HM530281 Series

Preliminary

331,776-word × 8-bit Frame Memory

The HM530281 series memory products provide completely asynchronous I/O and operate at the high speed of 50 MHz. A 0.8 µm CMOS process is used in their fabrication.

The HM530281 series memory products provide reset, jump, and line increment/hold pointer control functions that can be used in synchronization with independent clocks on each of the I/O ports. Memory can be accessed immediately without any waiting period after the execution of these functions.

In addition to the FIFO function, the 281 series products support an address structure that is compatible with HDTV, NTSC, and PAL standards, and can be used in a wide range of applications, such as noise reducers, TBC (time-based correction), inter-frame YC separation, and special function modes (e.g., multi-freeze, P-in-P) in the digital TV, VCR, and video camera application.

They are also appropriate for use as inter-system speed conversion buffer memories in communications systems.

Features

- Organization: 331,776 words × 8 bits
- Completely asynchronous operation of the serial read port and write port.
  - — Internal generation of read and write addresses
  - — Internal memory operation control provided on-chip
- High speed read/write cycle time: 50 MHz
- Reset, jump functions
  - — Independent execution for read and write ports
  - — Can be executed with arbitrary timing
  - — Allow immediate access after execution (read/write) (for the jump function, when the address setup is complete)
  - — Jump address specifiable in 32-word units
- 2 dimensional address
- Line increment/hold address pointer control function
- Window scan function
- Can handle HDTV, NTSC, and PAL standards
  - — Line length: Up to 1152 bits (Arbitrary line lengths can also be handled by using the line reset function.)
  - — Line count: Up to 324 lines
- Built-in self-refresh eliminates the need for external refresh control.
- Power supply voltage: $V_{CC} = 5.0$ V ± 10%.
- Low power consumption

Ordering Information

Product No.	Cycle time	Memory organization	Package
HM530281TT-20	20 ns	331,776 words × 8 bits^{*2}	44-pin TSOP
HM530281TT-25	25 ns	1152 dots × 288 lines × 8 bits^{*3}	(TTP-44DB)
HM530281TT-34	34 ns	1024 dots × 324 lines × 8 bits	
HM530281TT-45	45 ns		

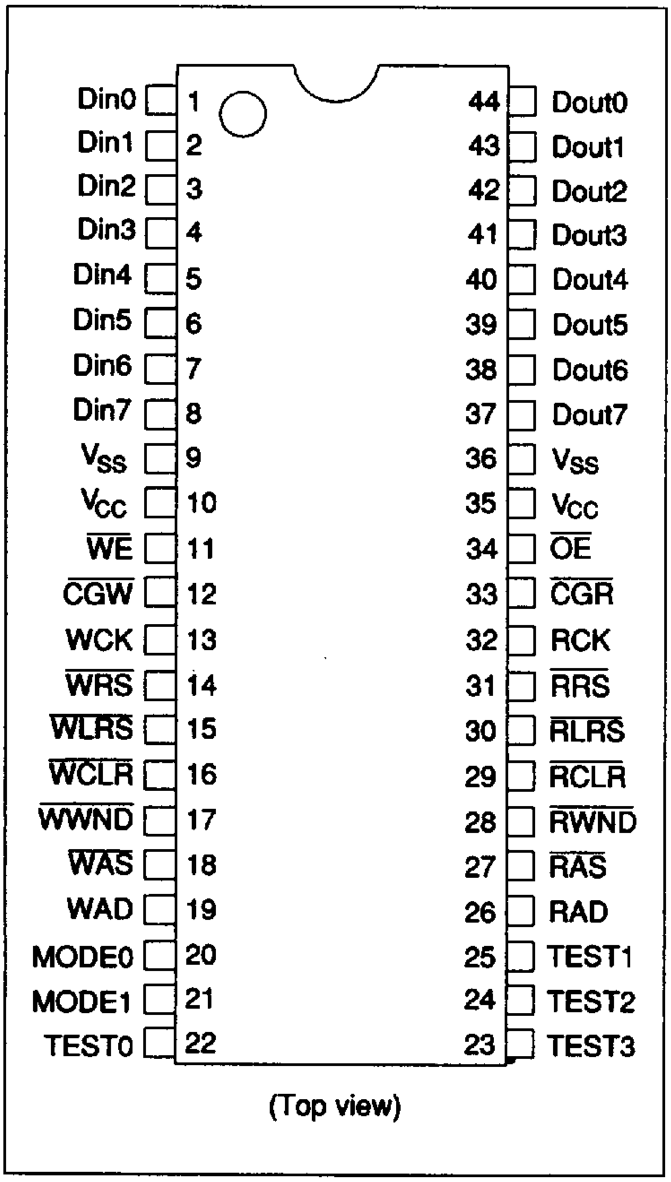
Notes: 1. Selectable following two kinds of addressing mode by mode pins

2. 1 dimensional addressing mode

3. 2 dimensional addressing mode

Note: The specifications of this device are subject to change without notice. Please contact your nearest Hitachi's Sales Dept. regarding specifications.

Pin Arrangement

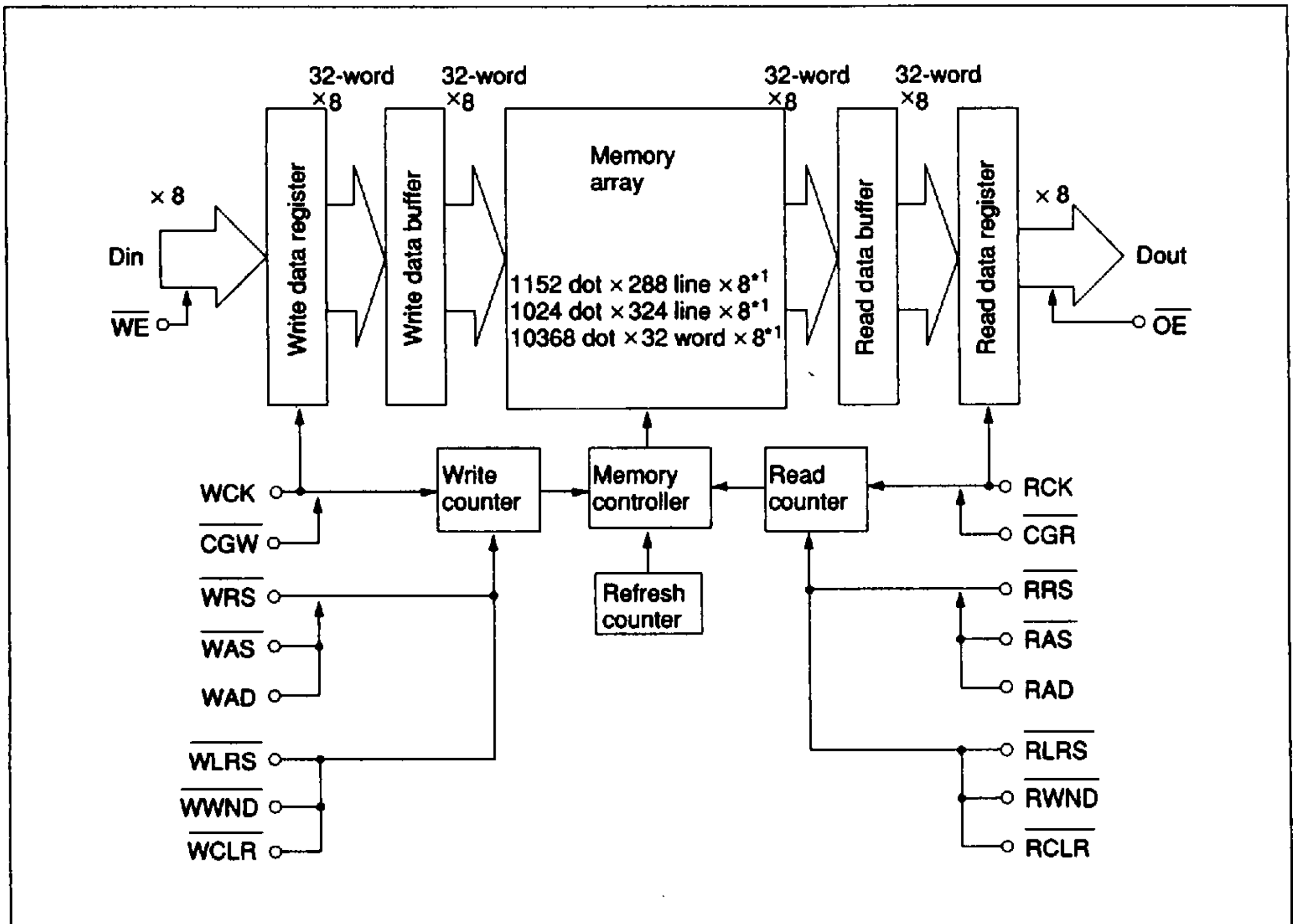

Din0 1
Din1 2
Din2 3
Din3 4
Din4 5
Din5 6
Din6 7
Din7 8
V_{SS} 9
V_{CC} 10
$\overline{\text{WE}}$ 11
$\overline{\text{CGW}}$ 12
WCK 13
$\overline{\text{WRS}}$ 14
$\overline{\text{WLRS}}$ 15
$\overline{\text{WCLR}}$ 16
$\overline{\text{WWND}}$ 17
$\overline{\text{WAS}}$ 18
WAD 19
MODE0 20
MODE1 21
TEST0 22
44 Dout0
43 Dout1
42 Dout2
41 Dout3
40 Dout4
39 Dout5
38 Dout6
37 Dout7
36 V_{SS}
35 V_{CC}
34 $\overline{\text{OE}}$
33 $\overline{\text{CGR}}$
32 RCK
31 $\overline{\text{RRS}}$
30 $\overline{\text{RLRS}}$
29 $\overline{\text{RCLR}}$
28 $\overline{\text{RWND}}$
27 $\overline{\text{RAS}}$
26 RAD
25 TEST1
24 TEST2
23 TEST3

(Top view)

Pin Functions

	Pin functions	
Symbol	**2 dim. add.**	**1 dim. add.**
Din0 to Din7	Data input	Data input
Dout0 to Dout7	Data output	Data output
WCK	Write clock	Write clock
RCK	Read clock	Read clock
$\overline{\text{WRS}}$	Write reset	Write reset
$\overline{\text{RRS}}$	Read reset	Read reset
$\overline{\text{WE}}$	Write enable	Write enable
$\overline{\text{OE}}$	Output enable	Output enable
$\overline{\text{CGW}}$	Write clock gate	Write clock gate
$\overline{\text{CGR}}$	Read clock gate	Read clock gate
$\overline{\text{WAS}}$	Write address set	Write address set
WAD	Write address	Write address
$\overline{\text{RAS}}$	Read address set	Read address set
RAD	Read address	Read address
$\overline{\text{WLRS}}$	Write line reset	V_{CC} or GND
$\overline{\text{RLRS}}$	Read line reset	V_{CC} or GND
$\overline{\text{WWND}}$	Write window mode	V_{CC} or GND
$\overline{\text{RWND}}$	Read window mode	V_{CC} or GND
$\overline{\text{WCLR}}$	Write clear	V_{CC} or GND
$\overline{\text{RCLR}}$	Read clear	V_{CC} or GND
MODE 0 to 1	Mode selection input	Mode selection input
V_{CC}	Power supply	Power supply
V_{SS}	Ground	Ground
TEST0 to TEST3	Connect to ground	Connect to ground

Block Diagram



Note : 1. Selected by the mode pin

Absolute Maximum Ratings

Item	Symbol	Rating	Unit
Pin voltage[*1]	V_T	–1.0 to +7.0	V
Power dissipation	P_T	1.0	W
Operating temperature	Topr	0 to +70	°C
Storage temperature	Tstg	–55 to +125	°C
Storage temperature (when biased)	Tbias	–10 to +85	°C

Note: 1. The permissible values with respect to V_{SS}.

Recommended DC Operating Conditions (Ta = 0 to +70°C)

Item	Symbol	Min	Typ	Max	Unit
Power supply voltage	V_{CC}	4.5	5	5.5	V
	V_{SS}	0	0	0	V
Input voltages	V_{IH}	2.7	—	6.5	V
	V_{IL}	–0.5*1	—	0.6	V

Note: 1. When the pulse width is under 10 ns, V_{IL} min = –3.0 V.

DC Characteristics (V_{CC} = 5.0 V ± 10%, V_{SS} = 0 V, Ta = 0 to +70°C)

		HM530281-20			HM530281-25			HM530281-34			HM530281-45				
Item	Symbol	Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	Unit	Test conditions
Operating power supply voltage	I_{CCA}	—	110	135	—	90	120	—	70	95	—	55	75	mA	Iout = 0, t_{WCC} = t_{RCC} = Min
Standby power supply voltage	I_{CCS}	—	15	25	—	15	25	—	15	25	—	15	25	mA	
Input leakage current	I_{LI}	–10	—	10	–10	—	10	–10	—	10	–10	—	10	mA	V_{CC} = 5.5 V, Vin = V_{SS} to V_{CC}
Output leakage current	I_{LO}	–10	—	10	–10	—	10	–10	—	10	–10	—	10	mA	$\overline{OE}$ = V_{IN} Vout = V_{SS} to V_{CC}
Output voltages	V_{OL}	—	—	0.4	—	—	0.4	—	—	0.4	—	—	0.4	V	I_{OL} = 2.1 mA
	V_{OH}	2.4	—	—	2.4	—	—	2.4	—	—	2.4	—	—	V	I_{OH} = –1.0 mA

Capacitances

Item	Symbol	Typ	Max	Units	Test conditions
Input capacitance	Cin	—	5	pF	Vin = 0 V
Output capacitance	Cout	—	7	pF	Vout = 0 V

Note: These parameters are sampled values, not values measured for all units.

AC Characteristics

Test Conditions

- Input pulse level: V_{SS} to 3.0 V
- Input rise/fall times: 3 ns
- I/O timing reference level: 1.5 V
- Output load: 1 TTL + 50 pF (including jig and scope capacitances)

		HM530281-20		HM530281-25		HM530281-34		HM530281-45		
Item	Symbol	Min	Max	Min	Max	Min	Max	Min	Max	Unit
Write clock cycle time	t_{WCC}	20	—	25	—	34	—	45	—	ns
Write clock pulse width (high)	t_{WC}	8	—	10	—	12	—	15	—	ns
Write clock pulse width (low)	t_{WCP}	8	—	10	—	12	—	15	—	ns
$\overline{\text{WRS}}$ setup time	t_{WRS}	8	—	10	—	10	—	10	—	ns
$\overline{\text{WRS}}$ hold time	t_{WRH}	8	—	10	—	10	—	15	—	ns
Data input setup time	t_{DS}	5	—	5	—	5	—	5	—	ns
Data input hold time	t_{DH}	10	—	12	—	12	—	15	—	ns
$\overline{\text{CGW}}$ setup time	t_{WGS}	8	—	10	—	10	—	10	—	ns
$\overline{\text{CGW}}$ hold time	t_{WGH}	8	—	10	—	10	—	15	—	ns
$\overline{\text{WE}}$ setup time	t_{WES}	5	—	5	—	5	—	5	—	ns
$\overline{\text{WE}}$ hold time	t_{WEH}	10	—	12	—	12	—	15	—	ns
Read clock cycle time	t_{RCC}	20	—	25	—	34	—	45	—	ns
Read clock pulse width (high)	t_{RC}	8	—	10	—	12	—	15	—	ns
Read clock pulse width (low)	t_{RCP}	8	—	10	—	12	—	15	—	ns
$\overline{\text{RRS}}$ setup time	t_{RRS}	8	—	10	—	10	—	10	—	ns
$\overline{\text{RRS}}$ hold time	t_{RRH}	8	—	10	—	10	—	15	—	ns
Access time from RCK	t_{RAC}	—	18	—	23	—	30	—	35	ns
Output hold time	t_{OH}	5	—	5	—	5	—	5	—	ns
Output enable time	t_{OLZ}	0	—	0	—	0	—	0	—	ns
Output enable access time	t_{OAC}	—	18	—	20	—	25	—	25	ns
Output disable time	t_{OHZ}	0	15	0	18	0	20	0	20	ns

AC Characteristics (cont)

		HM530281-20		HM530281-25		HM530281-34		HM530281-45		
Item	Symbol	Min	Max	Min	Max	Min	Max	Min	Max	Unit
$\overline{\mathrm{CGR}}$ setup time	t_{RGS}	8	—	10	—	10	—	10	—	ns
$\overline{\mathrm{CGR}}$ hold time	t_{RGH}	8	—	10	—	10	—	15	—	ns
$\overline{\mathrm{WAS}}$ setup time	t_{WSS}	8	—	10	—	10	—	10	—	ns
$\overline{\mathrm{WAS}}$ hold time	t_{WSH}	8	—	10	—	10	—	15	—	ns
$\overline{\mathrm{RAS}}$ setup time	t_{RSS}	8	—	10	—	10	—	10	—	ns
$\overline{\mathrm{RAS}}$ hold time	t_{RSH}	8	—	10	—	10	—	15	—	ns
Write address input setup time	t_{WAS}	5	—	5	—	5	—	5	—	ns
Write address input hold time	t_{WAH}	10	—	12	—	12	—	15	—	ns
Read address input setup time	t_{RAS}	5	—	5	—	5	—	5	—	ns
Read address input hold time	t_{RAH}	10	—	12	—	12	—	15	—	ns
$\overline{\mathrm{WLRS}}$ setup time	t_{WLS}	8	—	10	—	10	—	10	—	ns
$\overline{\mathrm{WLRS}}$ hold time	t_{WLH}	8	—	10	—	10	—	15	—	ns
$\overline{\mathrm{RLRS}}$ setup time	t_{RLS}	8	—	10	—	10	—	10	—	ns
$\overline{\mathrm{RLRS}}$ hold time	t_{RLH}	8	—	10	—	10	—	15	—	ns
$\overline{\mathrm{WCLR}}$ setup time	t_{WCLS}	8	—	10	—	10	—	10	—	ns
$\overline{\mathrm{WCLR}}$ hold time	t_{WCLH}	8	—	10	—	10	—	15	—	ns
$\overline{\mathrm{RCLR}}$ setup time	t_{RCLS}	8	—	10	—	10	—	10	—	ns
$\overline{\mathrm{RCLR}}$ hold time	t_{RCLH}	8	—	10	—	10	—	15	—	ns
$\overline{\mathrm{WWND}}$ setup time	t_{WWDS}	8	—	10	—	10	—	10	—	ns
$\overline{\mathrm{WWND}}$ hold time	t_{WWDH}	8	—	10	—	10	—	15	—	ns
$\overline{\mathrm{RWND}}$ setup time	t_{RWDS}	8	—	10	—	10	—	10	—	ns
$\overline{\mathrm{RWND}}$ hold time	t_{RWDH}	8	—	10	—	10	—	15	—	ns

Input and Output Pin Functions

$D_{IN}0$ to $D_{IN}7$ (data input) Input: The D_{IN} pins input 8 bits of data. Data is input on the rising edge of the cycle of WCK that follows a low level on both $\overline{\text{CGW}}$ and $\overline{\text{WE}}$.

$D_{OUT}0$ to $D_{OUT}7$ (data output) Output: The D_{OUT} pins output 8 bits of data. Data output is synchronized with the RCK clock, and the access time is specified from the rising edge of the RCK cycle.

WCK (write clock) Input: WCK is the write clock input pin. The input of write data is synchronized with this clock.

Write data is input on the rising edge of the cycle of WCK that follows a low level on both $\overline{\text{CGW}}$ and $\overline{\text{WE}}$, and when $\overline{\text{CGW}}$ is low, the internal write address pointer is incremented at the same time.

Input of the write jump address is also synchronized with this clock. The 14 bits of the write jump address are read in sequentially from the WCK cycle that set $\overline{\text{WAS}}$ low, irrespective of write data acquisition.

RCK (read clock) Input: RCK is the read clock input pin. Read data is output in synchronization with this clock when both $\overline{\text{CGR}}$ and $\overline{\text{OE}}$ are low, and when $\overline{\text{CGR}}$ is low, the internal read address pointer is incremented at the same time.

Input of the read jump address is also synchronized with this clock. The read jump address is read in sequentially starting at the RCK cycle in which $\overline{\text{RAS}}$ was set low, independently of read data output.

