UNIVERSITY OF TEXAS ARLINGTON | COLLEGE OF ENGINEERING

EE 5313: Microprocessor Systems

SPRING – 2023

INSTRUCTOR: Dr. Shahriar Shahabuddin

PROJECT TITLE:

Design of an SDRAM Controller for Asynchronous Memory Interface

Submitted By:

NAME	UTA ID
RANJITH THYLAGERE CHENNEGOWDA	1002068028
PRAJWAL GOWDA RAMPURA SURESH	1002064165

Date of Submission: May 1st, 2023

INDEX:

INTRODUCTION

SDRAM OVERVIEW:

- Synchronous Dynamic Random-Access Memory is a form of DRAM semiconductor memory which can run at faster speeds than conventional DRAM.
- SDRAM is synchronous interface RAM, which refers to the fact that it waits for the clock signal and usually accepts command at the rising edge of the clock.
- As the SDRAM is synchronized with the CPU's clock, this tends to increase the throughput and allows the SDRAM to run at higher speeds.
- The memory will be partitioned into banks and hence multiple bits of data can be accessed at the same time.
- SDRAM generally performs three main functions Read, Write and Refresh.
- Clocks during when CS is high otherwise ignored.

GOAL:

The goal of this project is to design an SDRAM controller that allows SDRAM memory to be interfaced with a microprocessor having only asynchronous memory support. This SDRAM controller is to be designed such that it can interface with one or more SDRAM memory devices. The minimum requirement is that it should support for a single MT48LC16M4A2 (SDRAM Memory) and interfacing is done by using 80386DX processor and we may interface with another32-bit data bus processor provided SDRAM memory support is not already provided for that processor.

The detailed requirements are as follows:

1. Support the MT48LC16M4A2 memory device is required.

2. Support for an 80386DX or similar processor with 32-bit data bus not having SDRAM support is required.

3. A complete controller solution including state machine, row, column, and bank signal generation, data masking, data flow, ready logic, and refresh support must be provided.

4. Interfacing with ~ADS, W/~R, and M/~IO control signals (or equivalent) will be required.

5. Support the sending of AUTO REFRESH commands at a rate sufficient to ensure memory integrity.

6. Support the READY logic of the selected microprocessor system, especially when refresh is occurring, unless a bus locking protocol is used.

7. Support is only required for burst length 8 transfers.

8.Extra Credit Steps.

a. Adding hardware external I/O control pins to support variable timing, with changes in both FSM timing and the LMR values being loaded.

b. Adding hardware external I/O to support 8- and 16-bit wide memories with burst lengths of 4 and 2 appropriately with all changes needed to the FSM and SDRAM interface.

FEATURES OF SDRAM

- The Micron 64Mb SDRAM is a high-speed CMOS, dynamic random-access memory containing 67,108,864 bits.
- It is a 4-Bank DRAM with a synchronous interface.
- Signals are sampled during the positive edge of the clock cycle.
- Configured as 4 Meg x 4 x 4 banks with 4096 rows and 1024 columns.
- Programmable with burst lengths of 1, 2, 4, 8, or full page. In this project we have used Burst length=8 and CAS Latency (CL) = 2.
- Refresh-64ms, 4,096-cycle refresh (15.6µs/row) for commercial applications and16ms, 4,096-cycle refresh (3.9µs/row) for automotive applications.
- Bank Address = 4 (BA0, BA1).
- Row Address = A0 – A11.
- Column Address = A0 – A9.

Timing values used in the project.

Timing Parameter	Timing/Parameter
T_{rfc}	66ns
T_{rp}	15ns
T_{mrd}	2 clock cycles
T_{wr}	14ns (7ns + 1clk)
T_{rcd}	15ns
CL (CAS Latency)	2
T_{ras}	5-16000
T_{rc}	8

SPECIFICATIONS OF 80386DX

- 32-Bit Address Bus.
- 32-Bit Data Bus.
- 66 MHz Clk Signal.

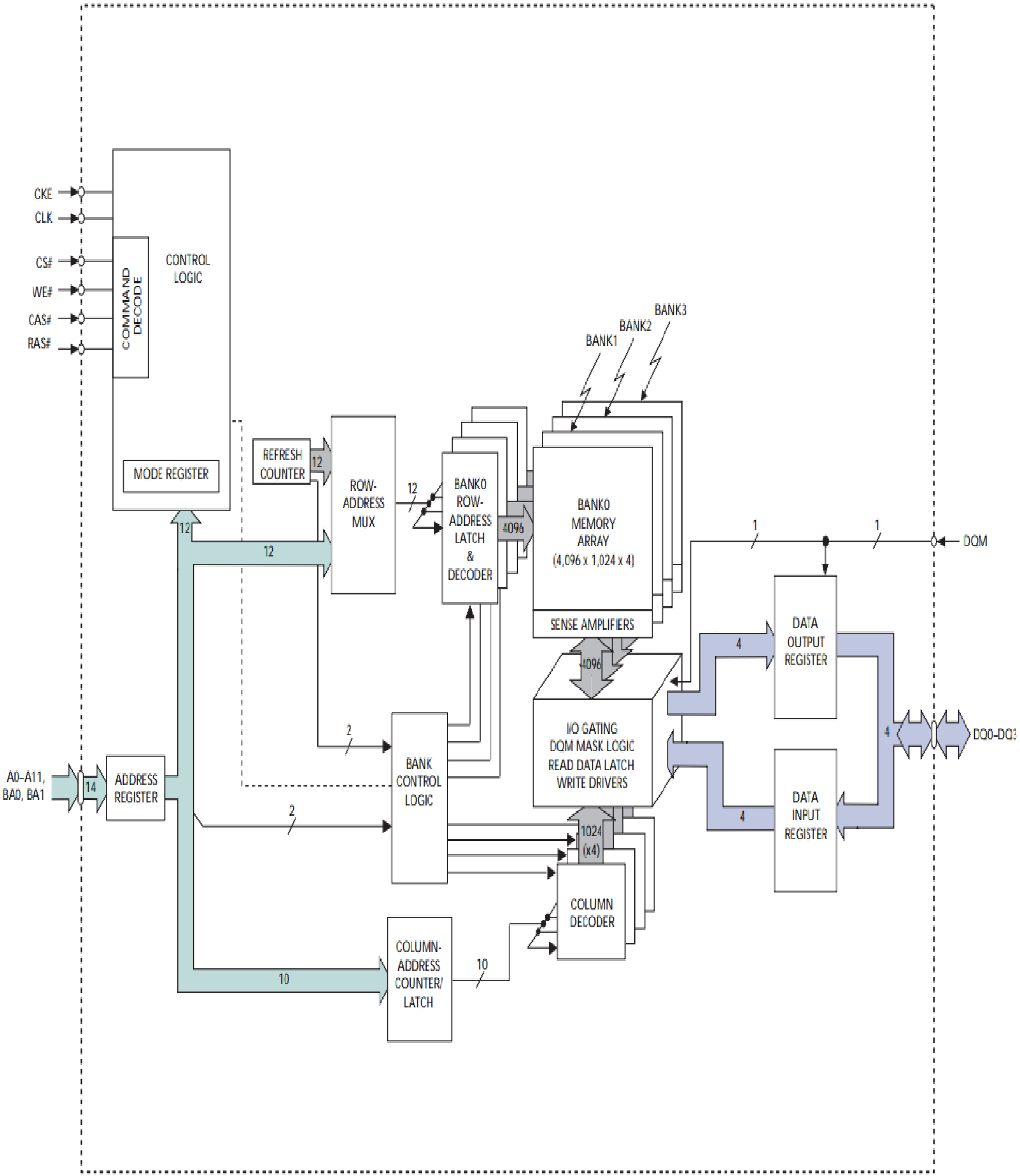
Outgoing Signals From processor

- Address Bus A31- 0.
- Data Bus A31- 0.
- Control Signals.

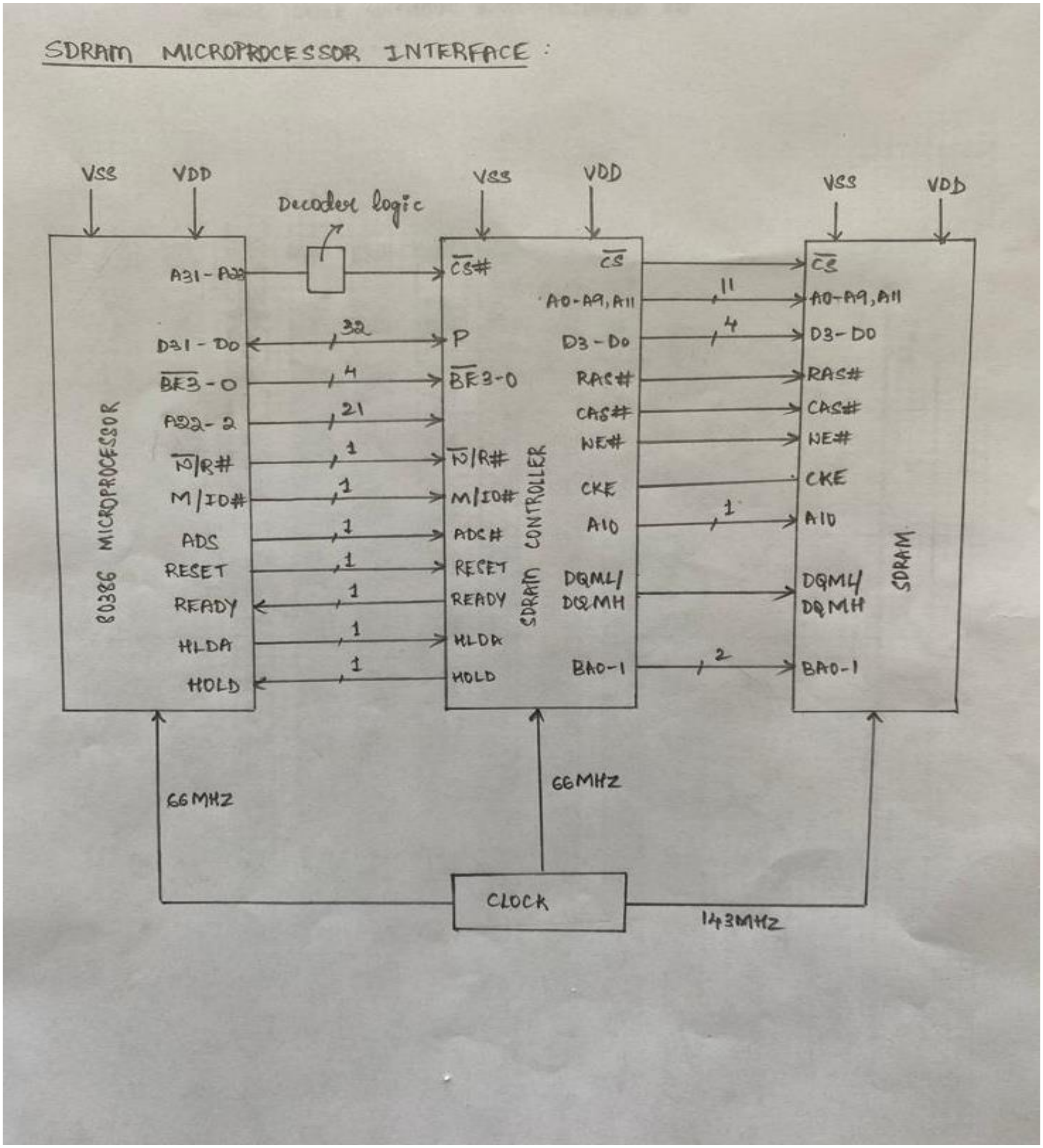
Incoming Signal to Processor

- Ready Signal

FUNCTIONAL BLOCK DIAGRAM.



SDRAM – MICROPROCESSOR Interface.



Address Bus Interfacing

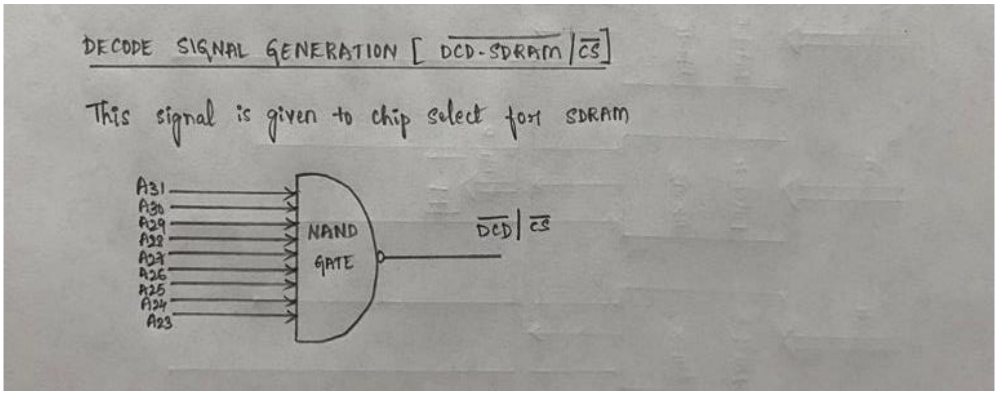
- The 80386DX has a 32 – Bit wide address bus and 32 – Bit wide data bus.

DCD_SDRAM	Bank Enable	Row_Address	Column_Address	X X
A31 – A23	A22 – A21	A20 – A9	A8 – A2	A0, A1

- The SDRAM that we are interfacing is 64Mb.

64Mb = 2^{26} Bits = 2^{23} Bytes

Log$_2$ (2^{23}) = So 23 Address lines are used to address memory.

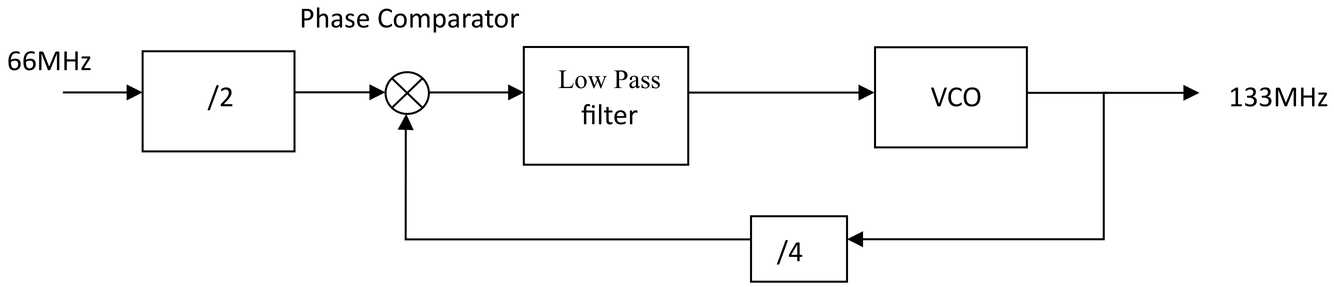


DECODE SIGNAL GENERATION [$\overline{\text{DCD-SDRAM}}$ / $\overline{\text{CS}}$]

This signal is given to chip select for SDRAM

PHASE LOCKED LOOP.

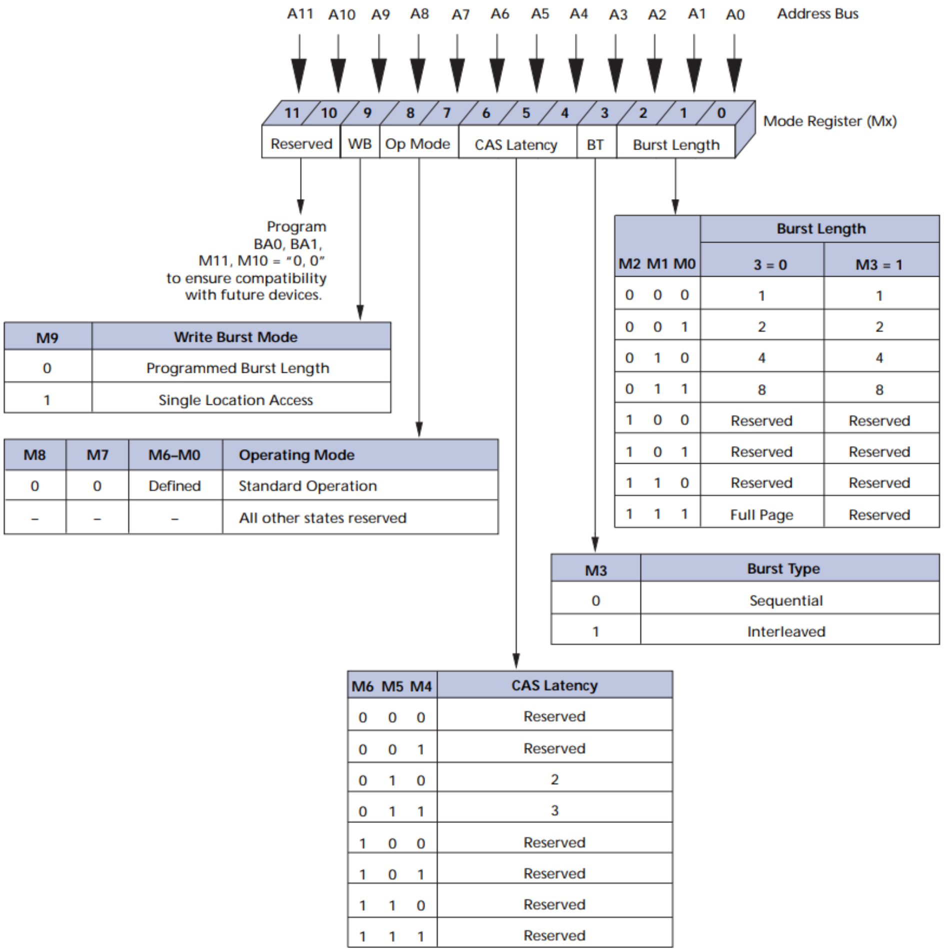
- The Phase Locked loop consists of a Low Pass Filter, Voltage controlled Oscillator and a Phase comparator (XOR Gate)
- This circuit is used until the desired frequency of 133MHz is obtained as the SDRAM requires a clock frequency of 133MHz to operate.
- It provides a Stable Clock to the SDRAM.



Clock Cycle Period = (1 / 133MHz) = 7.5ns

LOAD MODE REGISTER

- In our design CAS latency (CL) is 2, burst length (BL) is 8.
- Therefore, CAS latency (CL) = 010 and Burst length (BL) = 011.



M9	Write Burst Mode
0	Programmed Burst Length
1	Single Location Access

M8	M7	M6–M0	Operating Mode
0	0	Defined	Standard Operation
–	–	–	All other states reserved

	Burst Length	
M2 M1 M0	3 = 0	M3 = 1
0 0 0	1	1
0 0 1	2	2
0 1 0	4	4
0 1 1	8	8
1 0 0	Reserved	Reserved
1 0 1	Reserved	Reserved
1 1 0	Reserved	Reserved
1 1 1	Full Page	Reserved

M3	Burst Type
0	Sequential
1	Interleaved

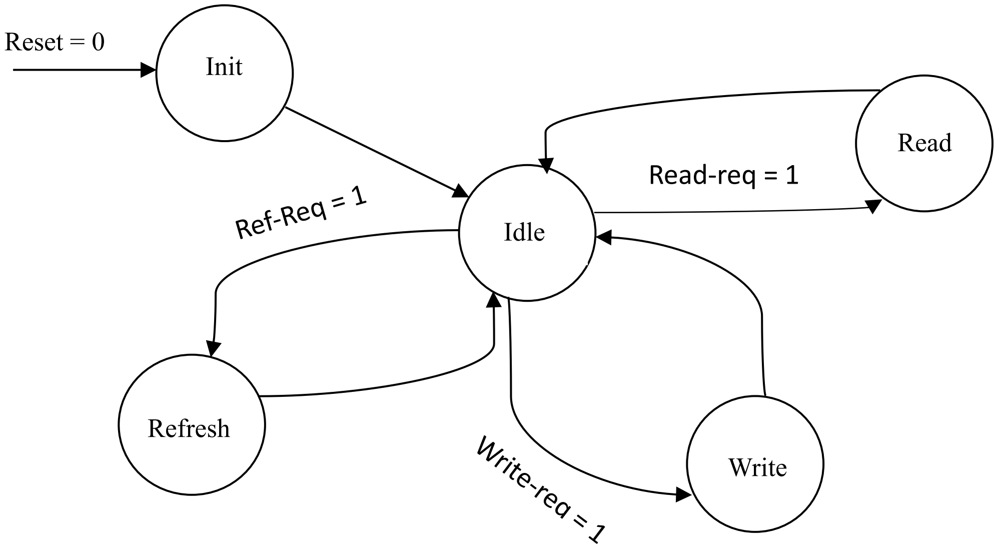
M6 M5 M4	CAS Latency
0 0 0	Reserved
0 0 1	Reserved
0 1 0	2
0 1 1	3
1 0 0	Reserved
1 0 1	Reserved
1 1 0	Reserved
1 1 1	Reserved

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		WB	Op Mode		CAS Latency			BT	Burst Length		
Value	0	0	0	0	0	0	1	0	0	0	1	1

Program: M11, M10: 0, 0: To ensure compatibility with future device

- Write Burst Mode: M9: 0: Program Burst Length
- Operating Mode: M8, M7: 0, 0: Standard Operation
- CAS latency: M6, M5, M4: 0, 1, 0: Two Clock Cycle
- Burst Type: M3: 0: Sequential
- Burst length: M2, M1, M0: 0, 1, 1.

