

SDRAM CONTROLLER DESIGN

Submitted in partial fulfillment of

EE 5313

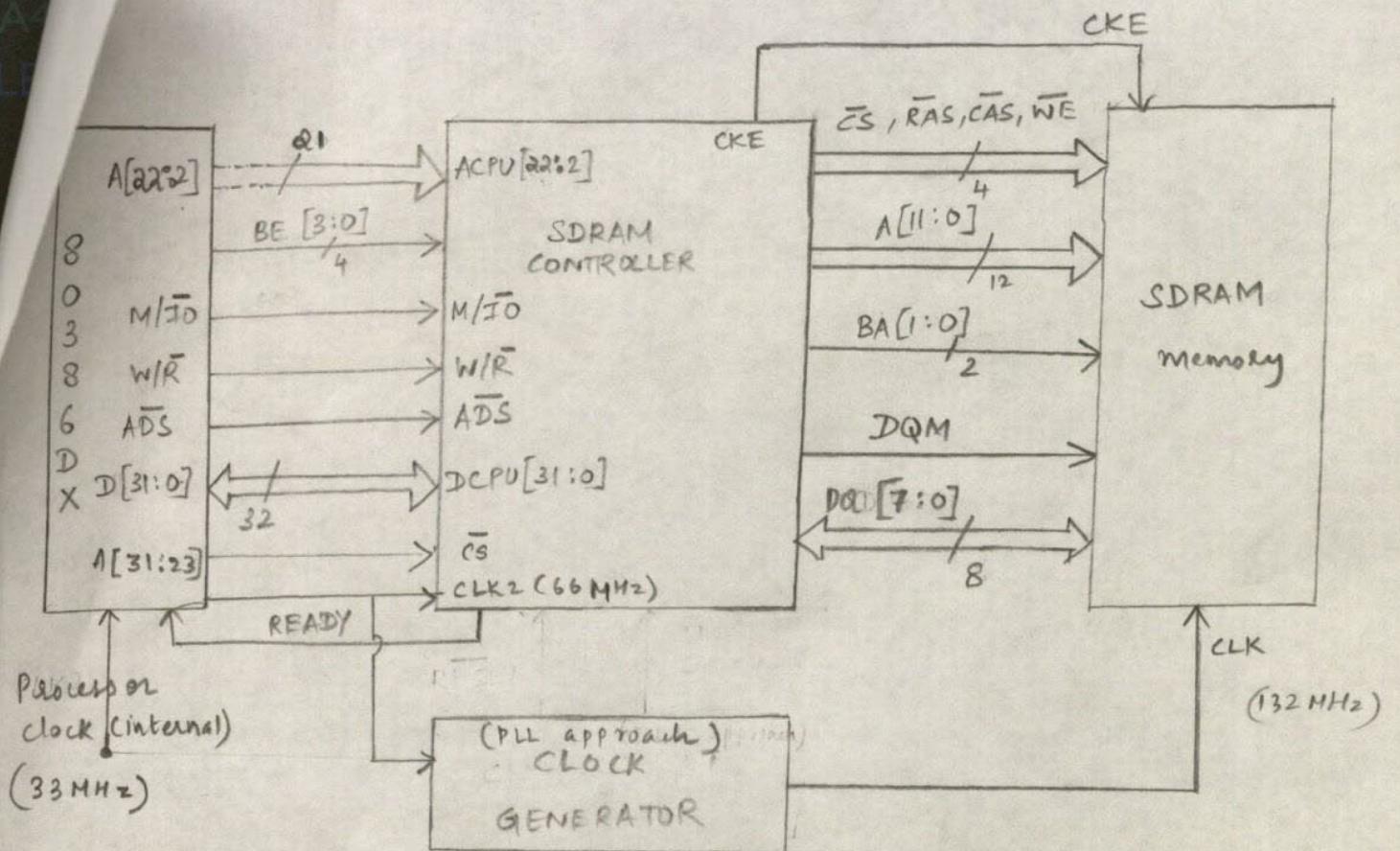
Microprocessor Systems

by

Lina Sera Varghese - 1001236490

Shrutilaya Ramesh- 1001156036

Farah Umme Kulsum Ahmed - 1001119162



HIGH LEVEL CONNECTION DIAGRAM

The SDRAM controller interfaces a 32 bit, CMOS Intel 386TM DX microprocessor with a synchronous DRAM MT48LC8M8A2 - [2 Meg x 8 x 4 banks] memory. Normally, 80386DX can interface with asynchronous memories only, but the SDRAM controller design enables the processor to be interfaced with synchronous memories as well.

Design features

Burst length BL=4

CAS latency, CL=2 clocks

CLK2 = 66 MHz, CLK = 132 MHz

SDRAM CONTROLLER DESIGN

Submitted in partial fulfillment of

EE 5313

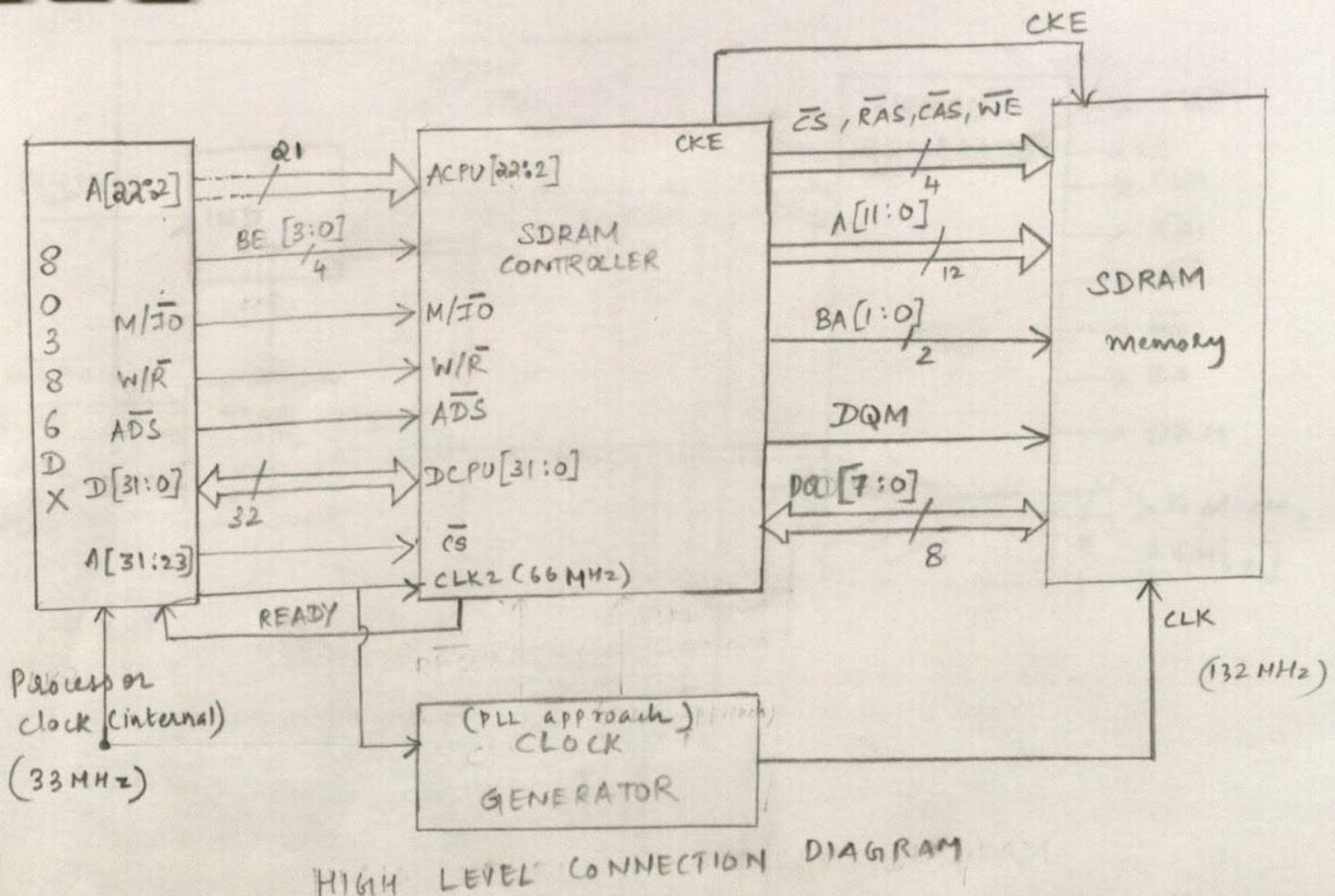
Microprocessor Systems

by

Lina Sera Varghese - 1001236490

Shrutilaya Ramesh- 1001156036

Farah Umme Kulsum Ahmed - 1001119162



HIGH LEVEL CONNECTION DIAGRAM

The SDRAM Controller interfaces a 32 bit, CMOS Intel 386TM DX microprocessor with a synchronous DRAM MT48LC8M8A2 - [2 Meg x 8 x 4 banks] memory. Normally, 80386DX can interface with asynchronous memories only, but the SDRAM controller design enables the processor to be interfaced with synchronous memories as well.

Design features

Burst length BL = 4

CAS latency, CL = 2 clocks

CLK2 = 66 MHz, CLK = 132 MHz

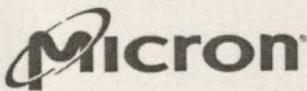
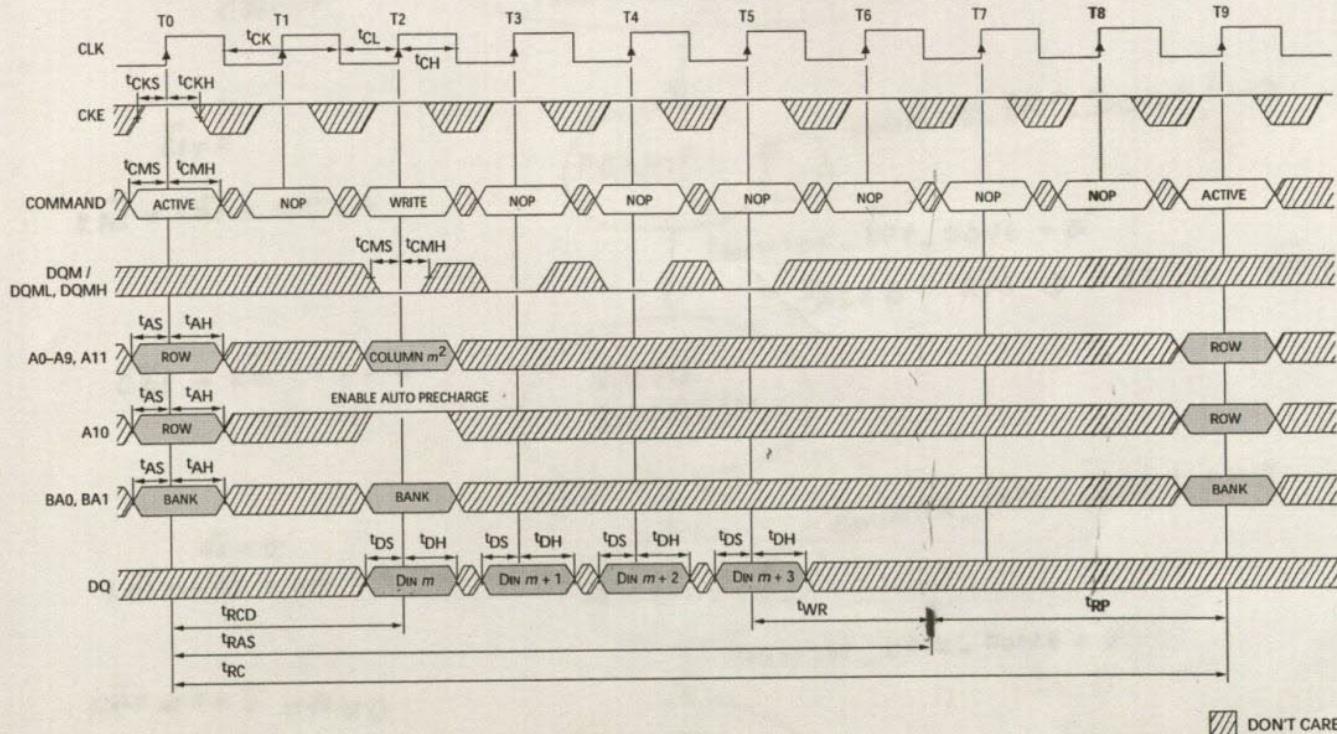
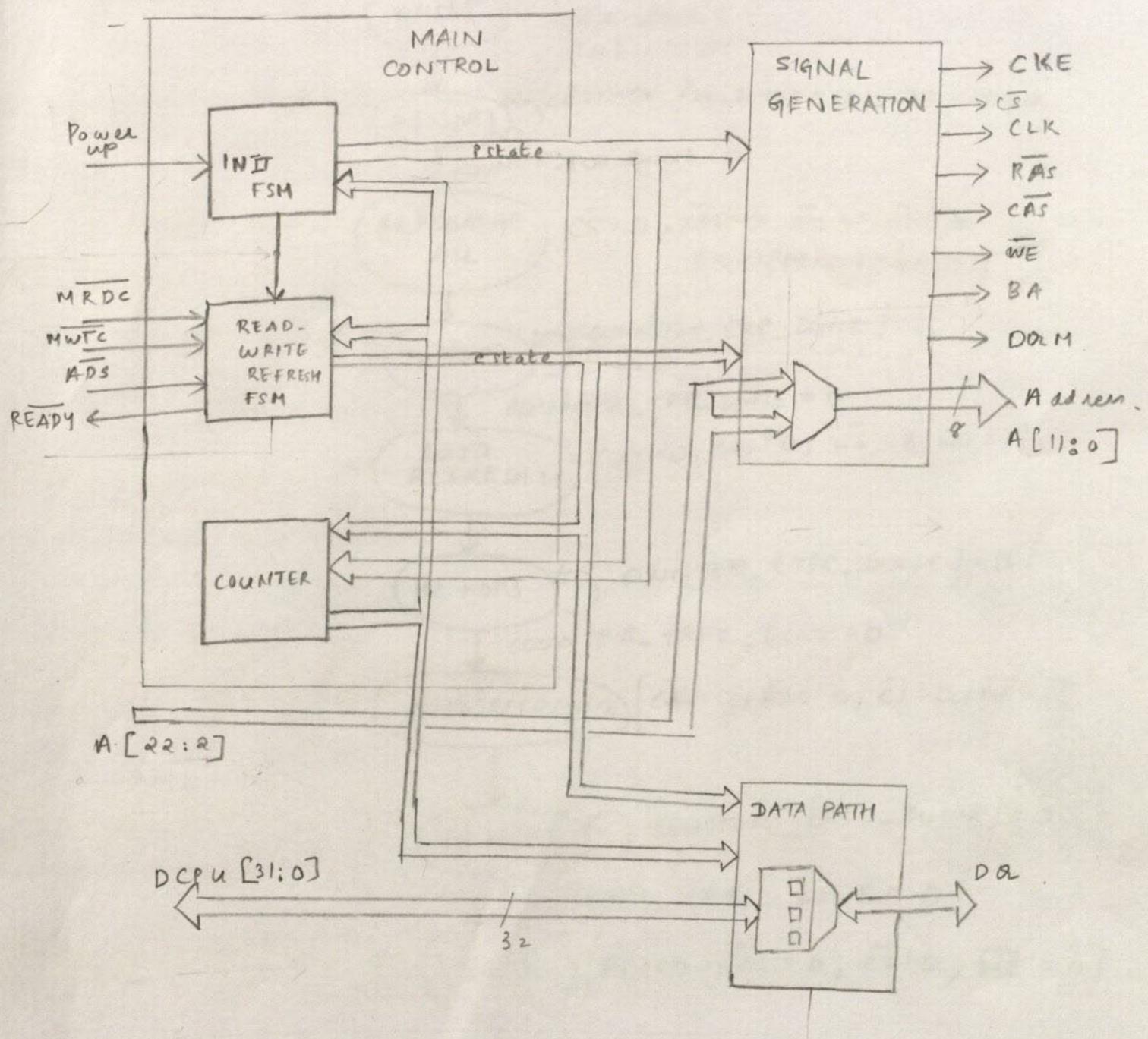