$\overline{\text{WRS}}$ (write address pointer reset) Input: $\overline{\text{WRS}}$ is a reset signal input that resets the write address pointer to 0 when $\overline{\text{WAS}}$ and $\overline{\text{WLRS}}$ are high, resets to the head of the line currently being written when $\overline{\text{WAS}}$ is high and $\overline{\text{WLRS}}$ is low, and jumps to the preset write jump address when $\overline{\text{WAS}}$ is low. *1

Only the falling edge of this reset input is detected, and, on the first WCK cycle following that falling edge, a write cycle to the set address is started immediately. Note that this reset operation is performed regardless of the values of $\overline{\text{CGW}}$ and $\overline{\text{WE}}$.

$\overline{\text{RRS}}$ (read address pointer reset) Input: $\overline{\text{RRS}}$ is a reset signal input that resets the read address pointer to 0 when $\overline{\text{RAS}}$ and $\overline{\text{RLRS}}$ are high, resets to the start of the line currently being read when $\overline{\text{RAS}}$ is high and $\overline{\text{RLRS}}$ is low, and jumps to the read jump address when $\overline{\text{RAS}}$ is low.*1

Only the falling edge of this reset input is detected, and, on the first RCK cycle following that falling edge, a read cycle at the set address is started immediately. Note that this reset operation is performed regardless of the levels of $\overline{\text{CGR}}$ and $\overline{\text{OE}}$.

$\overline{\text{WE}}$ (write enable) Input: $\overline{\text{WE}}$ is an input signal that controls the enabling/disabling of the data input pins. When $\overline{\text{WE}}$ is low, input data is acquired on the WCK cycle, and when $\overline{\text{WE}}$ is high, data input is disabled and the previous memory data is maintained.

Note that the write address pointer is incremented by the WCK write clock without regard for the level of $\overline{\text{WE}}$.

$\overline{\text{OE}}$ (output enable) Input: $\overline{\text{OE}}$ is an input signal that enables/disables the data output pins. When $\overline{\text{OE}}$ is low, data output is enabled, and when high, data output is disabled and the output pins go to the high impedance state.

Note that the read address pointer is incremented by the RCK read clock without regard for the level of $\overline{\text{OE}}$. Therefore, data can be jumped over during read simply by disabling output with $\overline{\text{OE}}$.

$\overline{\text{CGW}}$ (clock gate for write) Input: $\overline{\text{CGW}}$ is an input signal that enables/disables incrementing of the internal write address pointer. When $\overline{\text{CGW}}$ is low, the write address pointer is incremented in synchronization with the WCK write clock, and when high, incrementing is stopped. Therefore time axis compression can be easily implemented without stopping the write clock by using $\overline{\text{CGW}}$.

$\overline{\text{CGR}}$ (clock gate for read) Input: $\overline{\text{CGR}}$ is an input signal that enables/disables incrementing of the internal read address pointer. When $\overline{\text{CGR}}$ is low, the read address pointer is incremented in synchronization with the RCK read clock, and when high, incrementing is stopped. Therefore time axis expansion can be easily implemented without stopping the read clock by using CGR.

WAS (write address set and jump) Input: WAS is an input signal that initiates write jump address input when WRS is high and jumps to the previously input write jump address when WRS is low. The falling edge of this input signal is detected, and either a write jump address input is initiated or a jump to the previously input write jump address is executed on the first WCK cycle following the fall of WAS.

Note that this operating is executed without regard for the levels of CGW and WE.

WAD (write jump address) Input: WAD is the input pin for the write jump address. The 14/15 bits of the write jump address are read in sequentially from the high order bit, starting at the WCK cycle (when WRS was high) in which WAS was set low.*2

RAS (read address set and jump) Input: RAS is an input signal that initiates read jump address input when RRS is high and jumps to the previously input read jump address when RRS is low. The falling edge of this input signal is detected, and either the read jump address input is initiated or the jump to the previously input read jump address is executed on the first WCK cycle following the fall on RAS.

Note that operation is executed without regard for the levels of CGR and OE.

RAD (read jump address) Input: RAD is the input pin for the read jump address. The 14/15 bits of the write jump address are read in sequentially from the high order bit, starting at the RCK cycle (when RRS was high) in which RAS was set low.*2

WLRS (write line reset) Input (in 2 dimensional addressing mode): WLRS is an input pin for resetting the write address pointer to the start of the line from an arbitrary dot for each line.*3 Only the falling edge of this signal is detected, and, on the first WCK cycle following that falling edge, the write address pointer is set to the head of the next line when WRS is high, and to head of the current line when WRS is low.*3 Note that this reset operation is performed regardless of the levels of CGW and WE.

RLRS (read line reset) Input (in 2 dimensional addressing mode): RLRS is an input pin for resetting the read address pointer to the start of the line from an arbitrary dot for each line.*3 Only the falling edge of this signal is detected, and, on the first RCK cycle following that falling edge, the write address pointer is set to the head of the next line when RRS is high, and to head of the current line when RRS is low.*3 Note that this reset operation is performed regardless of the levels of CGR and OE.

WWND (write window scan) Input (in 2 dimensional addressing mode): WWND is an input signal that specifies the use of the window scan function. When executing a write jump with WRS and WAS low, if WWND is set low at the same time, a scan of the window region that takes that write jump address as its starting point will begin (see note below).

RWND (read window scan) Input (in 2 dimensional addressing mode): RWND is an input signal that specifies the use of the window scan function. when executing a read jump with RRS and RAS low, if RWND is set low at the same time, a scan of the window region that takes that read jump address as its starting point will begin.*4

WCLR (write clear) Input: WCLR is an input signal that, independently of the levels on all other input pins, resets the write address pointer to 0 and clears the window scan function. This function is executed immediately in the WCK cycle in which WCLR was set low. This clear operation should also be performed after applying power to the HM530281.

RCLR (read clear) Input: RCLR is an input signal that, independently of the levels on all other input pins, resets the read address pointer to 0 and clears the window scan function. This function is executed immediately in the RCK cycle in which

$\overline{\text{RCLR}}$ was set low. This clear operation should also be performed after applying power to the HM530281.

Notes: 1. The reset destination in window scan mode changes as follows.
Reset to 0: Reset to the window start.
Reset to line start: Reset to the point at the left edge of the window for the line

2.

Addressing mode	Address structure	Input address
1 dim. add. (FIFO)	0 to 10,367 blocks	Address bits A_{13} to A_0
2 dim. add. (1)	32 horizontal blocks by 324 vertical lines	Line address bits V_8 to V_0, horizontal address bits H_4 to H_0
2 dim. add. (2)	36 horizontal blocks by 288 vertical lines.	Line address bits V_8 to V_0, horizontal address bits H_5 to H_0

3. When window scan mode is set, the reset is to the point at the left edge of the window for the line.

4. When window scan is set, the horizontal address of the pointer reset destination when increment/hold is executed will be the left edge of the window. Also, when a reset is executed, the pointer will be reset to the starting point of the window.
Thus it is possible to scan arbitrary window regions within the screen independently for read and write by using these line reset and reset functions.

Memory Structure

The meomry is organized as 331,776 words of 8 bits each, and these words can be accessed sequentially, since the address pointer can be incremented by inputting a clock signal. Addresses are allocated corresponding to 32 word blocks.

The mode pins switch between the three addressing modes shown below.

Mode 0	Mode 1	Addressing mode	Address structure	Capacity
0	0	1 dim. add. (FIFO)	0 to 10,367 blocks	331,776 words
1	0	2 dim. add. (1)	32 horizontal blocks by 324 vertical lines	1024 dots by 324 lines
0	1	2 dim. add. (2)	36 horizontal blocks by 288 vertical lines	1152 dots by 288 lines

Notes: 1. In 1 dimensional addressing mode, blocks 0 to 10367 are accessed cyclically.
2. In the 2 dimensional addressing modes, the line head can be reset at an arbitrary dot on each line.

Operations

Write

Write operation: When the $\overline{\text{WE}}$ and $\overline{\text{CGW}}$ inputs are low, 8 bits of write data are input in synchronization with the WCK clock. The input data is read in to the word indicated by the address pointer on the next rising edge of the WCK cycle. This allows read data and write data to be handled with the same clock, and cascade connections to be easily implemented.

Reset operations (write): By setting $\overline{\text{WRS}}$ low, the write address pointer can be set immediately on that WCK cycle to the address 0 block head.

Address pointer increment operations (write): The write address pointer is incremented in synchronization with WCK hen $\overline{\text{CGW}}$ is low.

It is possible to apply a write mask in WCK clock units by setting the $\overline{\text{WE}}$ input high. In this case, the previous memory data will be retained. The write address pointer increment function can be stopped by setting the $\overline{\text{CGW}}$ input high. This allows time axis compression to be implemented easily.

If the reset interval specifications are met (see Notes 8 and 9), the write reset operation can be performed on an arbitrary WCK clock cycle without regard for the levels of the $\overline{\text{WE}}$ and $\overline{\text{CGW}}$ inputs.

$\overline{\text{WE}}$ and $\overline{\text{CGW}}$ Input Level, Write Address Pointer, and Data Input State Relationship

WCK rising edge		Internal write address pointer	Data input
$\overline{\text{CGW}}$	$\overline{\text{WE}}$		
L	L	Incremented	enable
L	H		disable (memory data is retained)
H	—	Stopped	

Note: Data is input when the $\overline{\text{WE}}$ input is low.

Read

Read operation: 8 bits of read data are output in synchronization with the RCK clock when the $\overline{\text{OE}}$ and $\overline{\text{CGR}}$ inputs are low. The access time is stipulated from the rising edge of the RCK clock.

Reset operations (read): By setting $\overline{\text{RRS}}$ low, the read address pointer can be set immediately on that RCK cycle to address 0 and the data will then be output.

Address pointer increment operations (read): The read address pointer is incremented in synchronization with RCK when $\overline{\text{CGR}}$ is low.

Data outputs go to the high impedance state when the $\overline{\text{OE}}$ input is set high. The reset address pointer increment function can be stopped by setting the $\overline{\text{CGR}}$ input high. This allows time axis expansion to be implemented easily.

If the reset interval specifications are met (see Notes 7), the write reset operation can be performed on a arbitrary RCK clock cycle without regard for the levels of the $\overline{\text{OE}}$ and $\overline{\text{CGR}}$ inputs.

Relation Between the $\overline{\text{OE}}$ and $\overline{\text{CGR}}$ Input Levels and the Read Address Pointer and Data Output States

RCK rising edge		Internal read address pointer	Data output
$\overline{\text{CGR}}$	$\overline{\text{OE}}$		
L	L	Incremented	Output
L	H		High impedance
H	L	Stopped	Output data held
H	H		High impedance

Note: Data is input when the $\overline{\text{OE}}$ input is low.

Line Reset (independent functions for read and write, in 2 dimensional addressing modes)

When the 281 series products are used in 2 dimensional addressing modes, the line length can be set to be either 1024 dots (2 dimensional (1)) or 1152 dots (2 dimensional (2)). In these modes, after accessing the data at the last dot (address) on each line, address pointer incrementing is stopped. Access is restarted at either the first dot at the head of the next line or at the first dot at the head of the current line by executing either a line increment or a line hold, respectively. Also, since these line reset operations can be executed at any arbitrary point in the middle of a line, an arbitrary line length (of between 64 dots and the actual line length) can be realized.

Line increment operation: The read and write line increment operations are executed by setting $\overline{\text{RLRS}}$ low and $\overline{\text{RRS}}$ high, and setting $\overline{\text{WLRS}}$ low and $\overline{\text{WRS}}$ high respectively. When these operations are executed, the next access goes immediately to the starting dot of the next line.

Line hold operation: The read and write line hold operations are executed by setting $\overline{\text{RLRS}}$ and $\overline{\text{RRS}}$ low, and setting $\overline{\text{WLRS}}$ and $\overline{\text{WRS}}$ low respectively. When these operations are executed, the next access goes immediately to the starting dot of the current line. Note that the read line hold operation is invalid on the first line following a 0 reset or jump. In this case, the same effect can be achieved by re-executing the reset or jump operation (resetting only the H address to 0).

If the reset interval specifications are met (see Notes on Usage 1 to 3), the line reset operation can be performed on an arbitrary RCK clock cycle without regard for the levels of the $\overline{\text{OE}}$, $\overline{\text{CGR}}$, $\overline{\text{WE}}$, and $\overline{\text{CGW}}$ inputs.

Jump (independent functions for read and write)

It is possible to set the address pointer to the start address of an arbitrary block in 32 word units. After initializing a jump address setup for read and/or write, after 64 WCK or 64 RCK cycles, it is possible to execute a jump to that address (random access in 32 word by 8 bit units) independently for read and write.

Jump address setup: The read and write jump addresses are serially input independently from the RAD and WAD pins in synchronization with the RCK and WCK clock inputs respectively. Address input start is enabled by setting the $\overline{\text{RAS}}$ and/or $\overline{\text{WAS}}$ inputs low for read and write respectively, and 14/15 bits of jump address are input sequentially starting with that cycle.*10 Note that the read and write operations can continue independently of this address input operation.

Following the start of address input, it is possible to mask the input of address bits below an arbitrary bit position by returning $\overline{\text{RAS}}$ or $\overline{\text{WAS}}$ to the high level at the desired bit position. This can be convenient in applications that need to jump a fixed interval, since the low order bits of the address will be fixed. When all 14 bits of an address are to be input, be sure to hold $\overline{\text{RAS}}$ and $\overline{\text{WAS}}$ low for the full 14-clock period.

Jump operation: The jump operation is executed by setting $\overline{\text{RRS}}$ and $\overline{\text{RAS}}$ low for read, and by setting $\overline{\text{WRS}}$ and $\overline{\text{WAS}}$ low for write, and the address set is accessed immediately from that RCK or WCK cycle. Note that as long as the interval specifications listed in Notes 7 to 9 are met, the jump operation can be executed on any RCK or WCK cycle without regard for the values of $\overline{\text{OE}}$ and $\overline{\text{CGR}}$ on read, and $\overline{\text{WE}}$ and $\overline{\text{CGW}}$ on write.

Window Scan (independent functions for read and write)

The window scan function can be used with either the 2 dimensional (1) or (2) addressing modes, and is a function which scans a rectangular region with an arbitrary starting point. The jump address setup function (see Jump address setup above) is used to specify the starting point

Initiating window scan: The window scan function is started by setting $\overline{\text{WWIND}}$ to low for read or $\overline{\text{RWIND}}$ low for write, and executing a read or write jump operation (see Jump operation above). Window scan will start immediately from that cycle.

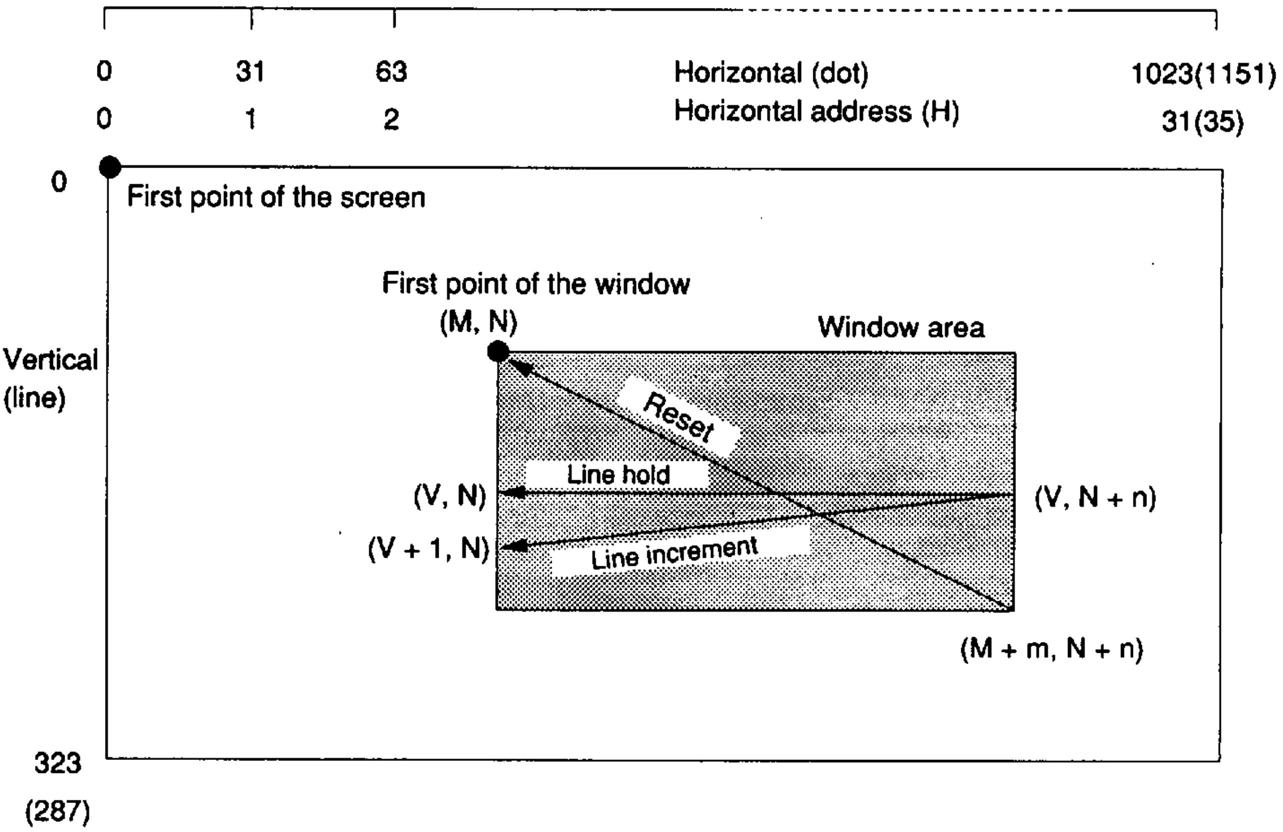
Window scan operation: When the window scan function is started, one of the functions described below will be executed independently for read and write.*11

Also note that as long as the interval conditions listed in Notes 7 to 9 are met, these operations can be executed at arbitrary dots without regard for the address block organization.

Clearing window scan: The window scan function is turned off either by executing a reset or jump with $\overline{\text{RWIND}}$ (for read) or $\overline{\text{WWIND}}$ (for write) set high, or by executing the clear operation described in section Clear below.

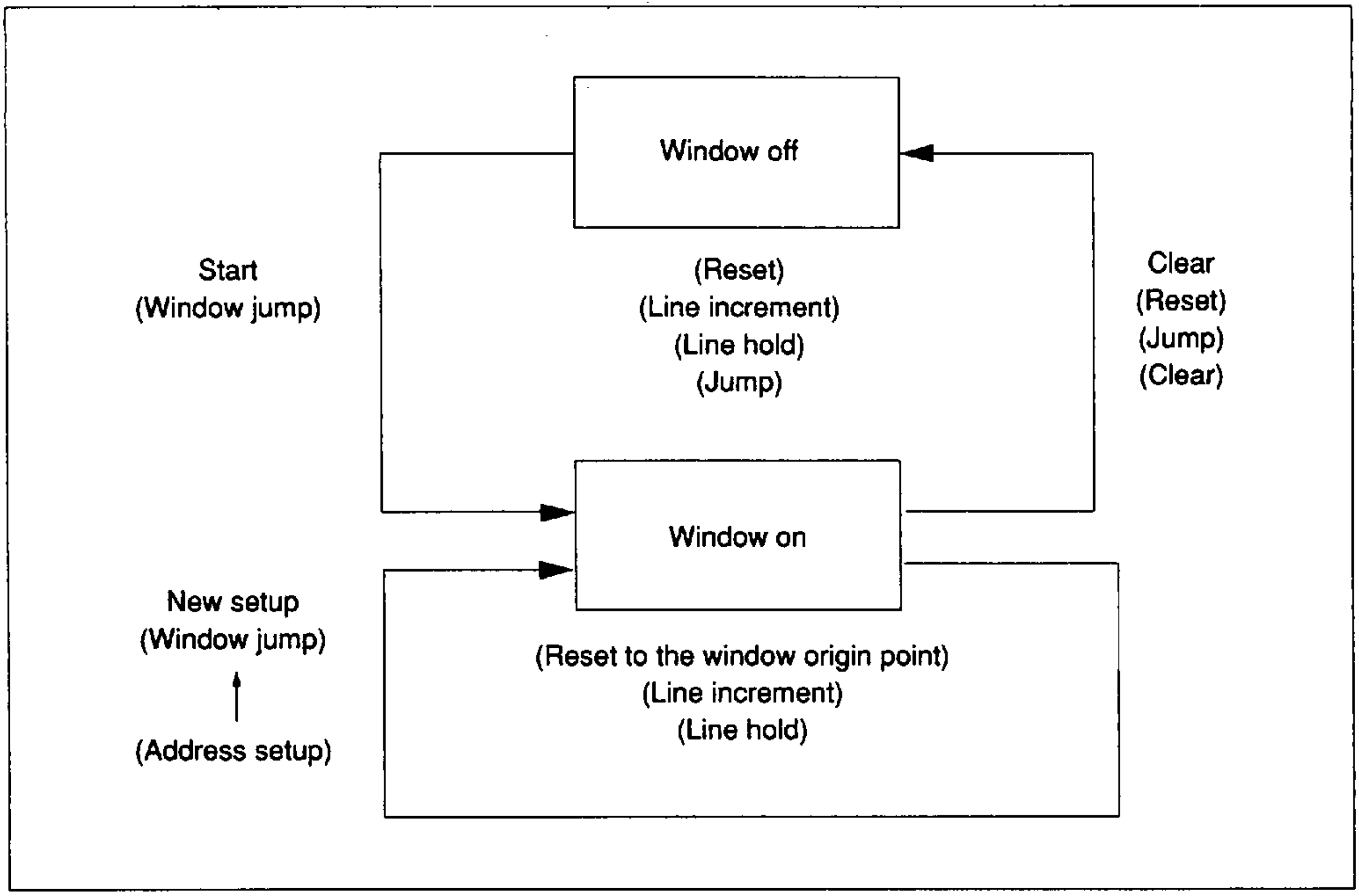
Operation	Address pointer control
Reset	Reset to the first dot at the start of the window.
Line increment	Reset to the first dot at the left edge of the window on the next line.
Line hold	Reset to the first dot at the left edge of the window on the current line.

