eMMC Technology Application

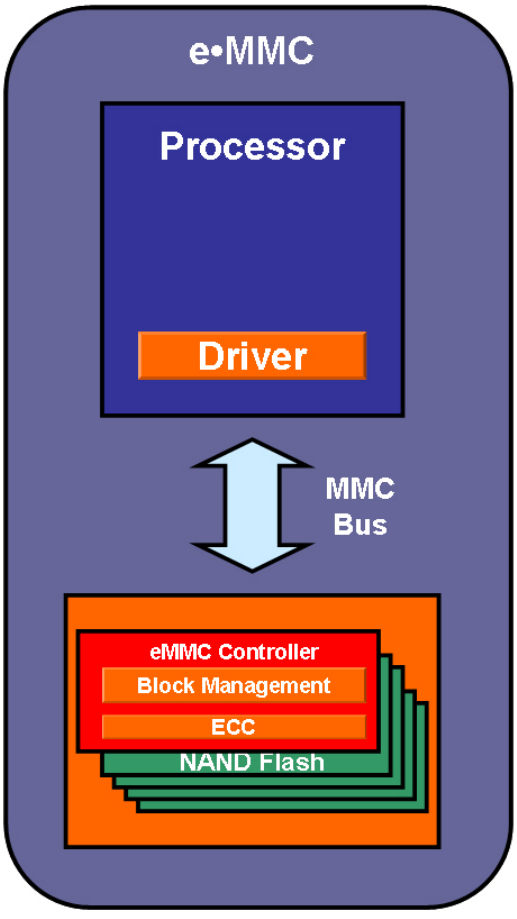
Preface

eMMC, short for Embedded MultiMediaCard, is the embedded memory standard specification defined by the MMC (MultiMediaCard Association). It is mainly applied in mobile devices, such as mobile phone, smart phone, tablet computer, notebook computer, etc., which are all characterized by light and thin. eMMC is designed just for reducing the component size.

eMMC packages the NAND Flash chip and Control chip into one chip by the MCP technology, so as to simplify the memory design, and further reduce the components and increase the PCB size.

eMMC Structure

eMMC consists of one embedded storage project. In its inside MMC Interface, NAND Flash and Control are packaged into one small chip by BGA. In short words, the Host processor only needs to transmit commands through the MMC Bus without doing any commuting about memory management, such as ECC, Wear-Leveling and BBM, which are done by the Control chip that packaged with the NAND Flash. This is a good way to save the R&D cost.



▲ **Fig.1: eMMC Structure**

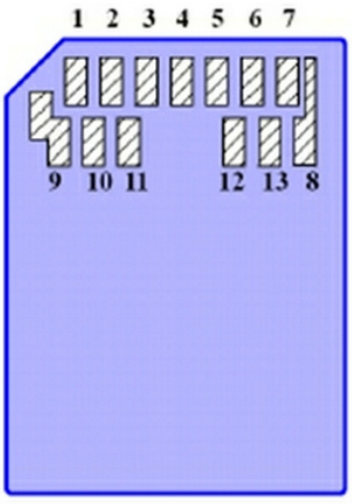
eMMC Specification

For eMMC, its transmission speed can be up to 52MB/s; it supports two voltages: 1.8v and 3.3v; and it supports updating. For its specification, eMMC 4.2 defines the eMMC's interface capacity (larger than 2GB) and speed (52MB/s); eMMC 4.3 provides new functions of Booting Function, Explicit Sleep Mode, Reliable Write, etc.; and the latest eMMC 4.4's double data rate (DDR) is strengthened, which increases the original 52MB/s to 100MB/s; besides, the Multiple Partitioning is added for defining the chip by two parts: High Density and High Performance. Users can store the codes and system data in the High Performance part, so as to use the block capacity more flexibly. And write-protection is also added to ensure the security.

Table 1 and Fig.2 show MMC's pins and appearance respectively.

Pin	Name
1	Data3
2	Command
3	VSS
4	VDD
5	Clock
6	VSS(2)
7	Data0
8	Data1
9	Data3
10	Data4
11	Data5
12	Data6
13	Data7

▲ **Table 1: MMC pins and their names.**



▲ **Fig.2: MMC appearance**

eMMC's characters and advantages

- Has these System Voltage (VDD) Ranges :

	High Voltage MultiMediaCard	Dual Voltage MultiMediaCard
Communication	2.7 - 3.6	1.70 - 1.95, 2.7 - 3.6 (Note 1)
Memory Access	2.7 - 3.6	1.70 - 1.95, 2.7 - 3.6
NOTE 1 VDD range: 1.95V - 2.7V is not supported.		

- Ten-wire bus (clock, 1 bit command, 8 bit data bus) and a hardware reset.
  - Clock frequencies of 0-200MHz
  - Three different data bus width modes: 1-bit , 4-bit, and 8-bit

- Data protection Mechanisms (Write Protection Types) :
  - Password
  - Permanent
  - Power-On
  - Temporary

- Different types of error protected read and write modes:
  - Single Block
  - Multiple Block

- Data Removal Commands:
  - Erase
  - Trim
  - Sanitize

- Enhance host and device communication techniques to improve performance
  - Power Off Notification
  - High Priority Interrupt
  - Background Operations
  - Partitioning
  - System data tagging

- Introduces dual data rate transfer (DDR mode)

www.zeroplus.com.tw TEL:+886 2-66202225 FAX:+886 2-22234362

eMMC Operating Mode

- The Host controls all the transmission between Host and Card; its control command has two types: Broadcast and Addressed (point-to-point transmission)

Broadcast commands	The Host transmits a command to all MMC cards, but only some cards will respond.
Addressed (point-to-point) commands	The Host transmits a command to some particular MMC card, and the card will respond to this command.

- eMMC has five operating modes: Boot Mode, Card Identification Mode, Interrupt Mode, Data Transfer Mode and Inactive Mode. See Table 2.

