1-1-1988

A writable programmable logic array

Yuan Iee Hwang

Rochester Institute of Technology
School of Computer Science and Technology

A Writable Programmable Logic Array

by

Hwang, Yuan Iee

A thesis, submitted to
The Faculty of the School of Computer Science and Technology,
in partial fulfillment of the requirements for the degree of
Master of Science in Computer Science

Approved by: James Heliotis 6/27/88

Dr. James Heliotis

Roy S. Czernikowski 6/27/88

Dr. Roy Czernikowski

George A. Brown 6/27/88

Prof. George Brown (Chairman)

June 27, 1988

TABLE OF CONTENTS

TABLE OF FIGURES

Appexdix A

Appexdix B

0. ABSTRACT

This thesis contains the analysis, design, and implementation of a writable programmable logic array integrated circuit. The WPLA is able to be reprogrammed any number of times as needed. A content addressable scheme is proposed to conduct READ, WRITE, and SEARCH operations in the WPLA. The WPLA is programmed by writing binary data into storage cells associated with each node in the AND/OR planes of the array; the binary data then form the *personalities* of the PLA. The layout of the WPLA will be implemented using Mentor Graphic's CHIPGRAPH layout editor with 2 μm NMOS technology and MOSIS design rules. The event-driven logic level simulator QUICKSIM, and a MOS circuit level simulator MSIMON, are used to verify the functional and timing behavior of the WPLA.

1. INTRODUCTION and BACKGROUND

This section provides a background of PLAs and their related technology. With the advent of VLSI technology, circuit complexity has been increasing exponentially, but using structured design concepts and design automation tools, design cycle times can be made shorter. Designers can use *standard cells* and/or a *silicon compiler* to create structured modules inorder to reduce development time. Moreover, through minor modification and with little overhead, PLAs [1]-[5] can be constructed into a testable architecture to improve *fault coverage* and reduce *test time*.

1.1 Overview of Programmable Hardware

The history of PLA development parallels the history of programmable ROMs. Due to programming limitations, early PLAs were available only in mask-programmed versions. Just as with a ROM, a logic designer would indicate on the vendor's PLA AND/OR logic map, where the desired connections were to be made. The vendor would then generate a custom mask for the PLA to implant the customer's logic.

The conventional PLA, as shown in Fig. 1.1, consists of input lines, bit lines (i.e., output lines of the input buffers which provide the true and the complement terms of the input variables), product lines, and output lines. The portion of the PLA consisting of the intersections of bit lines and product lines is called the AND plane.

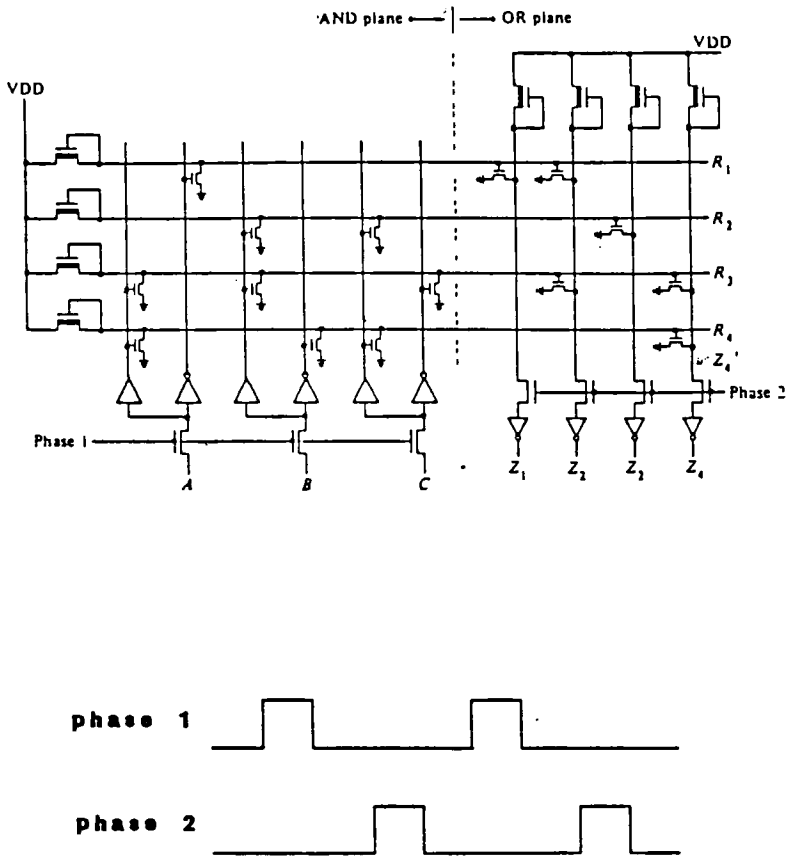


Fig. 1.1 The conventional PLA using a two phase clock

Similarly, that portion of the PLA consisting of the intersection of the product lines and the output lines is called the OR plane.

A particular combinational logic function is realized with a PLA by assigning pull-down devices at desired intersections in the AND plane and the OR plane. The pull-down device in the PLA represents the programmed code, called the PLA *personality*, which specifies boolean expressions as the sum of products of *literals*. The PLA's input lines are driven by either inverting or non-inverting superbuffers, which are controlled by a pass transistor clocked on *phase 1*. Each product line carries the NOR combination of all input signals that lead to the gate of the transistors attached to it. In a like manner, each output in the OR plane is the NOR combination of all the product lines connected to it and leads to an inverting output buffer through a pass transistor clocked on *phase 2*.

1.1.1 ROMs & PROMs

The first devices were one time Programmable Logic Arrays which could be programmed in much the same way as a Programmable Read Only Memory(PROM) is programmed: built-in fuses were blown by a special programming machine to implant programs or data into the PLA. NOR-NOR logic is used to create the AND plane and the OR plane architecture of an NMOS type Programmable Logic Array. It uses the sum of product forms to implement boolean functions. The advantages of a PLA are its design simplicity and the regularity of its structure, both of which reduce the complexity of logic function sections in integrated circuit design. On the

other hand, its main disadvantage is that PLAs are non-erasable after they have been programmed by the custom-made mask.

1.1.2 EPROMs & EEPROMs

The Eraseable Programmable Read Only Memory(EPROM) was the next development in VLSI technology, making it possible to reprogram the personalities of the chip. However, an EPROM has to be programmed in two steps. First, an EPROM must be bathed under the utraviolet light for about twenty minutes, to place the device in its erased or initialized state. It can then be reprogrammed in much the same way as a PROM was. As the VLSI technology moved on, the Electrically Eraseable Programmable Read Only Memory(EEPROM) was invented. The EEPROM employs a much simpler way of implementing logic functions in integrated circuits. This chip is similar to an EPROM except that the does not have the clear window; ultraviolet radiation is not required for erasure. Instead, a special voltage signal applied for specific times can erase an EEPROM, and this voltage can often be applied from within the host system.

1.2 Electrical Alterable Programmable Logic Array

An electrically alterable programmable logic array (EAPLA) has been designed by Wood[19] and Fong[6]. The use of the EEPROM technology in a PLA results from the idea of combining the electrically alterable nonvolatile memory devices within the PLA design. Although the EAPLA chip is not as dense as a PLA chip, the EAPLA

reduces the design effort considerably in the application field. The chip can be programmed or erased by applying high voltage to the control pin. The drawback of this scheme is the need to disconnect the chip from the socket or the board in order to reprogram the personality matrix. Removing a soldered chip from a circuit board almost always guarantees that it will be damaged.

1.3 Alterable Programmable Logic Array

Through a different approach, a RAM-like Alterable Programmable Logic Array (APLA) was designed by Marchand[16]. In Marchand's scheme, the APLA performs the same function as a standard programmable logic array. The APLA is programmed by writing the personality into a storage cell at each node of the AND/OR plane through an associated peripheral register. Unlike the EAPLA, in which a high voltage must be applied to change its personality, the Alterable PLA can be reprogrammed any number of times by rewriting the data into the storage cells at normal logic levels. Fig. 1.2 shows the typical arrangement of the APLA, which is composed of two parts: (a) the basic PLA function with AND and OR planes, and (b) the control logic needed to operate the "alterable" functions and to maintain the information in the dynamic storage cell.

Even though the writing scheme used to program the device is easier than its predecessors, the APLA still suffers from the long time required to alter the personality. In Fig. 1.3, an APLA with 22 inputs, 22 outputs, and 64 product terms[16] requires 2.2 ms for the entire personality to be written into the AND/OR

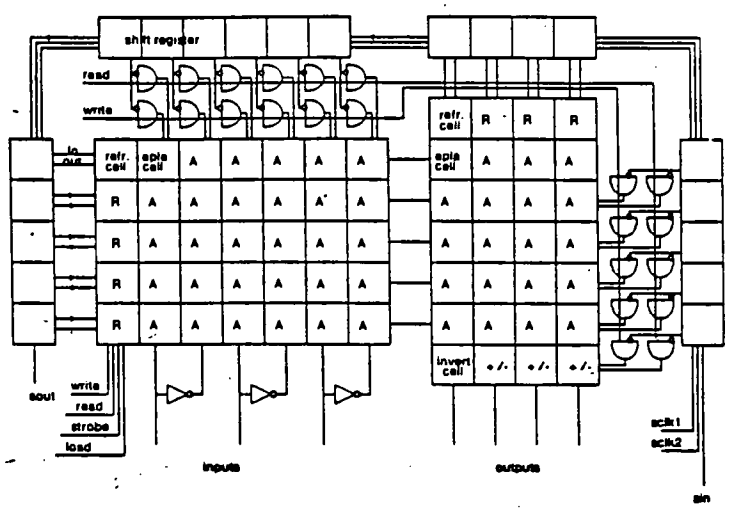


Fig. 1.2 An APLA structure

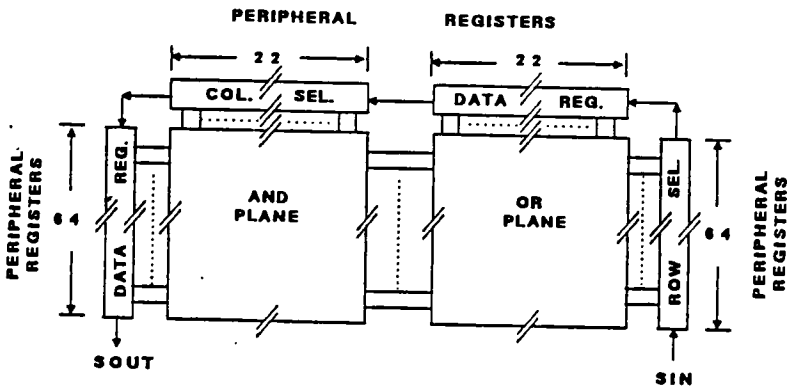


Fig. 1.3 An abstract block diagram
of a APLA

plane. Even writing a single row or column takes 34.4 μs. Furthermore, the write time grows linearly with respect to the size of the circuit. This means that the larger the configuration becomes, the longer the writing time will be. Another disadvantage is that the user must be aware of the locations of free or unprogrammed storage cells and must write the appropriate data into the correct peripheral registers, before transferring the data to the cell itself. Also, prior to updating the personality, the corresponding address of the row or the column with that personality needs to be known. It is inconvenient for a user to keep a table showing all occupied rows or columns and their associated personalities. Even using a table look-up instead of an exhaustive search is still time consuming.

1.4 Writable Programmable Logic Array

In order to avoid the problems of an APLA, such as the long writing time, the need to know the address of a free location etc., we will design a Writable Programmable Logic Array(WPLA) which employs a pseudorandom addressing scheme for fast WRITE and easy ERASE operations. To achieve fast access operations, the WPLA will use a *content addressable scheme* for SEARCH and READ operations. The end user does not need to know the physical address of a free location. Instead the WPLA will search the memory and allocate a free space for the user. In addition to the functional specification, the testability of the WPLA is considered in our design. Test vectors will be used to check the PLA circuit. As a result, improved testability will provide benefits, both in the design phase and in the field applications.

1.4.1 Improving Writing Speed

One of the core ideas of the WPLA is that the number of input variables or output variables of a PLA depends on system specification, so that segmenting the personalities is required. To update a memory cell, the system bus supplies data to the input variable dynamic latches by multiplexing the device input lines. The WRITE operation depends upon the number of multiplexed variables in the personalities instead of the number of peripheral registers as in the case of an APLA. The specific objective is to improve the WRITE speed by achieving a rate at least ten times faster than Marchand's scheme.

1.4.2 Random Selection of Free Locations

Each row is associated with a status tag which indicates whether the memory cells in that row are free or occupied. A row with a free tag will gate the writing clock to allow the personality to be written into its memory cells in the AND/OR planes. However, if more than one status tag has a free flag, the WRITE mechanism should be able to distinguish priorities and prevent multiple WRITES. The purpose is to omit the need to specify a free location, thus causing this chip work in a friendly manner.

1.4.3 Flexible Erasure and Updating Abilities

A search by content operation, like that in the content addressable memory(CAM), is conducted simultaneously on the AND/OR planes. A target tag is attached to each row and is set if the corresponding row is targeted by the searching key or by the same personality. Through target tags, erasing and updating personalities can be done fast enough and, further, the target tags help make the WPLA fault tolerant by using redundant rows. The objective is to implement the SEARCH operation and make it easy to operate. In addition, it is possible to examine all the personalities or just those associated with the targeted row.

1.5 The Overview of Each Chapter

This thesis consists of eight chapters. In chapter one, the motivation and objectives are presented by introducing ROM, EPROM, EEPROM, and APLA devices. In chapter two, the architecture of a WPLA is described, including the pseudorandom addressing scheme, the multiplexing input mechanism, and the design for testability etc.. The functions of the proposed WPLA are also summarized by each operation mode. In chapter three and four, the data portion and the control portion of the WPLA are discussed through their associated schematics and detailed timing diagram. In chapter five, the implementation is described, and the functional and timing verification is also evaluated in detail. In chapter six, a performance comparison among WPLA, PLA, APLA, and CAM is presented. In chapter seven, six application examples are presented. Finally, in chapter eight, the conclusion

and the trade-offs in the design of implementation and various application points of view are summarized. The performance impact of future VLSI technology improvements are also evaluated.

2. ARCHITECTURE DESCRIPTION

A Writable Programmable Logic Array basically uses volatile memory cells in the AND/OR planes as programmable nodes instead of using fixed pull-down devices as in the conventional PLA. The content or personality in the memory cell can be programmed, updated, and examined through versatile and powerful functions, such as WRITE, SEARCH, READ, and SCAN operations. The architecture of a WPLA and its associated operations will be described in the following subsections.

2.1 An Abstract Block Diagram

The proposed configuration of a WPLA is given in Fig.2.1. The AND plane and the OR plane perform the NOR-NOR operation like a conventional PLA. Using a non-overlapping two phase clock scheme, the primary inputs provide *literals* to the AND plane through input buffers. The outputs of the OR plane, representing the stored boolean functions, are located in the output buffers.

The programmable nodes in both planes are constructed on the basis of a *pseudo-static RAM cell.* The PLA *personality* is programmed into each memory cell through the input buffer and the master driver. The system data bus feeds a part of the primary input lines into the master buffer for formatting the desired personality or the search

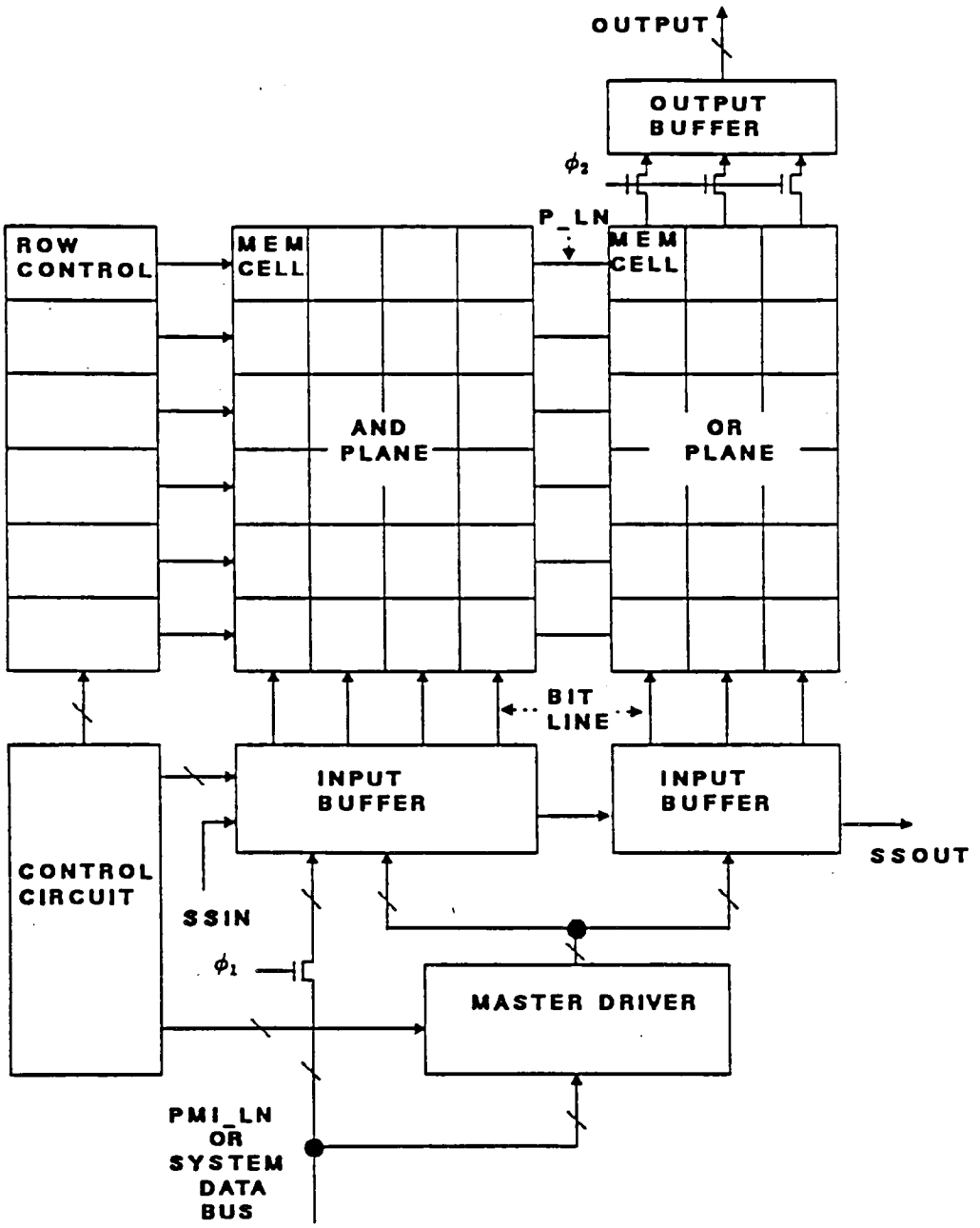


Fig. 2.1 AN ABSTRACT BLOCK DIAGRAM

argument. The row control circuit selects a vacant location and gates the write clock to store the prepared personality into the addressed location. Alternatively, it can select a targeted row, gate the read clock, and unload the personalities into the input buffers. The personalities are then scanned out from the input buffers for examination. Moreover, a priority mechanism in the row control circuit allows no multiple WRITES (READS) during the WRITE (READ) operation. The control circuit interfaces with the external mode control signals(CBT0, CBT1, CBT2) and the phase clocks(PHE1, PHE2) in Fig. 2.13, and generates internal control signals to activate the associated operations.

Detailed operation will be described in section 2.4 after NORMAL, WRITE, SEARCH schemes are described in the following section.

2.2 The Architecture of a WPLA

A row of a WPLA indicates the memory cells on the AND plane and the OR plane concatenated through a single product line, P_LN, as shown in Fig.2.2. Each row has its own row control circuit.

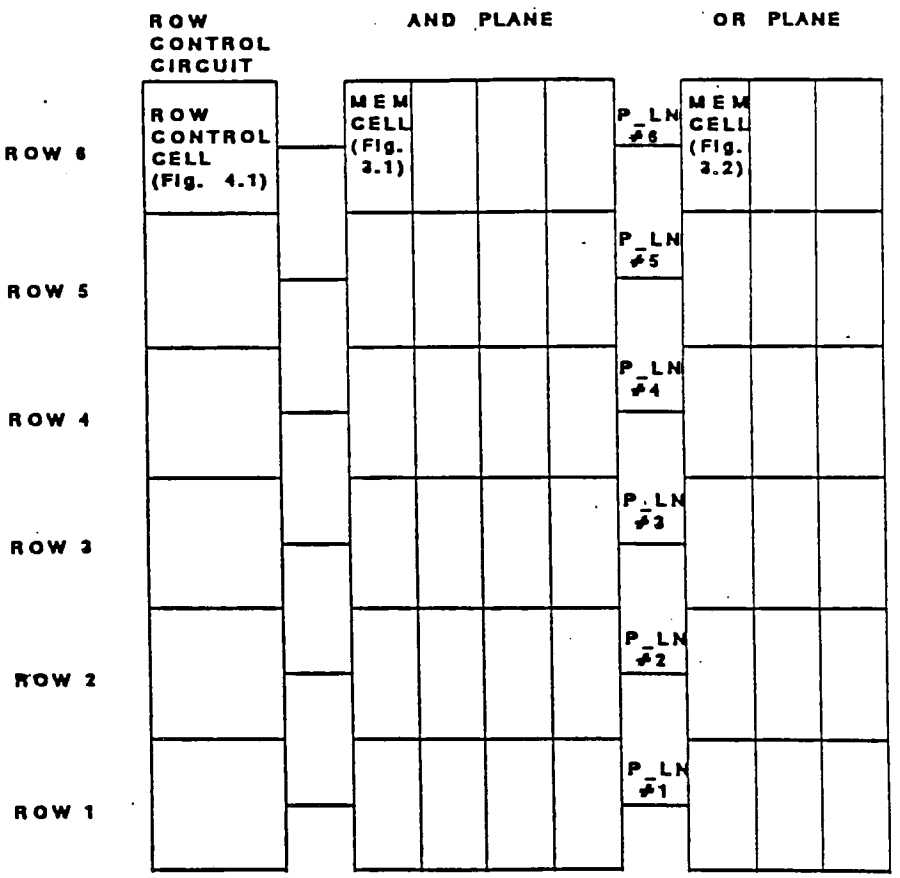


Fig 2.2 A WPLA WITH SIX ROWS

2.2.1 Storage Pattern for the PLA Personality

On the AND plane, a pair of memory cells is used to store the true term and its complement of a personality bit, or both cells store "logic 0" to represent "don't care". In the OR plane only a single memory cell is used to store a personality bit, with either "logic 1" or "logic 0". The basic storage pattern is similar to one used in a PLA using two nodes for each variable and one node for each output line. The pattern of personalities on all the programmed nodes in the AND and OR planes represents a boolean function in the sum of product form and is referred to as the *personality matrix*.