Overview of the window scan operation:



Note: M and N are addresses, M is in line units, N is units of 32 dots, and m and n are in line and dot units respectively.

Starting and clearing window scan:



Clear (independent functions for read and write)

The clear function both resets the address pointer to 0 without regard for the value on any other pin, and if window mode is set, clears window mode.

Clear Operation: The clear operation can be executed on any cycle by setting the $\overline{\text{RCLR}}$ pin low for read and the $\overline{\text{WCLR}}$ pin low for write. The data input following a write clear is valid data. If the interval conditions listed in Notes 7 to 9 are met by the operations including the clear operation, then the data preceding the clear operation will also be valid data.

However, note that read system resets (reset, jump, line reset, clear) and the read address setup operation cannot be executed for consecutive RCK clock cycles. Similarly, write system resets (reset, jump, line reset, clear) and the write address setup operation cannot be executed for consecutive WCK clock cycles.

Access of New and Previous Data

New data access (follow-up read out of data currently being written): Written data can be read out 160 WCK cycles after it is written. However, it is necessary to execute the read jump address setup operation outside the time period between 32 WCK cycles before write to that address is started and 32 WCK cycles after write to that address is completed.

- It is also possible to read out the new data when jumping to an address at least 160 WCK clock cycles after write to that address was started. Note that in this case, there is more than enough time for the read jump address setup operation even if it is begun 32 or more clock cycles after the completion of the write operation.

Previous data access (reading out data prior to that of the current write oepration): The previous data ca be read out up to 32 WCK clock cycles after the write operation.

Therefore, these memories can be used to provide delay times of between 160 and 331,808 (331,776 + 32) clock cycles.

Power On

Wait at least 100 µs after power-on to begin operation. At this time the write and read address pointers are undefined.

The following operation should be executed.

- $\overline{\text{CGW}}$ and $\overline{\text{CGR}}$ should be hold low.
- Reset cycle when 1 dimensional addressing mode.
- Clear cycle when 2 dimensional addressing mode.
- Dummy cycle of over 64 WCK and 64 RCK clock cycle.

Then, initiate the desired operating mode by providing the signal input combination given by the truth tables below.

Function Table

1 Dimensional Addressing Modes

- **Write**

WCK rising edge

$\overline{\text{WRS}}$	$\overline{\text{WAS}}$	Operation	
H	H	Normal state	In the normal state, the write address pointer is incremented in synchronization with WCK.
L	H	Reset	The write address pointer is reset to 0.
L	L	Jump	Jump to the address A to which the write address pointer is set.
H	L	Address setup	The write jump address is input.

- **Read**

RCK rising edge

$\overline{\text{RRS}}$	$\overline{\text{RAS}}$	Operation	
H	H	Normal state	In the normal state, the read address pointer is incremented in synchronization with RCK.
L	H	Reset	The read address pointer is reset to 0.
L	L	Jump	Jump to the address A to which the read address pointer is set.
H	L	Address setup	The read jump address is input.

2 Dimensional Addressing Modes (when window scan is not used)

- **Write** *1

Levels at the rise of WCK						Operation		
$\overline{\text{WRS}}$	$\overline{\text{WAS}}$	$\overline{\text{WLRS}}$	$\overline{\text{WWND}}$	$\overline{\text{WCLR}}$		**Write address pointer control**	**Write jump address**	**Notes**
H	H	H	H	H	Normal state	Incremented in synchronization with WCK	—	*2
L	H	H	H	H	Reset	Reset to (0, 0)	—	
L	L	H	H	H	Jump	Jump to the set address A	—	
H	L	H	H	H	Address set	—	Set	
H	H	L	H	H	Line increment	Reset to the first bit of the next line	—	*2
L	H	L	H	H	Line hold	Reset to the first bit of the current line	—	*2
—	—	—	—	L	Clear	Reset to (0, 0)		

(—: Don't care)

- **Read***1

Levels at the rise of WCK						Operation		
$\overline{\text{RRS}}$	$\overline{\text{RAS}}$	$\overline{\text{RLRS}}$	$\overline{\text{RWND}}$	$\overline{\text{RCLR}}$		**Read address pointer control**	**Read jump address**	**Notes**
H	H	H	H	H	Normal state	Incremented in synchronization with RCK	—	*3
L	H	H	H	H	Reset	Reset to (0, 0)		
L	L	H	H	H	Jump	Jump to the set address A	—	
H	L	H	H	H	Address set	—	Set	
H	H	L	H	H	Line increment	Reset to the first bit of the next line	—	*3
L	H	L	H	H	Line hold	Reset to the first bit of the current line	—	*3
—	—	—	—	L	Clear	Reset to (0, 0)	—	

(—: Don't care)

2 Dimensional Address Modes (when window scan is not used)

- Write

Levels at the rise of WCK					Operation	Write address pointer control		Write jump add-ress	Window mode after execu-tion	Notes
WRS	WAS	WLRS	WWND	WCLR		Window mode off	Window mode on			
L	H	H	H	H	Reset	Reset to (0, 0)		—	Off	
H	H	H	—	H	Normal state	Incremented in synchronization with WCK		—	—	*4
H	H	L	—	H	Line increment	To the first bit of the next line	To the left edge of the window on the next line	—	—	
L	H	L	—	H	Line hold	To the first bit of the current line	To the left edge of the window on the current line	—	—	
H	L	H	—	H	Address set	—		Set	—	
L	L	H	H	H	Jump	Jump to the set address A		—	Off	
L	L	H	L	H	Window jump	Jump to the set address A		—	On	*6
L	H	H	L	H	Reset	Reset to the window origin point A		—	—	
—	—	—	—	L	Clear	Reset to (0, 0)		—	Off	

(—: Don't care)

- **Read**

Levels at the rise of WCK					Operation	Read address pointer control		Read jump address	Window mode after execution	Notes
$\overline{\text{RRS}}$	$\overline{\text{RAS}}$	$\overline{\text{RLRS}}$	$\overline{\text{RWND}}$	$\overline{\text{RCLR}}$		Window mode off	Window mode on			
L	H	H	H	H	Reset	Reset to (0, 0)		—	Off	
H	H	H	—	H	Normal state	Incremented in synchronization with RCK		—	—	*5
H	H	L	—	H	Line increment	To the first bit of the next line	To the left edge of the window on the next line	—	—	
L	H	L	—	H	Line hold	To the first bit of the current line	To the left edge of the window on the current line	—	—	
H	L	H	—	H	Address set	—		Set	—	
L	L	H	H	H	Jump	Jump to the set address A		—	Off	
L	L	H	L	H	Window jump	Jump to the set address A		—	On	*6
L	H	H	L	H	Reset	Reset to the window origin point A		—	—	
—	—	—	—	L	Clear	Reset to (0, 0)		—	Off	

(—: Don't care)

Notes: 1. Hold the $\overline{\text{WWND}}$ and $\overline{\text{RWND}}$ pin high when window mode is not used.

2. The write address pointer is incremented up to the last dot on the current line, and then stopped. Writing is started immediately from the first dot on the next line by execution of the line increment operation. Also, writing is started immediately from the first dot on the current line by execution of the line hold operation.

3. The read address pointer is incremented up to the last dot on the current line, and then stopped. Reading is started immediately from the first dot on the next line by execution of the line increment operation. Also, reading is started immediately from the first dot on the current line by execution of the line hold operation.

4. The write address pointer is incremented up to the last address on the line, and then stopped. Writing is started immediately from the first dot on the next line or the left edge of the window by execution of the line increment operation.

5. The read address pointer is incremented up to the last address on the line, and then stopped. Reading is started immediately from the first dot on the next line or the left edge of the window by execution of the line increment operation.

6. It is possible to move directly from an old window to a new window in window mode by setting up a new jump address and executing a window setup jump operation. However, the new jump address should be input after access to the last line of the old window.
7. Read system reset (reset, jump, line reset, and clear) operations and read jump address set operations must be performed at times separated by at least 64 RCK clock cycles. (There is no absolute need to use only 32 word addressing units, and these operations can be performed on any clock cycle).
8. Write system reset operations (reset, jump, line reset, and clear) must be performed at times separated by at least 64 WCK clock cycles.
9. It is also possible to reset the write system in the middle of 32 word unit addressing. In this case, not only must the condition of note 8 be met, but furthermore, pairs of write system resets for units of less than 32 words must be separated by at least 160 WCK clock cycles. When the write system is reset at less than 32 words, the data up to the point to which the address pointer has advanced will be written, and the remaining data will retain the old values. (Note that after the completion of a write of less than 32 words, a write reset is required to write the data for the last address into the memory array.)
10.

Addressing mode	Address structure	Input address
1 dim. add. (FIFO)	0 to 10,367 blocks	Address bit A13 to A0
2 dim. add. (1)	32 horizontal blocks by 324 vertical lines	Line address bits V8 to V0, horizontal address bits H4 to H0
2 dim. add. (2)	36 horizontal blocks by 288 vertical lines	Line address bits V8 to V0, horizontal address bits H5 to H0

11. Specifiable window sizes
Horizontal: Between 64 dots and the length of the line.
Vertical: Between 1 line and the maximum number of lines.
12. Location 0 cannot be specified as a jump address. Use a reset to access location 0.
13. Read system and write system resets are performed when $\overline{\text{CGR}}$ and $\overline{\text{CGW}}$ are high, respectively. The desired reset is actually executed on the first RCK or WCK clock cycle following $\overline{\text{CGR}}$ or $\overline{\text{CGW}}$ going low.
However, do not input invalid reset pulses when $\overline{\text{CGR}}$ or $\overline{\text{CGW}}$ is high, even if RCK or WCK is low. It is possible for an invalid reset to be latched mistakenly.

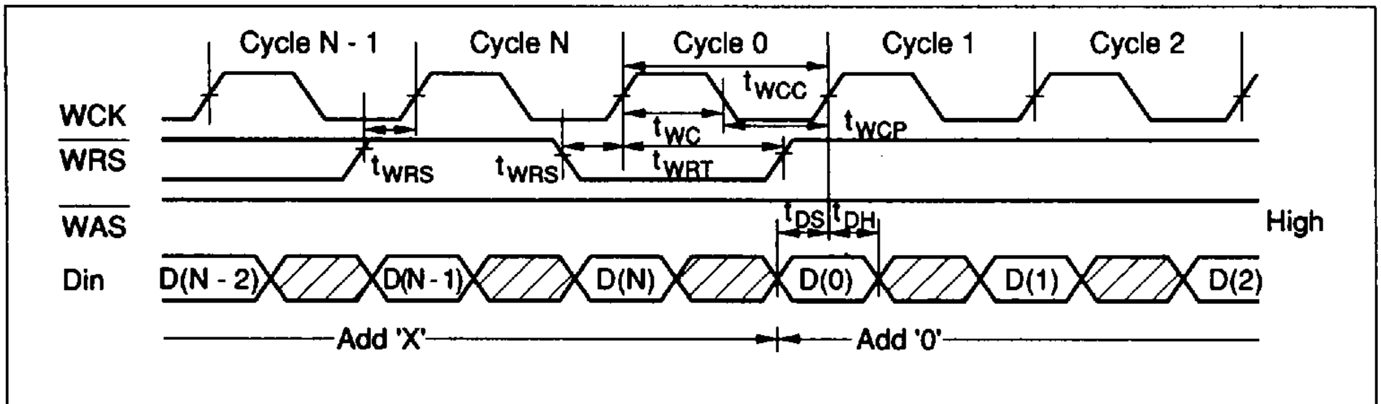
Supplement

If the read system reset interval (at least 64 RCK clock cycles) of note 7, or the write system reset interval for less than 32 word units (and at least 160 WCK clock cycles) are not provided (see note 9), it is possible for the 32 words of data of the first address after the reset to be invalid, or for the first write of less than 32 words following the write reset to fail to occur. However, even in this case, address pointer control will function corectly, and valid data will be output for the second and following addresses. (However, in this case the condition of note 8 and the 32 clock or longer read system reset/read jump address interval must be provided.)

Timing Waveforms

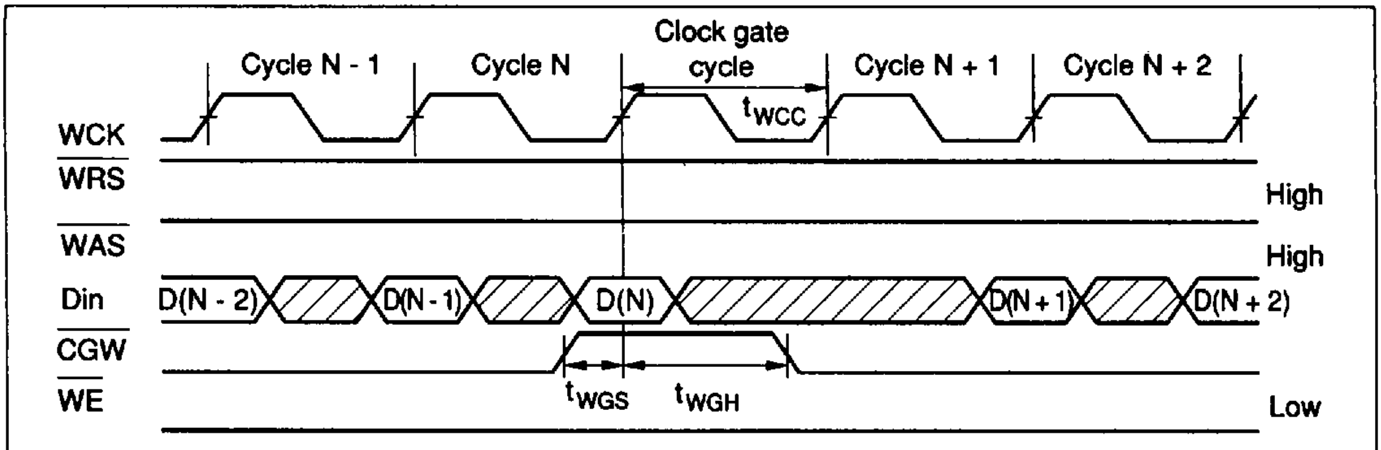
Write Cycle

- **Write address reset**



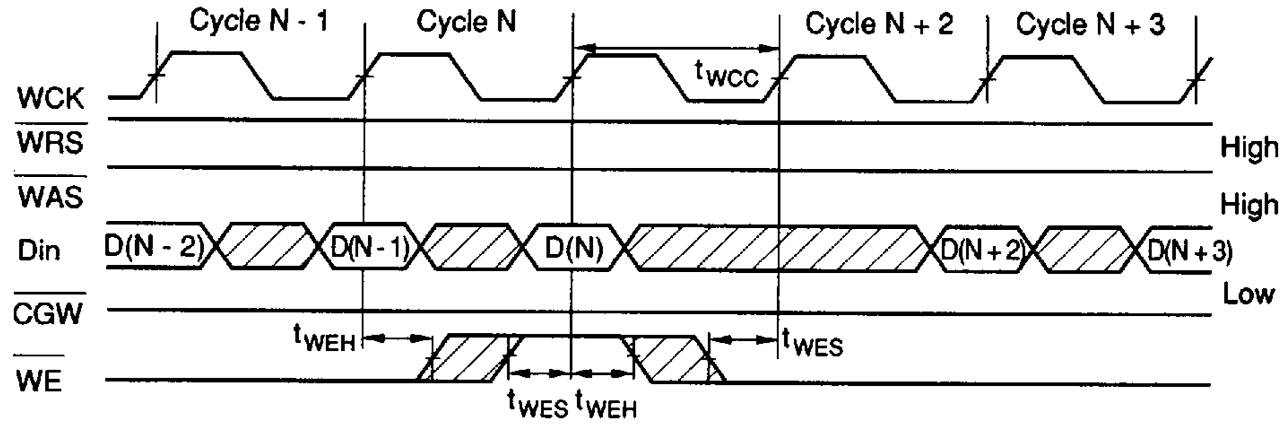
Note: The write address pointer is reset to 0 starting with the WRS low cycle. Only the falling edge of the WRS signal is detected. Adequate margin is provided if the rise occurs at least one clock cycle before the next fall.

- **Write clock gate**



Note: During cycles where CGW is high, the write address pointer is not incremented, and the D_{IN} data is not written.

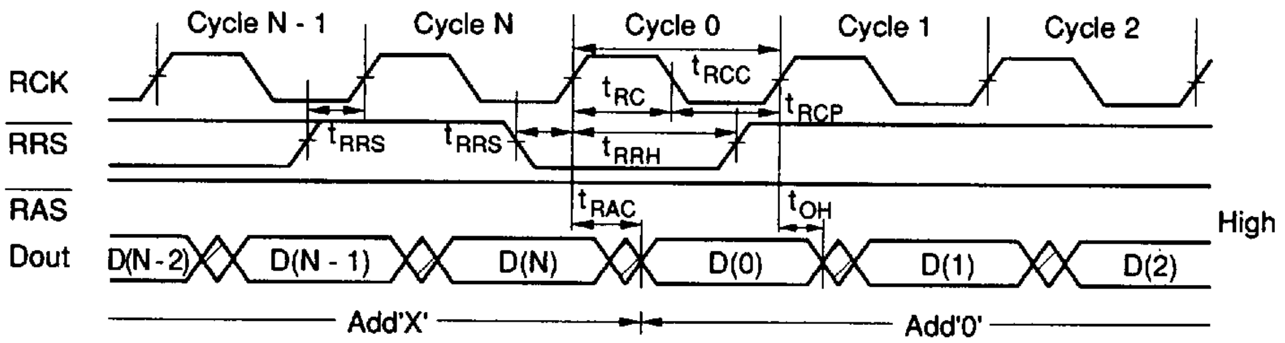
- **Write enable**



Note: Although the write address pointer is incremented on a cycle where $\overline{WE}$ is high, the D_{IN} data is not written, and the previous memory data is retained.