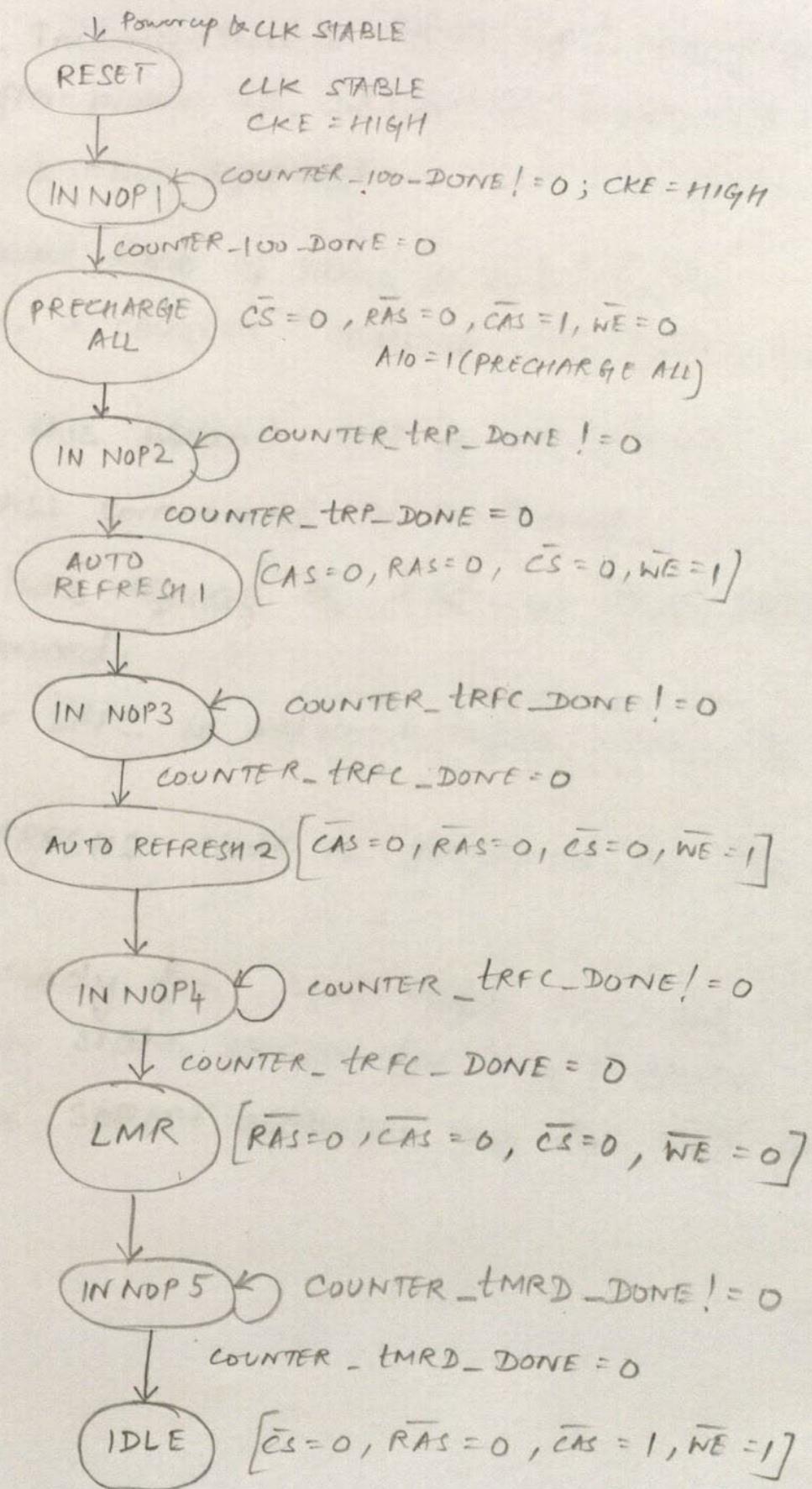
Figure 48: WRITE - With Auto Precharge



- Notes:
1. For this example, BL = 4.
 2. x16: A8, A9 and A11 = "Don't Care"
 - x8: A9 and A11 = "Don't Care"
 - x4: A11 = "Don't Care"



INITIALISATION OF SDRAM



Initialization

- (1) The first stage in Initialization is Power up & applying V_{DD} and V_{DDQ} . After power up, all counters are loaded.
- (2) A stable clock is now applied.
- (3) At this point, a wait time of 100us is applied. To enable this time, a counter "COUNTER_100_BN" is set to low.
- (4) At the end of this period, EKE is set to high.
- (5) A PRECHARGE ALL command is performed.
- (6) After a wait time equal to tRP, an AUTO REFRESH1 command is performed.
- (7) A time period = tRFC is allowed before moving to next state.
- (8) Another AUTO REFRESH2 state is performed for the same time frame.
- (9) The SDRAM is ready for LOAD MODE REGISTER (LMR) programming. This state persists for time = tMRD
- (10) After LMR, the SDRAM controller goes to IDLE mode.

STATE TRANSITION TABLE

<u>CURRENT STATE</u>	<u>NEXT STATE</u>	<u>CONDITION</u>
X	RESET = 0	RESET# = 0
RESET	INN0P1	Next rising edge of clock
INN0P1	INN0P1	COUNTER_100_DONE != 0
INN0P1	PRECHARGE ALL	COUNTER_100_DONE = 0
PRECHARGE ALL	INN0P2	Next rising edge of clock
INN0P2	INN0P2	COUNTER_ERP_DONE != 0
INN0P2	AUTO REFRESH1	COUNTER_ERP_DONE = 0
AUTOREFRESH1	INN0P3	Next rising edge of clock
INN0P3	INN0P3	COUNTER_ERFC_DONE != 0
INN0P3	AUTO REFRESH2	COUNTER_ERFC_DONE = 0
AUTOREFRESH2	INN0P4	Next rising edge of clock
INN0P4	INN0P4	COUNTER_ERFC_DONE != 0
INN0P4	LMR	COUNTER_ERFC_DONE = 0
LMR	INN0P5	Next rising edge of clock
INN0P5	INN0P5	COUNTER_EMRD_DONE != 0
INN0P5	IDLE	COUNTER_EMRD_DONE = 0

OUTPUT EQUATIONS FOR INITIALLISATION

OUTPUT

CKE = 1

COUNTER_100_EN = 1

A10 = 1

COUNTER_ERP_EN = 1

COUNTER_ERFC_EN = 1

COUNTER_tMRD_EN = 1

READY = 1

$$A[11:0] = (00, 0, 00, \underline{010}, \underline{0}, \underline{010})$$

CL BL

Burst length = 4 (010)

Burst type = 0, sequential

CAS latency = 2 (010)

Op Mode = Standard operation

Write Burst mode = Programmed Burst length

BA0, BA1 = ACPH[22:21]

ST

STATE == PRECHARGE ALL

RAS# = 0

(STATE == PRECHARGE ALL) ||

(STATE == AUTOREFRESH1) ||

(STATE == AUTOREFRESH2) ||

(STATE == LMR)

CAS# = 0

(STATE == AUTO REFRESH1) ||

(STATE == AUTO REFRESH2) ||

(STATE == LMR)

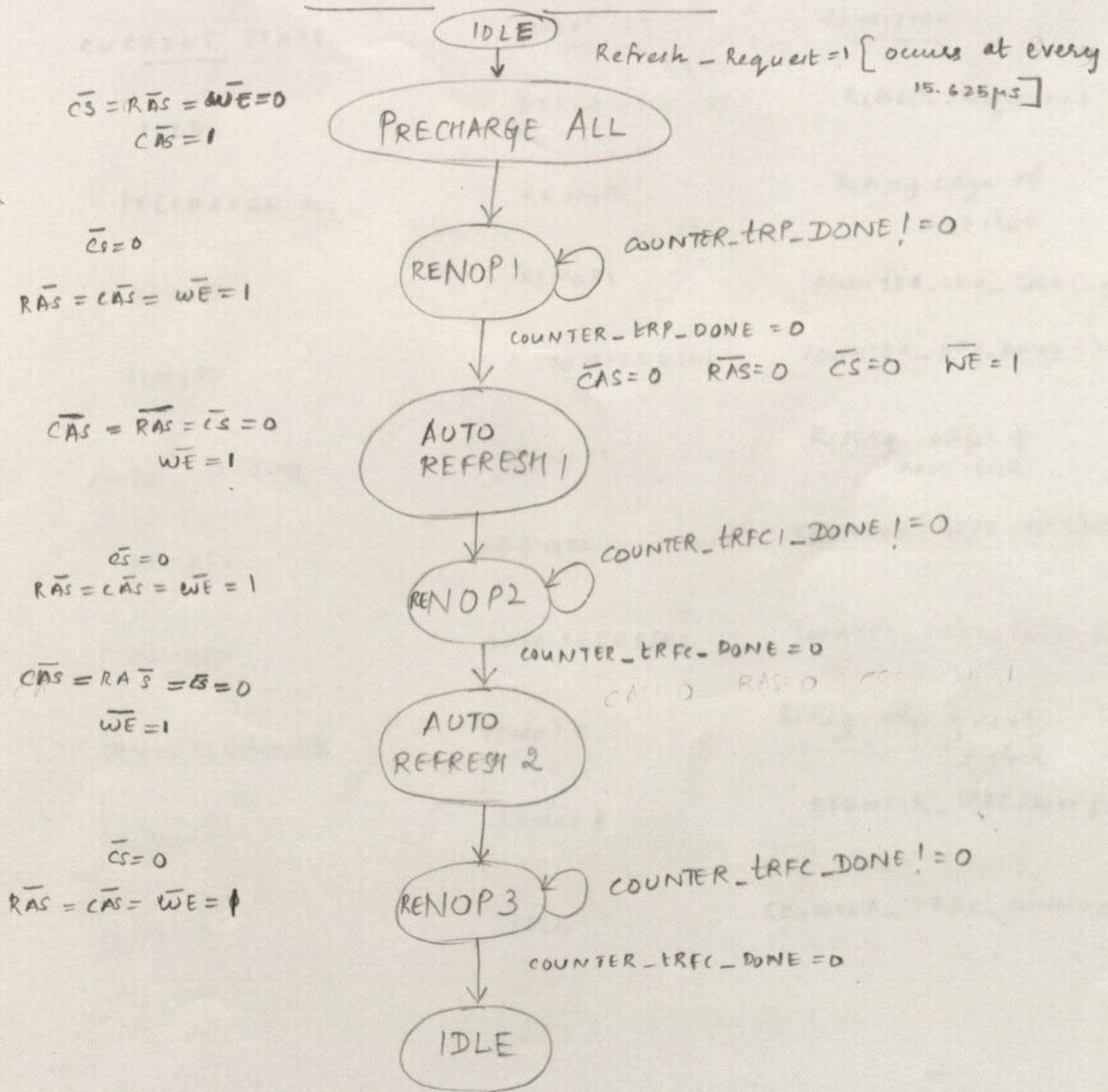
$\overline{WE} = 0$

(STATE == PRECHARGE ALL) ||

(STATE == LMR)

