THE UNIVERSITY OF TEXAS AT ARLINGTON

A Project Report

On

SDRAM Controller Design

Instructor :
Dr. Jason Losh

By-
Deepti Saxena (1001937586)

Shrutika Chaudhari (1002026761)

DEPARTMENT OF ELECTRICAL ENGINEERING
SPRING 2022

INDEX

❖ Objective of the Project:

The aim of this project is to interface the synchronous DRAM memory (MT48LC16M4A2) with microprocessor (80386DX) using the designed synchronous DRAM controller.

❖ Design Specifications:

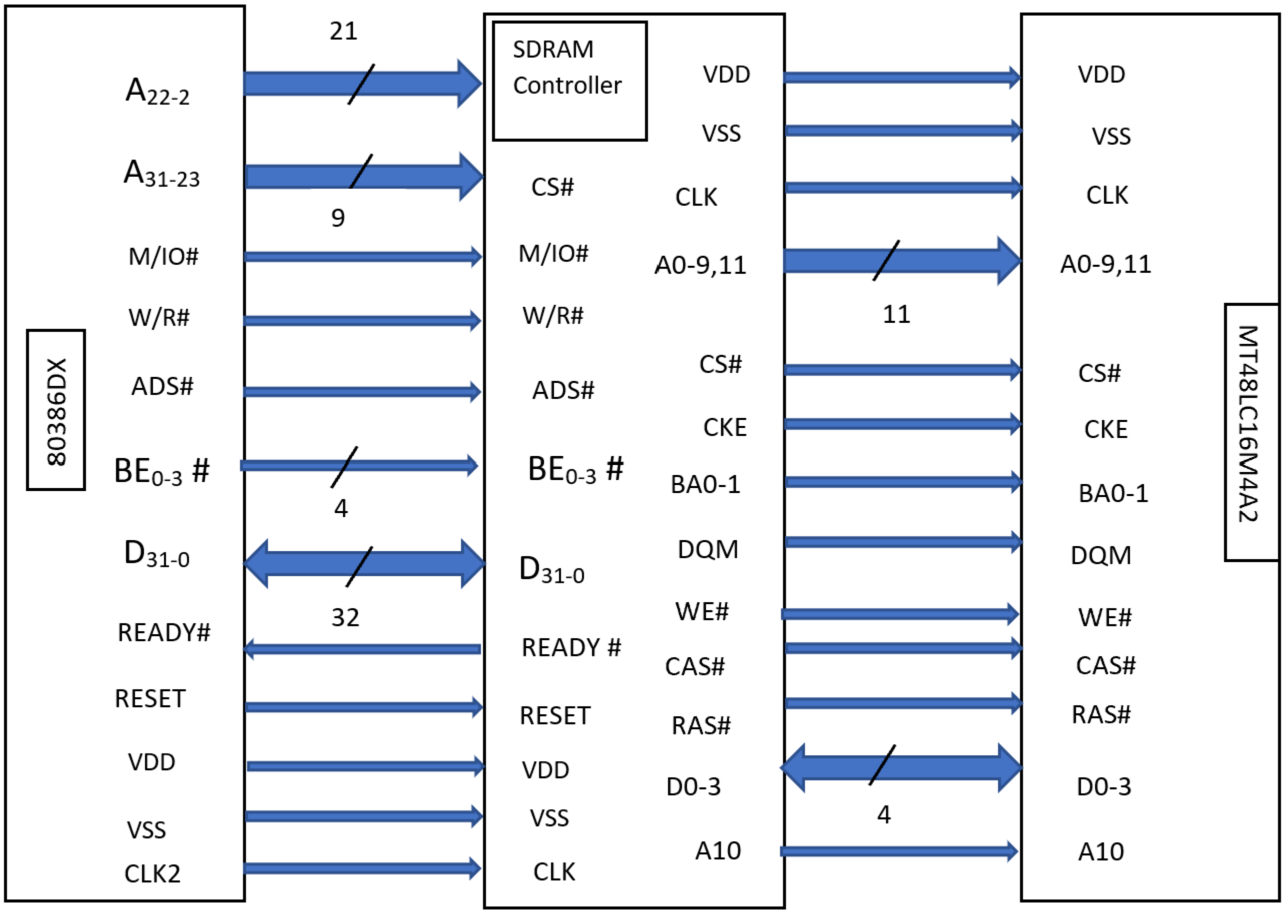
Memory Chip – MT48LC16M4A2	
Configuration	4 Meg x 4 x 4 banks
Speed Grade	-75
Clock Frequency	100 MHz
Clock Period	10n sec
CAS Latency	2
Burst Length	8

❖ Address Table:

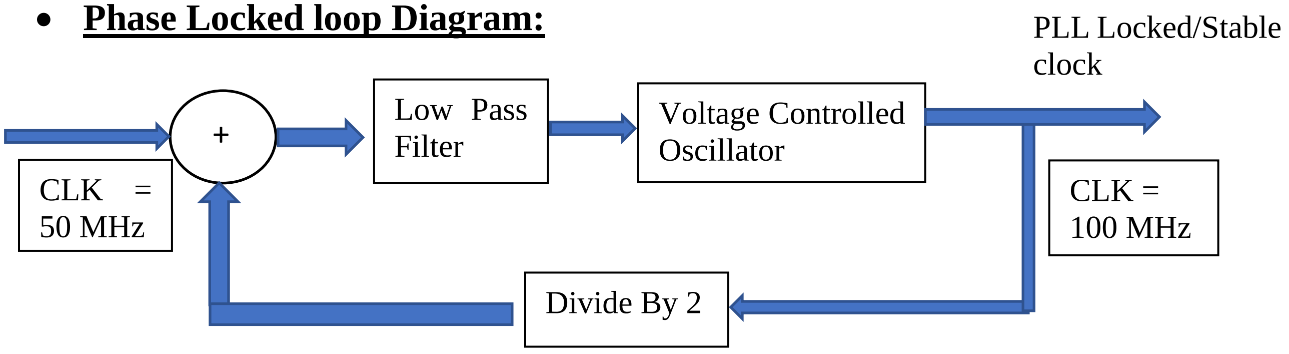
No of Rows	4K (4096)	2^{12}
No. of Columns	1K (1024)	2^{10}
No. of Banks	4	2^{2}
No. of Bits	4	2^{2}
Total		2^{26} = 64MiB

❖ Interfacing Diagram:

In this project SDRAM Controller is interfaced with 80386DX microprocessor (32 bit address and 32 bit data lines) and SDRAM memory. These 32 bit address lines are used to interface with SDRAM memory. The memory chip used (MT48LC16M4A2) has 4 banks of 16 MiB of each.

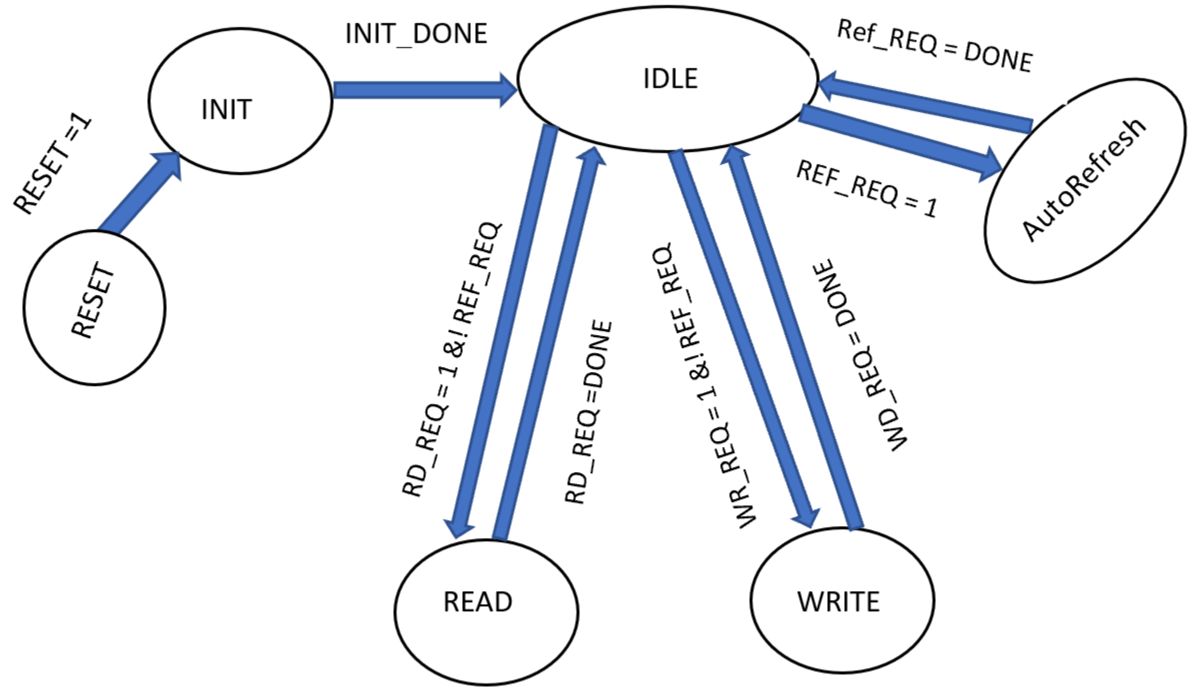


- ## Phase Locked loop Diagram:



Clock Cycle Time = Tclk= 1/100MHZ = 10n sec

❖ Top Level Diagram (Finite State machine) :