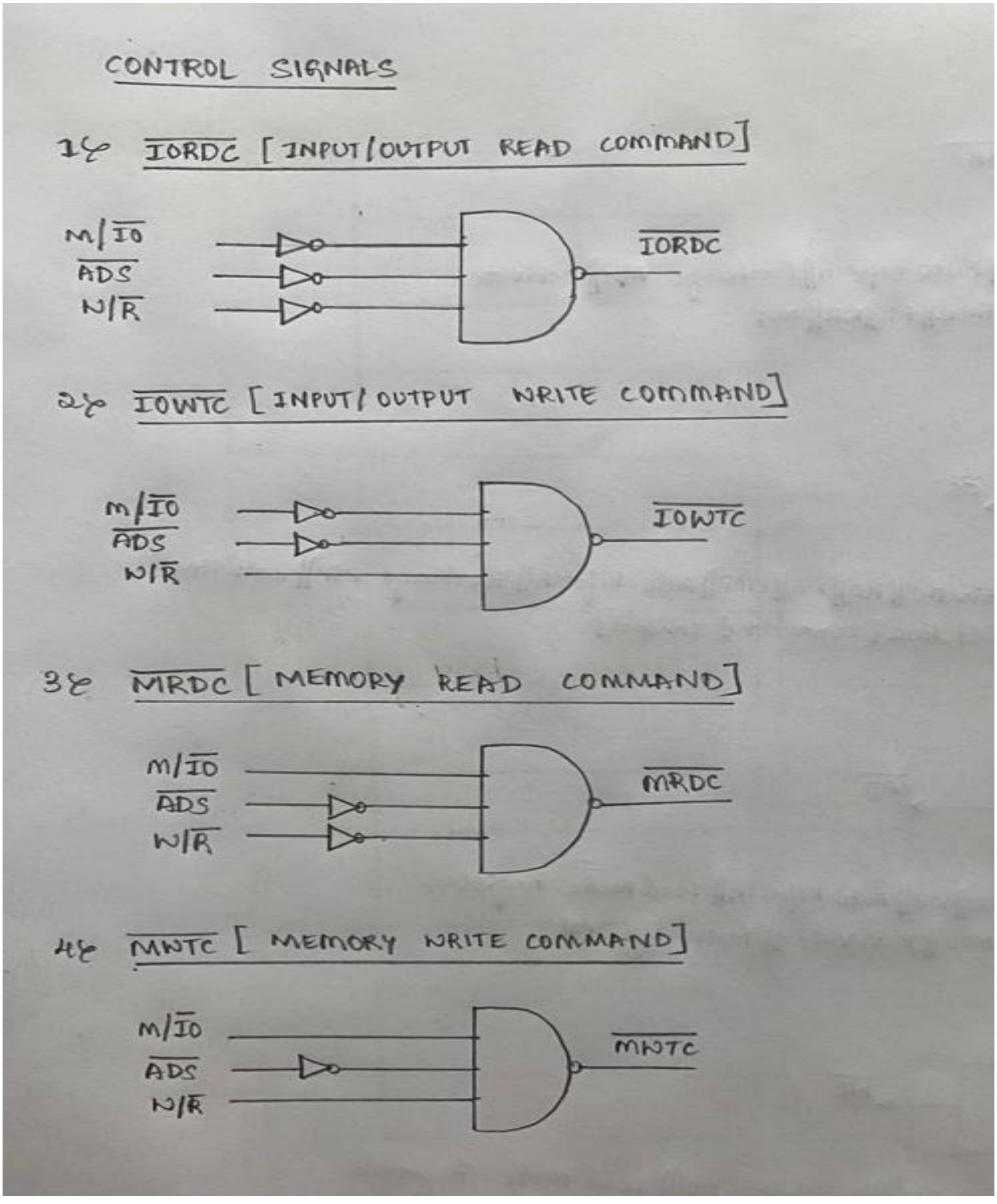
HIGHER LEVEL FINITE STATE MACHINE (FSM)



STATE TRANSITION TABLE FOR HIGHER LEVEL (FSM)

PRESENT STATE	SIGNAL CONDITION	NEXT STATE
X	Reset = 0	Init
Init	X	Idle
Idle	Read-Req = 1	Read
Read	X	Idle
Idle	Write-Req = 1	Write
Write	X	Idle
Idle	Ref-Req = 1	Refresh
Refresh	X	Idle

GENERATION OF CONTROL SIGNALS



CONTROL SIGNALS

1) $\overline{IORDC}$ [INPUT/OUTPUT READ COMMAND]

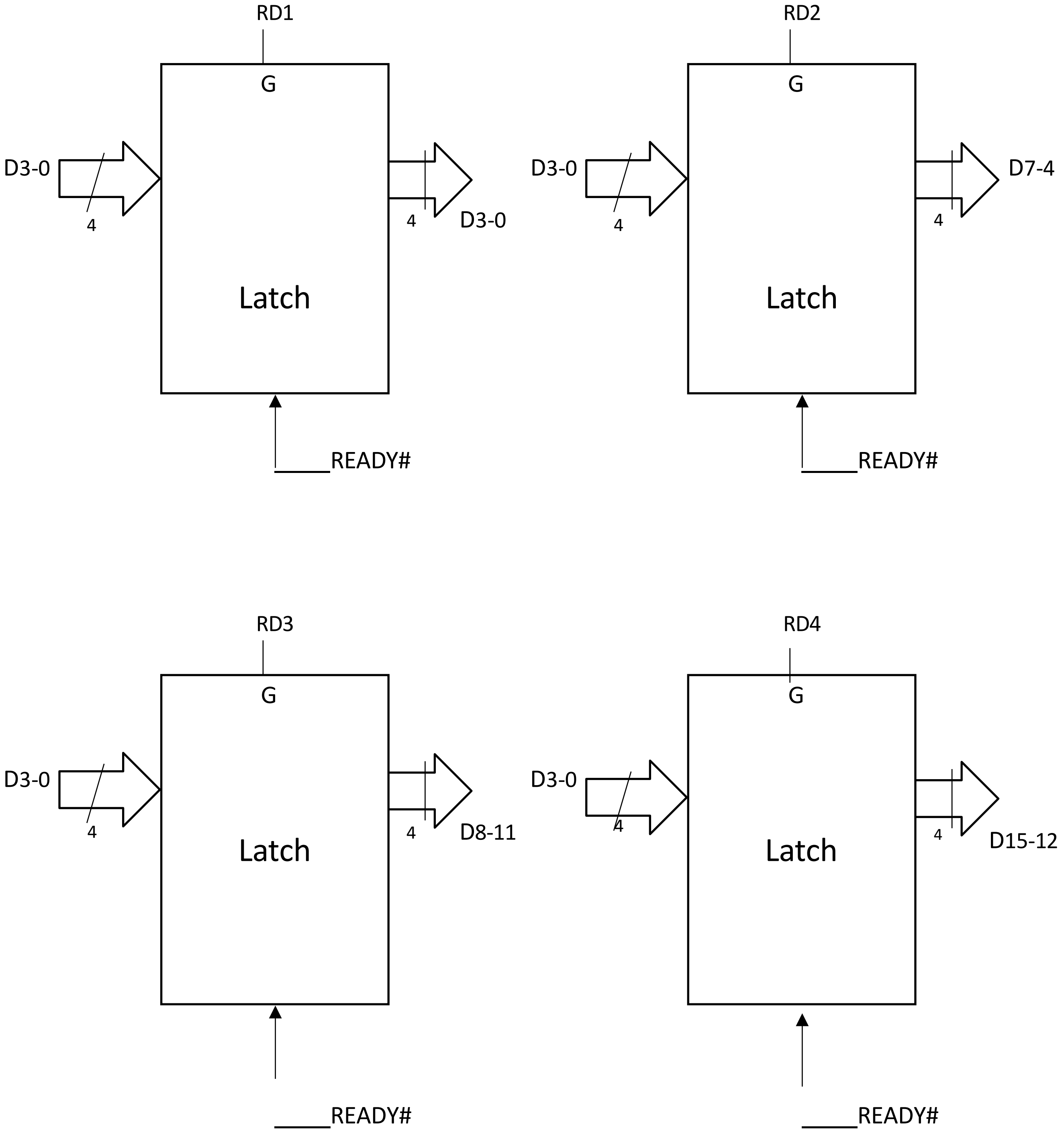
2) $\overline{IOWTC}$ [INPUT/OUTPUT WRITE COMMAND]

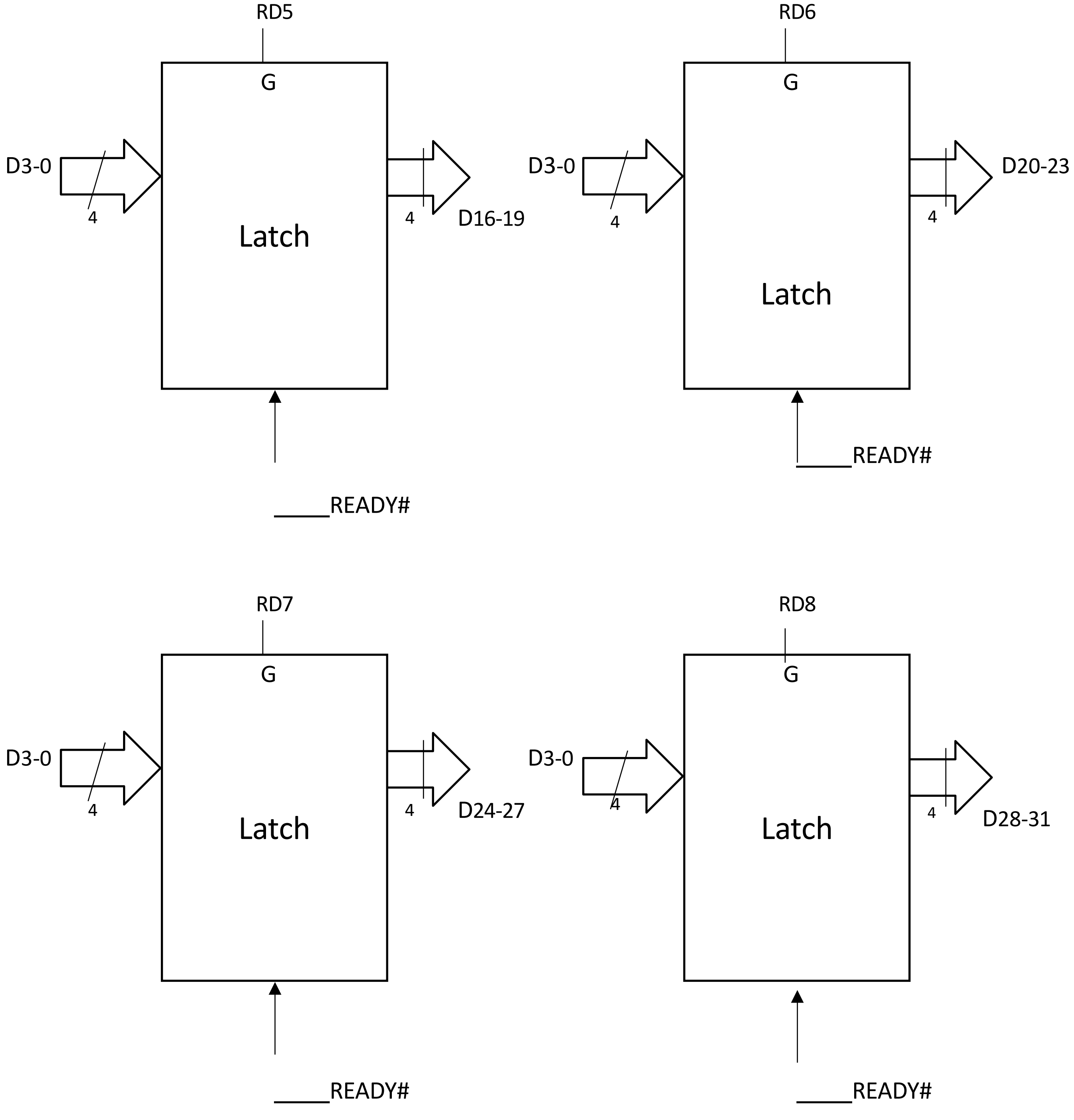
3) $\overline{MRDC}$ [MEMORY READ COMMAND]

4) $\overline{MWTC}$ [MEMORY WRITE COMMAND]

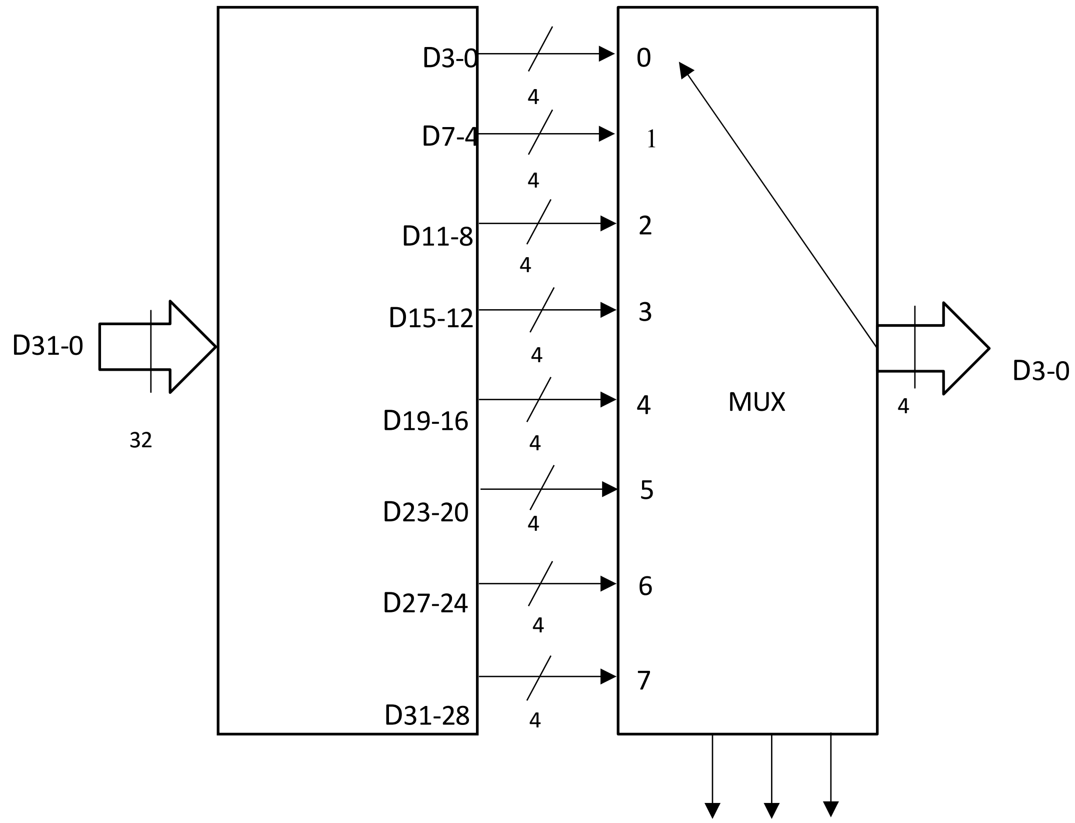
Generation of Chip Signals:

- **Read Data Latch:**




RD5
G
D3-0
4
Latch
4
D16-19
READY#
RD6
G
D3-0
4
Latch
4
D20-23
READY#
RD7
G
D3-0
4
Latch
4
D24-27
READY#
RD8
G
D3-0
4
Latch
4
D28-31
READY#

Write Data Latch:



0: STATE = WR_1

1: STATE = WR_2

2: STATE = WR_3

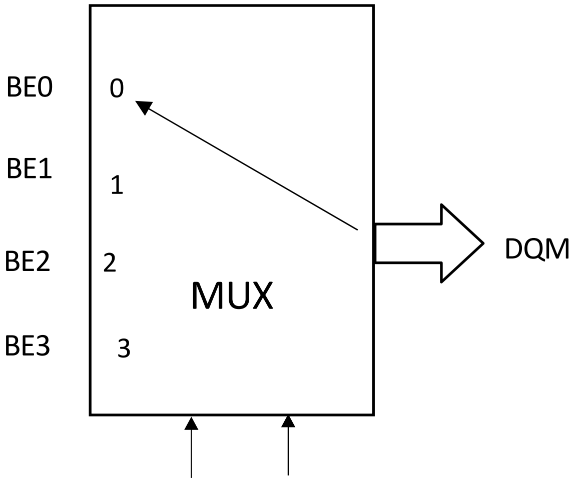
3: STATE = WR_4

4: STATE = WR_5

5: STATE = WR_6

6: STATE = WR_7

7: STATE = WR_8



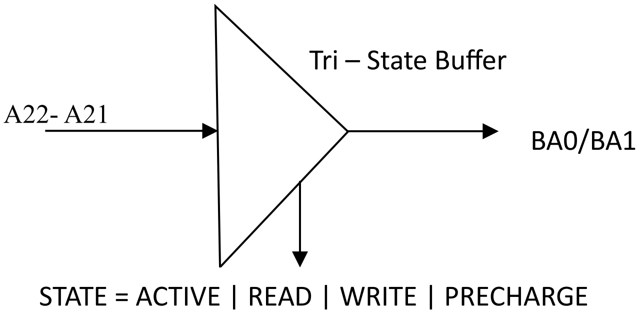
0: STATE = READ | NOP | WR_1 | WR_2

1: STATE = RD_1 | RD_2 | WR_3 | WR_4

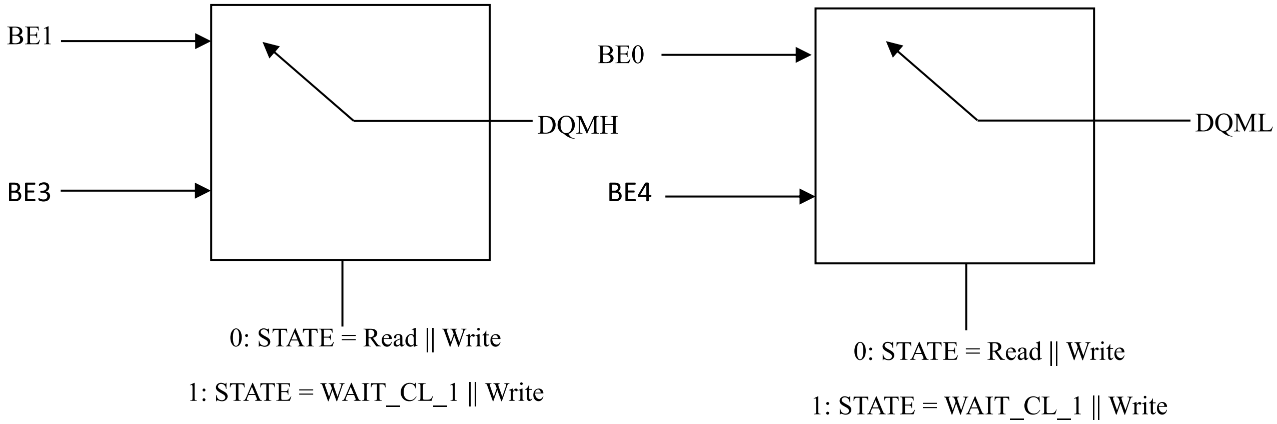
2: STATE = RD_3 | RD_4 | WR_5 | WR_6

3: STATE = RD_5 | RD_6 | WR_7 | WR_8

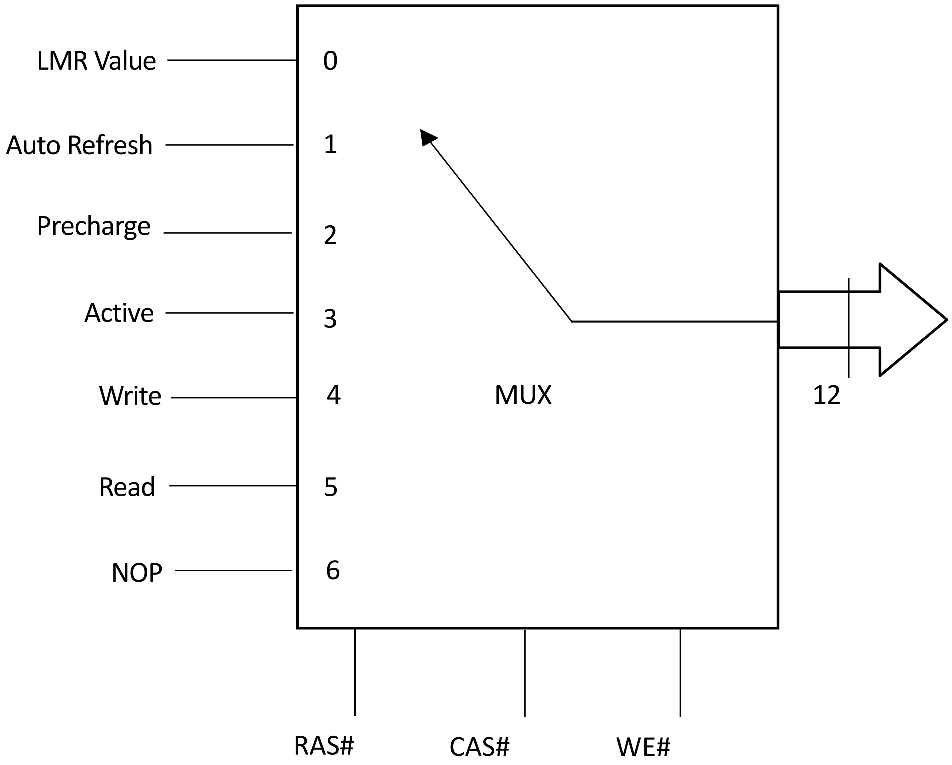
Generation of Bank Signals.



