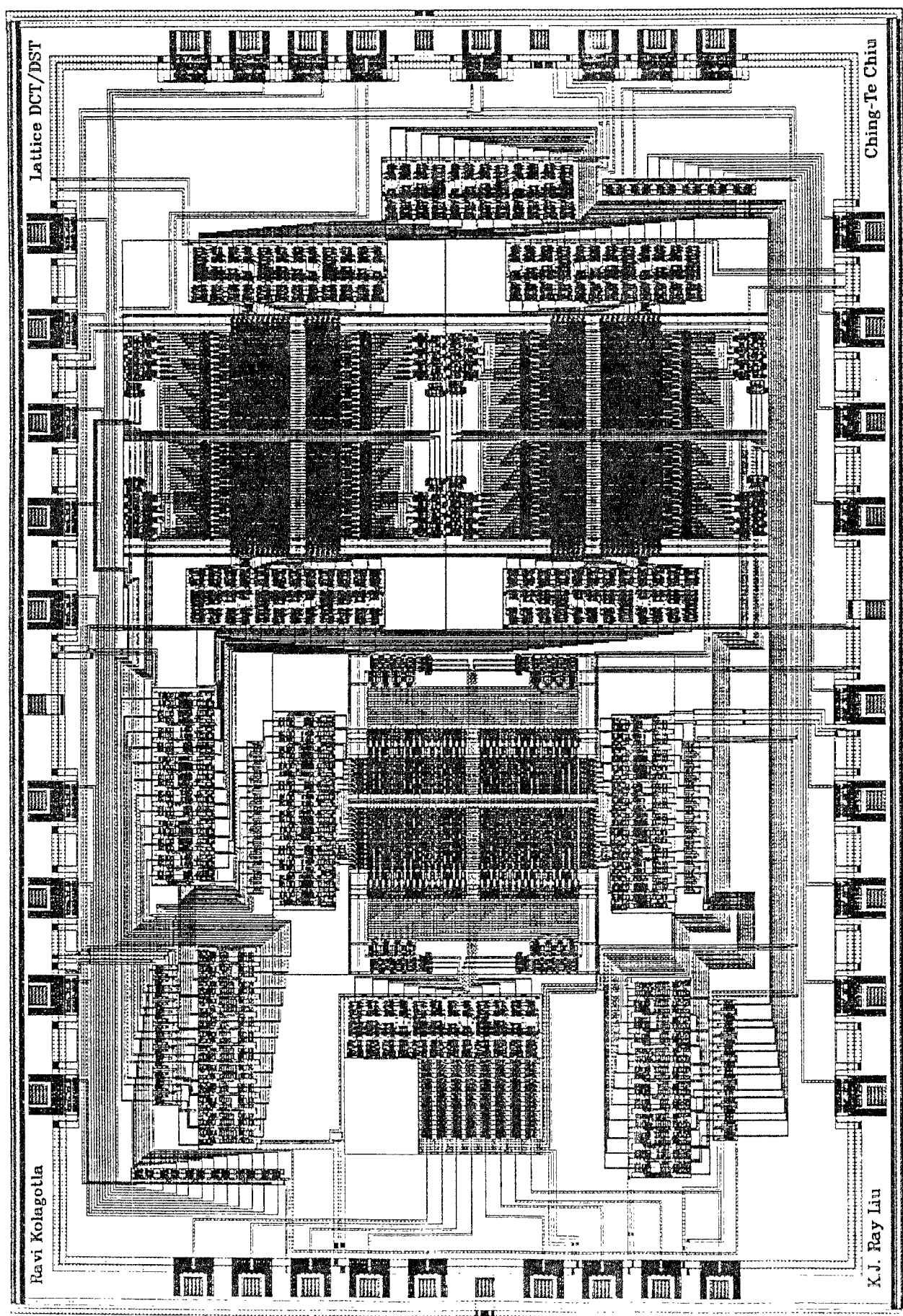


Fig.14 The layout diagram of the lattice module.