Boot mode	After power on, if the Card receives the CMD0(GO_Idle_State) with argument(0xF0F0F0F0), then it will enter into the Boot Mode if it supports, or enter into the Identification Mode.
Card identification mode	After power on, the Card will enter into this mode directly or after the Boot Mode, waiting the Host to transmit the CMD3(Send_Relative_Address).
Interrupt mode	The Host and Card will enter into this mode at the same time. There is no data transfer in this mode, which only permits the Host or the Card to transmit the interrupt service request.
Data transfer mode	After receives the RCA transmitted by the Host, the Card will enter into the Data Transfer Mode and wait for the read/write command.
Inactive mode	Caused by incorrect voltage and invalid inserting, the Card will enter into this mode; CMD15(GO_INACTIVE_ STATE) also can make a marked Card enter into the Inactive Mode.

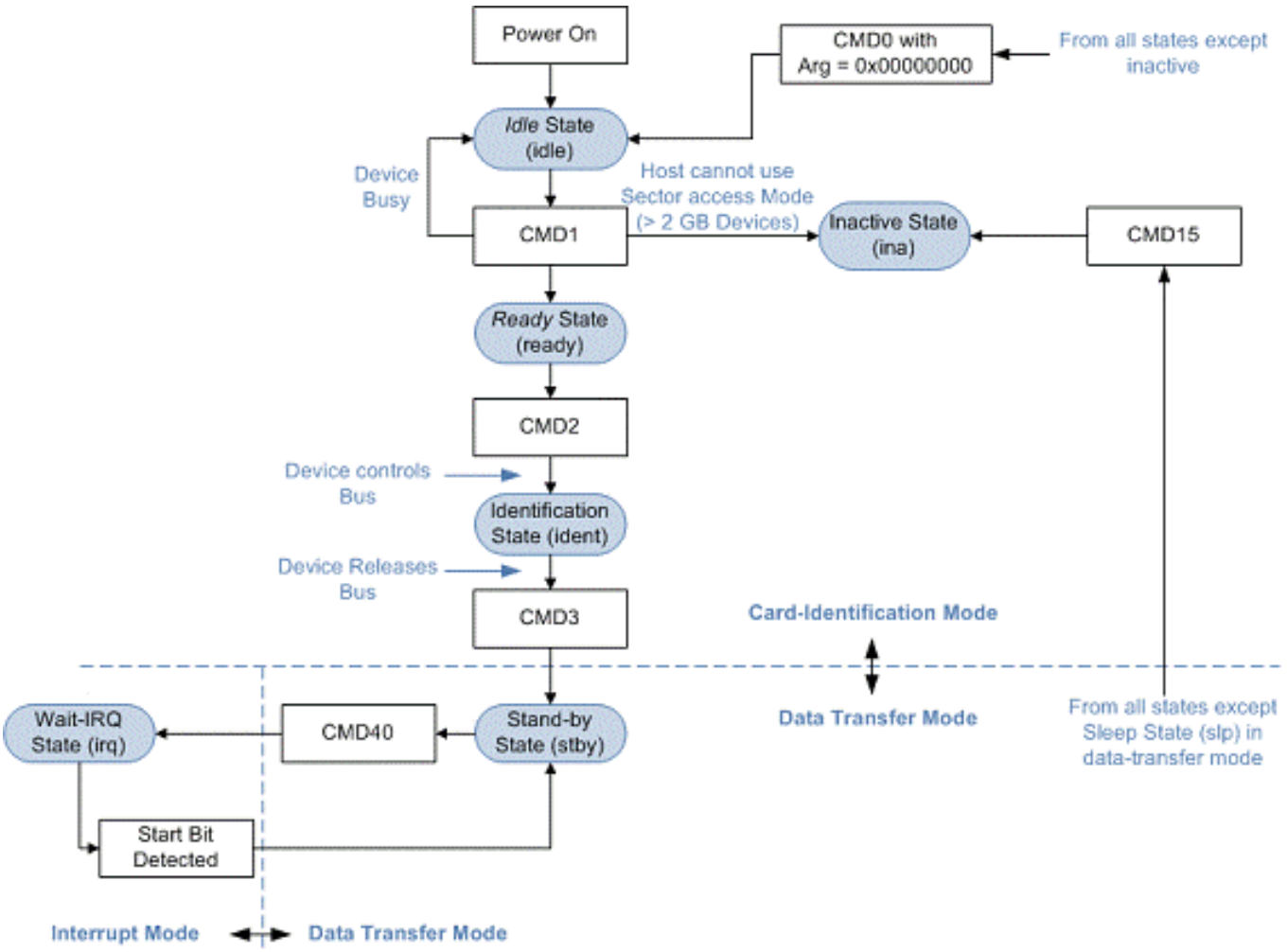
Status (Card)	Operating Mode	Cable Mode
Inactive State	Inactive mode	Open-drain (Output in low voltage)
Pre-Idle State	Boot mode	
Pre-Boot State		
Idle State	Device identification mode	
Ready State		
Identification State		
Stand-by State	Data Transfer mode	Push-pull (Output in high voltage)
Sleep State		
Transfer State		
Bus-Test State		
Sending-data State		
Receive-data State		
Programming State		
Disconnect State		
Boot State	Boot mode	
Wait-IRQ State	Interrupt mode	Open-drain

▲ **Table 2: Table of comparisons between operating mode and status.**

- Current the eMMC 4.5 Specification defines 4 speed modes, see table 3.