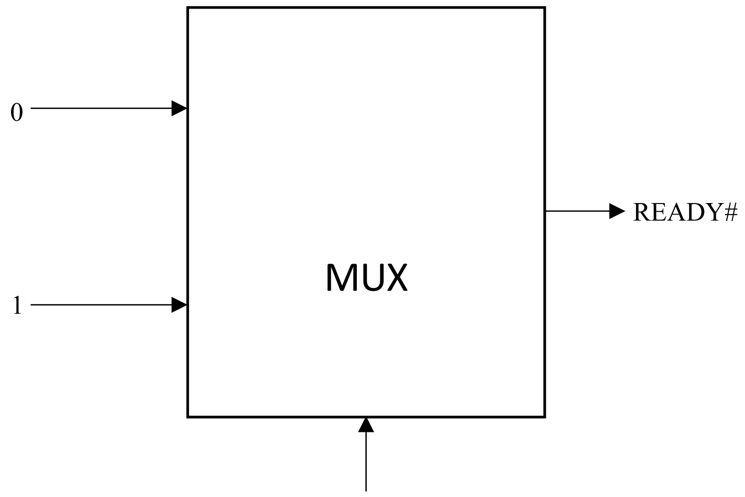
Generation of DQM/ DQMH, DQML Signals.



Generation of Command Signal



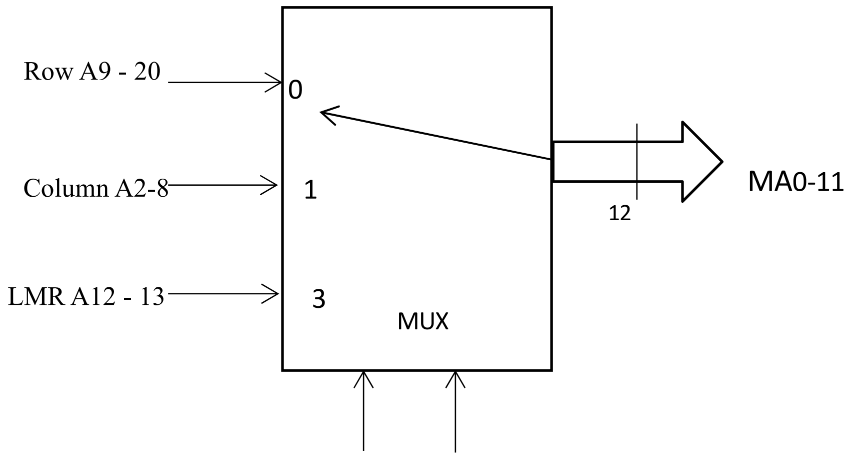
Generation of Ready Signal.



1 = STATE! = Read | Write

0 = STATE= Read | Write

Generation of ROW and Column Address

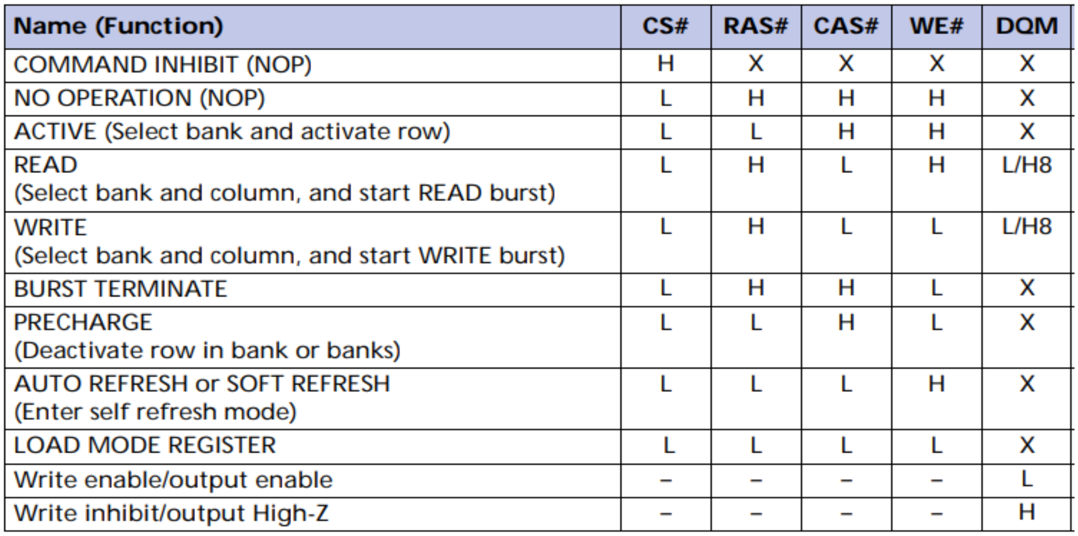


0: STATE = ACTIVE (ROW)

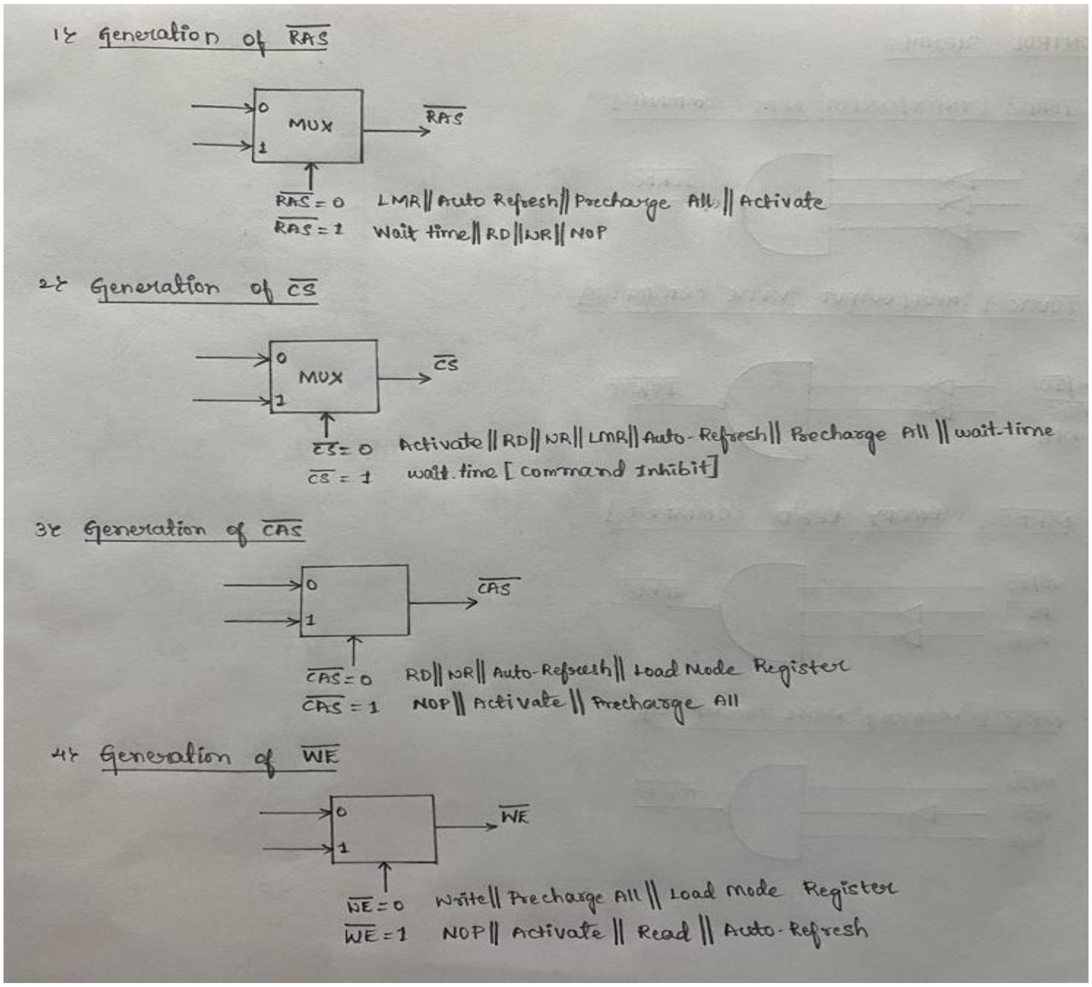
1: STATE = READ | WRITE

2: STATE = LMR

GENERATION OF SDRAM CONTROL SIGNALS



Name (Function)	CS#	RAS#	CAS#	WE#	DQM
COMMAND INHIBIT (NOP)	H	X	X	X	X
NO OPERATION (NOP)	L	H	H	H	X
ACTIVE (Select bank and activate row)	L	L	H	H	X
READ (Select bank and column, and start READ burst)	L	H	L	H	L/H8
WRITE (Select bank and column, and start WRITE burst)	L	H	L	L	L/H8
BURST TERMINATE	L	H	H	L	X
PRECHARGE (Deactivate row in bank or banks)	L	L	H	L	X
AUTO REFRESH or SOFT REFRESH (Enter self refresh mode)	L	L	L	H	X
LOAD MODE REGISTER	L	L	L	L	X
Write enable/output enable	–	–	–	–	L
Write inhibit/output High-Z	–	–	–	–	H



1> Generation of $\overline{RAS}$

$\overline{RAS}=0$ LMR || Auto Refresh || Precharge All || Activate

$\overline{RAS}=1$ Wait time || RD || WR || NOP

2> Generation of $\overline{CS}$

$\overline{CS}=0$ Activate || RD || WR || LMR || Auto-Refresh || Precharge All || wait-time

$\overline{CS}=1$ wait. time [command Inhibit]

3> Generation of $\overline{CAS}$

$\overline{CAS}=0$ RD || WR || Auto-Refresh || Load Mode Register

$\overline{CAS}=1$ NOP || Activate || Precharge All

4> Generation of $\overline{WE}$

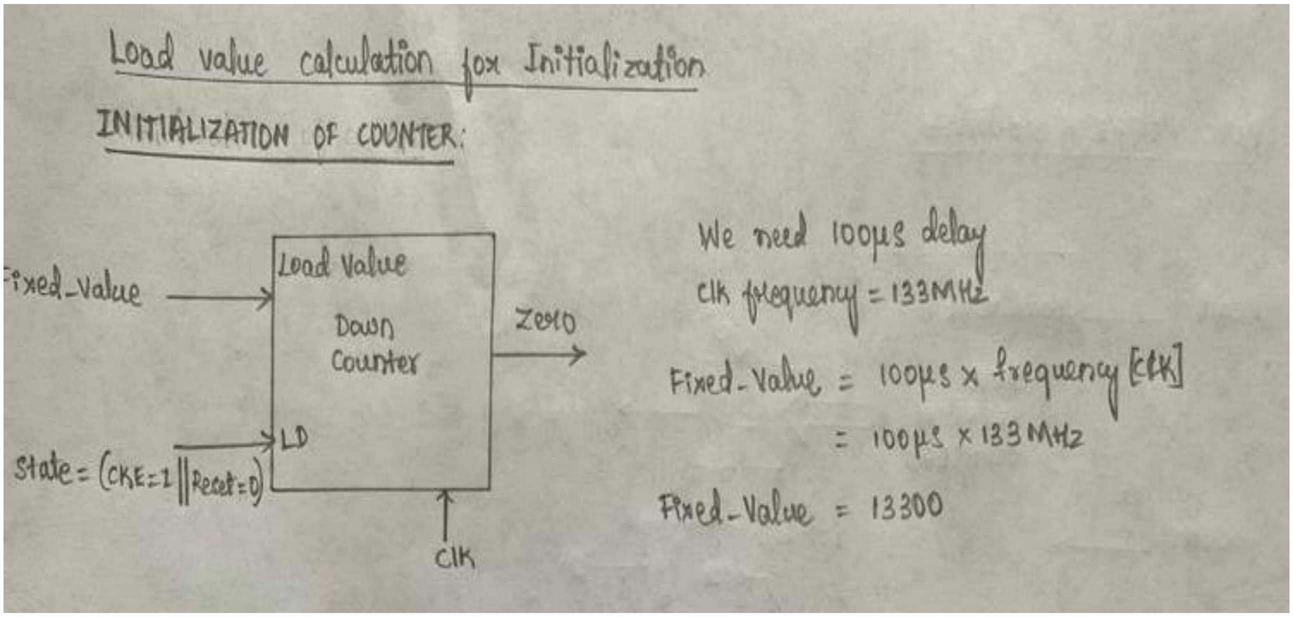
$\overline{WE}=0$ Write || Precharge All || Load mode Register

$\overline{WE}=1$ NOP || Activate || Read || Auto-Refresh

LOAD VALUE CALCULATION AND COUNTER

- We need a delay of 100us delay.
- The clock frequency is 133MHz.

Fixed_Value = 100µs x Frequency [Clk]
Fixed_Value = 100µs x 133MHz = 13300.
Clock frequency = 132 MHz
So, T_{clk} = 1/132Mhz = 7.5 ns
Therefore, T_{rfc} is given as: T_{rfc} = 66/7.5 = 8.8clk = 9clk



Load value calculation for Initialization

INITIALIZATION OF COUNTER:

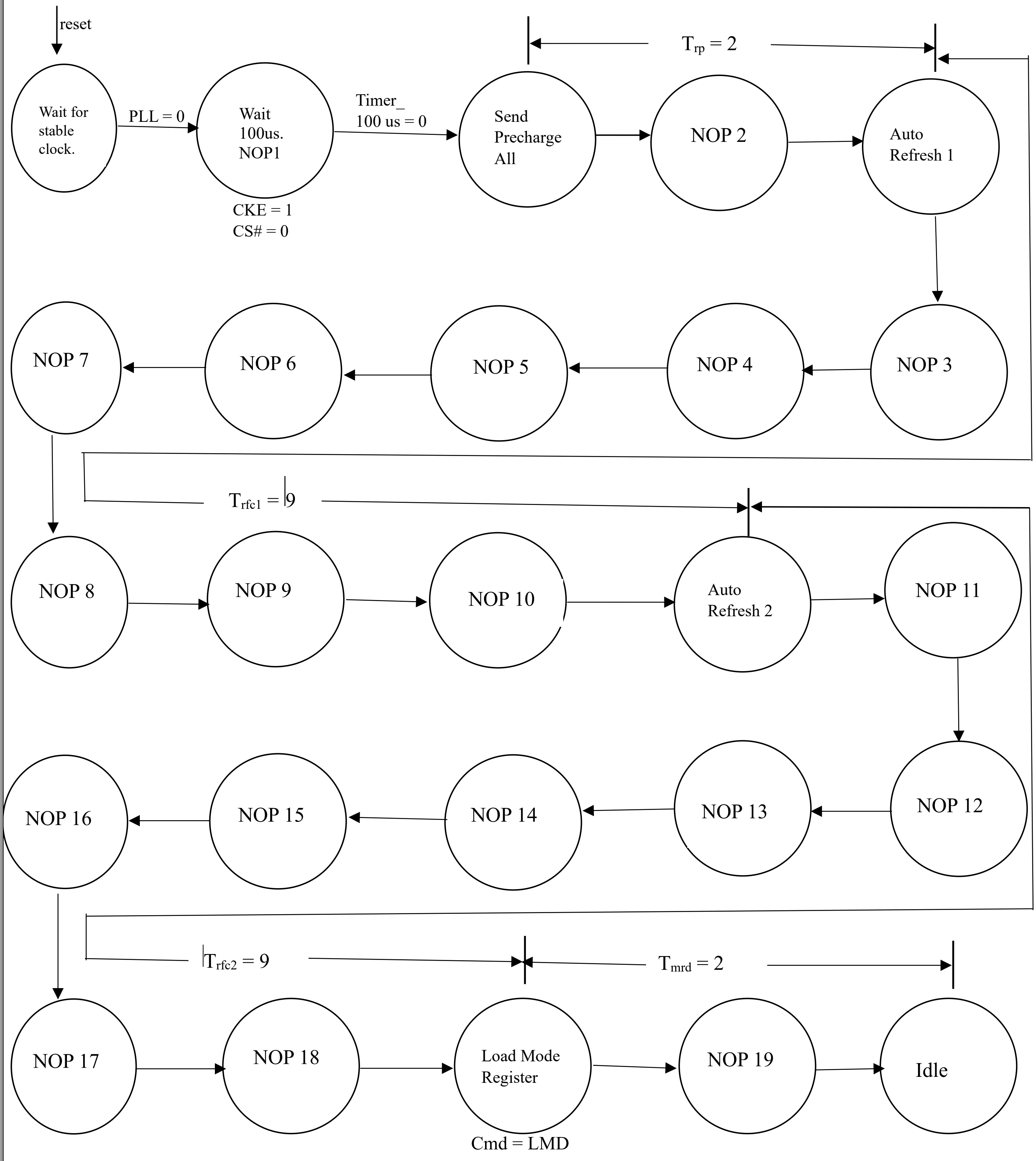
We need 100µs delay
clk frequency = 133MHz