2.2.2 Multiplexing Input Scheme

The input buffer attached to the AND and OR planes shown in Fig.2.3 is used to latch the formatted data provided by a master driver and then activates the AND and OR planes. Since the number of inputs and outputs of a WPLA is usually larger than the number of system data bus lines, a single row of the personality matrix or a search argument can not be loaded into the input buffers without partitioning.

In Fig.2.3, a multiplexer is used to input the partition of personalities(search argument) and is implemented by a set a pass transistors and multiplexing phase clock Ψ_i. The i^{th} partition of the personalities(the search argument) is latched into

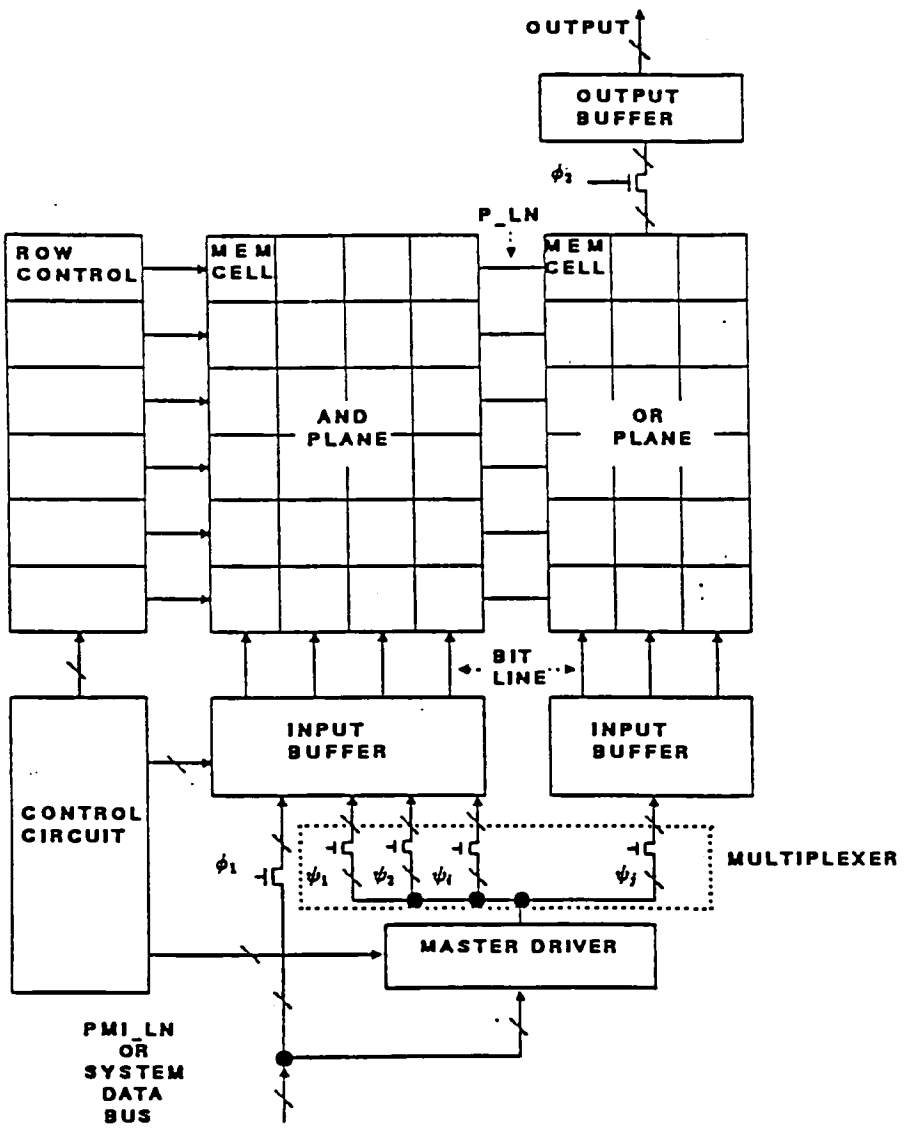


Fig. 2.2 MULTIPLEXING INPUT SCHEME

the correct input buffer by clocking its associated phase clock Ψ_i. If j indicates the minimum number of clock phase for this multiplexing operation, then.

$$j = [(n+q)/m]$$

Where

n: the number of the primary lines in the AND plane

q: the number of the primary lines in the OR plane

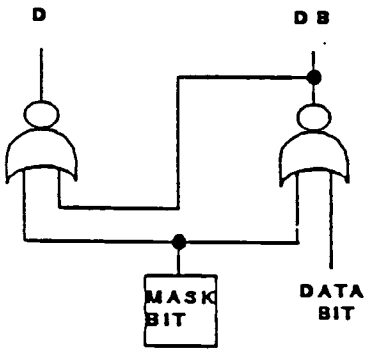
m: the number of the system data bus lines

After the j^{th} clock cycle, a single row of the personality matrix or a complete search argument is finally latched into the input buffers prior to the coming WRITE or SEARCH operation.

2.2.3 Data Formatting

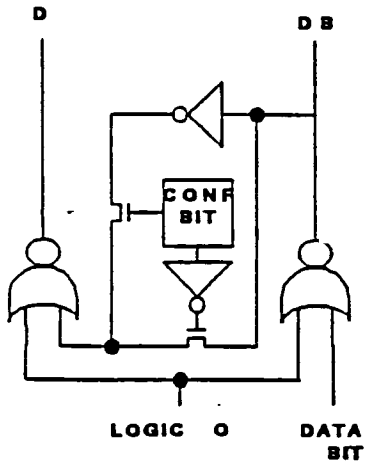
The purpose of data formatting in the master buffer comes from the need to encode the personality (search argument) during the WRITE or the SEARCH operations. Since the data format to be prepared for the OR plane is either "logic 1" or "logic 0", the required data format, i.e. the code for these two operations, is dominated by the data requirement in the AND plane and is illustrated in Fig.2.4.

It was difficult to design a data formatter circuit without using register storage which increases the area of the overall device. A dynamic storage circuit, as described in



MASK BIT	DATA BIT	D	DB	STATE REPRESENTED
0	0	0	1	logic 0
0	1	1	0	logic 1
1	x	0	0	"Don't care"

(a) WRITE OPERATION



MASK BIT	DATA BIT	D	DB	STATE REPRESENTED
0	0	0	1	logic 0
0	1	1	0	logic 1
1	1	0	0	"Don't care"
1	0	1	1	"Mask a bit"

(b) SEARCH OPERATION

Fig. 2.4 DATA FORMAT FOR THE AND PLANE DURING WRITE OR SEARCH OPERATION

chapter 3.4, was used to minimize area. The advantages of this arrangement are attributed to: (1) no modification in the circuit of the input buffer, (2) the centralization of the formatting capability, and (3) the compact layout size between the input buffer and the memory.

The formatting function is basically constructed by dual NOR gates, a mask bit and a configuration bit as shown in Fig.2.4. The *mask bit* contributes to formatting the data to be written, and the *configuration bit* contributes to formatting a search argument. The formatted data for WRITE and SEARCH operations are needed not only in a pair of complement states, but also in both "logic 0" and "logic 1".

In the WRITE operation(Fig.2.4.a), if the mask bit is "logic 0", the same state as the data bit will appear on the D, and its complement will appear on DB. However, if the mask bit is "logic 1", then "logic 0" will appear on both D and DB to represent a *don't care* state.

In the SEARCH operation(Fig. 2.4.b), If the configuration bit is "logic 0", D and DB will be in the complement states and D and the data bit have the same state. If the configuration is "logic 1", an additional inverter is added between two NOR gates to generate the complement state of the data bit on both D and DB. The mask or configuration bit is latched before the data bit is fed into the master buffer. Then, the data bit will determine the personality of the search argument. If the configuration

bit is logic 1, then the logic value of D and DB either the *don't care* state(both "logic 0") or the *masking bit* state(both "logic 1").

2.2.4 The Data Path

The data path for NORMAL operation in Fig.2.5 is similar to the conventional PLA as mentioned before. The primary input and the output are separately latched by ϕ_1 and ϕ_2. During the WRITE operation, a data path for a single bit in the AND plane and in the OR plane is illustrated in Fig.2.6. Ψ_i is a multiplexing clock phase which is used to latch the formatted data. In the AND plane, D and DB are stored into cell 1 and cell 2. In the OR plane, only the formatted data D goes through the inverting buffer and is stored into cell 3.

For the SEARCH operation (an abstract diagram is shown in Fig.2.7) the formatted data, as a search argument, is processed by the master driver and latched into the input buffer by the multiplexing clock phase Ψi. Next, the search argument in ~BIT_LN and BIT_LN is applied to the comparison circuit of the memory cells. P_LN is kept in "logic 1" only if the personality in node ~Q of the storage cells and the search argument are in opposite states; the results of this comparison P_LN(high) means "match" with the particular argument. The comparison circuit in the OR plane is different from that in the AND plane. A single storage cell is only reserved

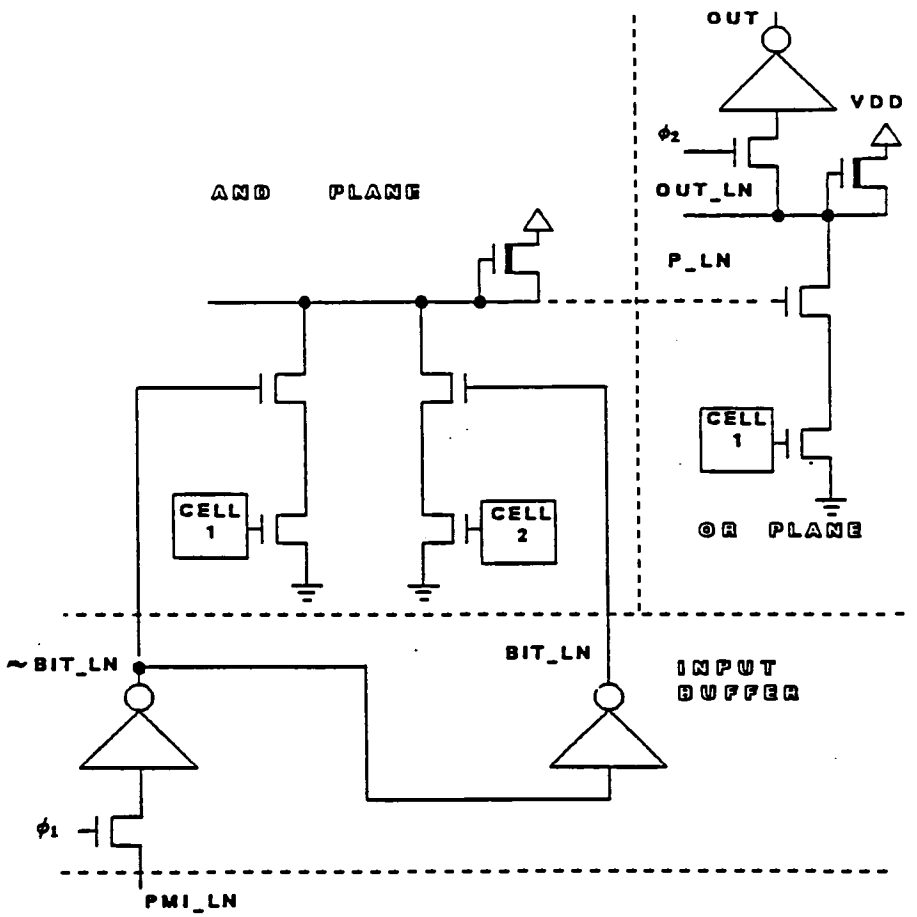


Fig. 2.5 DATA PATH FOR NORMAL OPERATION

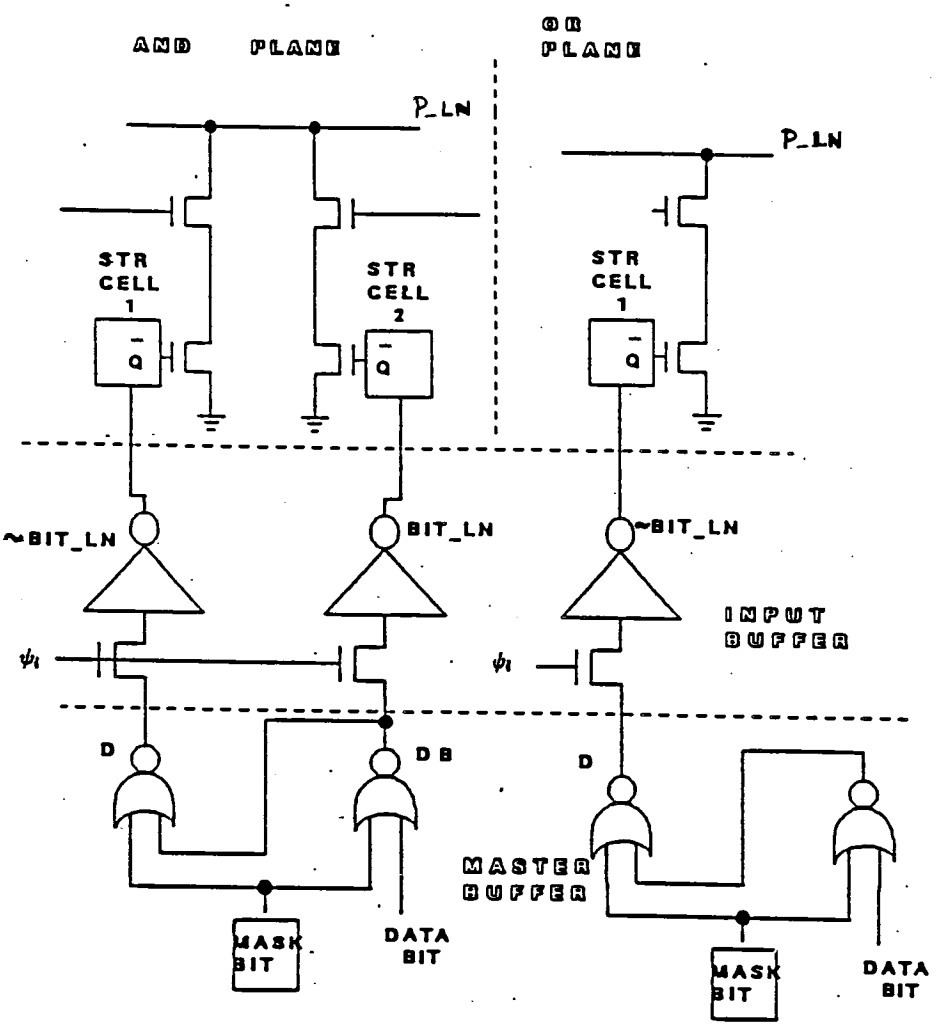


Fig. 2.8 DATA PATH FOR WRITE OPERATION

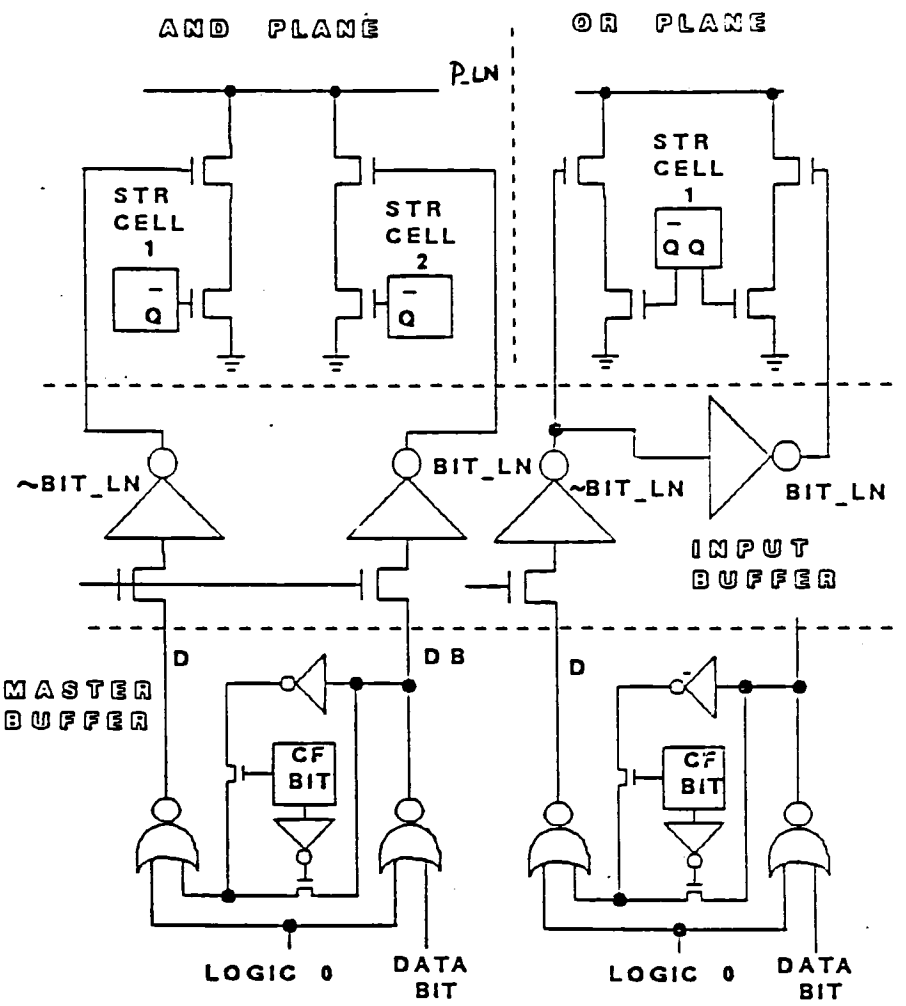


Fig. 2.7 DATA PATH FOR SEARCH OPERATION

for a bit in the OR plane, while the comparison of the "logic 0" state in the storage cell needs more complicated circuitry.

2.2.5 Pseudorandom Addressing Scheme

A single row control circuit in Fig.2.8 is composed of a *status tag*, a *target tag*, and a *priority circuit* with some control gates. The status tag is set to logic 1 initially through a master reset signal (RS) in order to indicate its associated row as "vacant". Whenever the row is gating the WRITE clock, the trailing edge of that clock will reset the status tag of this particular row to logic 0. A "logic 0" in status tag represents "occupied" and inhibits any activations by the following WRITE clocks.

The pseudorandom addressing scheme in Fig.2.9, implemented by a priority chain, is used to select a vacant row to write in the personalities. The physical order of the priority circuit chain in Fig.2.9(b) is arranged to let the lowest vacant row be written first. Since the initialization(Fig.2.9.a) sets all status tags to "vacant"(logic 1), the initial consecutive WRITE operations (Fig. 2.9.c) will store the personalities into row 1, row 2, and so on.

Next, erase operations (Fig.2.9.d) will release some vacant rows. Thus, the future personalities to be written will not constructed in the previous sequential order. Instead, the current lowest vacancy, addressed by the priority chain, will be written

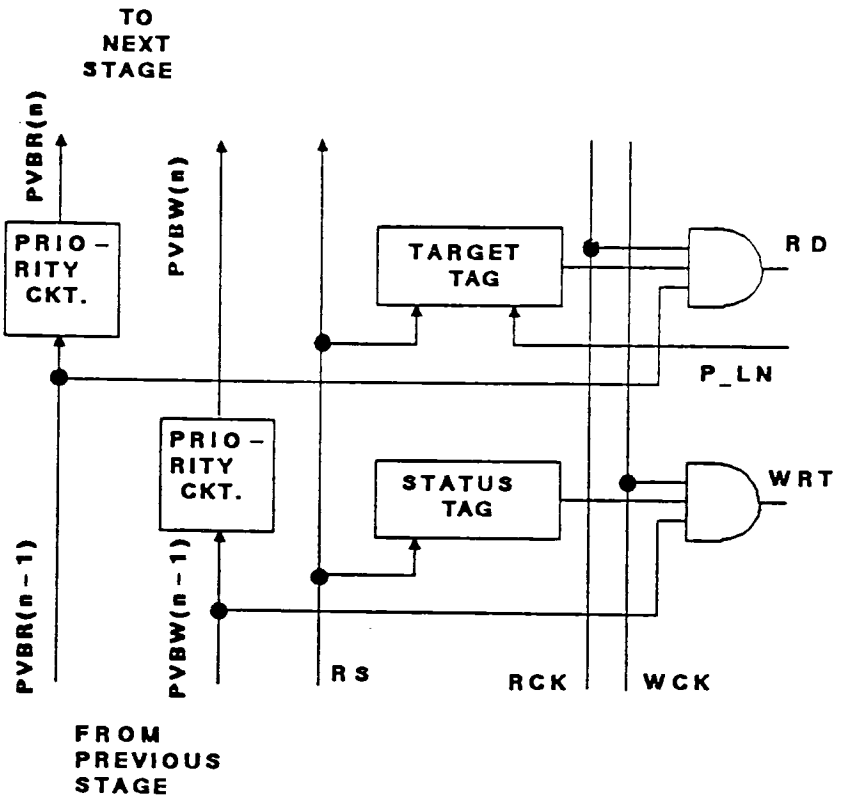


Fig. 2.8 A ROW CONTROL CIRCUIT

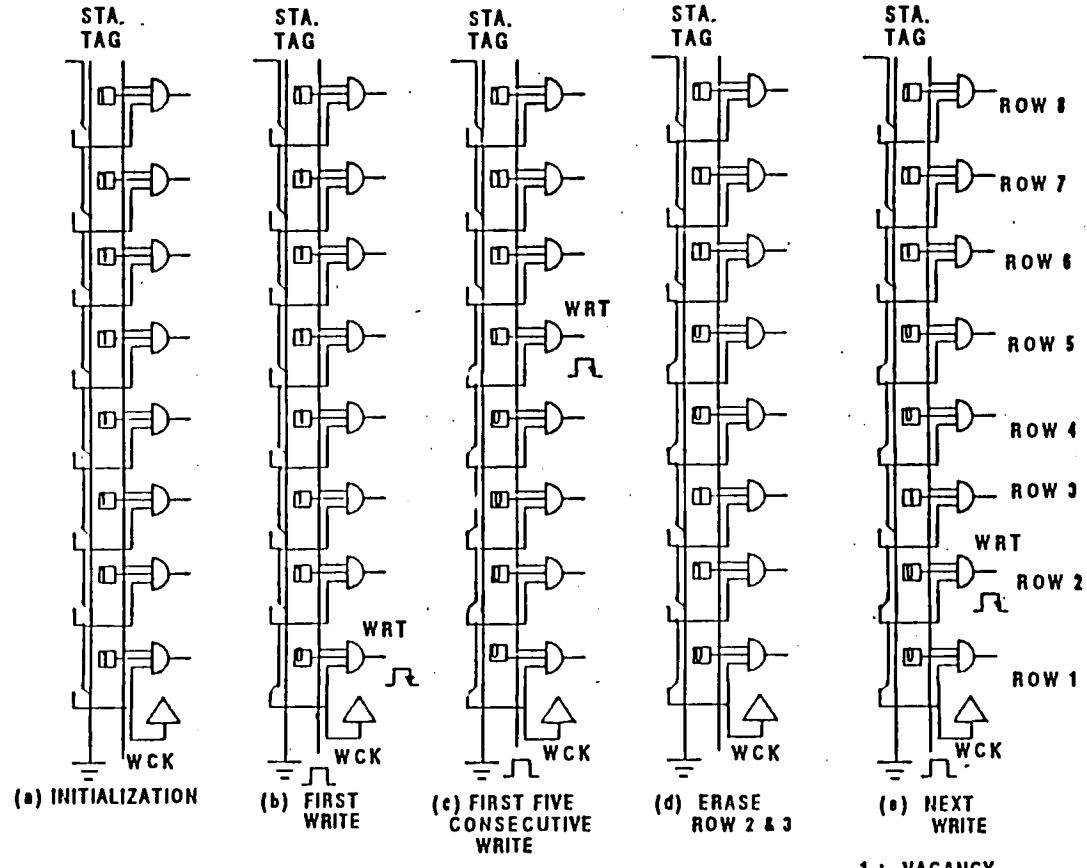


Fig. 2.9 PSEUDORANDOM ADDRESSING SCHEME

next (Fig.2.9.e). The more the erase operations are mingled with the WRITE operations, the more random the location of the personalities become. However, the upper most vacant row always has the least priority, and therefore accepts the fewest WRITE operations. We label this kind of architecture as a *pseudorandom addressing scheme*.