Mode Name	Data Rate	I/O Voltage	Bus Width (bit)	CLK Frequency	Max Data Transfer (implies x8 Bus width)
Backwards Compatibility with legacy MMC card	Single	3/1.8/1.2V	1, 4, 8	0-26MHz	26MB/s
High speed SDR	Single	3/1.8/1.2V	4, 8	0-52MHz	52MB/s
High Speed DDR	Dual	3/1.8/1.2V	4, 8	0-52MHz	104MB/s
HS200	Single	1.8/1.2V	4, 8	0-200MHz	200MB/s

▲ **Table.3: 4 speed modes.**



▲ **Fig.3: State flow of eMMC card after power on (Identification Mode).**

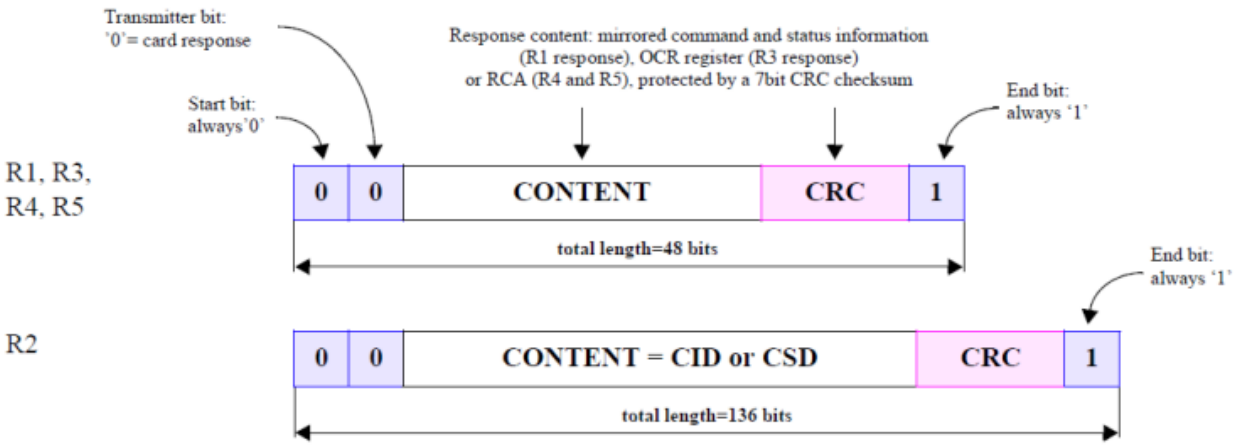
Normally, after power on, if not enter into the Boot Mode, the eMMC will enter into the Identification Flow immediately. Then the Host (Card Reader or Processor) will transmit the CMD1 (Command 1) to the Device (Card) immediately, requiring the working condition of idle device. The Device will return one of two status to the Host: Busy or Ready. If the Device returns Busy, then the Host will continuously transmit the CMD1 to the Device until it returns Ready and responds the required data. CMD2 and CMD3 follow that. CMD2 is to require the CID number; CMD3 is to transmit the Device RCA address for the Host. Now the registration between the Device and Host is completed; the Device will enter into the Data Transfer Mode and keep stand-by, waiting the Host to transmit any data transfer command or special command.

eMMC Packet and Response

- Packet : There are five packets in the eMMC protocol : Command, Response, Block, CRC Status and Busy. See below descriptions:

  - Command : Command transmitted by the Host with parameters.
  - Response : The response the Device makes to the Command; its inside parameters indicate the Device current status.
  - Block : The basic unit in the eMMC transmission protocol. All data is packaged in the Block and then transmitted.
  - CRC Status : It is always after writing the Block, used to notify the Host about the result of writing the Block.
  - Busy : It is always after writing the Block, used to notify the Host that the Device is busy and can't write the next Block.

- Responses : All responses are transmitted by the Device after receiving the Host command. The response type determines the response length. The basic response structure includes: Start bit, Transmission bit, Content bits, CRC bits and End bit. In the eMMC protocol there are 5 response types: R1, R2, R3, R4 and R5. See below detailed description.



▲ **Fig.4: The basic response packet structure.**

- R1 (normal response command):

Total 48 bits. [45:40] indicate the CMD value; [39:8] indicate the current status of Device. R1b, another format of R1, has a different structure, but owns a selectable Busy signal.

Bit position	47	46	[45:40]	[39:8]	7	0
Width (bits)	1	1	6	32	x	1
Value	"0"	"0"	x	x	CRC7	"1"
Description	**Start bit**	**Transmission bit**	**Command index**	**Device status**	**CRC7**	**End bit**

- R2 (CID, CSD register):

Total 136 bits. Used to respond the content of CID/CSD register. CID register matches with CMD2 and CMD10; and CSD register matches with CMD9.

Bit position	135	134	[133:128]	[127:1]	0
Width (bits)	1	1	6	127	1
Value	"0"	"0"	111111	x	"1"
Description	**Start bit**	**Transmission bit**	**Check bits**	**CID or CSD register incl. internal CRC7**	**End bit**

- R3 (OCR register):

Total 48 bits. Used to respond the content of ORC register. The format of the Device response is R3 only when the Host transmits the CMD1.

Bit position	47	46	[45:40]	[39:8]	[7:1]	0
Width (bits)	1	1	6	32	7	1
Value	"0"	"0"	"111111"	x	"1111111"	"1"
Description	**Start bit**	**Transmission bit**	**Check bits**	**OCR register**	**Check bits**	**End bit**

- R4 (Fast I/O):

Total 48 bits. Used to write or read 8 bits data segment into or from some register. The format of the Device response is R4 only when the Host transmitted the CMD39.

Bit position	47	46	[45:40]	[39:8] Argument field				[7:1]	0
Width (bits)	1	1	6	16	1	7	8	7	1
Value	"0"	"0"	"100111"	x	x	x	x	x	"1"
Description	**Start bit**	**Transmission bit**	**CMD39**	**RCA [31:16]**	**Status [15]**	**Register address [14:8]**	**Read register contents [7:0]**	**CRC7**	**End bit**

- R5 (Interrupt request):

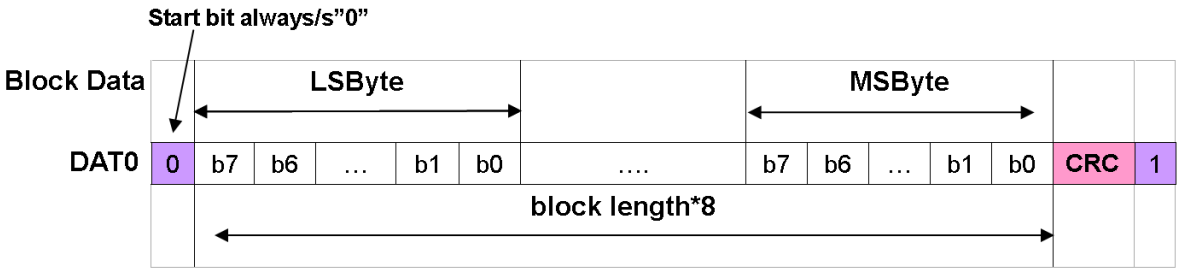
Total 48 bits. It is the response format for interrupt request. When the Host transmits the CMD40 to order some card to enter into the Interrupt Mode, and the Device receives it and responds in R5 format, then the Card can be considered be interrupted and the RCA is set to 0x0.

