Single-Data Rate SDRAM Project
(Part No. MT48LC2M32B2)

Document Ver. 1
04/02/2021

Table of Contents

Objective 1

Components/Tools 1

SDRAM Chip 1
- Memory Organization 1
- Pinout 2
- Mode Register 3
- Commands 4
- Timings 5

System 7
- Description 7
- Pin Mapping 8
- Timing 10
- Code 11
  - Command MCU's Code 11
  - Data MCU's Code 17
- Design Errors 19

Miscellaneous 20
- Links 20
- References 20

Objective

Set up a system to write and read data from a single address on the SDRAM chip.

Components/Tools

- SDR SDRAM chip (part no. MT48LC2M32B2)
- 86-pin TSOP socket
- 2 8-bit MCUs (part no. PIC18F45K50)
- Power supply
- Function generator

SDRAM Chip

Memory Organization

64 Mbit total memory

4 banks, 16 Mbit each

- 2^{11} rows $*$ 2^{8} columns $*$ 32 bits
- Address bits specify location at which operations take place, specific to the command issued (see A[10:0] in “Pinout” sub-section).

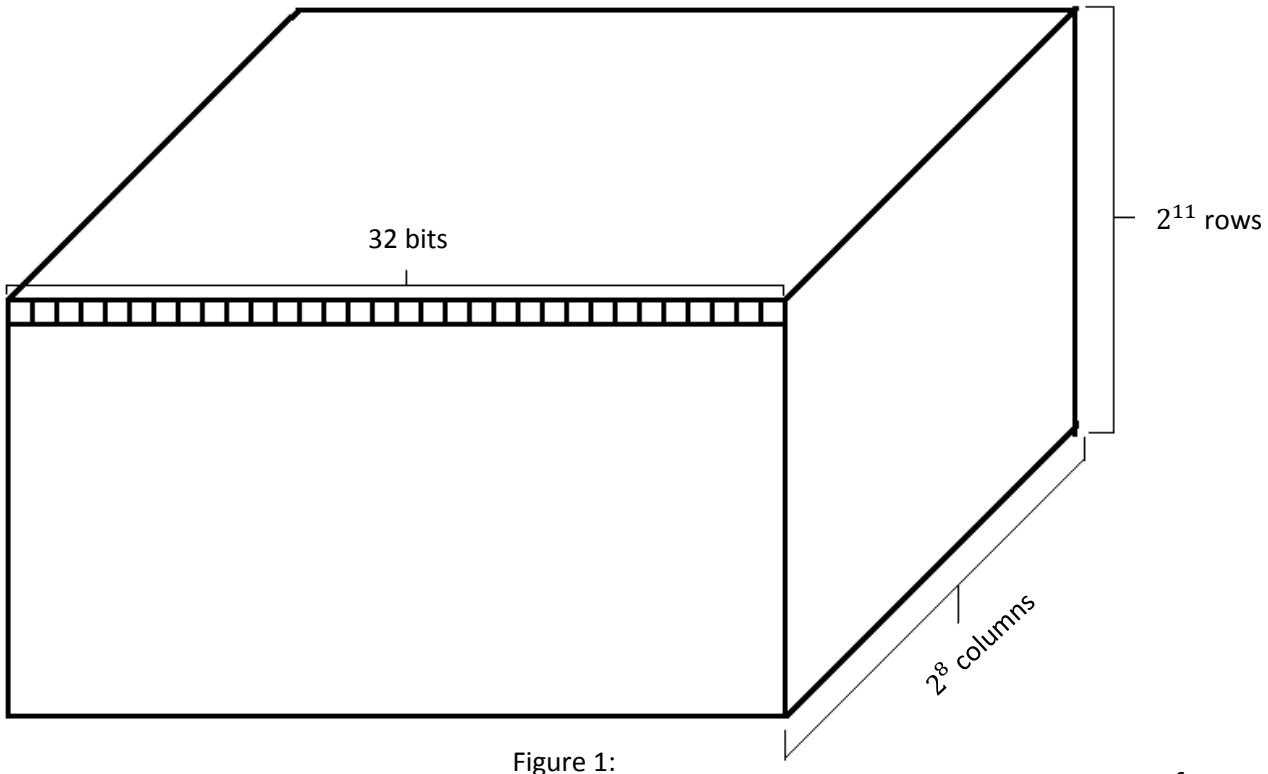


Figure 1:
Bank’s Memory Configuration

Pinout

Only included pinout description that is relevant to the current system's setup.