2.2.6 The ERASE and READ Scheme

The target tags in the row control circuitry, accompanied with another priority circuit chain, are used to control the ERASE and READ operations. Initially all target tags are reset to "logic 0" to indicate the rows as "untargeted".

Before the ERASE and READ operation, some of rows have already been written, and their associated status tags have been set to "occupied" (Fig.2.10.a). Accompanying the SEARCH operation, and through a strobe pulse(STB), the tag(s) of the associated targeted row(s) is (are) set to logic 1, through the P_LN(s), representing a "match" (Fig.2.10.b). To erase the occupied rows we easily shift the target tags to the status tags. The erased row(s) need not be cleared and will be ignored by the NORMAL and SEARCH operations until we write it(them) again.

For the READ operation shown in Fig.2.11, the target tags set by the previous SEARCH operation,(Fig.2.11.a and Fig. 2.11.b) can accept a READ clock and gate the

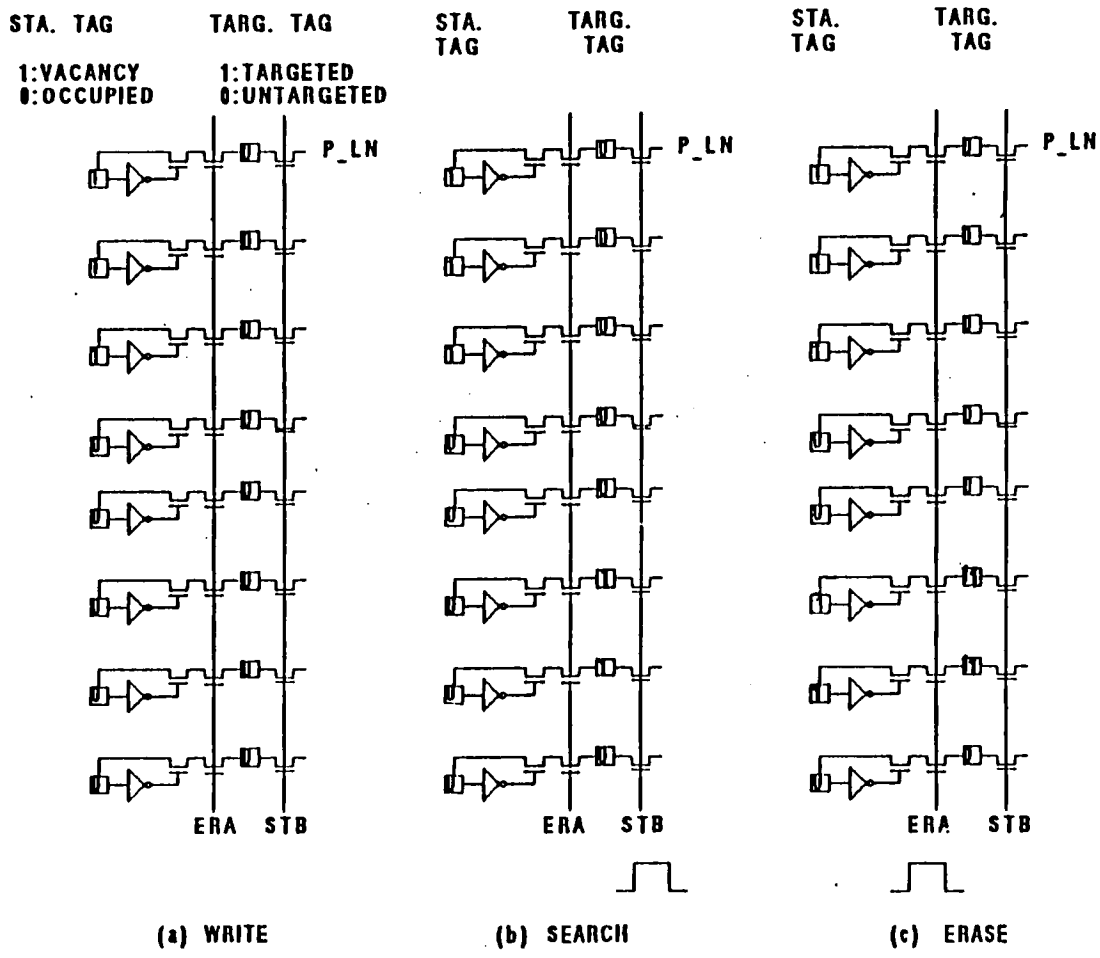


Fig 2.10 ERASE SCHEME

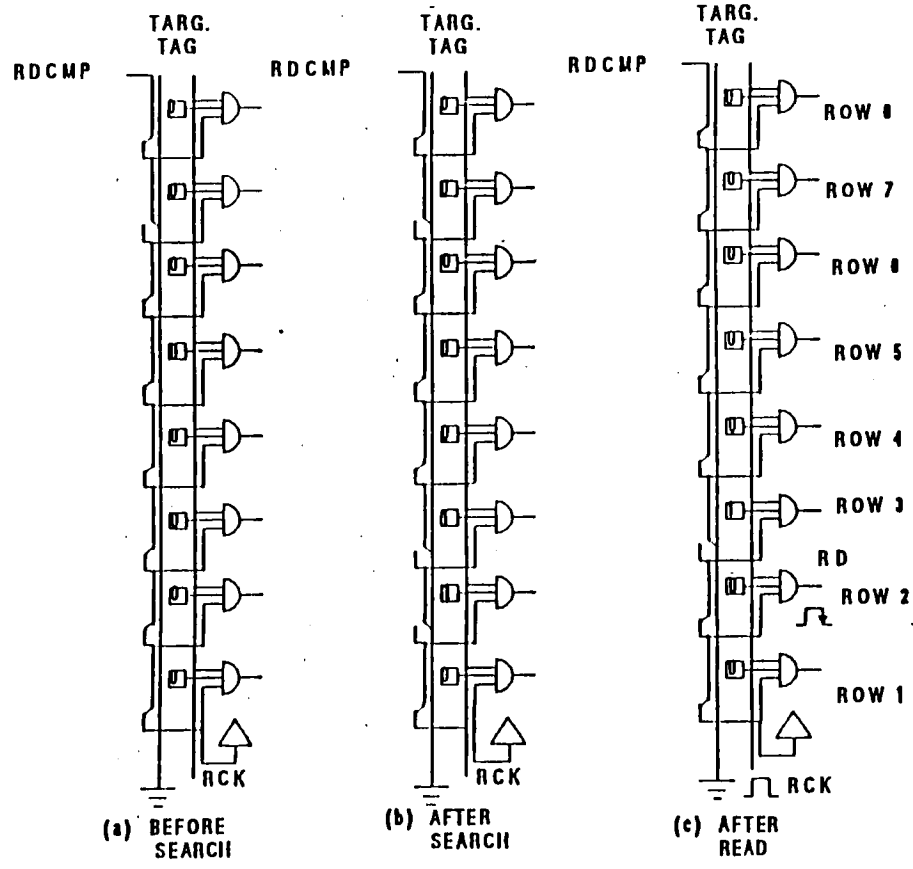


Fig. 2.11 READ SCHEME

1 : TARGETED
0 : UNTARGETED

data from one of targeted rows to the input buffers (Fig. 2.11.c). The priority chain also arranges the lowest targeted row to be read first. The trailing edge of the READ clock will reset the target to "untargeted"(logic 0) after its personalities are gated into the input buffers.

2.2.7 Design for Testability

During the SCAN or TEST operation, the input buffers in the AND and OR planes are configured into a serial shift register chain as shown in Fig.2.12.a. The personalities, which have been unloaded into the input buffer during the previous READ operation, are scanned out by clocking SSN1 and SSN2. SSOUT, the output pin of this shift register chain, can be used to examine the personalities in the series.

Similarly, all status tags and target tags in the row control circuit can also be chained together into a long shift register chain shown in Fig.2.12.b. In the TEST mode, the states of the status tag and the target tag for each control circuit can be observed from the output pin, SOUT, of this shift register chain. Moreover, the stand-alone test for the above two register chains can also be conducted simultaneously in the TEST mode. If the test vectors are serially scanned from the input pin SSIN and SIN of both shift register chains, then their test responses can be examined from the output pins SSOUT and SOUT.

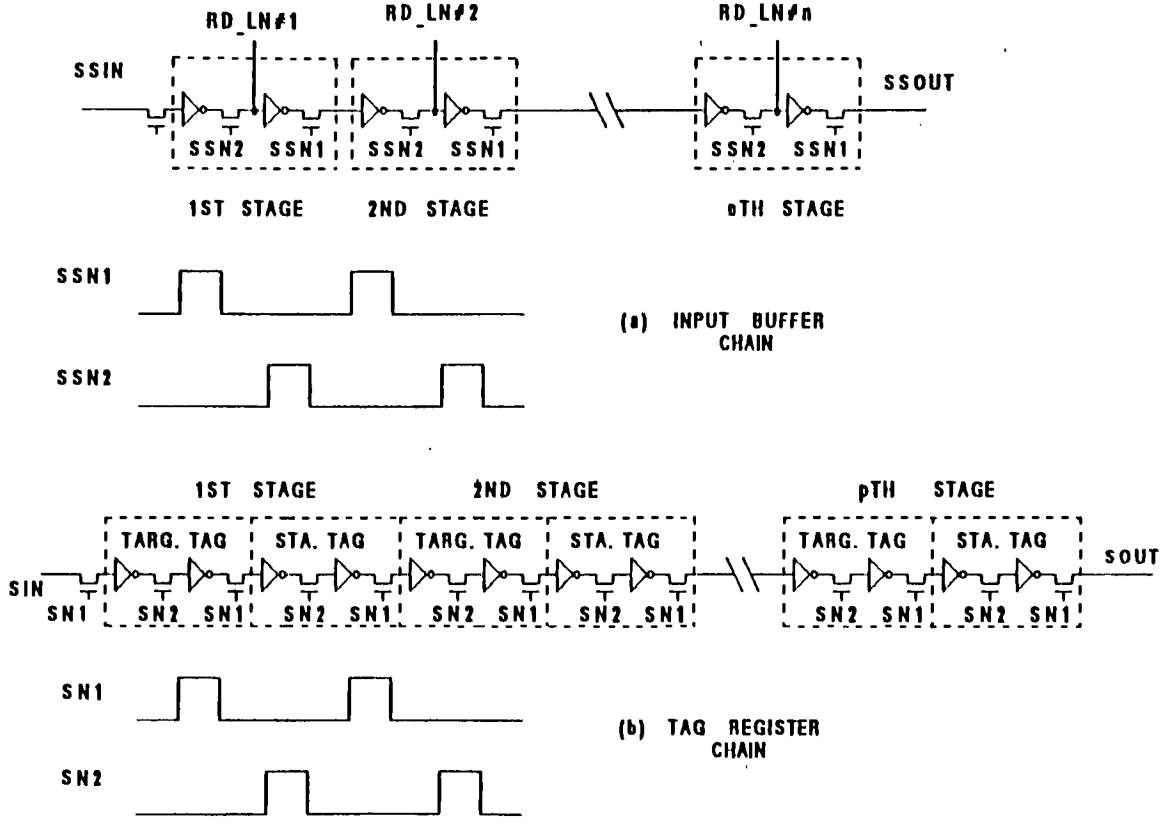


Fig. 2.12 SHIFT REGISTER CHAINS

Because the WPLA can be used to do stand-alone testing of the input buffers or of the tag registers, these features make it easier for the WPLA to be tested,

2.3 The extended Block Diagram of a WPLA

The extended block diagram of the proposed WPLA, given in Fig.2.13, shows the data and control connections among the modules and illustrates the overall architecture and components discussed in the last section. The detailed implementation of each module will be described in chapter three and four.

2.4 The Configuration of a Writable Programmable Logic Array

We will implement a WPLA with the same configuration as Marchand's chip(Fig.1.2) which has 22 inputs, 22 outputs, and 64 product terms. The bus-wide master drivers in the WPLA are constructed in a 16 bit structure, which provides for interface with a 16 bit system data bus. Therefore, the input buffers for the AND/OR planes total 44 stages, and are divided into 3 partitions of 16 stages, 16 stages, and 12 stages, all of which are used for latching data by multiplexing from the master drivers. The conceptual approach and objectives involved are described in the subsections that follow. A summary of the expected goals of the WPLA is:

(1) Fast writing speed.

(2) Random selection of free locations for the WRITE operation.

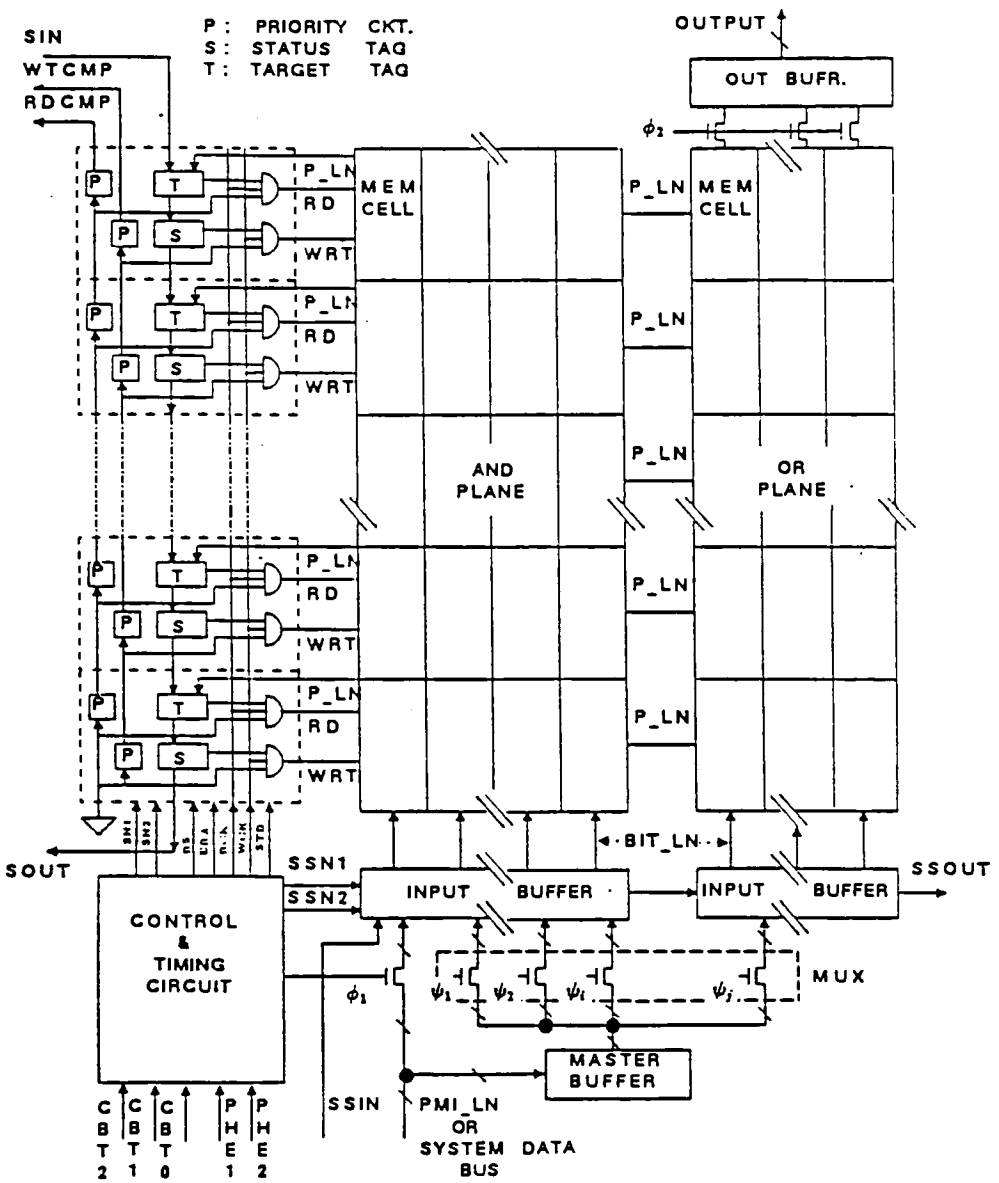


Fig. 2.13 THE EXTENDED BLOCK DIAGRAM OF WPLA

(3) A flexible content addressable search to erase and update WPLA.

2.5 Operational Modes

A WPLA should provide good functionality and fast operation for users. In this design there are eight basic operations. Some sequences of basic operations are meaningless or illegal to execute in a WPLA chip. A legal sequence of basic operations, called a *macro operation*, combines more than one basic operation. We shall first describe the basic ideas behind those operations, then describe two macro operations to illustrate how WPLA works.

2.5.1 Basic Operations

Normal

Feed the input from the primary input lines, activate the AND plane, and latch the outputs from the OR plane through two non-overlapping clock phases. (Performs the same function as a conventional PLA.)

Master Reset

Activate the reset pulse(RS) to set all the status tags to vacant (logic 1) and clear all the target tags to untargeted (logic 0).

Write

Load the personalities into the input buffers by multiplexing. Select a vacant row through pseudorandom addressing scheme. Write the personalities into the addressed row.

Search

Load the search argument or the search key into the input buffers by multiplexing. Strobe the targeted row(s) through the primary input line and set the associated target tag(s).

Erase

Activate the erase pulse(ERA) to shift the target tags of all occupied rows to the status registers.

Read

Following the priority chain, select a targeted row. Unload the personalities of this targeted row into the input buffers.

Scan

By specifing the contains of WPLA, the personalities of that row will be scanned out into the input buffers, which are reconfigured into a shift register chain.

Test

Test the two shift register chains which are configured from the input buffers and the tag registers, respectively. Apply test vectors to the control circuit and the input buffers and examine the test responses from their output ports through clocking two non-overlapping phase clocks.

2.5.2 Macro Operations

Search then Erase

Search then erase the occupied row(s) which have the specified search argument or the specified search key.

Search then Read then Scan

Serially examine those personalities which match on the search argument or on the search key in one of the targeted rows. If a row matches a key, then the contents of that row will be read.

3. THE DATA PORTION OF A WPLA

The circuitry of a WPLA can be divided into the data portion and the control portion. In this chapter, the circuitry of the data portion will be described. The associated control and timing portion will be discussed in chapter 4.

The data portion of the WPLA consists of (a) the basic memory cell, (b) the input buffer, (c) the output buffer, (d) the master driver, and (e) the glue logic circuit.

3.1 The Basic Memory Cell

The basic WPLA memory cell includes a *pseudo-static* memory cell and a comparison circuit. We chose the pseudo-static memory cell, instead of the dynamic one, to achieve fast operation, to avoid complicated refresh control timing, and to provide stable driving capability. Both the AND planes and the OR planes of the WPLA use the same storage cell, but each employs a different comparison scheme.

The memory cell in the AND plane is shown in Fig.3.1 and is composed of nine transistors. The two cascading inverters are formed by Q3-Q6 whereas Q2 provides the feedback path to construct a pseudo-static memory cell. The data on BIT_LN(~BIT_LN), coming from the input buffer, is then gated by Q1 as the

personality during the WRITE operation. The control signals WRT and ~(WRT + PH2B)on the gates of Q1 and Q2 assure that there is no data conflict or any ambiguous states on the input of the memory cell. During the READ operation, Q9 is turned on in order to gate the stored data to line ~RD_LN (RD_LN).

The comparison circuit is implemented by Q7 and Q8(shown in Fig.3.1). If the previous data stored in the memory cell at node ~Q is opposite of the current states on line ~BIT_LN (BIT_LN), the P_LN is kept high, which indicates a match. In the NORMAL mode, the state on the P_LN is utilized to activate the OR plane. Similarly, in the SEARCH mode, the state of P_LN is gated to the target tag through a strobe pulse(STB).

The memory cell in the OR plane shown in Fig.3.2 consist of 13 transistors. Q1-Q6 construct a pseudo-static memory cell. Q9 is used to gate the stored data to ~RD_LN for examining purposes. The comparison function in the NORMAL mode is performed through Q7 and Q8. In a "match" state, the OUT_LN is high, and the state of P_LN and the stored data on the node ~Q are opposite.