Bit position	47	46	[45:40]	[39:8] Argument field		[7:1]	0
Width (bits)	1	1	6	16	16	7	1
Value	"0"	"0"	"101000"	x	x	x	"1"
Description	**Start bit**	**Transmission bit**	**CMD40**	**RCA [31:16] of winning Device or of the host**	**[15:0] Not defined. May be used for IRQ data**	**CRC7**	**End bit**

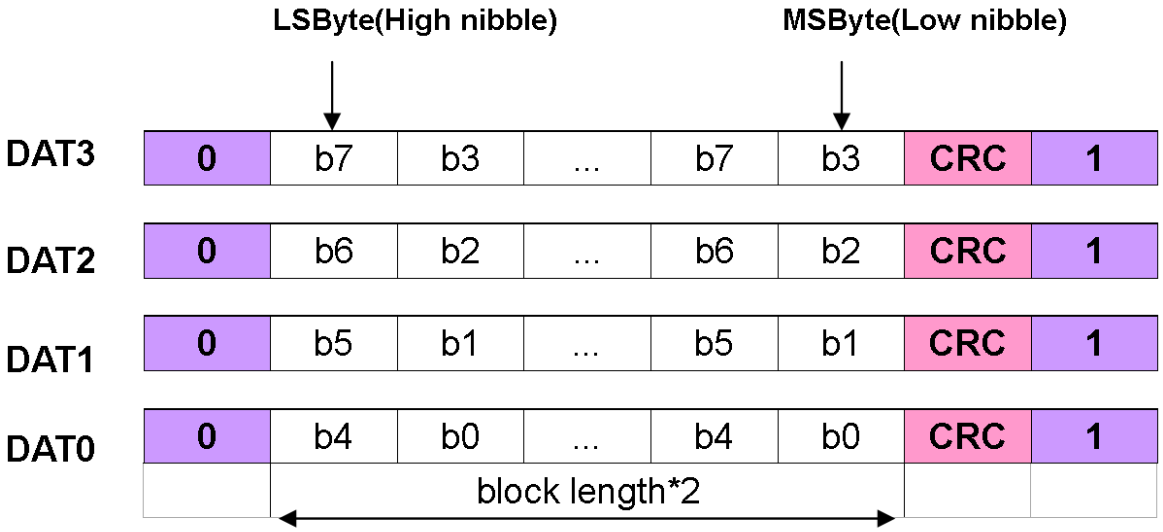
eMMC Transmission Mode

There are 5 transmission modes : 1 bit SDR (1 bit Bus sampling at rising edge), 4 bits SDR (4 bits Bus sampling at rising edge), 8 bits SDR (8 bits Bus sampling at rising edge), 4 bits Bus DDR and 8 bits Bus DDR. All five modes take the Block as the unit of data transmission.
See below diagrams:

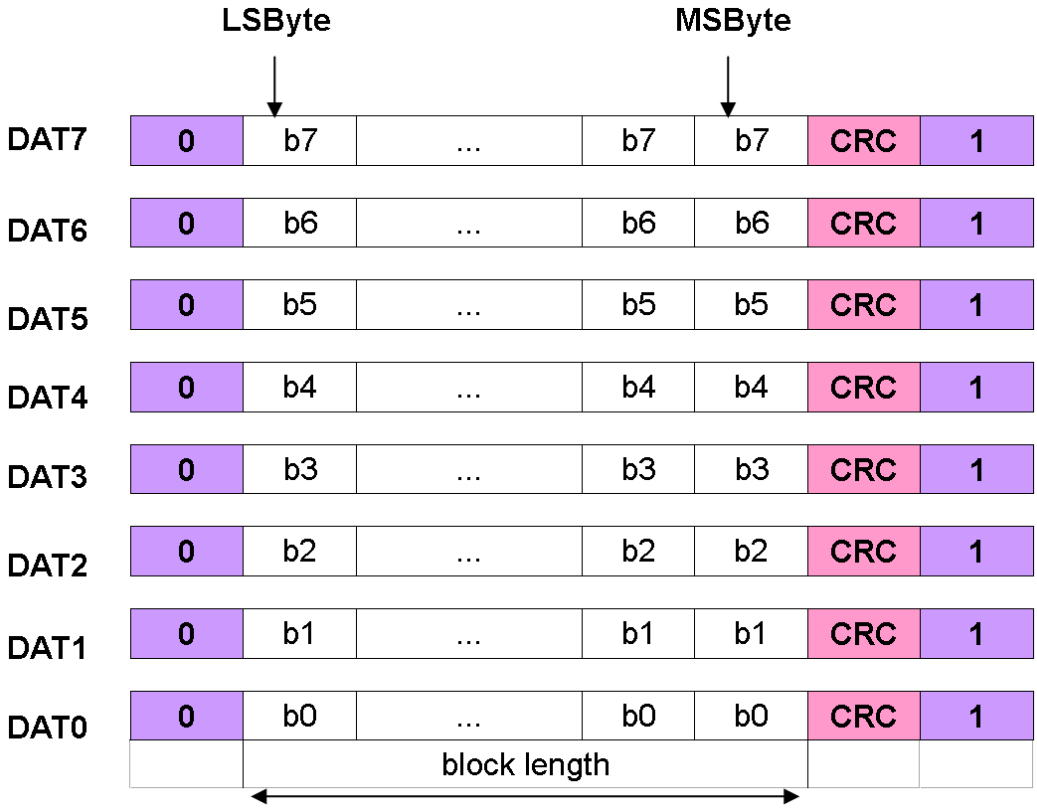
- 1 bit Bus SDR (Single Data Rate) : Sampling at rising edge.
  One data pin (Data 0) is enough.