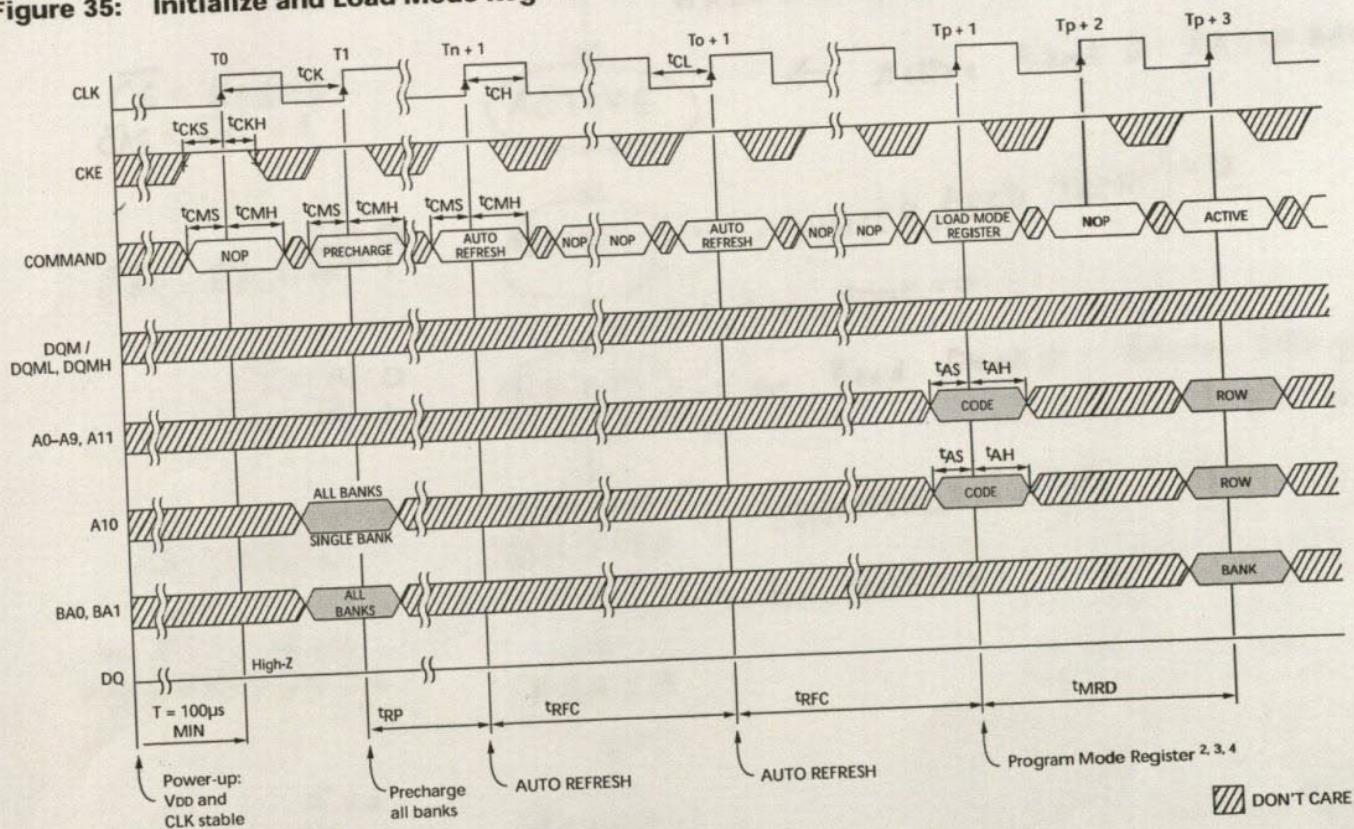
DQ = High Z state

AUTO REFRESH STATE DIAGRAM



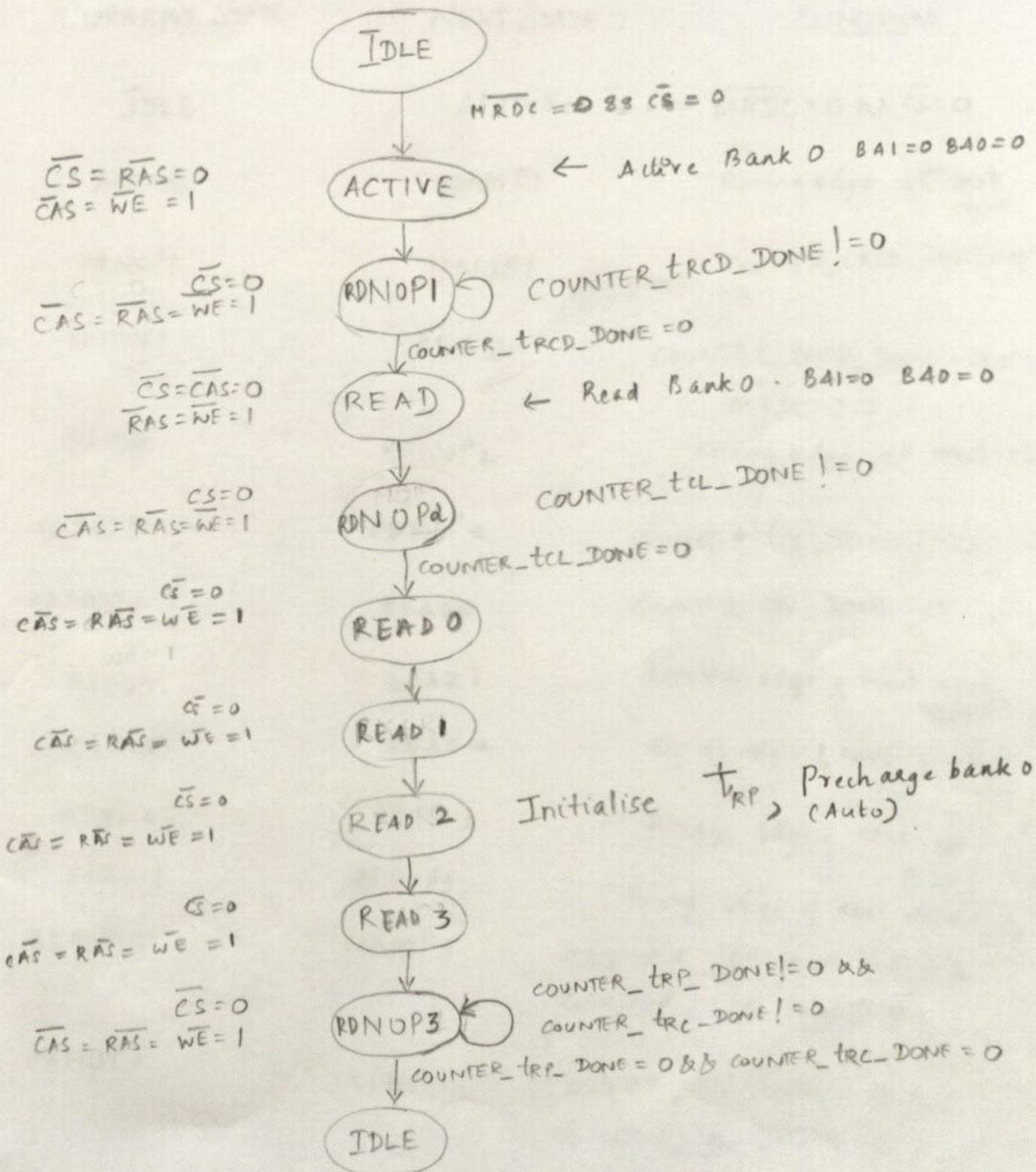
Timing Diagrams

Figure 35: Initialize and Load Mode Register



- Notes:
1. If CS# is HIGH at clock HIGH time, all commands applied are NOP.
 2. The mode register may be loaded prior to the AUTO REFRESH cycles if desired.
 3. JEDEC and PC100 specify three clocks.
 4. Outputs are guaranteed High-Z after command is issued.

READ STATE DIAGRAM



STATE TRANSITION TABLE FOR READ

CURRENT STATE	NEXT STATE	CONDITIONS
IDLE	ACTIVE	$\overline{MRDC} = 0 \text{ \& } \overline{CS} = 0$
ACTIVE	RDNOP1	Rising edge of next clock
RDNOP1	RDNOP1	COUNTER_TRCD_DONE != 0
RDNOP1	READ	COUNTER_TRCD_DONE = 0 $\text{ \& } \overline{MRDC} = 0$
READ	RDNOP2	Rising edge of next clock
RDNOP2	RDNOP2	COUNTER_TCL_DONE != 0
RDNOP2	READ0	COUNTER_TCL_DONE = 0
READ0	READ1	Rising edge of next clock (CLK)
READ1	READ2	Rising edge of next clock (CLK)
READ2	READ3	Rising edge of next clock (CLK)
READ3	RDNOP3	Rising edge of next clock (CLK)
RDNOP3	RDNOP3	COUNTER_TRP_DONE != 0 $\text{ \& } \overline{CS} = 0$
RDNOP3	IDLE	COUNTER_TRP_DONE = 0 $\text{ \& } \overline{CS} = 0$

OUTPUTS OF READ FSM

OUTPUT

$$BA[1:0] = ACPU[22:21]$$

$$A[11:0] = ACPU[20:9]$$

$$COUNTER_TRCD_EN = 1$$

$$COUNTER_TRAS_EN = 1$$

$$COUNTER_TRC_EN = 1$$

$$COUNTER_TCL_EN = 1$$

$$COUNTER_TRP_EN = 1$$

$$A[8:2] = ACPU[8:2]$$

$$A[1:0] = 0$$

A9 & A11 are don't cares

$$A10 = 1$$

$$RAS\# = 0$$

$$CAS\# = 0$$

$$DQ[7:0] = DCPU[7:0]$$

$$DQ[7:0] = DCPU[15:8]$$

$$DQ[7:0] = DCPU[23:16]$$

$$DQ[7:0] = DCPU[31:24]$$

STATE EQUATIONS

$$STATE == \{ ACTIVE, READ \}$$

$$STATE == ACTIVE$$

// Row address presented

$$STATE == ACTIVE$$

$$STATE == ACTIVE$$

$$STATE == ACTIVE$$

$$STATE == READ$$

$$STATE == READ2$$

$$STATE == READ$$

// column lines
addressing

$$STATE == READ$$

// Enable Auto Precharge

$$STATE == ACTIVE$$

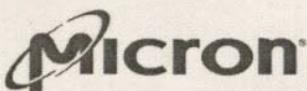
$$STATE == READ$$

$$STATE == READ0$$

$$STATE == READ1$$

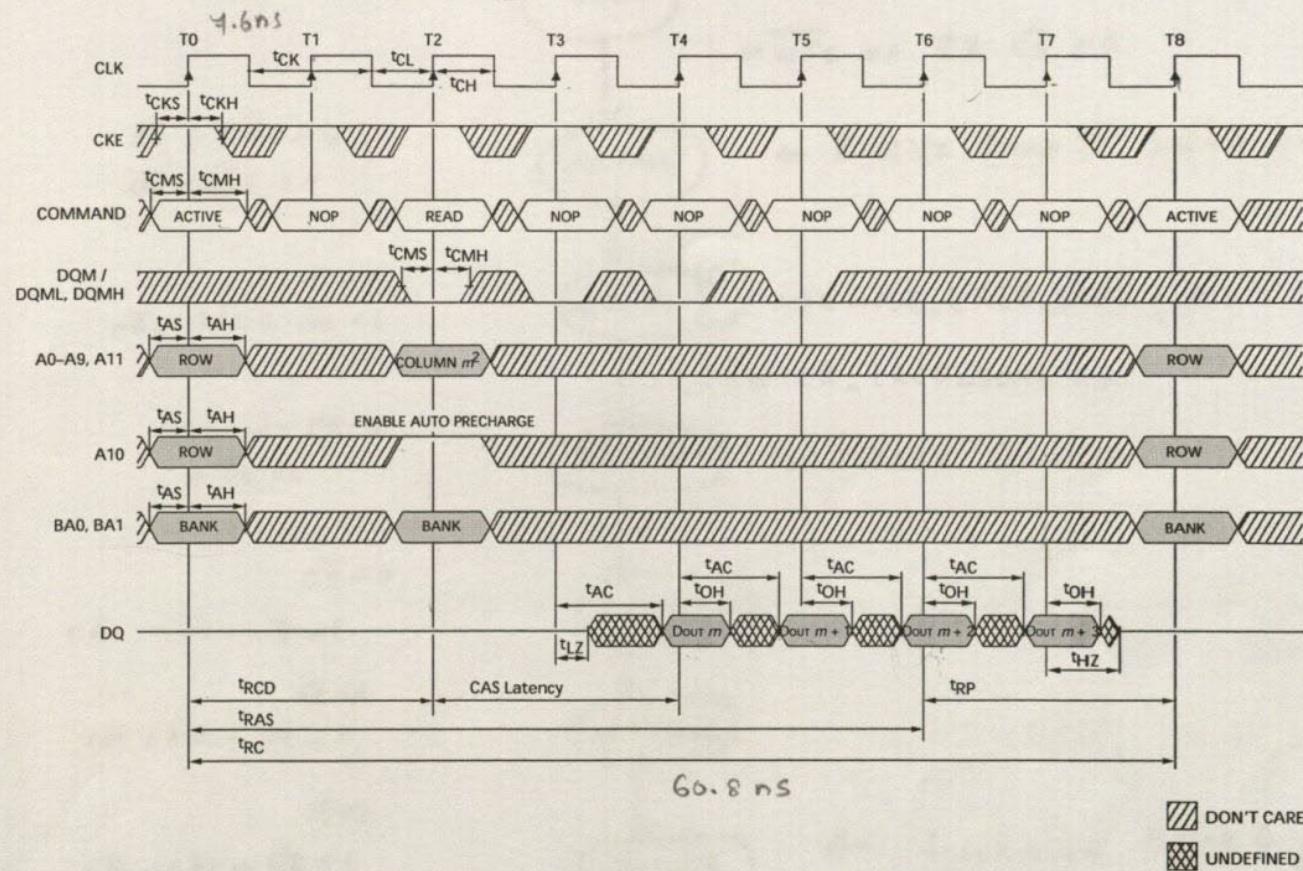
$$STATE == READ2$$

$$STATE == READ3$$



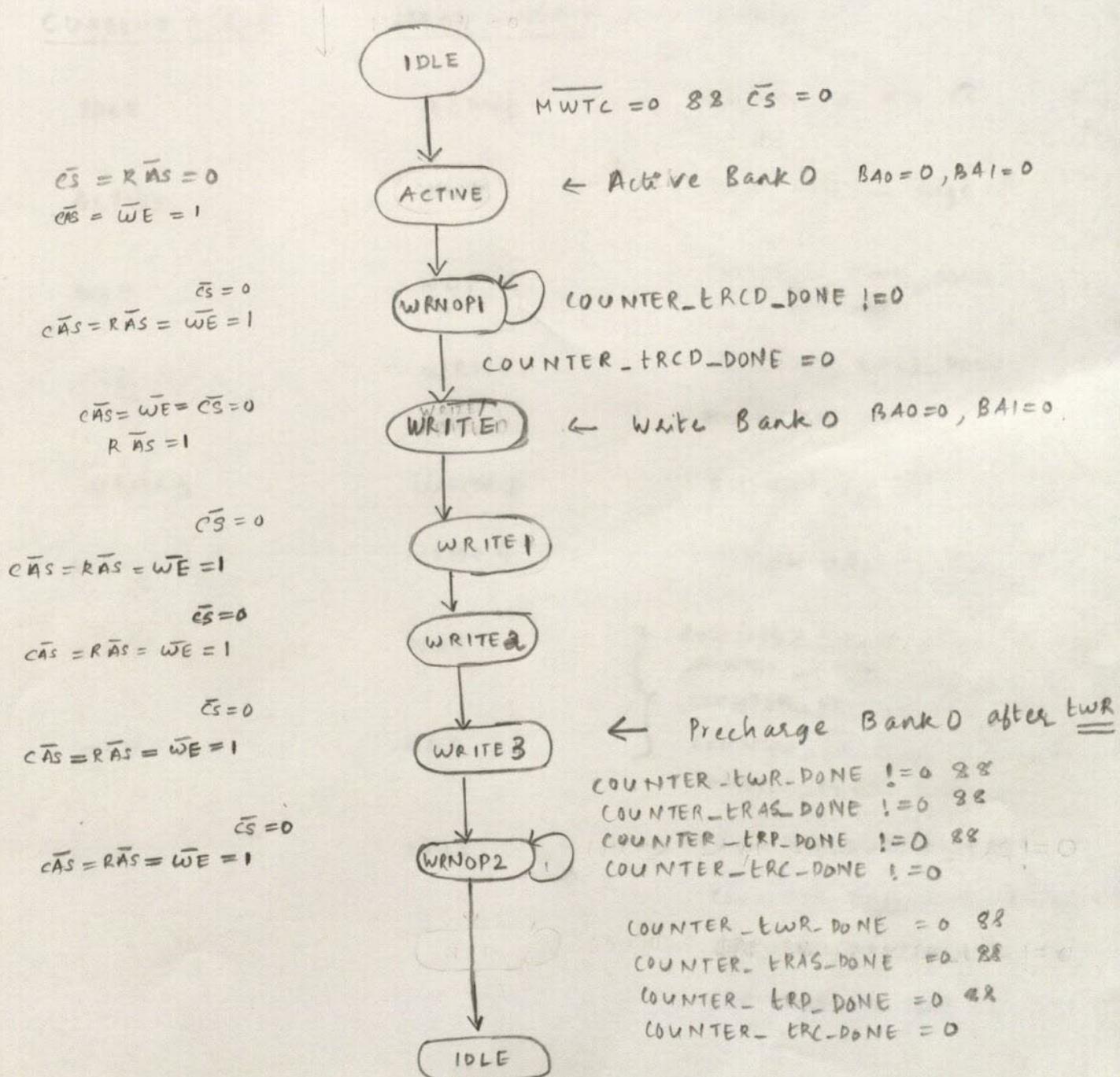
64Mb: x4, x8, x16 SDRAM Timing Diagrams

Figure 41: READ - With Auto Precharge



- Notes:
1. For this example, BL = 4, and CL = 2.
 2. x16: A8, A9 and A11 = "Don't Care"
 - x8: A9 and A11 = "Don't Care"
 - x4: A11 = "Don't Care"