Fixed-Value = 100µs x frequency [clk]
= 100µs x 133 MHz

Fixed-Value = 13300

Fixed_Value
Load Value
Down Counter
Zero
LD
State = (CKE=1 || Reset=0)
clk

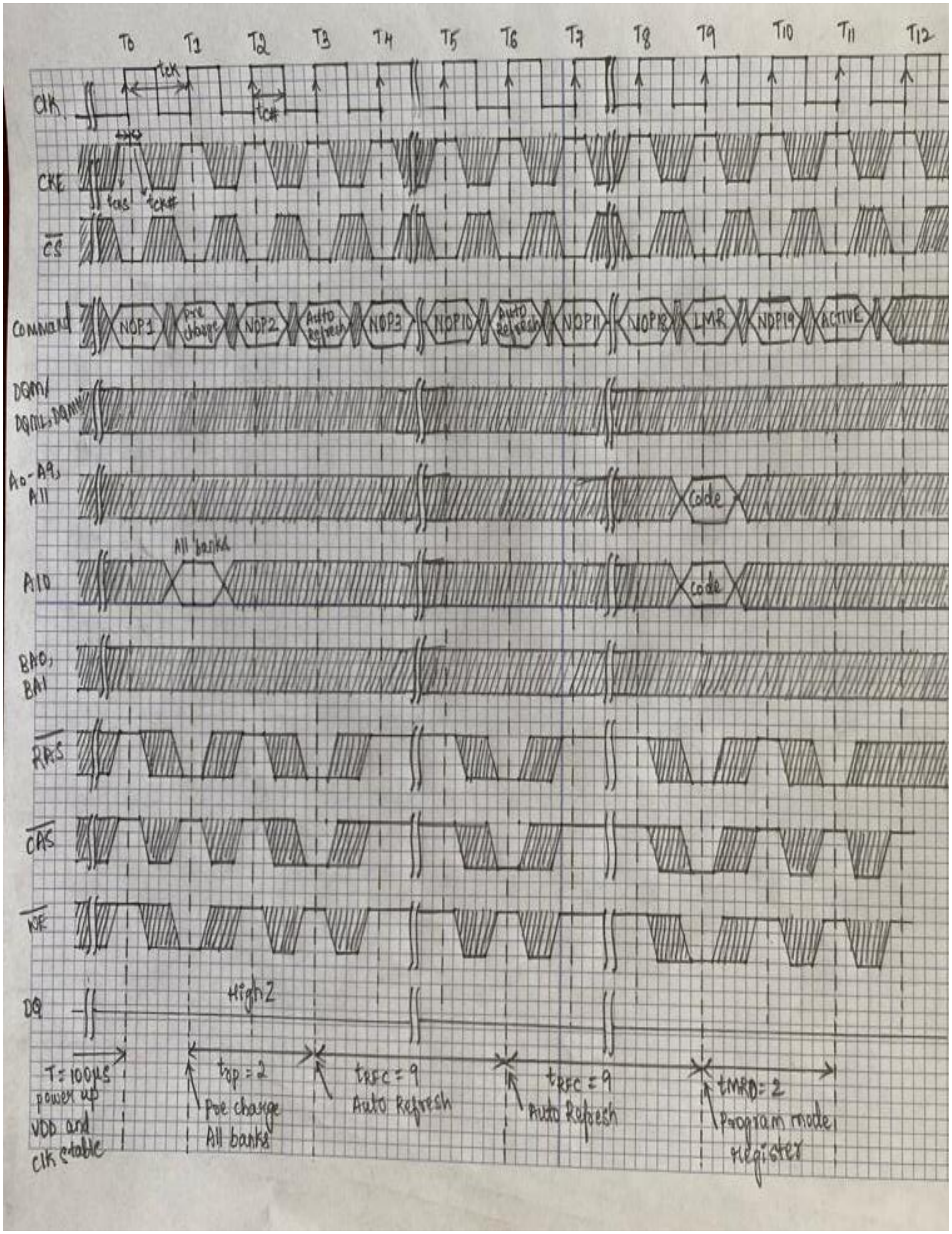
FSM FOR INITIALIZATION OF SDRAM



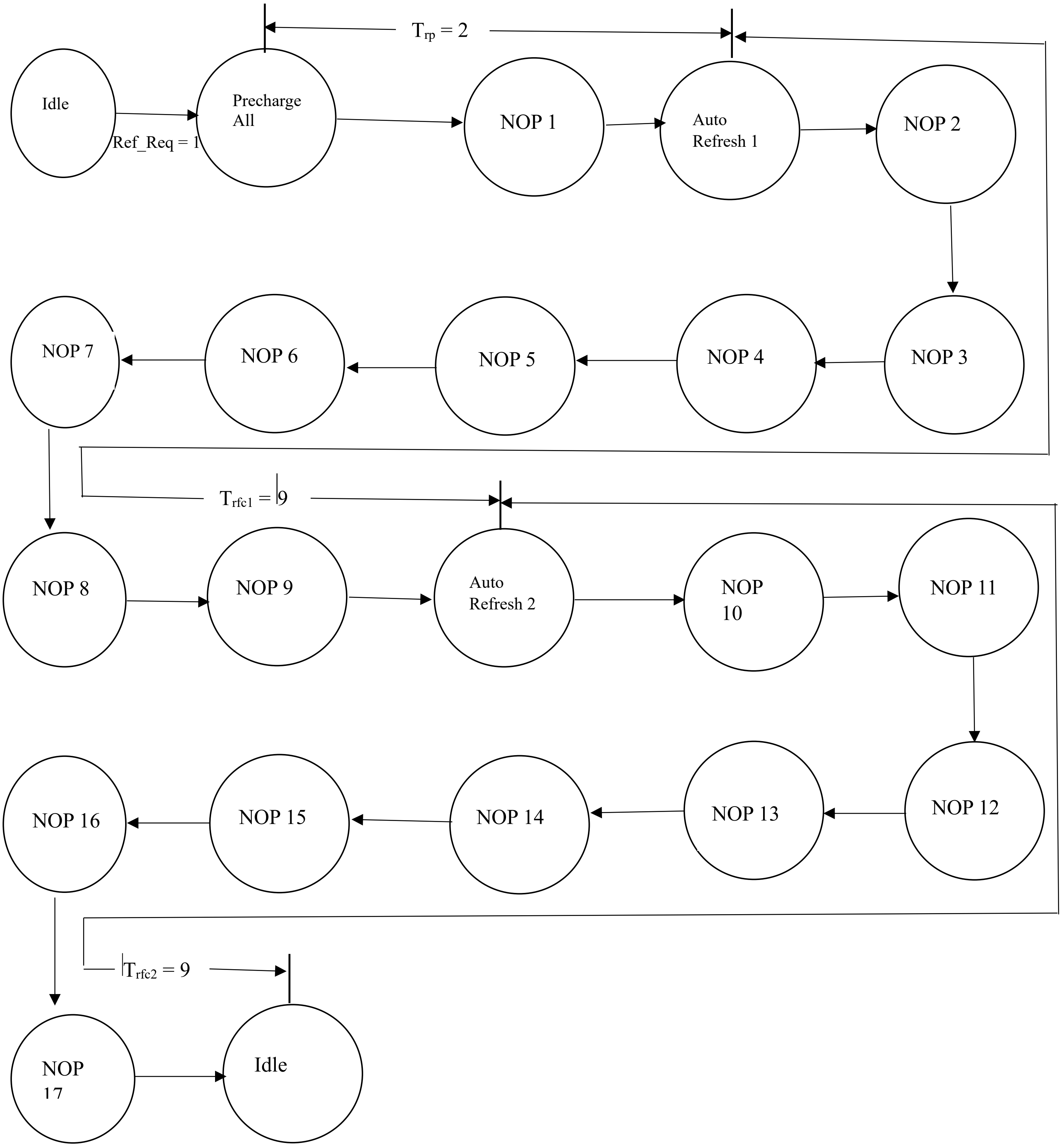
STATE TRANSITION TABLE FOR INITIALIZATION

PRESENT STATE	SIGNAL CONDITION	DESCRIPTION	NEXT STATE
X	Power up	-	Reset
Reset	Next Clock	-	Wait for stable clock
Wait for stable clock	! PLL = 0	Wait until stable clock is generates.	Wait for stable clock
Wait for stable clock	PLL = 0	Stable clock generation successful.	NOP1
NOP1	! Timer_100us = 0	Wait until 100us delay is completed.	NOP1
NOP1	Timer_100us = 0	100us delay completed move to next state.	Precharge ALL
Precharge ALL	Next clock	Wait for a time of T_{rp}, hence during this interval wait states are added.	NOP2
NOP2	Next clock	-	Auto refresh 1
Auto refresh 1	Next clock	Wait for a time of T_{rfc} and wait states are added if needed.	NOP3
NOP3 . . . . NOP9	Next clock . . . . Next clock	-	NOP4 . . . . Auto refresh 2
Auto refresh 2	Next clock	Wait for a time of T_{rfc} and wait states are added if needed.	NOP11
NOP11 . . . NOP18	Next clock . . . Next clock	-	NOP12 . . . LMR
LMR	Next clock	Wait for a time T_{mrd} and wait states are added accordingly.	NOP19
NOP19	Next clock	-	Idle

TIMING DIAGRAM FOR INITIALIZATION



FINITE STATE MACHINE FOR REFRESH



TRANSITION TABLE FOR REFRESH

PRESENT STATE	SIGNAL CONDITION	DESCRIPTION	NEXT STATE
Idle	Ref_req = 1	Enter precharge state if there is a refresh request	Precharge_ALL
Precharge_ALL	Next clock	Wait for a time of T_{rp}	NOP1
NOP1	Next clock	-	Auto refresh 1
Auto refresh 1	Next clock	Wait for a time of T_{rfc}	NOP2
NOP2 . . . NOP9	Next clock . . . Next clock	-	NOP3 . . . Auto refresh 2
Auto refresh 2 . . . NOP17	Next clock . . . Next clock	-	NOP10 . . . Idle

COUNTER FOR REFRESH SIGNAL

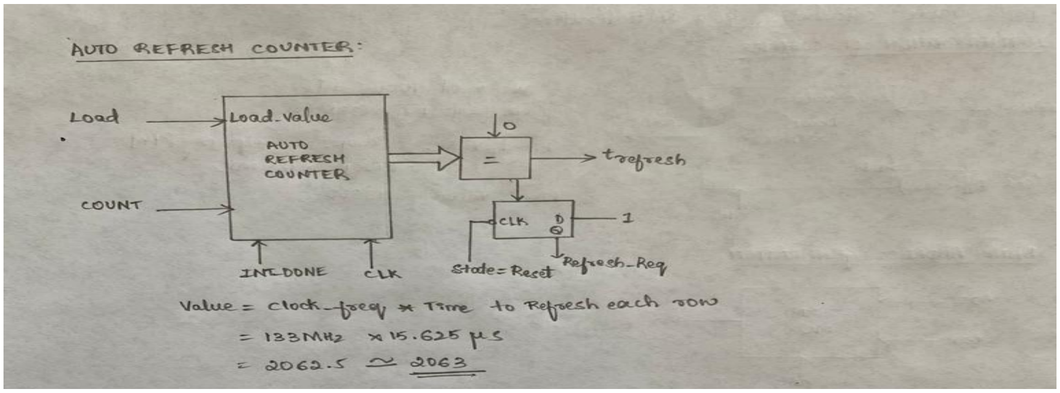
Val = Clock_Freq x Time_To_Refresh_Each_Row = 133MHz x 15.625us