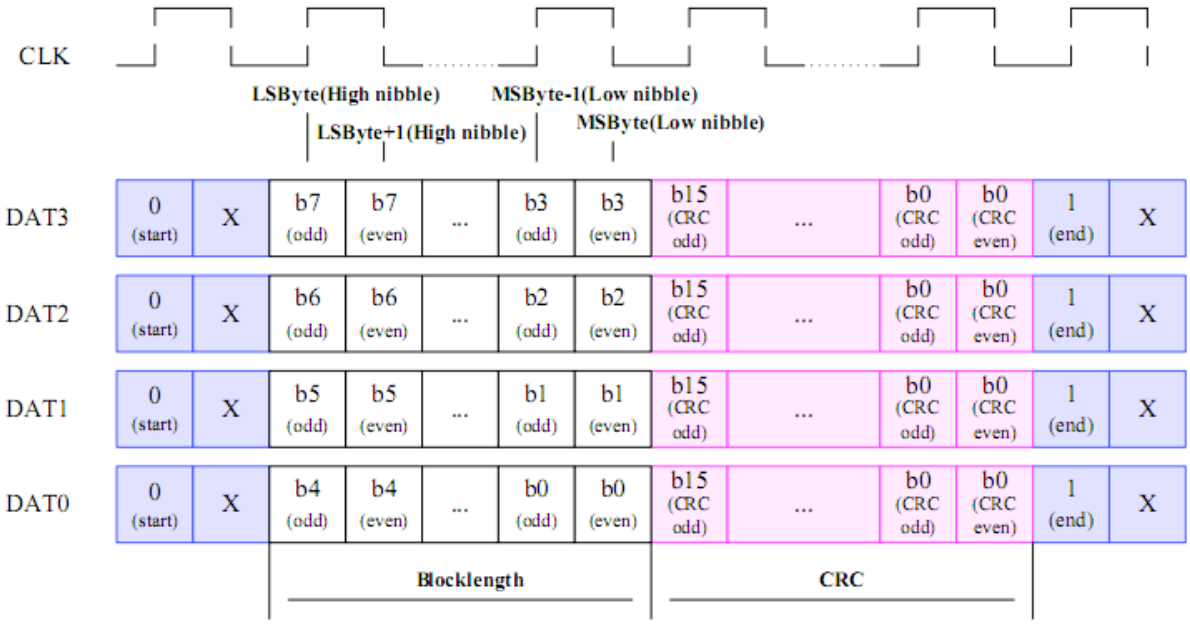
- 4 bit Bus SDR : Sampling at rising edge.
  4 data pins (Data 0-Data 3) are needed.



- 8 bit Bus SDR : Sampling at rising edge.
  4 data pins (Data 0-Data 3) are needed.



- 4 bit Bus DDR (Double Data Rate) :
  The Block and CRC are sampled at the rising/falling edge of CLK. But the start bit and end bit are only sampled at the rising edge.

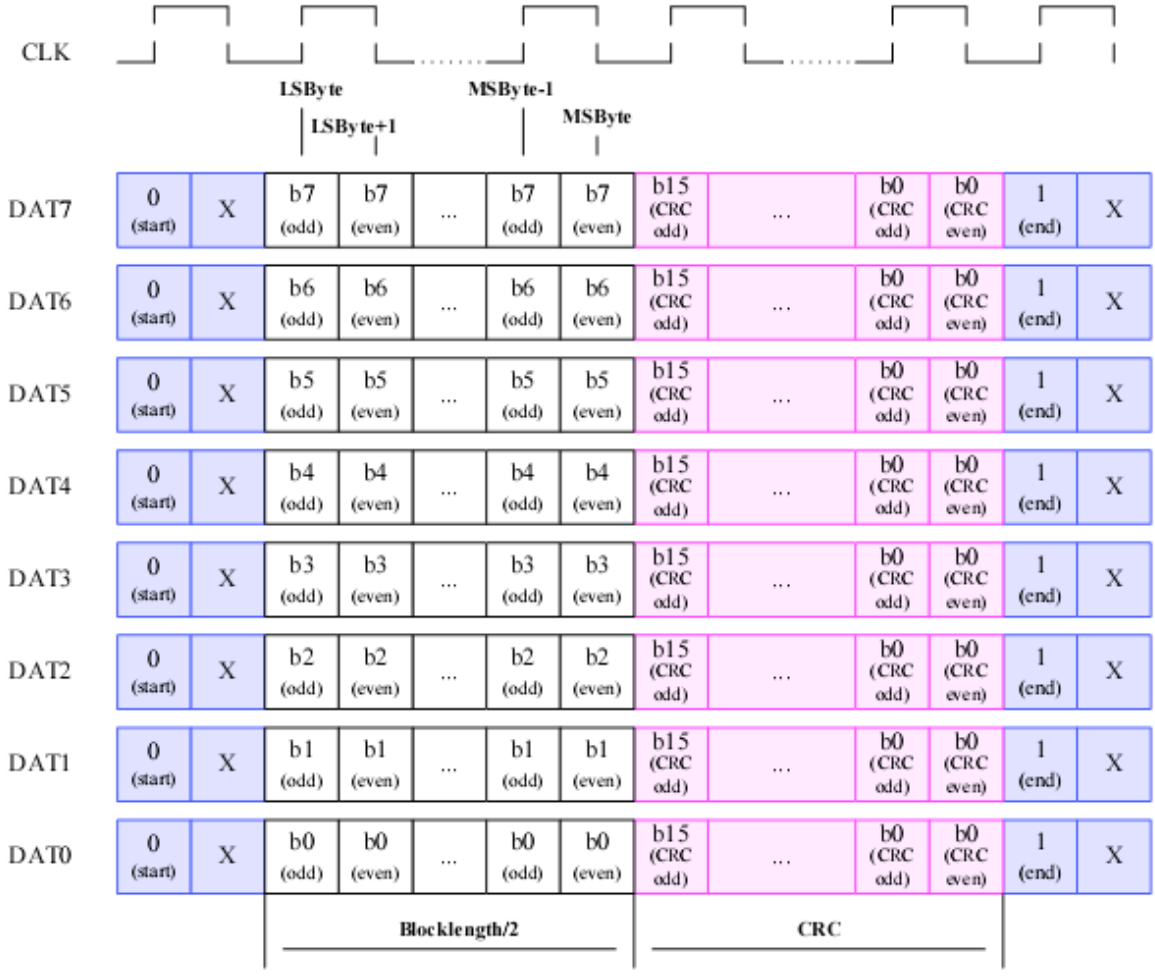


See below image for the DDR sampling mode. It captures the data at the rising/falling edge of CLK, to achieve double-data transmission. Please note that only the start bit and end bit are captured at the rising edge, the Data and CRC are captured at the rising/falling edge.