WRITE STATE DIAGRAM:



STATE TRANSITION TABLE FOR WRITE

<u>CURRENT STATE</u>	<u>NEXT STATE</u>	<u>CONDITION</u>
IDLE	ACTNE	$\overline{HWTC} = 0 \quad 88 \quad \overline{CS} = 0$
ACTIVE	NOP	Next RISING edge of clock (CLK)
NOP	NOP	COUNTER_ERCD_DONE != 0
NOP	WRITE1	COUNTER_ERCD_DONE = 0 88 $\overline{HWTC} = 0$
WRITE1	WRITE2	RISING edge of clock (CLK)
WRITE3	WRITE4	RISING edge of clock (CLK)
WRITE4	NOP	$\left. \begin{array}{l} COUNTER_EWR_DONE != 0 \\ COUNTER_ERAS_DONE != 0 \\ COUNTER_ERP_DONE != 0 \\ COUNTER_ERC_DONE != 0 \\ COUNTER_EDRAM_DONE != 0 \end{array} \right\}$
NOP	NOP	
NOP	IDLE	\rightarrow COUNTER_EWR_DONE = 0 88 $COUNTER_ERAS_DONE = 0 \quad 88$ $COUNTER_ERP_DONE = 0 \quad 88$ $COUNTER_ERC_DONE = 0$

OUTPUTS OF WRITE FSM

OUTPUTS.

$$1. BA[1:0] = ACPU[22:21]$$

$$2. A[11:0] = ACPU[20:9]$$

$$\text{COUNTER_TRCD_EN} = 1$$

$$\text{COUNTER_TRAS_EN} = 1$$

$$\text{COUNTER_TWR_EN} = 1$$

$$DQ[7:0] = DCPU[7:0]$$

$$DQ[7:0] = DCPU[15:8]$$

$$DQ[7:0] = DCPU[23:16]$$

$$DQ[7:0] = DCPU[31:24]$$

$$A[8:2] = ACPU[8:2]$$

$$A[1:0] = 0$$

A[9:8] all are don't cares.

$$A10 = 1$$

$$DQM = \overline{BE0}$$

$$= \overline{BE1}$$

$$= \overline{BE2}$$

$$= \overline{BE3}$$

$$\overline{RAS} = 0$$

$$\overline{CAS} = 0$$

$$\overline{WE} = 0$$

$$COUNT_EDQM_EN = 1$$

STATE EQUATIONS

$$\text{STATE} = \{\text{ACTIVE}, \text{WRITE}\}$$

$$\text{STATE} = \text{ACTIVE}$$

// when state is ACTIVE, Row address is presented.

$$\text{STATE} = \text{ACTIVE}$$

$$\text{STATE} = \text{ACTIVE}$$

$$\text{STATE} = \text{ACTIVE}$$

$$\text{STATE} = \text{WRITE 3}$$

$$\text{STATE} = \text{WRITE}$$

$$\text{STATE} = \text{WRITE1}$$

$$\text{STATE} = \text{WRITE2}$$

$$\text{STATE} = \text{WRITE3}$$

$$\text{STATE} = \text{WRITE}.$$

// Column liner addressing

$$\text{STATE} = \text{WRITE}$$

// Enable Auto Precharge.

$$\text{STATE} = \text{WRITE}$$

$$\text{STATE} = \text{WRITE1}$$

$$\text{STATE} = \text{WRITE2}$$

$$\text{STATE} = \text{WRITE3}$$

$$\text{STATE} = \text{ACTIVE}$$

$$\text{STATE} = \text{WRITE}$$

$$\text{STATE} = \text{WRITE}$$

$$\text{STATE} = \text{WRITE}$$

STATE TRANSITION TABLE

<u>CURRENT STATE</u>	<u>NEXT STATE</u>	<u>CONDITION</u>
IDLE	PRECHARGE_ALL	Refresh_Request = 1
PRECHARGE_ALL	RENOPI	Rising edge of next clock
RENOPI	RENOPI	COUNTER_ERP_DONE != 0
RENOPI	AUTO_REFRESH1	COUNTER_ERP_DONE = 0
AUTO_REFRESH1	RENOPI2	Rising edge of next clock
RENOPI2	RENOPI2	COUNTER_ERFC_DONE != 0
RENOPI2	AUTO_REFRESH2	COUNTER_ERFC_DONE = 0
AUTO_REFRESH2	RENOPI3	Rising edge of next clock
RENOPI3	RENOPI3	COUNTER_ERFC_DONE != 0
RENOPI3	IDLE	COUNTER_ERFC_DONE = 0

OUTPUTS FROM AUTO REFRESH

Outputs

$\overline{RAS} = 0$

state equations

(STATE == PRECHARGE ALL) ||

(STATE == AUTO REFRESH1) ||

(STATE == AUTO REFRESH2)

$\overline{CAS} = 0$

(STATE == AUTO REFRESH1) ||

(STATE == AUTO REFRESH2)

$\overline{WE} = 0$

STATE == PRECHARGE ALL

A10 = 1

STATE == PRECHARGE ALL

$\overline{READY} = 1$

STATE == IDLE ||

($\overline{MRDC} = 1$ || $\overline{MWTC} = 1$)

COUNTER_ERP_EN = 1

STATE == PRECHARGE ALL

COUNTER_ERFC_EN = 1

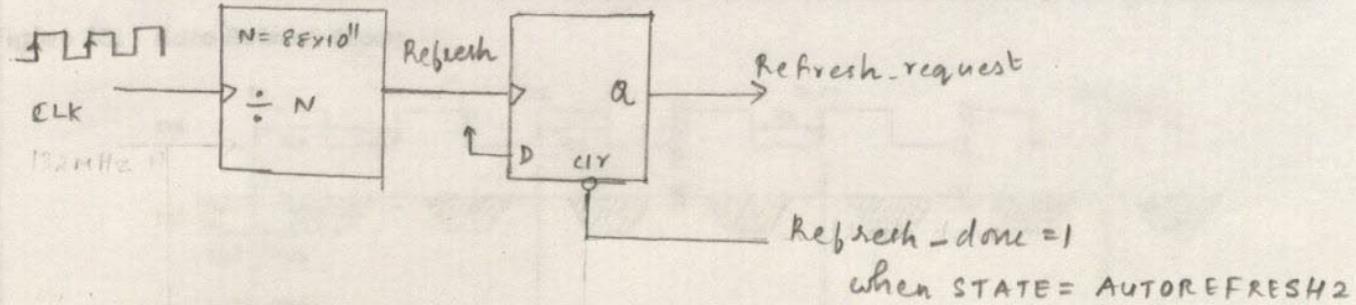
(STATE == AUTO REFRESH1) ||

(STATE == AUTO REFRESH2)

Refresh-done = 1

STATE == AUTO REFRESH2.

Design of Refresh counter



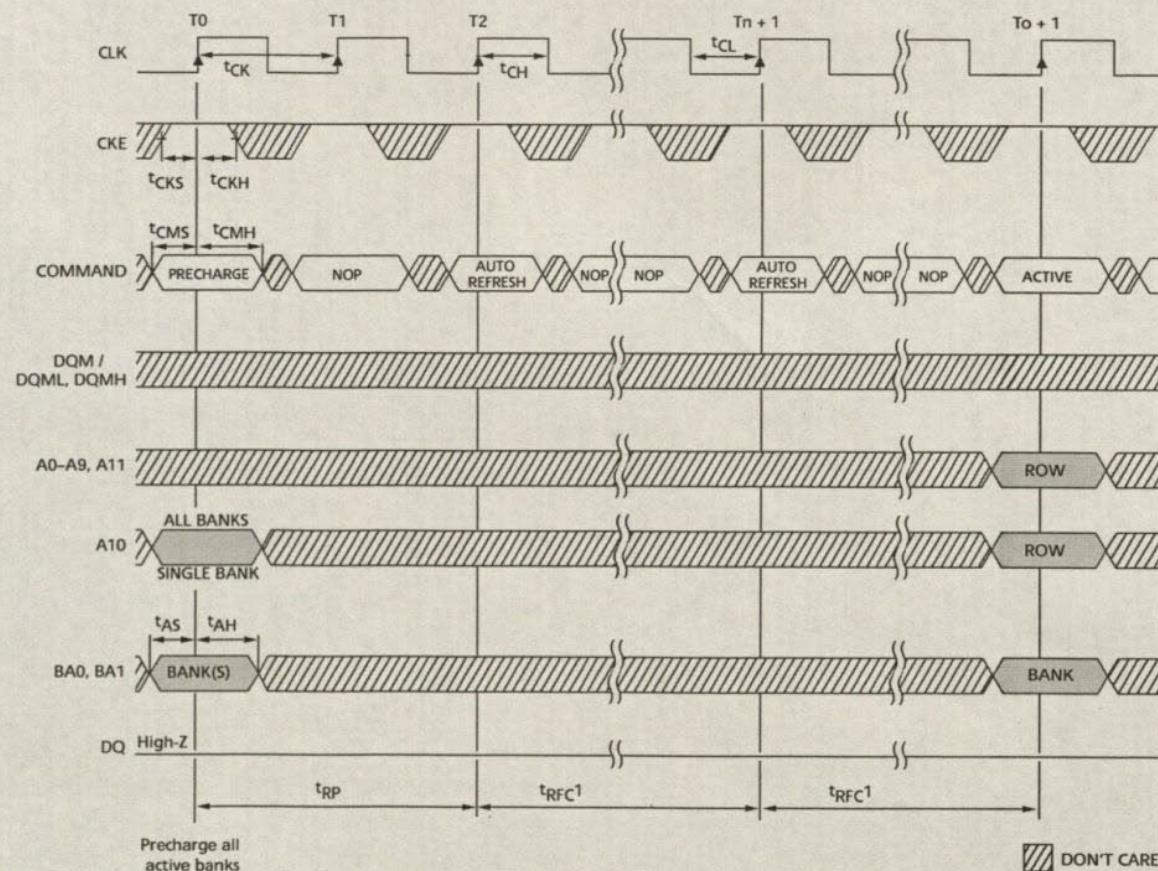
4096 Rows of SDRAM memory is to be refreshed at every 64 ms. Hence each row refresh rate = $\frac{64 \text{ ms}}{4096} = 15.625 \text{ ms}$.

The above design generates Refresh request at every 15.625 ms .
 when the No: of input clock cycles = $\frac{132 \text{ M}}{15 \mu} = 88 \times 10^11$,
 refresh request is generated. when STATE == AUTOREFRESH2,
 Refresh-done=1 and the request is cleared.