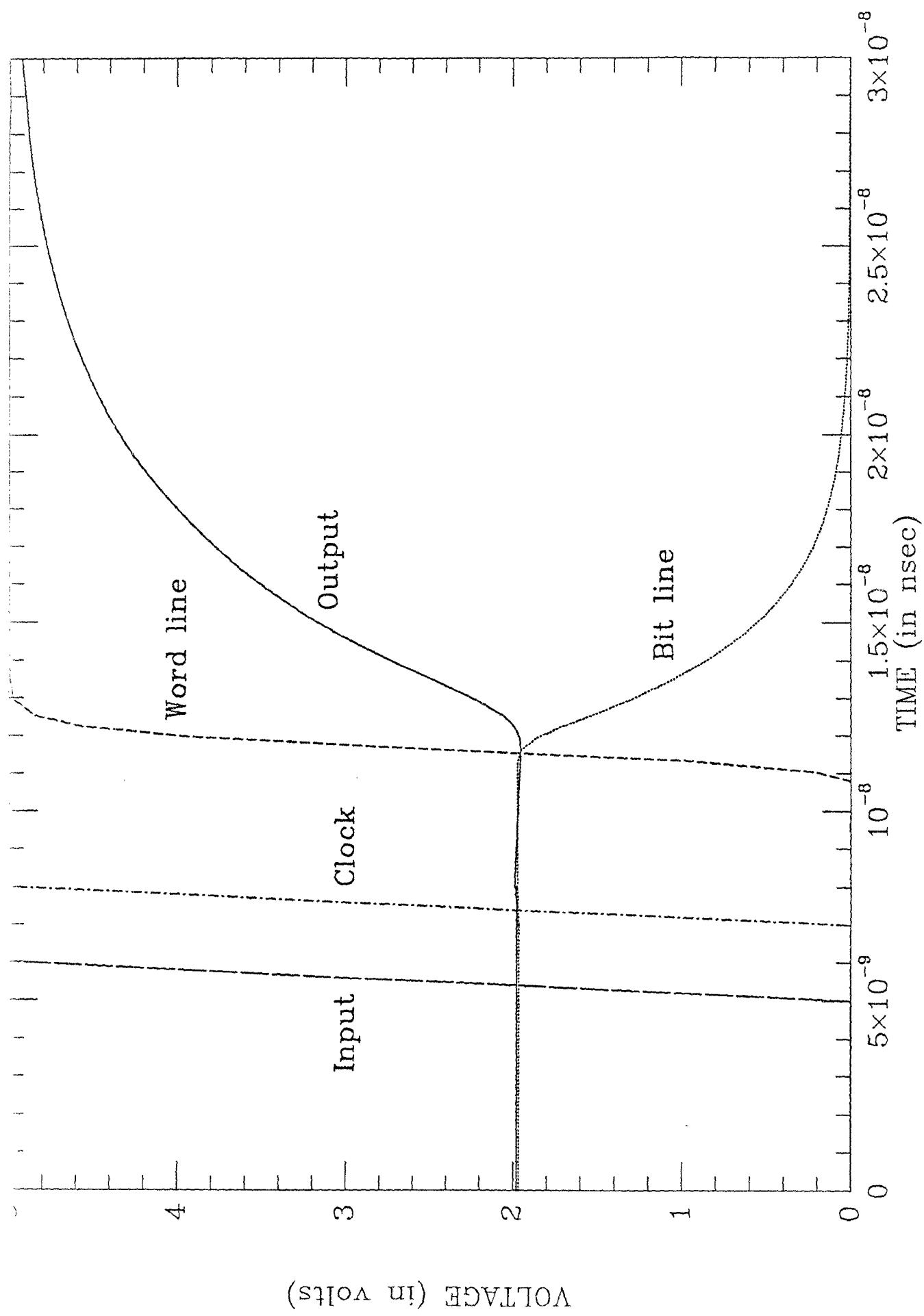


Fig.15(a) The Spice timing simulation of the ROM for output charged to Vdd.