- **State Transition Table:**

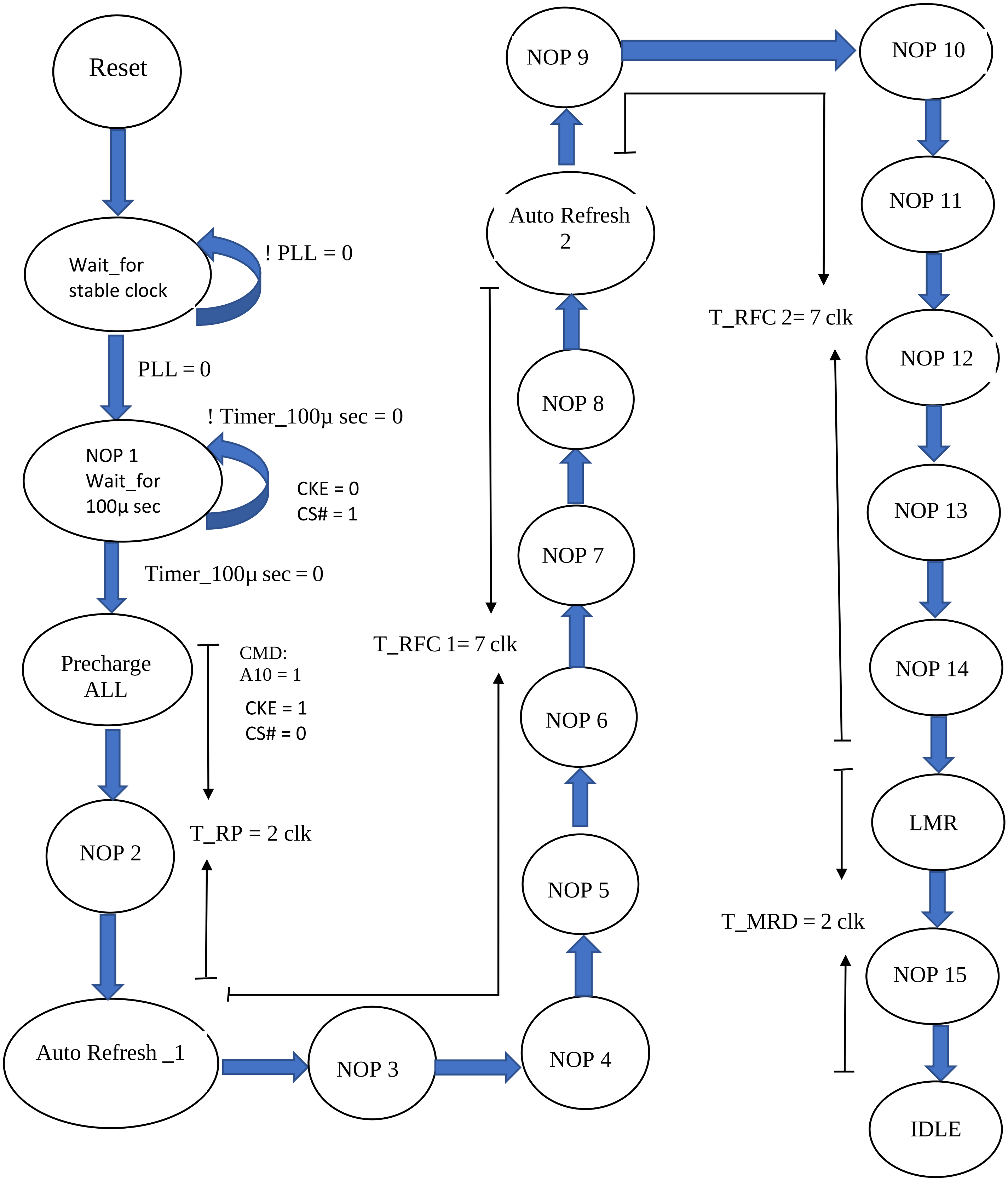
Current State	Condition	Next State
RESET	RESET = 1	INITIALIZATION
INITIALIZATION	INITIALIZATION = DONE	IDLE
IDLE	REF_REQ =1	AUTO REFRESH
AUTO REFRESH	REF_REQ= DONE	IDLE
IDLE	RD_REQ=1 &! REF_REQ	READ
READ	RD_REQ= DONE	IDLE
IDLE	WR_REQ=1 & ! REF_REQ	WRITE
WRITE	WR_REQ = DONE	IDLE

❖ Initialization

Initialization of SDRAM is done in few steps once the RESET Signal is asserted.

- Once the reset signal is given i.e the power is applied to VDD and VDQ. The Memory will wait for stable clock. Stable clock is required so that memory and processor run synchronously.
- After the PLL is lock/Stable clock it will wait for 100µ sec before next command.
- When the counter is zero (i.e 100µ sec wait completed), the SDRAM controller Precharge all banks. It will take 2 clocks (T_RP) for Precharge to get completed.
- Once the Precharge is completed, it will perform at least 2 Auto Refresh operation. Each refresh command will take 7 clocks (T_RFC) to complete.
- After Auto Refresh, SDRAM is ready for Load mode register programming where it defines the Burst Length, CAS Latency, Operating Mode, Burst type, etc. Load Mode Register will take 2 clocks (T_MRD) to complete the operation.
- When the T_MRD is satisfied , the controller will move to IDLE state and wait for any valid command (Read, Write, Auto Refresh).

- **Finite state machine**



- **State Transition table :**

Current State	Condition	Next State
X	Power up	Reset
Reset	Next clock	Wait for stable clock
Wait for stable clock	! PLL = 0	Wait for stable clock
Wait for stable clock	PLL = 0	NOP 1
NOP 1	! Timer_100µ sec = 0	NOP 1
NOP 1	Timer_100µ sec = 0	Precharge ALL
Precharge ALL	Next clock	NOP 2
NOP 2	T_RP Done	Auto Refresh 1
Auto Refresh 1	Next clock	NOP 3
NOP 3	Next clock	NOP 4
NOP 4	Next clock	NOP 5
NOP 5	Next clock	NOP 6
NOP 6	Next clock	NOP 7
NOP 7	Next clock	NOP 8
NOP 8	T_RFC 1 Done	Auto Refresh 2
Auto Refresh 2	Next clock	NOP 9
NOP 9	Next clock	NOP 10
NOP 10	Next clock	NOP 11
NOP 11	Next clock	NOP 12
NOP 12	Next clock	NOP 13
NOP 13	Next clock	NOP 14
NOP 14	T_RFC 2 Done	LMR
LMR	Next clock	NOP 15
NOP 15	T_MRD Done	IDLE

- **OUTPUT Table :**

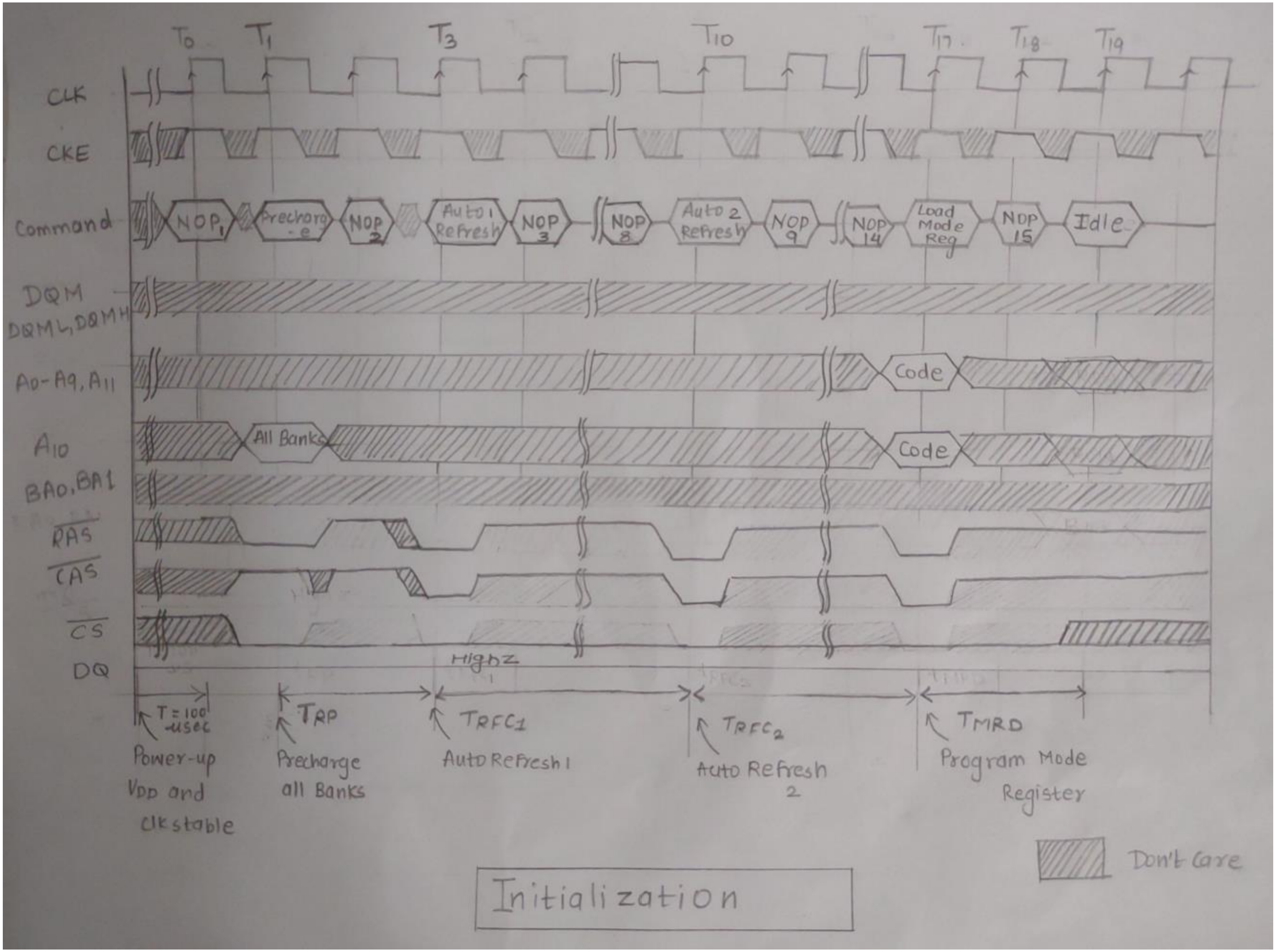
Output	Equation
CKE	STATE = Precharge ALL || NOP a || Auto Refresh b where, a = 2 to 15 || LMR || IDLE b = 1 & 2
CS#	STATE = Precharge ALL || NOP a || Auto Refresh b where, a = 2 to 15 || LMR || IDLE b = 1 & 2
RAS#	STATE = Precharge ALL || Auto Refresh b || LMR where b = 1 & 2
CAS#	STATE = Auto Refresh || LMR where b = 1 & 2
WE#	STATE = Precharge ALL || LMR
A0-9, A11	STATE = Precharge ALL || LMR
A10	STATE = Precharge ALL || LMR

