**EE 5313 MICROPROCESSOR PROJECT**

**SPRING 2016**

**SDRAM CONTROLLER DESIGN**

***By,***

***1001227227 - Navjot S. Saran***

***1001227227 - Kshitija Kulkarni***

***1001227274 - Mitali Salunkhe***

**CONTENTS:**

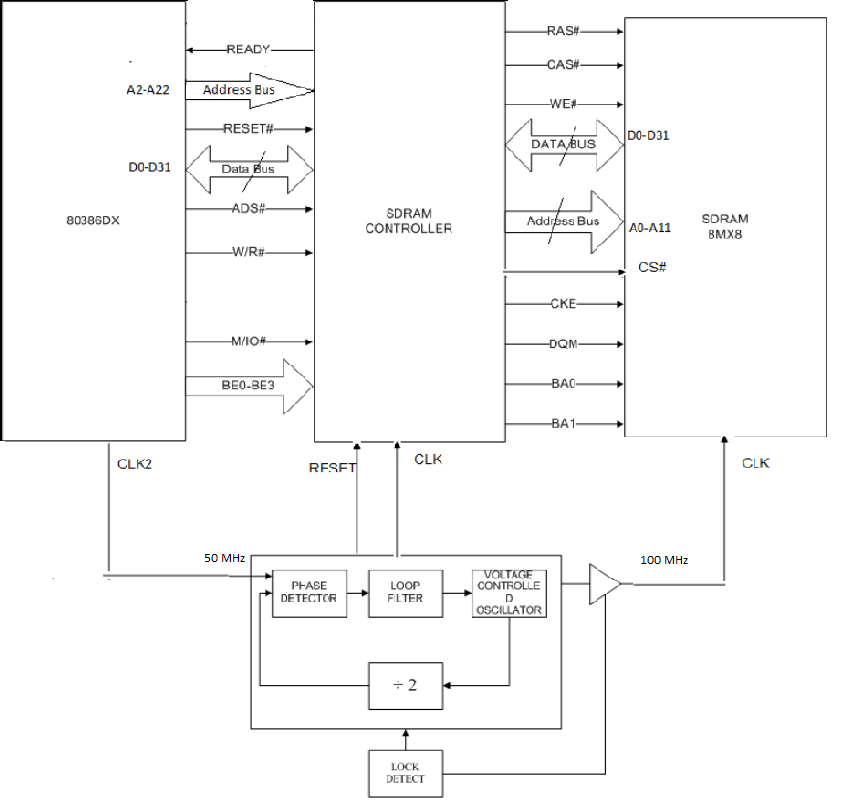
1. Introduction
2. Interfacing Diagram
3. Initialization
4. Row and Bank Activation
5. Reading from SDRAM memory
   1. Read with Auto Precharge
   2. Read state transistion table (Auto Precharge)
   3. Read with Manual Precharge
   4. Read state transistion table (Auto Precharge)
6. Writing to SDRAM memory
   1. Write with Auto Precharge
   2. Write state transistion table (Auto Precharge)
   3. Write with Manual Precharge
   4. Write state transistion table (Auto Precharge)
7. Auto Refresh
8. Address Signal Generation
9. Command Signals
10. Control Signals
11. Extra Credits

**INTRODUCTION:**

80386DX has only asynchronous memory support. In this project we are basically designing a SDRAM controller that interfaces the MT48LC8M8A2 SDRAM (2M X 8 X 4banks) memory with 80386DX microprocessor. The SDRAM controller located between the processor and memory and enables the conversion of 80386 commands into synchronous SDRAM memory command and control words. A burst length of 4 is supported by this design. In the project, we have used -75 MT48LC8M8A2 SDRAM Configuration. Our 80386DX microprocessor is working at 25MHz frequency (CLK2 = 50 MHz).

Fig 1. Shows the basic interference diagram between the 80386DX processor, the SDRAM controller and SDRAM memory.

