



THE UNIVERSITY OF TEXAS AT ARLINGTON

Submitted By:

Srinidhi Kadapanatham: 1002111809  
Kirtan Patel: 1002091306  
Jinit Soni: 1002127453

*Under the Guidance of the instructor:  
Dr. Shahriar Shahabuddin*

## Contents.

1. AIM.....	3
2. COMPONENTS REQUIRED AND SPECIFICATIONS.....	3
3. INTERFACING.....	3
4. CLOCK GENERATION.....	4
5. INTRODUCTION.....	4
6. FUNCTIONAL BLOCK DIAGRAM.....	7
7. IMPLEMENTATION.....	8
8. TOP LEVEL VIEW OF FINITE STATE MACHINE.....	9
9. TOP-LEVEL STATE TRANSITION TABLE.....	10
10. MEMORY ADDRESS MAPPING.....	10
11. INITIALIZATION.....	11
a. STATE MACHINE FOR INITIALIZATION.....	12
b. STATE TRANSITION TABLE FOR INITIALIZATION.....	13
c. TIMING DIAGRAM.....	14
12. LOAD MODE REGISTER.....	15
13. AUTO REFRESH.....	16
a. STATE MACHINE FOR AUTO REFRESH.....	17
b. STATE TRANSITION TABLE FOR AUTO REFERESH.....	18
c. TIMING DIAGRAM.....	19
14. READ.....	20
a. CALCULATIONS FOR READ.....	20
b. STEPS FOR READ.....	21
c. STATE MACHINE FOR READ.....	21
d. STATE TRANSITION TABLE.....	22
e. TIMING DIAGRAM.....	23
15. WRITE.....	24
a. CALCULATION FOR WRITE.....	24
b. STEPS FOR WRITE.....	24
c. STATE MACHINE FOR WRITE.....	24
d. STATE TRANSITION TABLE.....	25
e. TIMING DIAGRAM.....	26
16. SCHEMATIC.....	27

## AIM:

The project aims to design an SDRAM controller that allows SDRAM memory to be interfaced with a microprocessor (80386Dx) having synchronous memory support. This report shows the complete design and theory of operations.

## COMPONENTS REQUIRED AND SPECIFICATIONS:

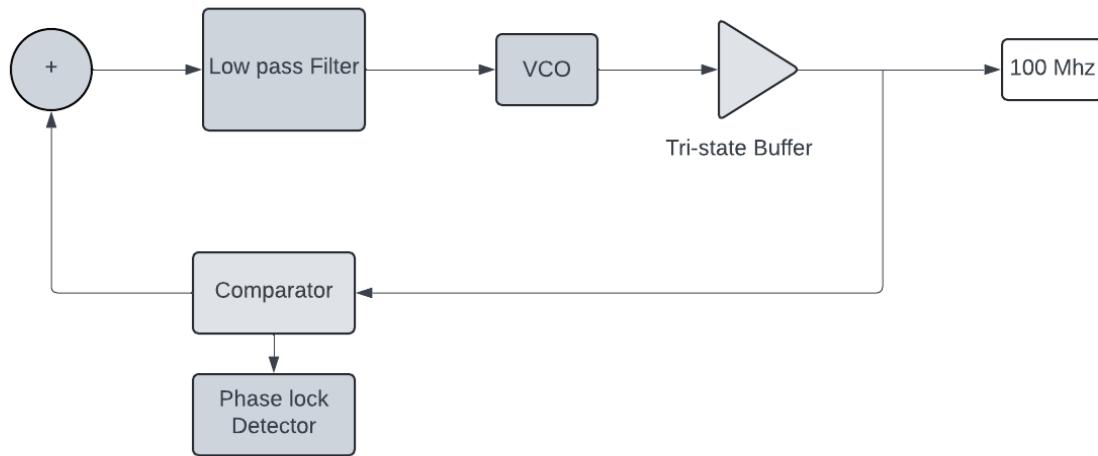
- SDRAM Controller
- SDRAM
  - MT48LC16M4A2 – 4 Meg x 4 x 4 banks 4096 rows x 1024 columns x 4 banks
  - Operating Frequency = 100 MHz
  - Clock cycle time T<sub>clcl</sub>: 10 ns
  - Speed Grade: -75 with access time: 5.4 ns
  - Setup time: 1.5 ns and hold time: 0.8 ns.
  - Burst length: 8
  - CAS Latency = 2
  - Refresh time: 64 ms for 4096 rows
- 80386DX processor
  - 32-bit address line and 32-bit data line.

## INTERFACING:

- Clk (66 MHz) from the microprocessor is given to the controller.
- Clk for the SDRAM is 100 MHz this is attained by passing Clk2 to phase locked loop.
- 8086Dx microprocessor consists of 32-bit address lines in which A0 and A1 act as bank enable signals for BE0 to BE3.
- 32-bit data bus D0-D31 of 8086Dx microprocessor is meant for data transfer and receive from the SDRAM device.
- Ready signal is given to the microprocessor from the controller to indicate the termination of current CPU bus cycle information.
- HOLD is also given to the processor from the controller.
- HLDA is hold acknowledgment is given to controller form 8086Dx processor.
- ADS, RD, and WT signals generate read and write signals for controllers.
- CAS, RAS, and WE are the signals given to the SDRAM for memory read and write operations.
- CKE is a clock-enabled signal used to enable the operations in the SDRAM.
- VDD is supplied to all devices.
- VSS is the ground.

## CLOCK GENERATION:

- The CLK2 of the 8086Dx processor is given as an input clock with a 66MHz clock frequency.
- To convert the Clk frequency from 66 MHz to 100 MHz we are using Phase Locked Loop (PLL).
- PLL consists of the lowpass filter, Voltage controlled Oscillator (VOC), Phase Lock Detector, Tri-State Buffer, and divide-by-phase comparators.
- This is used to generate an output signal related to an input signal's phase.
- The 66 MHz from the processor is sent to the PLL unit to obtain the desired frequency of 100 MHz.
- When a 100 MHz frequency is attained, it is given as an input to the controller and SDRAM.



## INTRODUCTION:

### SDRAM Overview

Synchronous Dynamic Random-Access Memory (SDRAM) is a type of DRAM that operates in sync with the microprocessor's clock speed. It's a popular choice in contemporary computer setups due to its efficient bandwidth and energy-saving features. To work with a microprocessor using a 32-bit asynchronous memory data interface, SDRAMs typically need a specialized controller.

## SDRAM Memory Details

The MT48LC16M4A2 is a 64MB SDRAM module with a 16-bit data bus and an internal structure featuring 4 banks. Each bank comprises 4,096 rows and 256 columns. This memory operates at clock speeds of up to 133 MHz.

## Microprocessor Overview

The 80386DX is a 32-bit microprocessor with a 32-bit data bus. However, it doesn't support SDRAM due to its asynchronous memory data interface. Thus, to connect the 80386DX to SDRAM, an SDRAM controller must be developed.

## SDRAM Controller

An SDRAM controller serves as the intermediary between a microprocessor and SDRAM modules. It's responsible for generating the necessary signals for reading from or writing to the SDRAM, as well as managing tasks like addressing rows and columns, refreshing the memory, and transferring data. The controller incorporates a state machine.

- ***State Machine:*** This component governs the data exchange between the microprocessor and SDRAM. The state machine operates through various states to manage this data flow effectively.
- **Initial State:** This serves as the starting point for the SDRAM controller. Upon system startup, the state machine enters this phase, resetting all internal registers to establish a consistent state.
- ***Configuration State:*** Following the initial state, the controller progresses to the configuration state. Here, it sets up the necessary control signals and registers to interact effectively with the SDRAM. This setup encompasses defining the memory's bank count, rows, columns, refresh intervals, and clock rates.
- ***Standby State:*** After configuration, the controller moves to the standby state. This state is adopted when the SDRAM is not engaged in any active operations. The controller remains in this state until prompted for a read or write task.
- ***No-Operation (NOP) State:*** A variant of the standby state, the NOP state involves the controller pausing operations without executing any tasks. It's employed to introduce delays or pauses between operations. During this phase, all inputs to the SDRAM are disregarded, maintaining the memory's current status.
- ***Reading State:*** Upon receiving a read command, the controller shifts to the reading state. Here, it dispatches the requisite signals to the SDRAM to kickstart the read process. The controller then awaits the data retrieval from the SDRAM and forwards it to the processor once available.
- ***Writing State:*** The controller switches to the writing state for a write command. Here, it transmits the required signals to the SDRAM to initiate the writing process. The data intended for writing is also sent to the SDRAM. Once the write operation concludes, the controller reverts to the standby state.

## Microprocessor Processor

- ***Auto Refresh State:*** During normal operation, the SDRAM controller periodically sends an auto-refresh command to the SDRAM.

Auto-refresh Operation: To preserve data integrity, the controller switches to an auto-refresh mode when required. During this mode, it dispatches the appropriate signals to the SDRAM to trigger the auto-refresh process. Once the task is done, the controller reverts to the standby state.

Row Addressing: The state machine handles the generation of row addresses and activates the RAS signal accordingly.