Val = 2062.5(can be rounded of to 2063) = 2063

Here, CL=2, T_{rcd} = 2 Clk

BL = 8 T_{rc} = 2+2+8 = 12clks

Therefore, ACTIVE-to-ACTIVE command period T_{rc} = 12 x 7.5 = 90ns

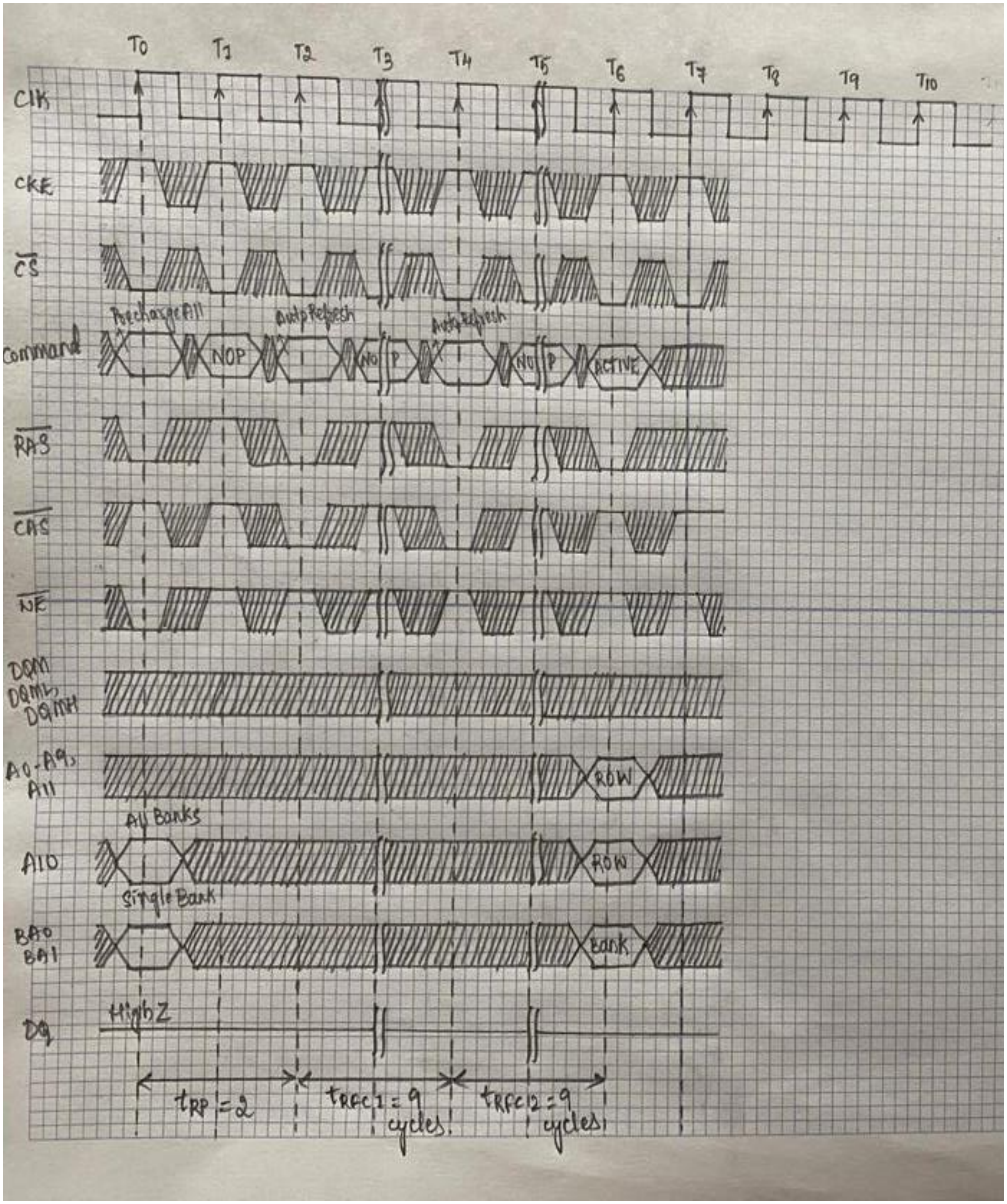


Value = clock_freq * Time to Refresh each row

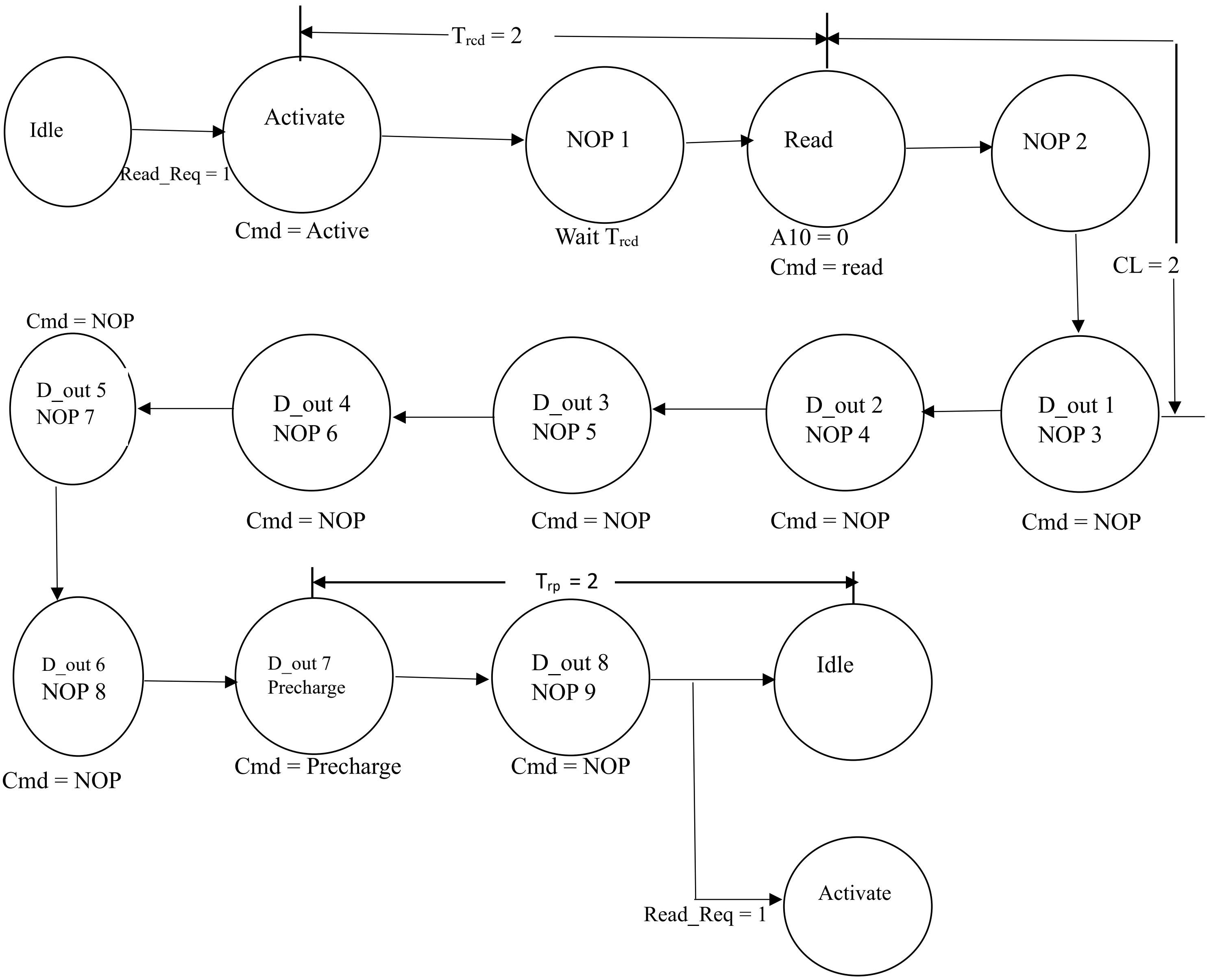
= 133MHz × 15.625 μs

= 2062.5 ≃ 2063

TIMING DIAGRAM FOR AUTO REFRESH



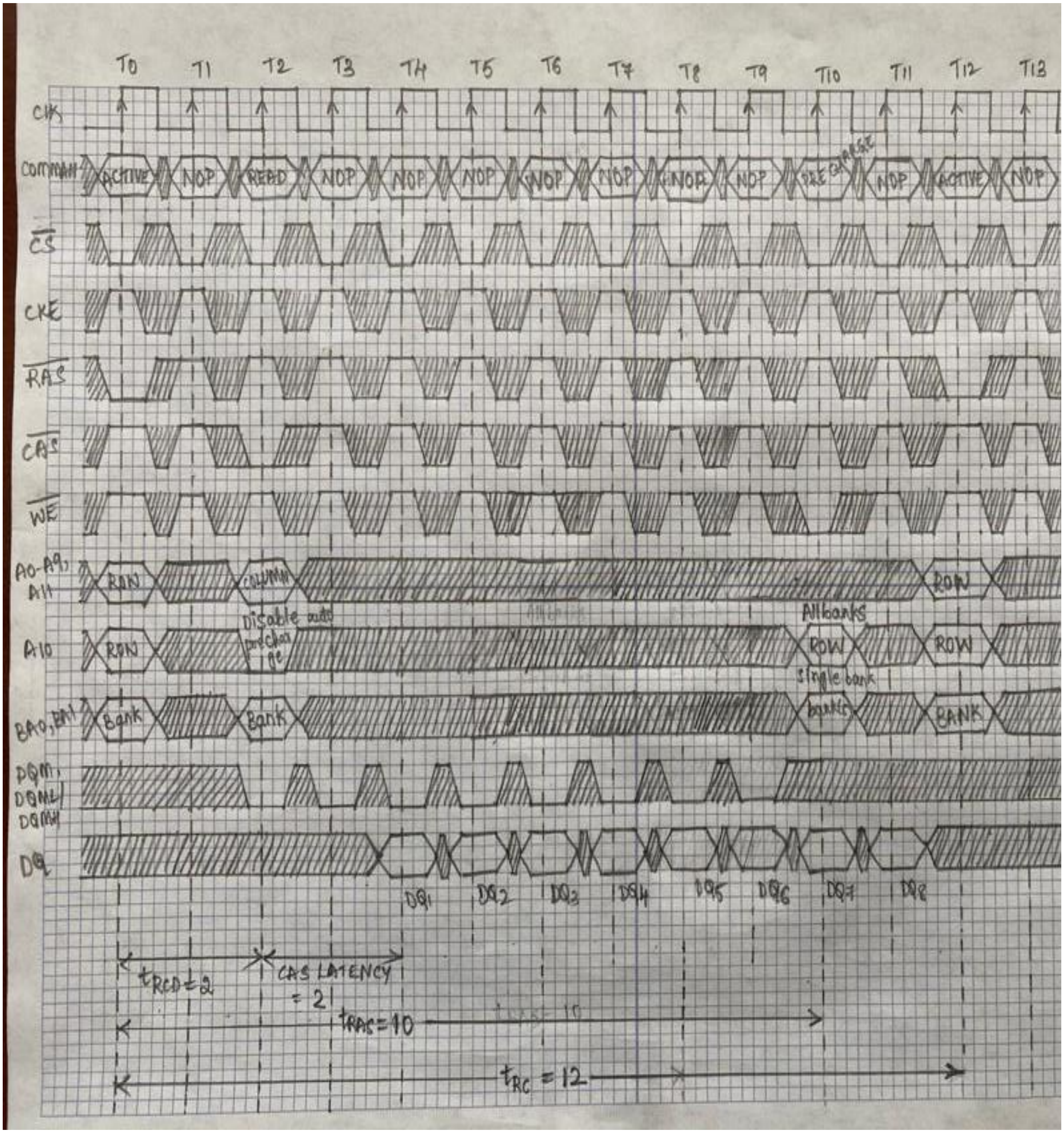
FSM FOR READ WITHOUT AUTO-PRECHARGE



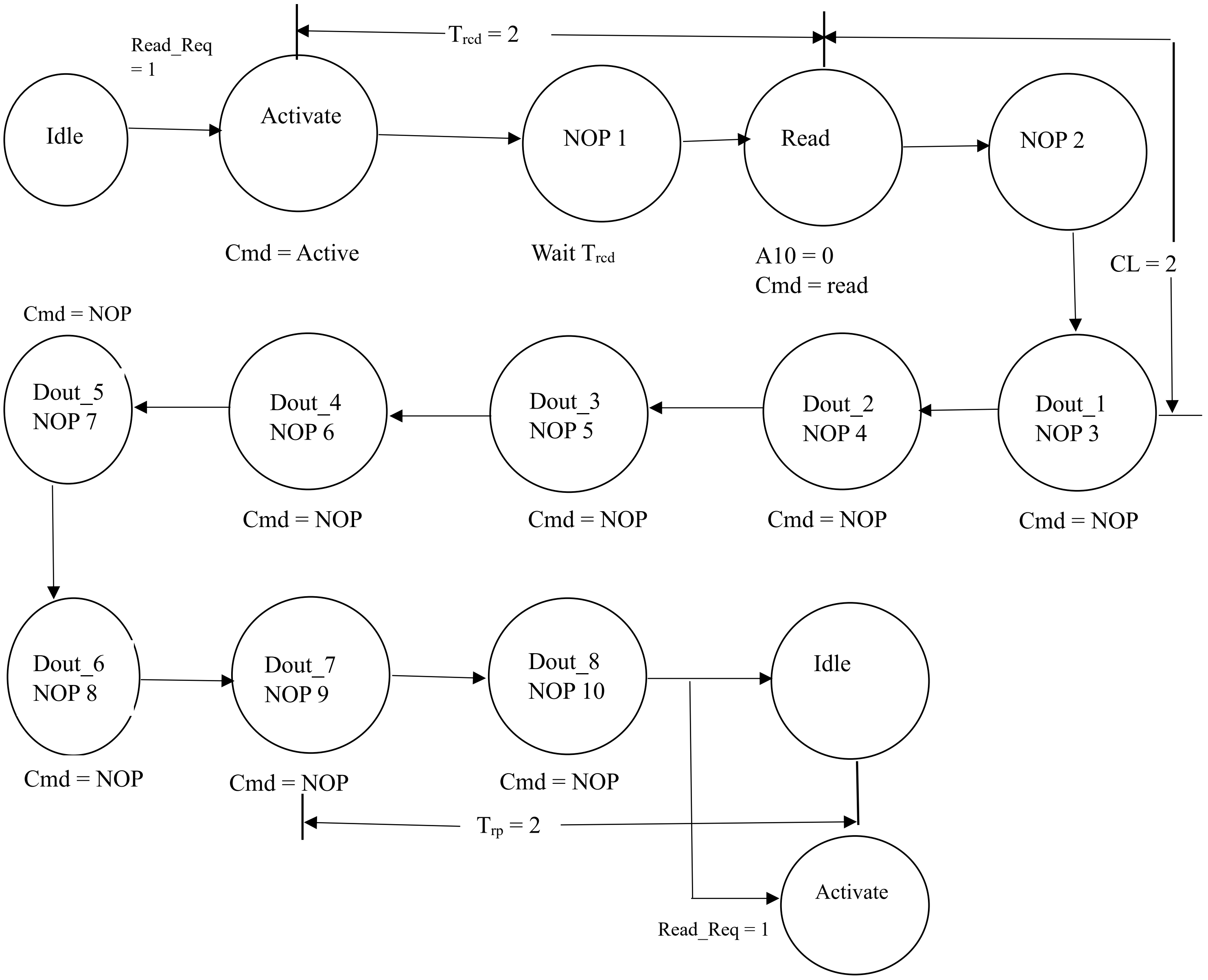
TRANSITION TABLE FOR READ WITHOUT AUTO-PRECHARGE

PRESENT STATE	SIGNAL CONDITION	DESCRIPTION	NEXT STATE
Idle	Read_req = 1	Activate only if read request	Activate
Activate	Next clock	-	NOP1
NOP1	Next clock	Start reading only if RD=1 A10=0 denotes auto pre-charge is disabled	Read
Read	Next clock	Wait for 2 clocks as the CL=2	NOP2
NOP2	Next clock	Data starts bursting for every clock pulse after CL=2	Data-out 1
Data-out 1	Next clock	-	Data-out 2
Data-out 2 . . . . Data-out 6	Next clock . . . . Next clock	-	Data-out 3 . . . . Data-out 7(Start of Precharge)
Data-out 7 (Start of Precharge) (T_{rp})	Next clock	-	Data-out 8 NOP9
Data-out 8 NOP9	Read_req != 1	-	Idle
Data-out 8 NOP9	Read_req = 1		Activate

TIMING DIAGRAM FOR READ WITHOUT AUTO-PRECHARGE



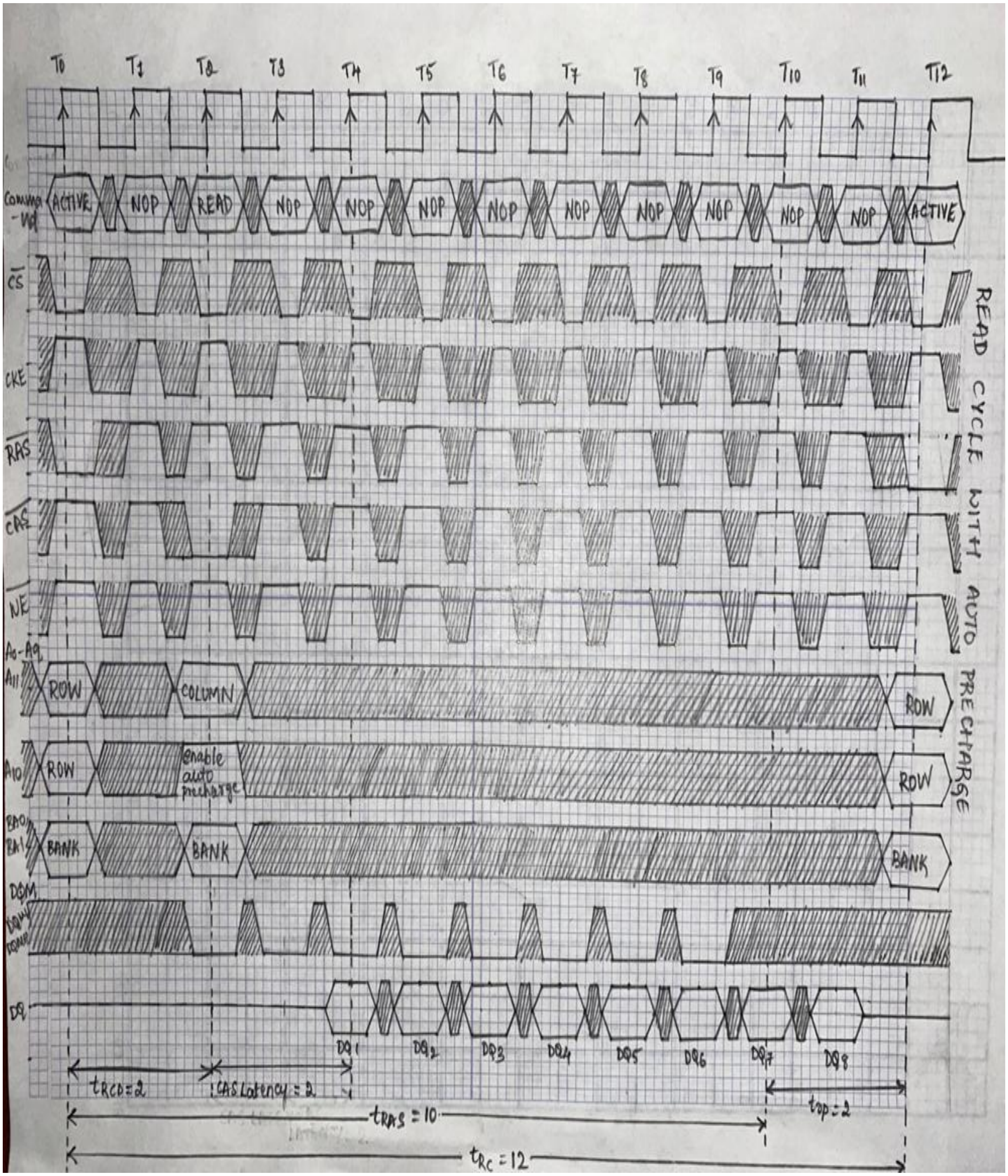
FSM FOR READ WITH AUTO-PRECHARGE



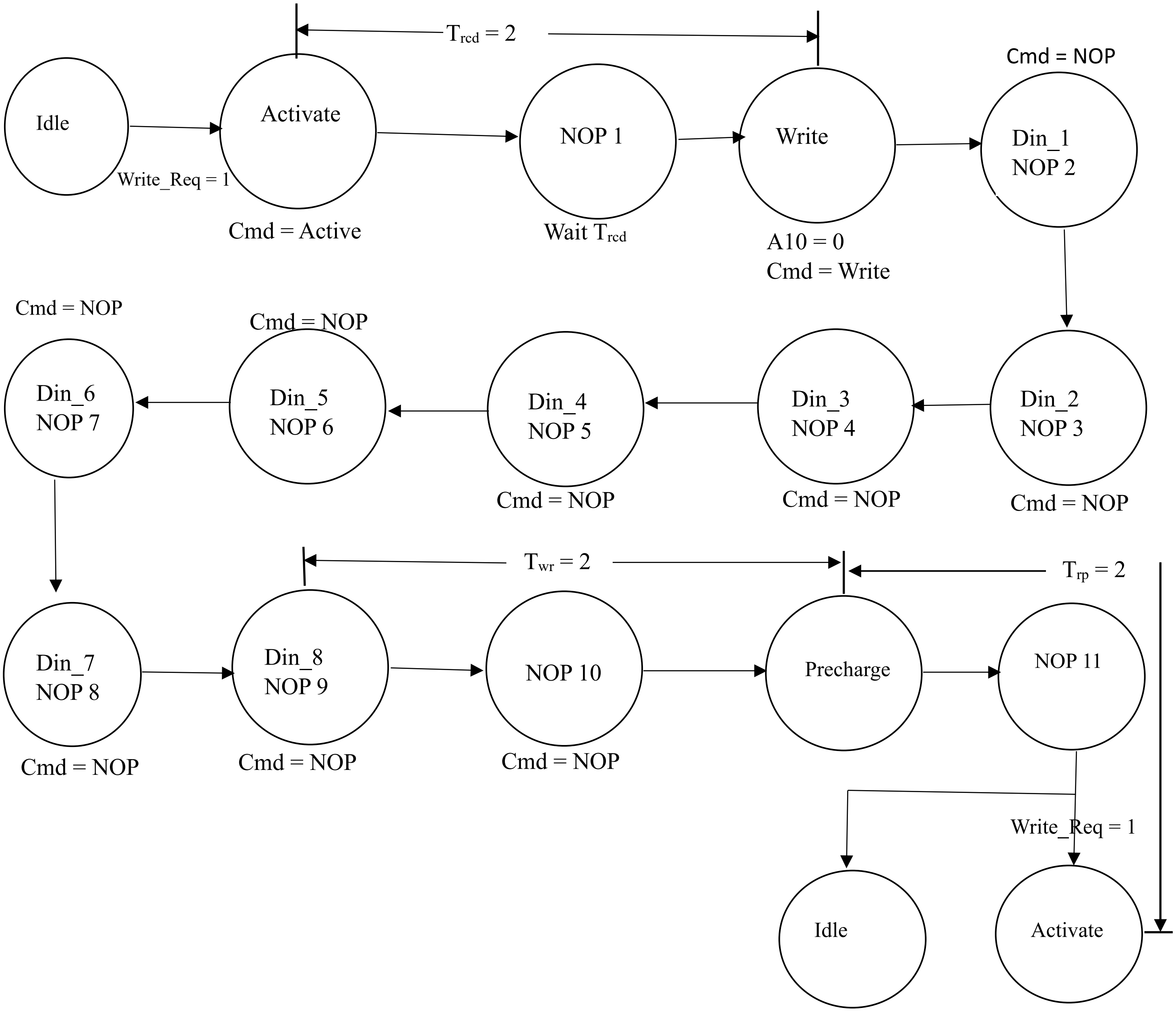
TRANSITION TABLE FOR READ WITH AUTO-PRECHARGE

PRESENT STATE	SIGNAL CONDITION	DESCRIPTION	NEXT STATE
Idle	Read_req = 1	Activate only if read request	Activate
Activate	Next clock	-	NOP1
NOP1	Next clock	Start reading only if RD=1 A10 = 1 denotes auto pre-charge is enabled.	Read
Read	Next clock	Wait for 2 clocks as the CL=2	NOP2
NOP2	Next clock	Data starts bursting for every clock pulse after CL=2	Data-out 1
Data-out 1	Next clock	-	Data-out 2
Data-out 2 . . . . Data-out 6	Next clock . . . . Next clock	-	Data-out 3 . . . . Data-out 7
Data-out 7	Next clock	-	Data-out 8
Data-out 8	Read_req != 1	-	Idle
Data-out 8	Read_req = 1	Activate only if read request	Activate

TIMING DIAGRAM FOR READ WITH AUTO-PRECHARGE



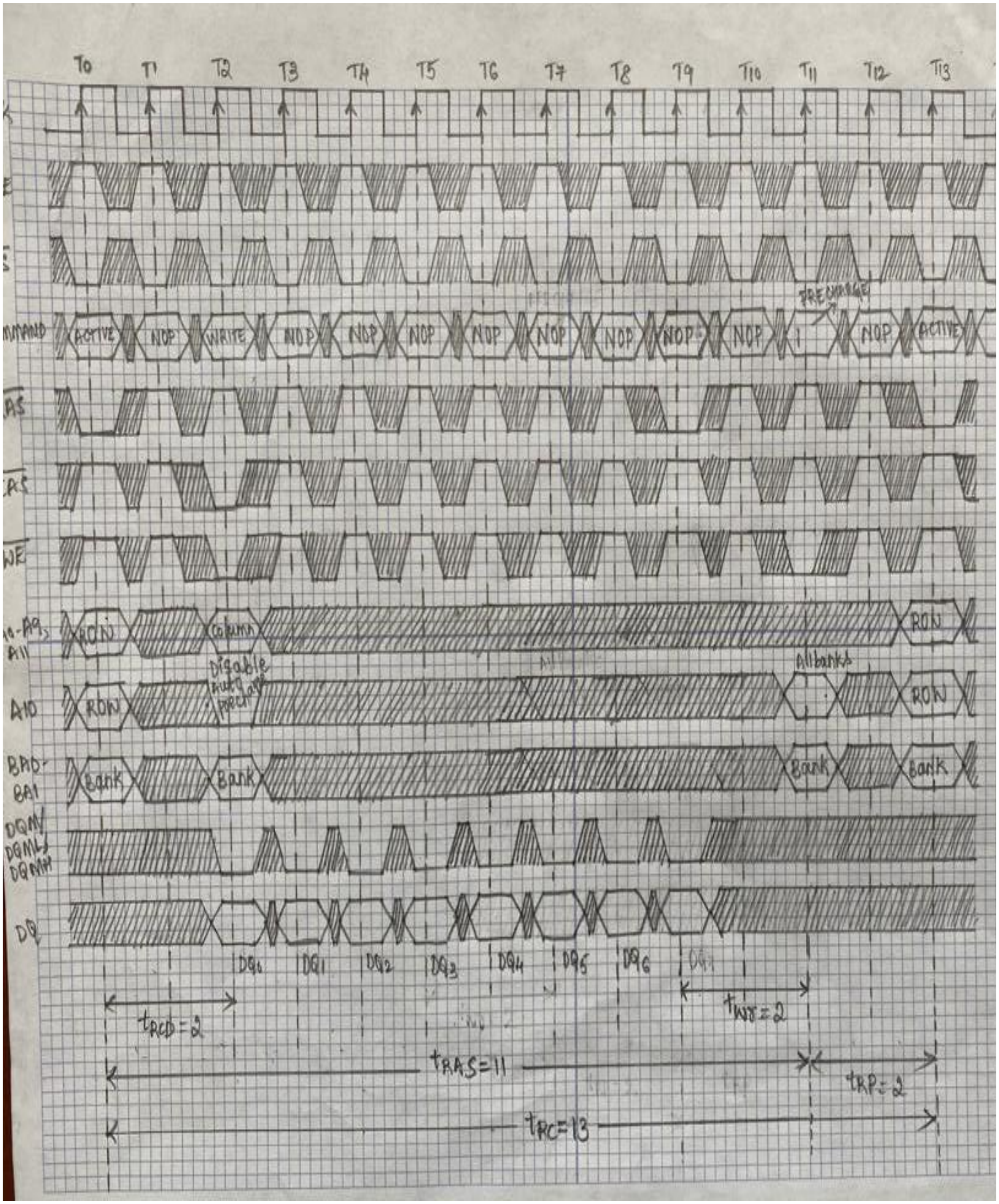
FSM FOR WRITE WITHOUT AUTO-PRECHARGE ON



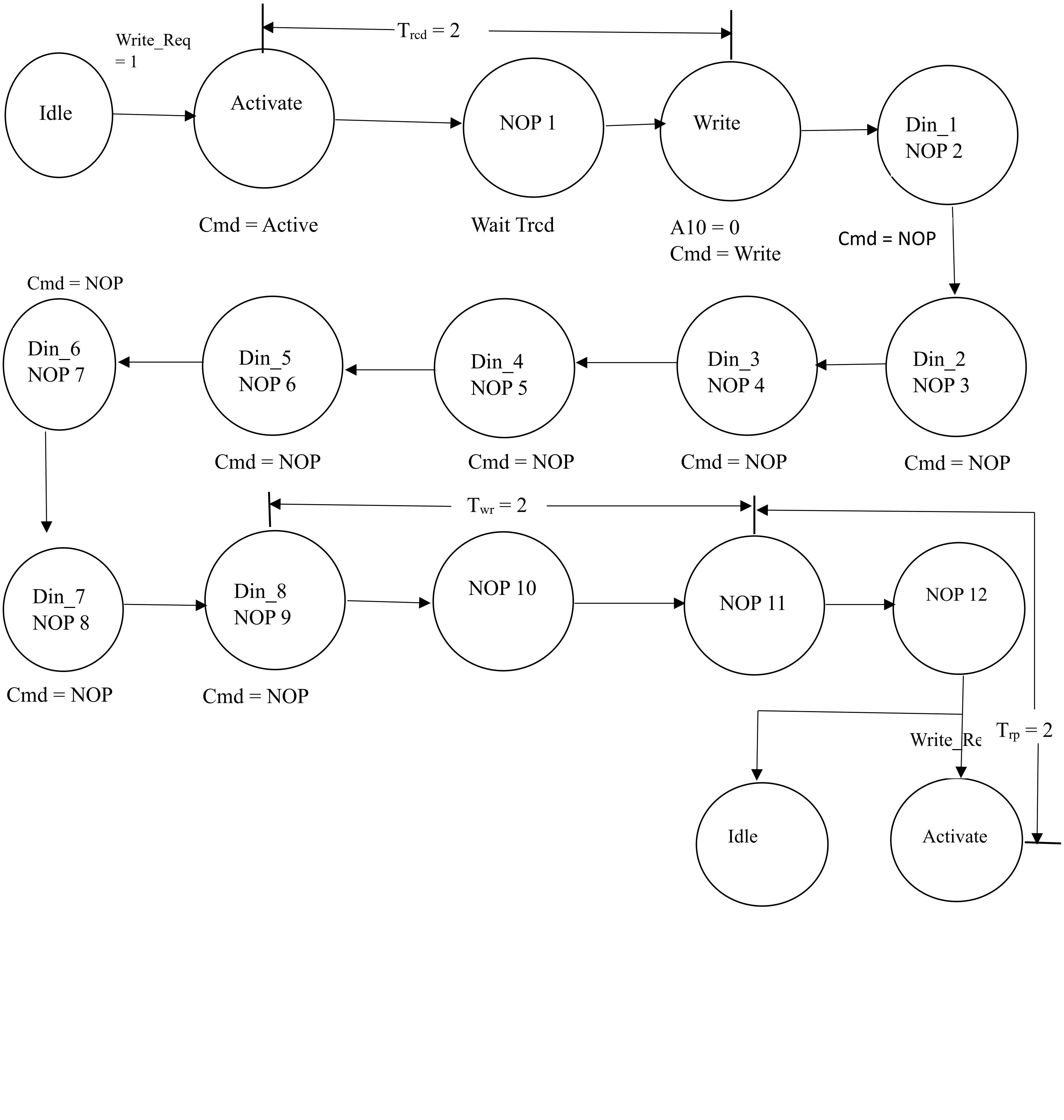
TRANSITION TABLE FOR WRITE WITHOUT AUTO-PRECHARGE

PRESENT STATE	SIGNAL CONDITION	DESCRIPTION	NEXT STATE
Idle	Write_req=1	Activate only if write Request.	Activate
Activate	Next clock	-	NOP1
NOP1	Next clock	Start writing only if wr=1 A10=0 denotes auto pre-charge is disabled	Write
Write	Next clock	Unlike read operation, data starts immediately writing once wr=1	Data-in 1
Data-in 1	Next clock	-	Data-in 2
Data-in 2 . . . . . Data-in 6	Next clock . . . . . Next clock	-	Data-in 3 . . . . . Data-in 7
Data-in 7	Next clock	-	Data-in 8
Data-in 8	Next clock	Wait for write Recovery time.	NOP10
NOP10	Next clock	-	Precharge.
Precharge.	Next clock	Wait for T_{rp}	NOP11
NOP11	Write_req != 1	-	Idle
NOP11	Write_req=1	Activate only if write Request.	Activate