**INTERFACING DIAGRAM:**

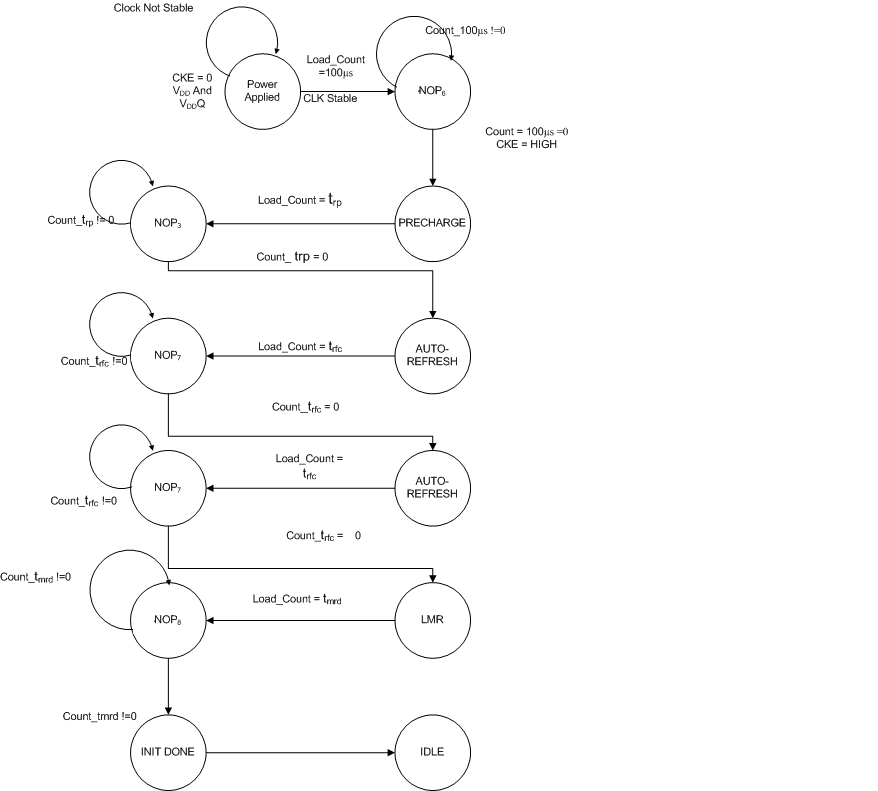
**

*Fig 1: Basic Block diagram of SDRAM – 80386DX Interface*

**MEMORY INITIALIZATION:**

NOP1 = trcd NOP3 = trp NOP5 = Start Auto PrechargeNOP7 = 100 us stateNOP9 = tmrd

NOP2 = tcl NOP4 = twr NOP6 = Clock StableNOP8 = Auto Refresh Delay

**

*Fig 2: State diagram: FSM initialization*

**Initialization Transition Table**

|  |  |  |
| --- | --- | --- |
| Current State | Conditions | Next State |
| X | Vdd,Vddq,CKE=0,CLK Stable | NOP6 |
| NOP6 | Count\_100μs=0 | Precharge |
| Precharge | X | NOP3 |
| NOP3 | Count\_trp=0 | Auto-Refresh |
| Auto Refresh | X | NOP8 |
| NOP8 | Count\_trfc=0 | Auto-Refresh |
| Auto-Refresh | X | NOP8 |
| NOP8 | Count\_trfc=0 | LMR |
| LMR | X | NOP8 |
| NOP8 | Count\_tmrd=0 | INIT DONE |

Load Mode Register Fields:

1. Burst Length (BL) = 4 : 0 1 0
2. Burst Type (BT) = Sequential : 0
3. CAS Latency = 2 : 0 1 0
4. Op Mode = Standard Operation : 0 0
5. WB = Programmed Burst Length : 0
6. Reserved : 0 0