64Mb: x4, x8, x16 SDRAM Timing Diagrams

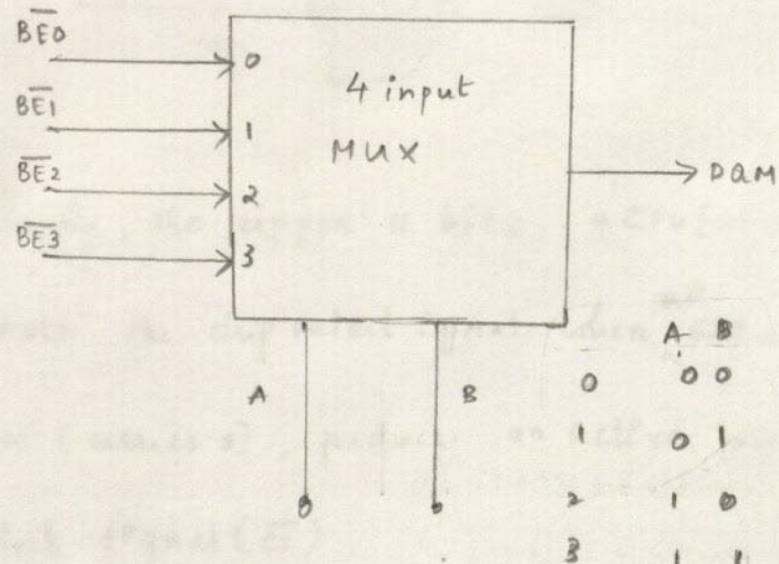
Figure 38: Auto Refresh Mode



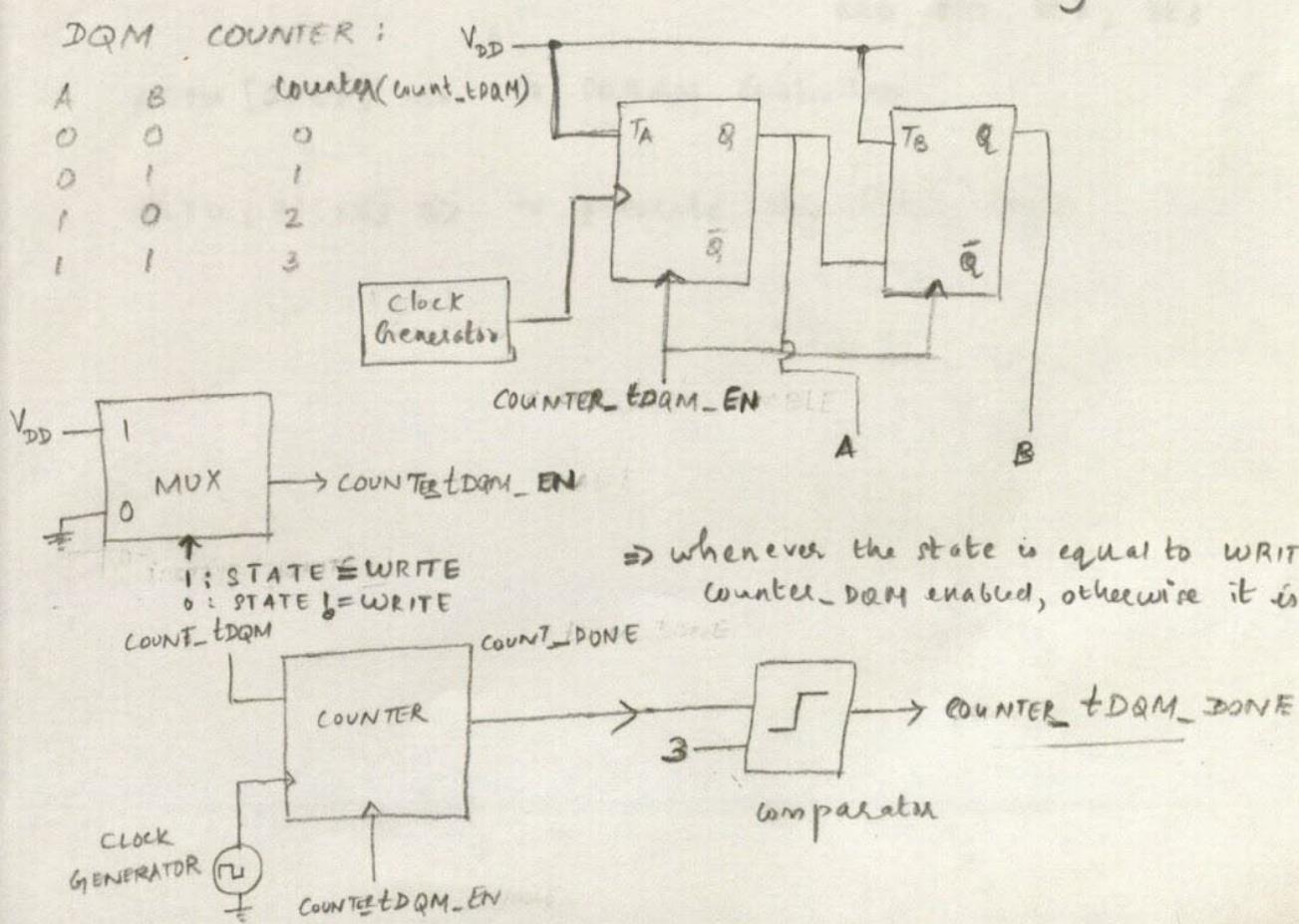
Notes:

1. Each AUTO REFRESH command performs a refresh cycle. Back-to-back commands are not required.

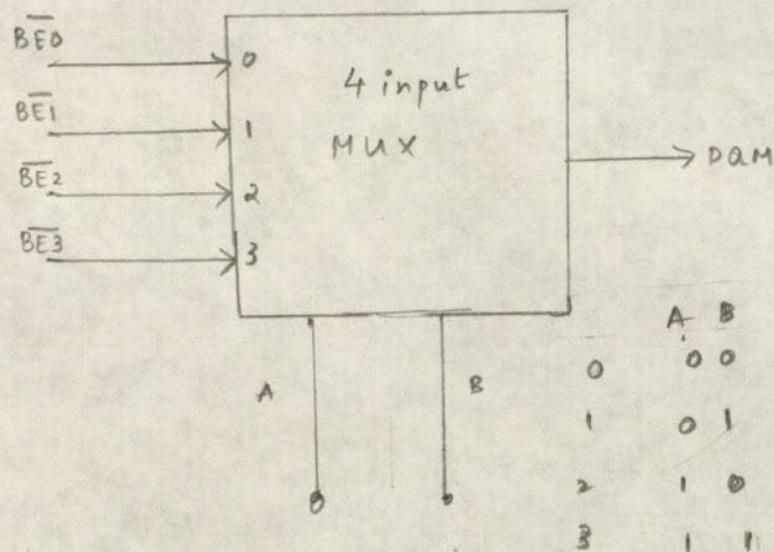
Generation of DQM signal



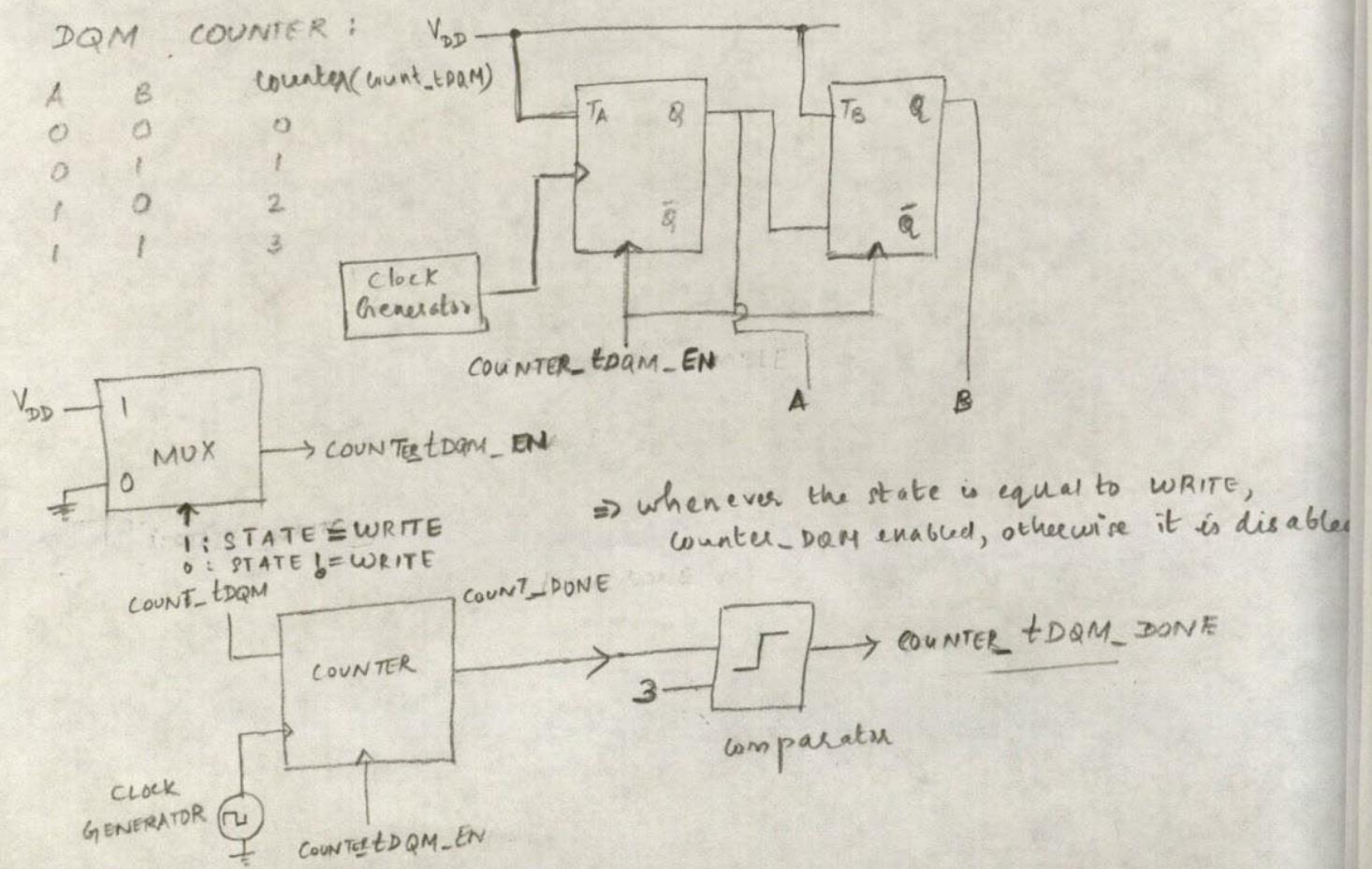
Design of an UP COUNTER which counts rising edges of 4 clocks when the STATE == WRITE, the Counter begins counting.



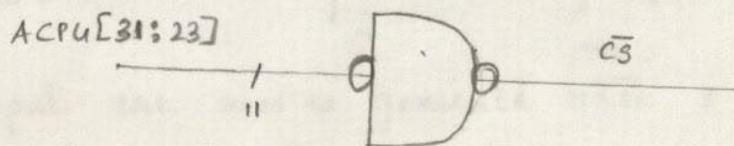
Generation of DQM signal



Design of an UP COUNTER which counts rising edges of 4 clocks
when the STATE == WRITE, the counter begins counting.



GENERATION OF CHIP SELECT SIGNAL [\bar{CS}]



As shown in fig , the upper 11 bits $ACPU[31:23]$ are used to generate the chip select signal. when ^{all} address lines $ACPU[31:23]$ are low (carries 0) , produces an active low signal named as chip select signal (\bar{CS}) .

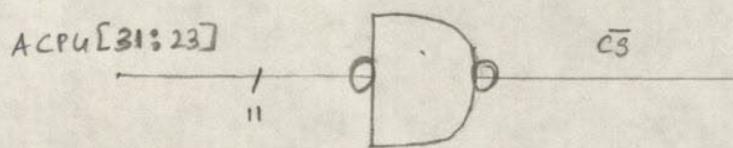
of 31 address lines of 80386DX processor

$ACPU[1:0] \Rightarrow$ used to generate Bank Enable signals $\overline{BE_0}, \overline{BE_1}, \overline{BE_2}, \overline{BE_3}$

$ACPU[22:2] \Rightarrow$ to SDRAM controller

$ACPU[31:23] \Rightarrow$ to generate chip select logic

GENERATION OF CHIP SELECT SIGNAL [\overline{CS}]



As shown in fig, the upper 11 bits $ACPU[31:23]$ are used to generate the chip select signal. When ^{all} address lines $ACPU[31:23]$ are low (carries 0), produce an active low signal named as chip select signal (\overline{CS}).

of 31 address lines of 80386DX processor

$ACPU[1:0] \Rightarrow$ used to generate Bank Enable signals $\overline{BE}_0, \overline{BE}_1, \overline{BE}_2, \overline{BE}_3$

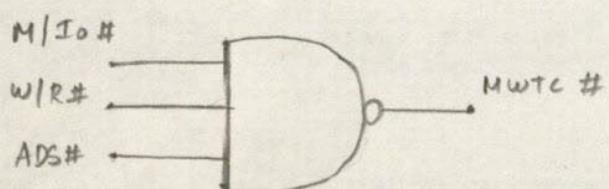
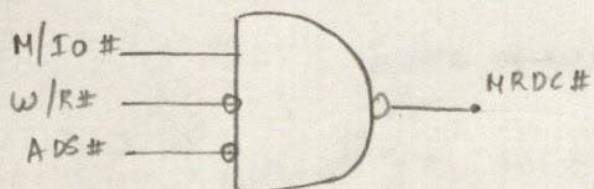
$ACPU[22:2] \Rightarrow$ to SDRAM Controller

$ACPU[31:23] \Rightarrow$ to generate chip select logic.

Generation of \overline{MRDC} & \overline{MWTC}

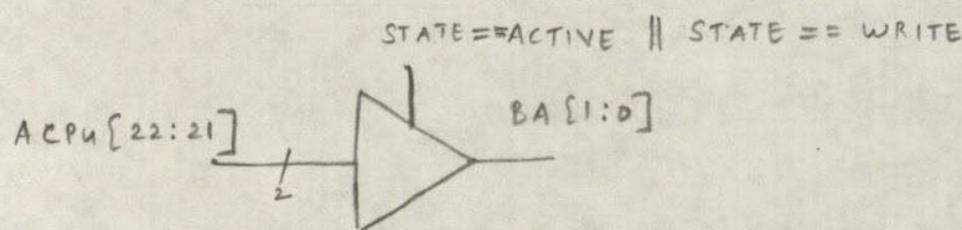
The 80386DX has 3 outputs, M/IO#, W/R# and ADS#.

These outputs are used to generate \overline{MRDC} & \overline{MWTC} signals which enables triggering of Read Finite state Machine & Write Finite state machine respectively.



Generation of BANK, ROW AND COLUMN ADDRESSES

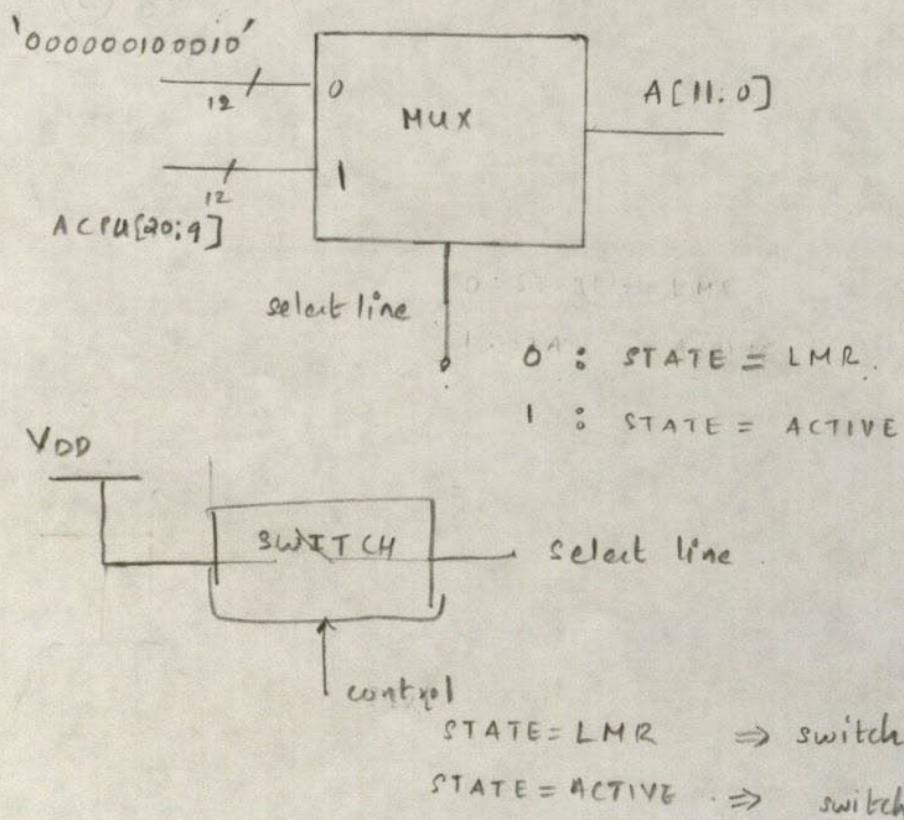
Generation of Bank select signals $BA[1:0]$



multi-bit Tri state Buffer.