TIMING DIAGRAM FOR WRITE WITHOUT AUTO-PRECHARGE



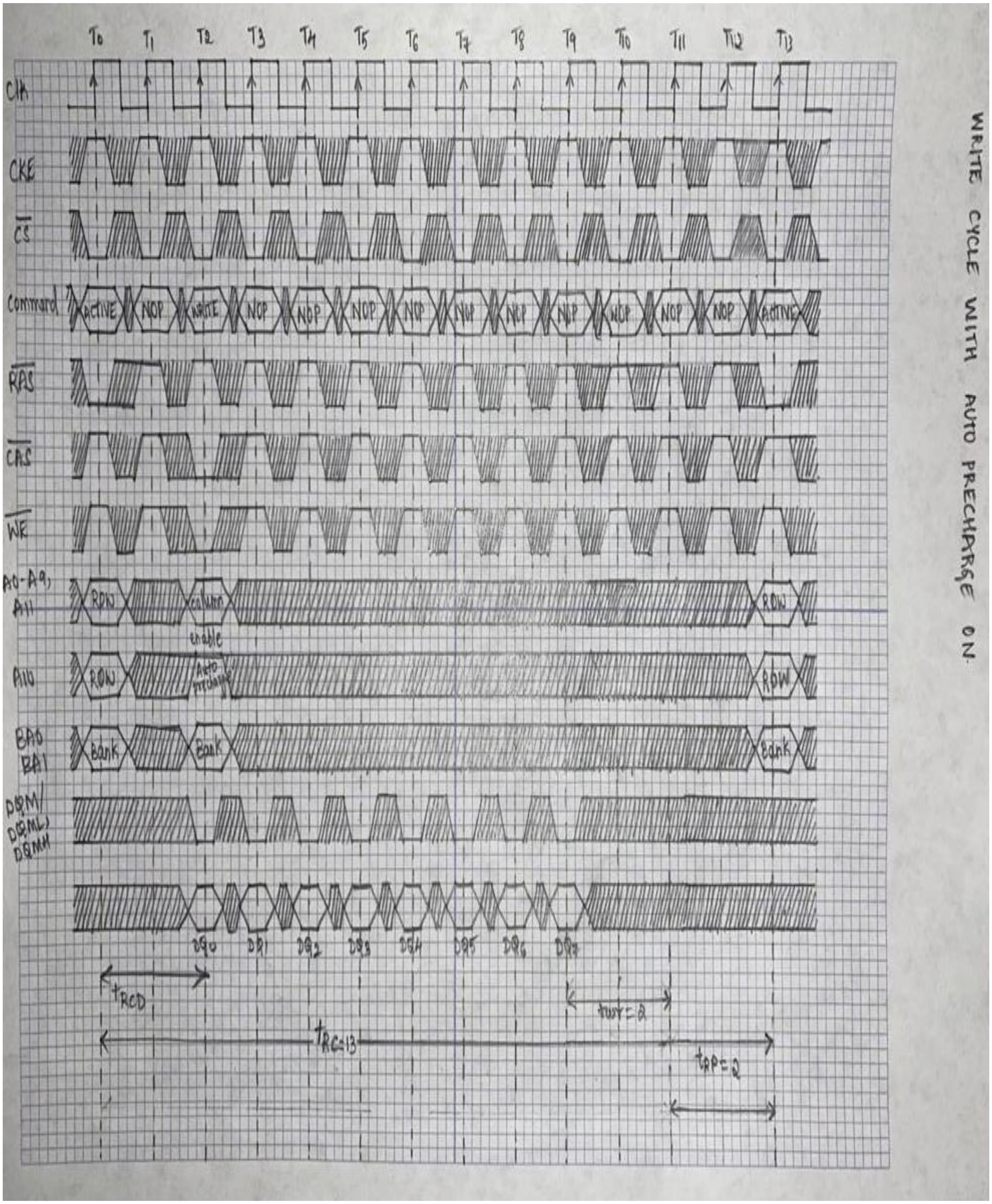
FSM FOR WRITE WITH AUTO-PRECHARGE ON



TRANSITION TABLE FOR WRITE WITH AUTO-PRECHARGE

PRESENT STATE	SIGNAL CONDITION	DESCRIPTION	NEXT STATE
Idle	Write_req=1	Activate only if write Request.	Activate
Activate	Next clock	-	NOP1
NOP1	Next clock	Start writing only if wr=1 A10=1 denotes auto pre-charge is enabled	Write
Write	Next clock	Unlike read operation, data starts immediately writing once wr=1	Data-in 1
Data-in 1	Next clock	-	Data-in 2
Data-in 2 . . . . . Data-in 6	Next clock . . . . . Next clock	-	Data-in 3 . . . . . Data-in 7
Data-in 7	Next clock	-	Data-in 8
Data-in 8 NOP9	Next clock	-	NOP10
NOP10	Next clock	Wait for write recovery time.	NOP11
NOP11	Next clock	Wait for precharge (T_{rp}) to complete.	NOP12
NOP12	Write_req != 1	-	Idle
NOP12	Write_req=1	Activate only if write Request.	Activate

TIMING DIAGRAM FOR WRITE WITH AUTO-PRECHARGE



EXTRA CREDITS

BURST LENGTH: 4

SPECIFICATIONS:

80386DX MICROPROCESSOR:
- 32-bit Address Bus
- 32-bit Data Bus
- 66MHz clk2 signal

Outgoing Signals from Processor:
- Address bus A31-0
- Data bus D31-0
- Control Signals

Incoming Signal to Processor:
- READY

Load Mode Register:
- In our Design CL-CAS LATENCY = 2 and BL-BURST LENGTH = 4
- Hence CAS latency = 010 and BURST LENGTH = 010

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		WB	Op Mode		CAS Latency			BT	Burst Length		
Value	0	0	0	0	0	0	1	0	0	0	1	0

Program: M11, M10: 0, 0: To ensure compatibility with future device

- Write Burst Mode: M9: 0: Program Burst Length
- Operating Mode: M8, M7: 0, 0: Standard Operation
- CAS latency: M6, M5, M4: 0, 1, 0: Two Clock Cycle
- Burst Type: M3: 0: Sequential
- Burst length: M2, M1, M0: 0, 1, 0 : Four Burst Length.

SYNCHRONOUS DRAM

Memory Configuration

MT48LC8M8A2: 2Meg*8*4Banks

Refresh

- 64ms, 4096 cycle refresh(15.6us/row)
- 16ms, 4096 cycle refresh(3.9us/row)

Refresh Count: 4K

Row Addressing: 4K(A0-A11)

Bank Addressing: 4(BA0, BA1)

Column Addressing: 1K(A0-A9)

DESIGN

Speed Grade: -7E

Clock Frequency: 133MHz

CL = 2(5.4ns) Clock Cycle Time: 7.5ns

Burst Length: 4

ADDRESS BUS INTERFACING

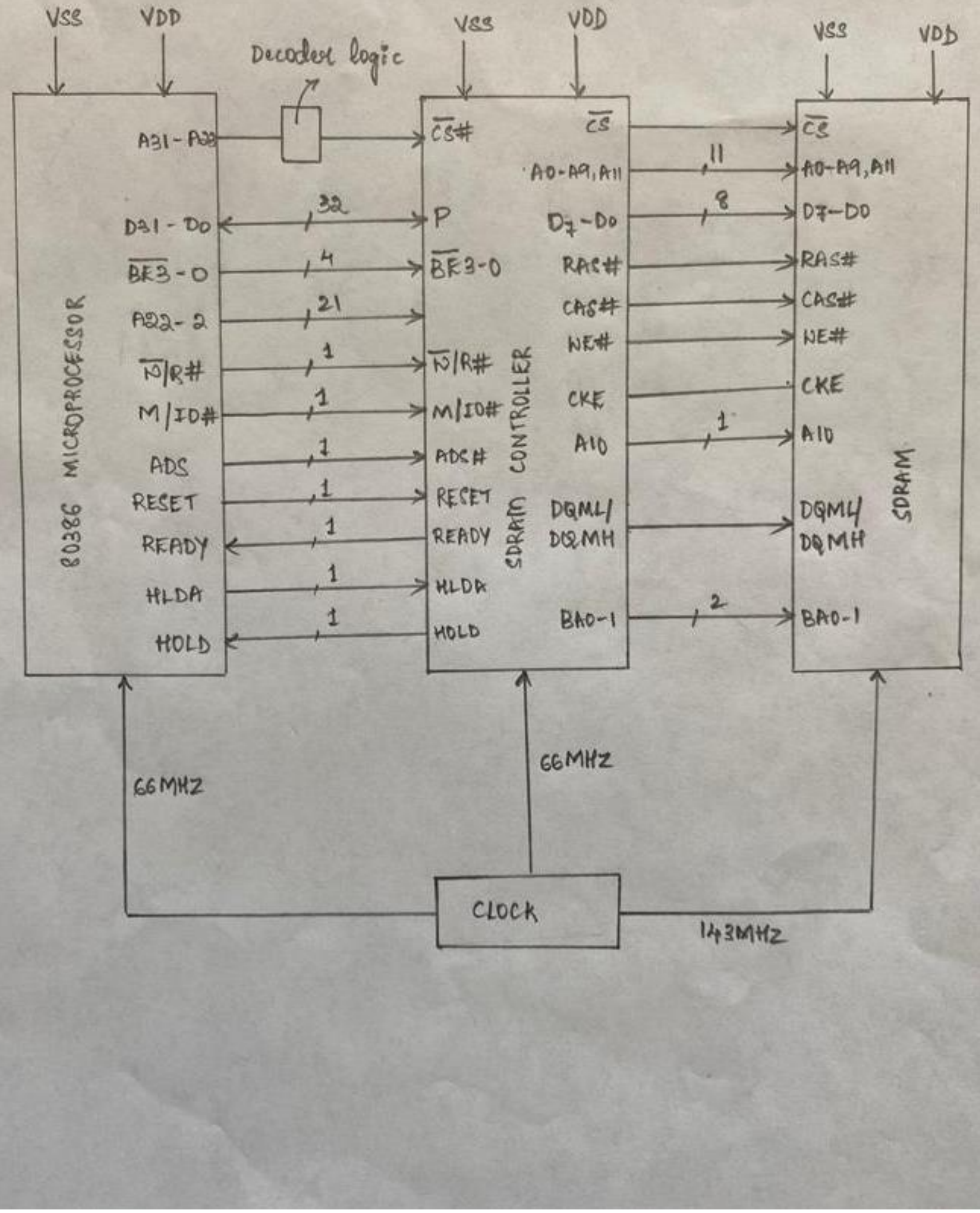
64Mib: 64*1024*1024

64Mib = 2^26 bits = 2^23 Bytes

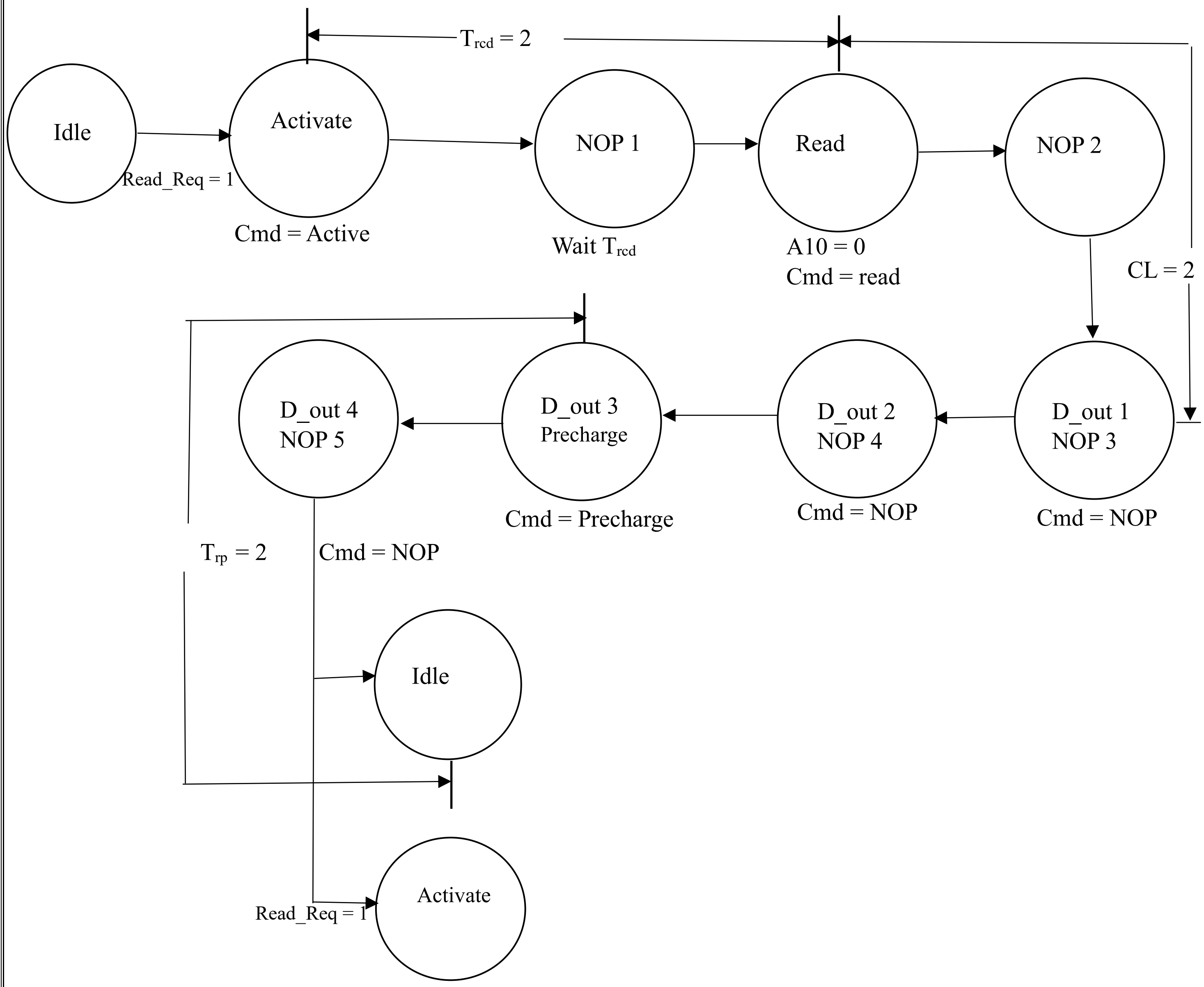
Log_2 (2^23) = 23 Address lines going to the address memory.

Burst length = 4:

SDRAM MICROPROCESSOR INTERFACE :



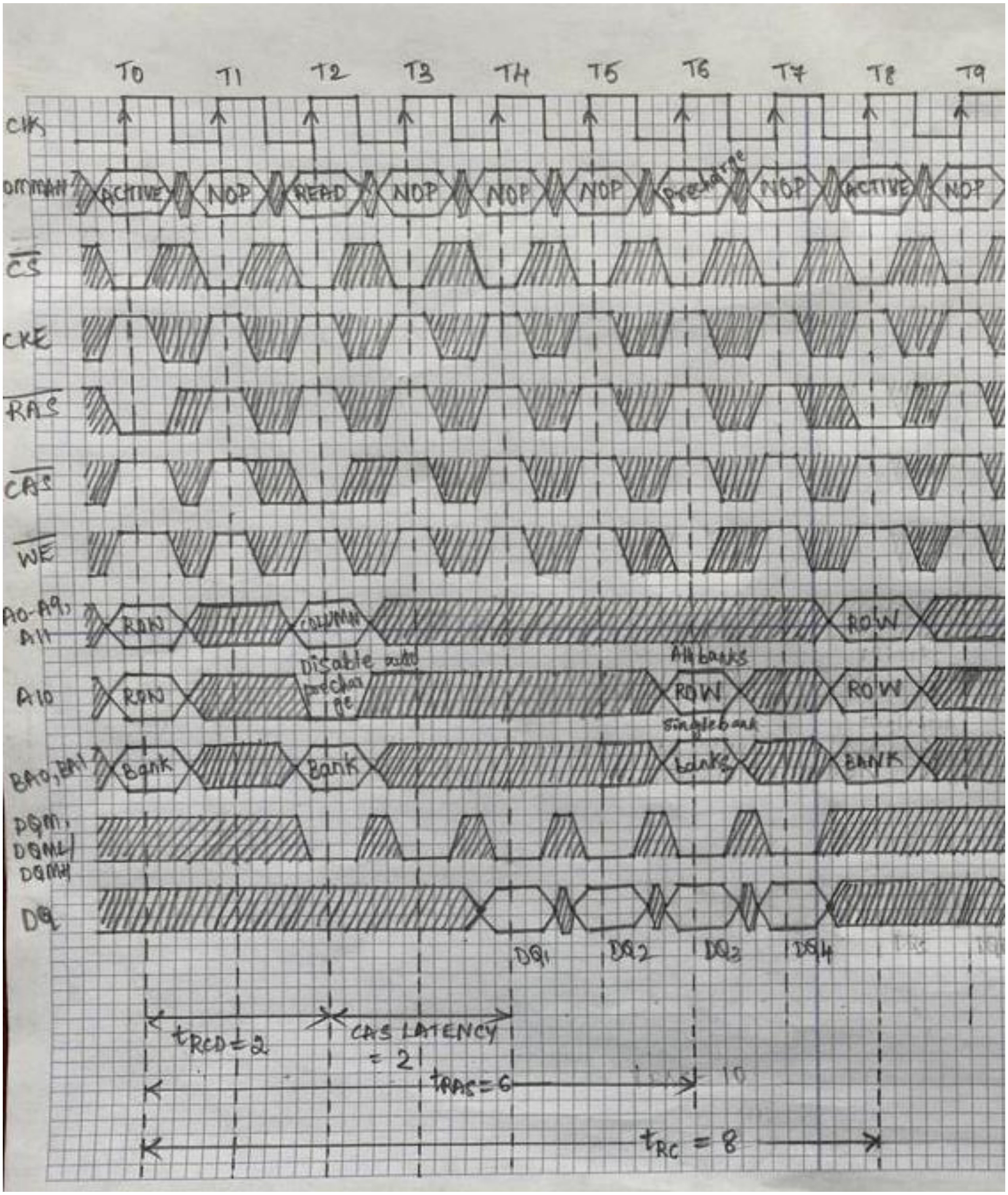
FSM FOR READ WITHOUT AUTO-PRECHARGE



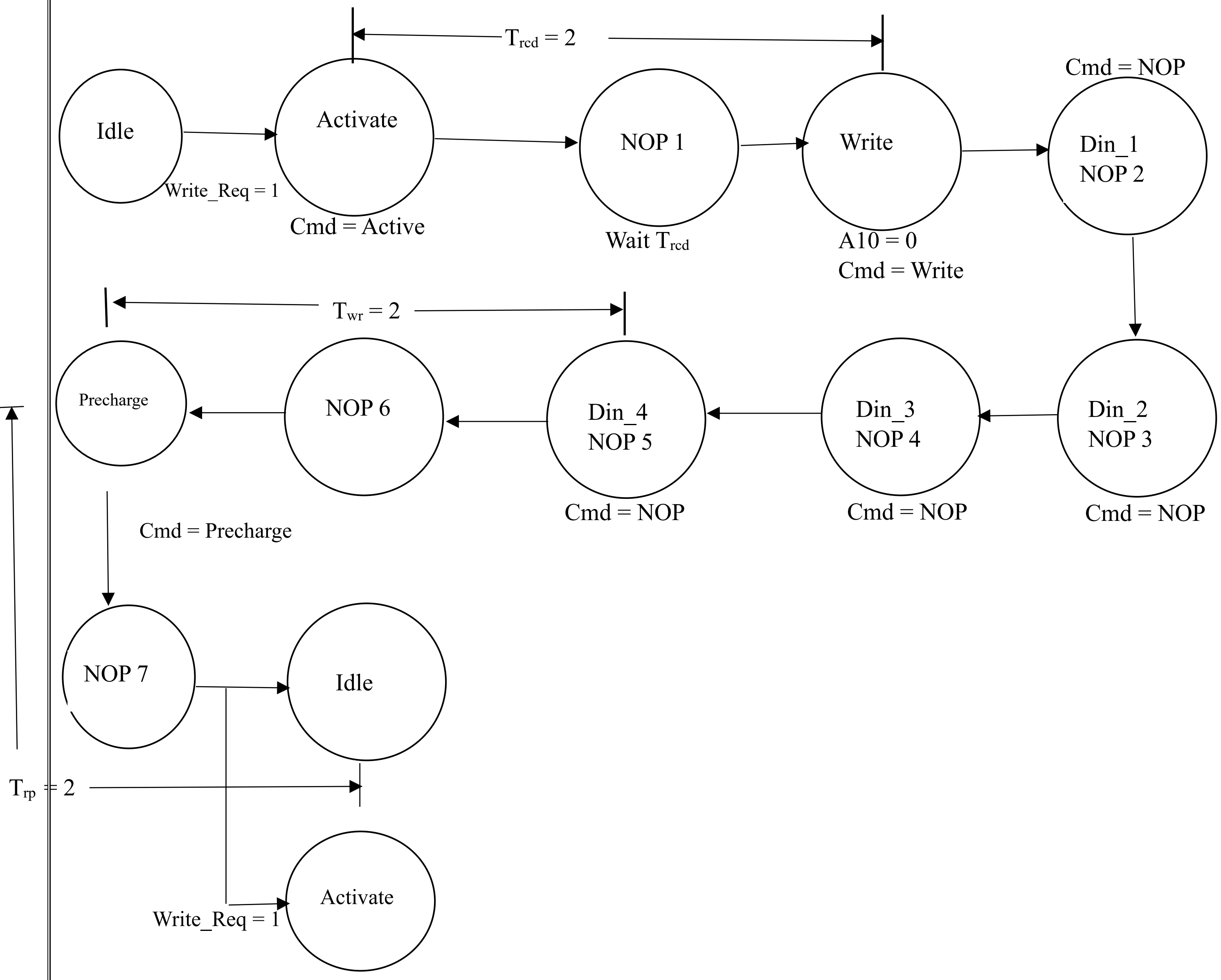
TRANSITION TABLE FOR READ WITHOUT AUTO-PRECHARGE

PRESENT STATE	SIGNAL CONDITION	DESCRIPTION	NEXT STATE
Idle	Read_req = 1	Activate only if read request	Activate
Activate	Next clock	Wait Until T_{rcd} is Completed	NOP1
NOP1	Next clock	Start reading only if RD=1 A10=0 denotes auto pre-charge is disabled	Read
Read	Next clock	Wait for 2 clocks as the CL=2	NOP2
NOP2	Next clock	Data starts bursting for every clock pulse after CL=2	Data-out 1
Data-out 1	Next clock	-	Data-out 2
Data-out 2	Next clock	-	Data-out 3(Start of Precharge)
Data-out 3(Start of Precharge)	Next clock	-	Data-out 4
Data-out 4	Read_req != 1	-	Idle
Data-out 4	Read_req = 1	Activate only if read request	Activate