- **CLOCK CALCULATION :**

Parameter	Symbol	Value	No. of Clocks
For 100µ sec	T_100µ	100µ sec	10000
Precharge ALL	T_RP	20 n sec	2
Auto Refresh 1&2	T_RFC 1 & 2	66 n sec	14
Load mode register	T_MRD		2
Total Clocks			10018

- **Load Mode Register :**

Address Bus	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
MODE Register	M11	M10	M9	M8	M7	M6	M5	M4	M3	M2	M1	M0
Parameter	Reserved		Write Burst	Operating Mode		CAS Latency			Burst Type	Burst Length		
Value	0	0	0	0	0	0	1	0	0	0	1	1



Initialization

❖ **AUTO REFRESH**

The steps required to perform the Auto Refresh operation –

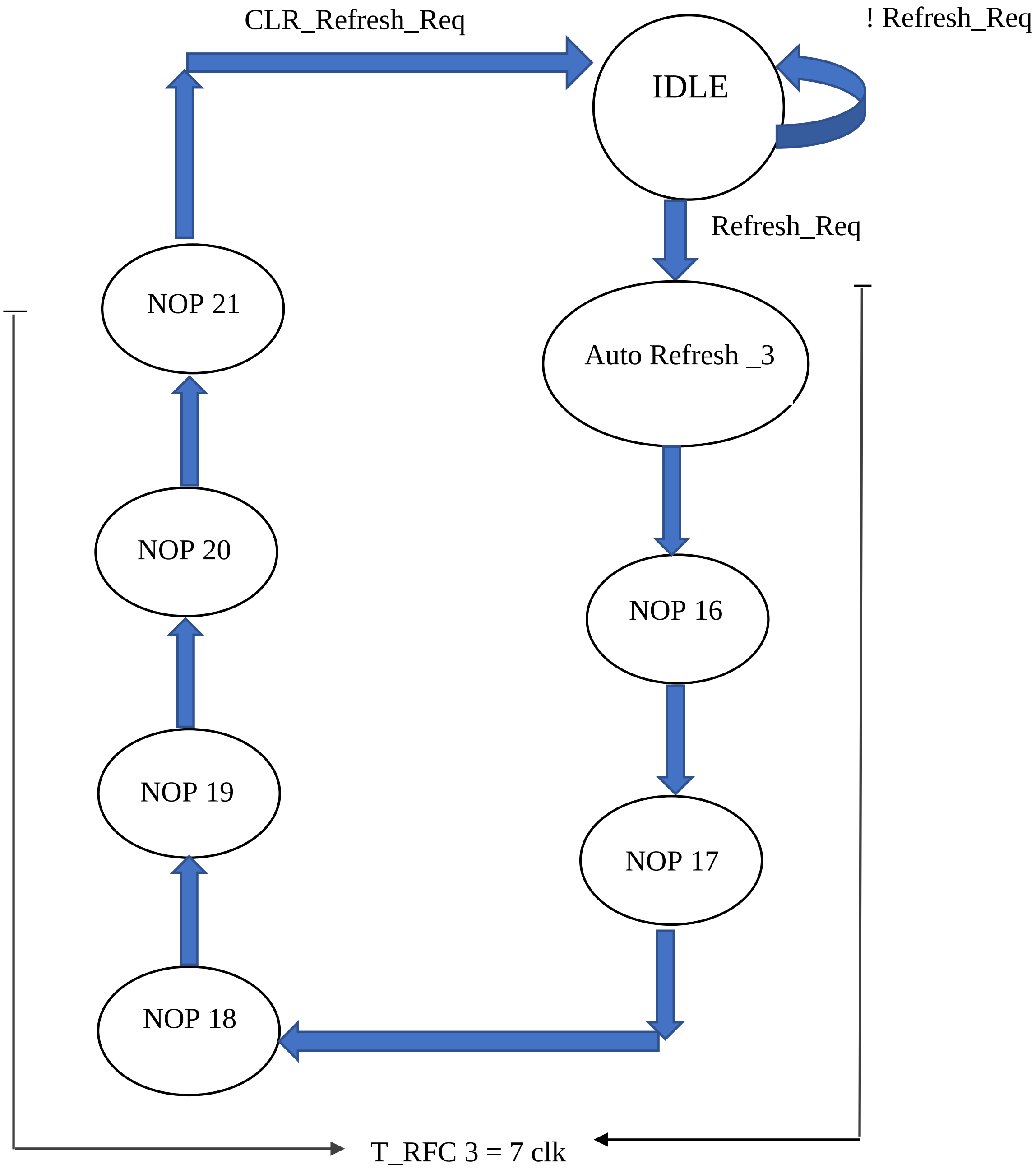
- The memory chip used in this project has 4096 rows and 1024 columns in each of the 4 banks.

- To refresh the 4096 rows, it takes 64 m sec. So, each row will take 64 msec/4096 rows = 15.62μ sec.

- When the count of Refresh request counter is zero, it will send the refresh request command and perform the Auto Refresh operation.

- The Auto Refresh operation will take 7 clocks (T_RFC) to complete the task. Once the operation is done the refresh counter (attached to the latch) clears the refresh request. So that it will not interrupt the read or write operation.

- **Finite State Machine :**



- **State Transition table :**

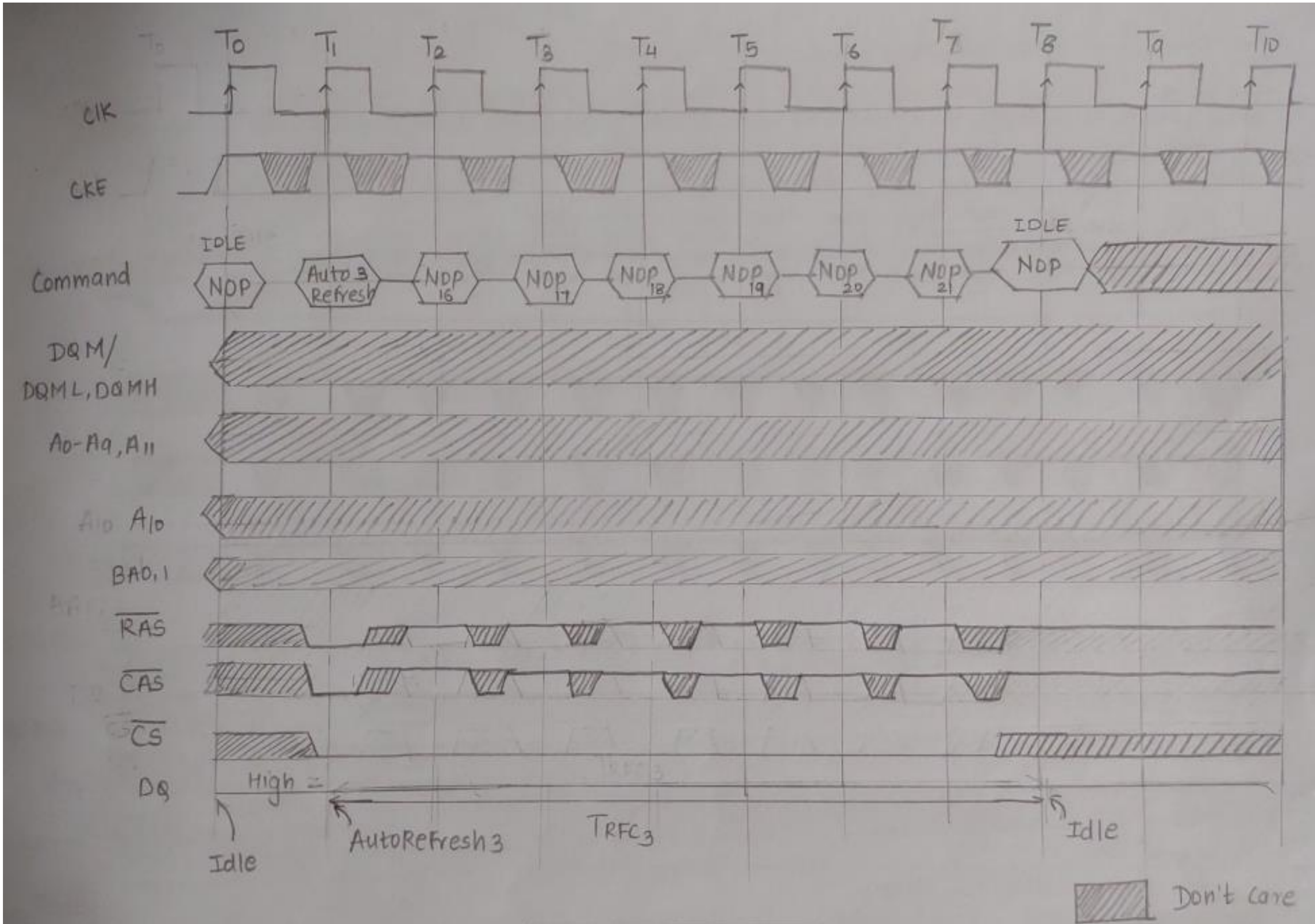
Current State	Condition	Next State
IDLE	! Refresh_Req	IDLE
IDLE	Refresh_Req	Auto Refresh 3
Auto Refresh 3	Next clock	NOP 16
NOP 16	Next clock	NOP 17
NOP 17	Next clock	NOP 18
NOP 18	Next clock	NOP 19
NOP 19	Next clock	NOP 20
NOP 20	Next clock	NOP 21
NOP 21	T_RFC 3 Done	IDLE