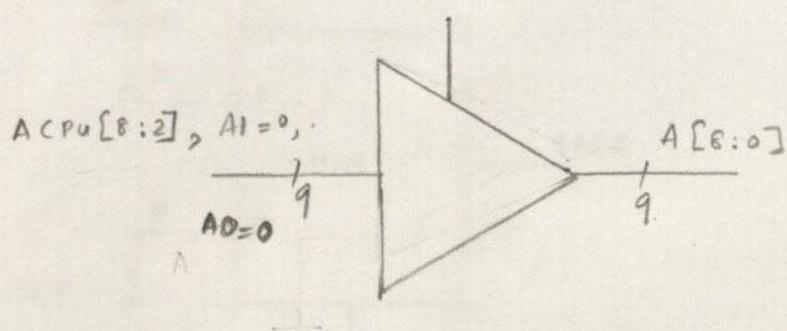
when state is active or write, the upper 2 address lines
 $A[CPU[22:21]]$ behave as bank select signals $BA[1:0]$.

Generation of Row addresses



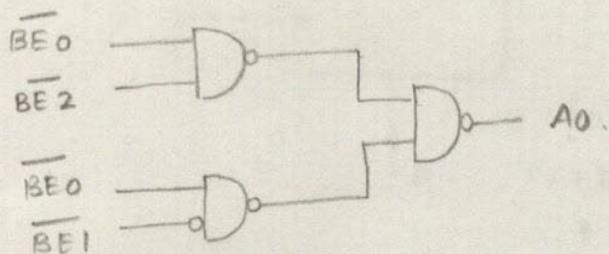
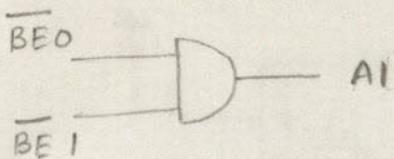
Generation of column addresses.

(STATE = WRITE || STATE == READ)

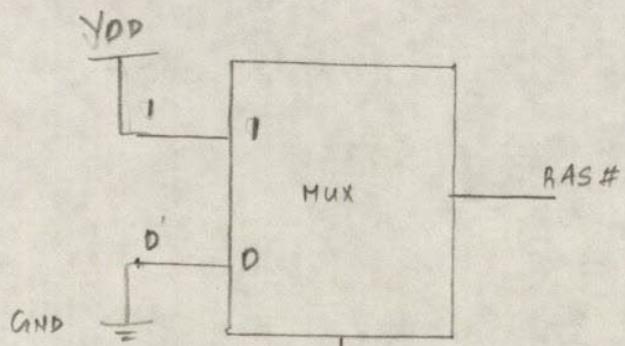


multi-bit tristate buffer

- * $A_2 - A_8$ select the block of 4 burst
- $A_0 - A_1$ select the starting column within the bank
- $A_0 \& A_1 = 0$ since addresses are a multiple of 4.
- Generation of lower address bits $A_1 \& A_0$ from bank enable signals

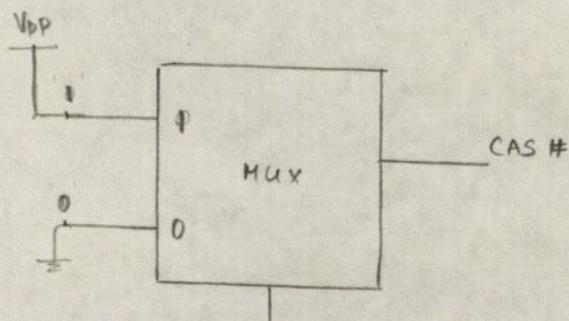


GENERATION OF SDRAM COMMAND AND CKE SIGNALS



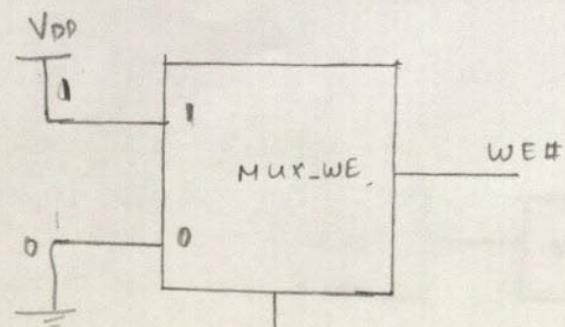
O: STATE == PRECHARGE ALL || STATE == AUTO REFRESH1
 || STATE == AUTO REFRESH2 || STATE == LMR ||
 STATE == ACTIVE

I: STATE != PRECHARGE ALL || STATE != AUTO REFRESH1 ||
 STATE != AUTO REFRESH2 || STATE != LMR ||
 STATE != ACTIVE .



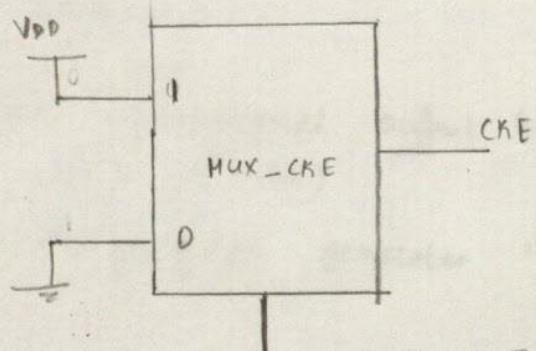
O: STATE == AUTO REFRESH1 || STATE == AUTO REFRESH2
 || STATE == LMR || STATE == READ || STATE == WRITE

I: STATE != AUTO REFRESH1 || STATE != AUTO REFRESH2 ||
 STATE != LMR || STATE != READ



0: STATE == PRECHARGE ALL || STATE == LMR ||
STATE == WRITE

1: STATE != PRECHARGE ALL || STATE != LMR ||
STATE != WRITE

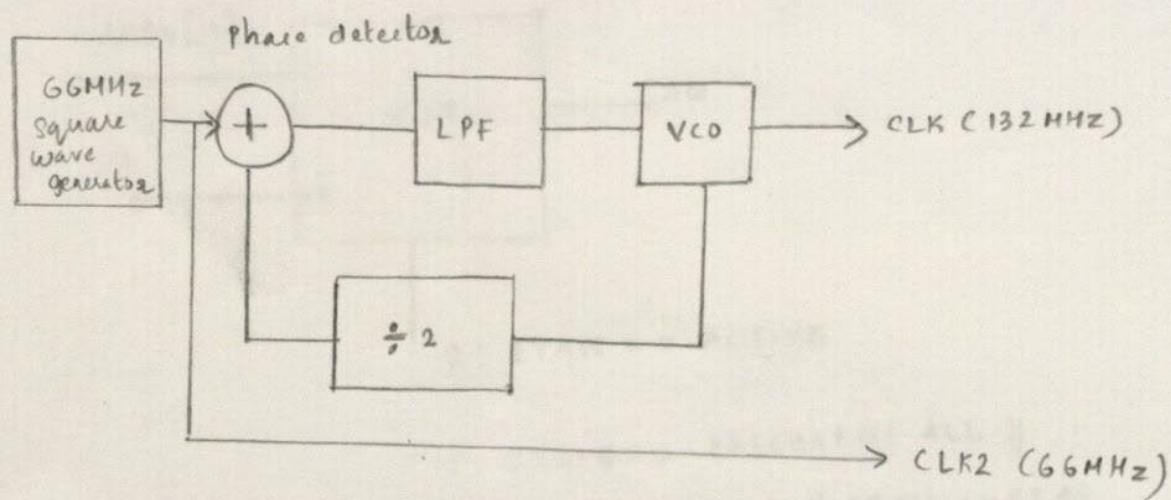


0: STATE == RESET

1: STATE != RESET



MASTER CLOCK GENERATOR

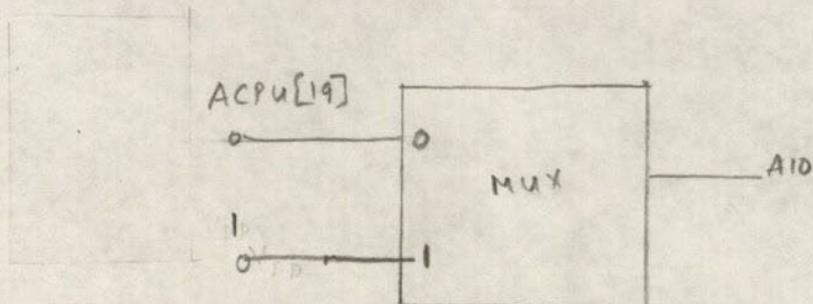


The clock generator design uses a PLL approach which generates a stable 132 MHz clock for SDRAM memory (CLK).

The 66 MHz clock signal output from 80386DX processor, (CLK2)

with the design of PLL generates 132MHz clock for SDRAM.

GENERATION OF A10



0 : STATE == ACTIVE

1 : STATE == PRECHARGE ALL ||

STATE == WRITE || STATE == READ

A10 line carries row addresses when state == ACTIVE.
 whereas A10 = 1 to enable /Auto Precharge during Read or
 write state. During state Precharge All , A10=1 closes out or
 deactivate all banks.

COUNTERS

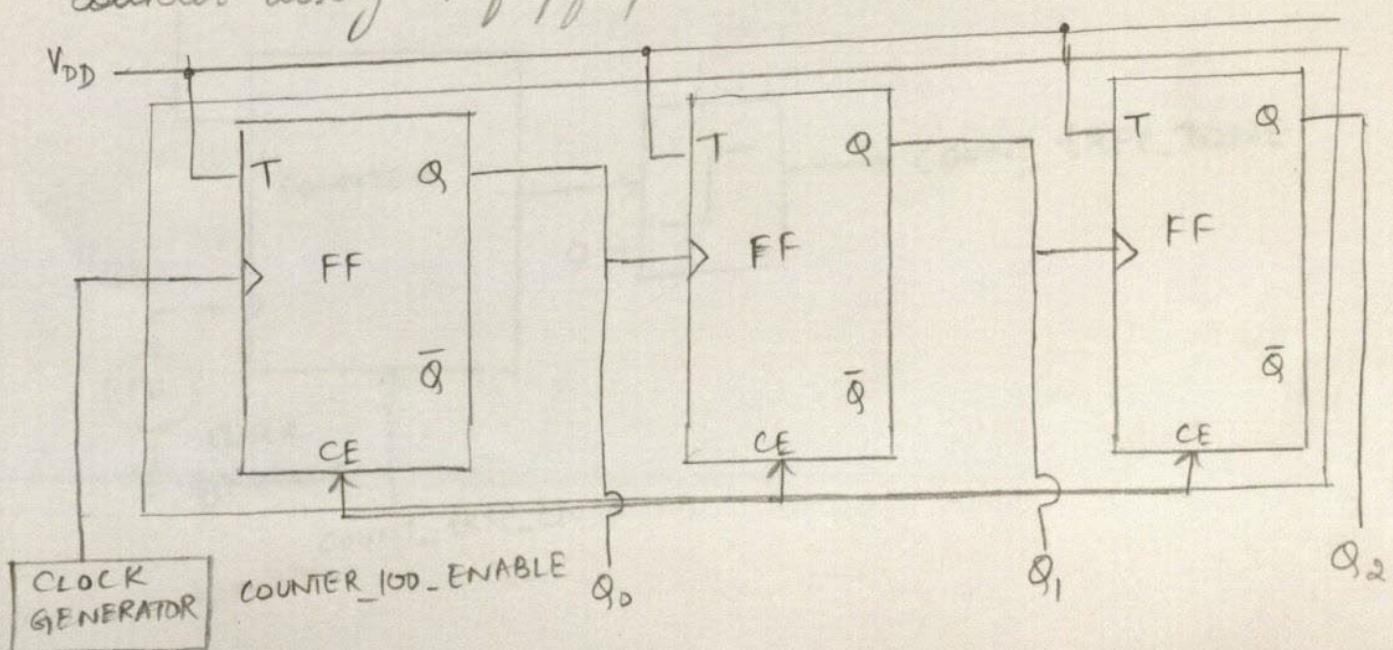
Below is shown the design of a 3 bit DOWN COUNTER. Set the basis of all the counters declared in the FSM.