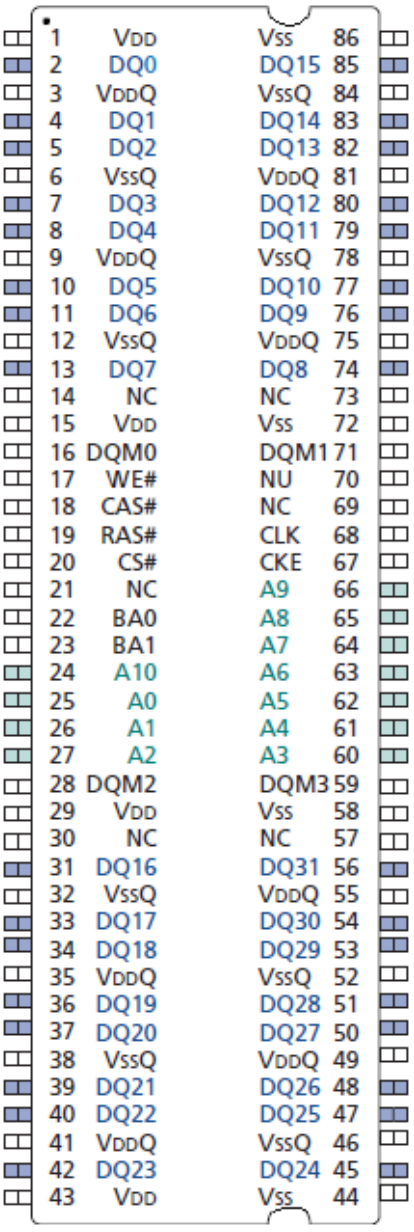


Figure 2:
TSOP Pinout

Pin	Type	Description
CLK	Input	Supplied clock signal. All SDRAM input is sampled on the rising edge.
CKE	Input	Clock enable; HIGH = enable CLK, LOW = disable CLK.
CS#	Input	Chip select; LOW = select SDRAM, HIGH = deselect SDRAM. All commands ignored when deselected.
CAS#, RAS#, WE#	Input	Command inputs (along with CS#) that determine the command being issued to the SDRAM.
DQM[3:0]	Input	DQ mask. Each DQM bit corresponds to 8 bits on DQ: DQM[3] → DQ[31:24], DQM[2] → DQ[23:16], DQM[1] → DQ[15:8], DQM[0] → DQ[7:0]. HIGH = high-Z, LOW = valid I/O (see Figure 3)
BA[1:0]	Input	Bank address; SDRAM has 4 banks. Defines which bank to issue command to. Also used for LMR command (see Figure 4).
A[10:0]	Input	Address bits. During ACTIVE command, A[10:0] = row address. During READ/WRITE command, A[7:0] = column address and A[10] enables (HIGH) or disables (LOW) auto-precharge. Also used for LMR command (see Figure 4).
DQ[31:0]	I/O	Data I/O to/from SDRAM.
V_{DDQ}	Supply	DQ power.
V_{SSQ}	Supply	DQ ground.
V_{DD}	Supply	SDRAM power.
V_{SS}	Supply	SDRAM ground.
NC	-	Not connected.
NU	-	Not used.

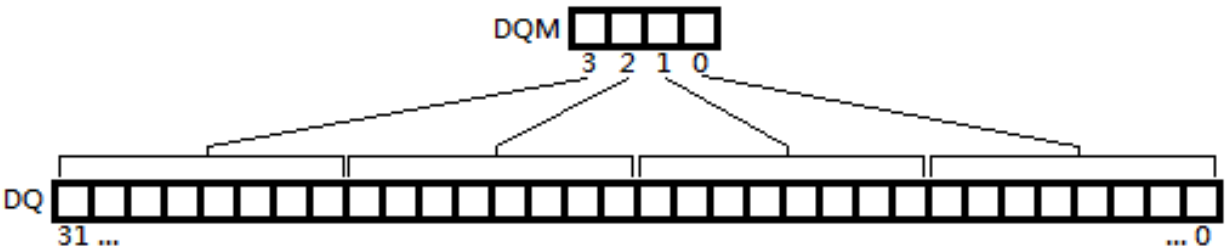


Figure 3:
DQM to DQ Relation

Mode Register

The mode register defines the operation of the SDRAM chip which involves the write-burst mode, the CAS latency (number of clock cycles between a READ and the data being output on DQ), the burst type, and the burst length.