CLK	↑	↓	↑	↓	↑	↓	.....	↑	↓	↑	↓	.....	↑	↓	↑	↓
DTA3	0	X	b7	b7	b3	b3	.....	b15	b15	b14	b14	.....	b0	b0	1	X
DTA2	0	X	b6	b6	b2	b2	.....	b15	b15	b14	b14	.....	b0	b0	1	X
DTA1	0	X	b5	b5	b1	b1	.....	b15	b15	b14	b14	.....	b0	b0	1	X
DTA0	0	X	b4	b4	b0	b0	.....	b15	b15	b14	b14	.....	b0	b0	1	X
	S	DATA						CRC							E	

- 8 bit Bus DDR (Double Data Rate):

  8 data pins (Data 0-Data 7) are needed. It has the same transmission mode of 4 bits Bus DDR.



eMMC Practical Measurement and Signal Tracing

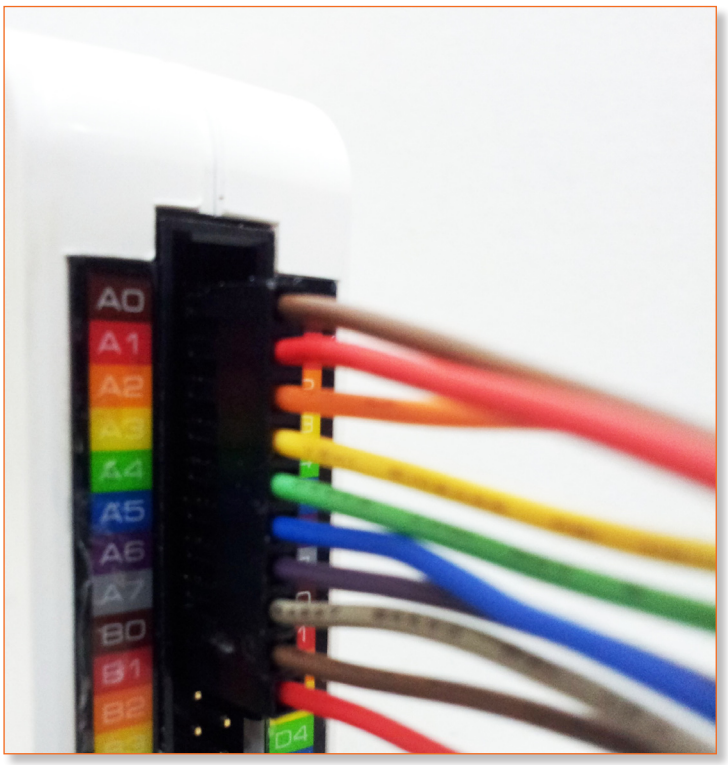
eMMC Protocol Analyzer module, designed by Zeroplus Logic Analyzer, can help engineers measuring the eMMC signal packet. They only need to connect the test line of Logic Analyzer with the eMMC to measure the signal; then, with the humanized software interface, the eMMC can trace any eMMC Command, Response or Data, helping engineers to quicken the R&D process. See Fig.5:

▲ **Fig.5: Connect the card reader and eMMC with the test line, then start the LA to capture.**

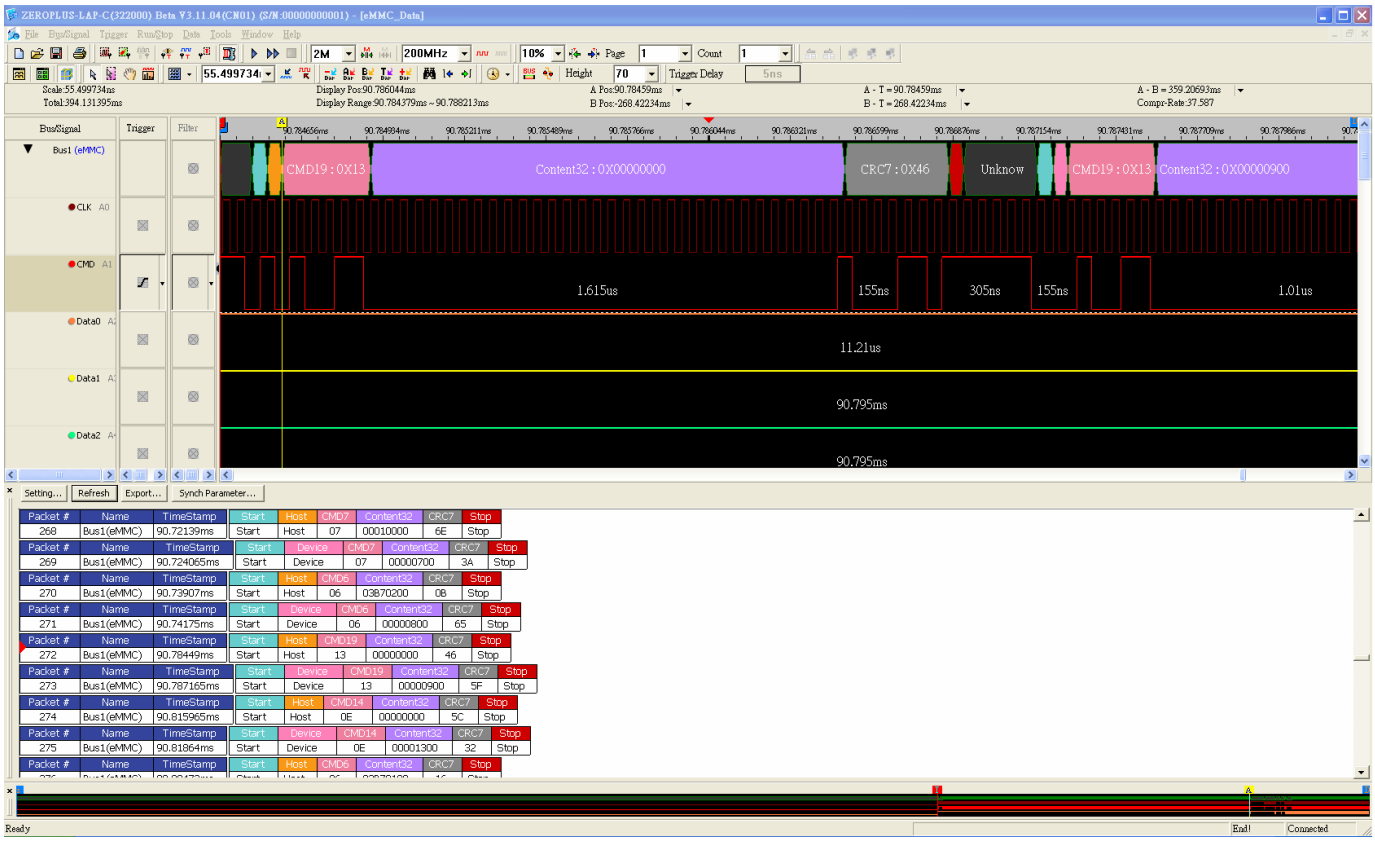
Refer the table 4 for the pins and their names, connect the lines of Command, Clock and Data0-Data7 with A0, A1, A2, A3, A4, A5, A6, A7, B0 and B1. See Fig.6.