DESIGN OF COUNTER_TRC :

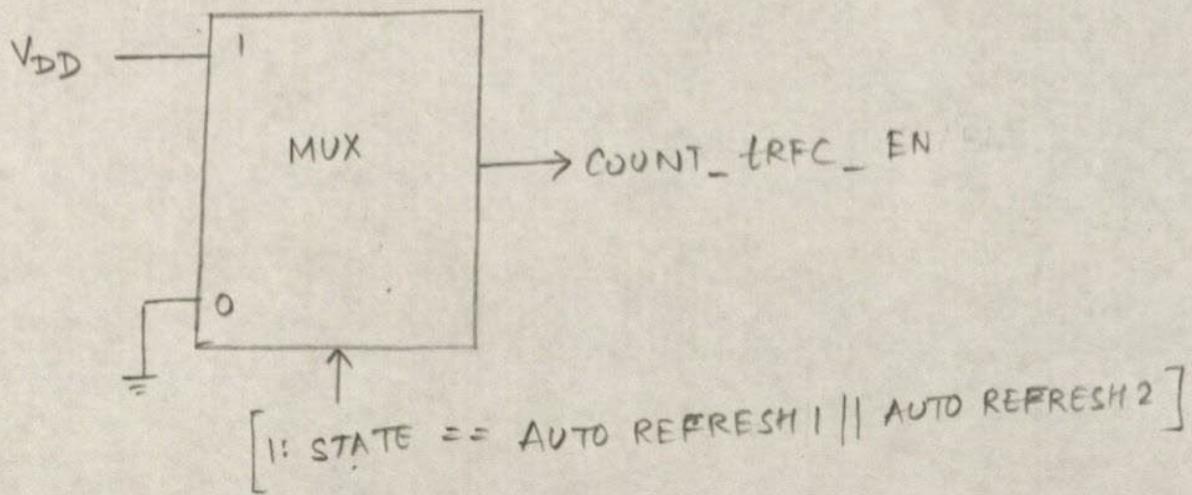
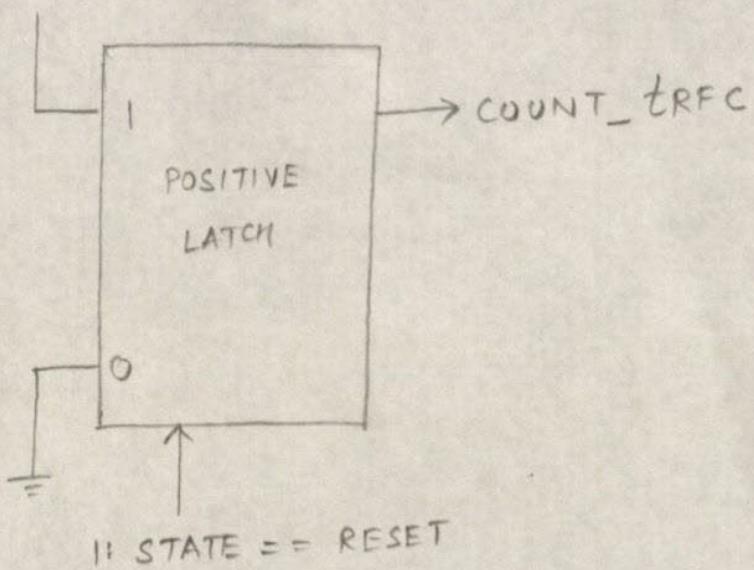
TRUTH TABLE

Q_0	Q_1	Q_2	COUNTER
0	0	0	7
0	0	1	6
0	1	0	5
0	1	1	4
1	0	0	3
1	0	1	2
1	1	0	1
1	1	1	0

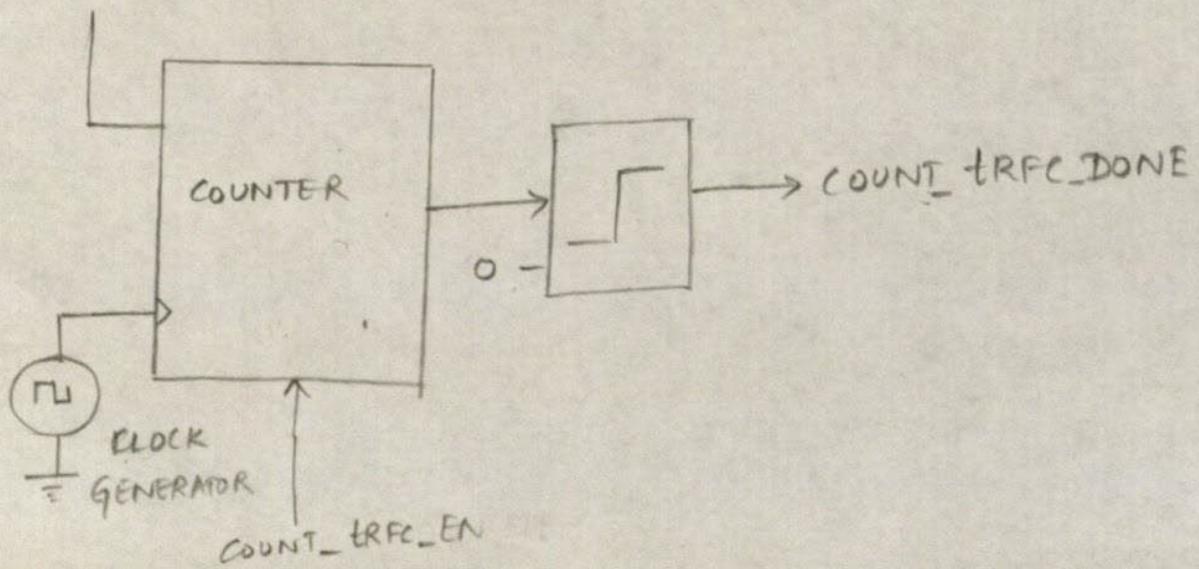
- * The counter counts the pulse for the time period defined as TRC in the READ FSM.
- * TIMER is implemented via a 3 bit asynchronous down counter using T-flipflops

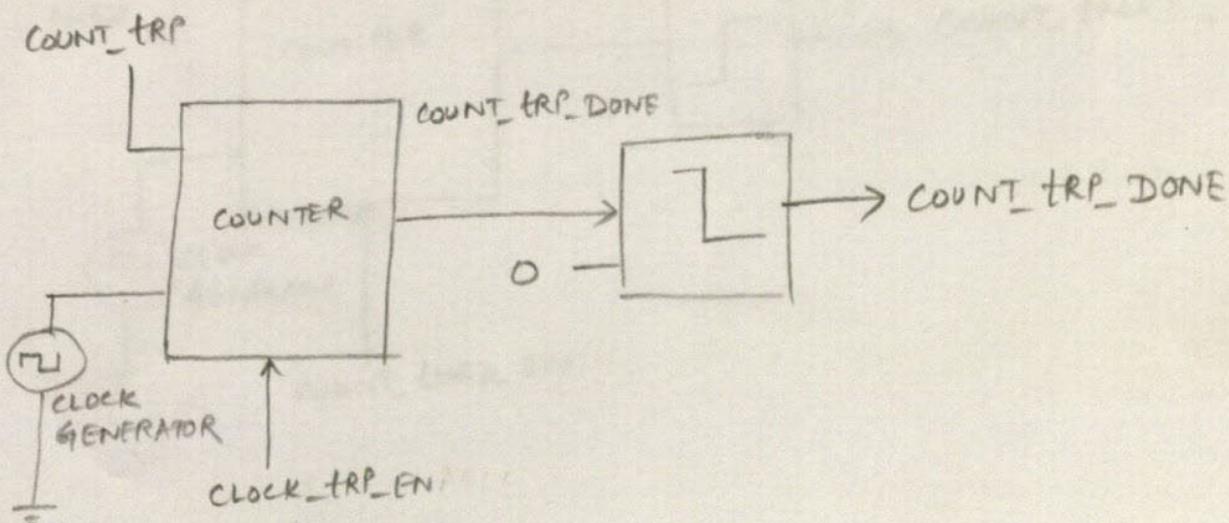
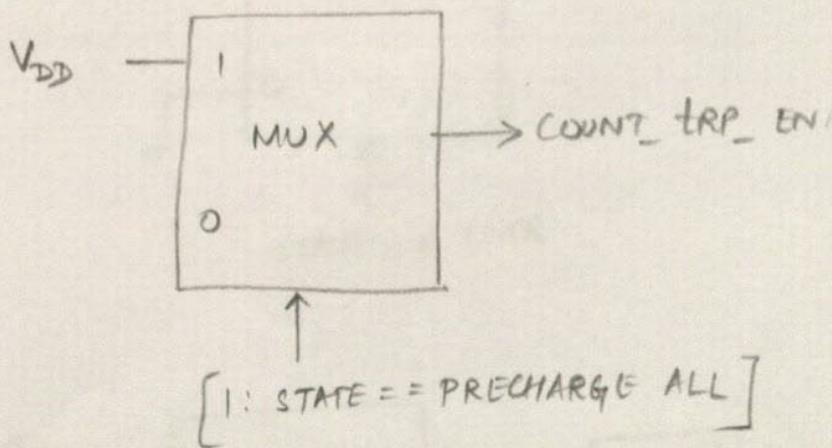
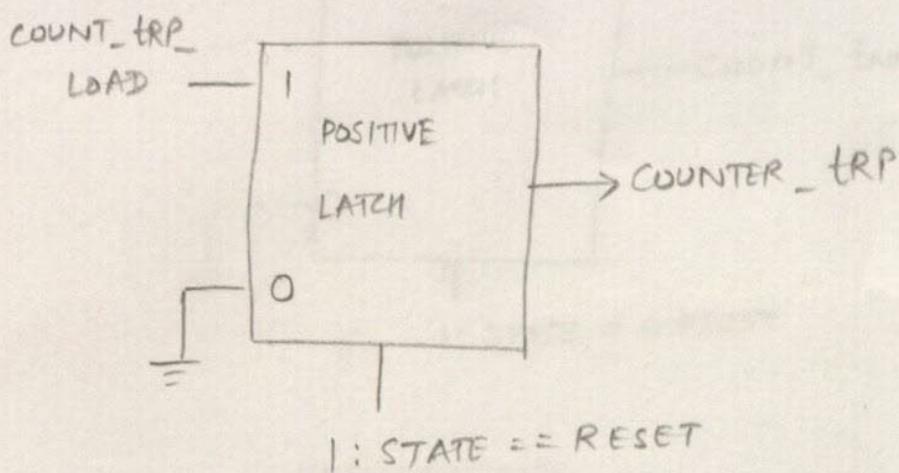


COUNT_TRFC_LOAD

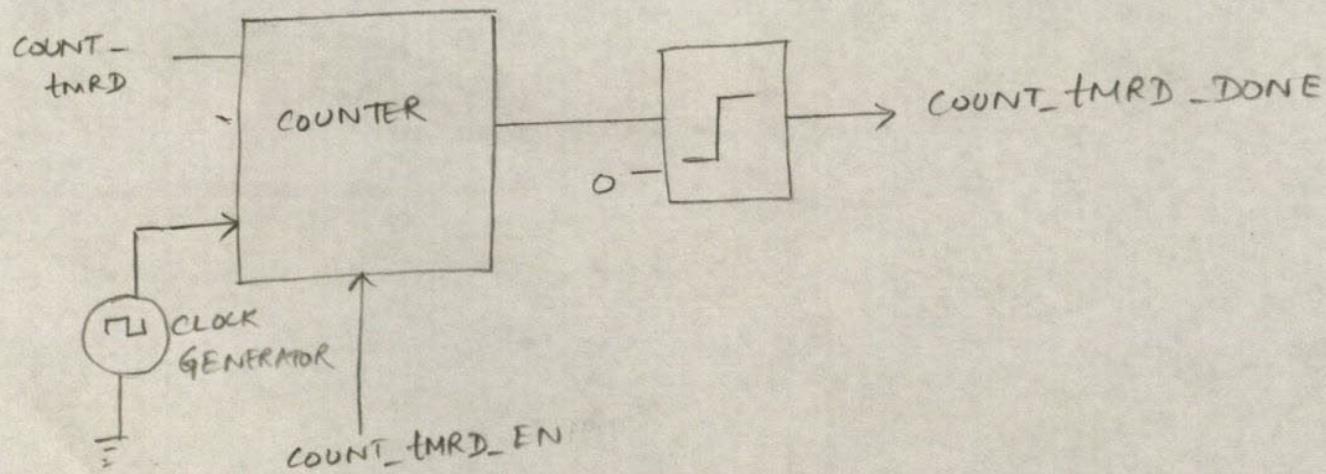
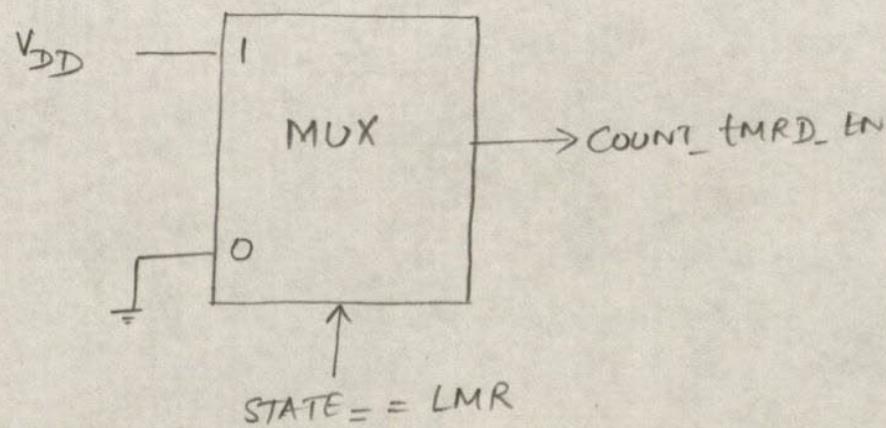
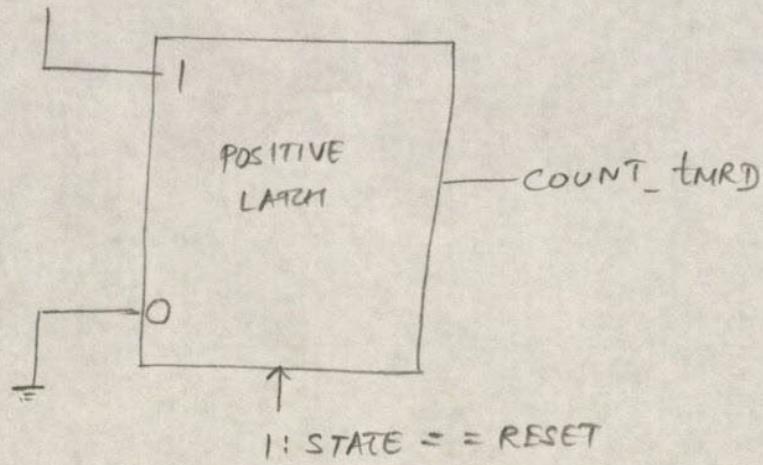


COUNT_TRFC

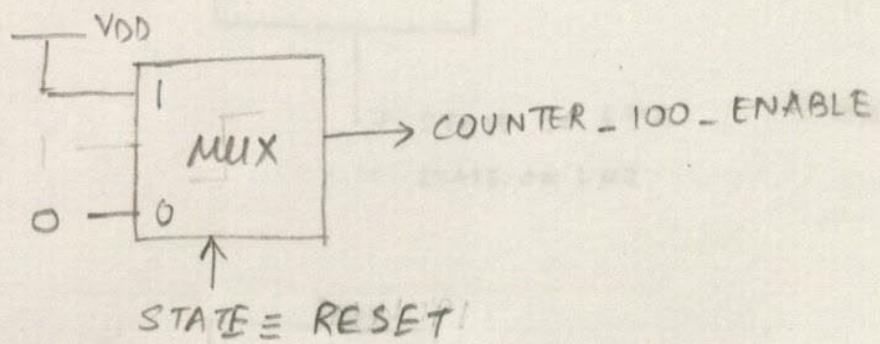
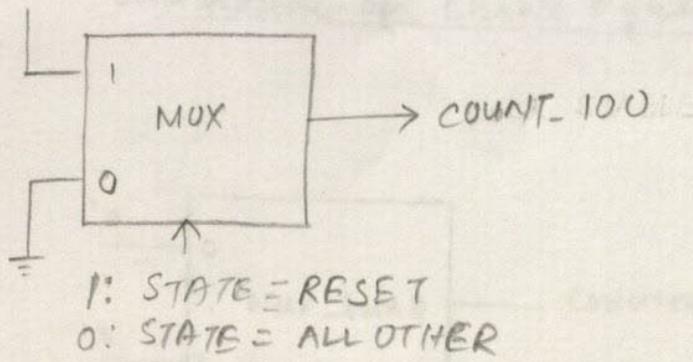