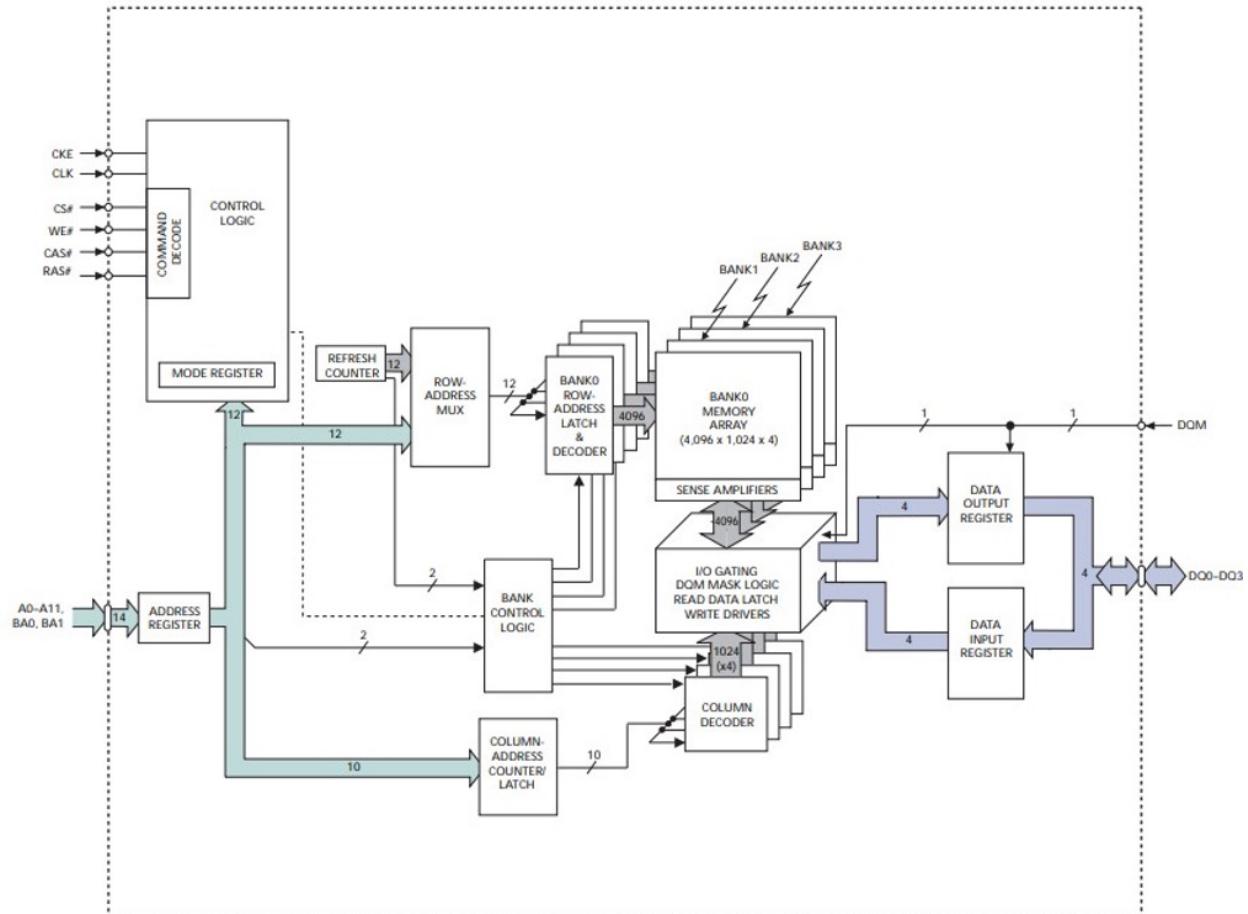
Column Addressing: The state machine manages the creation of column addresses and triggers the CAS signal.

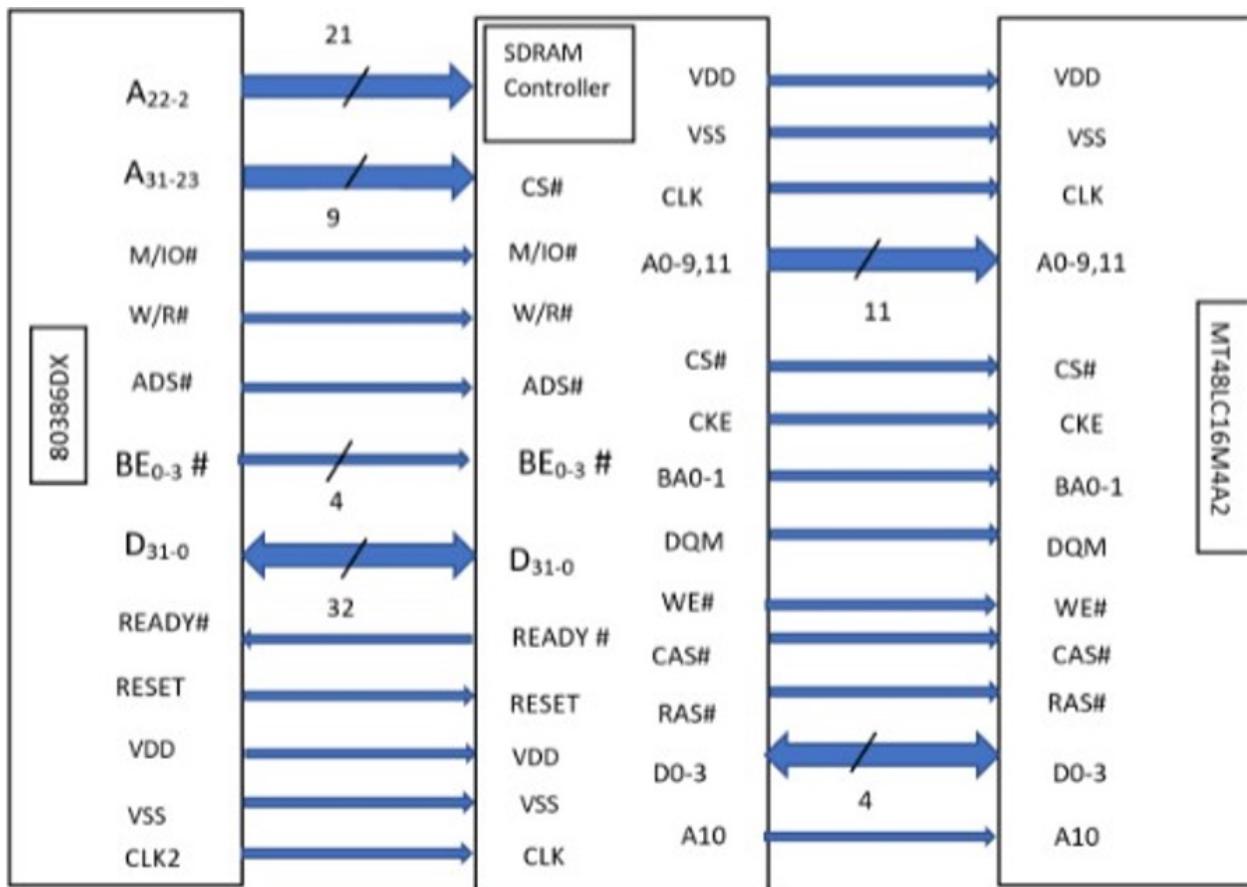
Data Transmission: The state machine oversees the data exchange between the microprocessor and the SDRAM.

Refreshing: The state machine issues an auto-refresh instruction to the SDRAM to maintain its operational state.

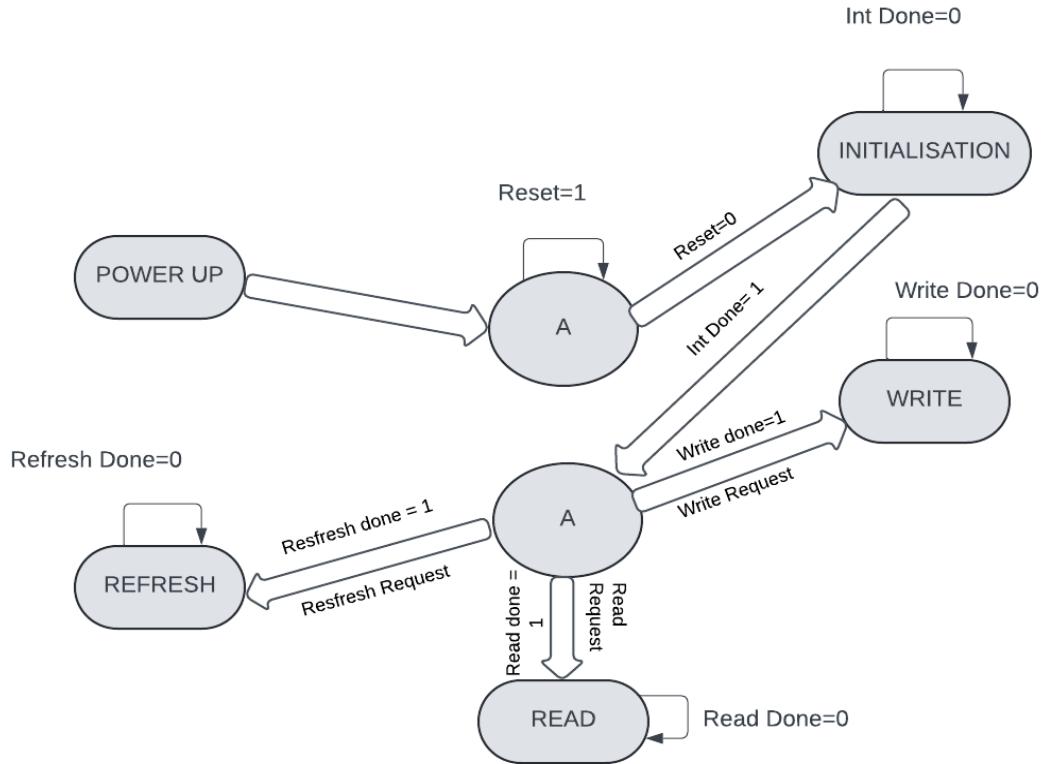
## FUNCTIONAL BLOCK DIAGRAM:

Figure 1: 16 Meg x 4 SDRAM



**IMPLEMENTATION:**

## TOP LEVEL VIEW OF FINITE STATE MACHINE



The SDRAM Controller FSM consists of main state diagram. It has 4 sub-state machines INITIALIZATION, READ, WRITE and REFRESH.

### TOP-LEVEL STATE TRANSITION TABLE:

Current State	Conditions	Next State
Power Up	Null	Reset
Reset	Reset# = 1	Reset
Reset	Reset# = 0	Initialization
Initialization	Int_Done = 0	Initialization
Initialization	Int_Done = 1	Idle
Idle	Read_Request	Read
Read	Read_Done = 0	Read
Read	Read_Done = 1	Idle
Idle	Write_Request	Write
Write	Write_Done = 0	Write
Write	Write_Done = 1	Idle
Idle	Refresh_Request	Refresh
Refresh	Refresh_Done = 0	Refresh
Refresh	Refresh_Done = 1	Idle

### MEMORY ADDRESS MAPPING:

DCD_SDRAM	BA [1:0]	ROW	COL
A31-23	A22-21	A20-9	A8-2
9bits	2bits	12bits	7bits

## INITIALIZATION:

When initializing SDRAMs upon powering up, it's crucial to adhere to the recommended procedures to prevent unexpected outcomes. Once power is applied and a stable clock signal is confirmed, a minimum delay of 100 microseconds is advised before executing any command aside from NOP. During this delay, NOP commands should be consistently applied. After this delay and at least one NOP command, a PRECHARGE command can be introduced.