Pin	Name
1	Data3
2	Command
3	VSS
4	VDD
5	Clock
6	VSS(2)
7	Data0
8	Data1
9	Data3
10	Data4
11	Data5
12	Data6
13	Data7

▲ **Table 4**

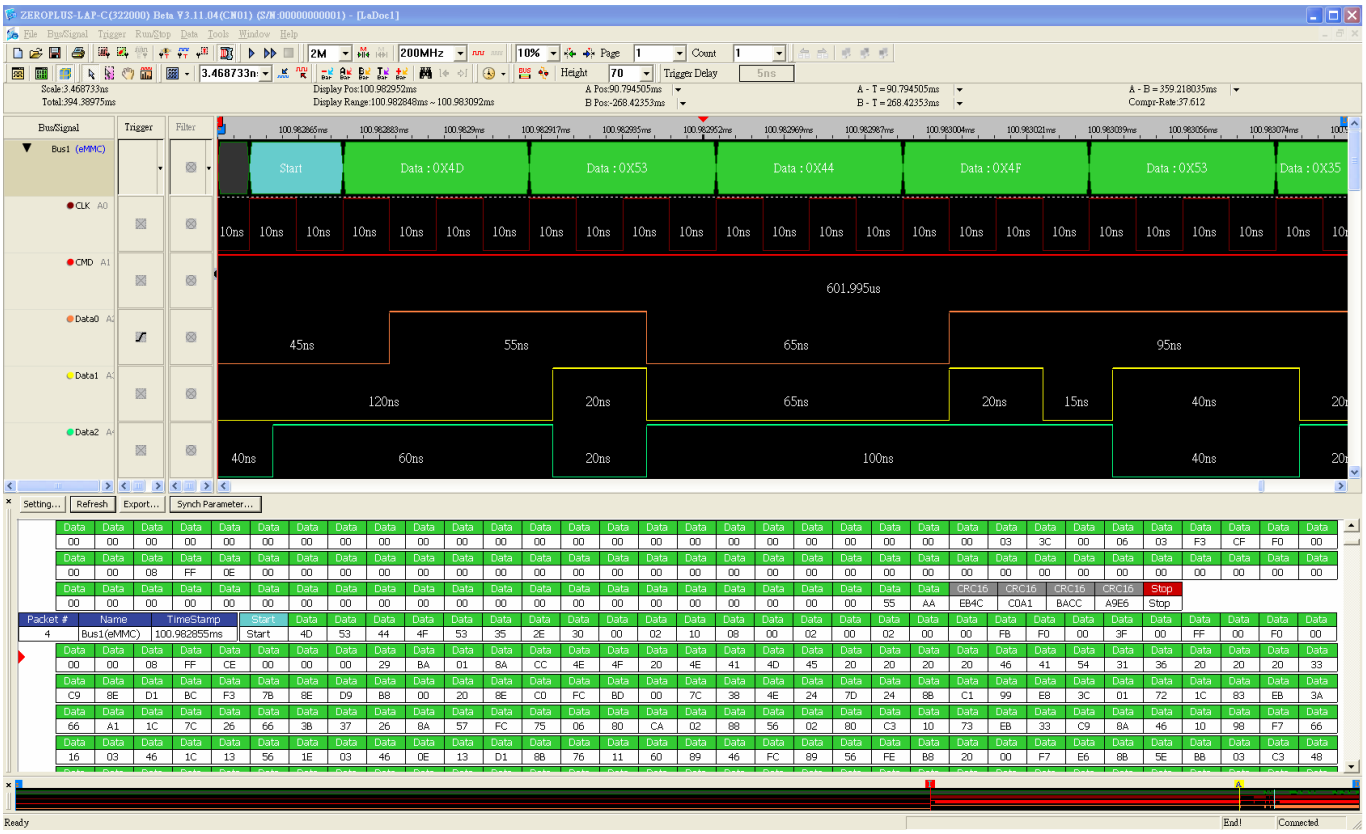


▲ **Fig.6**



▲ **Fig.7: Use Zeroplus Logic Analyzer to measure the eMMC CMD.**

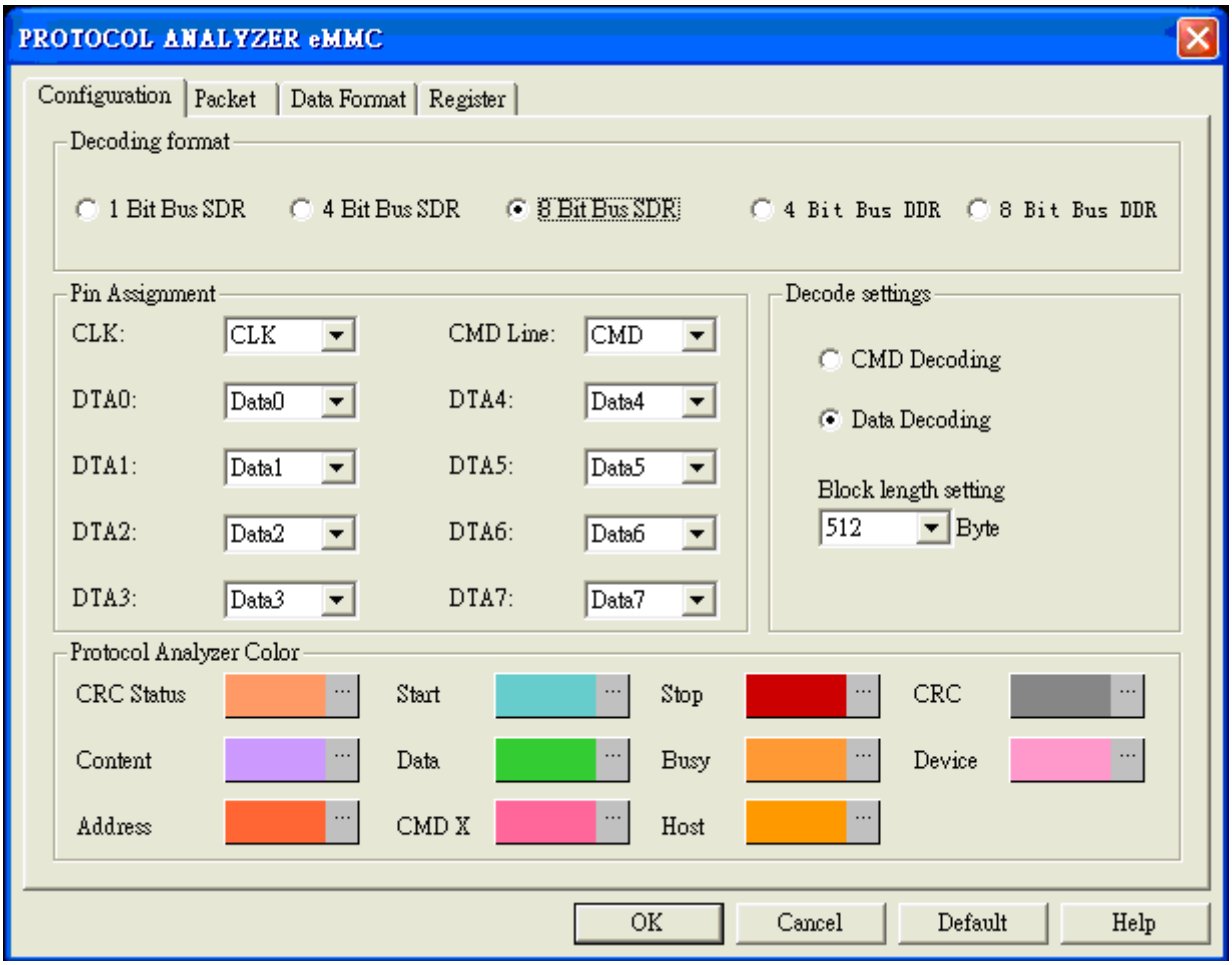
www.zeroplus.com.tw TEL:+886 2-66202225 FAX:+886 2-22234362



▲ **Fig.7: Use Zeroplus Logic Analyzer to measure the eMMC Data.**

eMMC Protocol Analyzer Module Introduction

The eMMC Protocol Analyzer module of Zeroplus Logic Analyzer can analyze the Command and Data of