**READ OPERATION WITH AUTO PRECHARGE:**



*Fig 3: State diagram: Reading from SDRAM memory (with auto precharge)*

**READ WITH AUTO PRECHARGE TRANSISTION TABLE:**

|  |  |  |
| --- | --- | --- |
| PRESENT STATE | CONDITION | NEXT STATE |
| ACTIVE | LOAD VALUE tRCD | NOP1 |
| NOP1 | Count\_tRCD = 0 | READ |
| READ | LOAD VALUE tCL | NOP2 |
| NOP2 | tCL VALUE = 0 | Rxfer1 |
| Rxfer1 | X | Rxfer2 |
| Rxfer2 | X | Rxfer3 |
| Rxfer3 | X (next clock) | Rxfer4 |
| Rxfer3 | X(same clock) | AUTO PRECHARGE (A10 = 1) |
| AUTO PRECHARGE | Load\_count=tRP | NOP3 |
| NOP3 | Count\_tRP = 0 | IDLE |

*Fig 4: State Table: Read with Auto Precharge*

**READ OPERATION WITH MANUAL PRECHARGE***:*



*Fig 5: State diagram: Reading from SDRAM memory (with manual precharge)*

**READ WITH MANUAL PRECHARGE TRANSISTION TABLE:**

*Fig 6: State Table: Read with Manual Precharge*

|  |  |  |
| --- | --- | --- |
| PRESENT STATE | CONDITION | NEXT STATE |
| ACTIVE | LOAD VALUE tRCD | NOP1 |
| NOP1 | tRCD VALUE = 0 | READ |
| READ | LOAD VALUE tCL | NOP2 |
| NOP2 | tCL VALUE = 0 | Rxfer1 |
| Rxfer1 | X | Rxfer2 |
| Rxfer2 | X | Rxfer3 |
| Rxfer3 | X (next clock) | Rxfer4 |
| Rxfer3 | X(same clock) | PRECHARGE (A10 = 0) |
| PRECHARGE | Load\_count=tRP | NOP3 |
| NOP3 | tRP VALUE = 0 | IDLE |

**WRITE OPERATION WITH AUTO PRECHARGE:**



*Fig 7: State diagram: Writing to SDRAM memory (with auto precharge)*

**WRITE WITH AUTO PRECHARGE TRANSISTION TABLE:**

|  |  |  |
| --- | --- | --- |
| PRESENT STATE | CONDITION | NEXT STATE |
| ACTIVE | LOAD VALUE tRCD | NOP1 |
| NOP1 | tRCD VALUE = 0 | WRITE , |
| WRITE | X(same clock) | Wxfer1 |
| Wxfer1 | X(next clock) | Wxfer2 |
| Wxfer2 | X | Wxfer3 |
| Wxfer3 | X | Wxfer4 |
| Wxfer4 | LOAD VALUE tWR | NOP4 |
| NOP4 | tWR VALUE = 0 | NOP5 |
| NOP5 | LOAD VALUE tRP | NOP3 |
| NOP3 | tRP VALUE = 0 | IDLE |

*Fig 8: State Transition Table: Write with auto precharge*

**WRITE OPERATION WITH MANUAL OPERATION:**



*Fig 9: State diagram: Writing to SDRAM memory(with manual precharge)*

**WRITE WITH MANUAL PRECHARGE TRANSISTION TABLE:**

|  |  |  |
| --- | --- | --- |
| PRESENT STATE | CONDITION | NEXT STATE |
| ACTIVE | LOAD VALUE tRCD | NOP1 |
| NOP1 | tRCD VALUE = 0 | WRITE , |
| WRITE | X(same clock) | Wxfer1 |
| Wxfer1 | X(next clock) | Wxfer2 |
| Wxfer2 | X | Wxfer3 |
| Wxfer3 | X | Wxfer4 |
| Wxfer4 | LOAD VALUE tWR | NOP4 |
| NOP4 | tWR VALUE = 0 | PRECHARGE |
| NOP5 | LOAD VALUE tRP | NOP3 |
| NOP3 | tRP VALUE = 0 | IDLE |