The mode register is programmed with the LOAD MODE REGISTER (LMR) command and it must be done when all banks are idle (done with PRECHARGE command).

LMR command uses BA[1:0] as the most significant 2 bits and A[10:0] as the least significant 11 bits for the mode register.

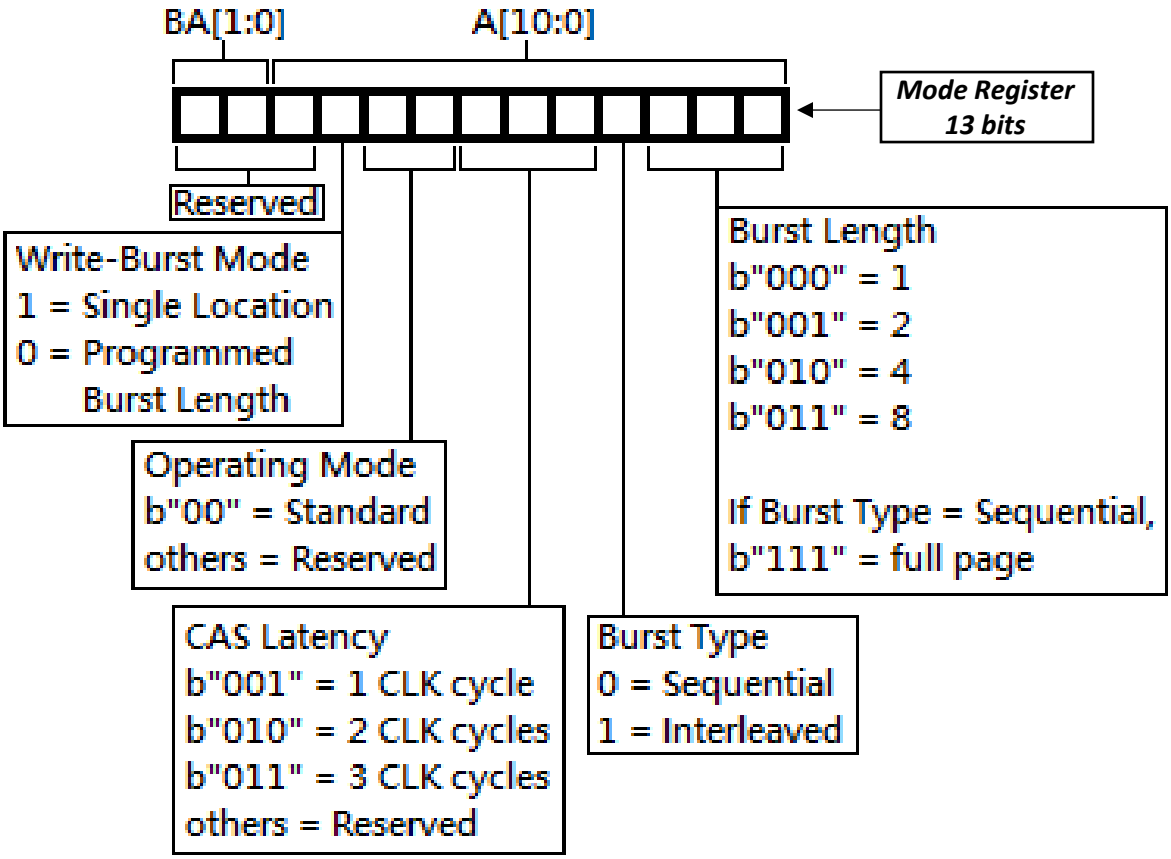


Figure 4:
Mode Register

Commands

Command Name	CS#	RAS#	CAS#	WE#	DQM	BA[1:0]	A[10:0]	DQ
COMMAND INHIBIT	H / 1	X	X	X	X	X	X	X
NO OPERATION	L / 0	H / 1	H / 1	H / 1	X	X	X	X
ACTIVE	L / 0	L / 0	H / 1	H / 1	X	Bank	Row	X
READ	L / 0	H / 1	L / 0	H / 1	L/H / 0/1	Bank	Column	Output
WRITE	L / 0	H / 1	L / 0	L / 0	L/H / 0/1	Bank	Column	Input
BURST TERMINATE	L / 0	H / 1	H / 1	L / 0	X	X	X	X
PRECHARGE	L / 0	L / 0	H / 1	L / 0	X	X	All/Single	X
REFRESH	L / 0	L / 0	L / 0	H / 1	X	X	X	X
LOAD MODE REGISTER	L / 0	L / 0	L / 0	L / 0	X	MR[12:11]	MR[10:0]	X
READ/WRITE Enable	X	X	X	X	L / 0	X	X	Valid
READ/WRITE Inhibit	X	X	X	X	H / 1	X	X	High-Z