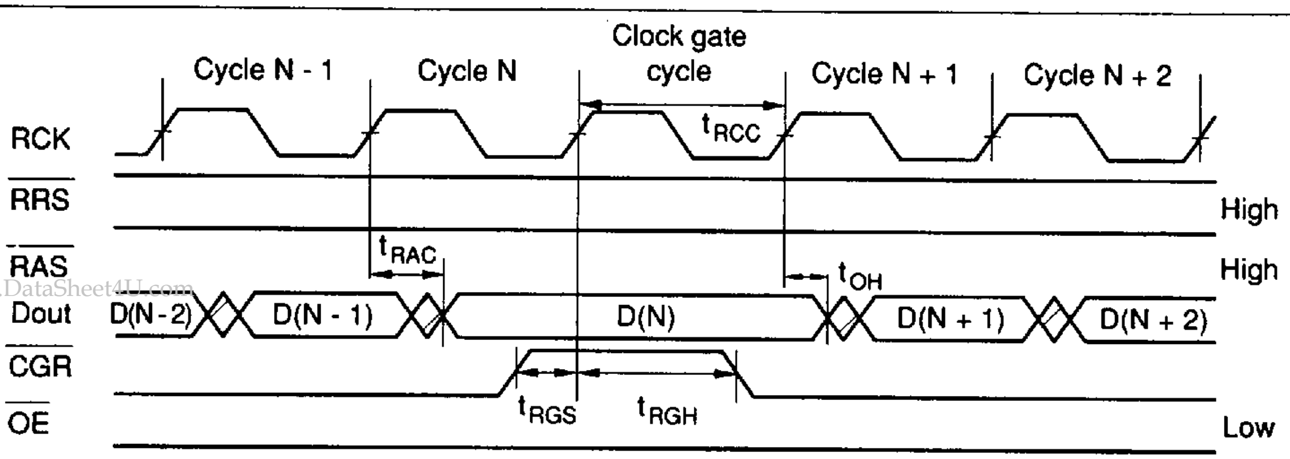
Read Cycle

- **Read address reset**



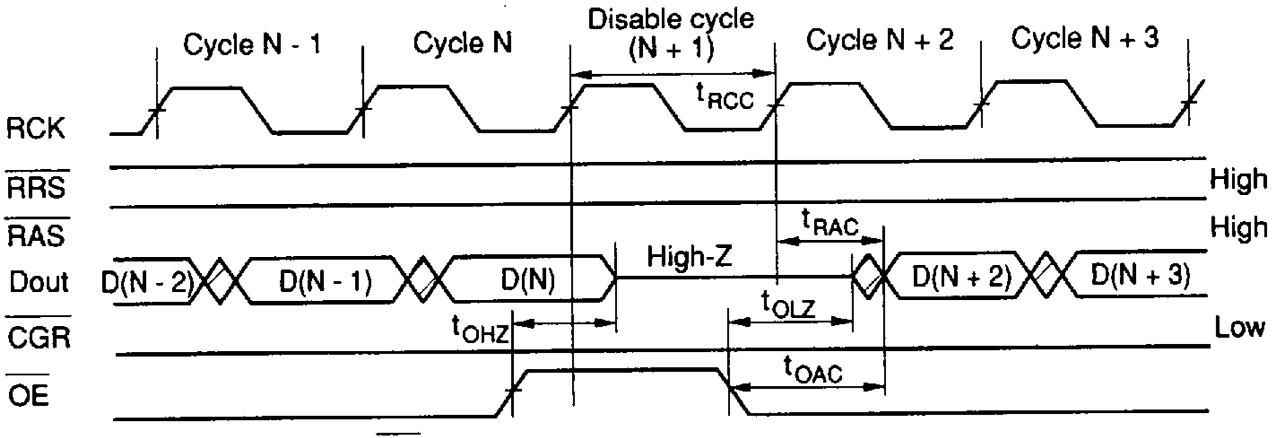
Note: The read address pointer is reset to 0 from the cycle where $\overline{RRS}$ was low. Only the falling edge of the $\overline{RRS}$ signal is detected. Adequate margin is provided if the rise occurs at least one clock cycle before the next fall.

- **Read clock gate**



Note: During cycles where CGR is high, the read address pointer is not incremented, and the output data is retained.

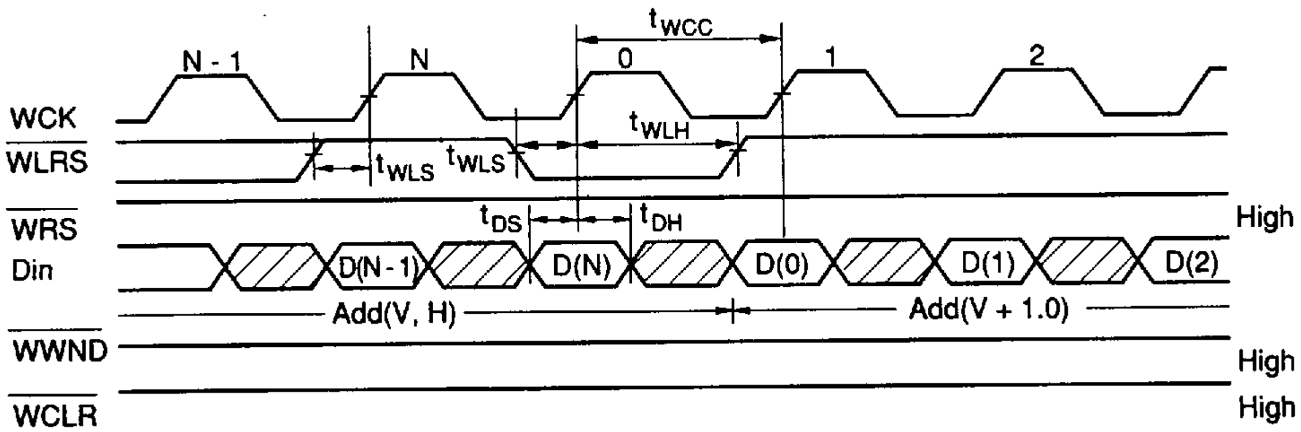
- **Output enable**



Note: During cycles where $\overline{OE}$ is high, the output goes to the high impedance state, and the read address pointer is incremented.

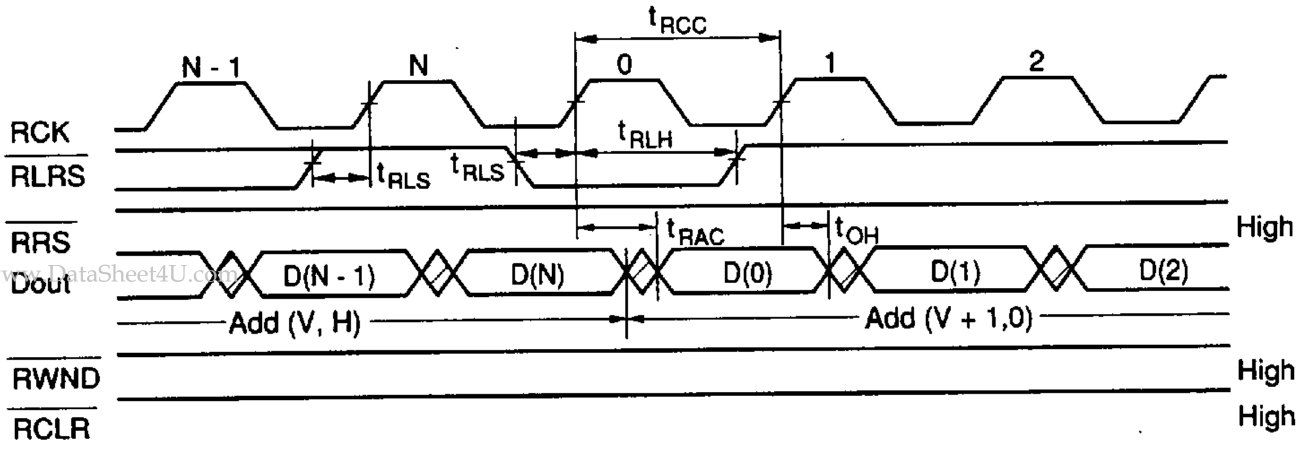
Line Reset

- **Write line increment**



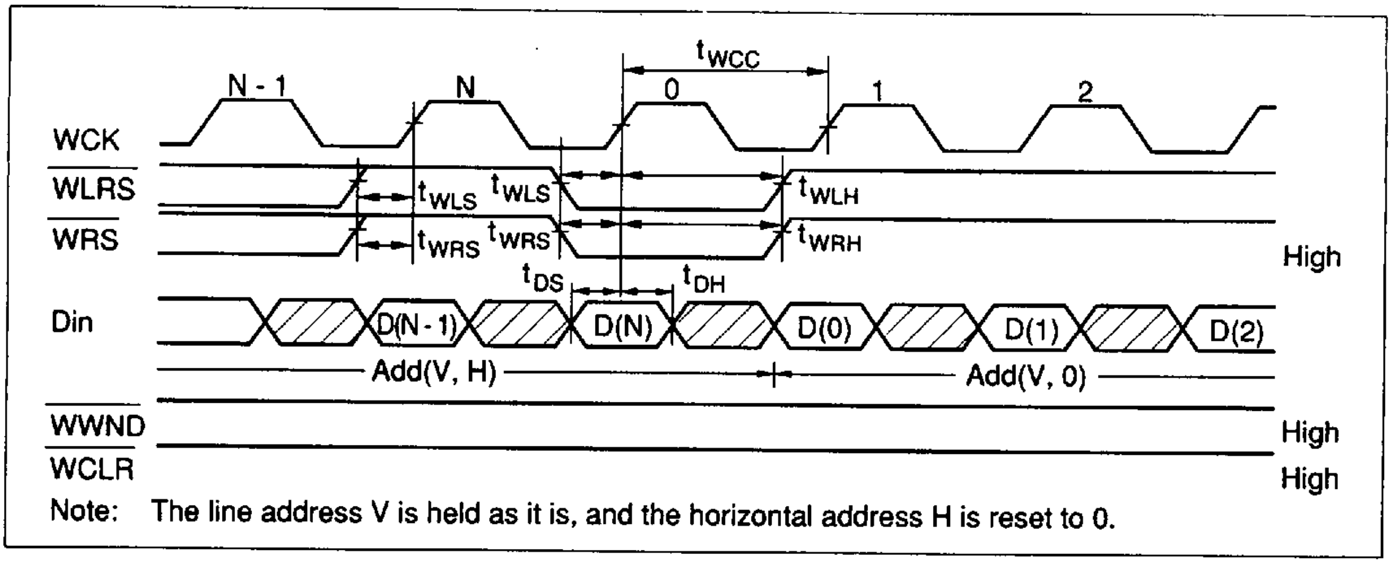
Note: The line address V is incremented, and the horizontal address H is reset to 0.

- **Read line increment**



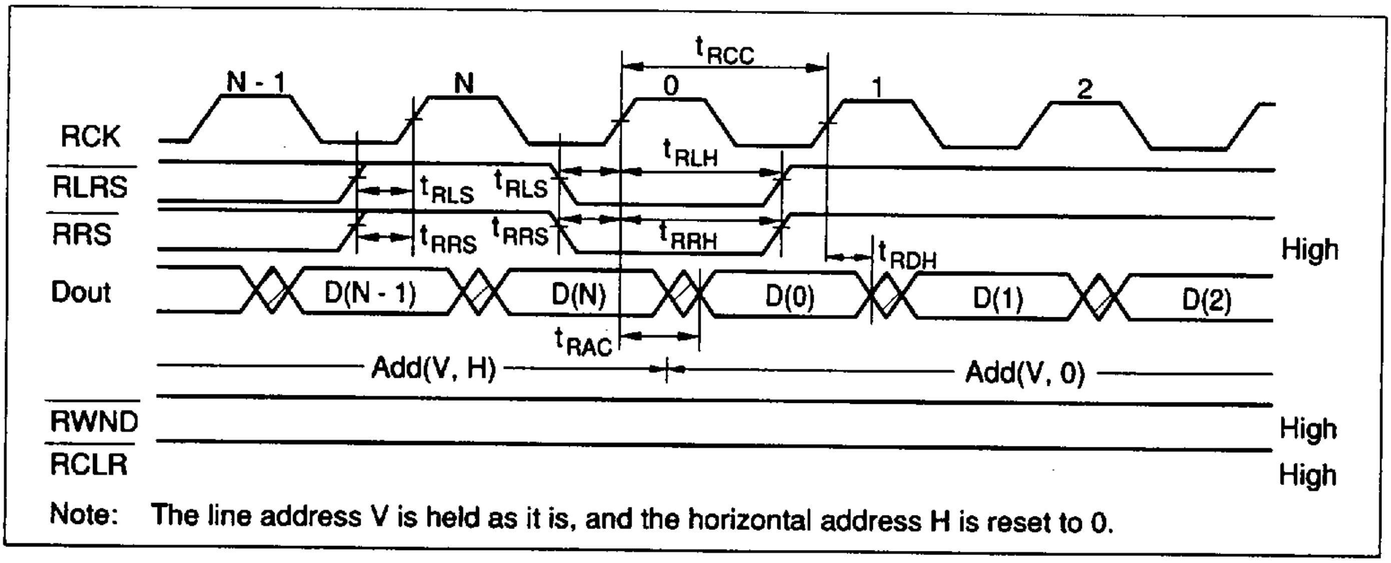
Note: The line address V is incremented, and the horizontal address H is reset to 0.

- **Write line hold**



Note: The line address V is held as it is, and the horizontal address H is reset to 0.

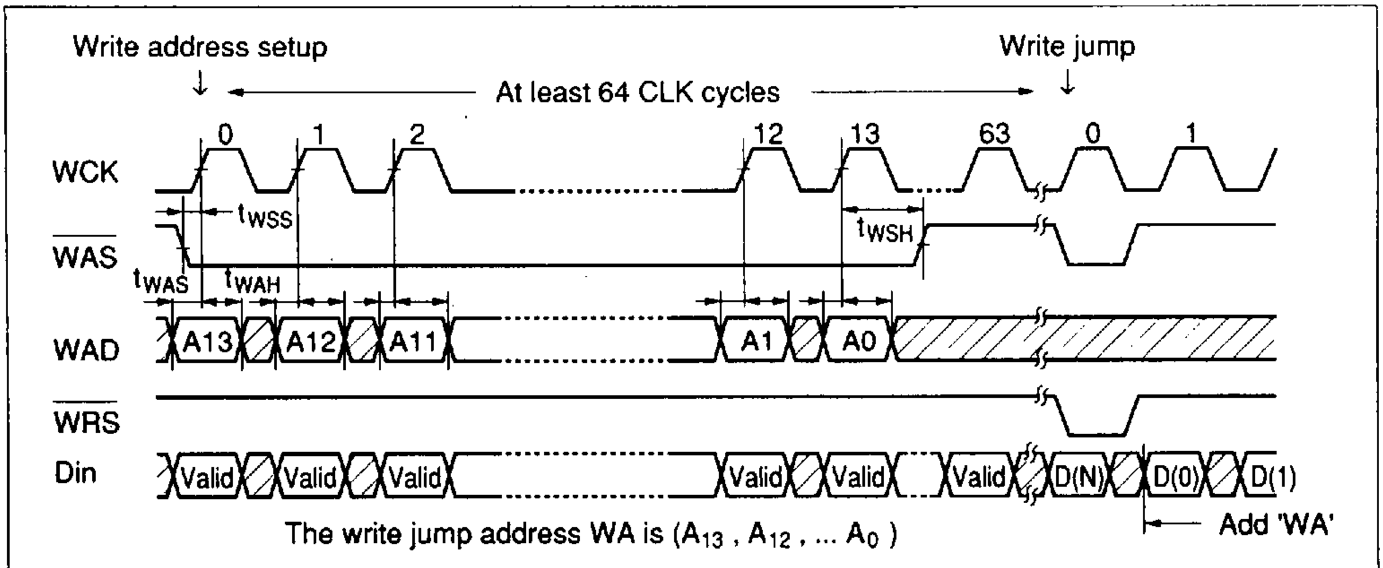
- **Read line hold**



Note: The line address V is held as it is, and the horizontal address H is reset to 0.

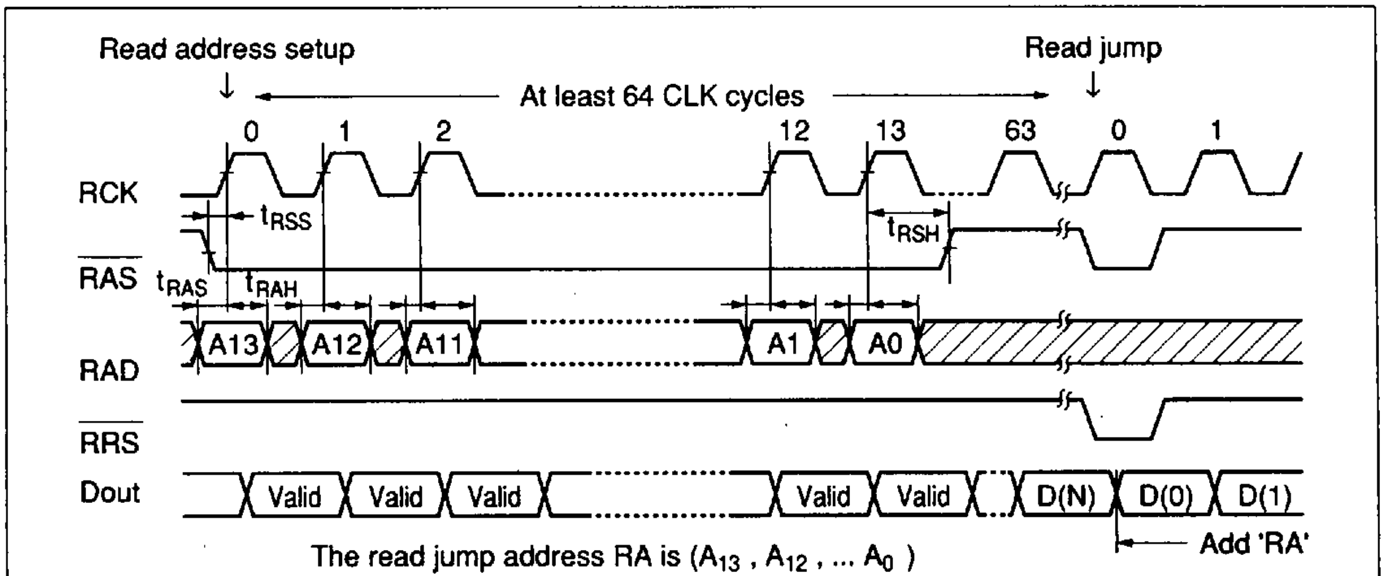
Jump Address Setup (1 Dimensional Addressing Mode)

- **Write address setup**



Note: After 64 cycles have passed following the start of write address setup, a jump to the set address can be performed at any time.

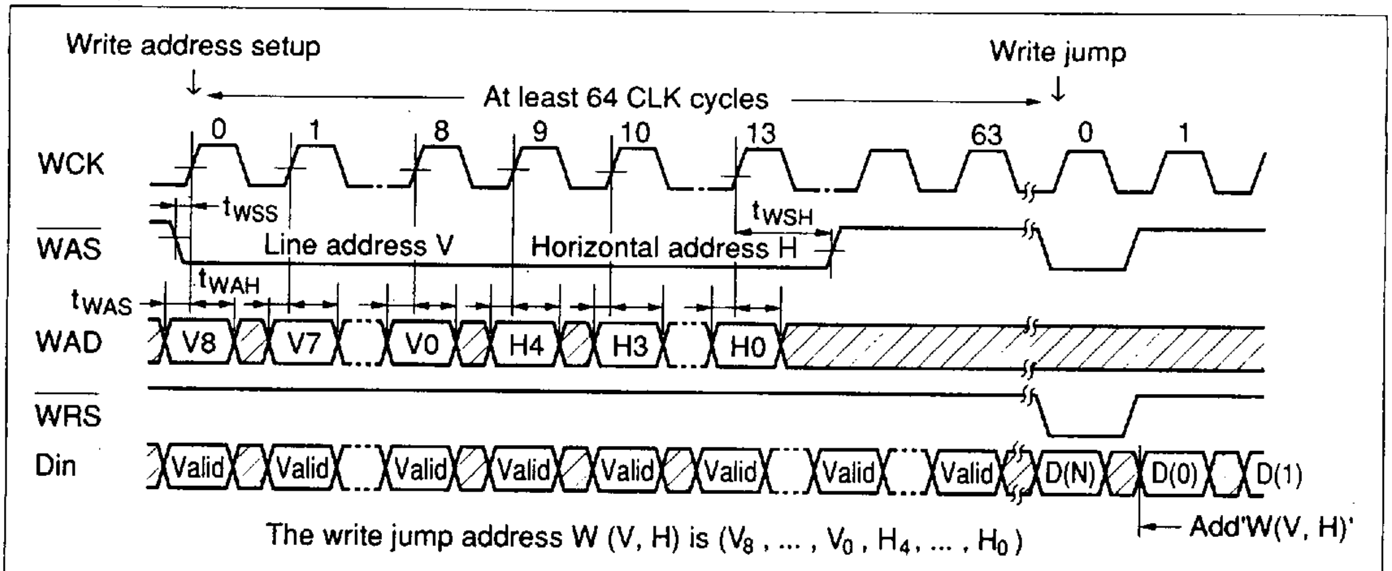
- **Read address setup**



Note: After 64 cycles have passed following the start of read address setup, a jump to the set address can be performed at any time.
Read and write address setup can be performed asynchronously.