- **OUTPUT Table :**

Output	Equation
CKE	STATE = Auto Refresh 3 || NOP a || IDLE where, a = 16 to 21
CS#	STATE = Auto Refresh 3 || NOP a || IDLE where, a = 16 to 21
RAS#	STATE = Auto Refresh 3
CAS#	STATE = Auto Refresh 3

- **CLOCK CALCULATION:**

Parameter	Symbol	Value	No. of Clocks
Auto Refresh 3	T_RFC 3	66 n sec	7
From IDLE to Auto Refresh 3			1
Total Clocks			8



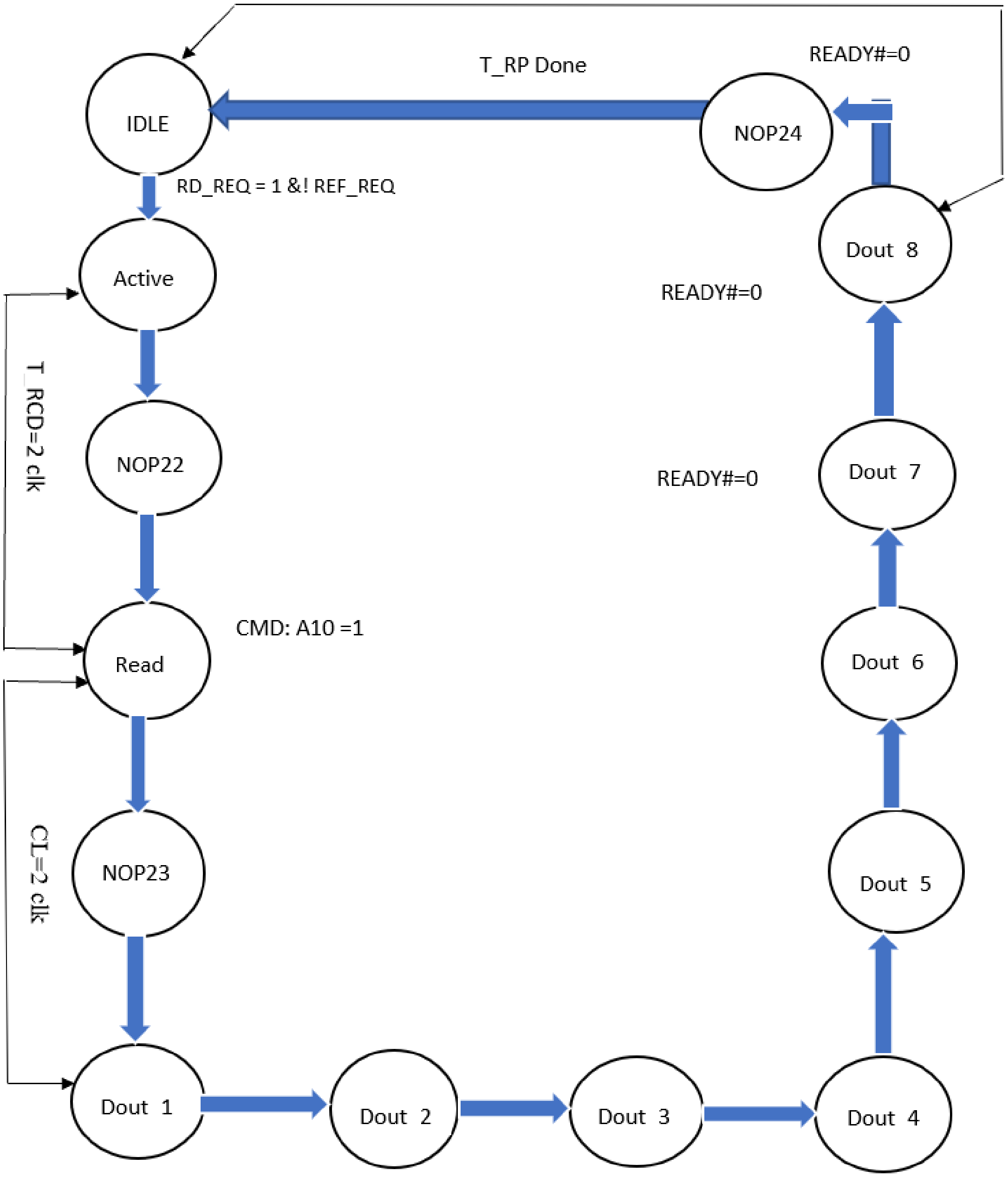
Auto Refresh

❖ **READ OPERATION**

Steps to perform the Read operation –

- When the processor gets the memory read request and not refresh request, then it will start the read operation by activating the row and bank signals.
- Once the T_RCD time (Active to read delay) (2 clocks) completed, It will perform the read command and in this command, Precharge will be performed by enabling the A10 signal.
- After the CAS latency (2 Clk), the microprocessor will read the data from memory in nibble and latched it.
- Once the 32-bit data is latched, the SDRAM controller will send a ready signal to the microprocessor.
- Once the data is out and also precharge is done i.e row will get close then the SDRAM is ready for next operation.

- **Finite State Machine:**



- **State Transition Table :**

Current State	Condition	Next State
IDLE	RD_REQ = 1 &! REF_REQ	Active
Active	Next clock	NOP22
NOP22	T_RCD Done	Read
Read	Next clock	NOP23
Nop23	T_CL Done	Dout_1
Dout_1	Next clock	Dout_2
Dout_2	Next clock	Dout_3
Dout_3	Next clock	Dout_4
Dout_4	Next clock	Dout_5
Dout_5	Next clock	Dout_6
Dout_6	Next clock	Dout_7
Dout_7	Next clock	Dout_8
Dout_8	Next clock	NOP24
NOP24	T_RP	IDLE

- **OUTPUT Table :**

Output	Equation

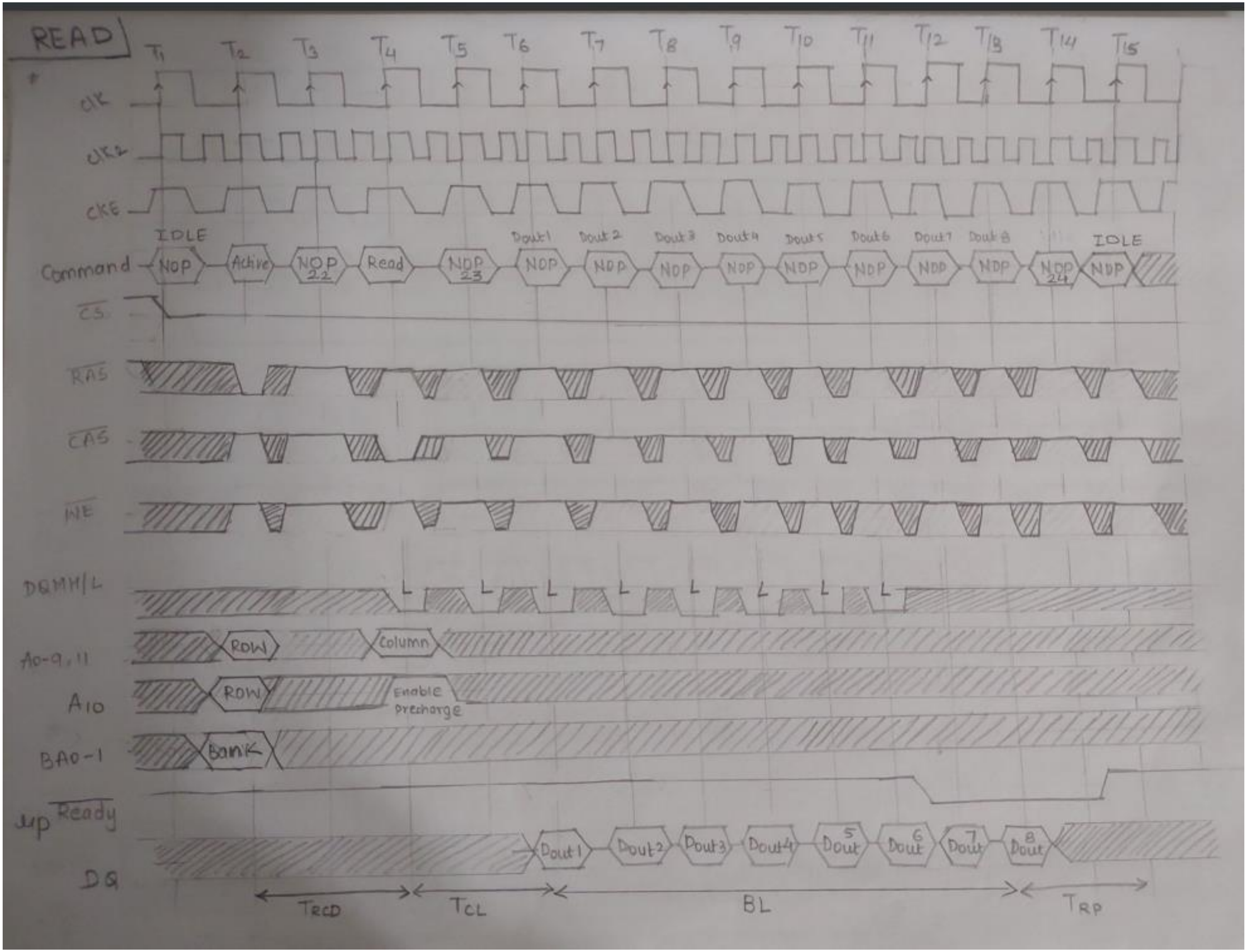
CKE	STATE = IDLE || ACTIVE || READ || Dout$_n$ || NOP$_a$ Where, n = 1 to 8
	a = 22 to 24