Notes:

PRECHARGE
If A[10] = 1, precharge all banks (deactivate all rows).
If A[10] = 0, precharge bank determined by BA[1:0] (deactivate all rows in that specific bank).

REFRESH
If CKE = 1, REFRESH is AUTO REFRESH command and must be explicitly issued.
If CKE = 0, REFRESH is SELF REFRESH command and the SDRAM refreshes itself.

Timings

CAS Latency, CL

CAS – Column Address Strobe (deprecated term)

It is the number of clock cycles between sending a column address to the SDRAM and the data being available on DQ. [1, 2]

Can be set to 1, 2, or 3 clock cycles by programming the mode register.

Mode Register Delay, t_{MRD}

Delay between programming the mode register and issuing another command.

Row Active Time, t_{RAS}

The minimum number of clock cycles required between ACTIVE and issuing PRECHARGE. In SDRAM modules, this is t_{RCD} + CL. This is the time needed to internally refresh the row and overlaps with t_{RCD}. [1]

Row Cycle Time, t_{RC}

The minimum time in cycles it takes a row to complete a full cycle, which is $t_{RC} = t_{RAS} + t_{RP}$. If this is set too short it can cause corruption of data and if it is too high, it will cause a loss in performance, but increase stability. [3]

Row Address to Column Address Delay, t_{RCD}

The minimum number of clock cycles required between opening a row of memory (ACTIVE) and accessing columns within it (READ/WRITE). The time to read the first bit of memory from a DRAM without an active row is t_{RCD} + CL. [1]

Row Refresh Cycle Timing, t_{RFC}

Determines the number of cycles to refresh a row on a memory bank. If this is set too short, it can cause corruption of data and if it is too high, it will cause a loss in performance, but increased stability. [3]

Row Precharge Time, t_{RP}

The minimum number of clock cycles required between issuing PRECHARGE and opening the next row. (1)

Timings according to the datasheet:

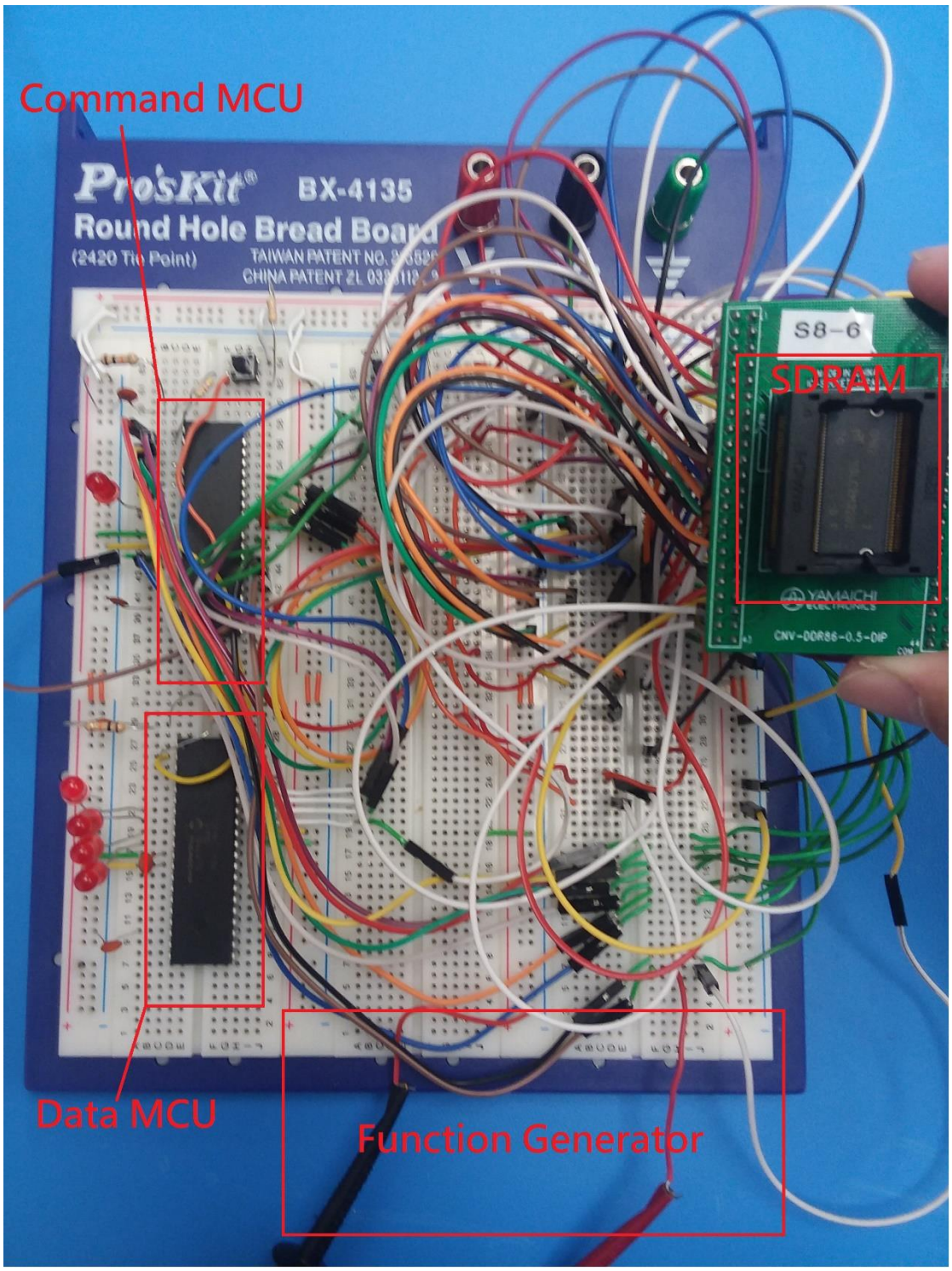
Parameter		143 MHz		167 MHz		167 MHz		183 MHz		200 MHz	
		Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.
ACTIVE to PRECHARGE	t_{RAS}	42	120k	42	120k	42	120k	38.7	120k	38.7	120k
Mode Register Delay	t_{MRD}	14	-	12	-	12	-	11	-	10	-
ACTIVE to ACTIVE	t_{RC}	70	-	60	-	60	-	55	-	55	-
ACTIVE to READ/WRITE Delay	t_{RCD}	20	-	18	-	18	-	16.5	-	15	-
AUTO REFRESH Period	t_{RFC}	70	-	60	-	60	-	60	-	60	-
PRECHARGE Period	t_{RP}	20	-	18	-	18	-	16.5	-	15	-

ns

System

Description

This system uses two MCUs: one for issuing commands to the SDRAM chip and to handle timing (“Command MCU”), and one to connect to the SDRAM’s DQ I/O bus (“Data MCU”). A function generator is used as the SDRAM’s clock signal. A power supply at 3.3 V is used to power the whole system.



Pin Mapping

The PIC18F45K50 has 5 8-bit ports: PORTA through PORTE.

PORTA, PORTB, and PORTD each have all their 8 bits available for general-purpose use.

PORTC has 5 bits and PORTE has 3 bits available for general-purpose use.

To lessen the wires in the design, only 4 pins of DQ are being used.