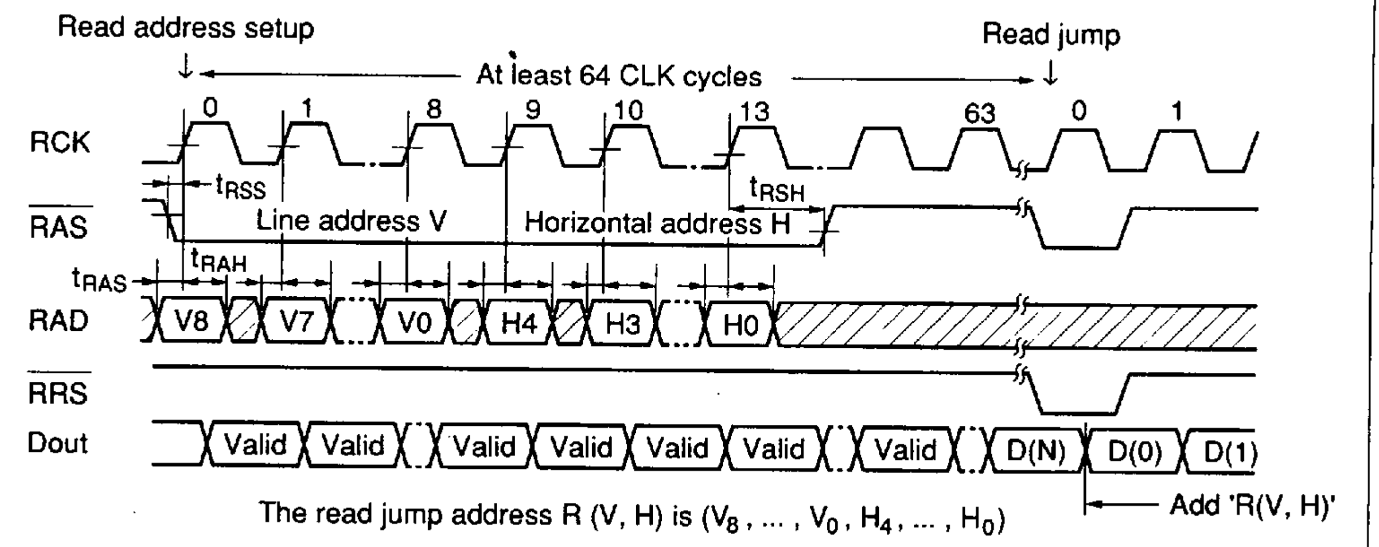
Jump Address Setup (2 Dimensional Addressing Mode 1)

- **Write address setup (2 dimensional addressing: 324 line × 1024 dot mode)**



Note: The jump to the set address can be performed at any time once the required 64 cycles have passed following the start of write address setup.

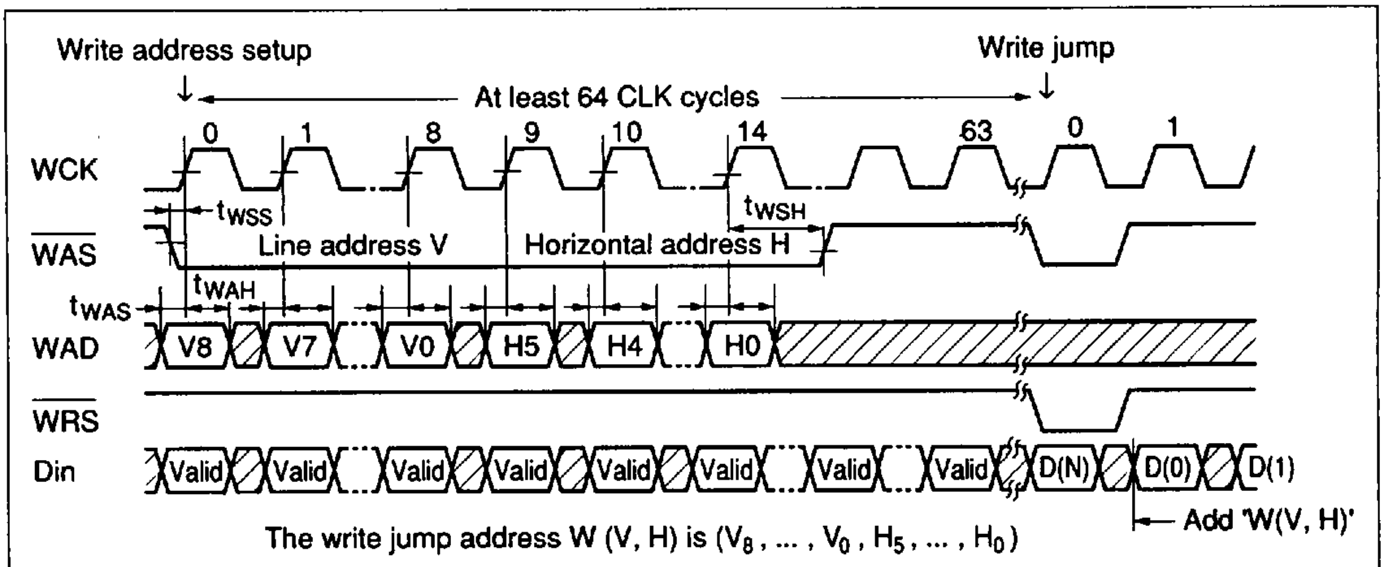
- **Read address setup (2 dimensional addressing: 324 line × 1024 dot mode)**



Note: The jump to the set address can be performed at any time once the required 64 cycles have passed following the start of read address setup.
Read and write address setup can be performed asynchronously.

Jump Address Setup (2 Dimensional Addressing Mode 2)

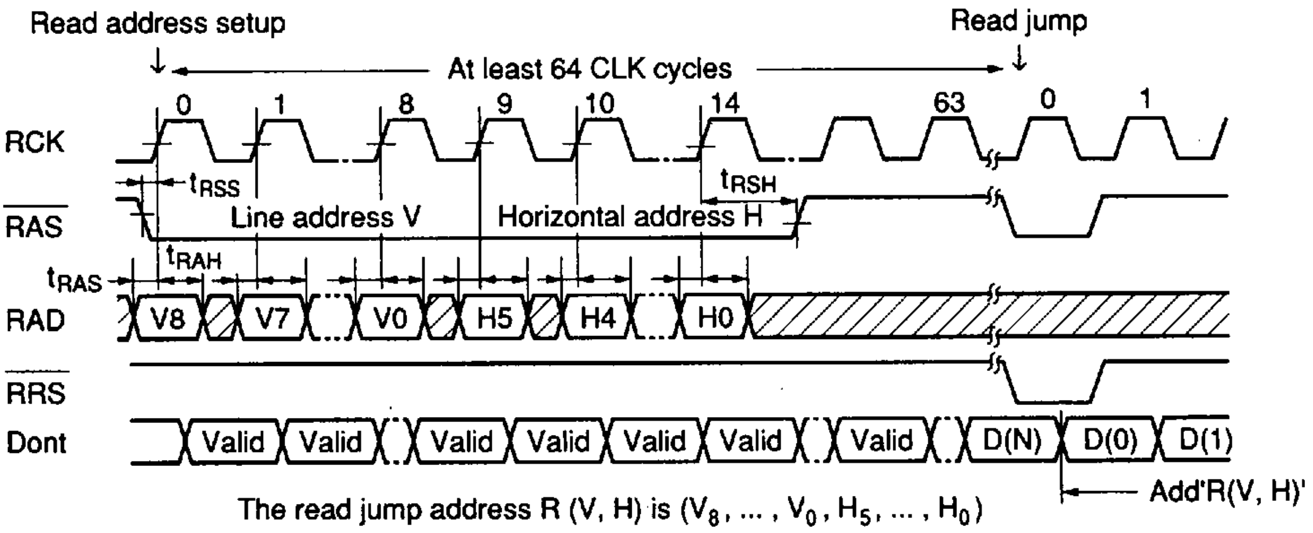
- **Write address setup (2 dimensional addressing: 288 line × 1152 dot mode)**



The write jump address W (V, H) is (V_8, ... , V_0, H_5, ... , H_0)

Note: The jump to the set address can be performed at any time once the required 64 cycles have passed following the start of write address setup.

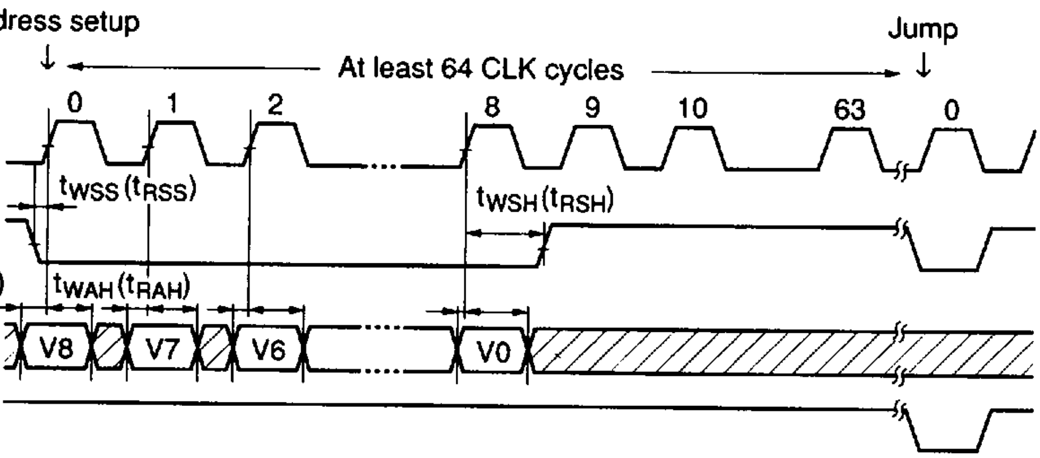
- **Read address setup (2 dimensional addressing: 288 line × 1152 dot mode)**



The read jump address R (V, H) is (V_8, ... , V_0, H_5, ... , H_0)

Note: The jump to the set address can be performed at any time once the required 64 cycles have passed following the start of read address setup.
Read and write address setup can be performed asynchronously.

- **Address input mask**

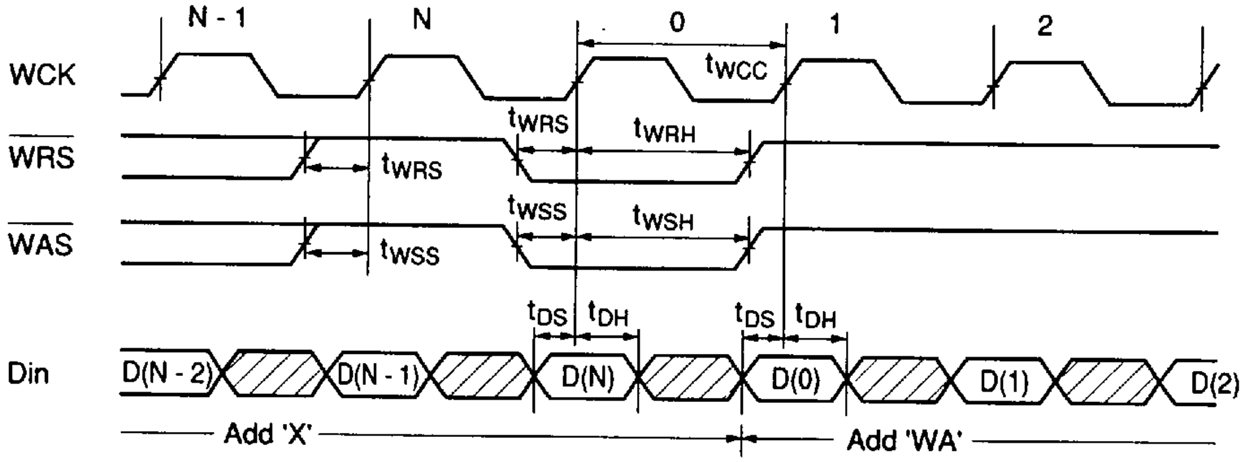


In this example, only the line address is re-input, and the horizontal address retains its previously set value.

Note: After the start of read or write jump address setup, if $\overline{\text{RAS}}$ or $\overline{\text{WAS}}$ respectively is returned to the high level at an arbitrary bit position, the address bits input thereafter are masked, and the corresponding bits retain their previous values.

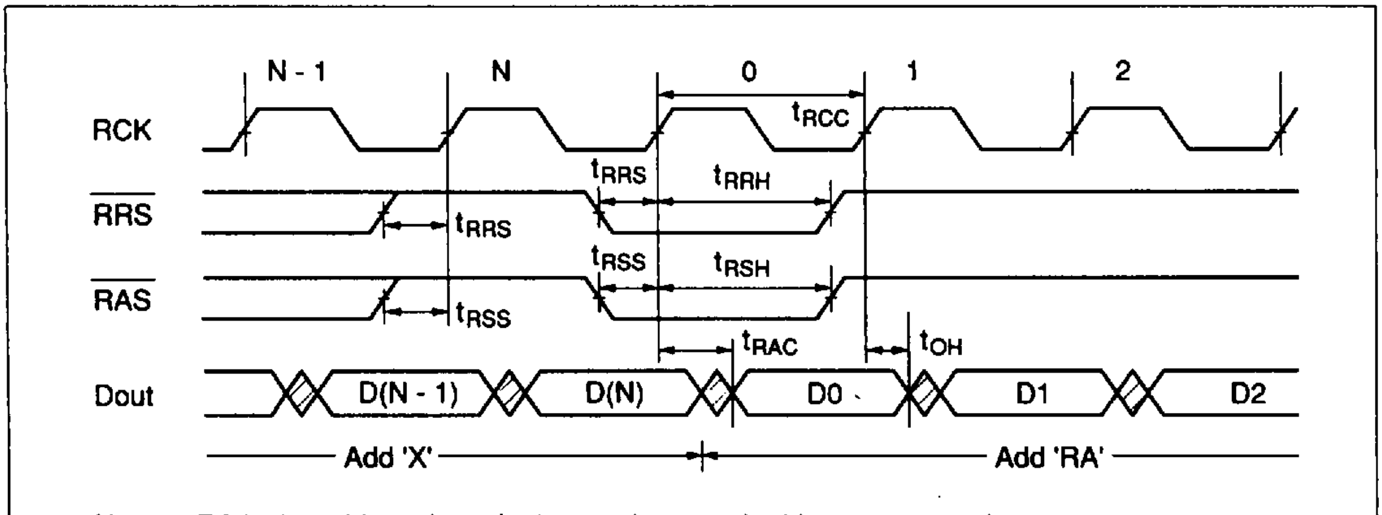
Jump

- **Write jump**



Note: WA is the address input in the previous write address setup cycle.

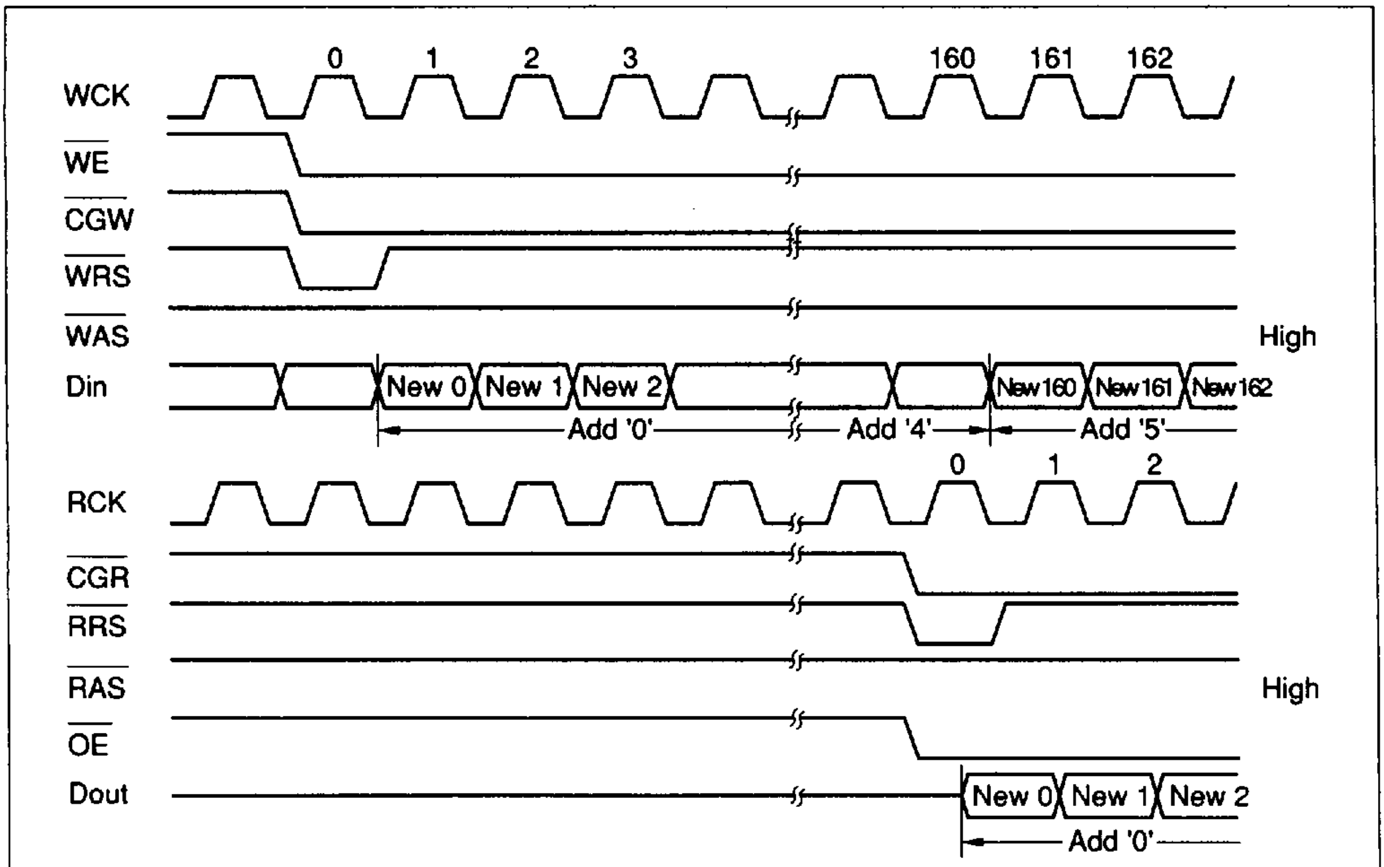
- **Read jump**



Note: RA is the address input in the previous read address setup cycle.