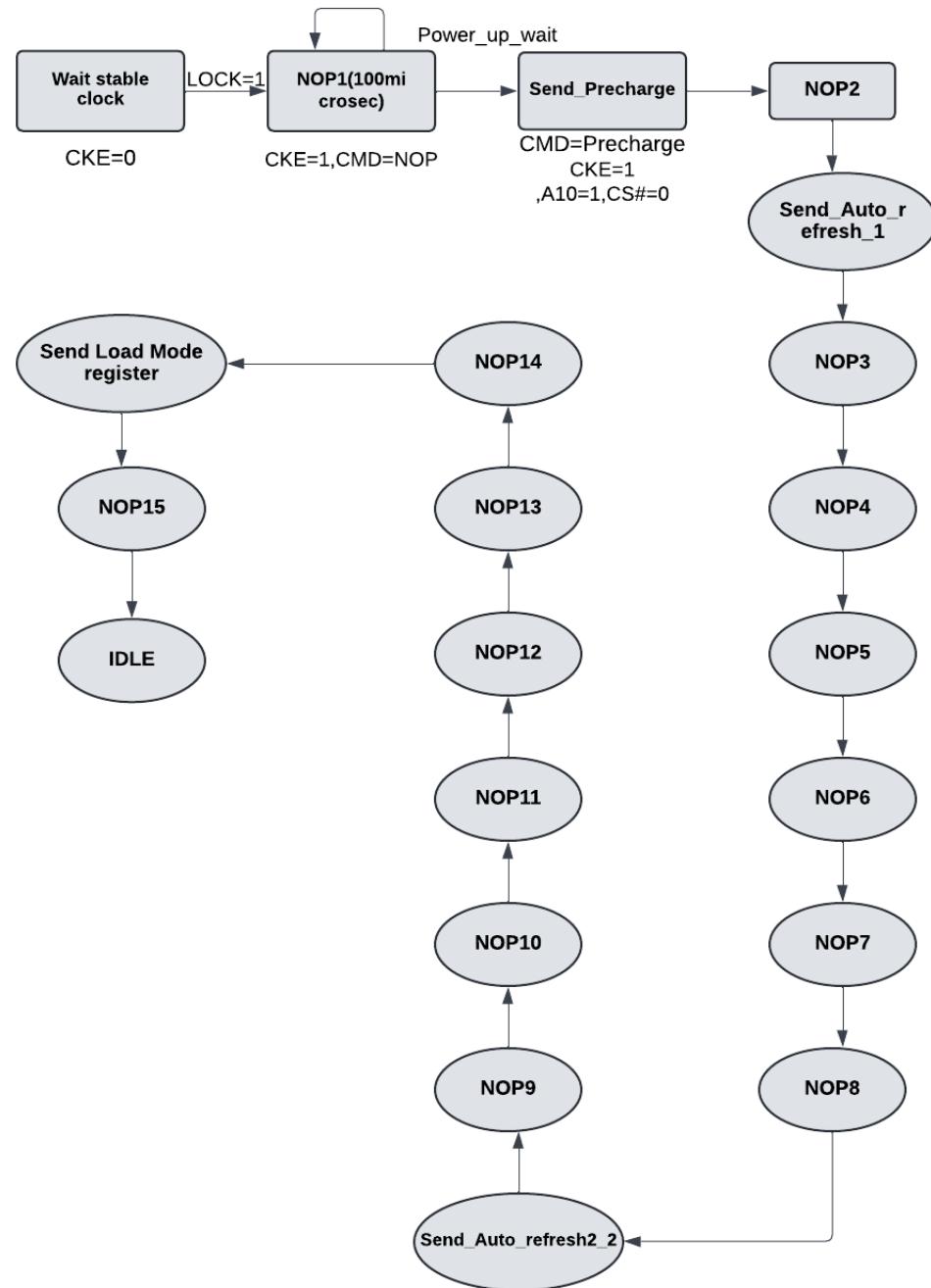
Following the PRECHARGE command, it's essential to await a duration of at least tRP. Throughout this waiting period, an NOP2 command is recommended to ensure all memory banks complete their precharge tasks. Once charging is finalized, the SDRAM should progress to the auto-refresh command phase. Here, a wait of at least tRFC time is needed, equivalent to 7 clock cycles, accompanied by NOP3 through NOP8 commands.

After this interval, a minimum of two auto-refresh cycles should be conducted. During this process, commands ranging from NOP3 to NOP14 should be implemented.

Once the auto-refresh cycles are completed, the SDRAM transitions to mode register configuration. Given that the mode register initializes in an undetermined state, it's imperative to set the desired bit configurations before any operational command. This can be achieved using the LOAD MODE REGISTER command.

Following mode register configuration, it's essential to observe a delay of at least tMRD time. Given tMRD is equivalent to 2 clock cycles, a NOP15 command should be applied during this period. After this wait, the SDRAM is primed to accept any valid command.

## STATE MACHINE FOR INITIALIZATION:

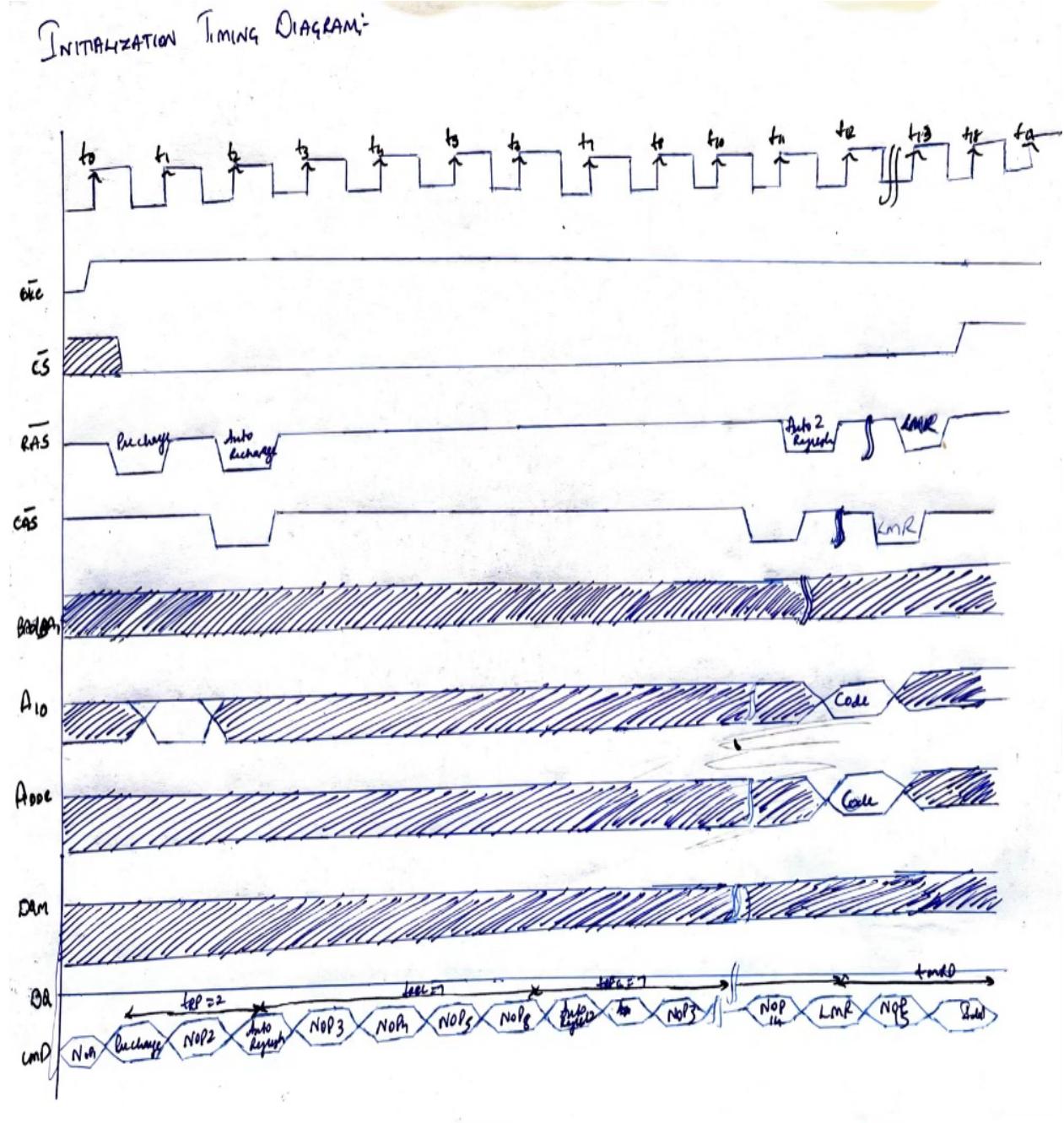


**STATE TRANSITION TABLE FOR INITIALIZATION:**

Current State	Condition	Next State
<b>NA</b>	Reset#=0	Reset
<b>Reset</b>	Next Clock, CKE=0	Wait Stable Clock
<b>Wait Stable Clock</b>	Lock=1	NOP1(100microsec)
<b>NOP1</b>	Power_up_wait!=0	NOP1
<b>NOP1</b>	Power_up_wait=0, CKE=1	Send Precharge
<b>Send Precharge</b>	Null	NOP2
<b>NOP2</b>	Null	Send_Auto refresh 1
<b>Send-Auto refresh 1</b>	Null	NOP3
<b>NOP3</b>	Null	NOP4
<b>NOP4</b>	Null	NOP5
<b>NOP5</b>	Null	NOP6
<b>NOP6</b>	Null	NOP7
<b>NOP8</b>	Null	Send_Auto refresh2
<b>Send-Auto refresh2</b>	Null	NOP9
<b>NOP9</b>	Null	NOP10
<b>NOP10</b>	Null	NOP11
<b>NOP11</b>	Null	NOP12
<b>NOP12</b>	Null	NOP13
<b>NOP13</b>	Null	NOP14
<b>NOP14</b>	Null	Send Load Mode register
<b>Send Load Mode register</b>	Null	NOP15
<b>NOP15</b>	Null	IDLE