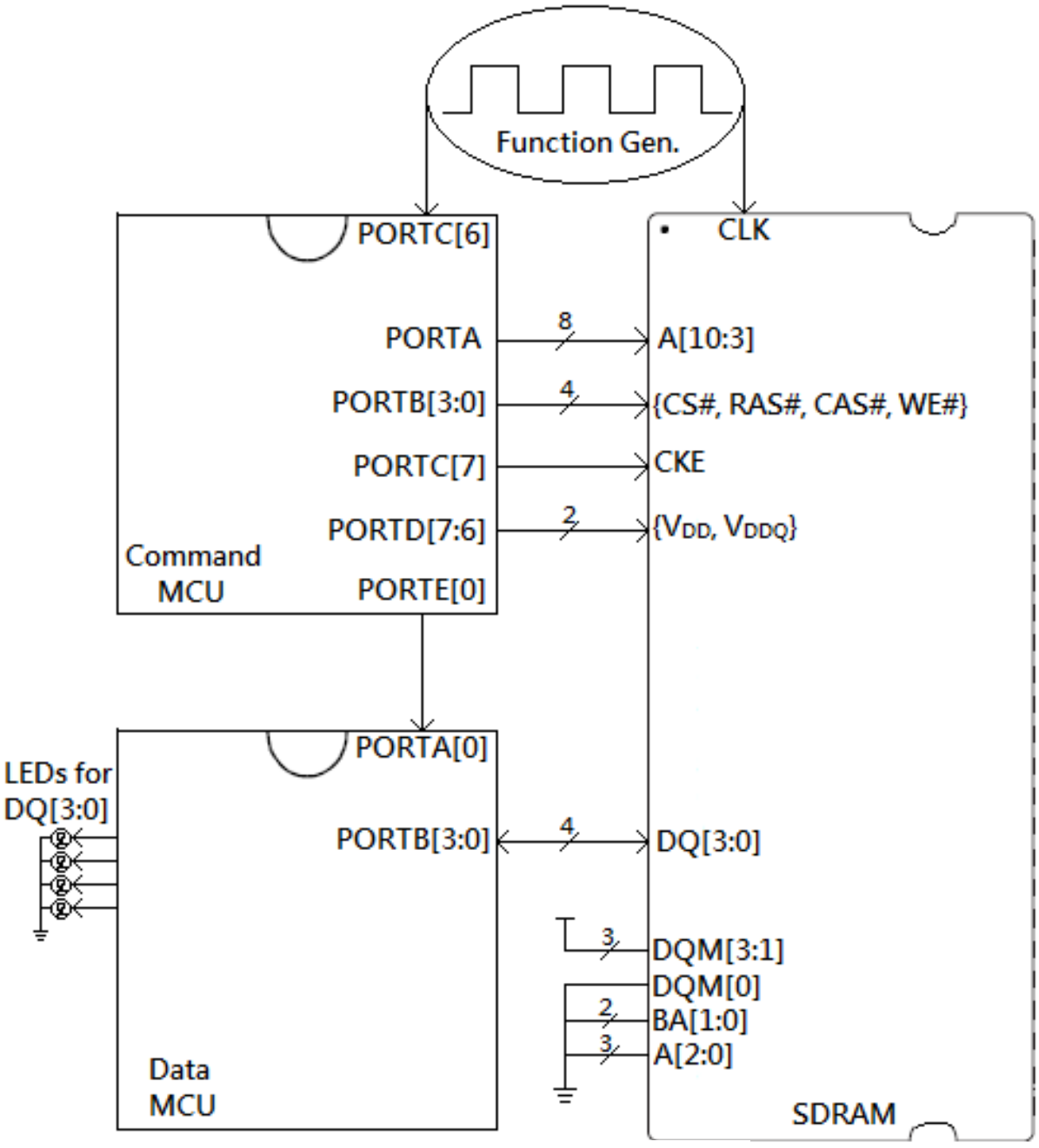


Figure 5:
System Diagram

Command MCU	Data MCU	SDRAM
PORTA →		→ A[10:3]
PORTB[3:0] →		→ {CS#, RAS#, CAS#, WE#}
PORTC[7] →		→ CKE
PORTD[7:6] →		→ {V_{DD}, V_{DDQ}}
PORTE[0] →	→ PORTA[0]	
	PORTB[3:0] ↔	↔ DQ[3:0]
		V_{DD} → DQM[3:1]
		GND → DQM[0]
		GND → BA[1:0]
		GND → A[2:0]

See page 11 and 17 for the pin mapping code.

Timing

The PIC18F45K50 MCUs are using their 16 MHz internal oscillator.

The MCU's smallest instruction (16 bits) uses 4 cycles, so 1 instruction cycle = 4 clock cycles, meaning 4 M instruction cycles per second.

Setting up one of the 8-bit ports in the MCU (loading and outputting a value) requires two instruction cycles.

This is important for constructing the code because of the SDRAM's timing constraints; the SDRAM expects cycles of NOPs the cycle immediately after issuing some commands.

The commands the system is currently using that have timing constraints are PRECHARGE, AUTO REFRESH, LOAD MODE REGISTER, ACTIVE, READ, and WRITE.