In order to provide the content addressable function on this plane, two additional comparison circuits, accompanied by BIT_LN and CMP_LN, are needed. CMP_LN is connected to all memory cells in the same row of the OR plane. It is charged (or discharged) to the same state as the P_LN when either the path of Q10

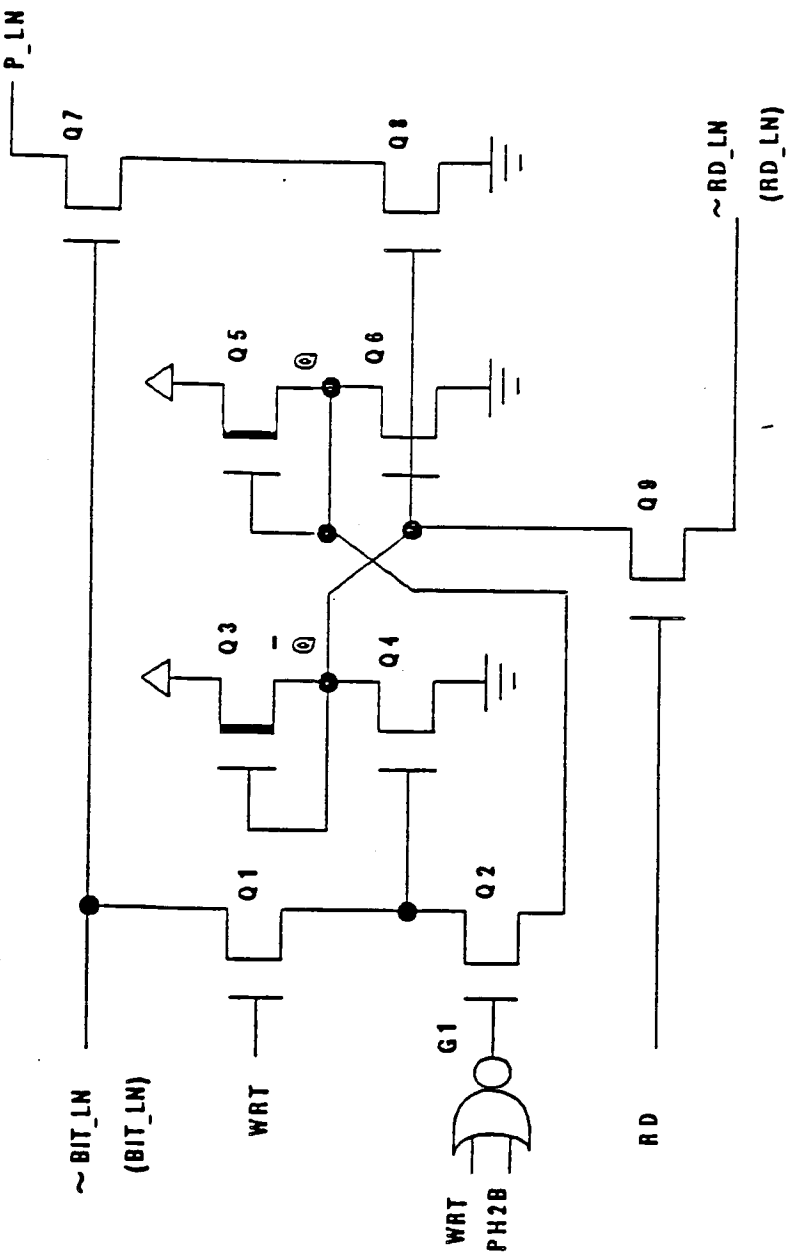
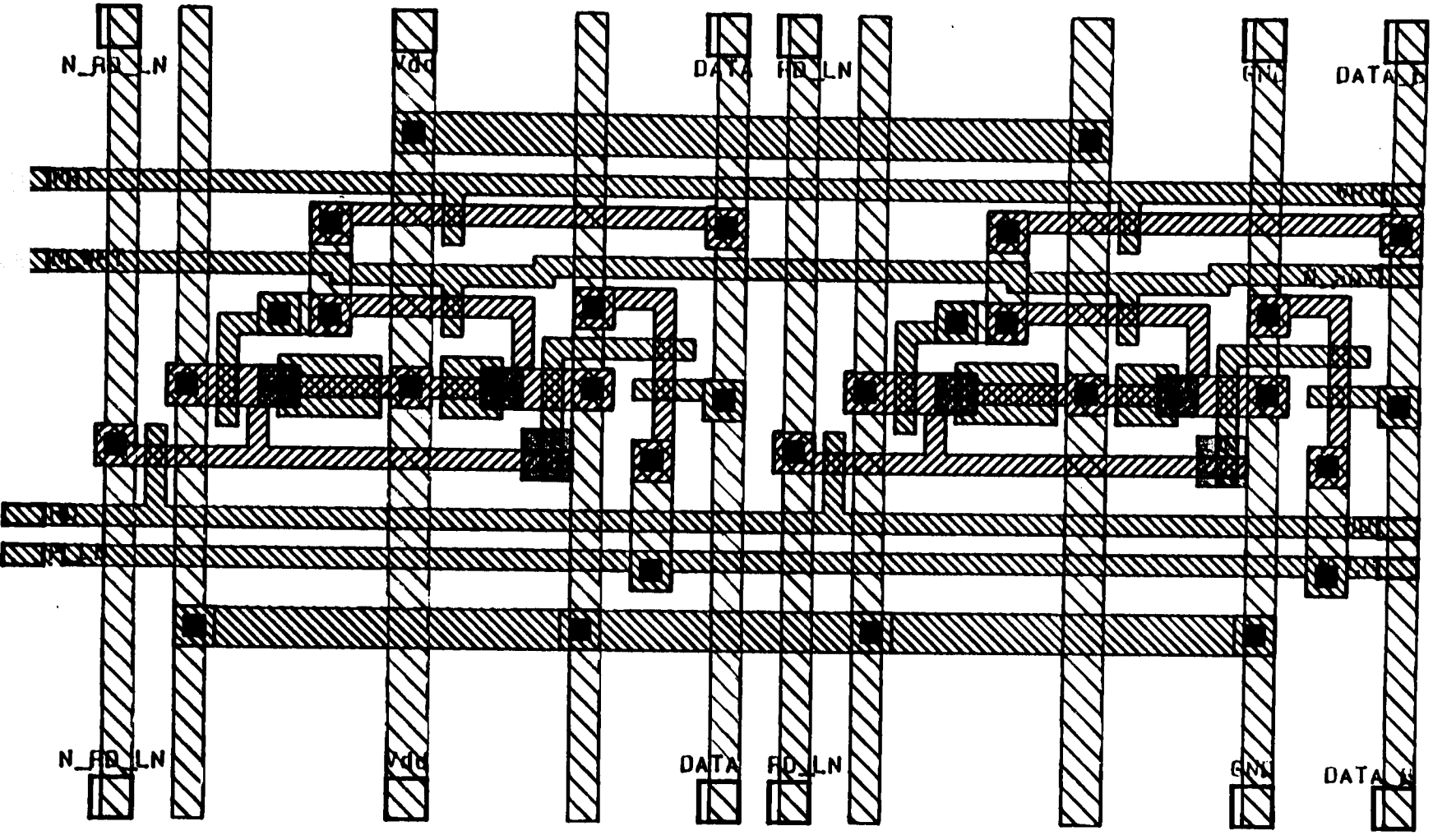


Fig. 2.1 A MEMORY CELL OF AND PLANE



AND_CELL

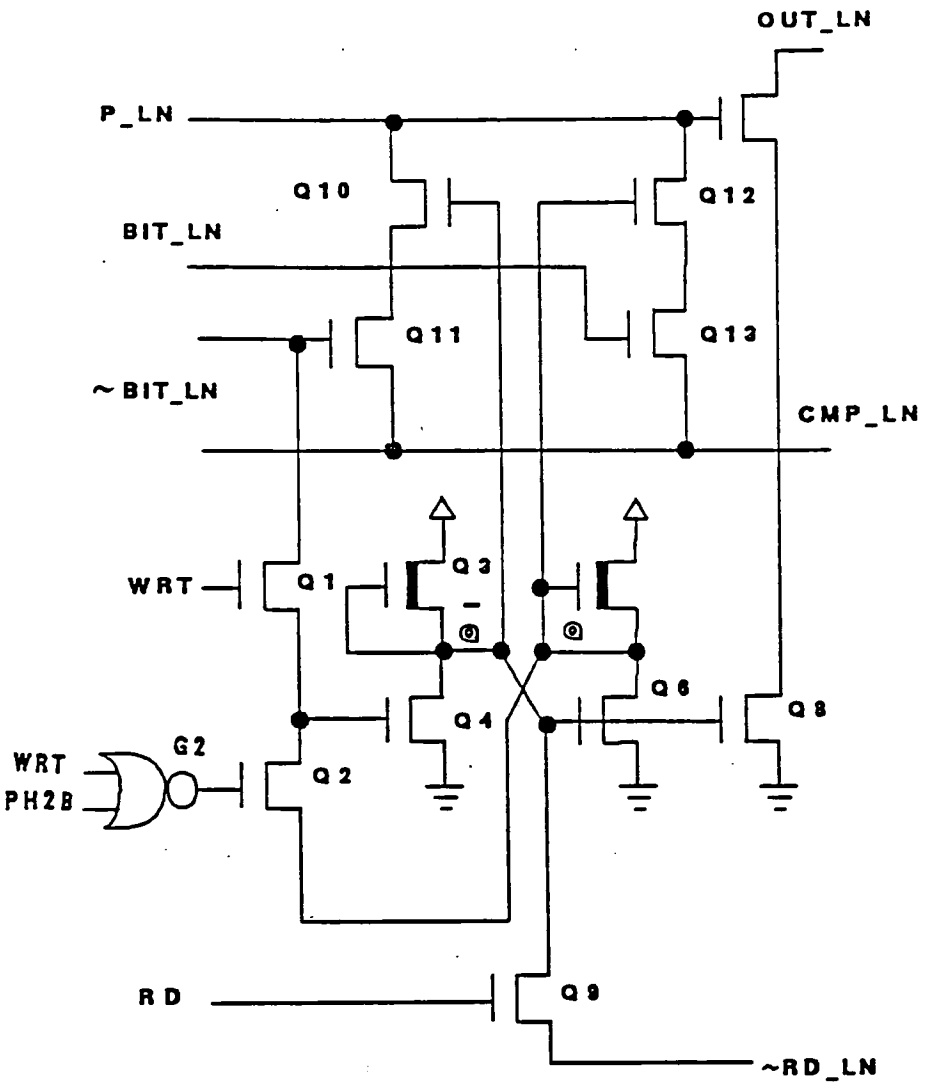
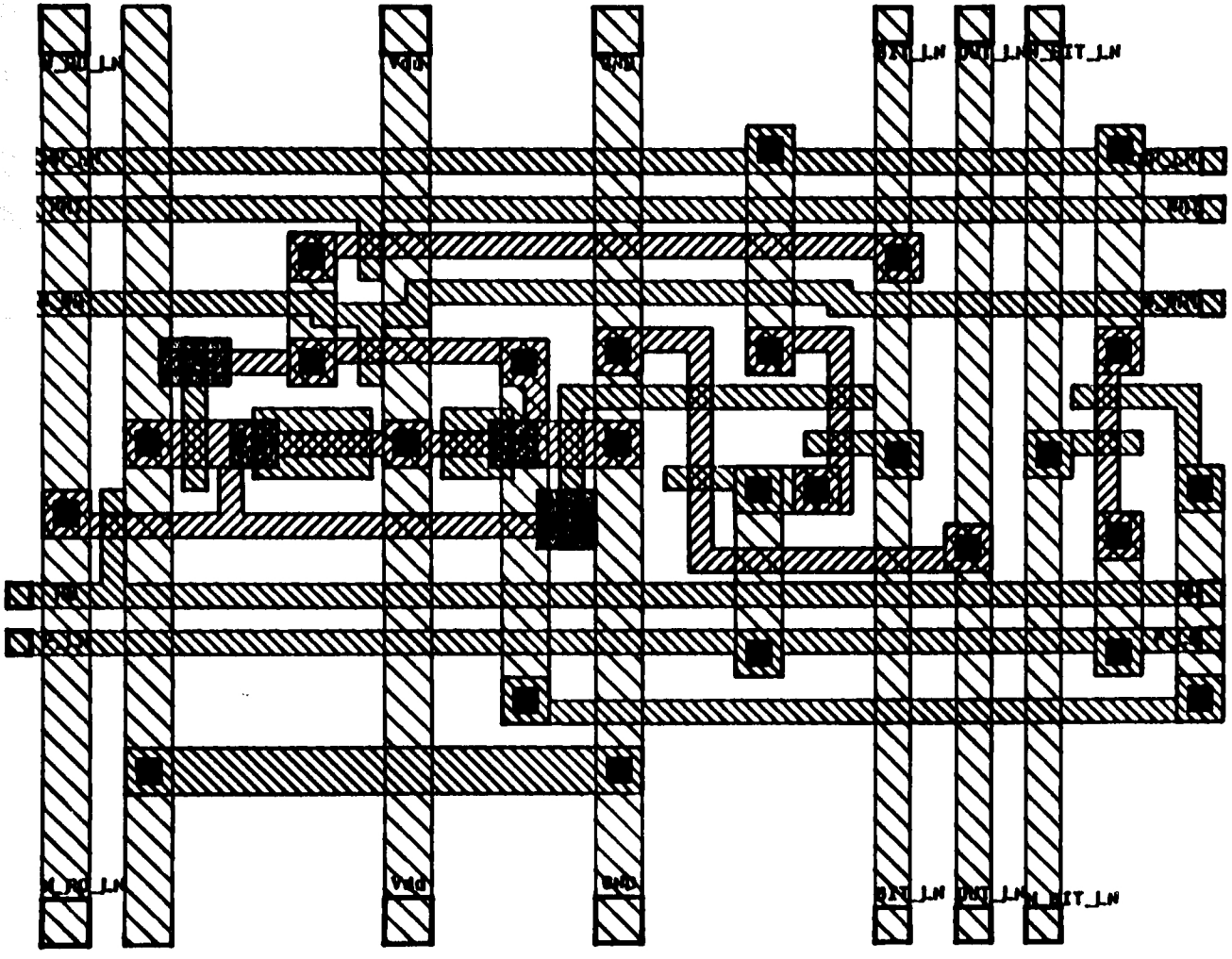


Fig. 3.2 A MEMORY CELL OF OR PLANE



OR_CELL

Table 1. THE SIZE COMPARISON WITH DIFFERENT MEMORY CELLS

	APLA	WPLA		PLA	CAM	SRAM
		AND	OR			
Technology	NMOS	NMOS	NMOS	NMOS	NMOS	NMOS
Structure	Dynamic	Static	Static	-	Static	Static
No. of Tr's Dep. Enh.	5 0 5	9 2 7	13 2 11	1 0 1	9 2 7	6 2 4
Interface Lines	5	6	8	2	4	3
Size ($\mu m * \mu m$)	54*54	112*130	152*130	14*14	124*90	84*64
Area (μm^2)	2,916	14,560	19,760	196	11,160	5,376
Area Factor	1	4.99	6.78	0.069	3.83	1.84

and Q11 or the path of Q12 and Q13 is turned on. If these two paths are both off, the state of CMP_LN is floating or temporarily kept on the previous state until entering into the SEARCH mode.

During the WRITE operation, the state of BIT_LN and ~BIT_LN will be stored into the node ~Q and Q. In the SEARCH mode, CMP_LN is connected to GND through a control gate Q1(Fig.3.10). If the current state of ~BIT_LN (BIT_LN) is opposite of the state of the memory cell on the node ~Q (on the node Q), P_LN will be kept at high. Through a strobe pulse(STB), P_LN is gated to the target tag. On the other hand, if no match is found, P_LN is pulled to GND through CMP_LN and clears the target tag associated with this particular row.

Fig.3.3 shows the layout of these two memory cells. The size of comparison among PLA[6], APLA[16], CAM[14], and SRAM[15], which is normalized by the size of the APLA, is shown on table 1. The higher overhead on WPLA memory cells compared with APLA is due to the usage of a pseudo-static memory cell and the augmentation of WPLA by content addressable functions.

3.2 The Input Buffer

There are two different kinds of input buffers, one for the AND plane and one for the OR plane. The input buffer for the AND plane is shown in Fig.3.4. It provides the

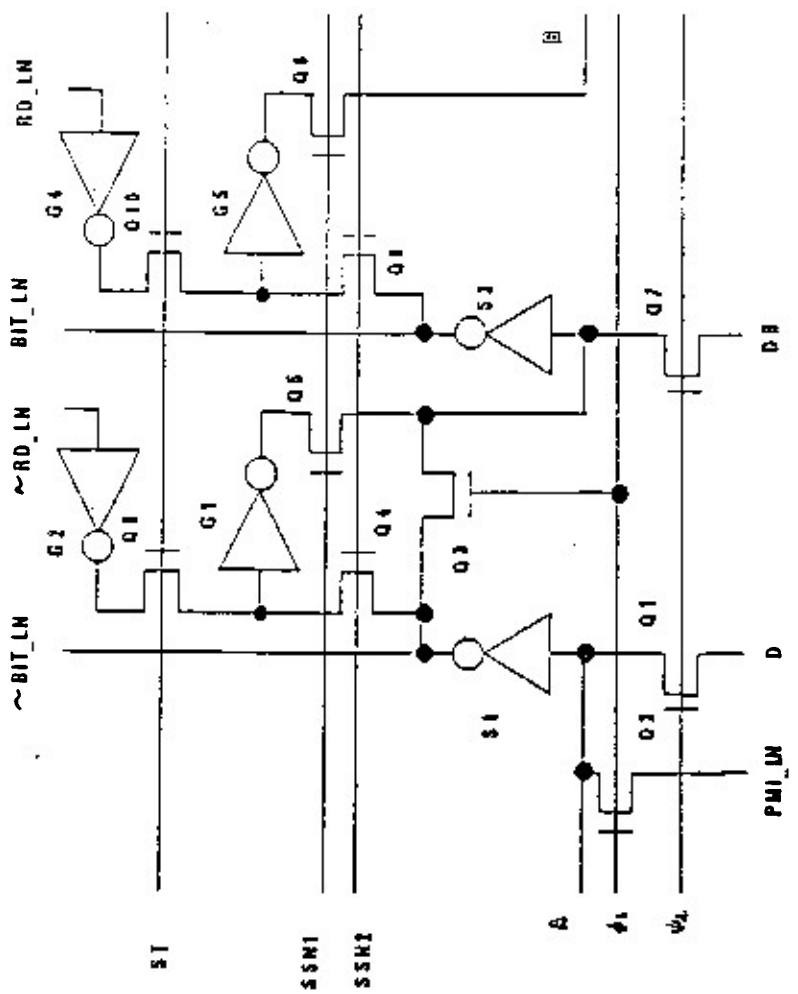


Fig. 3.4 INPUT BUFFER FOR AND PLANE

data to a pair of memory cells or receives the data from them. In the NORMAL operation, the gates of Q2 and Q3 is turned on by Φ_1. The primary input on PMI__LN goes through the superbuffers S1 and S2 to supply the complement and the true terms of the input variable, called *literals*, to ~BIT__LN and BIT__LN.

In the WRITE or the SEARCH operation, D and DB, the data to be written or to be searched respectively, come from the master driver. Since Q3 is off, the superbuffers S1 and S2 latch the data individually from D and DB through Q1 and Q7 by clocking Ψ_i. BIT__LN and ~BIT__LN, the outputs of the superbuffers S1 and S2, lead to all memory cells in these two particular columns and provide the personality (the search argument) to be written(searched).

In the READ mode, ST is active (Fig.3.4) and turns on Q6(Q10). The personality bit is immediately propagated to G2(G4) and Q6(Q10). By activating ST, the personality is latched onto the input of the inverter G1(G5). If a SCAN mode is activated next, Q6(Q10) will be turned off, and the two non-overlapping clock phases SSN1 and SSN2 are activated. The personalities previously latched, then will be scanned out to the next buffer stage in the shift register chain.

The input buffer for the OR plane is shown in Fig.3.5. It is similar to the right half portion of the input buffer in Fig.3.4, except for an additional transistor Q2, a control

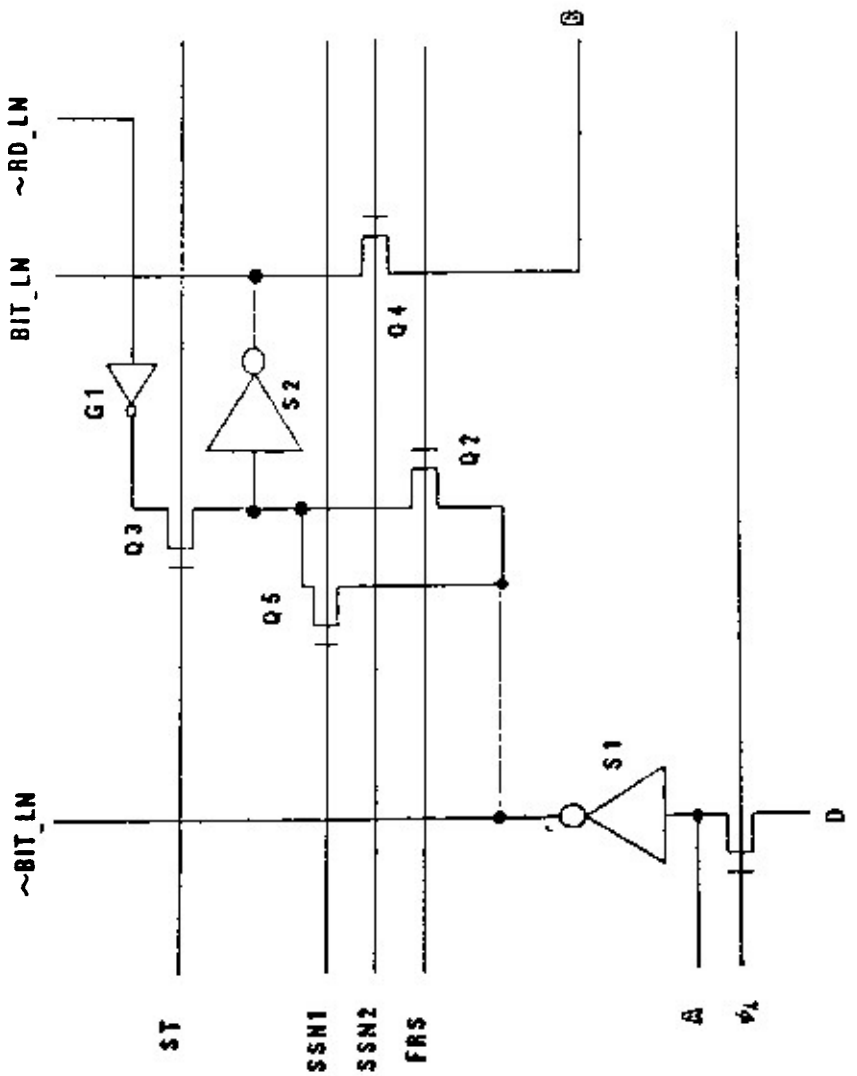


Fig. 3.5 INPUT BUFFER FOR OR PLANE

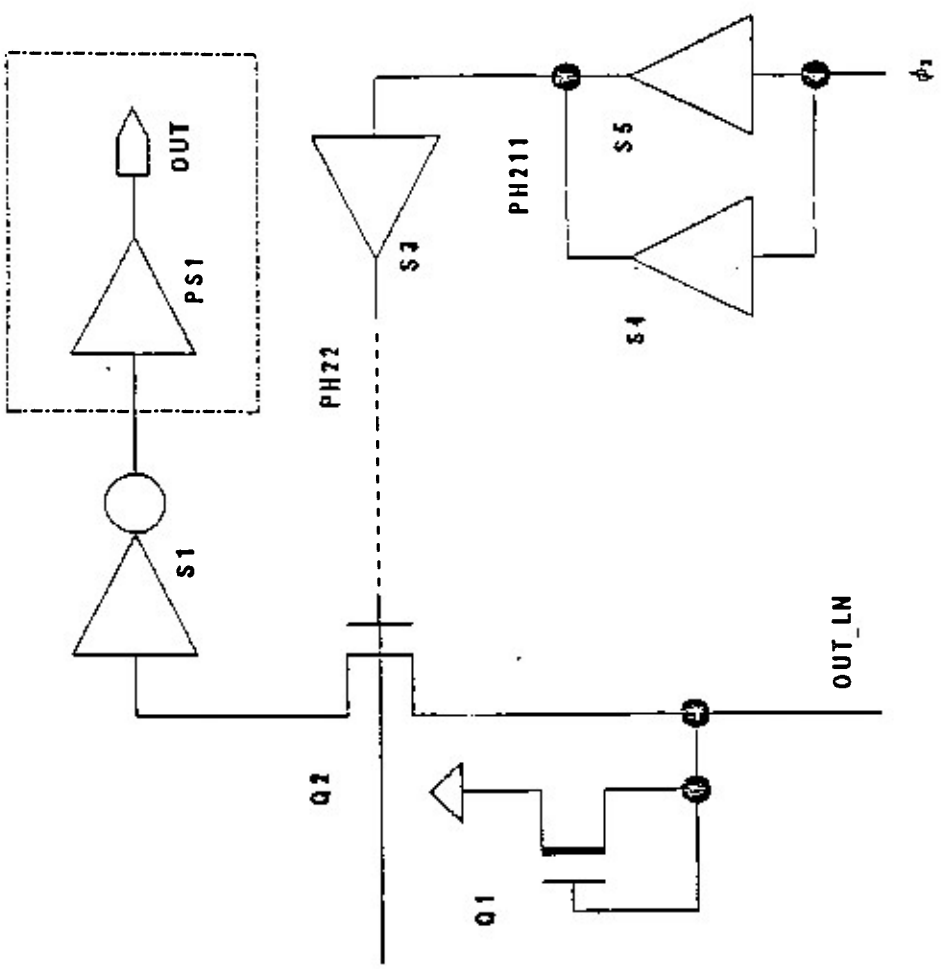


Fig. 3.6 OUTPUT BUFFER

signal FRS, and a BIT__LN coming from the superbuffer S2. In the NORMAL mode, the input buffer of this plane is idle and is a redundant circuit.