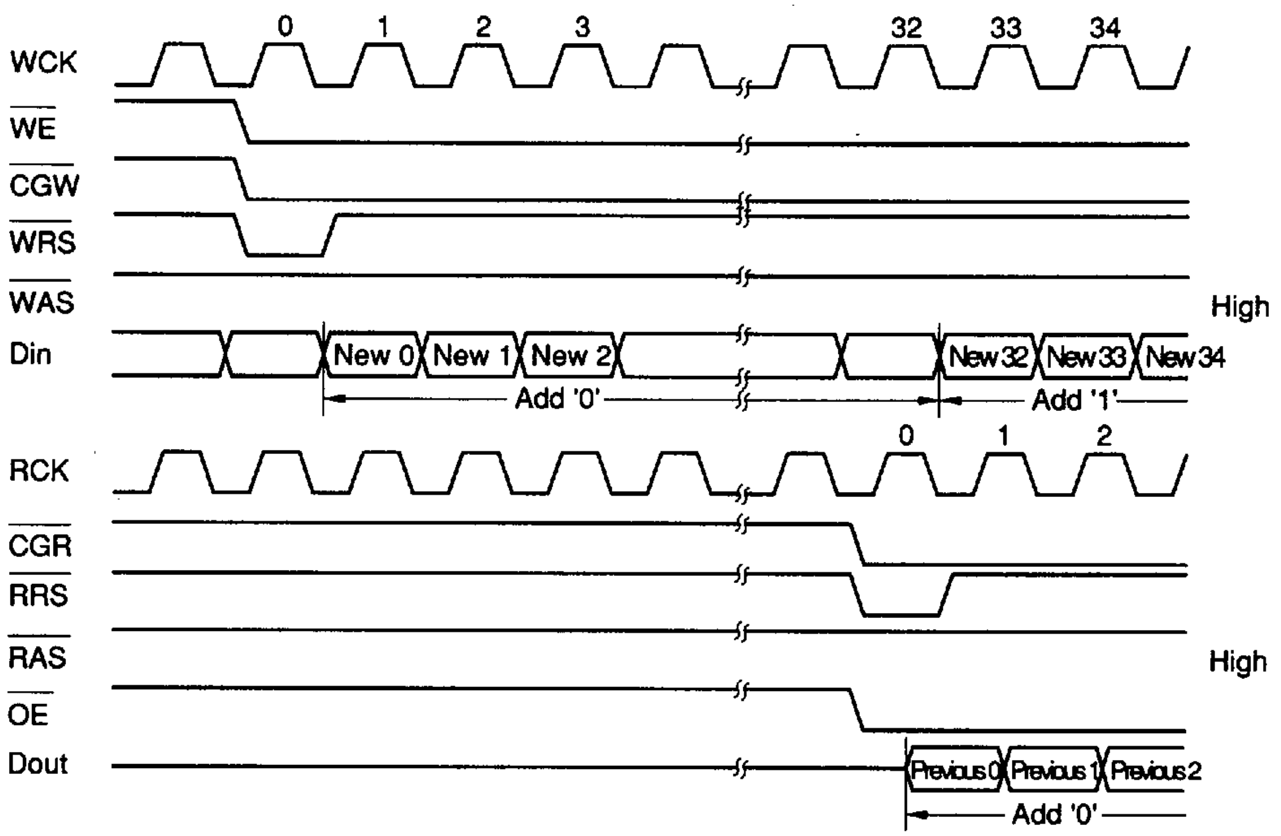
New/Previous Data Access

- **New data access (address reset)**



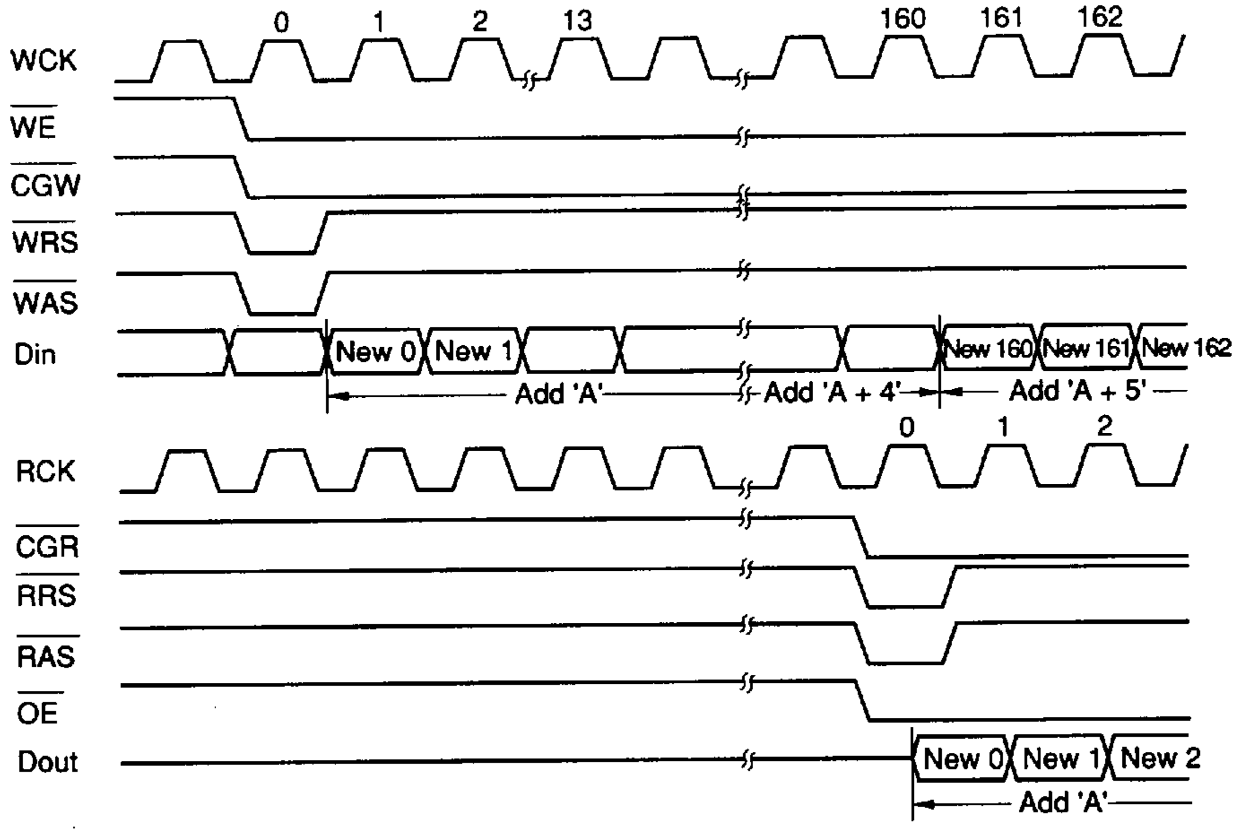
Note: Written data can be read out 160 WCK clock cycles after it was written.

- **Previous data access (address reset)**



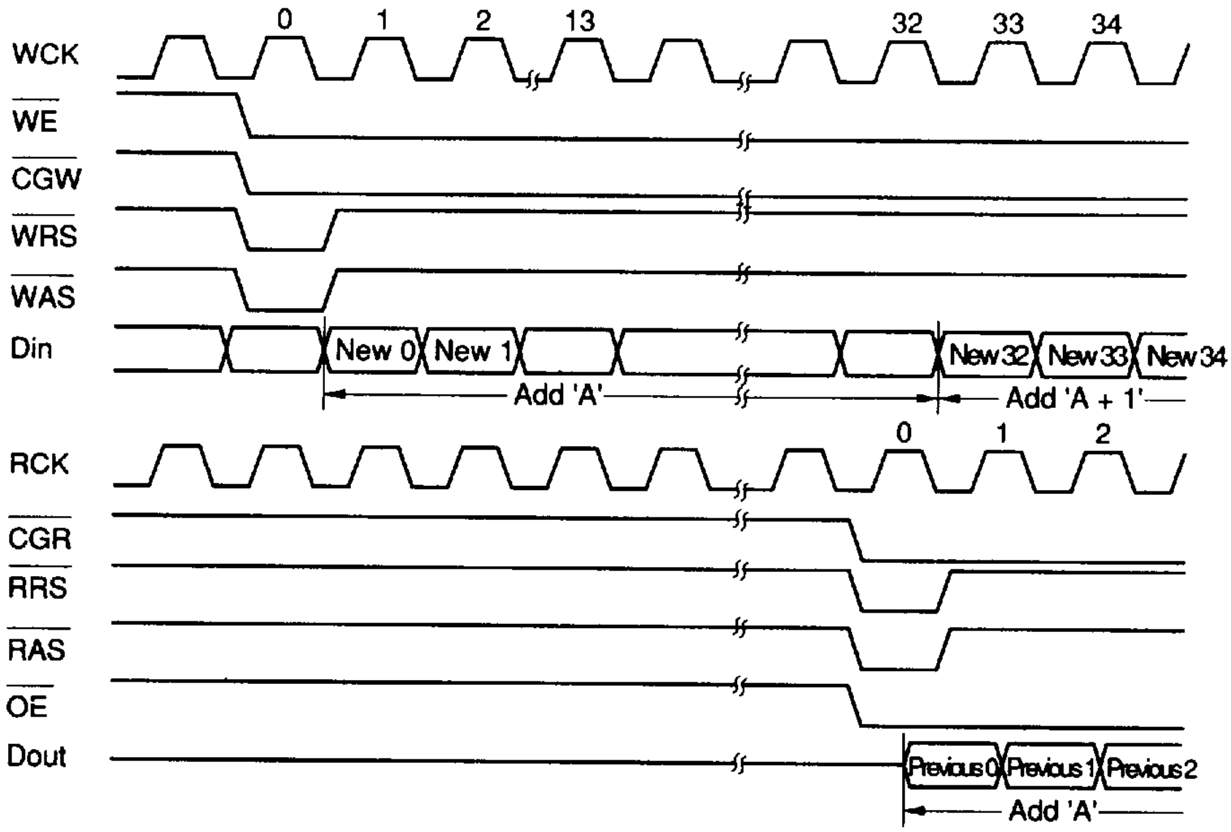
Note: Previous data can be read out up to 32 WCK clock cycles after the write operation.

- **New data access (address jump)**
  (example where the read and write jump addresses are to the same location)



Note: Written data can be read out 160 WCK clock cycles after it was written. However, it is necessary to execute the read jump address setup operation outside the time period between 32 WCK cycles before the start of write to that address and 32 WCK cycles after the completion of write to that address.

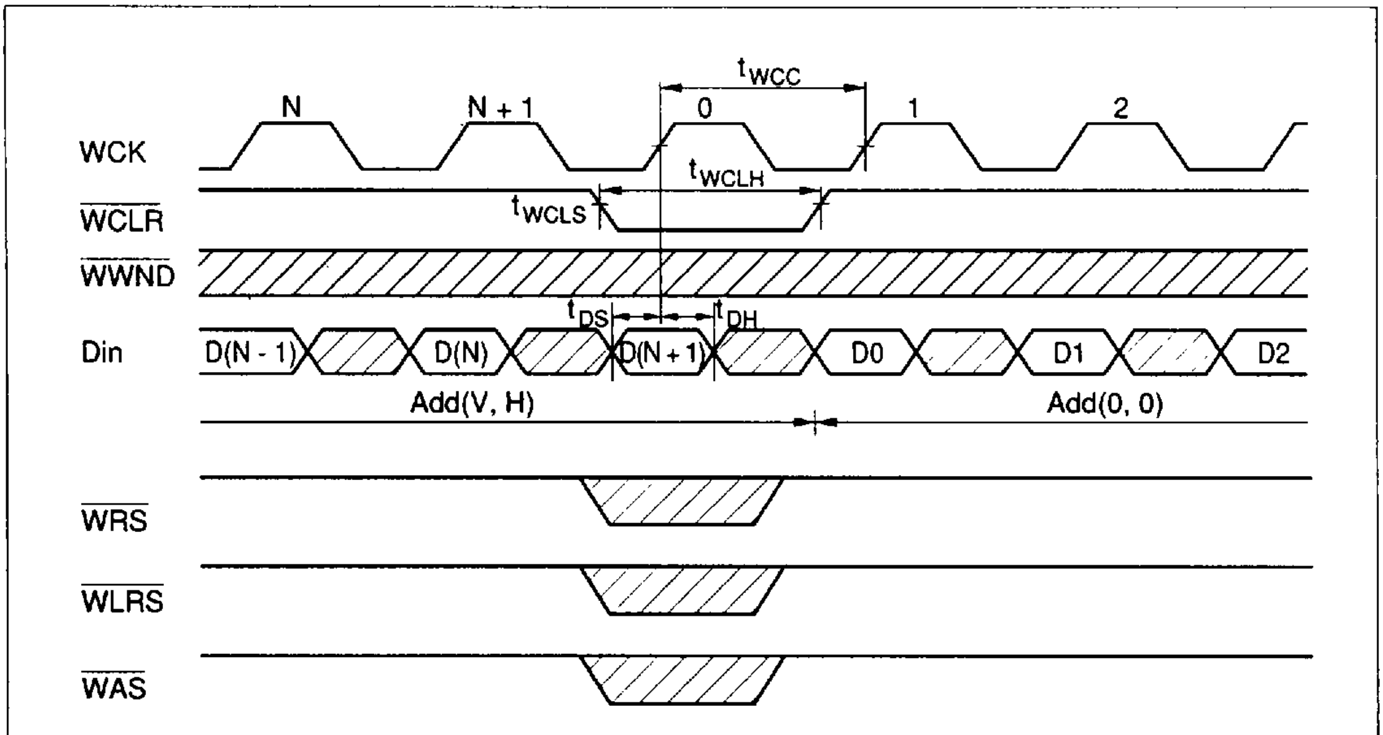
- **Previous data access (address jump)**
  (example when the read and write jump addresses are to the same location)



Note: Previous data can be read out up to 32 WCK clock cycles after the write operation.

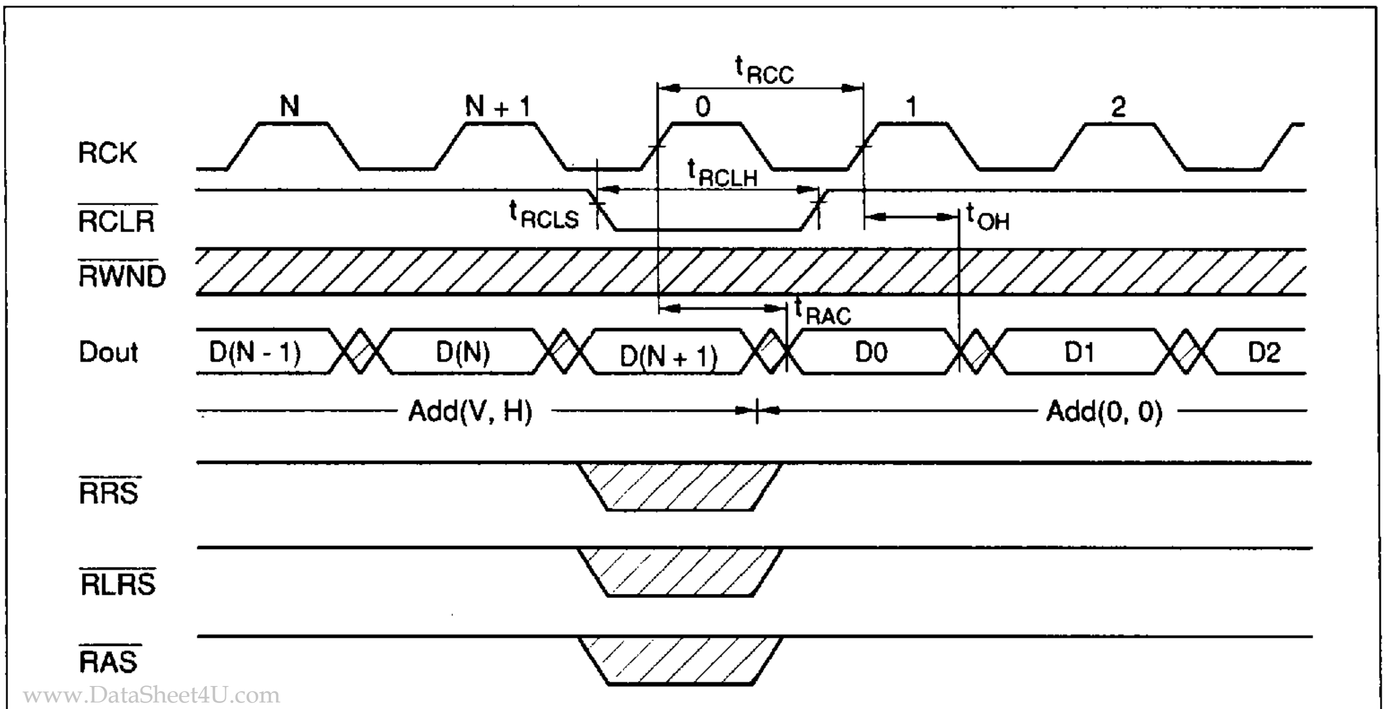
Clear

- **Write clear**



Note: The write address pointer is reset to (0, 0), and window mode is turned off if it was on.

- **Read clear**



Note: The read address pointer is reset to (0, 0), and window mode is turned off if it was on.

Window Scan Function

Combined Window Scan Example

In window scan mode, the destination address of a jump will be the first point in the window region, and line reset and reset operate as follows.

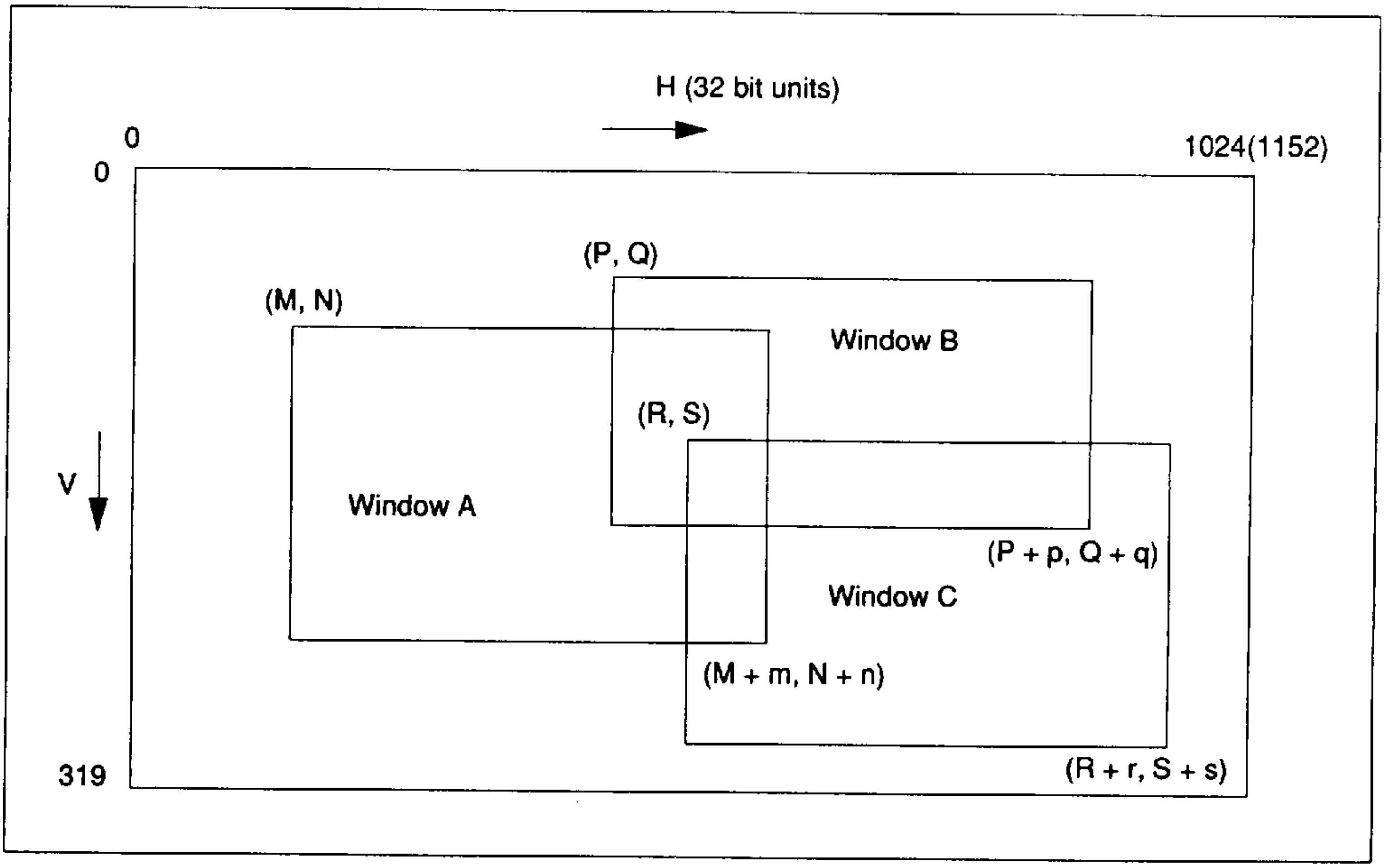
Line reset: Resets to the left edge of the window on the next line.

Reset: Resets to the first point in the window.

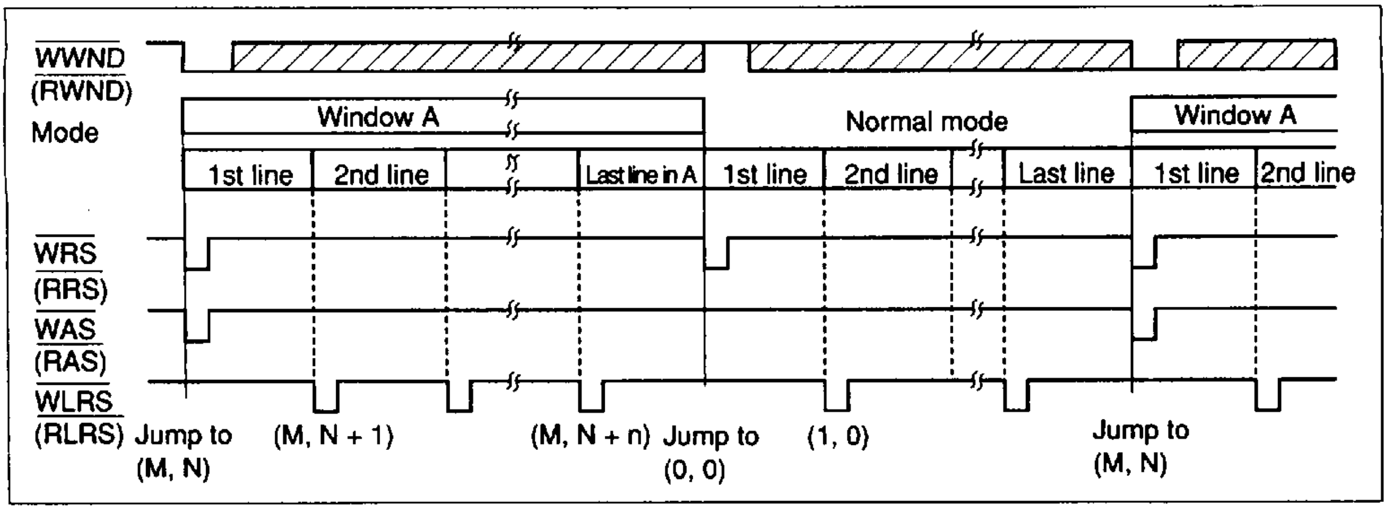
In this mode, addresses are generated automatically internally, so this function is useful in applications that need to scan a window region.