Constraint	Requirement	Commands
t_{MRD}	2 cycles	LOAD MODE REGISTER
t_{RCD}	3 cycles	ACTIVE
t_{RFC}	2 cycles	AUTO REFRESH
t_{RP}	3 cycles	PRECHARGE, READ/WRITE (when auto PRECHARGE is enabled)

For example, if I issue a PRECHARGE command, the SDRAM needs the next 3 cycles to be NOP commands before issuing the next command. As mentioned before, setting up a port, specifically PORTB which is being used for issuing commands, takes 2 cycles. Because of that, the NOP command cannot be issued the cycle immediately after issuing a PRECHARGE which will cause unwanted behavior from the SDRAM.

To try and solve this problem, the clock signal used for the SDRAM is 400 kHz, which is a tenth of the MCUs' instruction cycles per second, meaning there is a buffer time between the SDRAM clock's rising edge in which the command MCU can set up its ports and fulfill the SDRAM's timing constraints (see Figure 6).

I haven't done an exact timing analysis, but I think the 10 MCU buffer cycles between each SDRAM cycle should be sufficient.

Also, the SDRAM requires a REFRESH every 15.625 µs to maintain data integrity.

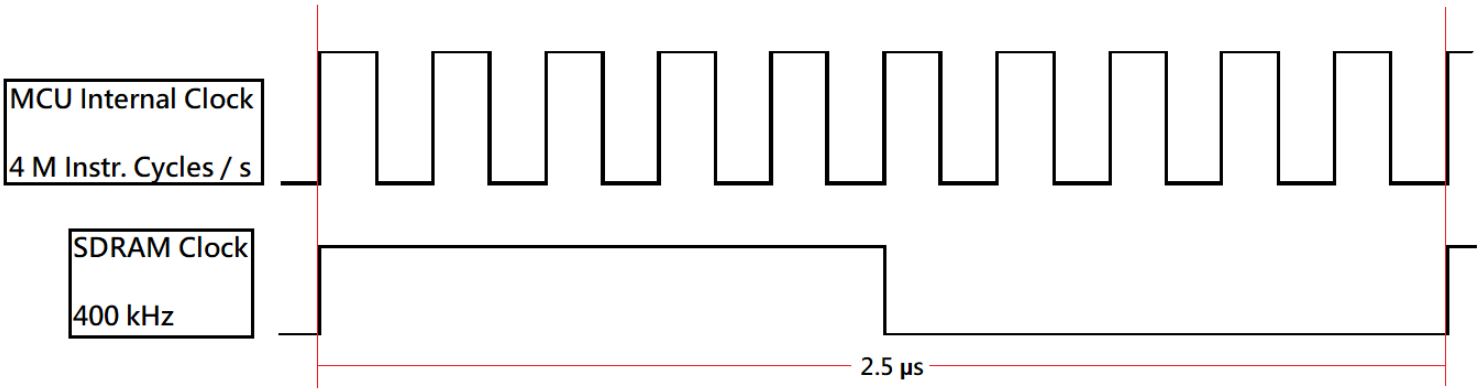


Figure 6:
MCU to SDRAM Clock Signal Visual

Code

[Code documentation omitted]

Design Errors

- The current system ties DQM[3:1] high and DQM[0] low. I've observed that V_{DD} stays high due to this, regardless of the output from the Command MCU's POWER port.

- The GO button works on the first iteration. Pressing STOP and GO usually produces incorrect results. The design needs to be disconnected from power before pressing GO again which will produce consistent, correct results.

- Supplying power to the design with my demo board produced random, incorrect results when reading back data from the SDRAM. Switching over to a standalone power supply fixed this and gave consistent, correct data; the data that was written was the same data that was read.

- The DQ signals read from the SDRAM are weak and jittery, probably because of the low clock frequency / low refresh rate.

Miscellaneous

References

1.) **Memory Timings – Wikipedia**
https://en.wikipedia.org/wiki/Memory_timings#:~:text=The%20timing%20of%20modern%20synchronous,%2D8%2D8%2D24.

2.) **What Is CAS Latency in RAM? CL Timings Explained – Tom's Hardware**
https://www.tomshardware.com/reviews/cas-latency-ram-cl-timings-glossary-definition,6011.html

3.) **Memory Timings Explained – TechPowerUp**
https://www.techpowerup.com/articles/overclocking/AMD/memory/131/2