In the WRITE or the SEARCH operation, FRS turns on Q2. The data D goes through Q1 and S1 and provide the personality(the search argument) to be written(searched) on ~BIT__LN. Moreover, ~BIT__LN through S2 provides its complement state on BIN__LN which is utilized by the comparison circuit of the memory cell in the OR plane as mentioned in section 3.1.

In the READ mode, Q2 is turned off by FRS. After the personality has been read through G1 and Q5, it is latched onto the input of S2. The SCAN operation is performed, by clocking SSN1 and SSN2. As a result, the data will be shifted to the next buffer stage in the chain or will be observed at SSOUT pin.

3.3 The Output Buffer

The output buffer is shown in Fig.3.6. Q1 is a pull-up load for OUT__LN. The state of OUT__LN is clocked by PH22 to an inverting superbuffer, S1, during the NORMAL operation and finally drives the OUT pad to show the state of the boolean function. The Φ_2 is originally generated by the timing control circuitry. Through the line driver S3-5, Φ_2 is propagated to the output buffer and is denoted as PH22.

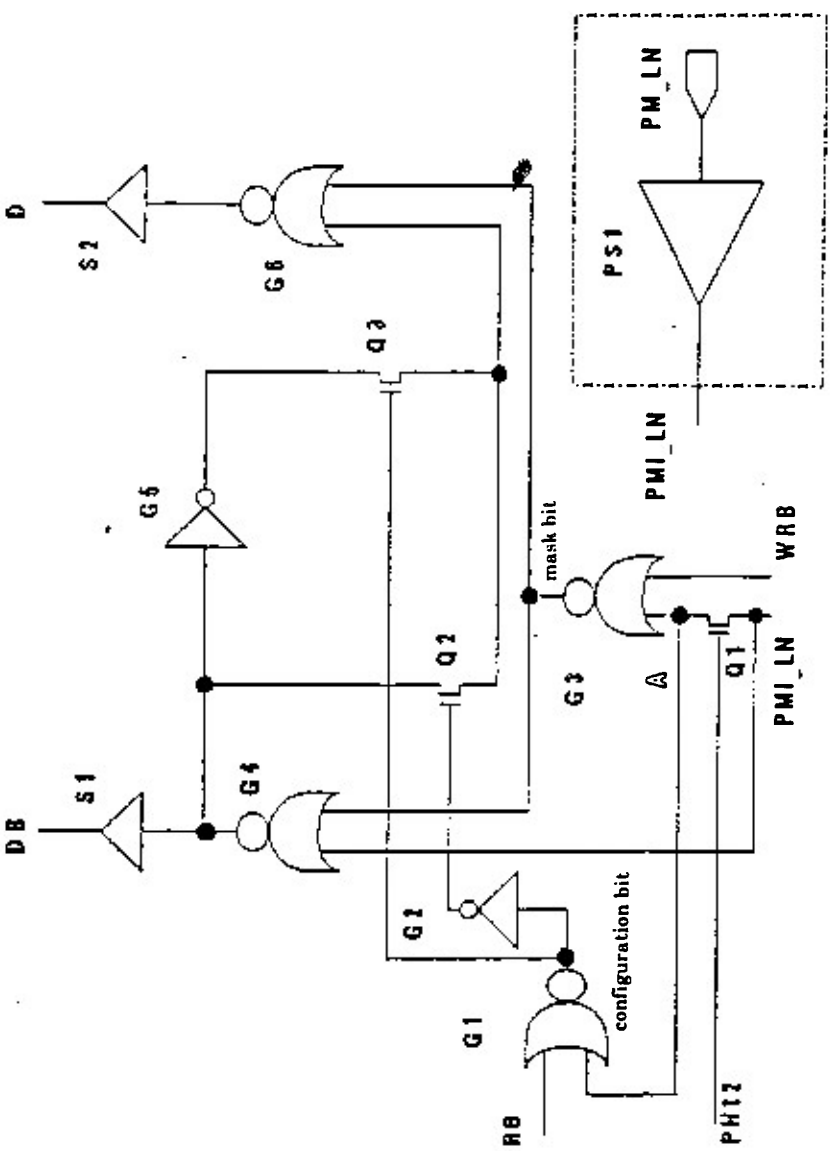


Fig. 3.7 MASTER DRIVER

3.4 The Master Driver

The master driver in Fig.3.7 is connected to the system data bus through PMI_LN. G3-G6 and Q1-Q3 construct a *data formator*. G1 and G2 control the pass transistors Q2 and Q3 to perform this function. The non-inverting superbuffers S1 and S2 enhance the driving capability of the formatted data D and DB. D and DB, like bus lines, are distributed to the input buffers of different partitions. By clocking PH12, the data bit is latched onto the node B as a *mask bit* and C as a *configuration bit.* As soon as PH12 is changed to low, the NOR gates G4 and G6 will generate the desired formatted data, as shown in Fig.2.4, through the masking bits and PMI_LN.

In the WRITE operation, since the SRB is high, Q2 is on and Q3 is off. The WTB is low and enables the NOR gate G3. The output G4 and G6 will be both low if the mask bit latched on the node A is "logic 0". On the other hand, if the latched state of the mask bit is high, the output of G3 will enable G4 and G6. Since G4 and G6 are cascaded through Q2, the opposite states will appear on the output of G4 and G6.

In the SEARCH operation SRB is low. The transistor Q2 will be off and Q3 will be on if the configuration bit on the node A is low. An additional inverter G5 is appended between G4 and G6. Meanwhile, WTB is high at this mode and the output of G3 is kept in a low state. Therefore, the output of G4 and G6 will both be in a state that is the opposite of the state of the PMI_LN. On the other hand, if the configuration bit

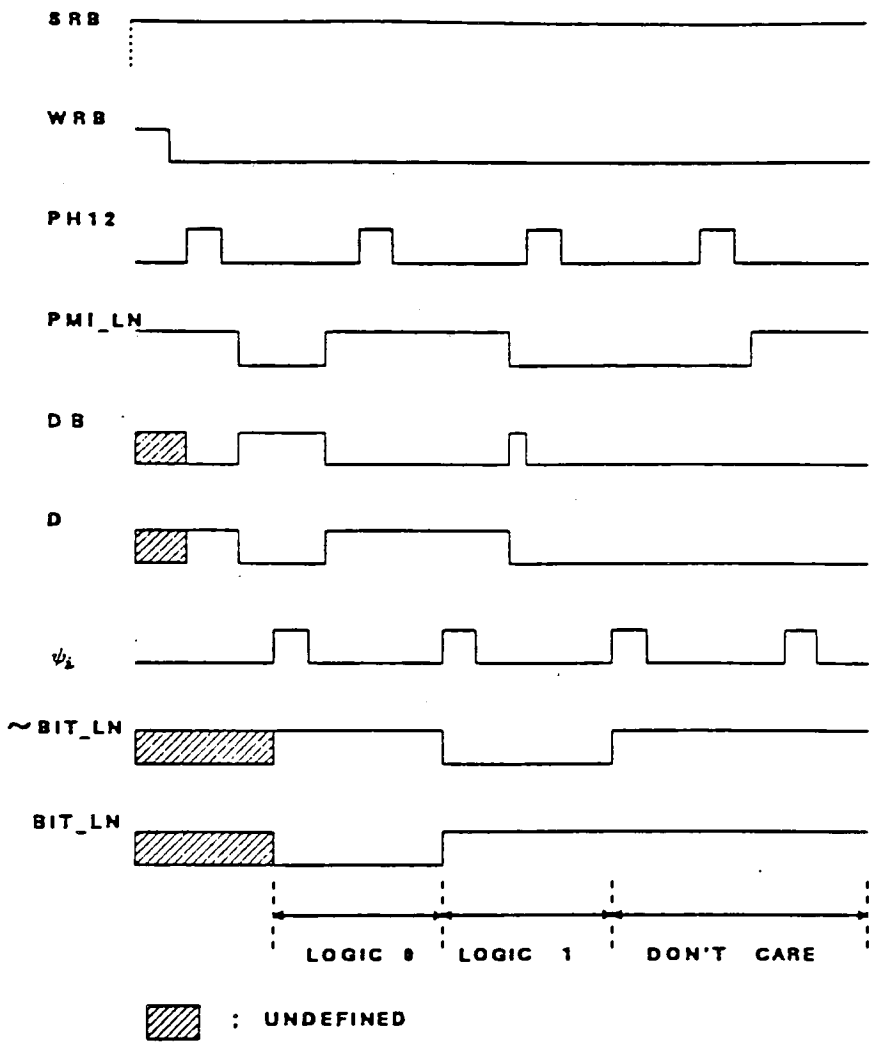


Fig. 3.8 DATA FORMAT FOR WRITE OPERATION

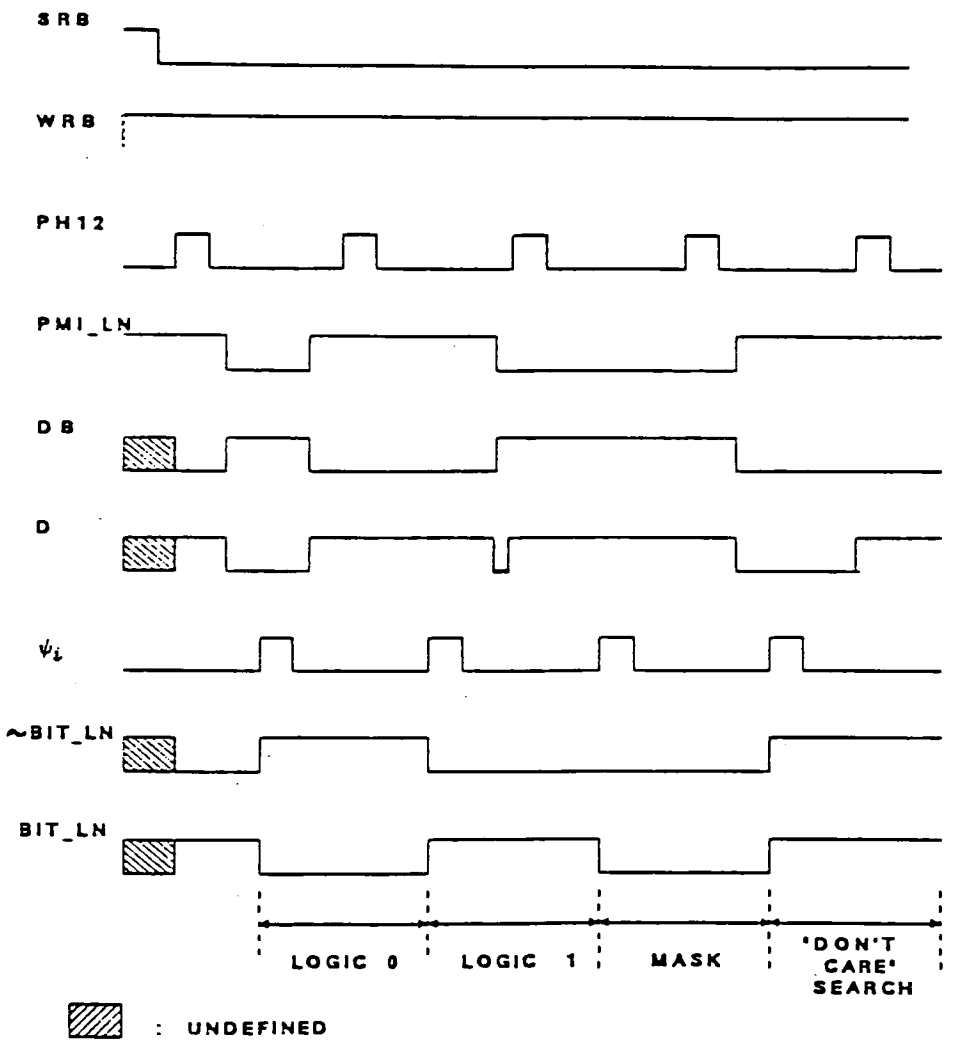


Fig. 3.8 DATA FORMAT FOR SEARCH OPERATION

is high, Q2 is on and Q3 is off. Again, the same state as PMI__LN will appear in the output of G6 and the opposite state of PMI__LN will appear in the output of G4.

The detailed timing diagrams for the data formatting for the WRITE and SEARCH operation are shown in Fig.3.8 and Fig.3.9. The dependency relation between D(DB) of the master driver and ~BIT__LN (BIT__LN) of the input buffer is also shown in these two figures. The related circuit diagram are shown in Fig.3.4, Fig.3.5, and Fig.3.7.

3.5 The Glue Logic Circuit

The pull-up load of P__LN and CMP__LN in Fig.3.10 is connected with the memory bank of each row. In the SEARCH mode, CMP__LN is kept in GND by clocking PH12 to gate the control signal SEREN1. This activates the comparison function in the memory cells of the OR plane as mentioned in section 3.1.

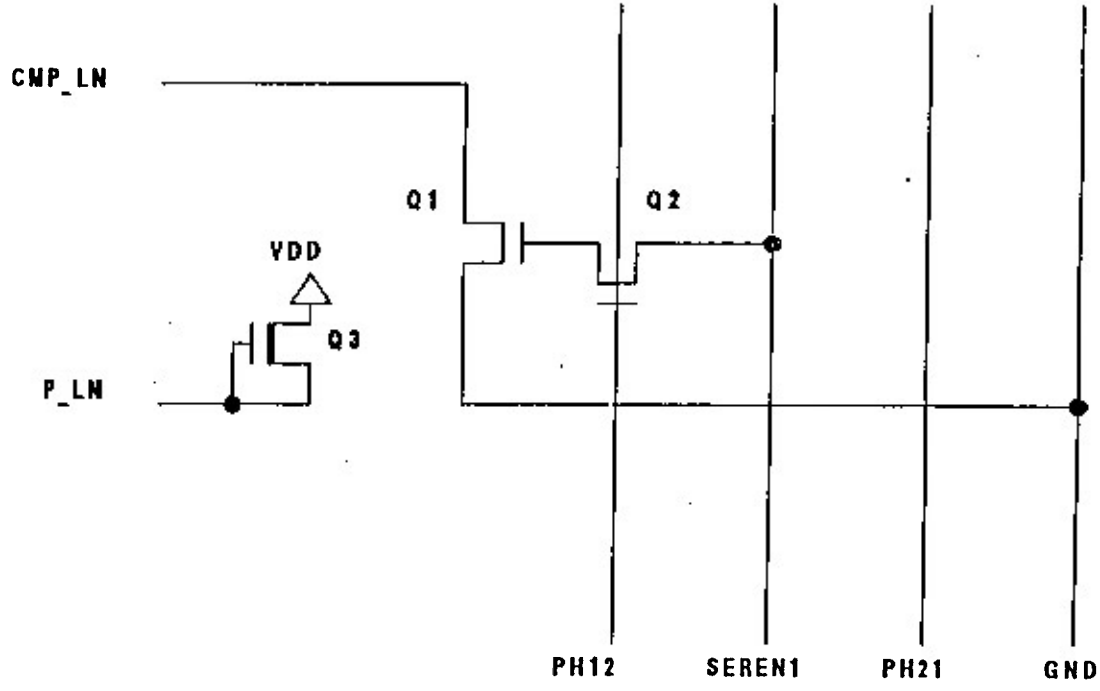


Fig. 3.10 THE GLUE LOGIC CIRCUIT

4. THE CONTROL PORTION OF A PLA

The control portion of a programmable logic array provides the associated control and timing signals to the data portion. It is composed of (a) the row control circuit, (b) the mode selection and main phase clock circuit, (c) the multiplexing control circuit, and (e) the glue logic circuit.

4.1 The Row Control Circuit

As mentioned in chapter two, each row in the WPLA has its associated control circuit as shown in Fig.4.1. The *status register* consists of two inverters, G8 and G9, and three pass transistors, Q11-13. Similarly, the *target register* consists of G2, G3, and Q3-Q5. Normally, the control signal SNB and SN2 turn on Q12 and Q13(Q3 and Q5) unless the WPLA is in the TEST mode. The function of the NOR gate G7 (G1) is to turn on Q11(Q4) to maintain the feedback path of the tag register, and to disconnect it whenever any operation control signal is applied to the input of G7(G1) for updating the state of the tag registers.

Initially, a master signal RS (Fig.4.2.b) sets the node Y, the output of the *status register*, to "logic 1", and resets the node V, the output of the *target register*, to "logic

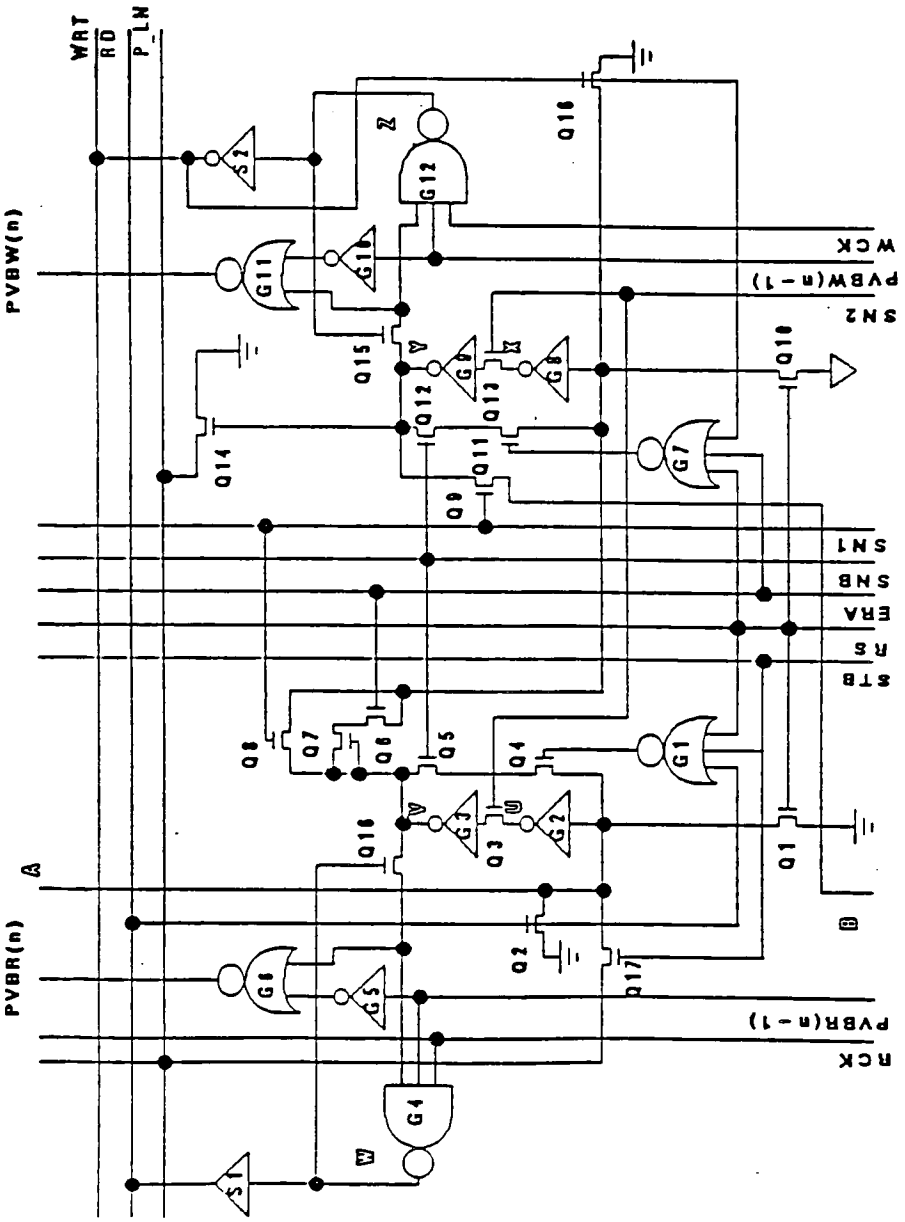


Fig. 4.1 A ROW CONTROL CIRCUIT

0". Since Q14, controlled by the node Y, is turned on, the P__LN is pulled down to *GND* at this moment and ignores the successive operations.

During the WRITE operation, if this specific row is vacant ("logic 1" on the node of Y) and receives the WRITE permission("logic 1" from PVBW(n-1)), the following WRITE clock (WCK) will be gated to the node Z to generate a negative WRITE pulse. Through superbuffers S2 and S3, the WRT will clock the associated memory cells on this row and will reset the node of Y to "logic 0" representing "occupied" through Q16. As soon as the trailing edge of WCK occurs, Q15 is turned on by the node Z. The "logic 0" in the node Y passes to the gate G12 and inhibits the following WCK to activate this row again.

In a like manner, during the SEARCH operation the target row will let its associated P__LN be high. By applying a strobe pulse(STB), the state of P__LN will be gated through Q17 to the target register. The state of the target register will be set to "logic 1" (target flag) on the node V.

During the READ operation, if the row is targeted ("logic 1" on the node of V) and receives the READ permission ("logic 1" from PVBR(n-1)), the following READ clock(RCK) will be gated to the node W of G4 to form a negative READ pulse. Through an inverting buffer S1, RD unloads the personalities of the memory cell to the ~(RD__LN) or (RD__LN) of this particular row and simultaneously resets the node

of V through Q2. Like WCK, the trailing edge of RCK turns on Q16 through the node W and inhibits any following RCK from activating this row again.