Also, completely independent window regions can be scanned by the read and write systems.

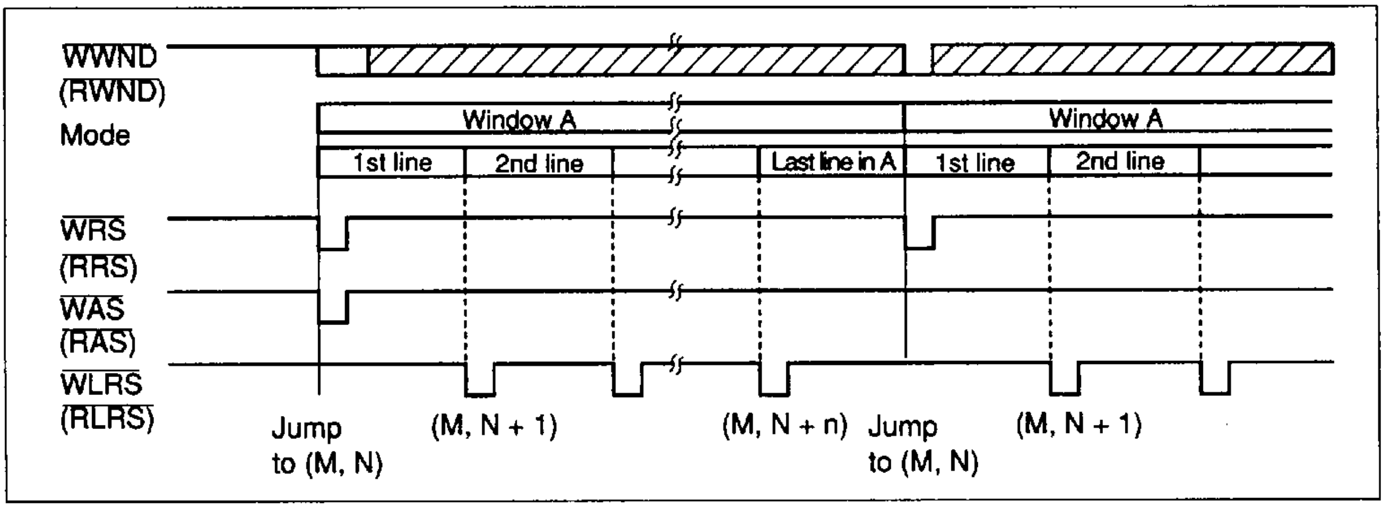
Representative application examples are presented below.



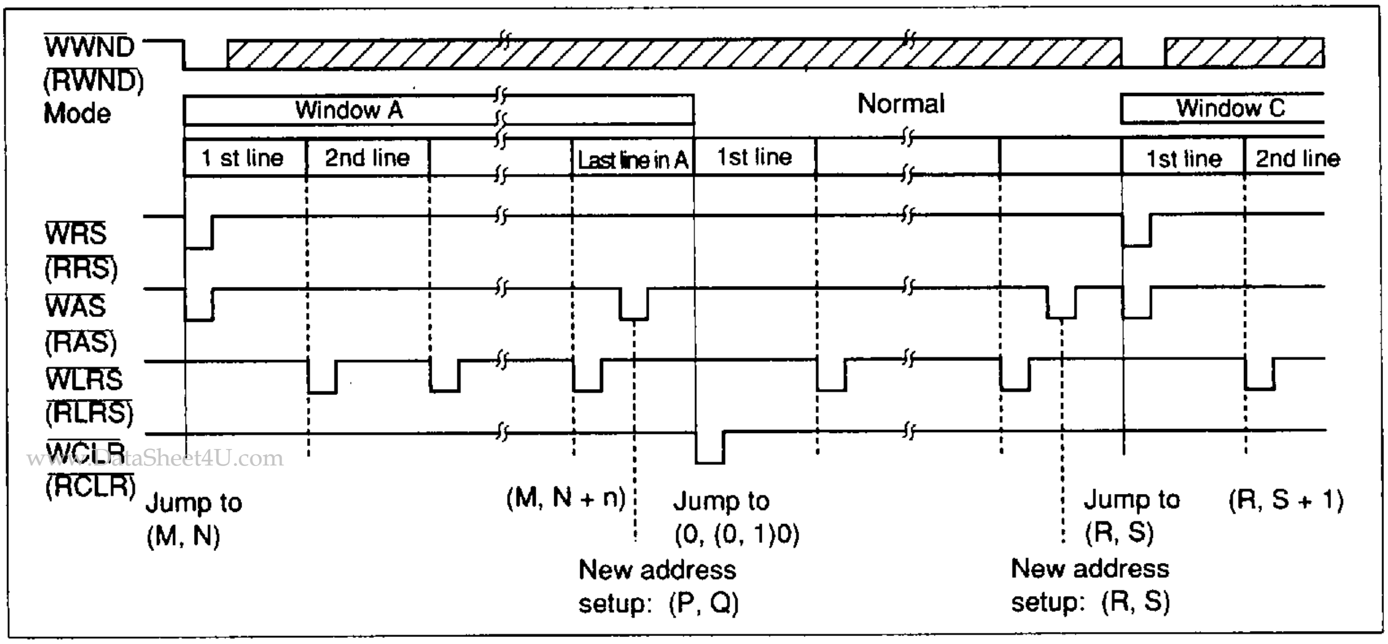
Case 1: Switching Between Normal and Window A Scan



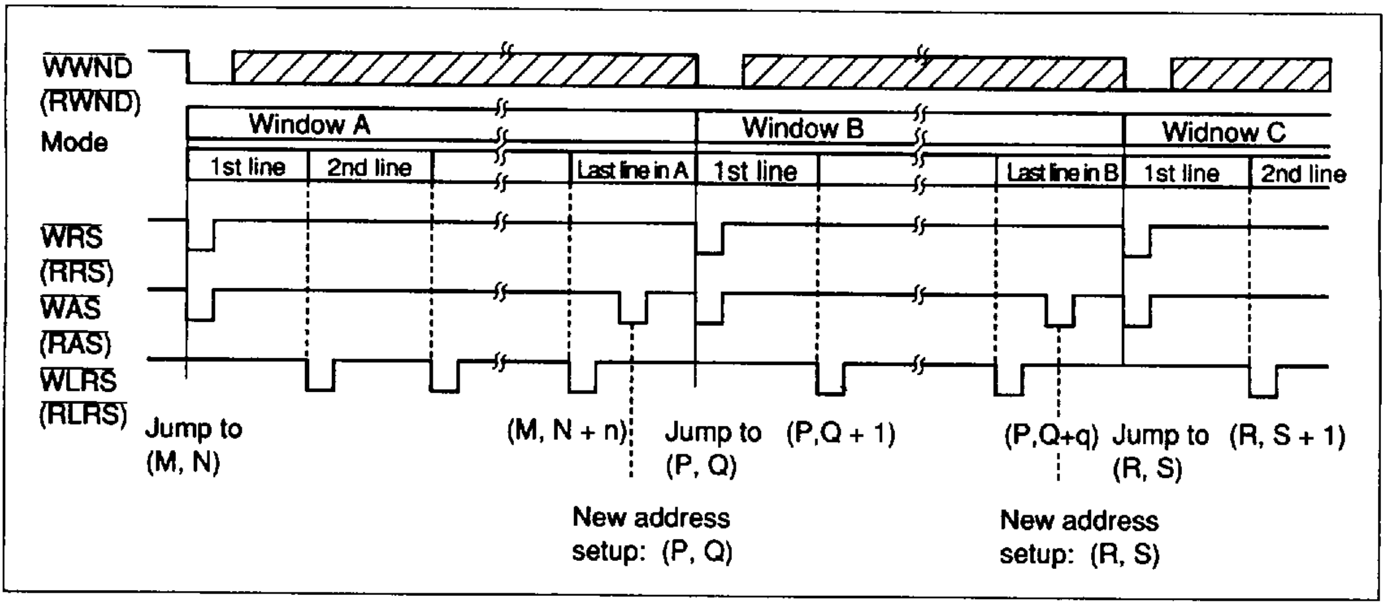
Case 2: Repeatedly Scanning Window A



Case 3: Switching from Window A Scan to Normal Scan to Window C Scan

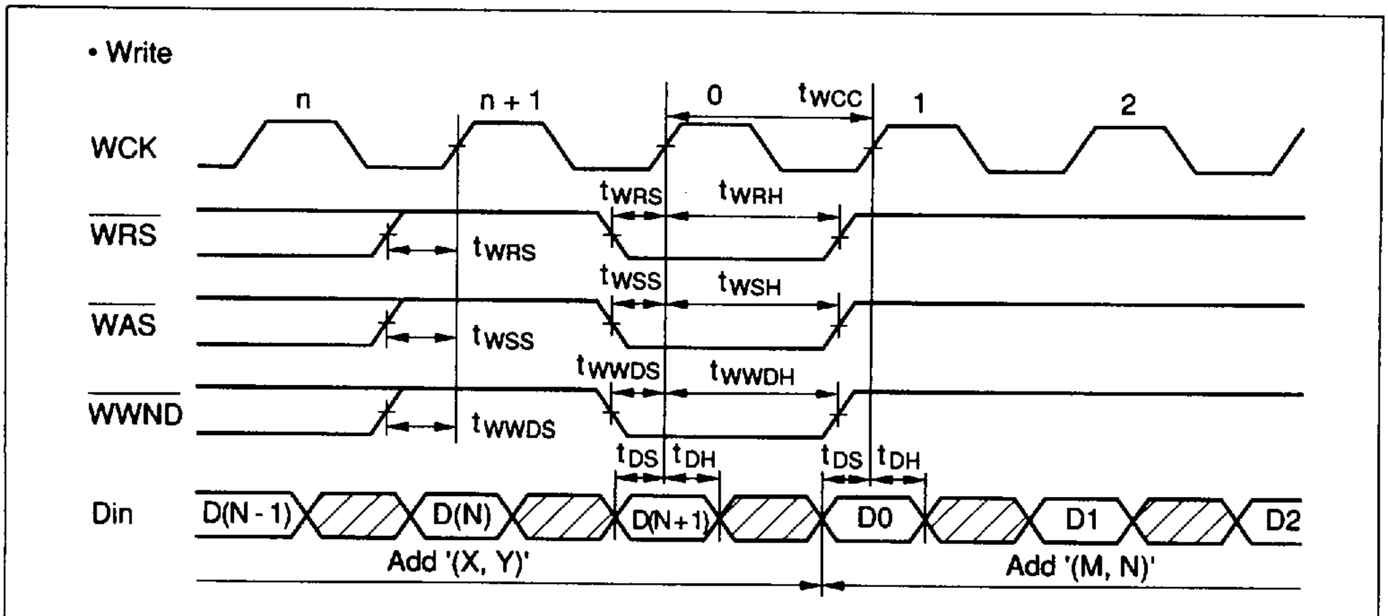


Case 4: Switching from Window A Scan to Window B Scan to Window C Scan



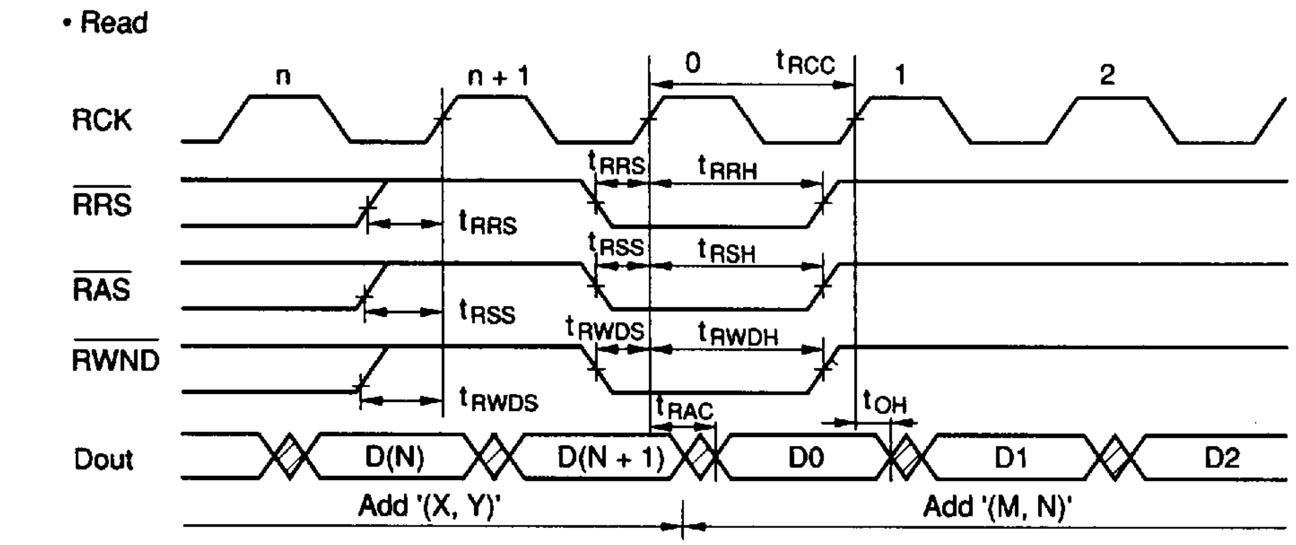
Window Scan Timing Charts

- **Window Jump (setup)**



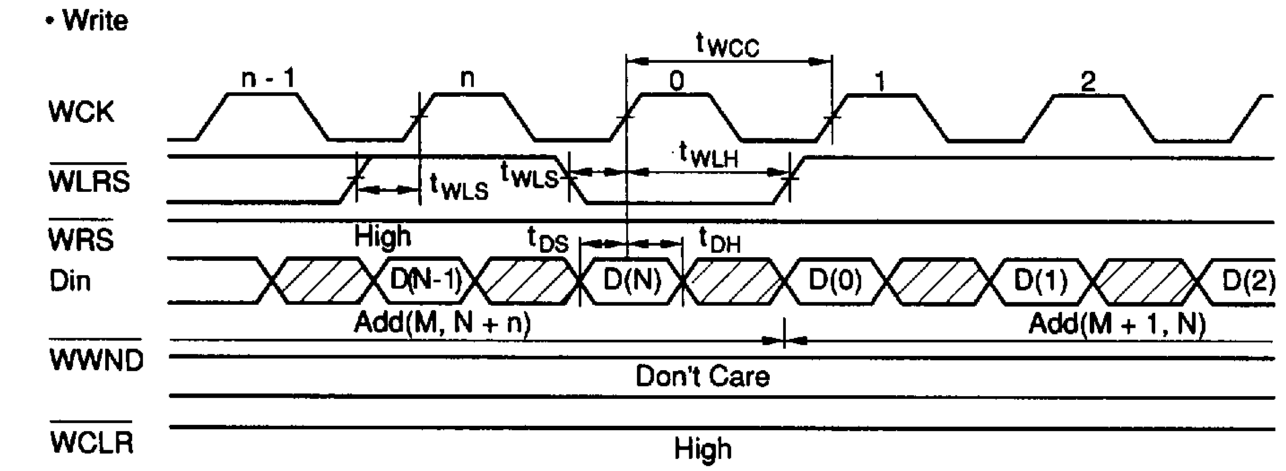
Note: The value (M, N) is the address input during the write address setup cycle.

- **Window Jump (setup)** (cont)



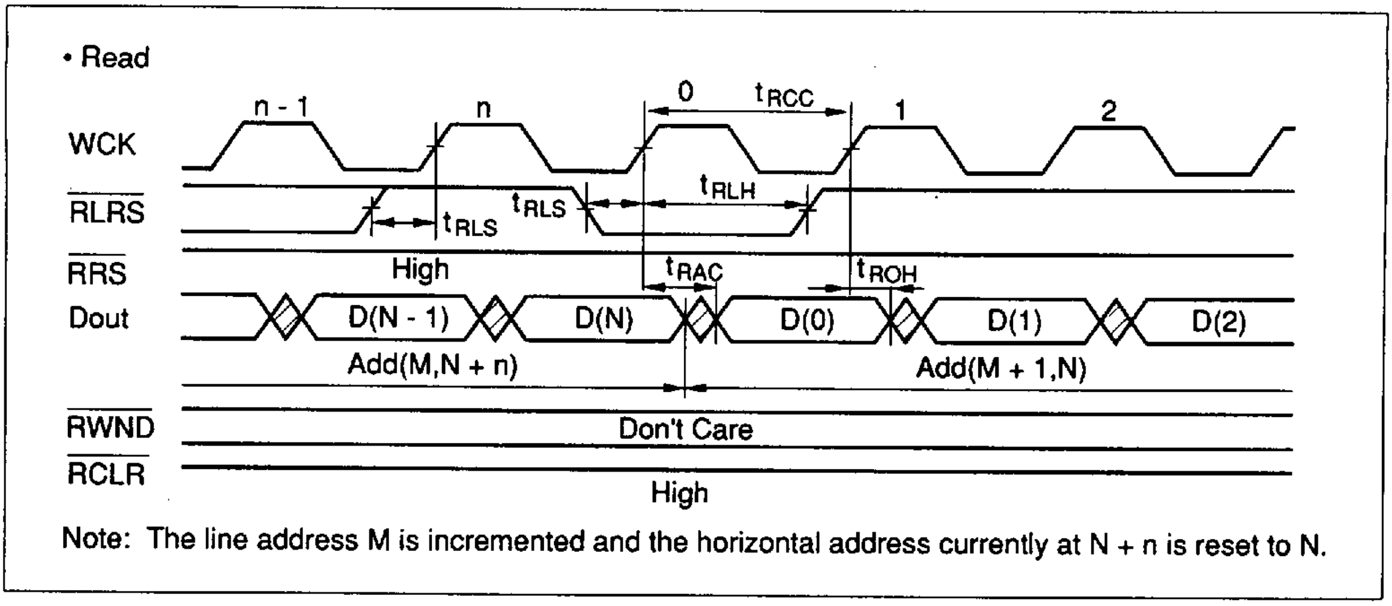
Note: The value (M, N) is the address input during the read address setup cycle.

- **Line Increment (in window mode)**



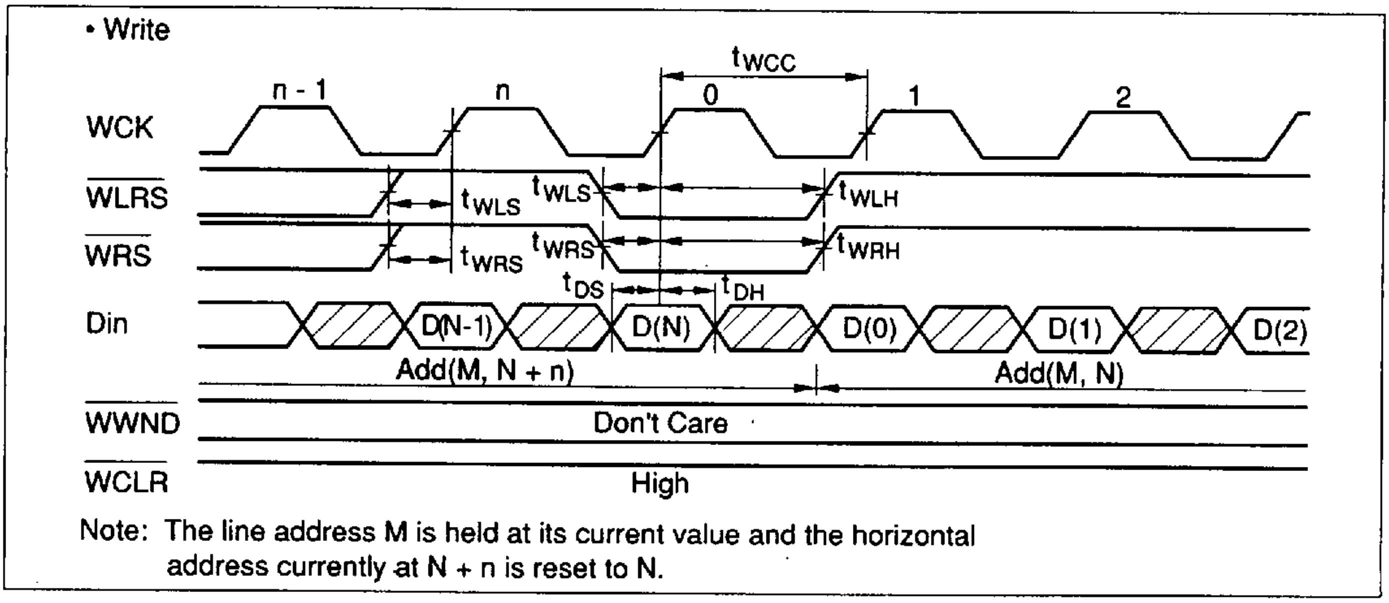
Note: The line address M is incremented and the horizontal address currently at N + n is reset to N.