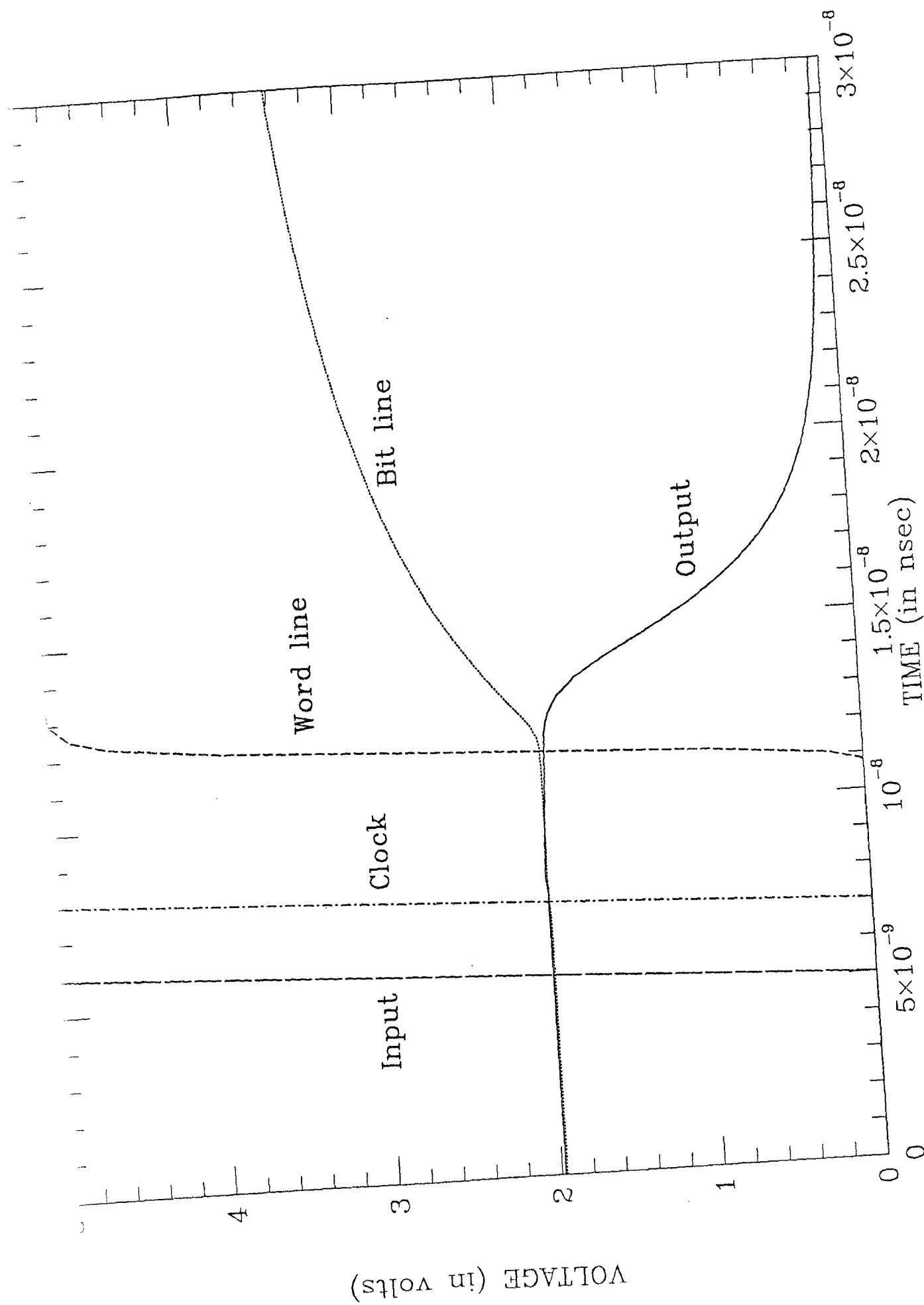


Fig.15(b) The Spice timing simulation of the ROM for output discharged to GND.