On the other hand, during the ERASE operation, if the target tag on the node V is high, it turns on Q6. The targeted tag, then, is shifted to the status register, and the status register is set to the vacancy ("logic 1") state.

The WRITE(WCK) and READ(RCK) permission signals of the current row control stage are generated by the priority circuit of the previous stage. The priority control signal PVBW[n-1] (PVBR[n-1]) provides the WRITE permission state (the READ permission state) for the current stage and the priority control signal PVBW[n] PVBR[n] provides the WRITE permission state (the READ permission state) for the next stage.

If a WPLA totally has k product lines, it indicates there are k rows in the chip. Each physical row is assigned a positive integer n ($1 \leq n \leq k$) and is chained together except for the two dummy priority signals PVBW[0] (PVBR[0]) and PVBW[k] (PVBR[k]). PVBW[0] (PVBR[0]) is constantly connected to *Vdd* and PVBW[k] (PVBR[k]) is assigned as the *handshaking signal* WTCMP (RDCMP) with the system controller.

The circuit serving the WRITE permission is implemented by gate G10 and G11. Row 1 will pass the WRITE permission to row 2 if the node Y in row 1 is "logic 0".

Otherwise, row 1 will write first before it releases the WRITE permission. The permission signal will follow the chain order and will propagate to the lowest vacant row. The lowest vacant row blocks the permission signal until it finishes the WRITE operation. After finishing this operation, the associated status register of this latest written row is set to "occupied" ("logic 0") and this lets gate G11 pass the WRITE permission to the next lowest vacant row until WTCMP is high and is received by the system controller.

Similarly, the priority circuit for the READ permission is implemented by G5 and G6. The lowest targeted row on the priority chain, through its target tag with the high state on the node V, will block the READ permission to the following rows and this lets RCK gate into this row. After finishing the operation, the associated target register of this latest read row is set to "logic 0" on the node V and this lets the NOR gate G6 pass the READ permission to the next lowest targeted row until RDCMP is high and is received by the system controller.

The function of the TEST mode is to check two tag register by shifting the test vectors from SIN and examining at SOUT as described in chapter two. SNB is changed to "logic 0" during this mode. The feedback path of both tag registers are disconnected by Q5 and Q12. But, through the pass transistor Q8 and Q9, a dynamic shift register chain is configured by the target register and the status register of each row control stage.

Node A of the current stage is connected to node B of its adjacent higher row except for the lowest and the highest rows. Node A of the highest row is assigned as SIN, and node B of the lowest row is assigned as SOUT.

4.2 The Mode Selection and Main Phase Clock Circuits

The mode selection and main phase clock circuit is shown in Fig.4.2. To reduce the pin count, the operation mode selection in Fig.4.2.a is implemented by a 3-to-8 decoder. The mode selection bits CBT0-2, and their decoded modes, have been shown in Table 2.

The bits CBT0-2, through the pad drivers, actives only one line to "logic 0" each time except in the TEST mode. At the TEST mode, CBT0-2 activates SN and SSN to concurrently test two dynamic register chains configured by the row control circuit and the input buffers. G1-6 are inverters that provide the complement state of each connected line in order to control each one's associated operation mode.

The two non-overlapping phase clocks, PHE1 and PHE2(shown in Fig.4.2.b), are supplied by the system controller and their driving capability is enhanced by the two pad drivers, PS1 and PS2. The inverting buffers, S1 and S2, provide the

complement forms of PHE1 and PHE2. The output of these superbuffers PH1B and PH2B are used in the internal control.

4.3 The Multiplexing Control Circuit

The multiplexing control circuit includes a module-*j* counter(*j* is the partition number of the input buffers as mentioned in section 2.2.2) and a decoder. It generates a set of multiplexing clock phases(MD0-2) for latching data into the different partitions of the input buffers.

In Fig.4.3, a modulo-3 counter and decoder are implemented. The modulo-3 counter is constructed by G6-10 and Q4-11. The decoder consists of G12-13 and Q12-20. The gates G1-5 and Q1-3 are used to gate the clock PH1B and the clear signal. If it is neither in the WRITE operation nor in the SEARCH operation, the output of G1 will be high to turn on Q1. Then, Q4 and Q5 will reset the counter output N2 and N4 to low. If it is either in the WRITE or in the SEARCH operation, the timing diagram for the counter and decoder is shown in Fig.4.4. The MD0, MD1, and MD2, the outputs of the decoder, are alternately changed to "logic 0". Through these three outputs of the decoder, the multiplexing phase clock $\Psi 1$-$\Psi 3$ and the WRITE pulse WCK (the STROBE pulse STB) can be generated as in Fig.4.5 and 4.6. The timing diagram is shown in Fig.4.7.

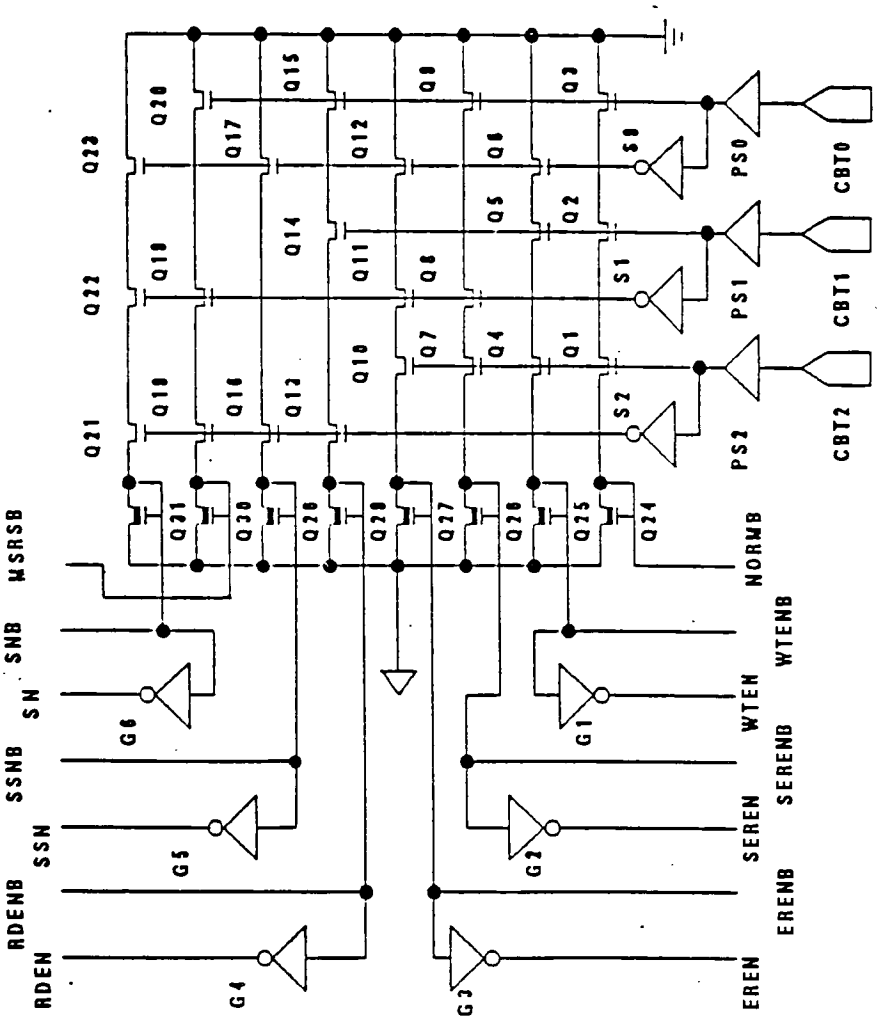


Fig. 4.2(a) MODE SELECTION

CBT2	CBT1	CBT0	MODE
0	0	0	TEST
0	0	1	MSRS
0	1	0	SCAN
0	1	1	READ
1	0	0	ERASE
1	0	1	SEARCH
1	1	0	WRITE
1	1	1	NORMAL

Table 2 Mode Select

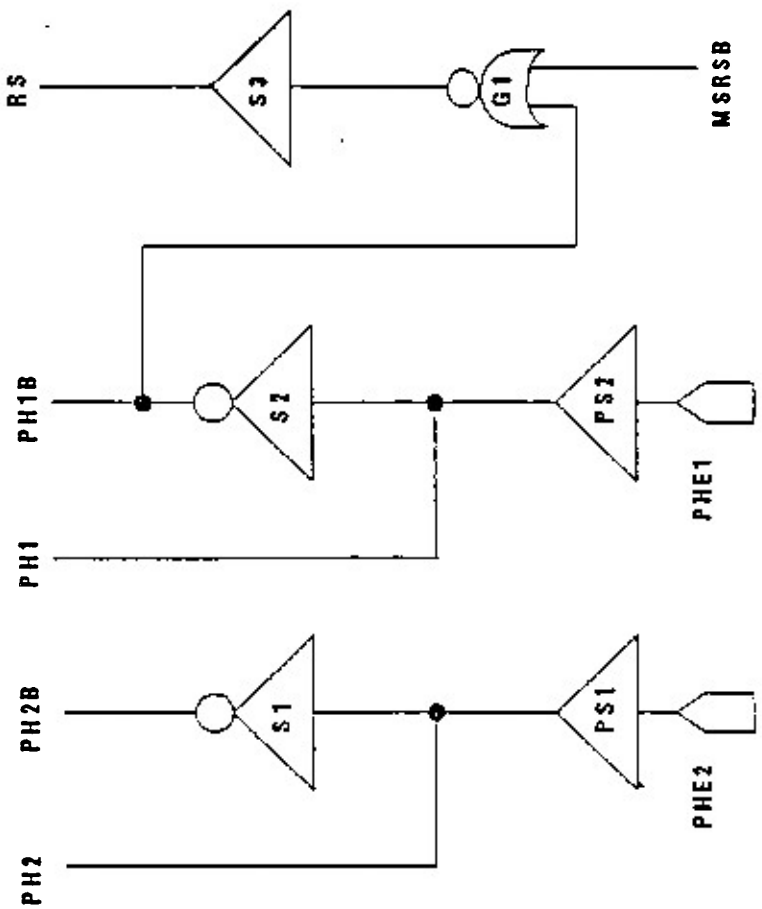


Fig. 4.2(b) PHASE CLOCK CIRCUIT

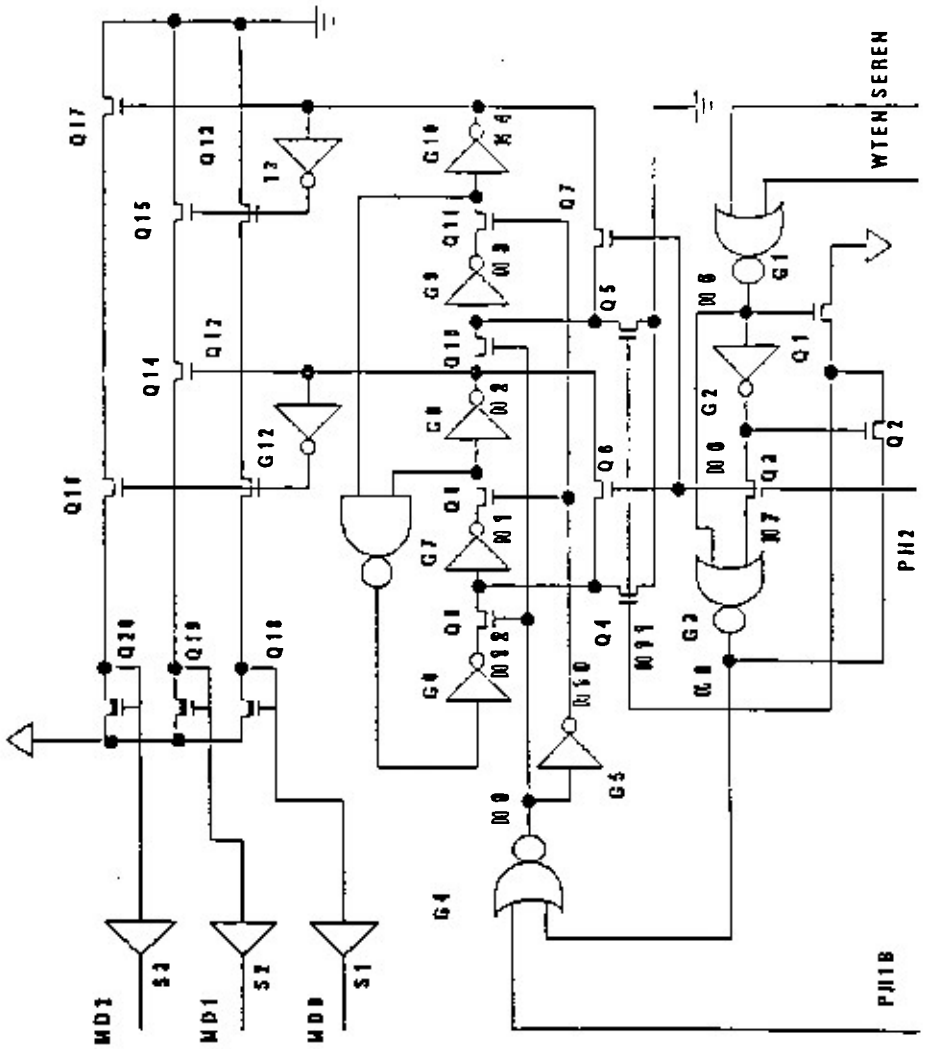


Fig. 4.2 MULTIPLEXING CONTROL CIRCUIT

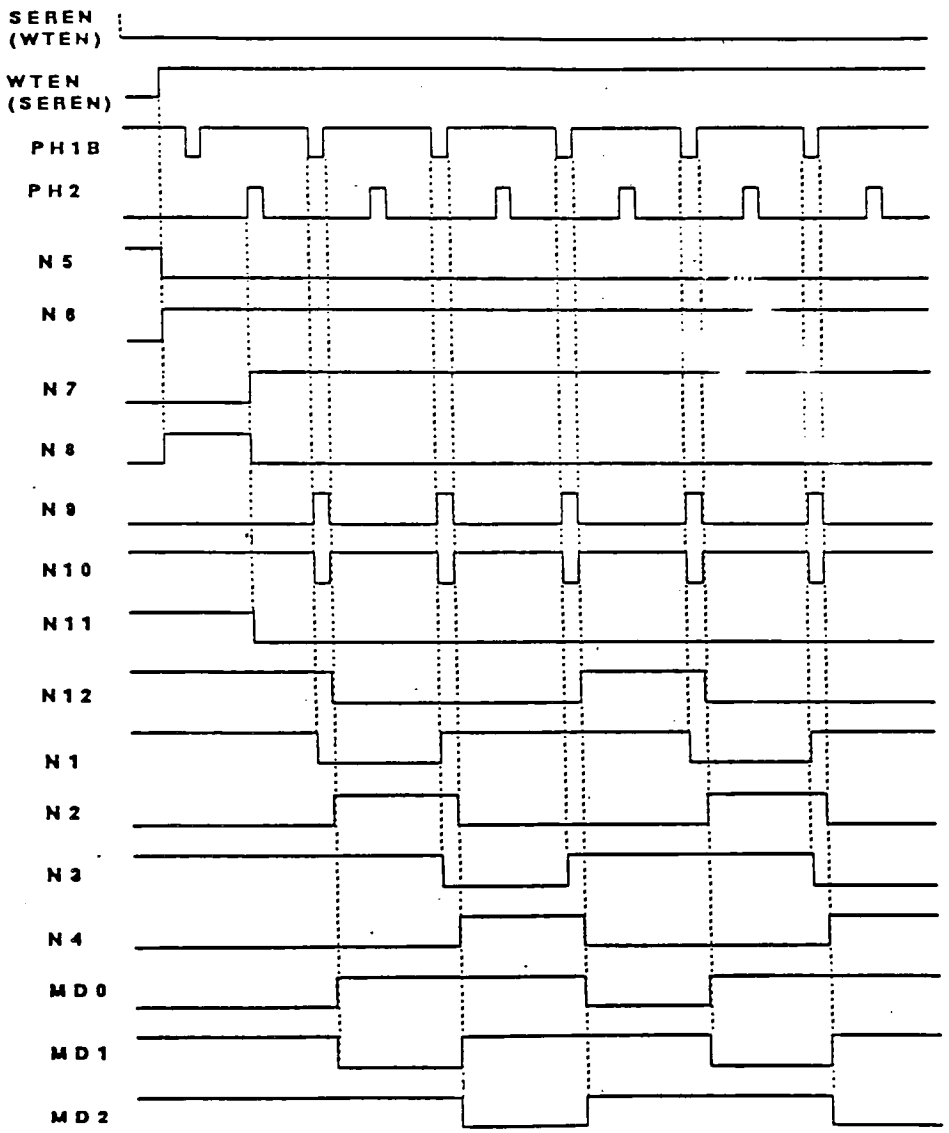


Fig. 4.4 TIMING DIRGRAM FOR MULTIPLEXING CONTROL CIRCUIT

4.4 The Glue Logic Circuit

Except for the control signals for the WRITE and the SEARCH operations, the other associated control signal is described next. The NOR gate G1 and a supper buffer S3(shown in Fig.4.2.b) use MSRSB and PH1B, generating the master reset pulse RS in order to initialize all status and target registers in the row control circuit.

In the READ mode or the ERASE mode, the relevant control signals are ST and FRS (shown in Fig.4.5) and RCK and ERA(shown in Fig.4.6). All of them are generated by the mode enable signals RDEN, RDENB, EREN and SSN, and by the two non-overlapping phase clocks, PH1 and PH2. Similarly, the SSN1 and SSN2 are generated by the scan enable signal, SSNB, and the clock phases, PH1B and PH2B.

The timing diagram associated with the READ, ERASE, and SCAN mode is shown in Fig.4.8. The other timing control signals shown in Fig.4.5 and Fig.4.6 are Φ_1, Φ_2, SN_1 and SN_2. The Φ_1 and Φ_2 are used individually to latch the input and output data during the NORMAL operation. SN1 and SN2 provide the two non-overlapping scan clock phase for the row control circuits during the TEST operation. The timing diagram to implement these two operations is shown in Fig.4.9.

To speed up the operation, line drivers are appended to some of the control and data signals as shown in Fig.4.10. Each input and output pin also has its own pad driver to enhance the driving capability of this design.

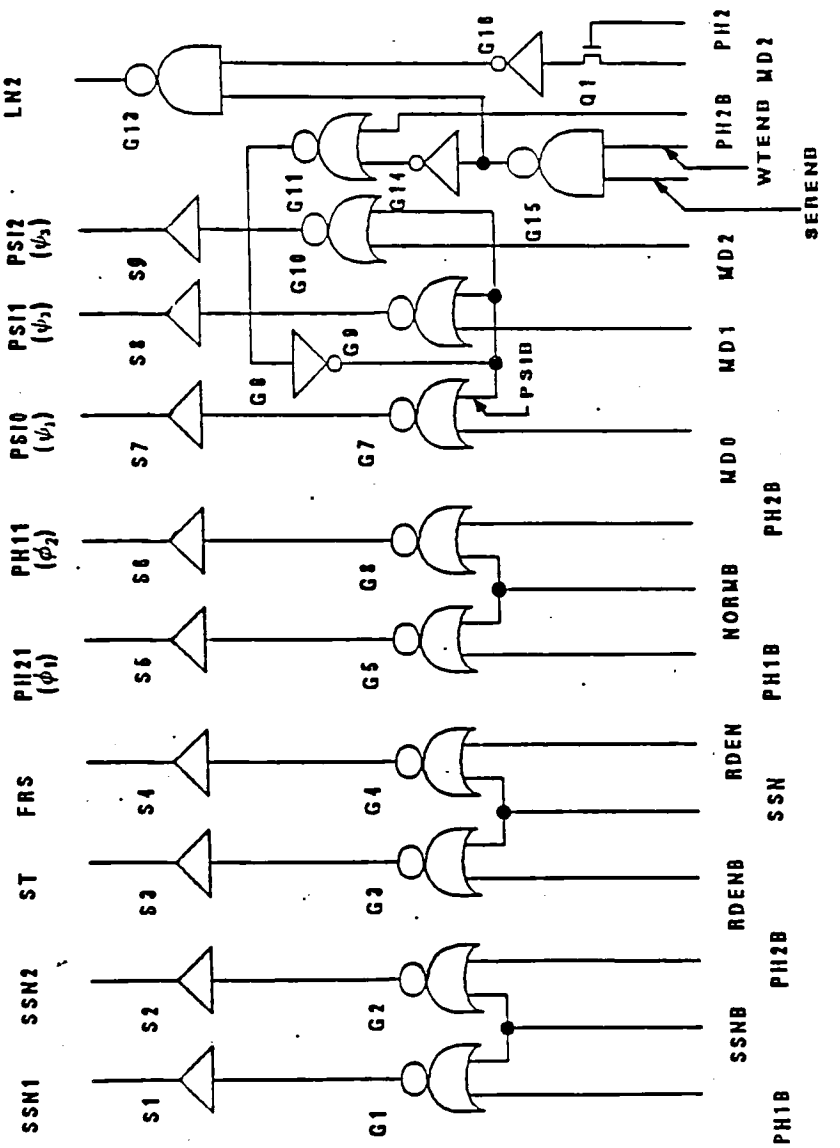


Fig 4.5 CONTROL CIRCUIT (PART 1)