TIMING DIAGRAM FOR READ WITHOUT AUTO-PRECHARGE



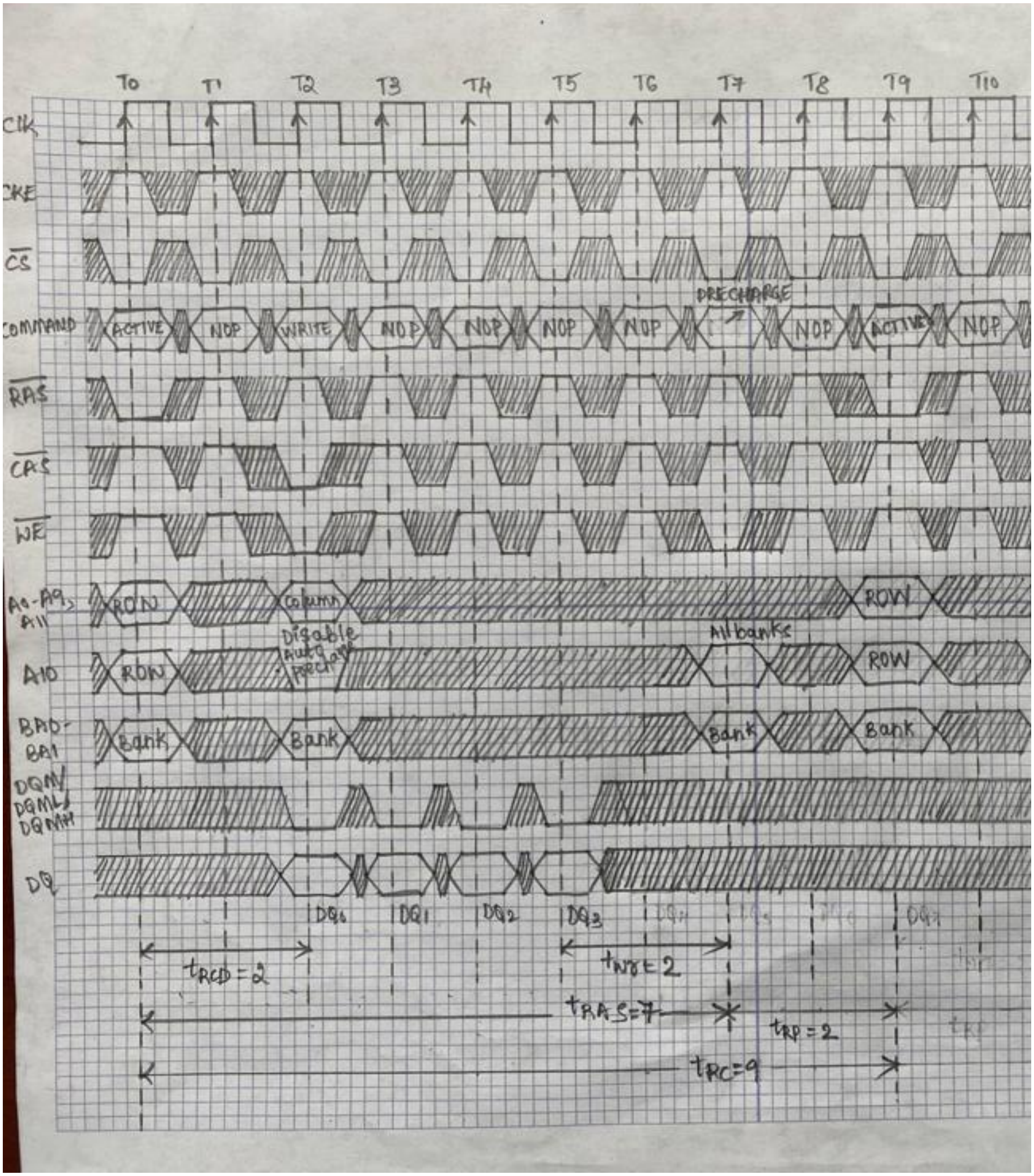
FSM FOR WRITE WITHOUT AUTO-PRECHARGE



TRANSITION TABLE FOR WRITE WITHOUT AUTO-PRECHARGE

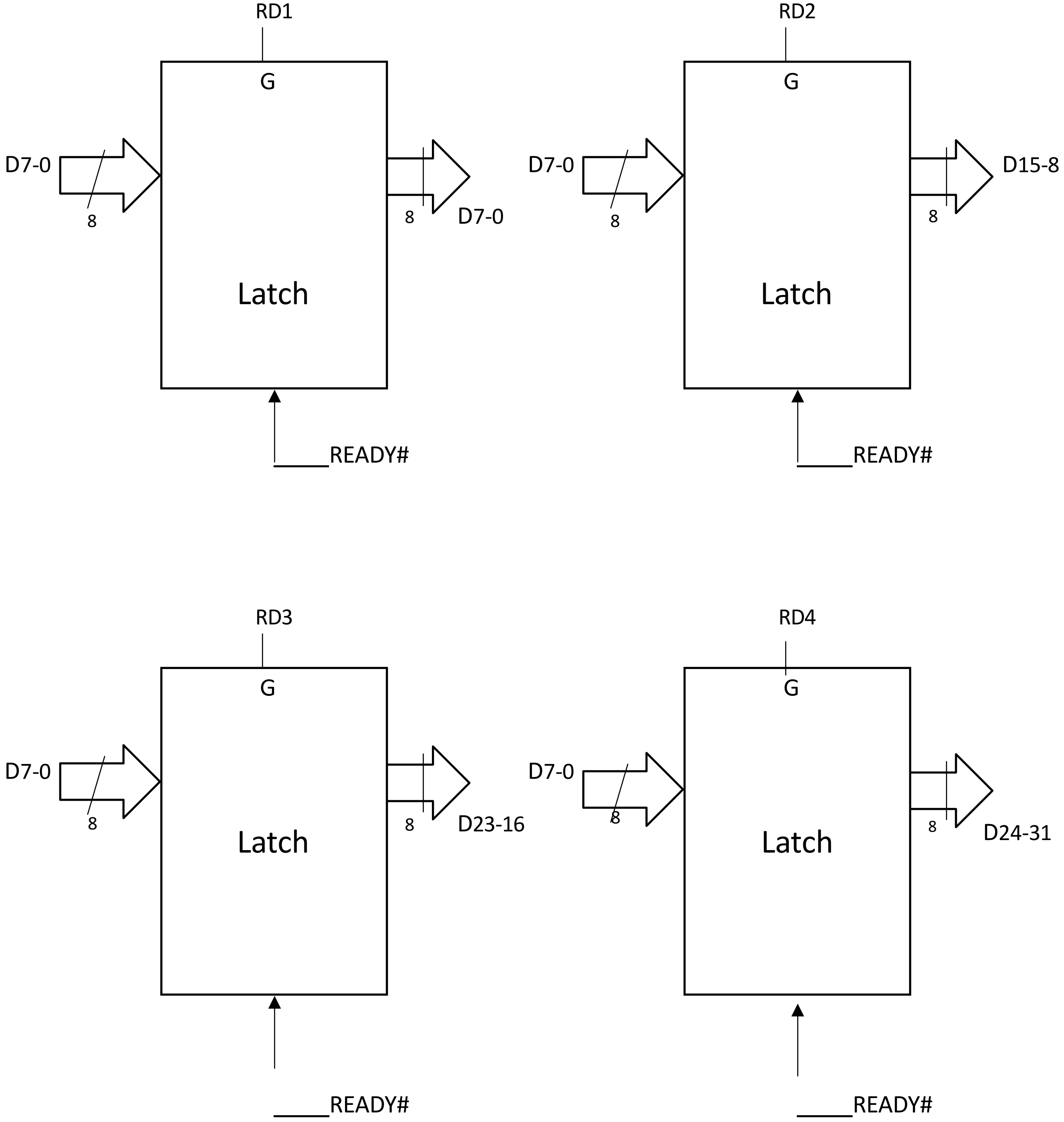
PRESENT STATE	SIGNAL CONDITION	DESCRIPTION	NEXT STATE
Idle	Write_req=1	Activate only if write Request.	Activate
Activate	Next clock	Wait T_{rcd}	NOP1
NOP1	Next clock	Start writing only if wr=1 A10=0 denotes auto pre-charge is disabled	Write
Write	Next clock	Unlike read operation, data starts immediately writing once wr=1	Data-in 1
Data-in 1	Next clock	-	Data-in 2
Data-in 2	Next clock	-	Data-in 3
Data-in 3	Next clock	-	Data-in 4
Data-in 4	Next clock	Wait for write recovery time.	NOP6
NOP6	Next clock	-	Precharge
Precharge	Next Clock	Wait for precharge command period.	NOP7
NOP7	Write_req=1	Activate only if write Request.	Activate
NOP7	Write_req != 1	-	Idle

TIMING DIAGRAM FOR WRITE WITHOUT AUTO-PRECHARGE

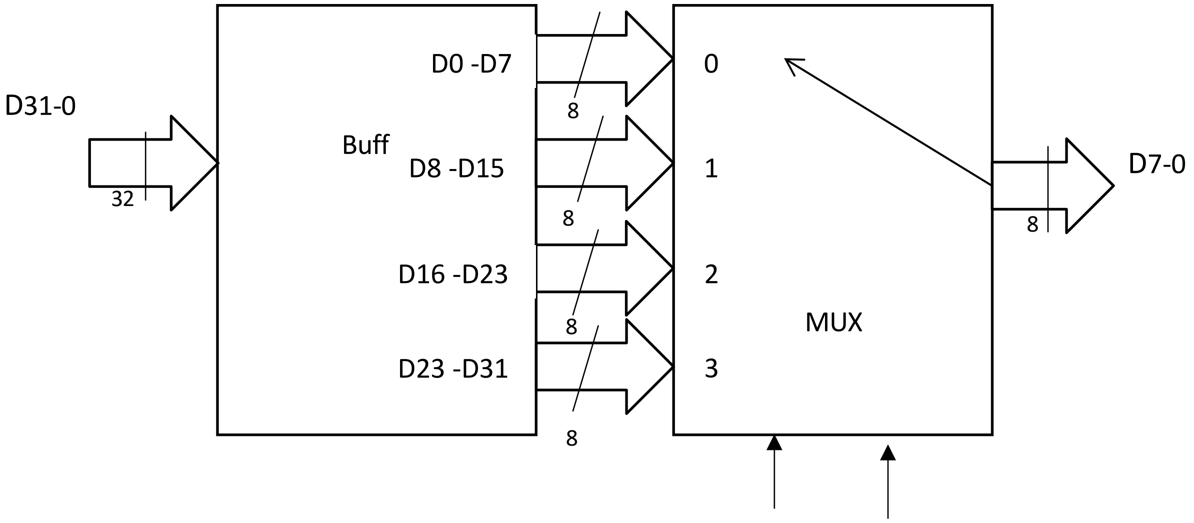


Generation of Chip Signals:

Read Data Latch:



Write Data Latch:

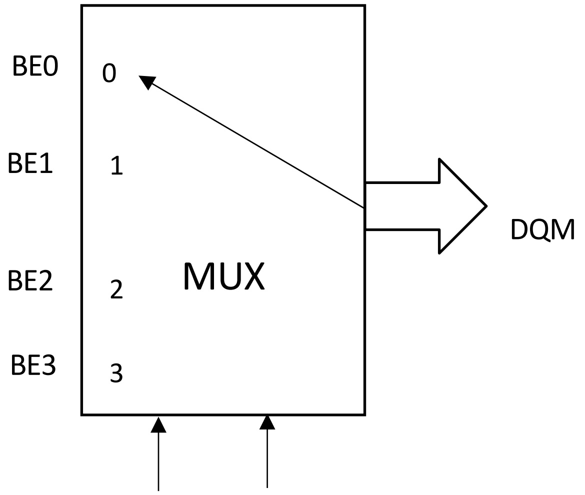


0: STATE = WR_1

1: STATE = WR_2

2: STATE = WR_3

3: STATE = WR_4



0: STATE = READ | WR_1

1: STATE = NOP | WR_2

2: STATE = RD_1 | WR_3

3: STATE = RD_2 | WR_4

BURST LENGTH: 2

SPECIFICATIONS:

80386DX MICROPROCESSOR:
- 32-bit Address Bus
- 32-bit Data Bus
- 66MHz clk2 signal

Outgoing Signals from Processor:
- Address bus A31-0
- Data bus D31-0
- Control Signals

Incoming Signal to Processor:
- READY

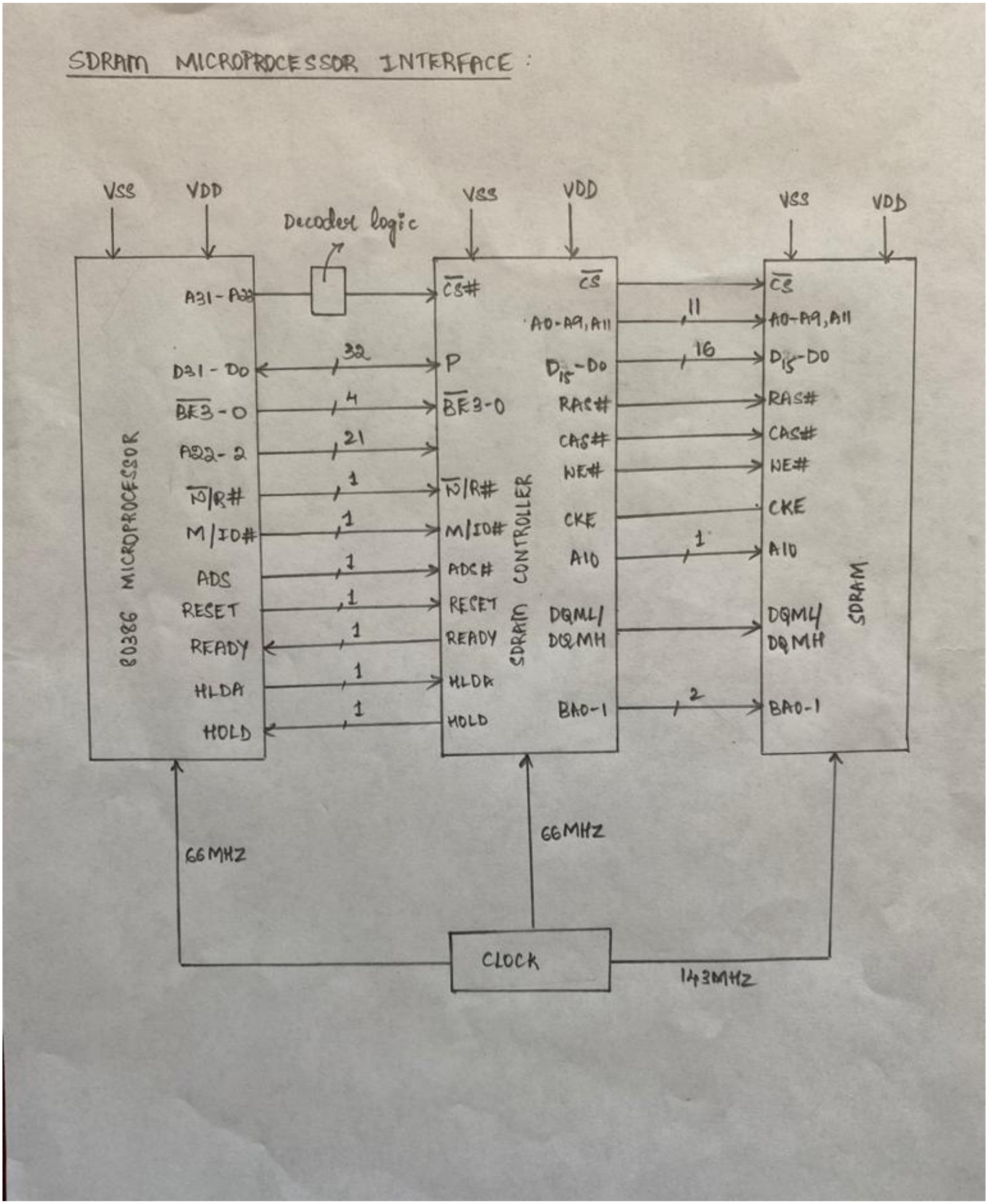
Load Mode Register:
- In our Design CL-CAS LATENCY = 2 and BL-BURST LENGTH = 2
- Hence CAS latency = 010 and BURST LENGTH = 001

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		WB	Op Mode		CAS Latency			BT	Burst Length		
Value	0	0	0	0	0	0	1	0	0	0	1	0

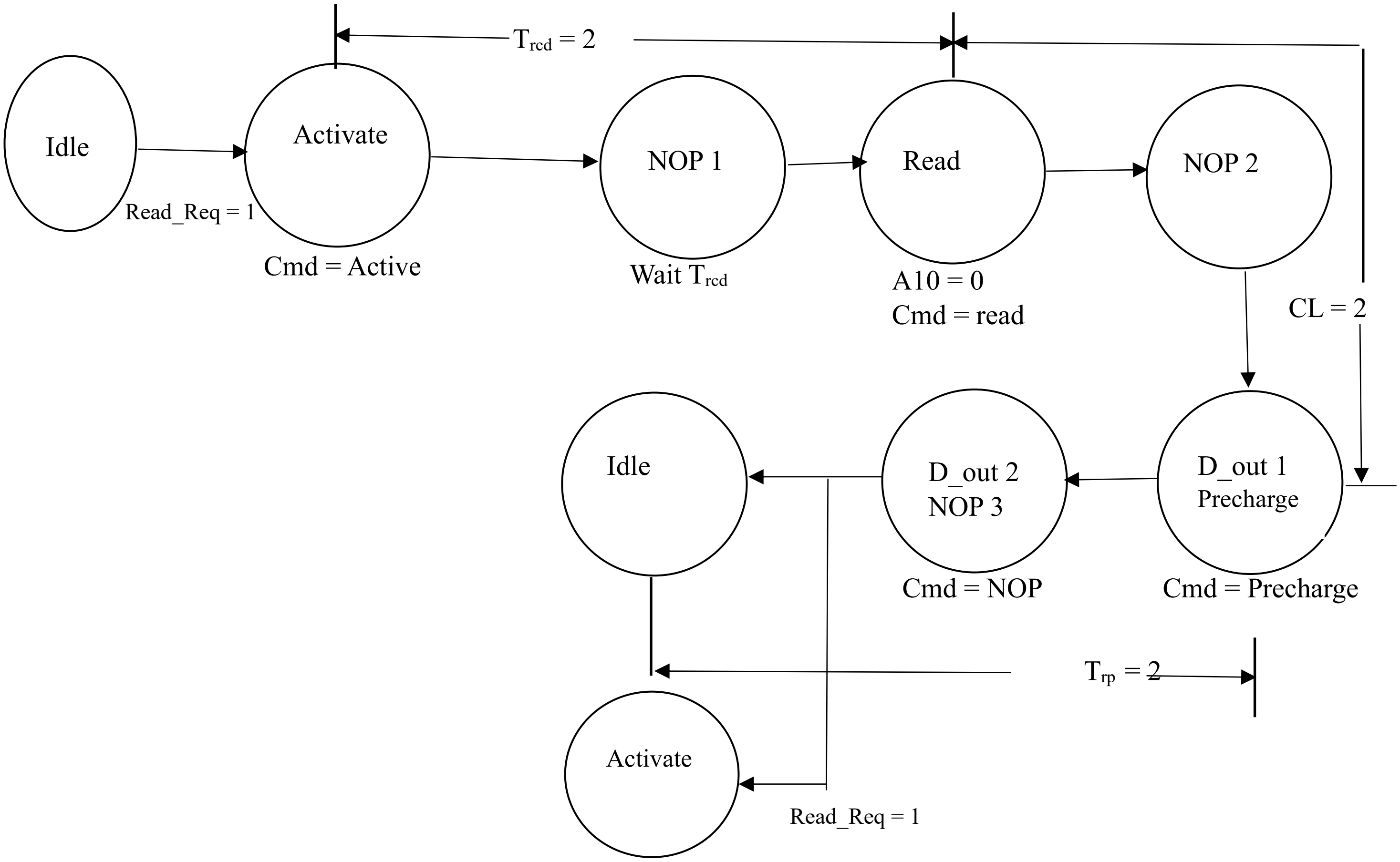
Program: M11, M10: 0, 0: To ensure compatibility with future device

- Write Burst Mode: M9: 0: Program Burst Length
- Operating Mode: M8, M7: 0, 0: Standard Operation
- CAS latency: M6, M5, M4: 0, 1, 0: Two Clock Cycle
- Burst Type: M3: 0: Sequential
- Burst length: M2, M1, M0: 0, 0, 1: Two Burst Length.