CS#	STATE = IDLE || ACTIVE || READ || Dout$_n$ || NOP$_a$ Where, n = 1 to 8
	a = 22 to 24

RAS#	STATE = ACTIVE
CAS#	STATE = READ
A0-A9/A11	STATE = ACTIVE || READ
A10	STATE = READ || ACTIVE
DQs	STATE = Dout_n where n=1-8
BA0, BA1	STATE = ACTIVE

- **Clock Calculation:**

Parameter	Symbol	Value	No. of Clocks Required
ACTIVE-to-READ delay	T_RCD	20 (n sec)	2
Burst Length	BL	-	8
CAS Latency	CL	-	2
NOP24 to IDLE	-	-	1
IDLE to Active	-	-	1
Total Clocks			14 Clocks


READ
T1 T2 T3 T4 T5 T6 T7 T8 T9 T10 T11 T12 T13 T14 T15
CK
CK2
CKE
IDLE
Dout1 Dout2 Dout3 Dout4 Dout5 Dout6 Dout7 Dout8
IDLE
Command
NOP Active NOP 22 Read NOP 23 NOP NOP NOP NOP NOP NOP NOP NOP NOP 24 NOP
CS
RAS
CAS
WE
DQMH/L
L L L L L L L L
A0-9, 11
ROW
Column
A10
ROW
Enable Precharge
BA0-1
Bank
up Ready
DQ
Dout1 Dout2 Dout3 Dout4 Dout 5 Dout 6 Dout 7 Dout 8
TRCD
TCL
BL
TRP

❖ **WRITE OPERATION**

Steps to perform the Write operation –

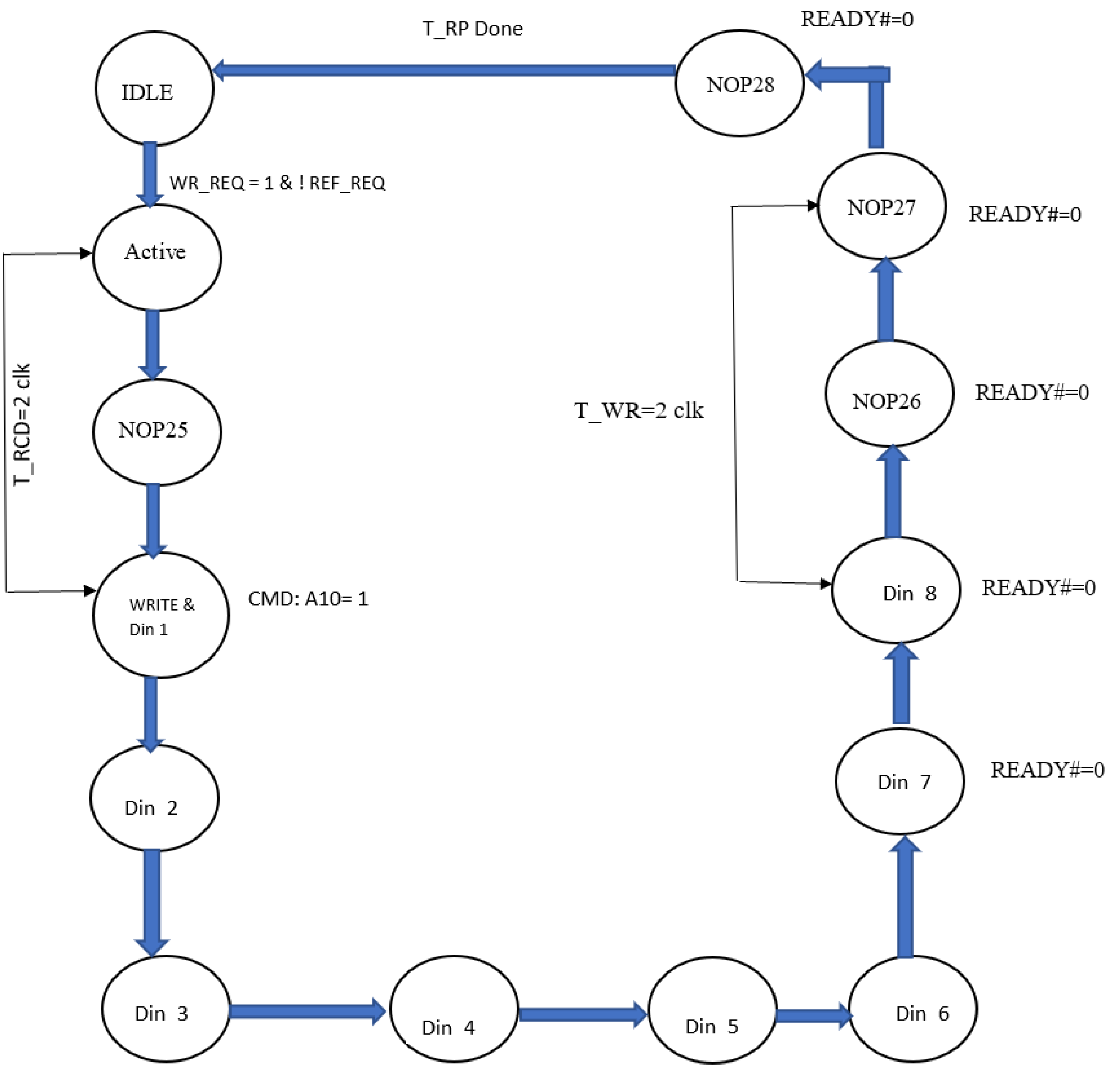
- When the processor gets the memory write request and not refresh request, then it will start the write operation by activating the row and bank signals.

- Once the T_RCD time (Active to write delay) (2 clocks) completed, It will perform the write command. During which the data will come from microprocessor to memory and also the Precharge is enabled by giving A10 signal high.

- Once the 32-bit data is buffered, the SDRAM controller will send a ready signal to the microprocessor.

- When all the data is written to memory, It will take atleast two clock cycle for the write recovery (T_WR = 2 Clk) to get completed. Once write recovery is completed and also precharge is done i.e row will get close then the SDRAM is ready for next operation.

- **Finite State Machine :**



- **STATE Transition Table :**

Current State	Condition	Next State
IDLE	WR_REQ = 1 &! REF_REQ	Active
Active	Next clock	NOP24
NOP25	T_RCD Done	WRITE
WRITE & Din 1	Next clock	Din 2
Din 2	Next clock	Din 3
Din 3	Next clock	Din 4
Din 4	Next clock	Din 5
Din 5	Next clock	Din 6
Din 6	Next clock	Din 7
Din 7	Next clock	Din 8
Din 8	Next clock	NOP26
NOP26	Next clock	NOP27
NOP27	T_WR Done	NOP28
NOP28	T_RP Done	IDLE

- **OUTPUT Table:**

Output	Equation

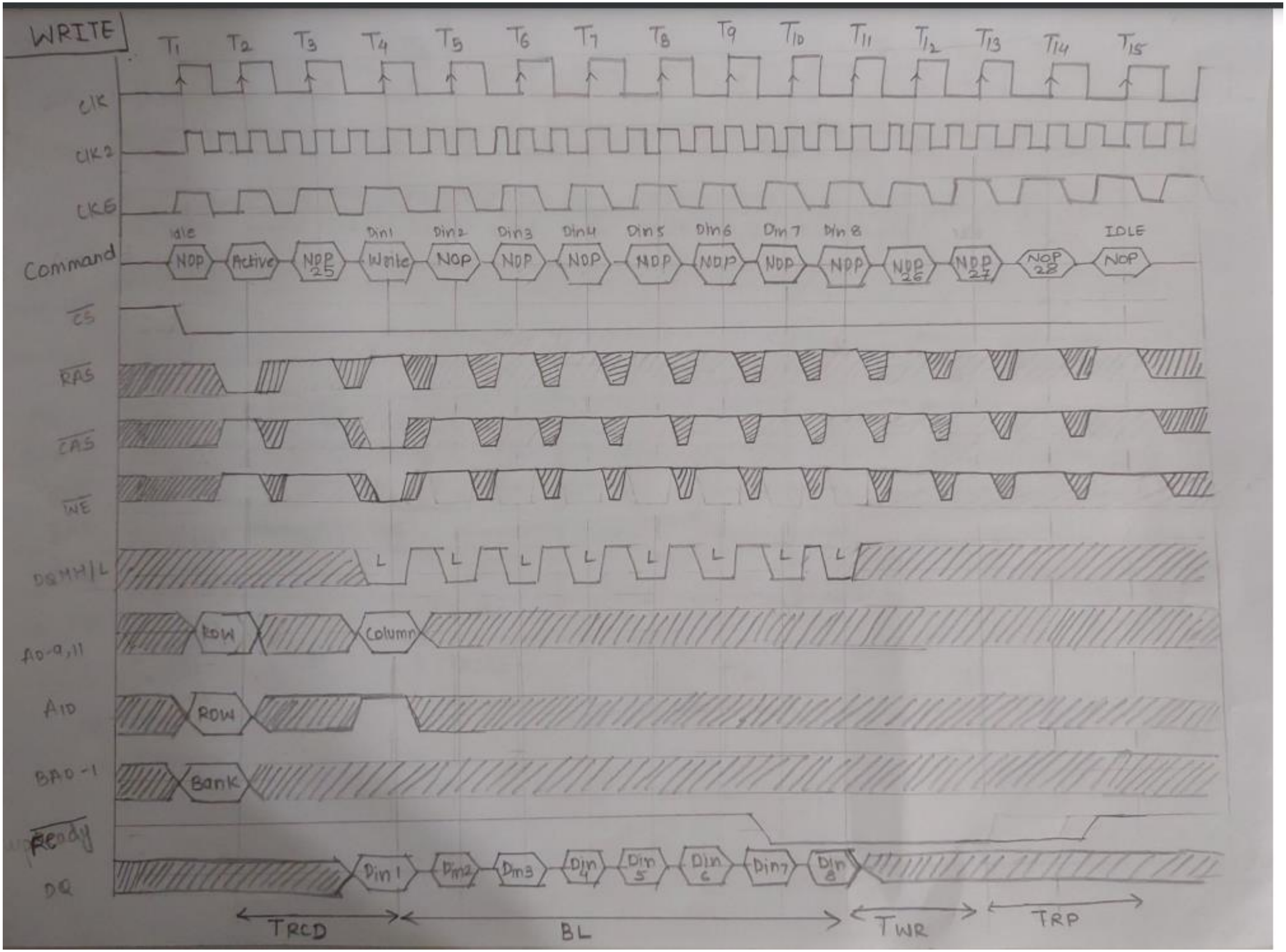
CKE	STATE = IDLE || ACTIVE || WRITE || Din$_n$ || NOP$_a$ Where, n = 1 to 8
	a = 25 to 28