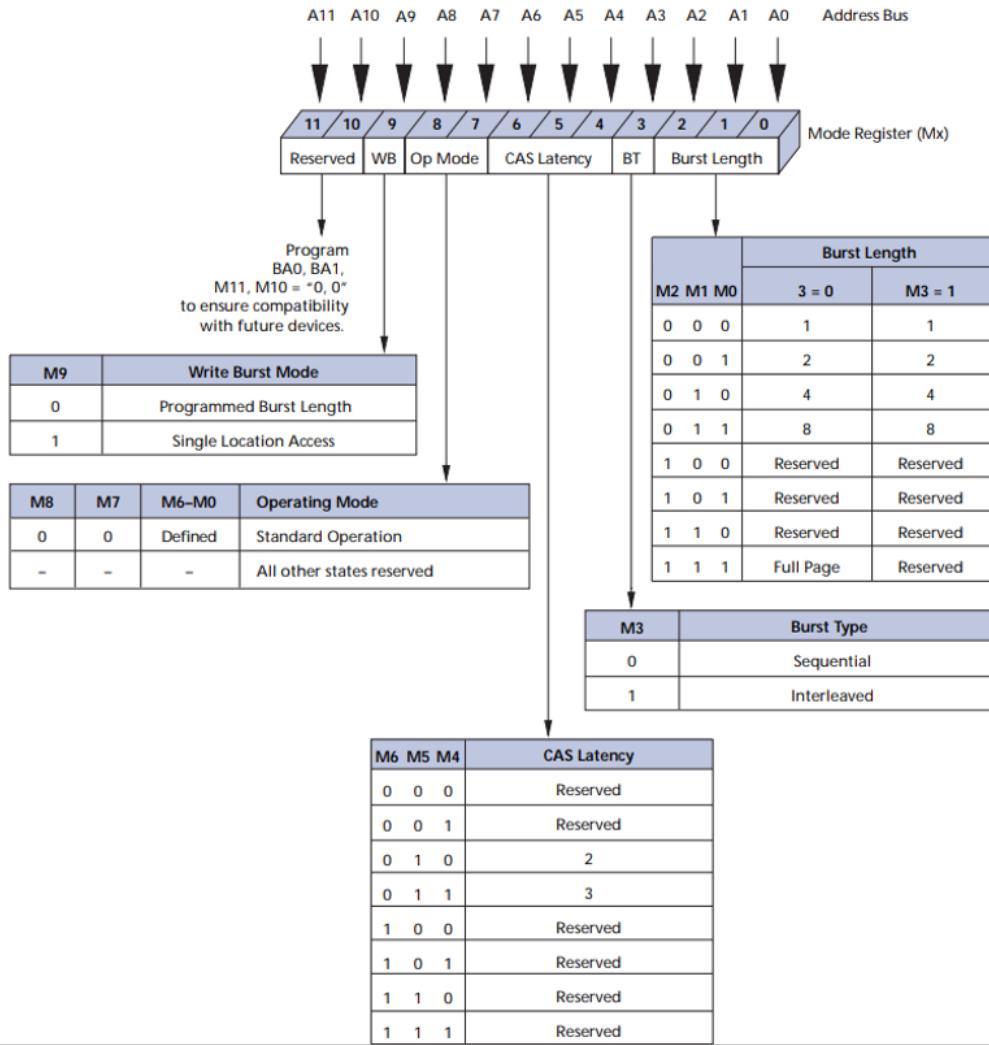
Clock Calculations	Value(ns)	No. of clocks
<b>NOP1</b>	100000ns	83,000
<b>Precharge Period (Trp)</b>	20ns	2
<b>Auto Refresh Period (Trfc)</b>	66ns	7
<b>Load Mode Register Period (Tmrdrd)</b>	20ns	2

**TIMING DIAGRAM:**



## LOAD MODE REGISTER:

Figure 6: Mode Register Definition



According to our specifications, LMR for initialization is:

Function	Function Value	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
Burst length (BL)	8										0	1	1
Burst Type	Sequential									0			
CASLatency	2						0	1	0				
Op mode	Standard operation				0	0							
WriteBurstMode	Programmed Burst Mode			0									
Reserved	--	0	0										

LMR: 00 0 00 010 0 011

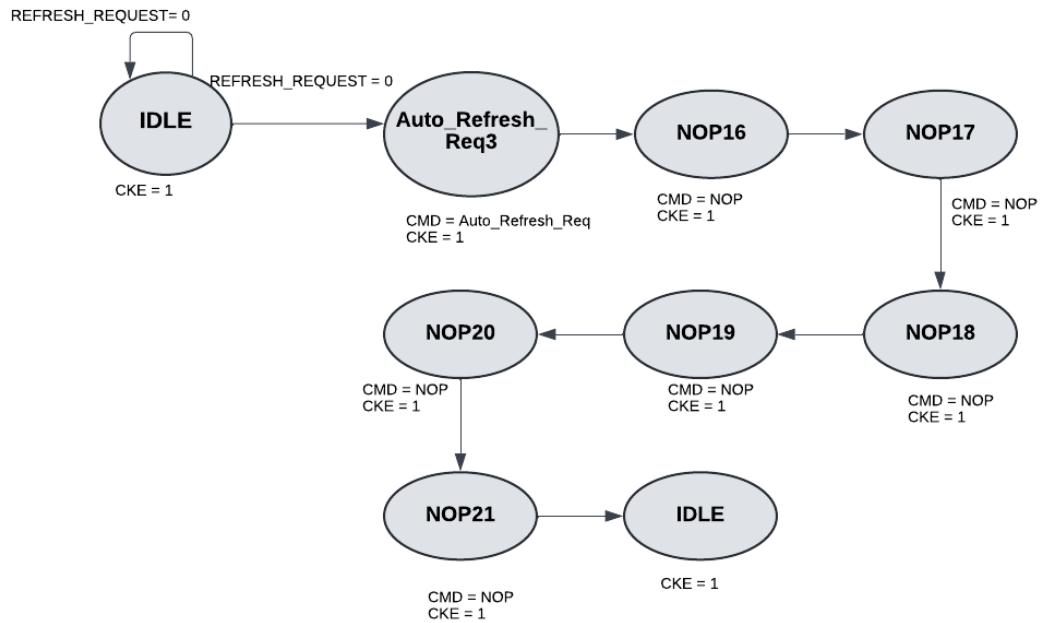
## AUTO REFRESH:

In the SDRAM's regular operation, the AUTO REFRESH command serves to rejuvenate the memory, resembling the CAS#-BEFORE-RAS# (CBR) refresh in conventional DRAMs. Unlike some commands, AUTO REFRESH isn't stored persistently in memory and needs to be activated each time a memory refresh is due. Before triggering an AUTO REFRESH command, it's crucial to PRECHARGE all active banks.

Following the PRECHARGE command, a waiting period of at least the specified tRP time is necessary before initiating an AUTO REFRESH command, as outlined in the Operational guidelines.

During the AUTO REFRESH operation, the internal refresh mechanism handles the addressing, rendering the address bits irrelevant or "Don't Care". For a 64Mb SDRAM, irrespective of its data width, conducting 4,096 AUTO REFRESH cycles every 64ms is imperative. To satisfy this refresh frequency and guarantee each memory row's refresh, an intermittent AUTO REFRESH command can be dispatched approximately every 15.625µs, signaled by the REFRESH\_REQUEST# command. When the REFRESH\_REQUEST# signal drops, the SDRAM transitions to the Auto\_Refresh\_Req mode. In this mode, the SDRAM undergoes NOP16-22 cycles, spanning the tRFC time equivalent to 7 clocks. After this designated period, the SDRAM reverts to its standby or idle mode.