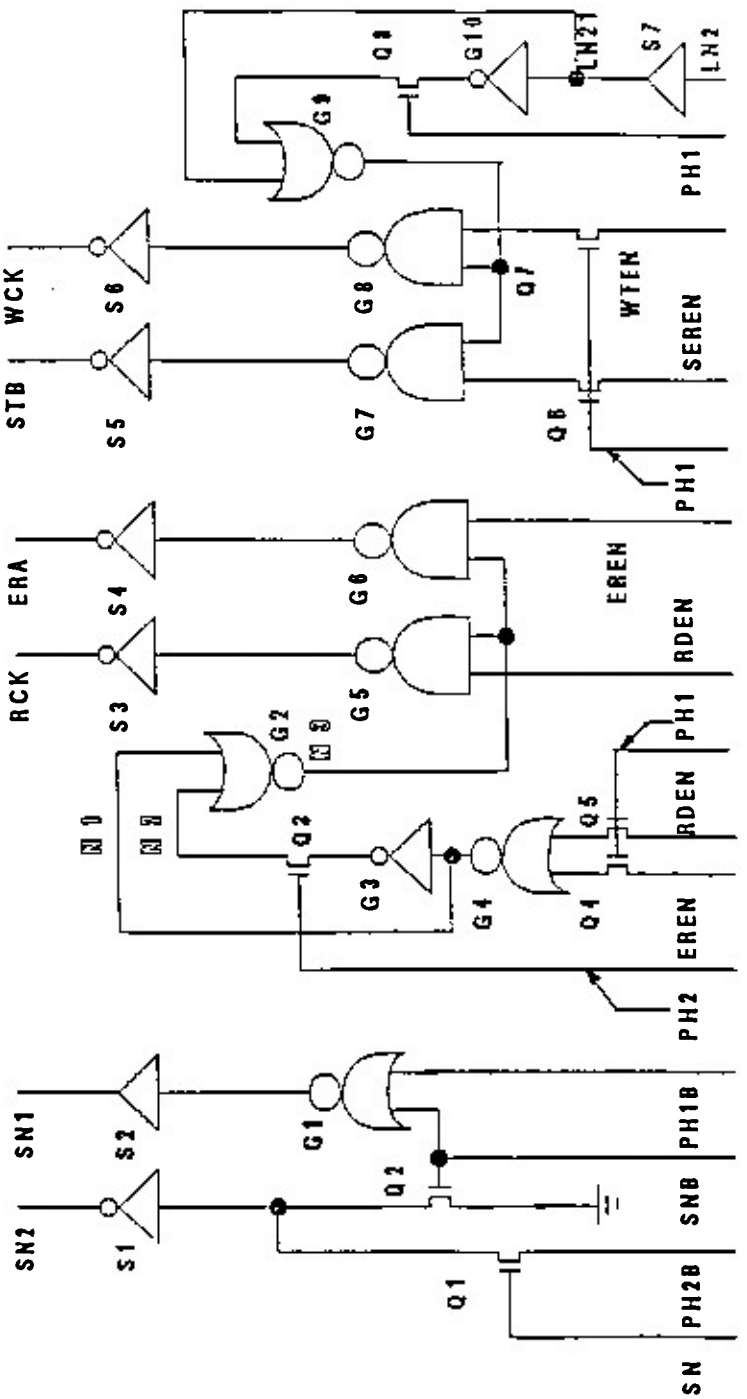


Fig. 4.6 CONTROL CIRCUIT (PART 2)

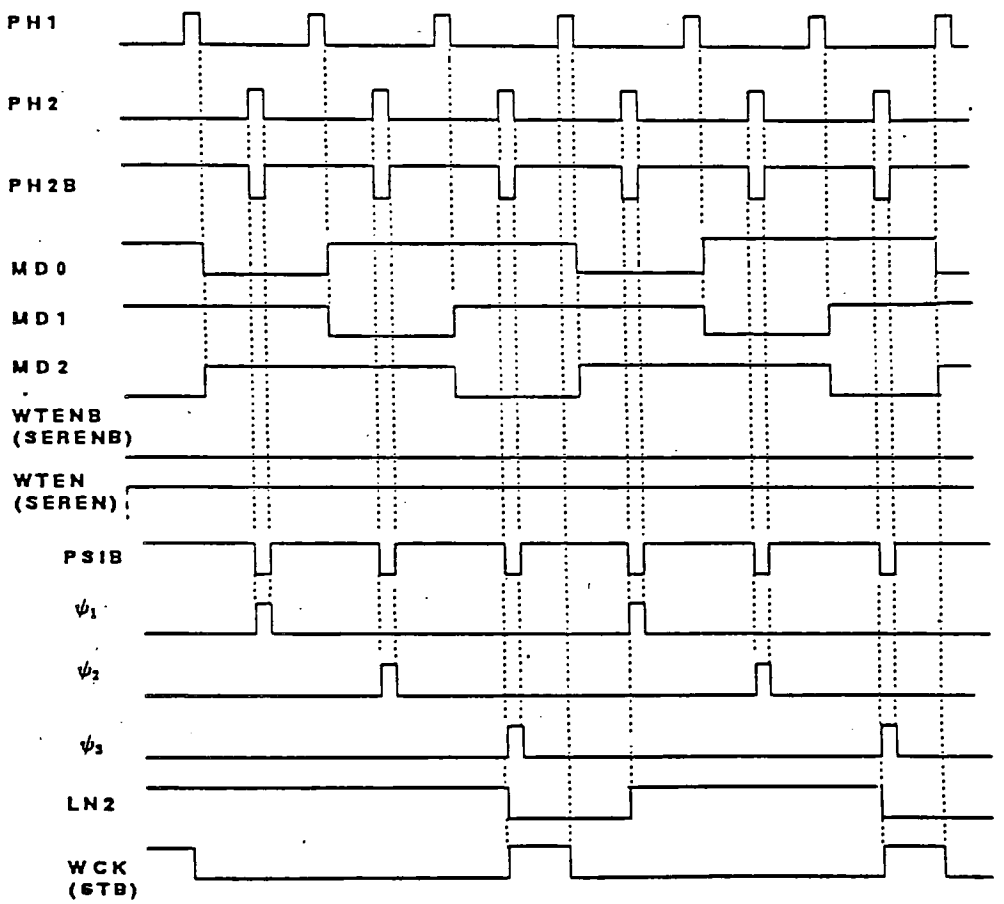


Fig. 4.7 TIMING DIAGRAM FOR WRITE /SEARCH MODE

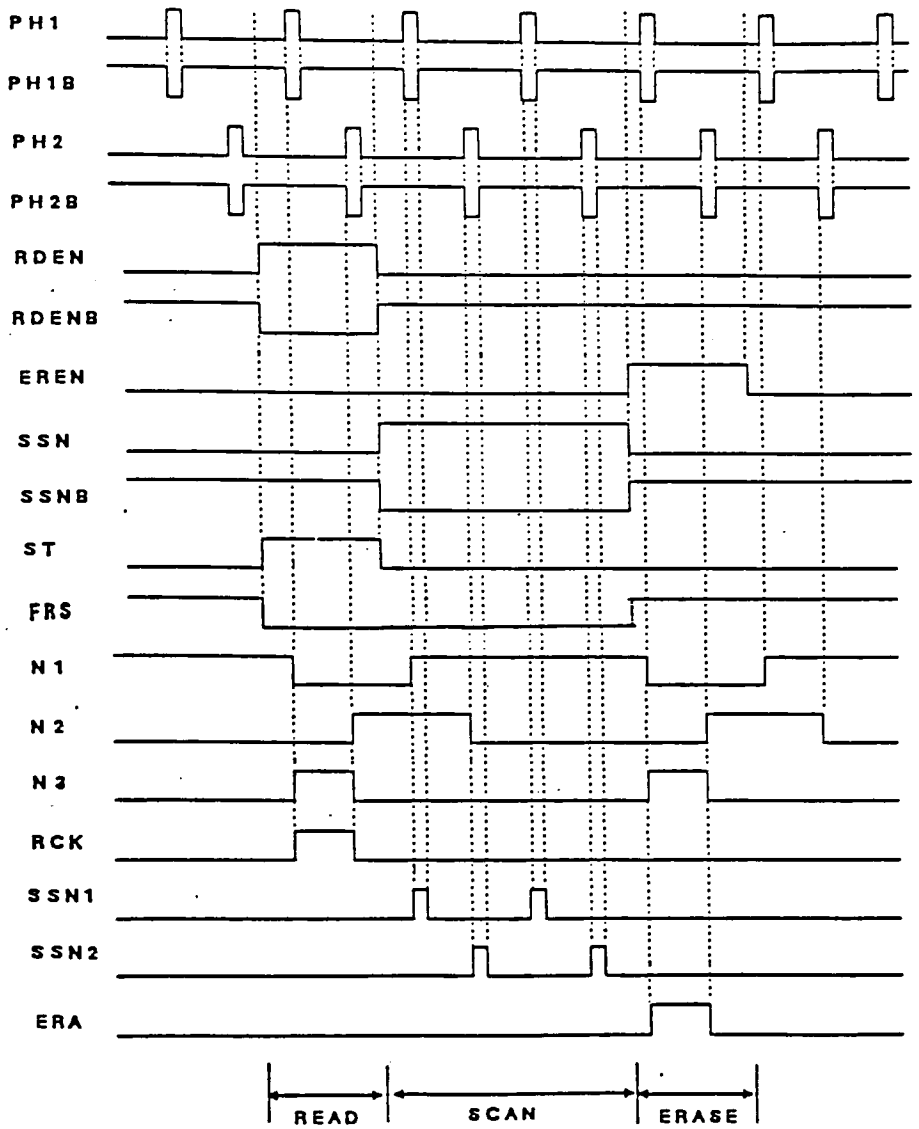


Fig. 4.8 TIMING DIAGRAM FOR READ, ERASE, AND SACN MODE

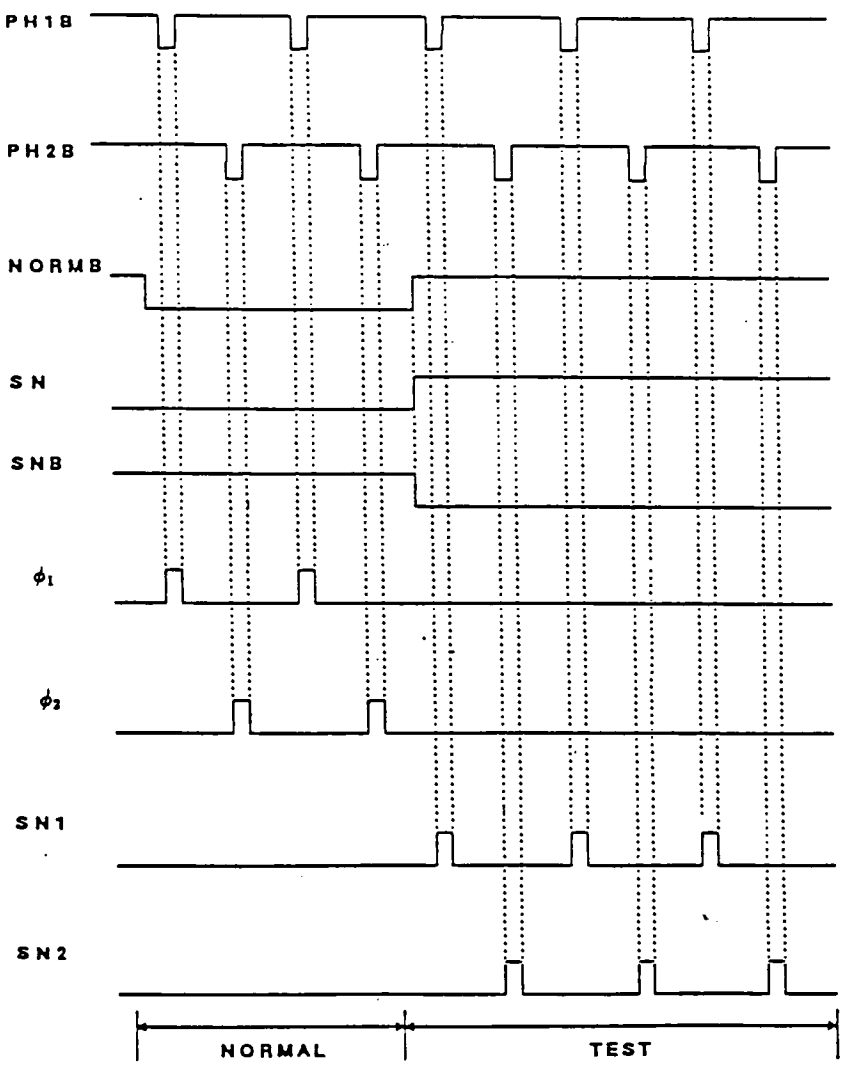


Fig. 4.9 TIMING DIAGRAM FOR NORMAL AND TEST MODE

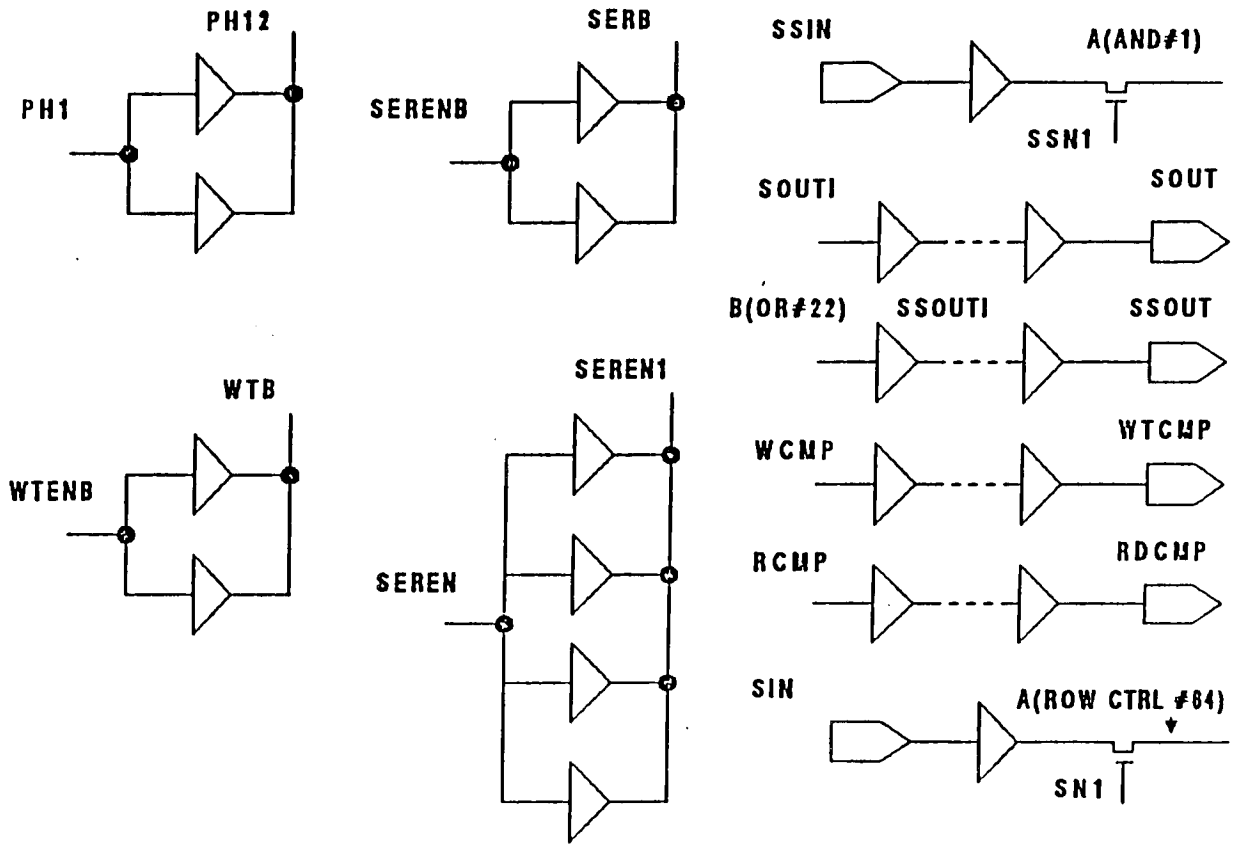


Fig. 4.10 GLUE LOGIC CIRCUIT

5. IMPLEMENTATION AND VERIFICATION

A WPLA with 22 inputs, 22 outputs, and 64 product terms, the same configuration as Marchand's chip[16] has been implemented here. The bus wide master drivers in the WPLA are constructed with a 16 bit structure which provides for an interface with a 16 bit system data bus. The input buffers for the AND and OR planes are divided into three partitions of 16 stages, 16 stages, and 12 stages. These 44 stages are available for latching data by multiplexing from the master drivers. The functional and timing behavior has been evaluated using Mentor Graphics logic simulator(QUICKSIM) and circuit simulator(MSIMON), and will be described in the following sections.

5.1 Implementation

The layout of the WPLA is implemented using Mentor Graphics layout editor CHIPGRAPH with 2 μm NMOS technology and MOSIS NMOS design rules. ECAD corporation DRACULA IC layout verification system was used to insure that the MOSIS design rules were correctly followed. The overall chip size, including bonding pads, is 10.3mm*10.5mm. A drawing of the pad assignments for the total

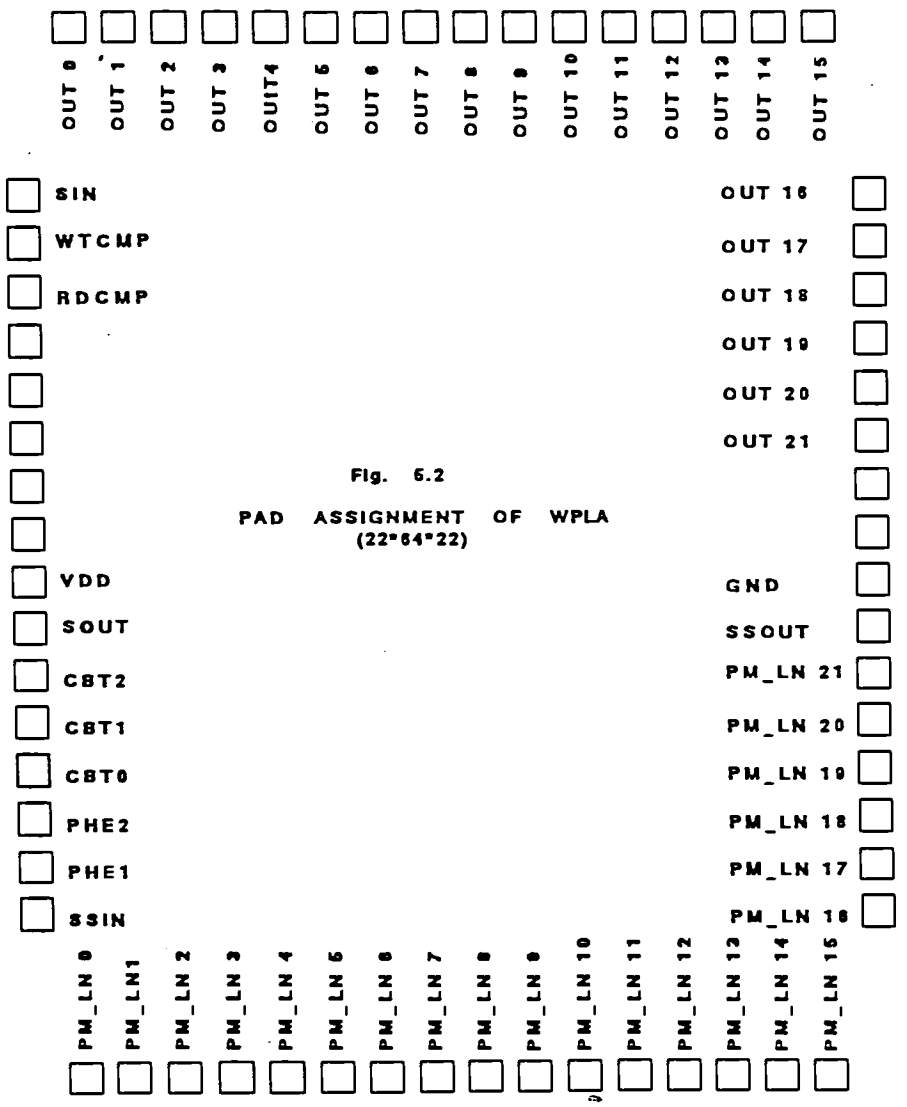


Fig. 5.2

PAD ASSIGNMENT OF WPLA
(22*64*22)

57 pins are shown in Fig.5.1. The transistor count in this chip is a total of 50,028, including 10,430 depletion transistors and 39,589 enhancement transistors.

The event-driven logic level simulator, QUICKSIM, and the timing analyzer, MSIMON, have been successfully used to verify the functional and timing behavior of the implemented WPLA.

5.2 Functional Verification

The eight basic operations and four of the macro operations were checked using seven subtests:

(1) MASTER RESET_then_WRITE

(2) WRITE_then_NORMAL

(3) NORMAL

(4) SEARCH

(5) SEARCH_then_ERASE

(6) SEARCH_then_READ_then_SCAN

(7) TEST

The test set-up, test vector, and test responses for each subtest are in the prepared manual. Subtest(4) is especially used in QUICKSIM to test the targeted P_LNs

and their associated target tags. The PROBE function(QUICKSIM) allows the states of P__LNs and the target tags to be observed directly without an additional TEST operation. In the test of the finished product, users can conduct a macro operation SEARCH-TEST(instead of a basic operation) in order to observe the same results.

5.3 Timing Evaluation

MSIMON is a powerful CAD tool used to analyze the worst case timing behavior of a circuit. This is especially useful for circuits designed using multiple non-overlapping clocks. In this test, the worst case duration for the phase clocks, PHE1 and PHE2, is determined for the WPLA.

Since the WRITE/SEARCH operation took more time to activate the additional counter and the decoder, as mentioned in chap four, these two operational modes as well, as the NORMAL operation, dominated the cycle time of the phase clocks. According to these observations, there are some different duration requirements for the phase clocks between the NORMAL operation and the WRITE/SEARCH operations. In the NORMAL operation, the duration of PHE1 should be longer than that of PHE2. This will be reversed in the WRITE/SEARCH operations. Therefore, there are two alternative clock strategies, and both of them have their PROs and CONs.

The first strategy is the dual clock scheme, which uses two sets of non-overlapping phase clocks, one set for NORMAL operation, and the other set for the remaining operations. The advantage of this strategy is that the operation speed is faster in the NORMAL operation mode. However, the system controller has to monitor the operation mode to provide different clock phases to the WPLA.

Next, a single set of phase clocks is used for all operations. Unlike the dual clock scheme, the system controller need not supervise the operation modes in order to deliver the different phase clock sets. On the other hand, the worst case operation frequency will be slower than that in the dual clock scheme.

6. Performance Evaluation and Comparison

A WPLA belongs to the application memory chip class. It not only includes the properties of the PLA and the SRAM, but also combines some functions of the APLA and the general purpose CAM. Comparisons among the WPLA and different memory types, based on their functionalities, speeds and chip sizes, are shown in Table 6-8.

The WPLA can perform the same function as the PLA and further is reconfigurable by programming the different personalities through its versatile operation modes. The memory cell of the WPLA is similar to SRAM, but it includes additional comparison circuits. Due to the additional internal control circuits and the augmentation of the memory cell, there are penalties in the slower speed performance and the larger chip size. The trade-offs among WPLA, PLA, SRAM are listed in Table 3 and Table 4. The performance evaluation, in the following sections, will focus on the comparisons between WPLA and APLA[16] and between WPLA and CAM[13].

6.1 Comparison between WPLA and APLA

In term of functionalities, both WPLA and APLA can implement the sum of product boolean equations and the state machines. The personalities in both chips can be programmable and updatable. The differences between WPLA and APLA are:

(a) In the WRITE mode the WPLA uses segmented write by multiplexing through a bus-wide length instead of using the serial scan approach in APLA. In comparison to APLA, the write speed of WPLA(in Table 7) is 18 times faster when using a single set phase clock, and 22 times faster when using a two set phase clock.

(b) The pseudorandom addressing scheme in WPLA alleviates the necessity of using a table or exhaustive search to find a free location for a WRITE operation. Without knowing where a free location is, the self addressing scheme provides a free location through a priority chain. In the APLA, the addressing scheme, provided by the scan-in and decoding circuit, is time consuming and the vacant rows or columns need to be known by the users.