CS#	STATE = IDLE || ACTIVE || WRITE || Din$_n$ || NOP$_a$ Where, n = 1 to 8
	a = 25 to 28

RAS#	STATE = ACTIVE
CAS#	STATE = WRITE
A0-A9/A11	STATE = ACTIVE || WRITE
A10	STATE = WRITE || ACTIVE
DQs	STATE = Din_n where n=1-8
BA0, BA1	STATE = ACTIVE

- **CLOCK Calculation:**

Parameter	Symbol	Value	No. of Clocks Required
ACTIVE-to- WRITE delay	T_RCD	20n sec	2
WRITE Recovery time	-	1 Clk + 7.5n sec	2
Burst Length	BL - 1	-	7
NOP27 to IDLE	-	-	2
IDLE to Active			1
Total Clocks			14 Clocks

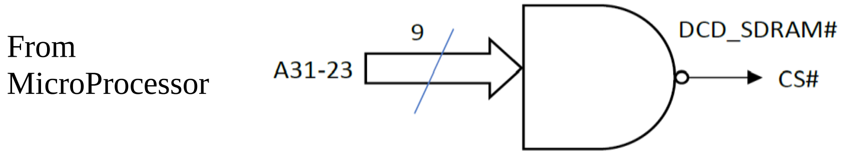

WRITE
T1 T2 T3 T4 T5 T6 T7 T8 T9 T10 T11 T12 T13 T14 T15
CLK
CLK2
CKE
Command
Idle
Din1 Din2 Din3 Din4 Din5 Din6 Din7 Din8
IDLE
NOP
Active
NOP 25
Write
NOP
NOP 26
NOP 27
NOP 28
CS
RAS
CAS
WE
DQMH/L
L
A0-9,11
Row
Column
A10
ROW
BA0-1
Bank
Ready
DQ
Din1
Din2
Din3
Din4
Din5
Din6
Din7
Din8
TRCD
BL
TWR
TRP

❖ GENERATION OF CONTROL SIGNALS :

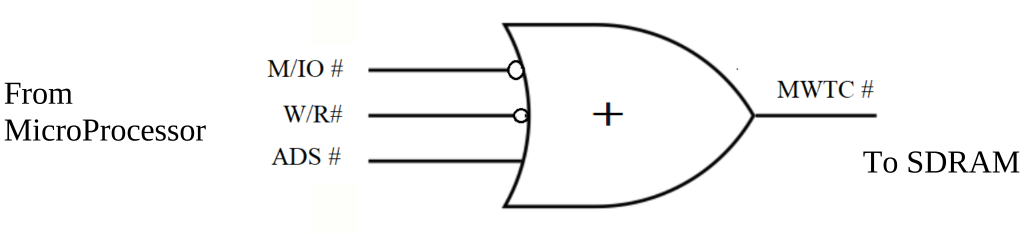
Decoder	Bank Signals Address	Row Address	Column Address	× ×
A31 A23	A22 A21	A20 A9	A8 A2	A1 A0

- Decoder Address – A31 – A23 (9 bits)
- Bank Signal Address – A22-A21 (2 bits)
- Row Address – A20 – A9 (12 bits)
- Column Address – A8 – A0 (10 bits)