*Fig 10: State Transition Table: Write with manual precharge*

**AUTO REFRESH:**

1. This command must be issued each time a refresh is required. All active banks must be precharged before issuing an auto refresh command.
2. Address bits are “Don’t Care” during AUTO REFRESH since the internal refresh controller does the addressing part.
3. Regardless of the SDRAM memory configuration, there are 4096 rows. Hence it requires 4096 auto refresh cycles every 64ms (commercial and industrial) and 16ms (automotive).

NOTE: SELF REFRESH command is initiated like AUTO REFRESH command except CKE is disabled (LOW).



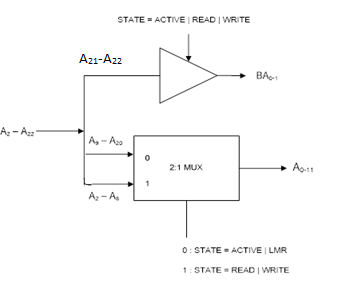
*Fig 11: State Diagram: Auto Refresh*

**AUTO REFRESH TRANSISTION TABLE:**

|  |  |  |
| --- | --- | --- |
| PRESENT STATE | CONDITION | NEXT STATE |
| IDLE | X | AUTO REFRESH |
| AUTO REFRESH | LOAD VALUE tRFC | NOP |
| NOP | tRFC VALUE = 0 | IDLE |

*Fig 12: State Transition Diagram: Auto Refresh*

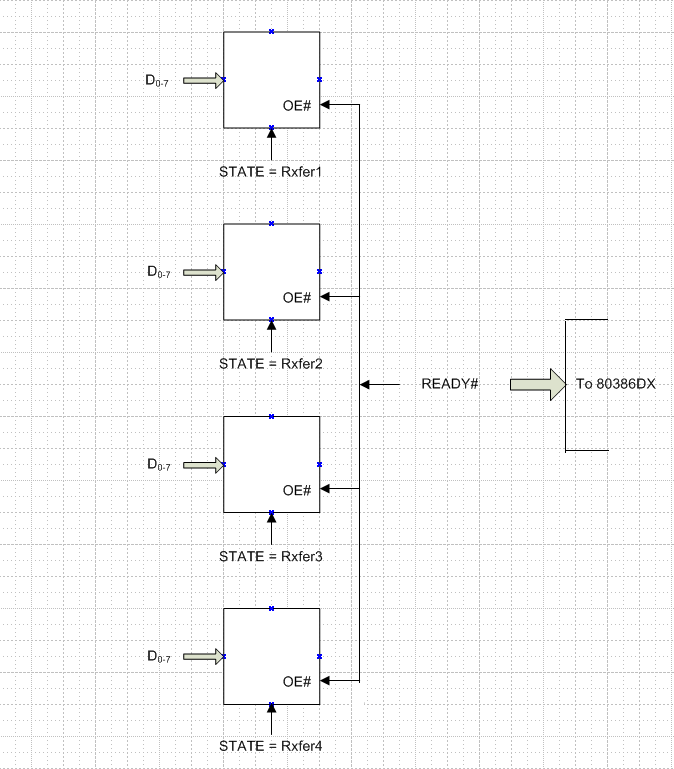
**Address Signal Generation:**

**

*Fig 13: Block Diagram for Address Generation*

In this memory address latching with microprocessor is done as following: A2-A8 forms column address (LSBs of column are hardcoded), A9-A20 forms row address and A21-A22 forms BA0-BA1 signals to the memory.