(c) No background pattern is required to put into the vacant rows. The WPLA provides ease of initialization and an ERASE operation. In the APLA, The

vacant rows and columns must be programmed for the background pattern during the initialization and ERASE operation in order to avoid malfunctions in the NORMAL mode.

(d) A powerful content addressable search in the WPLA makes the erase and updating of personalities easy and amendable to fault tolerant applications. On the contrary, the APLA has no way to perform a content addressable search.

(e) The control and timing interface of the system controller in the WPLA is easy. Unlike the WPLA, the system controller of the APLA needs to generate complicated timing in order to control the operations and the refresh.

(f) The disadvantages of the WPLA is a larger chip size and slower speed in the NORMAL operation. The pseudo-static memory cell of WPLA has been compared with the dynamic memory of the APLA in Table 2. The area factors of the memory cells in the AND plane and the OR plane are 4.99 and 6.78.

6.2 Comparison between WPLA and CAM

The CAM that will be compared with the WPLA is a general purpose VLSI CAM[13] with READ, WRITE, ASSOCIATE and other content-oriented operations. Both the WPLA and CAM have a content addressable capability and provide the ability to examine the targeted rows. However, there are some variations between these two architectures, and the major differences are listed below.

(a) In the functional aspect, a NORMAL operation for the WPLA drives the AND plane, like a search plane, by applying a search argument or key through the primary inputs. The AND plane then activates the OR plane as a functional plane to construct the boolean functions. For CAM, there is no functional plane in it. An associative search is conducted on the single plane and the targeted row(s) or record(s) is(are) retrieved immediately, and are followed by a READ operation. Without modification in the architecture, CAM can not perform the sum of product boolean functions or implement state machines. However, CAMs provide more versatile and faster data retrieval capabilities, especially in associated-linked and order retrieval, which may be more suitable in applications to LISP or DATABASE machine.

(b) The WRITE mode in CAM operates just like a standard RAM WRITE. The address mask register is loaded with the word address. Then the data mask register is loaded with the word data. Finally the match control register is loaded with the write signal. In the WPLA, the row to be written is pseudo-randomly selected by the priority chain in the row control circuits. On the other hand, CAM can perform overwrite operation through the above addressing scheme, but the WPLA always writes into the current lowest vacant row, regardless of the released rows during the previous ERASE operation, unless the lowest vacancy is the same as one of the released rows.

(c) A bit in the AND plane of the WPLA is constructed by a pair of memory cells to allow a mask information(don't care state) pre-programmed during the WRITE operation. The CAM uses only a single memory cell for each bit and has no capability to pre-program the mask information. The mask capability of CAM is implemented only in the SEARCH operation and depends on a search argument and a mask register. The WPLA also includes this mask scheme, implemented through a pre-latched configuration bit and a data bit in the master driver. Moreover, the formatted search argument generated from the master driver is capable of searching the "don't care" state or masking a bit as well as searching "logic 1" and "logic 0" in the AND plane. These four types of formatted data, "logic 1", "logic 0", "don't care search"(both "logic 1"), and "mask a bit"(both

"logic 0"), not only provide a more powerful search capability for the WPLA to update personalities, but are also applicable to minimizing the two level boolean functions stored; they even support PLA partitioning and folding.

(d) At a READ operation immediately following the SEARCH/ASSOCIATE operation, the WPLA as well as the CAM determines which targeted row or column is read first through a priority circuit. The personality in the WPLA is examined through a serial scan path, but the data in the CAM is examined in parallel through the bit lines. It takes more time for WPLA to examine the target row(s). Since the macro operation, SEARCH-READ-SCAN, in the WPLA is only used for off line test purposes, it will not affect the NORMAL operation in the WPLA because there is no need to examine the personalities of the target row(s) in the WPLA at this moment.

(e) The chip size of the WPLA is somewhat larger than the CAM due to the larger number of I/O pins and the overhead in the control circuit.

Table 3 FUNCTIONAL COMPARISON

	PLA	APLA	WPLA	CAM	SRAM
Normal Operation	Content-Addressing in AND Plane	Content-Addressing in AND Plane	Content-Addressing in AND Plane	Content-Addressing in Whole Plane	Address Decoding in Whole Plane
Write	No	Yes	Yes	Yes	Yes
* Address Scheme		Scan & Decode	Pseudo-random	Decode	Decode
* Data Scheme		Serial	Parallel & Mux	Parallel	Parallel
Read	Depend	Serial	Serial	Parallel	Parallel
Content-Addressability	No	No	Yes	Yes	No
* Bit mask			(Yes)	(Yes)	
* Link Associated			(No)	(Yes)	
* Read First			(Yes)	(Yes)	
* Multi-Write			(No)	(Yes)	
* Multi-Write Flagged			(No)	(Yes)	
Erase	No	Yes	Yes	Yes	Yes
* Stand-Alone thru Addressing		(Yes)	(No)	(Yes)	(Yes)
* Master Reset		(No)	(Yes)	(No)	(No)
* Multi-Erase thru Searching		(No)	(Yes)	(Yes)	(No)
Observability (Design for Testability)	Depend	Serial	Serial	Parallel	Parallel
* Data Portion		(Yes)	(Yes)	(Yes)	(Yes)
* Control Portion		(No)	(Partial)	(No)	(No)

Table 4. CHIP SIZE COMPARISON

	PLA	APLA	WPLA	CAM	SRAM
Configuration	22*64*22	22*64*22	22*64*22	16*8	28*8
Size (mm*mm)	9.92*0.90	5.62*5.34	10.30*10.46	6.11*6.17	5.56*4.34
Area (mm^2)	0.83	30.01	107.78	37.69	24.16
No. of memory Cells		4,224	4,224	8,192	16,384
Area Factor		1	3.59	0.65	0.21
Note	1	2	2	3	3

Notes:

1 The bonding pads and input/output transistors are not included.
2 The bonding pads are included.
3 The size is unknown whether the bonding pads are included or not.

7. The Applications of Writable Programmable Logic Array

A WPLA has the advantage over a combinational logic circuit, not only in its regularity of structure, but also in its ability to program personalities. In addition, the operations of pseudorandom WRITE, content addressable SEARCH and ERASE abilities of the WPLAs can be used to advantage in many applications.

7.1 Reconfigurable Combinational Circuit

A WPLA can implement combinational functions as well as a PLA can implement them. However, the function in a WPLA can be modified through reprogramming the personalities, without requiring any change, either in the design or in the layout of the structure. Moreover, the SEARCH_ERASE_WRITE operations can dynamically reconfigure the implemented functions within a single WPLA chip if macro operations can operate on-line or can concurrently operate with the other executing modules in the system. This macro-operation will highly promote WPLA in dynamically reconfigurable systems and save the hardware overhead at the system design level. Thus, the larger silicon area of WPLA, in comparison to PLA, may be overlooked when considering its contribution at the system level.

7.2 Programmable finite state machine

The a WPLA can be used to implement a finite state machine with two non-overlapping phase clocks(PH1, PH2). The outputs of a sequential machine, under a given input sequence in a WPLA, are reconfigurable in contrast to the fixed code in a PLA. A WPLA's ability to be reconfigured extends the product life cycle and provides a modification capability for field applications.

Similar to a reconfigurable combinational circuit, a sequential machine can also dynamically change the personalities to implement the complicated functions within a single WPLA chip with its associated reconfigurable firmware. This contribution means the WPLA is also applicable to a dynamically reconfigurable state machine design instead of a statically reconfigurable one.

7.3 Peripheral Device Controller

The peripheral device normally has low speed compared with CPU operations. An interface between the computer and the peripheral device, implemented by a WPLA in Fig.7.1 has the benefit of good programmability and updatability. The personalities through a microprocessor or a computer data bus can be written into a WPLA chip or be modified to upgrade the performance. Moreover, a WPLA-based

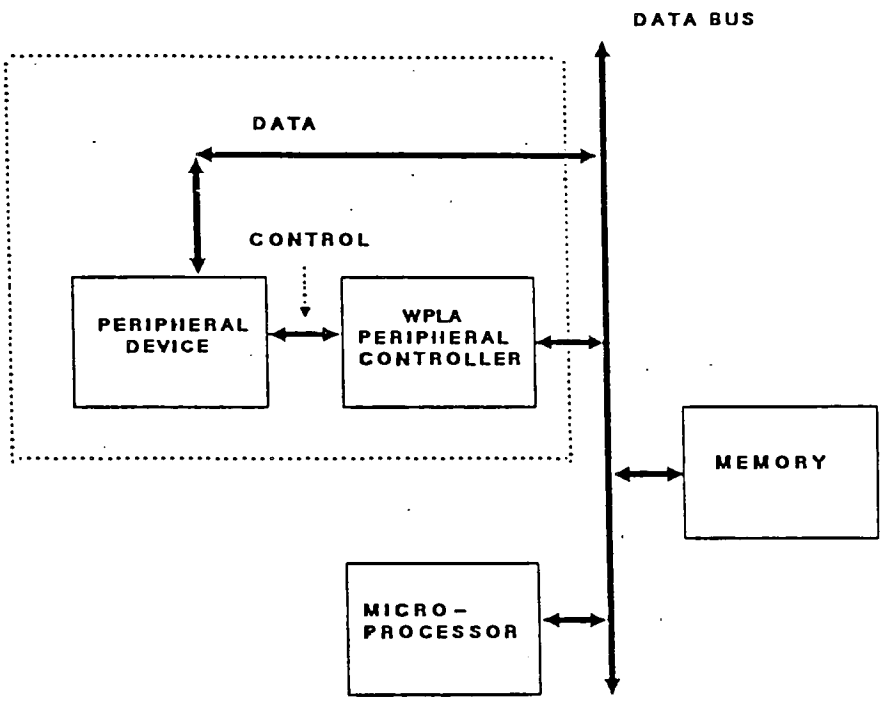


Fig. 7.1 A WPLA PERIPHERAL CONTROLLER

design can become a standard interface for all slow speed peripheral devices. The changeable personalities of a WPLA provide the different control sequences for the associated peripheral devices.

7.4 Fast Turn around Time Custom Design

While designing a computer or a digital system, the complete machine instruction set, or the control sequence, may not be well defined until interactively dealing with the customers and getting the final approval for the prototype. At design phase, a WPLA can easily change its personalities to fulfill fast turnaround time, such as SEARCH, ERASE and WRITE.

In the final product, if the speed requirements are not critical and not over the limit of maximum operation frequency of a WPLA, the WPLA-based design will continue to provide the capability of fast engineering change and low modification cost for the customer. On the contrary, if the maximum operation frequency of a WPLA in the final product is slower than the specifications, the WPLA can also be easily substituted by a PLA with the same personalities and still contribute to fast turn around time at the design phase.

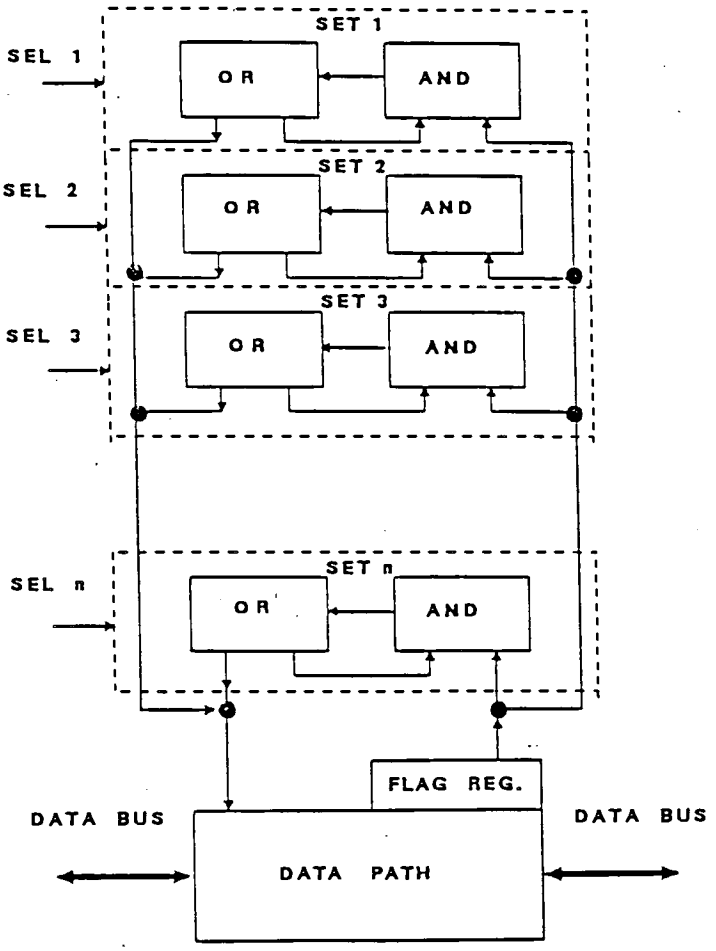
7.5 An Emulator for Different Computer Systems

The need to execute the instruction set of another computer for which the user has existing programs is quite common. Regardless of the control portion of the computer, which is conducting this operation using PLAs or micro stores, the instructions could be executed by simulation. They can be interpreted through a program written in the original machine instruction set. But, the simulation usually requies high performance penalties. Even, in the case of using multiple PLA chips, to emulate the different instruction set following the chip select(as shown in Fig.7.2), the hardware still demands high cost as well as having a lack of updatability and is limited to emulating few machines.

Using a WPLA and a database on the disk to store the different control sequences for each emulated system, a system can emulate a number of different systems with a single chip WPLA. It can also use online operations to program or update the personalities.

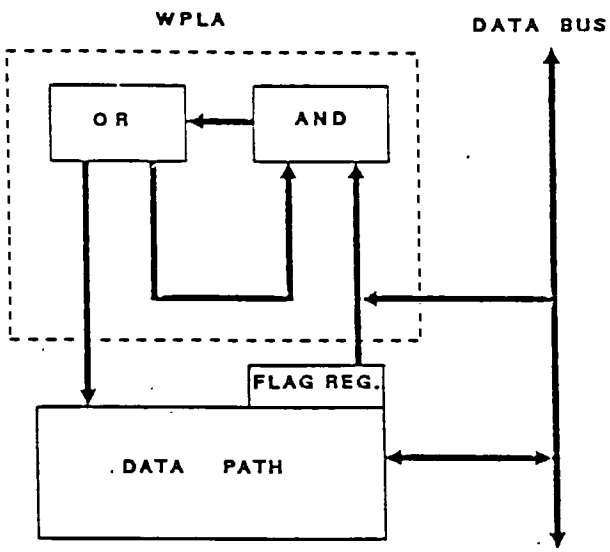
7.6 Testability and Reliability

Design for testability is one of the most important issues in VLSI design. Good controllability and observability in the WPLA makes the system easy to test, and a



(a) A PLA APPROACH

Fig. 7.2 AN EMULATOR FOR DIFFERENT COMPUTER SYSTEMS



(b) A WPLA APPROACH

Fig. 7.2 AN EMULATOR FOR DIFFERENT COMPUTER SYSTEMS
(Continued)

testable design can save an engineer a lot of work. The tests can be concluded either in the design phase or during the field application period.

During the field application period, a testable design will provide good maintainability and availability for the system. Moreover, if the macro operations are allowed to execute online, and the control sequences of the system are issued from the WPLA, then fault tolerance tests, in the data portion, can be conducted through reconfiguring the control sequences or using the redundant rows in the WPLA to recover the fault. For example, if a built-in hardware multiplier in the system is faulty and is detectable, an adder with the control sequences which simulate the function of the multiplier can replace the faulty multiplier immediately and backup this operation. The original control sequences for the fast multiplier are erased through a macro operation, SEARCH-ERASE.

The performance degradation is unavoidable in this approach, but it achieves fault tolerance at the functional level instead of the circuit level. Since multiple P__LNs and OUT__LNs will possibly be activated concurrently, the online fault isolation and correction will be difficult to perform in WPLA itself. The WPLA design, so far, has not explored the fault tolerant capability. But, by modifying the current WPLA architecture, or using error-correcting coding schemes, we may achieve a fault

tolerant capability in the future and may attribute fault tolerance both to the data portion and to the control portion of a WPLA-based design.

8. Conclusion

A WPLA uses a PLA architecture with pseudo-static memory cells and employs a set of macro operations. Without changing the design and the chip layouts, the WPLA based design can be reconfigured with respect to its embedded boolean functions only by dynamically modifying or reprogramming the stored personalities.

In contrast to an APLA, a WPLA uses an improved writing scheme in both the data portion and the addressing mechanism. In the data portion of a WPLA, the personality to be written is fed into the input buffers in parallel instead of serially. In the addressing mechanism, a new pseudorandom addressing technique was introduced to replace the address decoding approach and to accelerate the WRITE speed by at least a factor of 18. Besides, the augmented functions provided such as content-addressable SEARCH and ERASE operations can also contribute to a fast updatability. Moreover, the ease of interface and user friendly features, as well as a good testability design, facilitates a WPLA-based system in situations that demand fast turnaround time, customized design, and low volume applications.

Because of additional circuits used to implement the additional functions in a WPLA, as mentioned above, more silicon area is needed in comparison to a PLA or to an

APLA. But, following the increase of the inputs, outputs, and product lines, the internal control circuits are shared by the augmented data portion except for these needed to duplicate the row control circuit. This reduces the overhead percentage for the internal control circuits in the overall chip size.

Another difference between a WPLA and a PLA is that the rows in a WPLA may not be fully occupied. The physical area for the vacant rows remains in the chip to maintain updatability. Therefore, a WPLA based design with a good functionality may not be dense enough. On the contrary, a PLA based approach can implement a set of specific boolean functions or control sequences through the fixed number of product lines and hence no redundancy is embedded for future updating.

Generally, the chip size in a PLA based design is more dense and compact than that in a WPLA based design. Although the trade off between functionality and chip size is decided by the user, WPLAs with different configurations, like standard devices, can be manufactured. The various forms of WPLAs provide the users with choices that best fit their designs, with the least overhead possible. A sufficient number of redundant rows for updating or reconfiguring the personalities.

On the other hand, the low speed in the NORMAL operation is another disadvantage of a WPLA. This penalty results from more internal control circuits leading to the

same phase clock, and from the large chip size, and the long wire connection. If the pin count is not a critical constraint in the future package technology, then the speed of a WPLA, for all operation modes, can be improved by externally providing the multiplexing phase clocks and mode control signals, instead of a built-in multiplexer and a mode decoder. If the WPLA design is scaled for the coming 1 μm silicon technology, its predicted speed of operation is improved, The WPLA may be operated at approximately 7.5MHz for a dual phrase clocking scheme and at 4.7 MHz for a single phrase clocking scheme. Then, the functionality, speed, performance, and smaller chip size may expand the use of WPLAs in some other applications in addition to the fast turnaround custom design field.

REFERENCE

1. [Agrawal 1983]
V. D. Agrawal, and S. K. Jain, "Test Generation for MOS Circuits Using D-Algorithm," Proc., 20th Design Automation Conference, June 1983, pp. 347-354.

2. [Bose 1982]
P. Bose, J. A. Abraham, "Test Generation for Programmable Logic Arrays," Proc., 19th Design Automation Conference, June 1982, pp. 574-580.

3. [Bozorgui-Nesbat 1984]
Saied Bozorgui-Nesbat, Edward J. McCluskey, "Lower Overhead Design for Testability of Programmable Logic Arrays," International Test Conference, Proc., 1984, pp. 856-863.

4. [Bryant 1984]
R. E. Bryant, "A Switch-Level Model and Simulator for MOS Digital Systems," IEEE Trans. on Computers, vol. c-33, no. 2, Feb. 1984, pp. 160-177.

5. [Daehn 1981]
W. Daehn and J. Mucha, "A Hardware Approach to Self-Testing of Large Programmable Logic Arrays," IEEE Trans. Computing, vol. c-30, Nov. 1981, pp. 829-833.

6. [Fong 1984]
E.Fong, M.Converse, and P.Denham, "An Electrically Reconfigurable Programmable Logic Array using a CMOS/DMOS Technology," IEEE J. Solid-State Circuits, vol. sc-19, no. 6, Dec. 1984, pp. 1041-1043.

7. [Fujiwara 1985]
Hideo Fujiwara, R. Treuer and V. K. Agarwal, "A Low Overhead, High Coverage, Built-In, Self-Test PLA Design," Digest 15th Int. Symp on Fault-Tolerant Computing, June 1985,

8. [Fujiwara 1983]
Hideo Fujiwara, Kewal K. Saluja, K. Kinoshita, "An Easily Testable Design of Programmable Logic Arrays for Multiple Faults," IEEE Trans. Computers, vol. c-32, no. 11, Nov. 1983, pp. 1038-1046.

9. [Fujiwara 1981]
Hideo Fujiwara, K. Kinoshita, "A Design of Programmable Logic Arrays with Universal Tests," IEEE Trans. Computers, vol. c-30, Nov. 1981, pp.823-828.

10. [Hong 1980]
S. J. Hong, D. L. Ostapko, "FITPLA: A Programmable Logic Array for Functional Independent Testing," Digest 10th Int. Symp. Fault-Tolerant Computing, 1980, pp. 131-136.

11. [Hua 1984]
K. A. Hua, J. Y. Jou and J. A. Abraham, "Build-In Tests for VLSI Finite State Machine," Digest 14th Int. Symp on Fault-Tolerant Computing, June 1984, pp. 422-425.

12. [Hwang 1985]
Kai Hwang, Faye' A. Briggs, Computer Architecture and Parallel Processing, Reading, Mass., 1985.

13. [Khakbaz 1983]
J. Khakbaz, "A Testable PLA Design with Low Overhead and High Fault Coverage," Digest 13th Int. Symp. on Fault-Tolerant Computing, June 1983, pp. 426-429.

14. [Kohnen 1982]
Teuvo Kohonen, Content Addressable Memory, Springer-Verlag, Reading, Mass., 1982.

15. [Kohnen 1977]
Teuvo Kohonen, Associative Memory, Springer-Verlag, Reading, Mass., 1977.

16. [Marchand 1985]
J. F. Phippe Marchand, "An Alterable Programmable Logic Array," J. Solid-State Circuits, vol. sc-20, no. 5, Oct. 1985, pp. 1061-1066.

17. [Min 1984]
Y. Min, "A PLA Design for Ease of Test Generation," Digest 14th Int. Symp. on Fault-Tolerant Compouting, June 1984, pp. 436-442.

18. [Treuer 1985]
Robert Treuer, Hideo Fujkwara, Vinod K. Agarwal, "Implementing a Built-in Self-Test PLA Design," IEEE Design & Test, Apr. 1985, pp. 37-48.

19. [Wood 1981]
R. A. Wood, Y. Hsieh, C. A. Price and P. Wang, "An Electrically Alterable PLA for Fast Turnaround-Time VLSI Development Hardware," IEEE J. of Solid-State Circuits, vol. sc-16, no. 5, Oct. 1981, pp. 570-577.