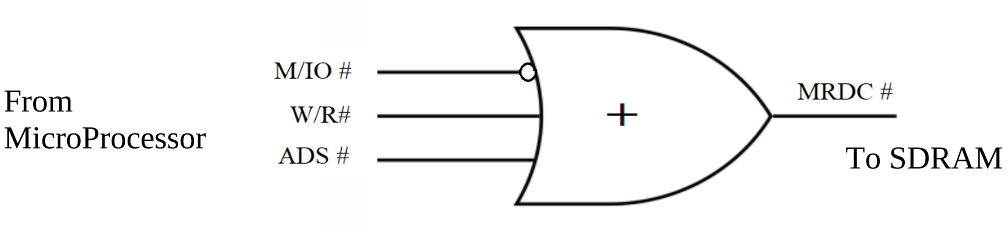
- **Chip select signal :**



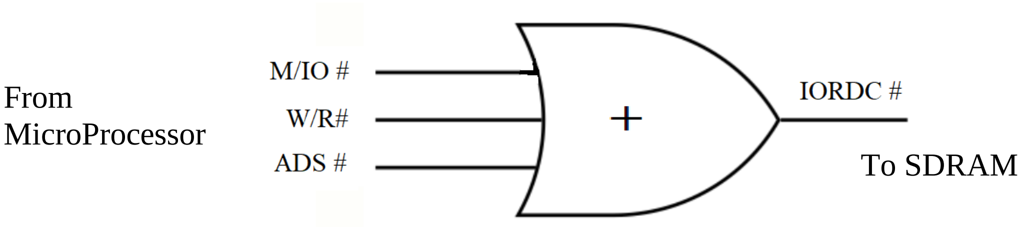
- **Memory write control signal :**



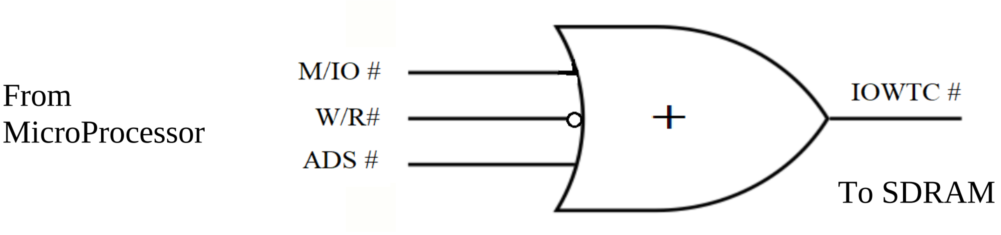
- **Memory Read control signal :**



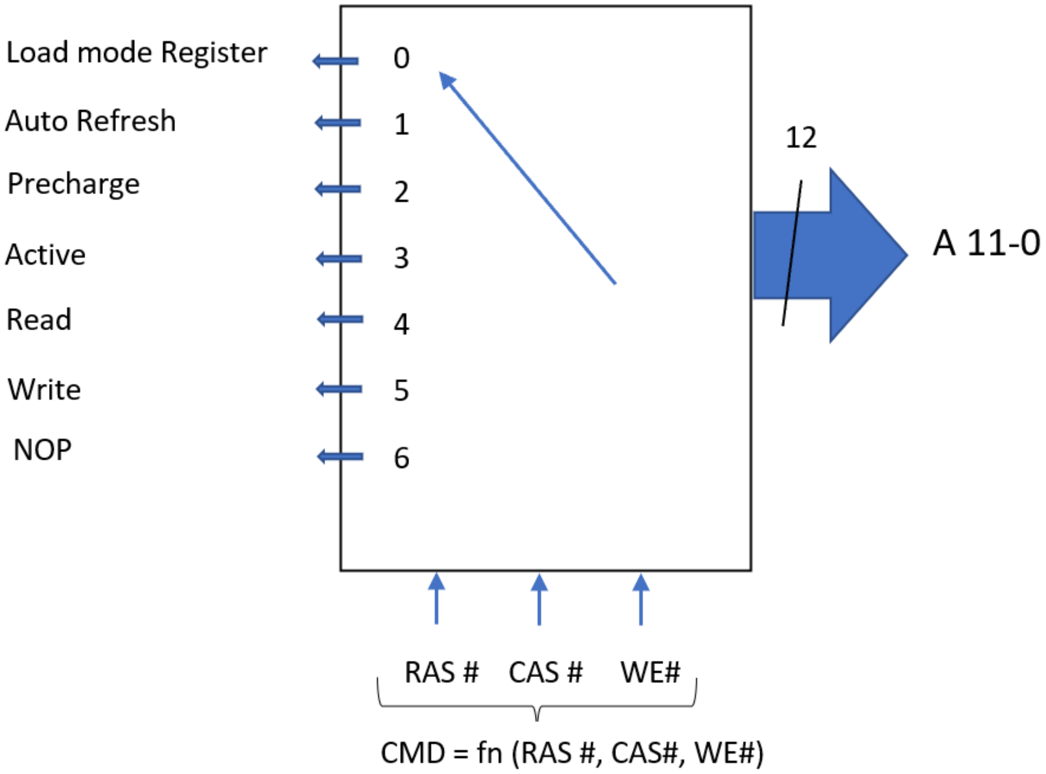
- **I/O Read control signal :**



- **I/O Write control signal :**



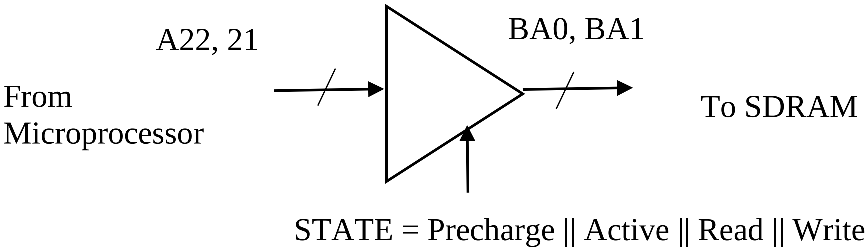
- **Command Signals :**



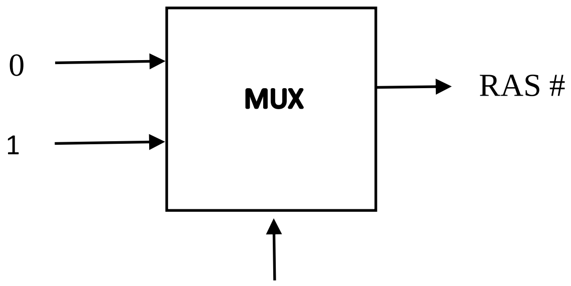
- **Command state table :**

STATE	COMMAND	RAS #	CAS #	WE #
0	Load Mode Register	0	0	0
1	Auto Refresh	0	0	1
2	Precharge	0	1	0
3	Active	0	1	1
4	Read	1	0	1
5	Write	1	0	0
6	NOP	1	1	1

- **Bank Signals:**



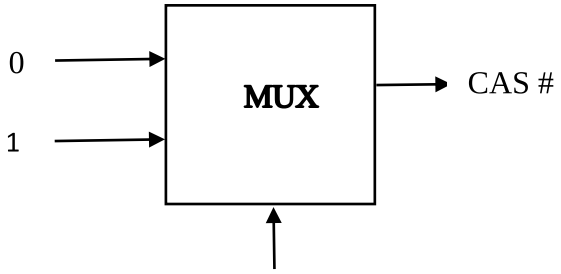
- **RAS # Signal:**



CMD = fn (Active, Precharge, Auto Refresh, LMR, NOP, Read, Write)

For 0 – STATE = Active || Precharge || Auto refresh$_b$ || Load mode Register
For 1 – STATE = NOP$_a$ || Read || Write Where, a = 2 to 28 & b = 1 to 3

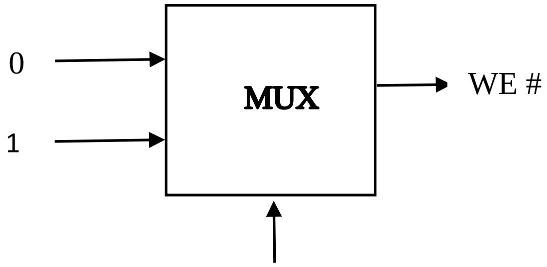
- **CAS # Signal:**



CMD = fn (Active, Precharge, Auto Refresh, LMR, NOP, Read, Write)

For 0 – STATE = Auto refresh$_b$ || Read || Write || Load mode Register
For 1 – STATE = Active || NOP$_a$ || Precharge   Where, a = 2 to 28 & b = 1 to 3

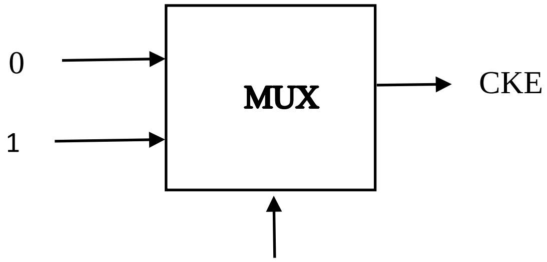
- **WE # Signal:**



CMD = fn (Active, Precharge, Auto Refresh, LMR, NOP, Read, Write)

For 0 – STATE = Precharge || Write || Load mode Register
For 1 – STATE = Active || NOP$_a$ || Auto refresh$_b$ || Read   Where, a = 2 to 28 & b = 1 to 3

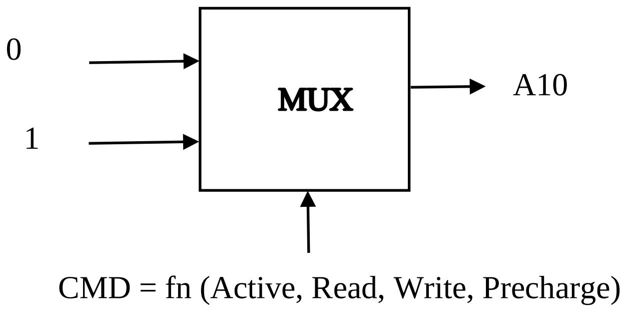
- **Clock Enable (CKE) Signal:**



CMD = fn (Active, Precharge, Auto Refresh, LMR, NOP, Read, Write)

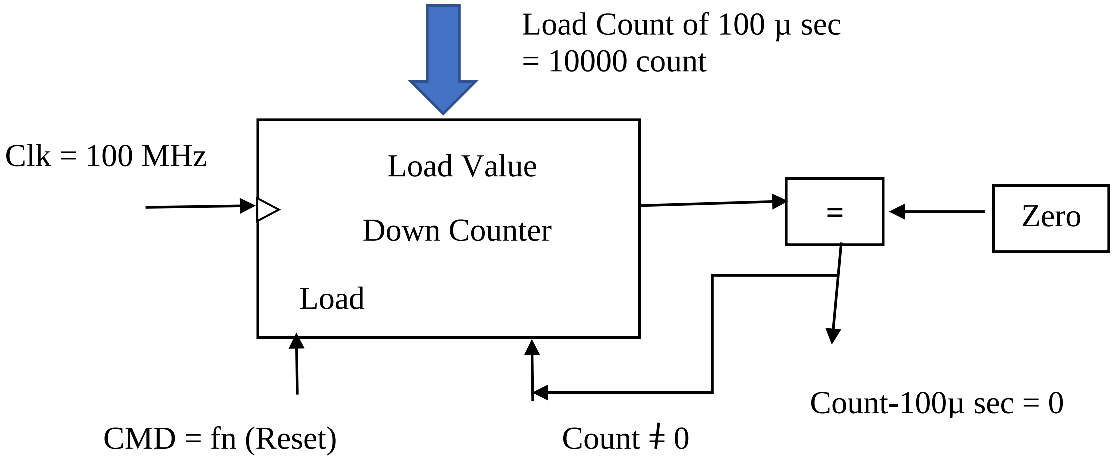
For 0 – STATE = Reset || Wait for stable clock
For 1 – STATE = Active || NOP$_a$ || Auto refresh$_b$ || Read || Precharge || Write || Load mode Register   Where, a = 1 to 28 & b = 1 to 3