## STATE MACHINE FOR AUTO REFRESH:

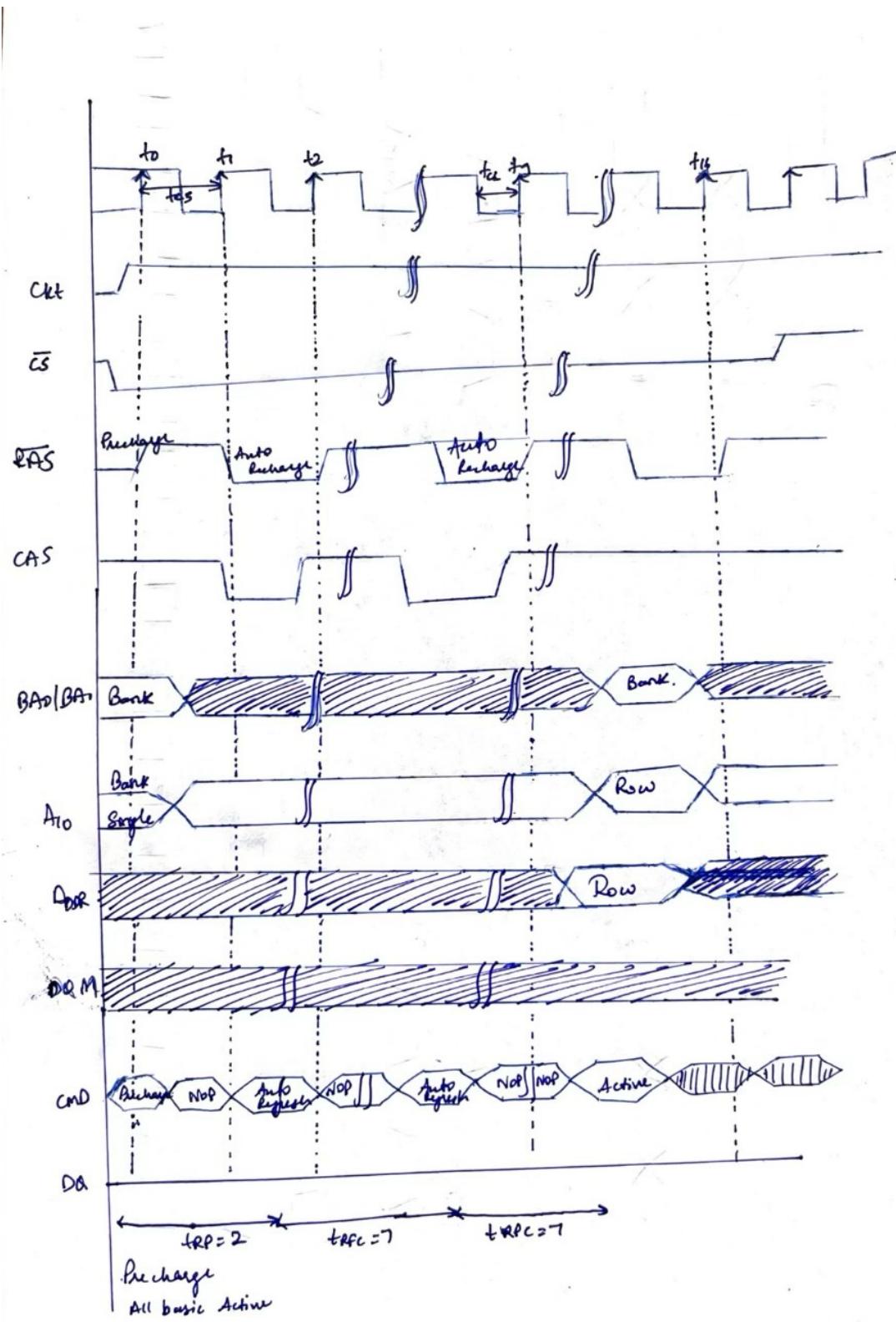


### STATE TRANSITION TABLE FOR AUTO REFRESH:

Current State	Condition	Next State
IDLE	Refresh_Request! = 0	IDLE
IDLE	Refresh_Request = 0	Auto_Refresh_Req
Auto_Refresh_Req	Null	NOP16
NOP16	Null	NOP17
NOP17	Null	NOP18
NOP18	Null	NOP19
NOP19	Null	NOP20
NOP20	Null	NOP21
NOP21	Null	IDLE

Clock Calculations	Value(ns)	No. of clocks
Precharge Period (Trp)	20ns	2 Clocks
Auto Refresh Period (Trfc)	66ns	7 Clocks

**TIMING DIAGRAM:**



**READ:****CALCULATIONS FOR READ:**

Read cycle calculation

$$\text{Time period} = 1 \text{ second}$$

$$\text{Clock frequency} = 100 \text{ MHz}$$

$$\text{Clock Variant} = -75$$

$$T_{\text{clk}} = \frac{1}{100 \times 10^6} = 10 \text{ ns}$$

- Auto refresh period for -75 =  $66 \text{ ns}$

- Rows required to refresh =  $4096 \text{ ie } 2^{12}$

- Refresh time for all rows at once =  $4096 \times 66 \text{ ns}$   
 $= 270.336 \mu\text{s}$

→ Refresh period of 4096 rows in -75 is  $64 \text{ ms}$

$$\text{All rows to be refreshed in } 1 \text{ sec} = \frac{1}{64 \times 10^3} = 15.625 = 16 \text{ times}$$

$$\rightarrow \text{Time consumed for refresh in 1sec} = \frac{1}{64 \times 10^3} = 15 \times 270.336 \mu\text{s} \\ = 4.325 \text{ ms}$$

→ Time remaining for read cycle = 1 - Time consumed for refresh

$$1 - 4.325 \times 10^{-3} \text{ s} \\ = 0.996 \text{ sec}$$

→  $t_{RCD}$  (Row to Column Delay) for -75 =  $2 \text{ ns}$

$$\text{No of clocks required} = \frac{t_{RCD}}{T_{\text{clk}}} = \frac{20}{10} \text{ ns} = 2 \text{ clocks.}$$

Considering  $Cl = 2$   $t_{BL} = 8$

$$\text{No of clocks required for read cycle} = t_{RCD} + Cl + t_{BL} \\ = 2 + 2 + 8 \\ = 12 \text{ clocks //}$$

$$T_{RC} = 12 \times T_{\text{clk}} = 12 \times 10 = 120 \text{ ns}$$

$$\rightarrow \text{No of read cycles} = \frac{\text{Time remaining for read cycle}}{T_{RC}} = \frac{996 \text{ ms}}{120 \text{ ns}} \\ = 8.3 \times 10^6 = 8300000 \text{ cycles}$$

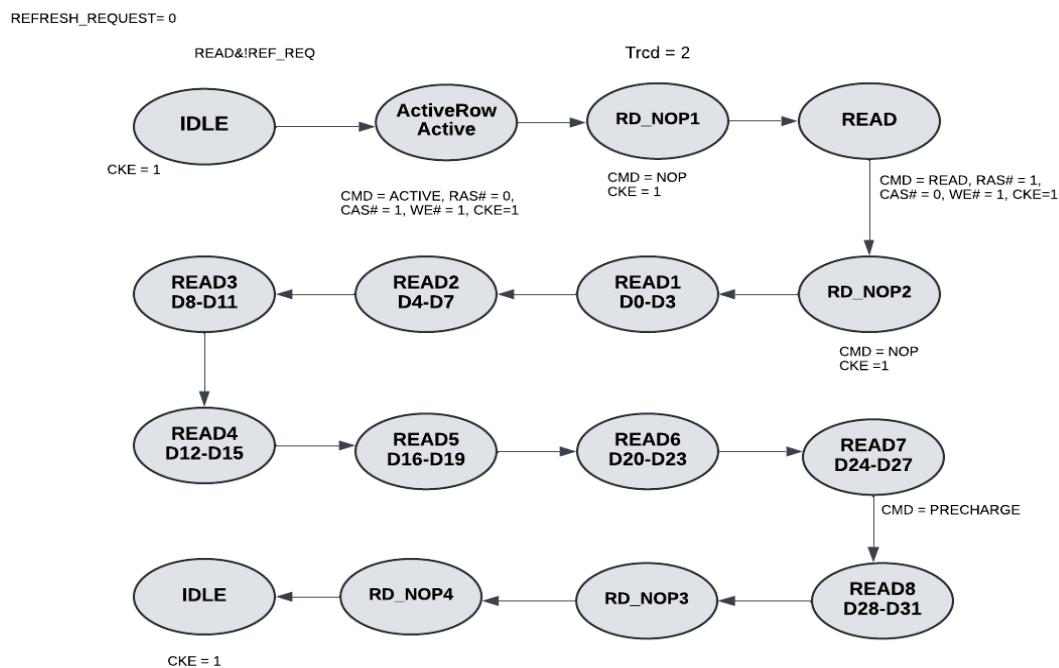
## STEPS FOR READ:

The READ command enables the retrieval of data from an active row within the SDRAM. When the READ&! REF\_REQ signal drops to a low state, the SDRAM transitions to the Active mode. During this transition, the row and bank become active by configuring RAS#=0, CAS#=1, and WE#=1. Subsequent to this, it enters the RD\_NOP1 phase before advancing to the READ phase.

In the READ phase, both the column address and the input value A10 are specified. A10's value determines the activation of auto precharge; for this operation, A10 is set high, indicating the use of auto precharge. Following the READ phase, the SDRAM awaits a duration based on the CAS Latency setting, which is set to 2 in this scenario. This leads the SDRAM to the RD\_NOP2 phase, transitioning to READ1 in the subsequent cycle. This sequence iterates through READ8, with each iteration fetching 4 bits of data, culminating in a total of 32 bits of data retrieved.

Upon completion of the READ process, the READY signal is triggered. Given that the SDRAM clock operates at double the speed of the microprocessor clock, a minimum of two wait cycles is essential to ensure synchronization between the two clocks.

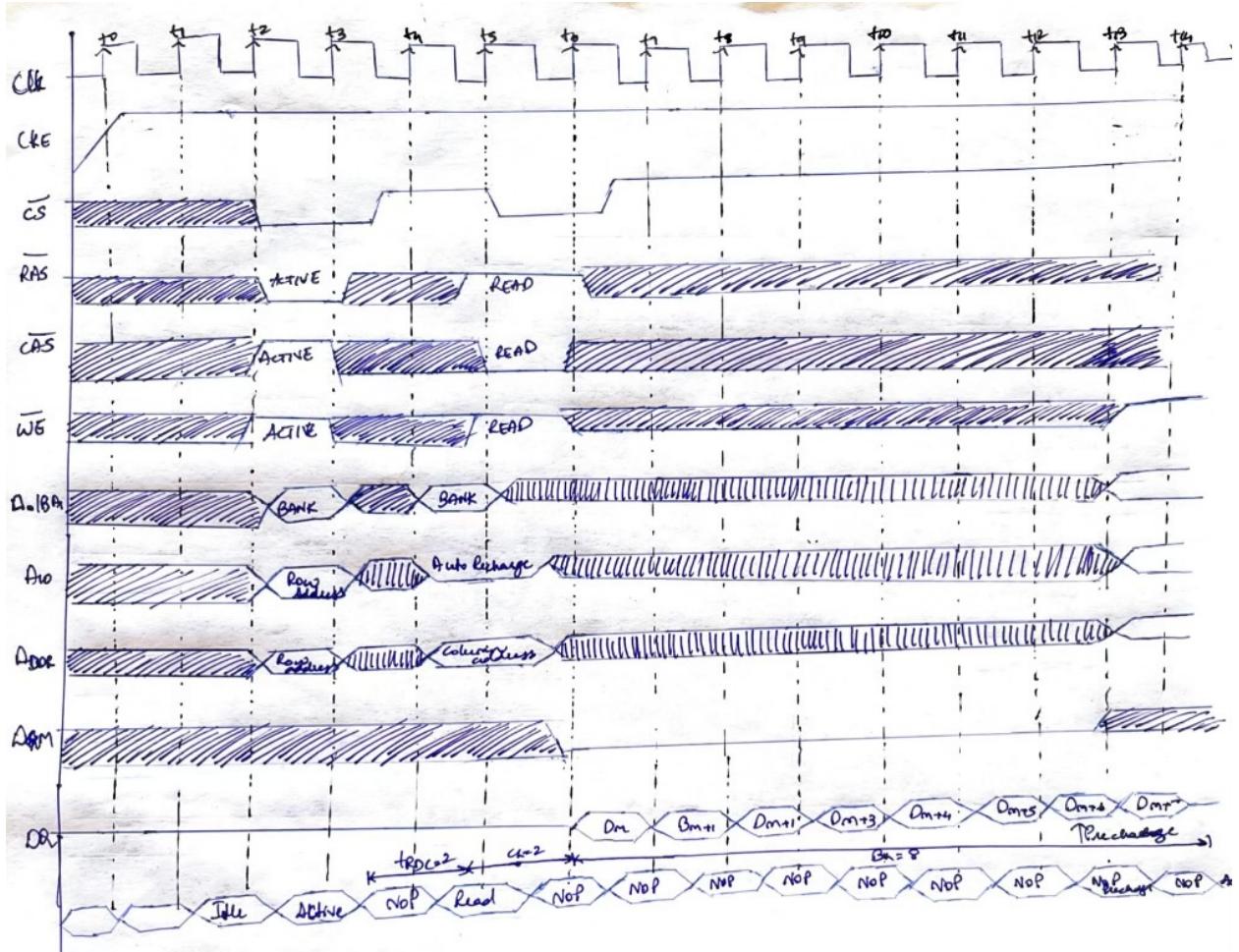
## STATE MACHINE FOR READ:



**STATE TRANSITION TABLE:**

Current State	Condition	Next State
IDLE	RD&! REF_REQ=0	ROW ACTIVE
ROW ACTIVE	null	RD_NOP1
RD_NOP1	null	READ
READ	null	RD_NOP2
RD_NOP2	null	READ1
READ1	null	READ2
READ2	null	READ3
READ3	null	READ4
READ4	null	READ5
READ5	null	READ6
READ6	null	READ7
READ7	null	READ8
READ8	null	RD_NOP3
RD_NOP3	null	RD_NOP4

### TIMING DIAGRAM:



## WRITE:

### CALCULATION FOR WRITE:

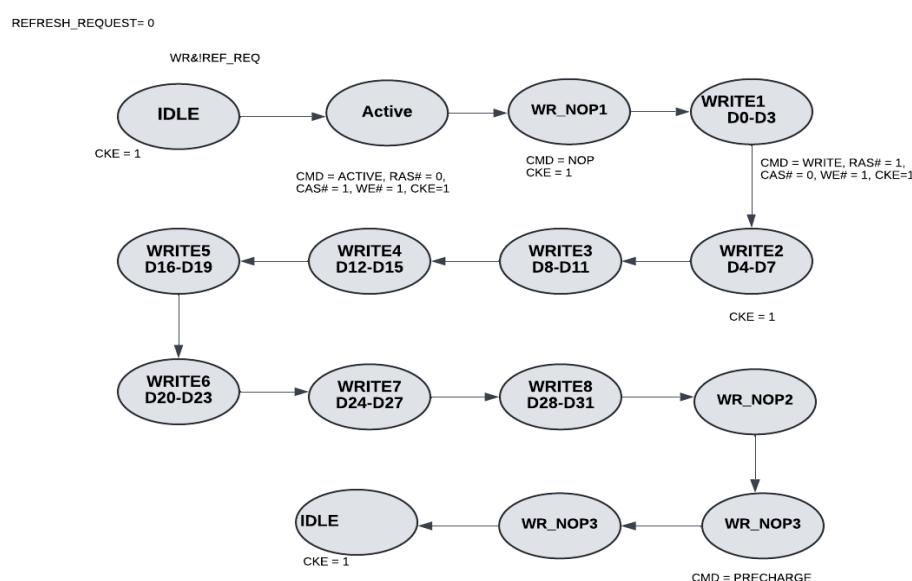
$T_{RD}$  Active to Read/Write delay = 20ns = 2 clock cycles  
 $T_{WR}$  Write Recovery Time = 1clk + 7.5ns = 2 clock cycles  
 $T_{RP}$  Precharge Command Period = 20ns = 2clock cycles  
 $T_{CL}$  = 10ns

### STEPS FOR WRITE:

The WRITE command serves to initiate writing to an active row within the SDRAM. Upon receiving the WRITE command combined with the WR&!REF\_REQ signal, the SDRAM transitions to the Active mode by activating the specified row and bank, with RAS#=0, CAS#=1, and WE#=1 settings. This is succeeded by the WR\_NOP1 phase, leading to the WRITE phase where both the column address and the intended data are specified. The writing process unfolds from WRITE1 through WRITE8, with each phase writing 4 bits of data, totaling 32 bits across all cycles.

Post the WRITE operation, in compliance with the tWR timing specifications, the SDRAM enters the WR\_NOP2 phase. It then progresses through two cycles from WRITE8 to reach the WR\_NOP3 phase. Following a trp delay of 2 cycles, the SDRAM transitions to the IDLE state.

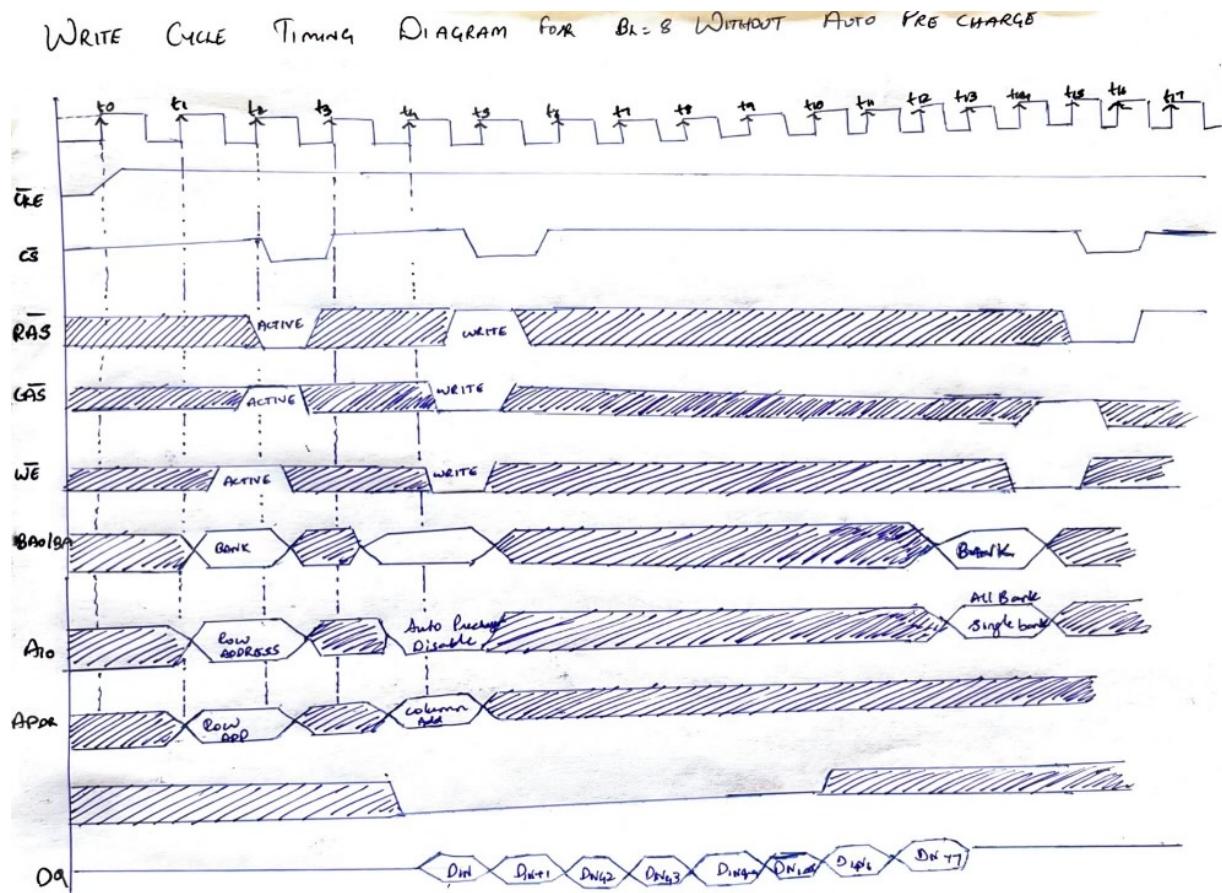
### STATE MACHINE FOR WRITE:



### STATE TRANSITION TABLE:

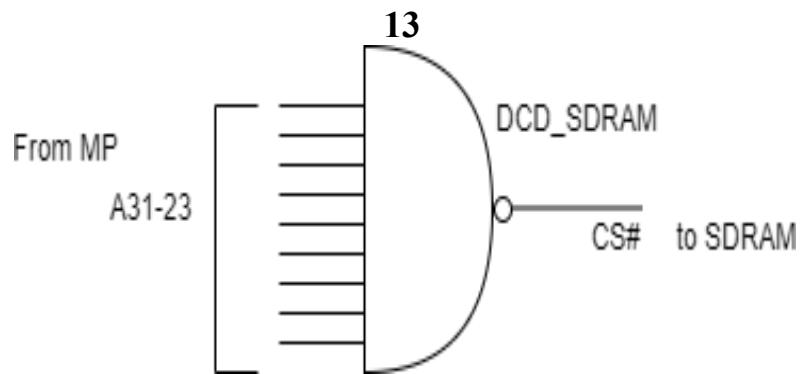
Current State	Condition	Next State
IDLE	WR&!REF_REQ=0	ACTIVE
ACTIVE	null	WR_NOP1
WR_NOP1	null	WRITE1
WRITE1	null	WRITE2
WRITE2	null	WRITE3
WRITE3	null	WRITE4
WRITE4	null	WRITE5
WRITE5	null	WRITE6
WRITE6	null	WRITE7
WRITE7	null	WRITE8
WRITE8	null	WR_NOP2
WR_NOP2	null	WR_NOP3
WR_NOP3	null	WR_NOP4
WR_NOP4	null	IDLE