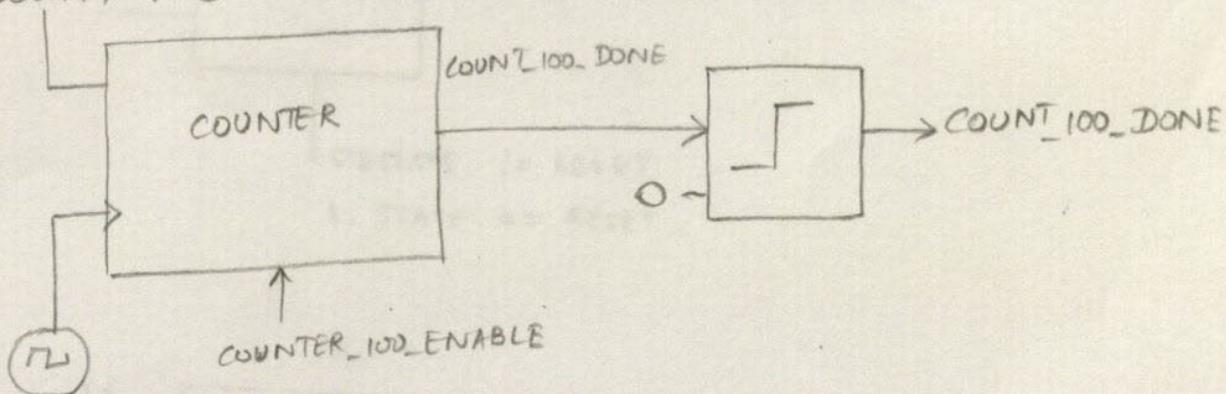
COUNT_tMRD_LOAD



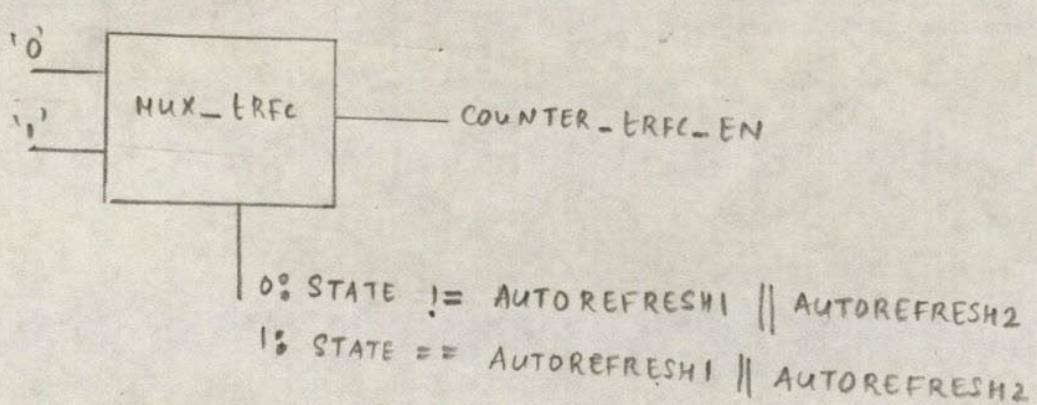
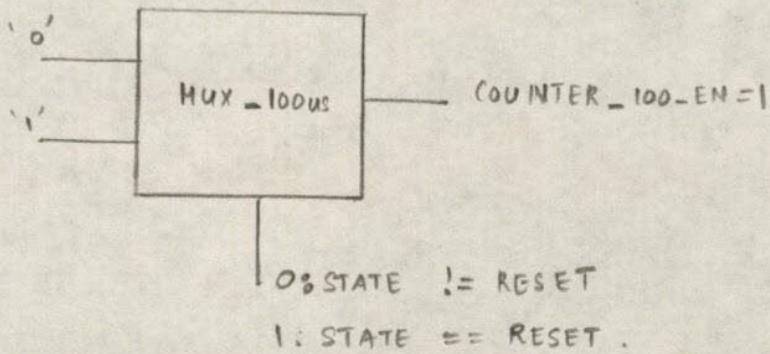
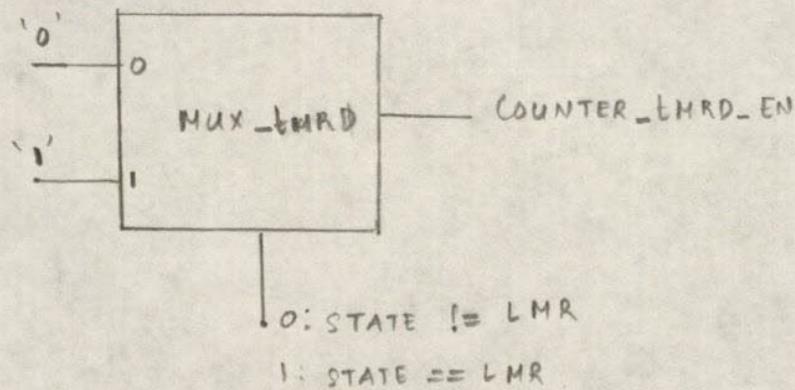
COUNT_100_LOAD

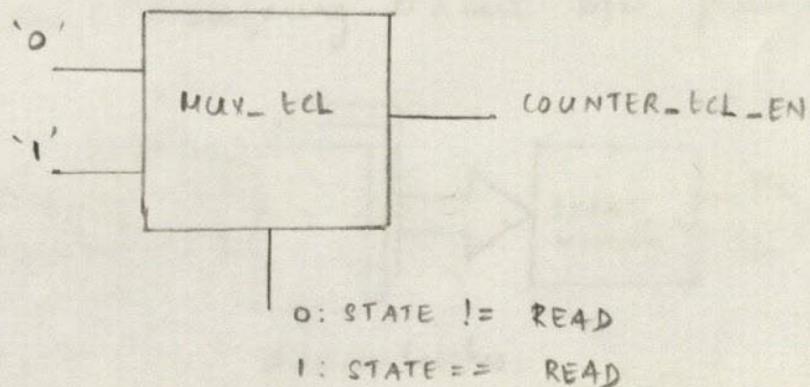
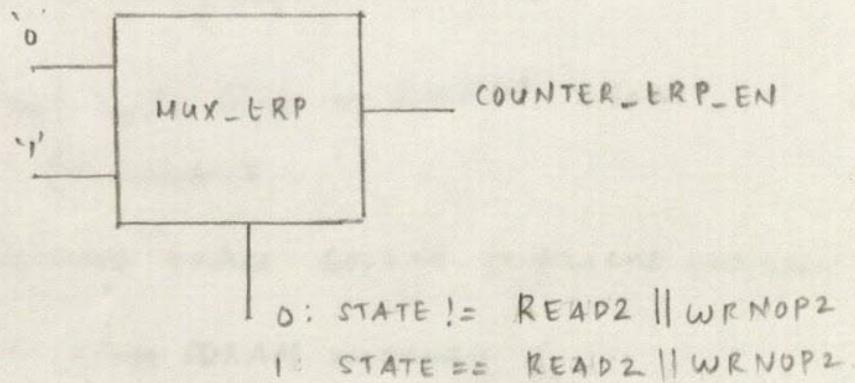
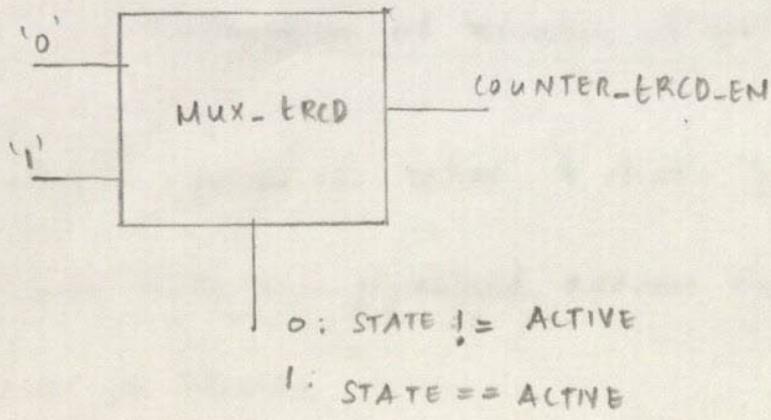


COUNT_100



Generation of enable signals of the internal counters





Estimation of number of wait states

The 80386DX processor takes 4 clocks for the fastest read cycle with non-pipelined address timing.

$$CLK_2 = 66 \text{ MHz}$$

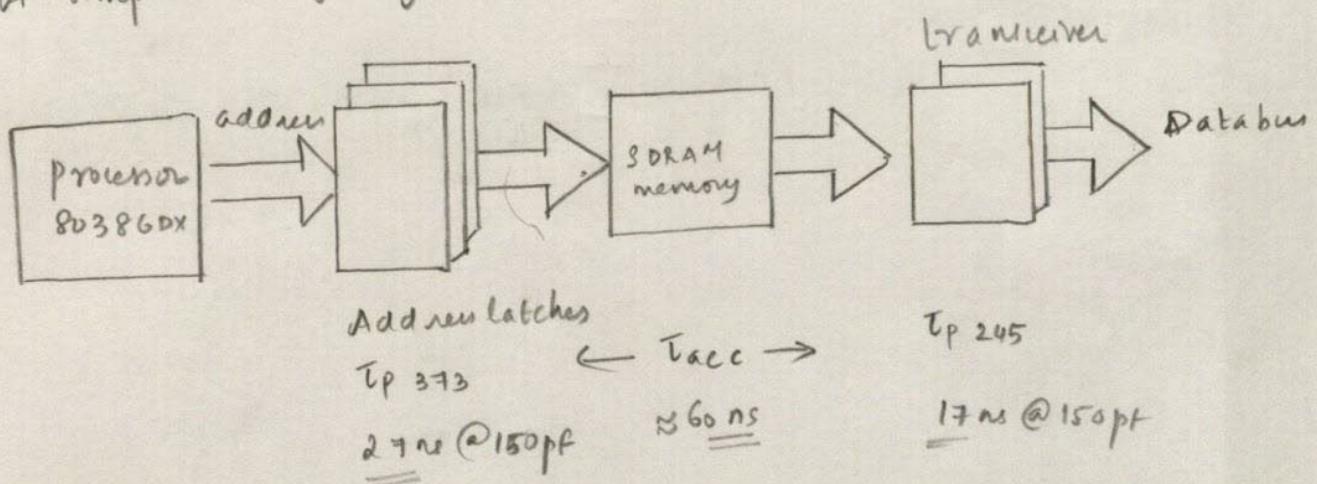
$$CLK_2 \text{ time period} = 15.15 \text{ ns}$$

$$\text{Read cycle time} = 60.6 \text{ ns} \approx 60 \text{ ns}$$

for processor

\therefore Processor takes 60.6 ns to present address and return data from SDRAM memory

A simple interfacing circuit between processor and memory

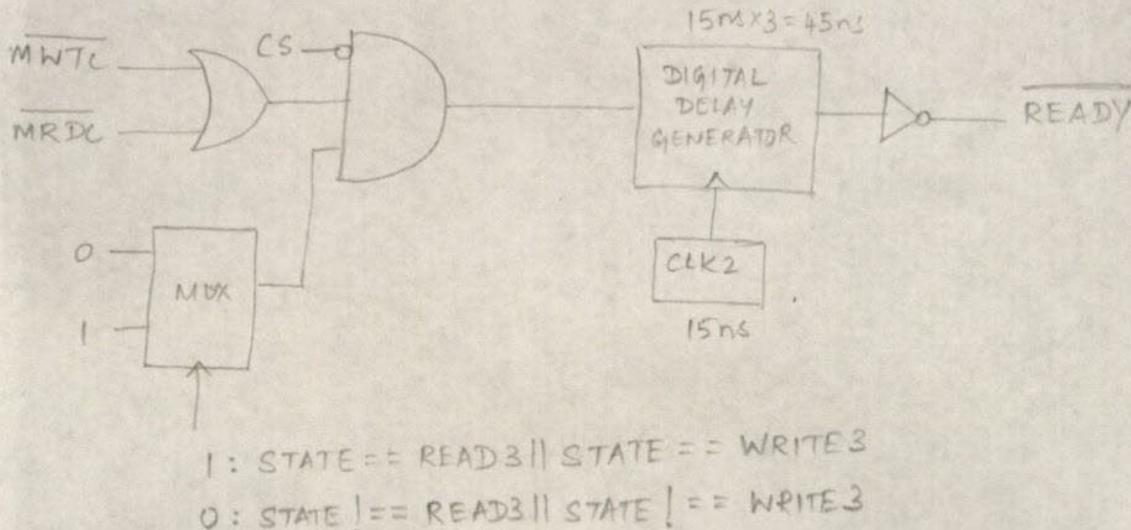


• SDRAM memory access time

SDRAM memory \rightarrow Read cycle of 8 clocks ($CLK = 132 \text{ MHz}$)

\therefore Memory access time $\approx 60 \text{ ns}$

Generation of READY signal



- 1) When MWTC or MRDC are active low, the write or read cycle are respectively generated.
- 2) At the end of the write or read cycle, the MUX outputs a value of 1
- 3) As CS is active low during the FSMs, the AND logic outputs as follows:

MWTC/MRDC	CS	MUX output	Output signal
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
,	1	1	1

- 4) The output is high only when CS=0, MWTC/MRDC=0 and the state is at Read3 or write 3. However, a delay stage of $15\text{ns} \times 3 = 45\text{ns}$ is introduced such that Ready is generated 3 clock cycles later than the memory access time. The assumption is made to accommodate all delays the data encounters on its way from memory to processor. We have made the assumption based on delays encountered when address lines are latched into 74S373. Delays = $27\text{ns} + 17\text{ns} = 44\text{ns} \approx 45\text{ns} \approx 3$ clock cycles.