- **Line Increment (in window mode)** (cont)

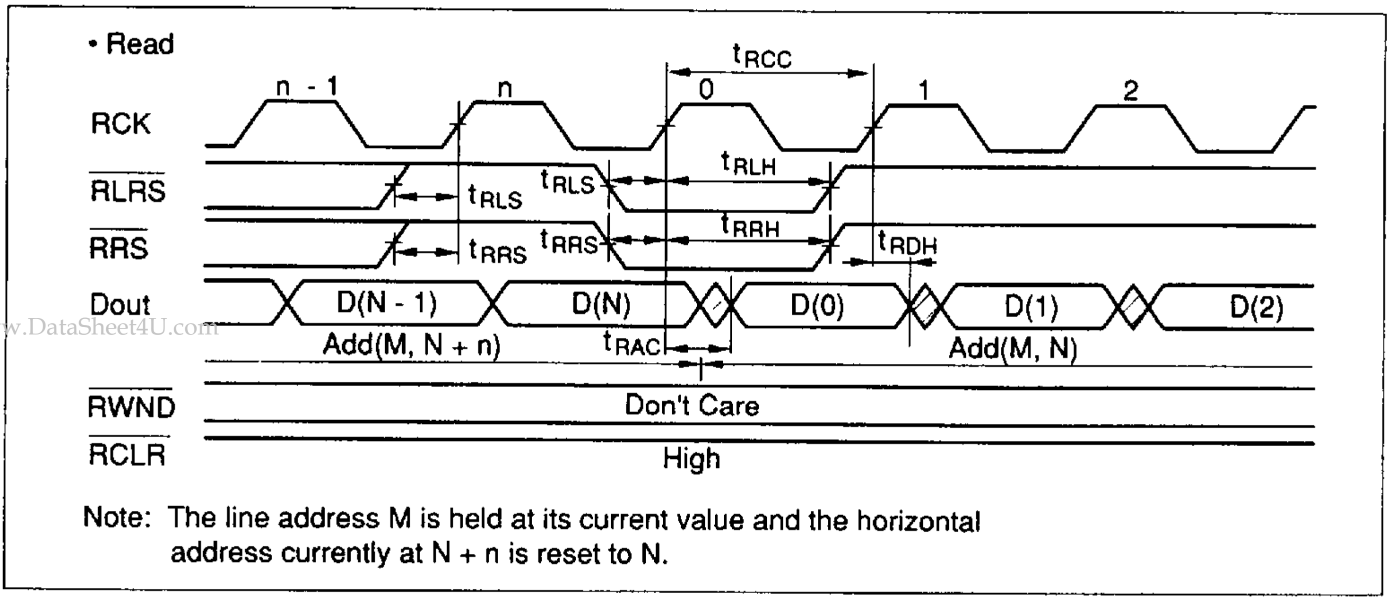


Note: The line address M is incremented and the horizontal address currently at N + n is reset to N.

- **Line Hold (in window mode)**



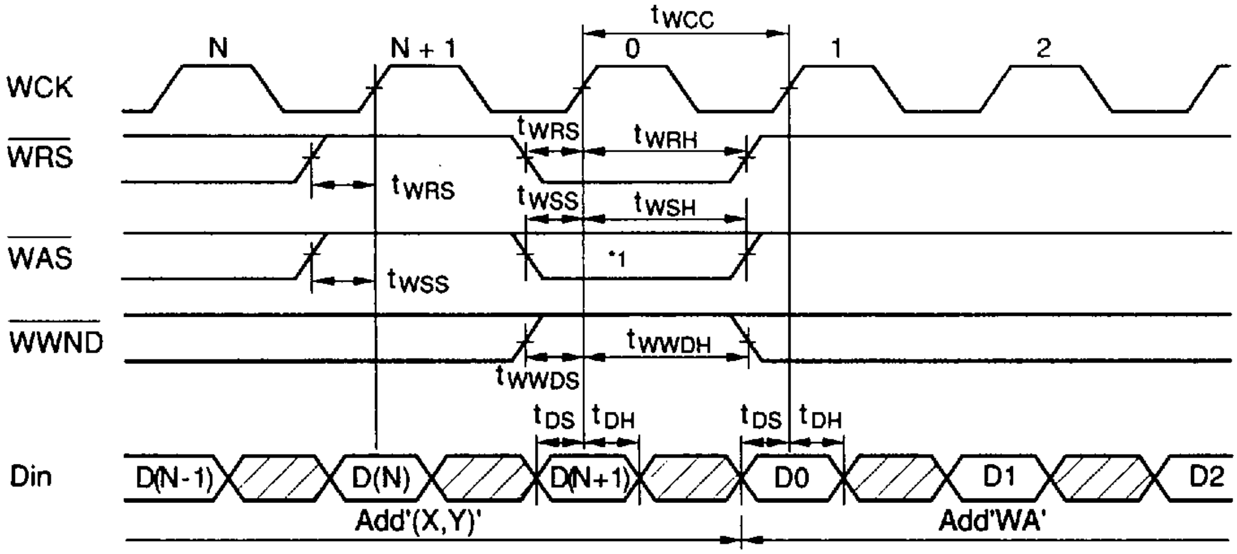
Note: The line address M is held at its current value and the horizontal address currently at N + n is reset to N.



Note: The line address M is held at its current value and the horizontal address currently at N + n is reset to N.

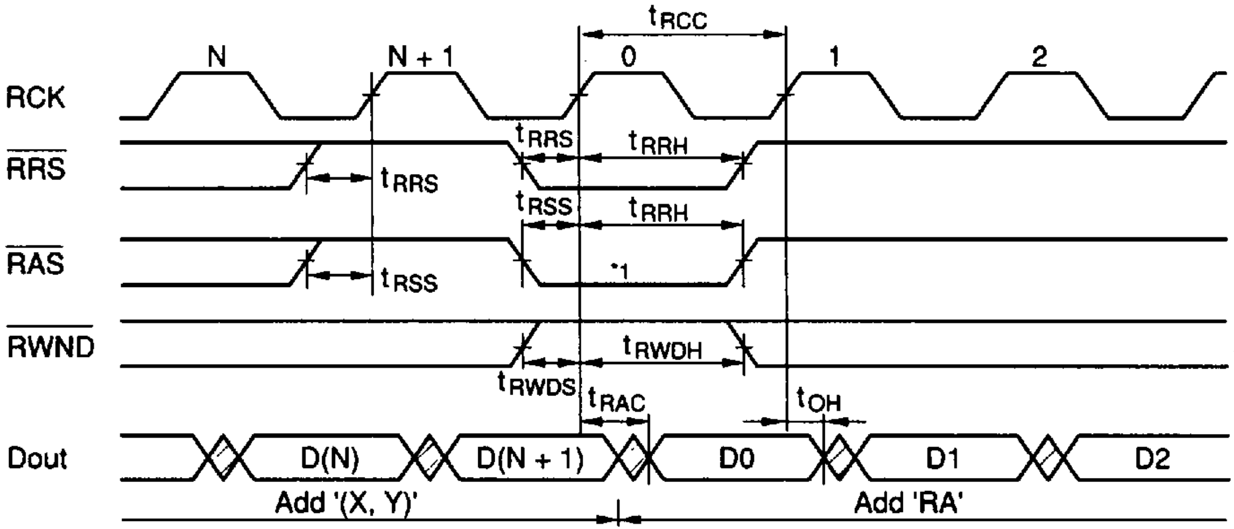
- **Window Clear**

• Write



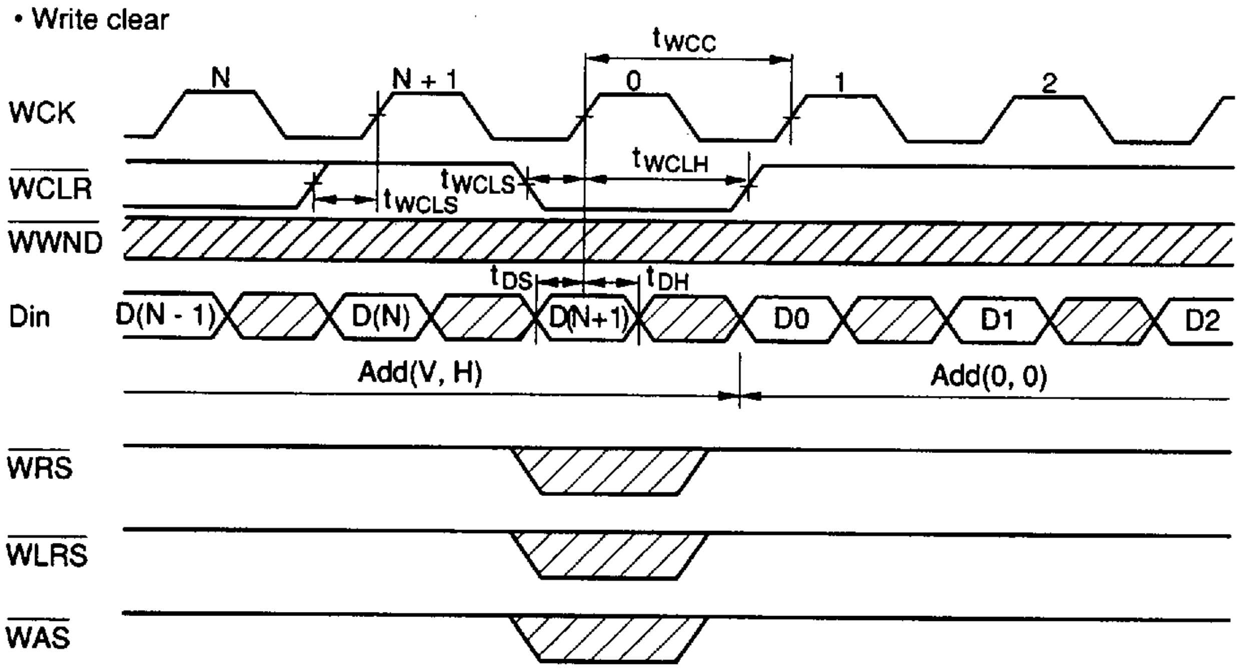
Note: 1. The write address is reset to (0, 0) when $\overline{\text{WAS}}$ is high.
When $\overline{\text{WAS}}$ is low, the write address jumps to WA, and in any case, the write window is cleared.

• Read

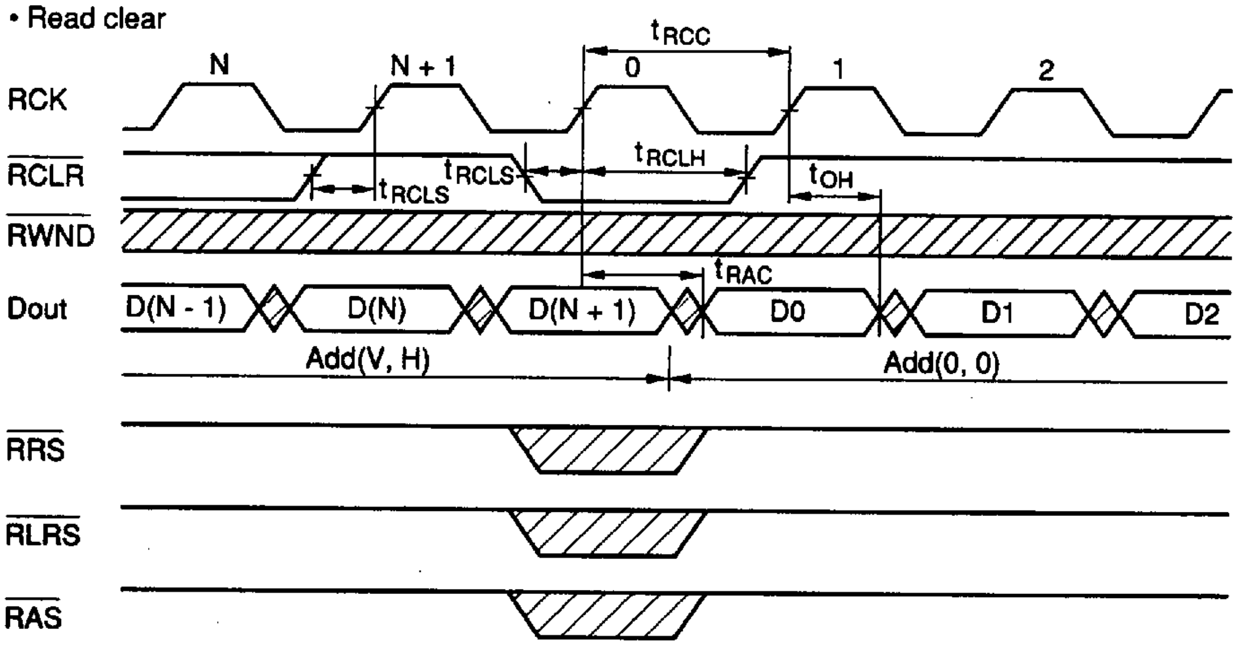


Note: 1. The read address is reset to (0, 0) when $\overline{\text{RAS}}$ is high.
When $\overline{\text{RAS}}$ is low, the read address jumps to RA, and in any case, the read window is cleared.

- **Clear**



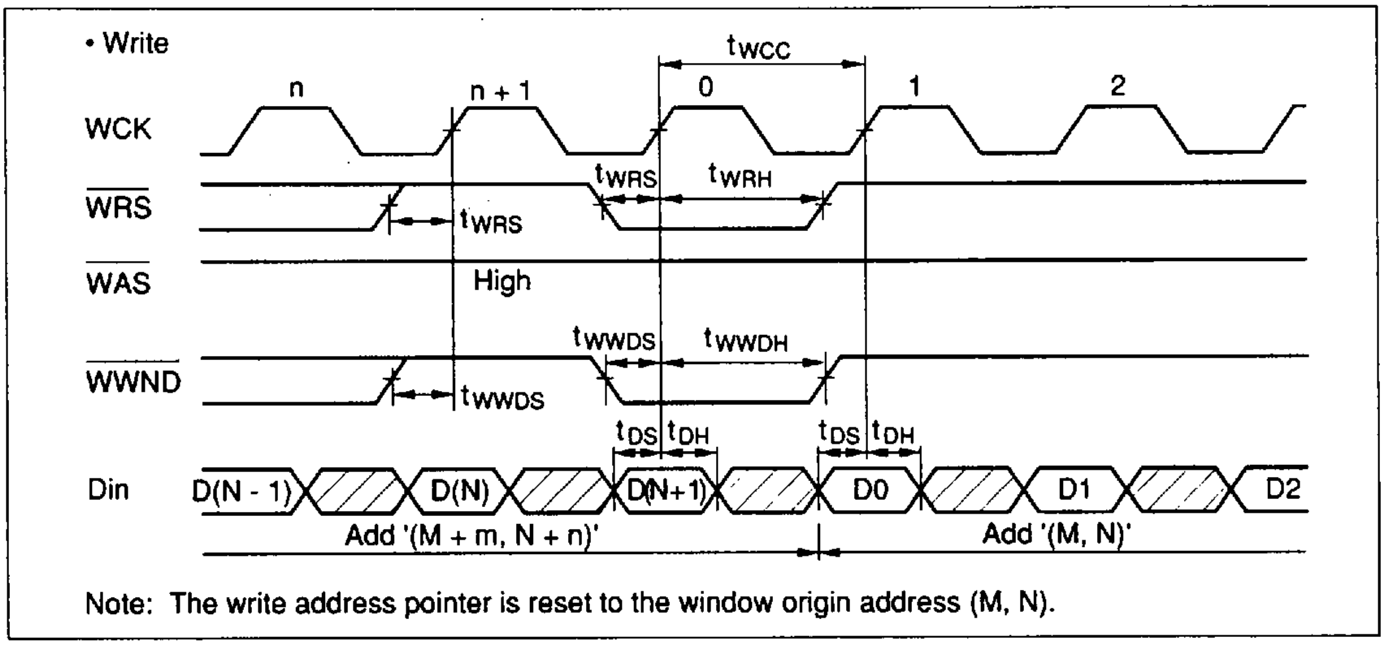
Note: The write address pointer is reset to (0, 0), and window mode is turned off if it was on.



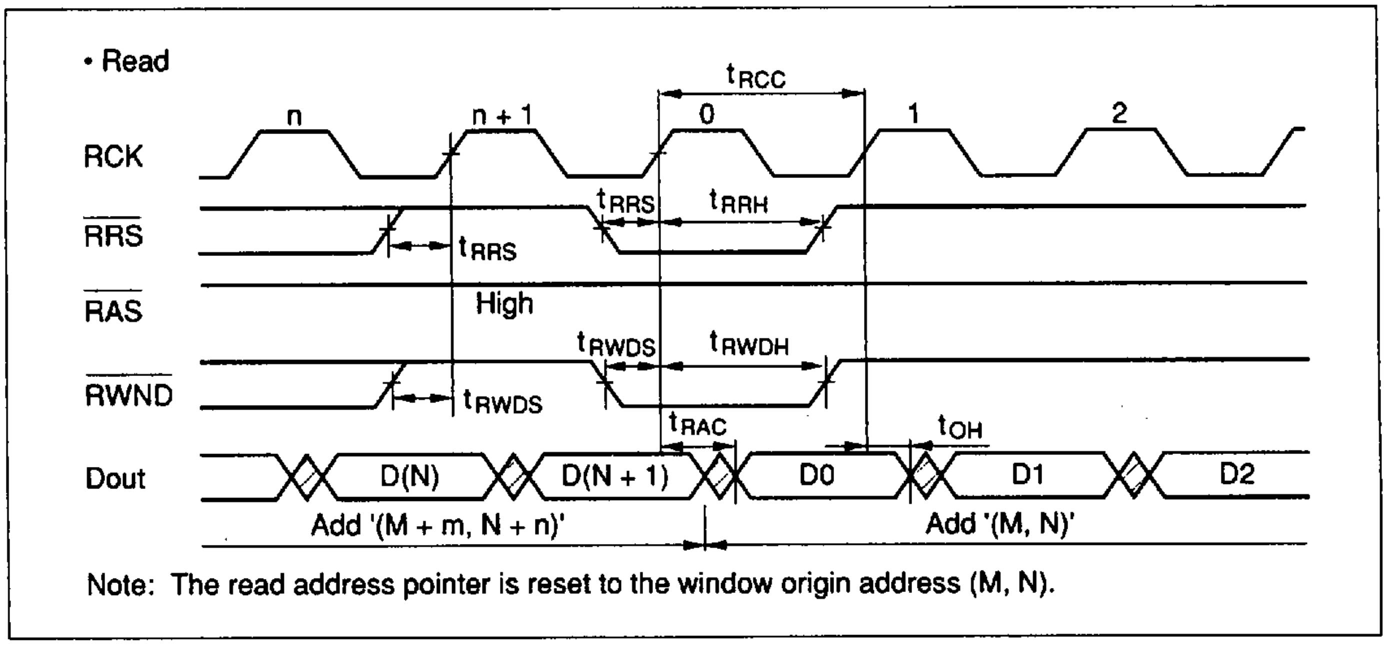
Note: The read address pointer is reset to (0, 0), and window mode is turned off if it was on.

- **Reset to the Window Origin**

These figures show the timing charts for resetting the address pointer to the window origin address (M, N) during window scan mode execution



Note: The write address pointer is reset to the window origin address (M, N).



Note: The read address pointer is reset to the window origin address (M, N).