### TIMING DIAGRAM:

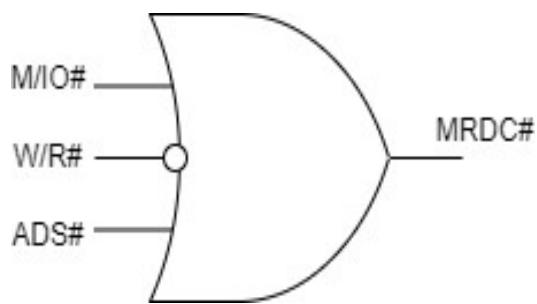


## GENERATION OF CHIP SIGNALS:

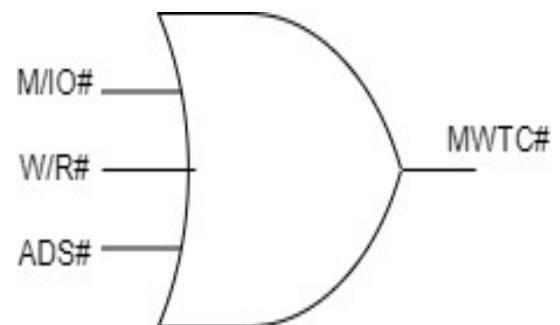
### Generation of Chip Select:



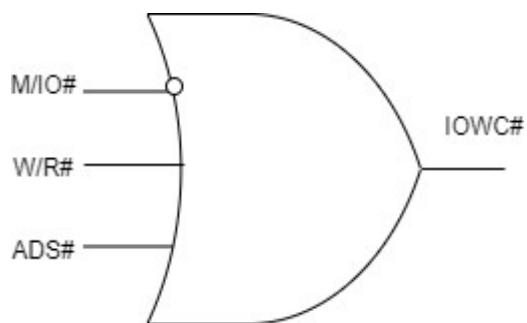
### Generation of MRDC:



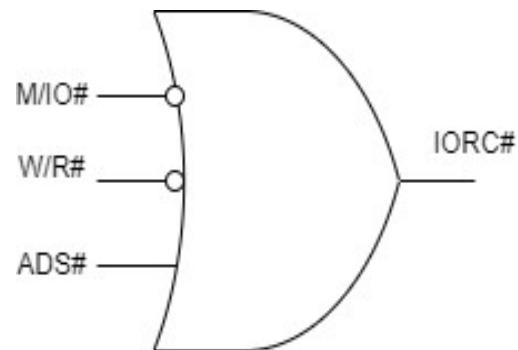
### Generation of MWTC:



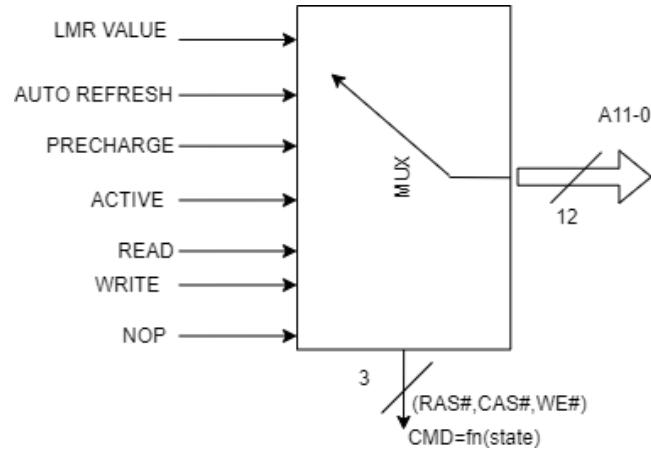
### Generation of IOWC:



### Generation of IORC:

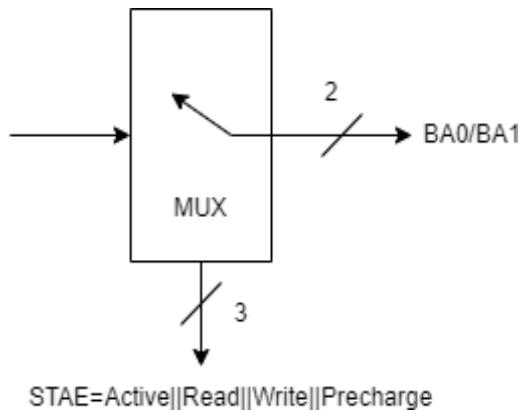


## GENERATION OF FUNCTIONAL COMMAND SIGNAL:

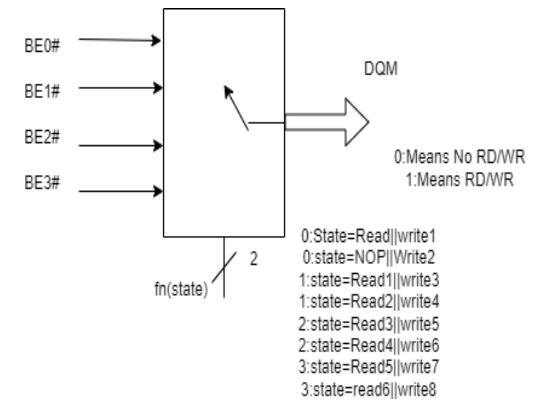


Function	RAS#	CAS#	WE#
LMR VALUE	L	L	L
AUTO REFRESH	L	L	H
PRECHARGE	L	H	L
ACTIVE	L	H	H
READ	H	L	H
WRITE	H	L	L
NOP	H	H	H

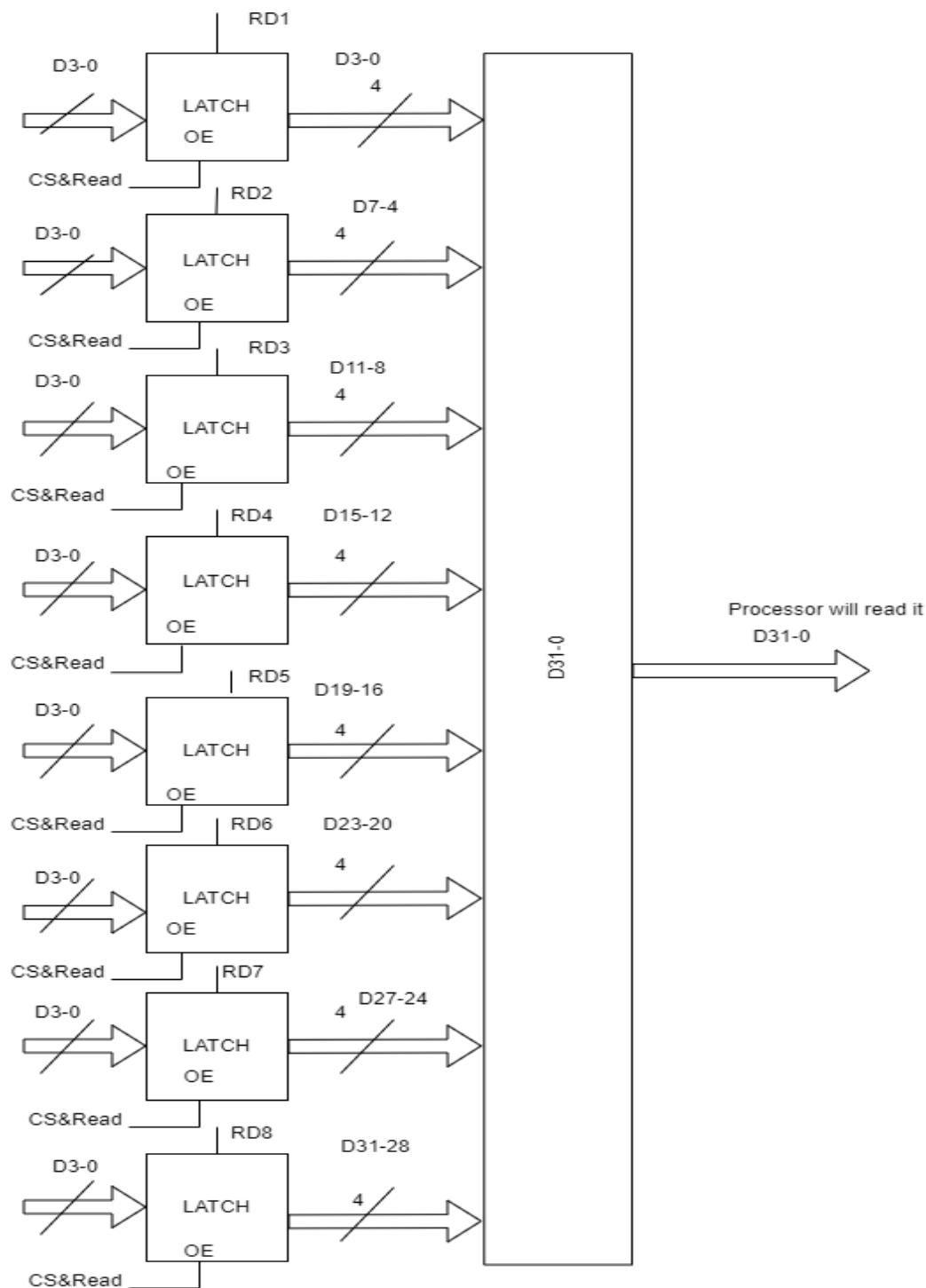
### Generation of Bank Address:



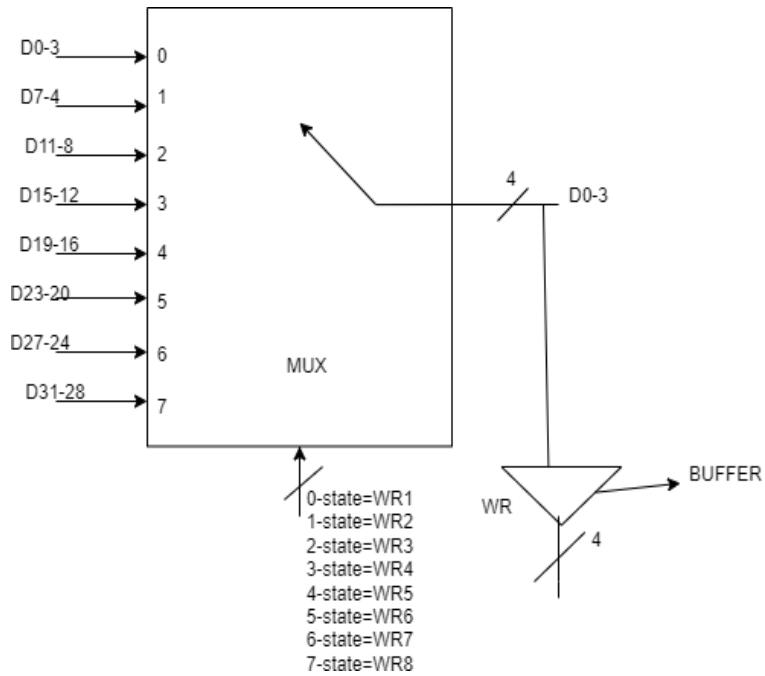
### Generation of DQM:



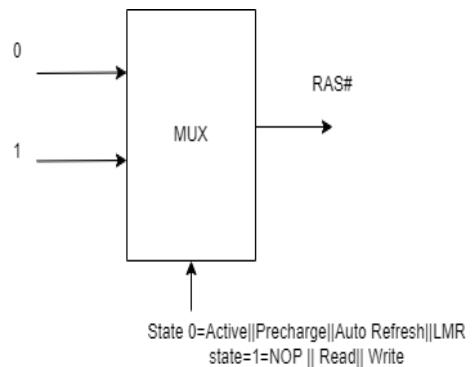
### Data Latch for READ:



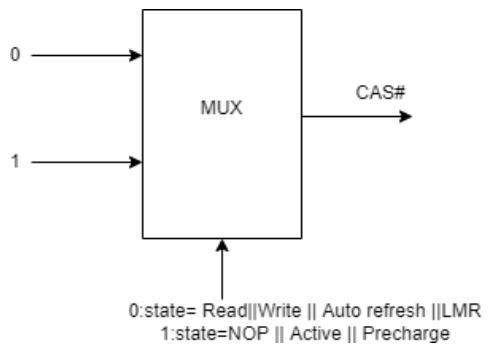
### Data Latch for WRITE:



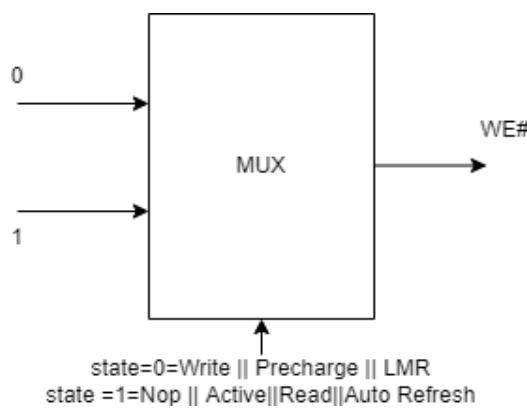
### Generation of RAS:



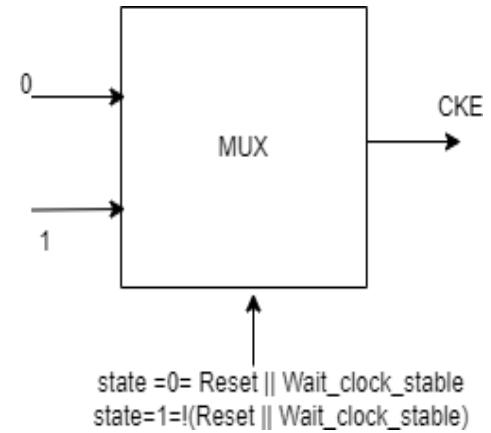
### Generation of CAS:



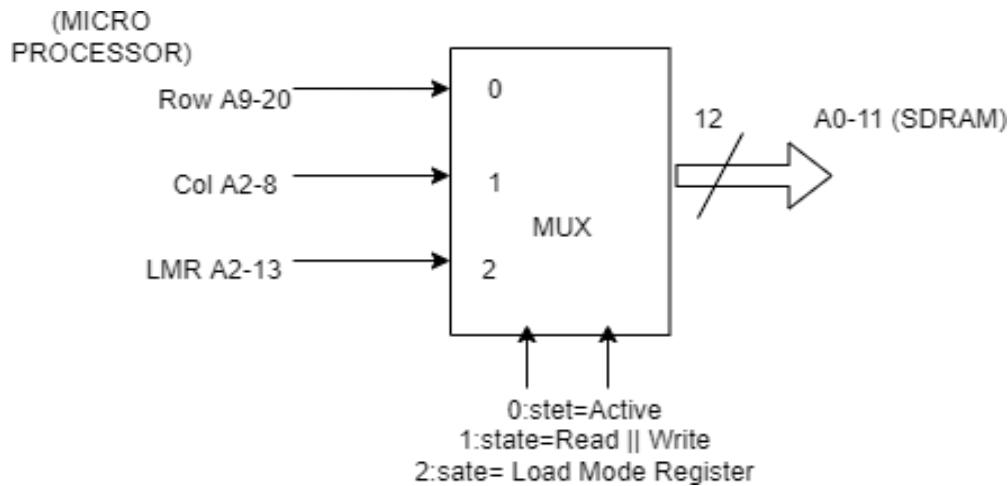
### Generation of WE:



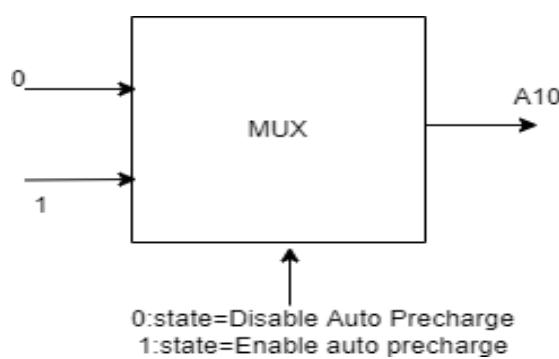
### Generation of CKE:



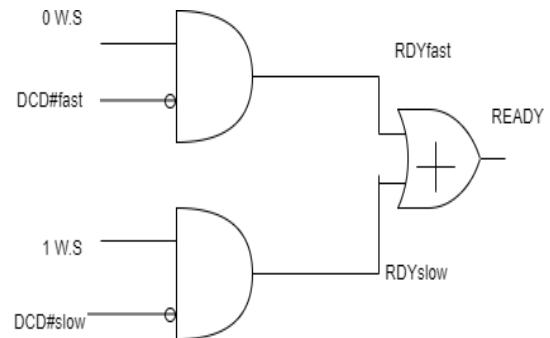
### Generation of Row and Column:



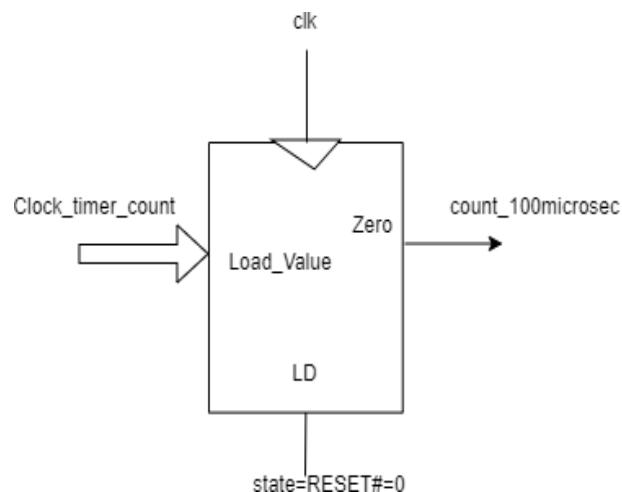
### Generation of Auto Precharge (A10):



### Generation of READY Signal:

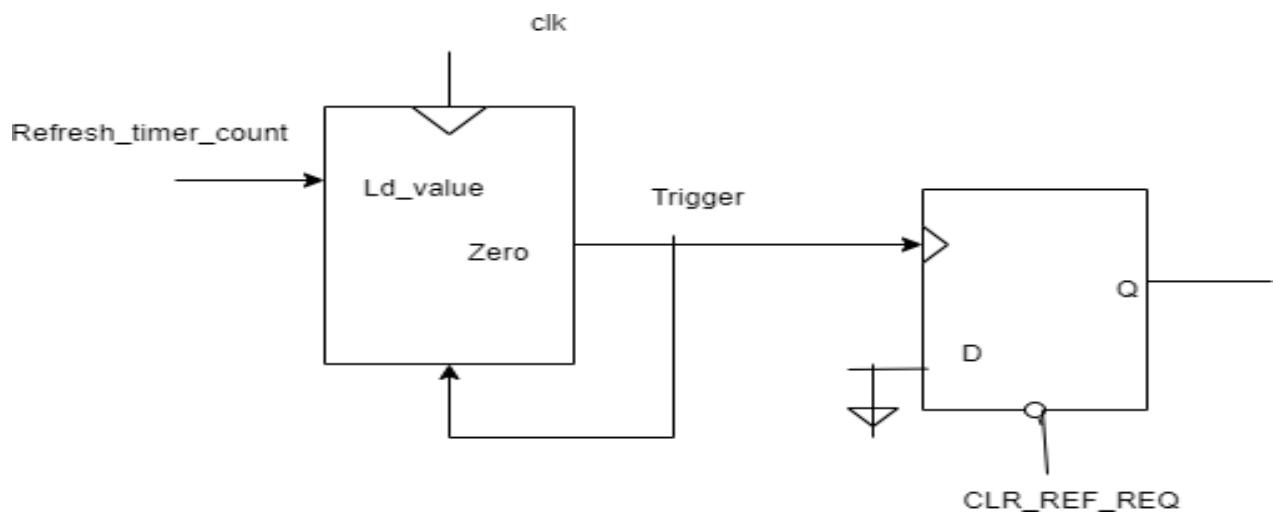


### Generation of Clock Timer:



$$\text{Clock\_timer\_count} = 100\text{microsec} * 100\text{MHz} = 10,000\text{counts}$$

### Generation of Refresh Time:



$$\text{Refresh\_timer\_count} = 64\text{ms} * 100\text{MHz} = 6,400$$

THANK YOU