- **A10 Signal:**

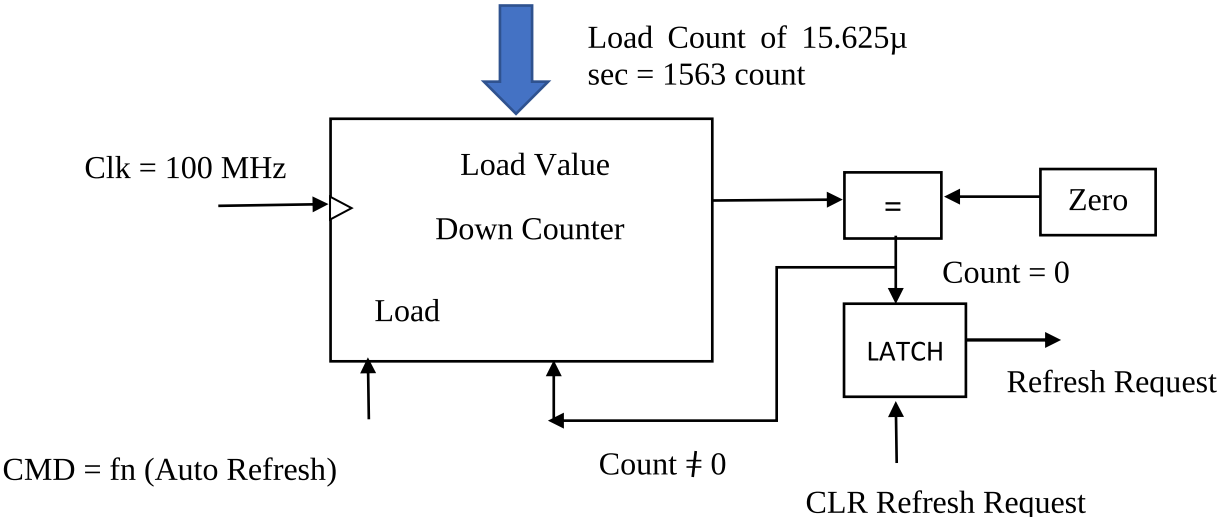


For 0 – STATE = Active
For 1 – STATE = Precharge || Read || Write

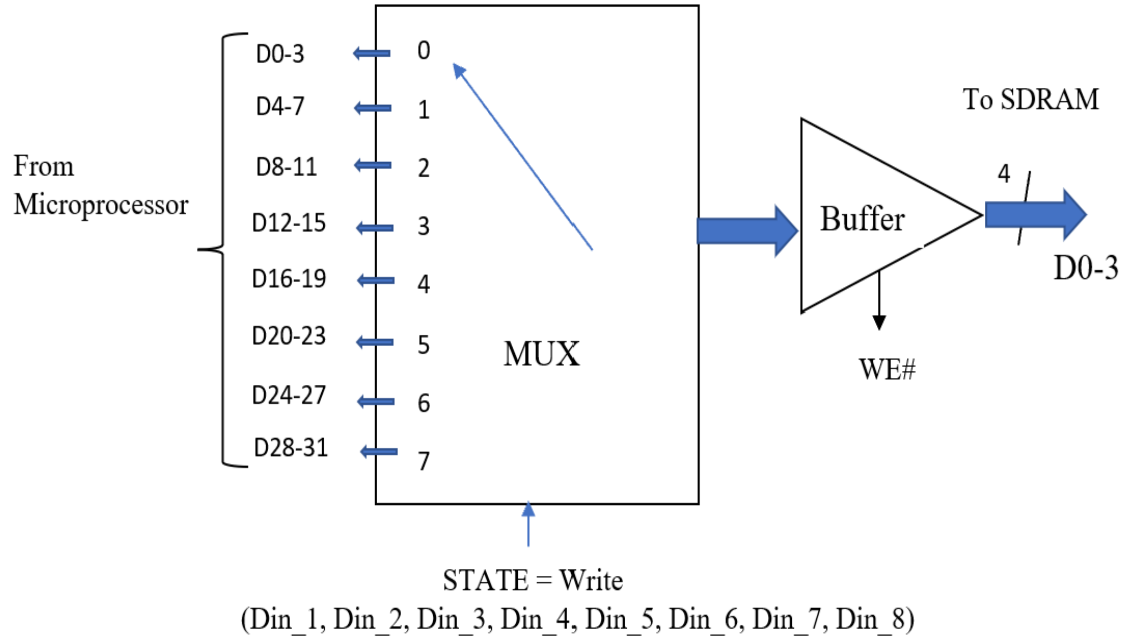
- **100µ sec wait Timer Counter:**



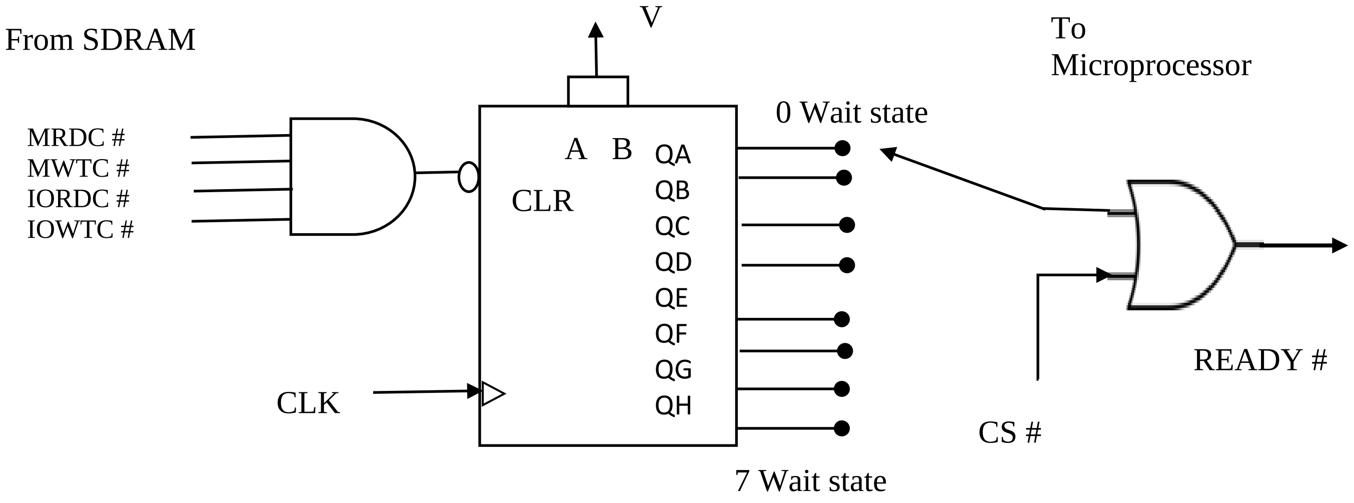
- **Refresh Request Timer Counter:**



- **Data BUFFER for Write:**



- **Ready Signal:**

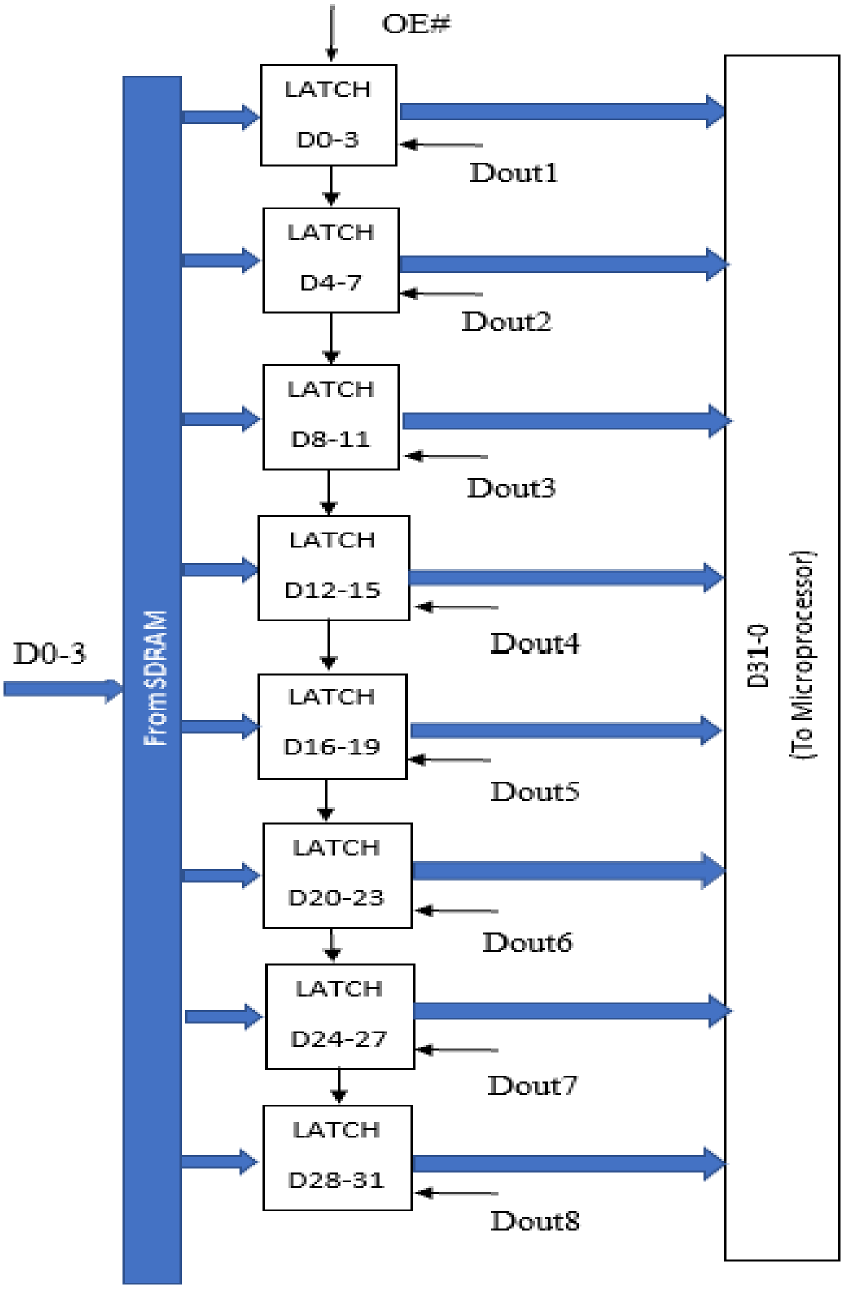


CMD = fn (Read, Write)

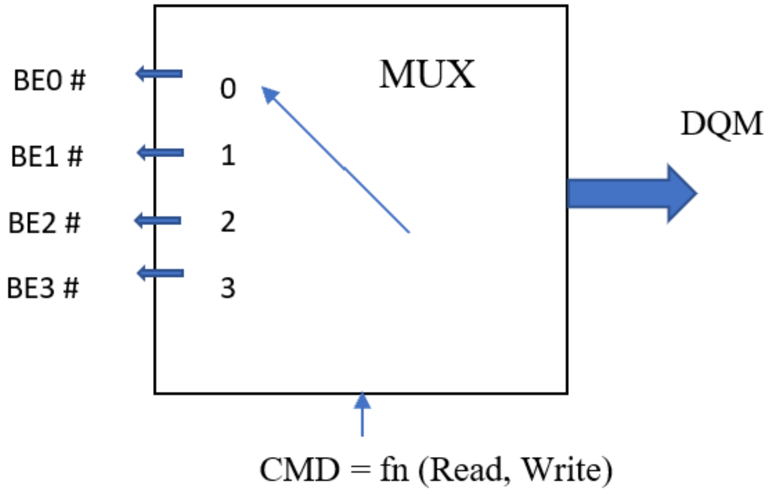
STATE = Dout_7 & Dout_8 & NOP_{24} || Din_7 & Din_8 & NOP_{26} & NOP_{27} & NOP_{28}

- **Data LATCH for Read :**

STATE = Read (Dout 1 to Dout 8)



- **DQM Signal:**



0 – STATE = Read & NOP23 || Din 1 & Din 2
1 - STATE = Dout 1 & Dout 2 || Din 3 & Din 4
2 – STATE = Dout 3 & Dout 4 || Din 5 & Din 6
3 – STATE = Dout 5 & Dout 6 || Din 7 & Din 8