Burst length = 2:



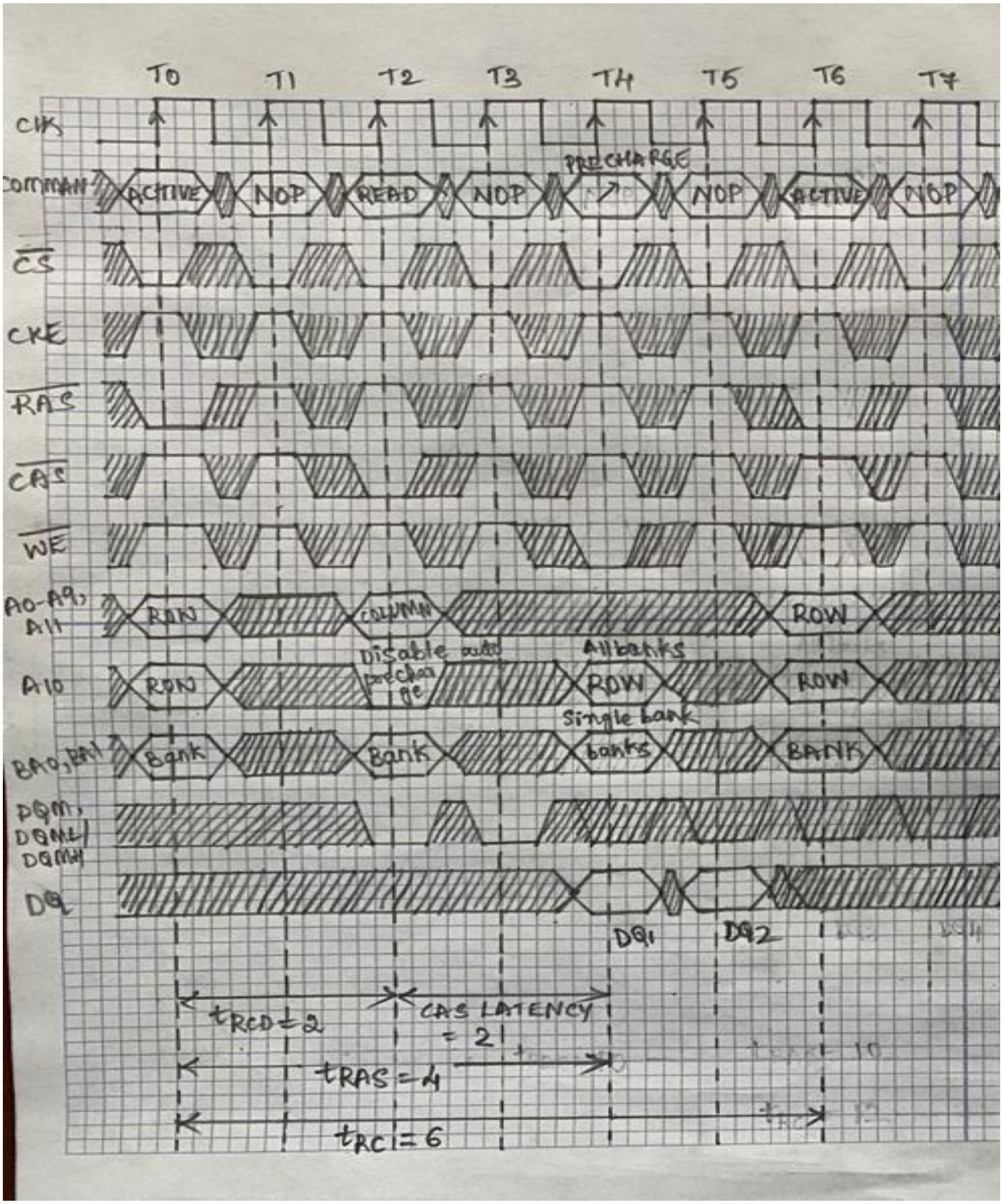
FSM FOR READ WITHOUT AUTO-PRECHARGE



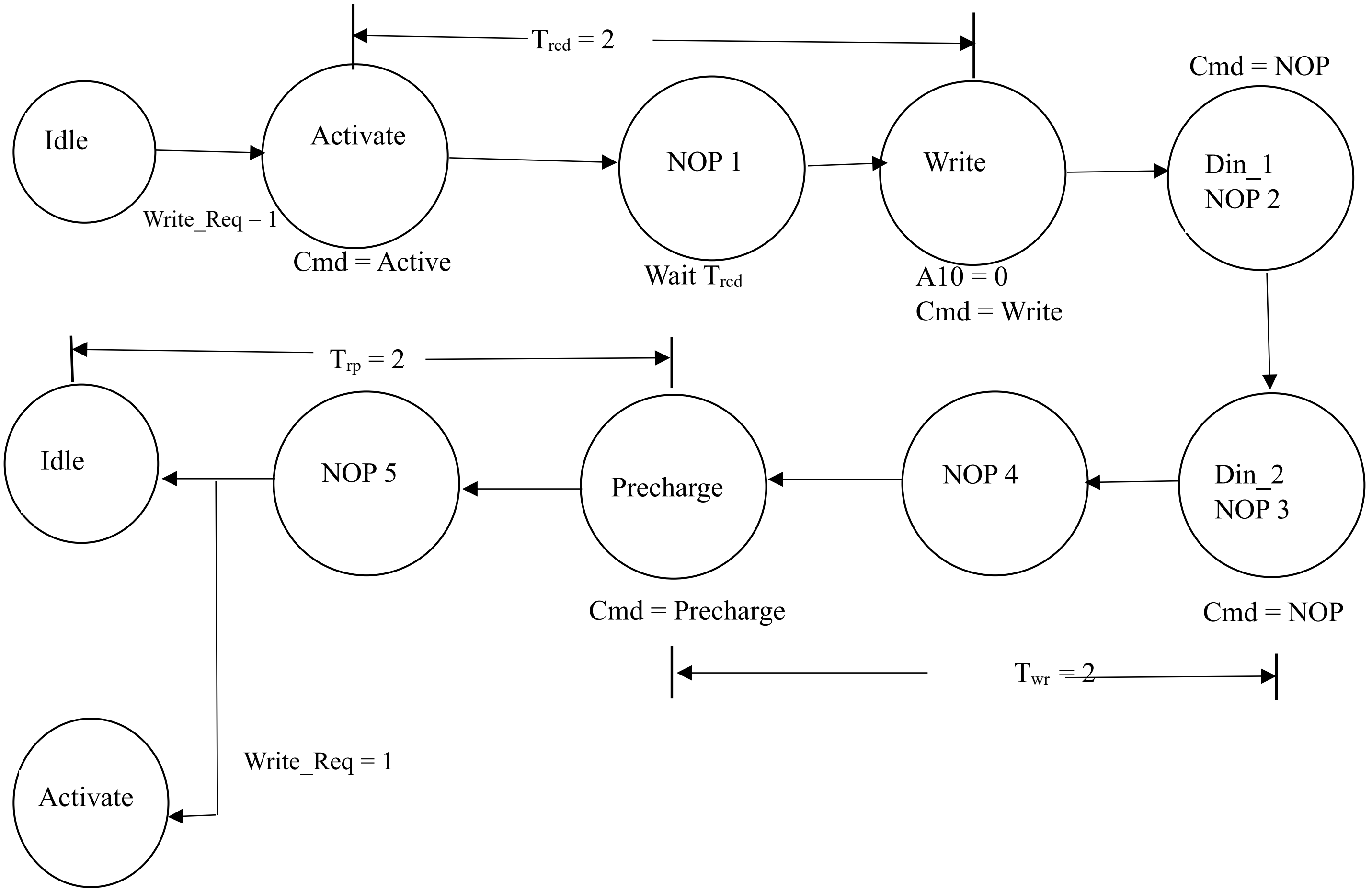
TRANSITION TABLE FOR READ WITHOUT AUTO-PRECHARGE

PRESENT STATE	SIGNAL CONDITION	DESCRIPTION	NEXT STATE
Idle	Read_req = 1	Activate only if read request	Activate
Activate	Next clock	-	NOP1
NOP1	Next clock	Start reading only if RD=1 A10=0 denotes auto pre-charge is disabled	Read
Read	Next clock	Wait for 2 clocks as the CL=2	NOP2
NOP2	Next clock	Data starts bursting for every clock pulse after CL=2	Data-out 1(Start of Precharge)
Data-out 1(Start of Precharge)	Next clock	-	Data-out 2
Data-out 2	Read_req != 1	-	Idle
Data-out 2	Read_req = 1	-	Activate

TIMING DIAGRAM FOR READ WITHOUT AUTO-PRECHARGE



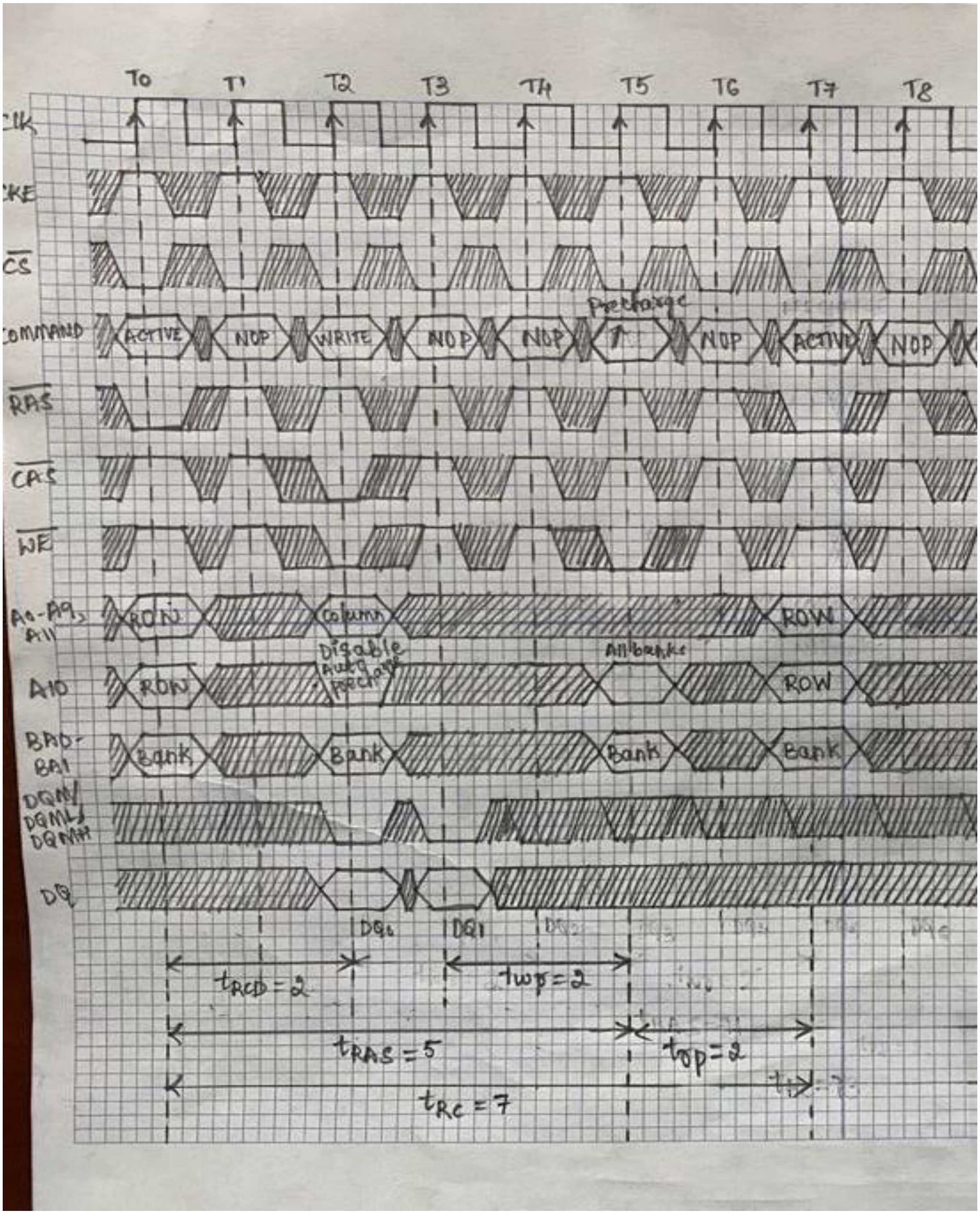
FSM FOR WRITE WITHOUT AUTO-PRECHARGE



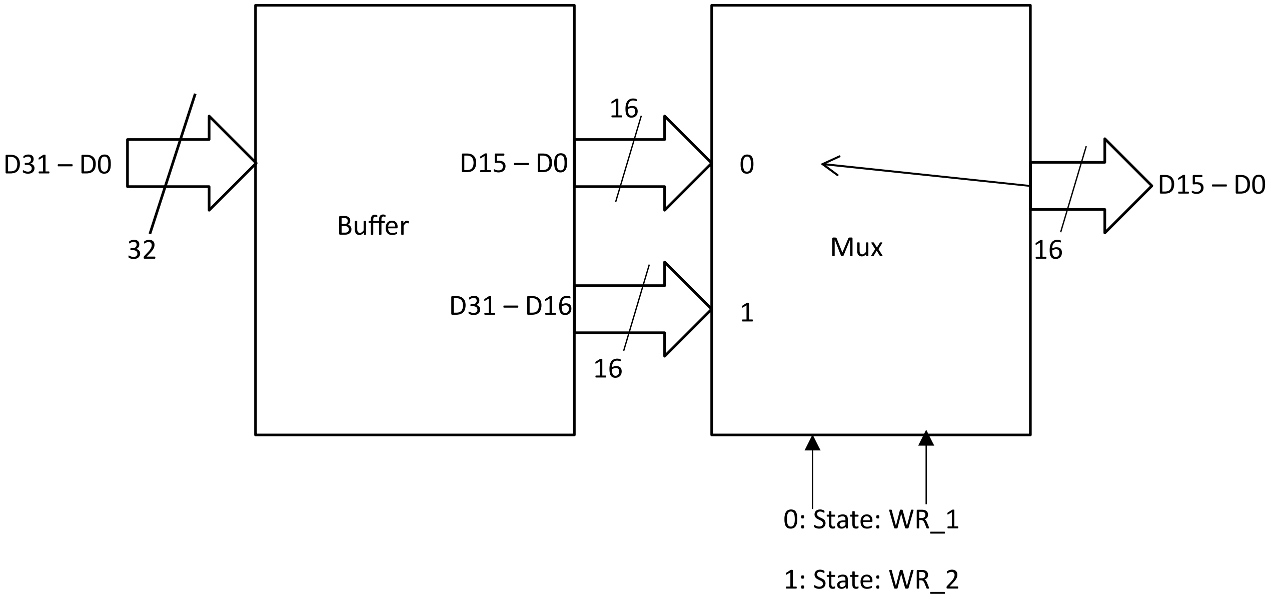
TRANSITION TABLE FOR WRITE WITHOUT AUTO-PRECHARGE

PRESENT STATE	SIGNAL CONDITION	DESCRIPTION	NEXT STATE
Idle	Write_req=1	Activate only if write Request.	Activate
Activate	Next clock	Wait T_{rcd}	NOP1
NOP1	Next clock	Start writing only if wr=1 A10=0 denotes auto pre-charge is disabled	Write
Write	Next clock	Unlike read operation, data starts immediately writing once wr=1	Data-in 1
Data-in 1	Next clock	-	Data-in 2
Data-in 2	Next clock	Wait for write Recovery Time	NOP4
NOP4	Next clock	-	Precharge
Precharge	Next clock	Wait until T_{rp} id completed.	NOP5
NOP5	Write_req != 1	-	Idle
NOP5	Write_req=1	Activate only if write Request.	Activate

TIMING DIAGRAM FOR WRITE WITHOUT AUTO-PRECHARGE

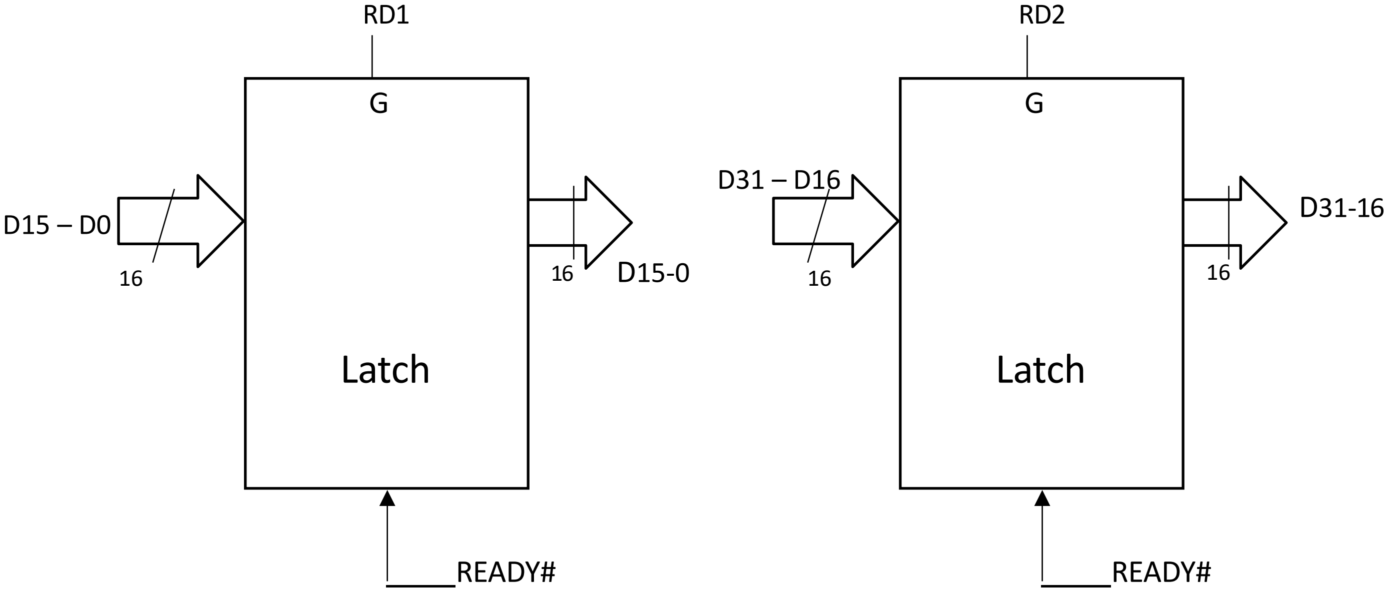


Data MUX for Write



Generation of Chip Signals:

Read Data Latch:



THANK YOU!