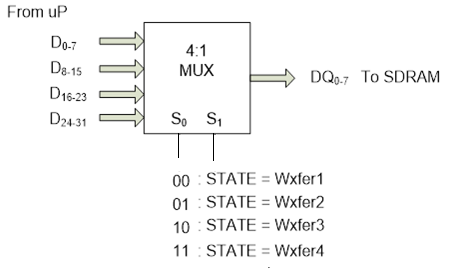
**Data Path Module (For Read):**



*Fig 14: Block Diagram of Data Flow (READ)*

In data line interface, while reading the data from the memory is latched byte by byte to 4 different latches corresponding to 4 different data for BL=4 and is outputted simultaneously from all 4 latches on READY# signal to give 32-bit data to the microprocessor.

**Data Path Module (For write):**



*Fig 15: Block Diagram of Data Flow (WRITE)*

Similarly, while writing each eight bit- data is given to the memory using a 4:1 multiplexer.

**Command Signals:**

Depending upon our state we are enabling various command signals such as:

**NOP:**



**LMR (LOAD MODE REGISTER):**



**ACTIVE:**



**READ:**



**Write:**



**Precharge:**

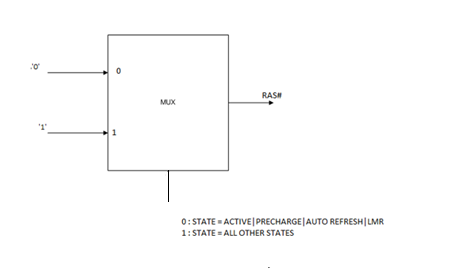


**Refresh:**

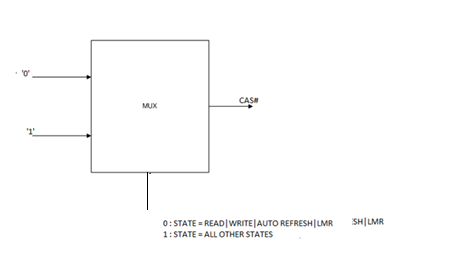


**CONTROL SIGNALS:**

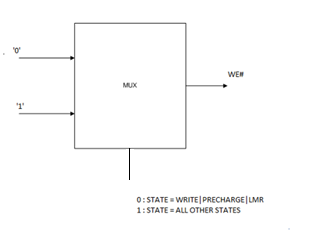
1. **RAS ( Row Address Strobe):**

**

1. **CAS ( Column Address Strobe):**

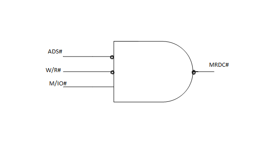
**

1. **WE ( Write Enable):**

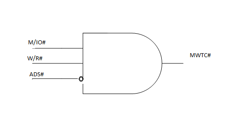
**

1. **MRDC (Memory Read Control)#:**

80386DX does not have MRDC# and MWTC# signals so we need to generate these signals using these logics.



1. **MWTC (Memory Write Control)#:**



1. **READY SIGNAL**



**Design of Various Delay Counters:**

1. Count to be loaded = 100 μs / 10 ns = 2710H



1. Auto Refresh Counter:

trefresh = 64 ms / (No of rows)

= 64 ms / 4096

µs

= 15.6 (Count to be loaded 0820H)



1. trcd = 15 ns / 10 ns = 2 clock cycles

Clk

1. tcl = 15 ns / 10 ns = 2 clock cycles



1. trp = 15 ns / 10 ns = 2 clock cycles



1. twr



1. LMR: tmrd = 15 ns / 7.5 ns = 2 clock cycles



1. trfc = 66 / 10 = 7 clock cycles



1. trc = 60/10 = 6 clock cycles



1. tLMR = 2trfc + trp



**EXTRA CREDITS:**

Showing Support for MT48LC16M4A2 – 4 Meg x 4 x 4 banks with higher burst length (BL=8).

Here, we will interface a 4bit width memory with 80386DX microprocessor.

Showing Data-Path Module(For Read Operation):



Data Path Module (For Write Operation):



Finite State Diagram:

1. Read Cycle:



1. Finite State Machine (Write Cycle):



Address Generation(for X 4 mode) :