eMMC Bus. Users only need to do some settings in the Bus setting interface, which has four sections: Decoding format, Pin Assignment, Decode settings, Protocol Analyzer Color. See Fig.9.



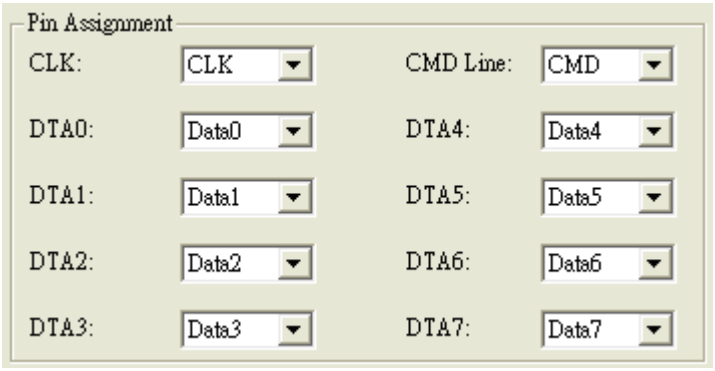
▲ **Fig.9: eMMC Protocol Analyzer module setting window.**

- Decoding format:

  Select the eMMC's transmission mode, which determines the Logic Analyzer can decode the data correctly or not.

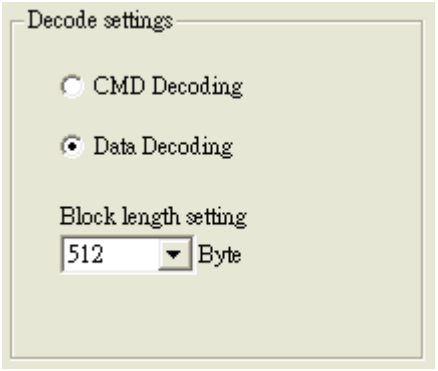
- Pin Assignment:

  Set the pin connection when analyzing the eMMC. Make sure the connection is correct.



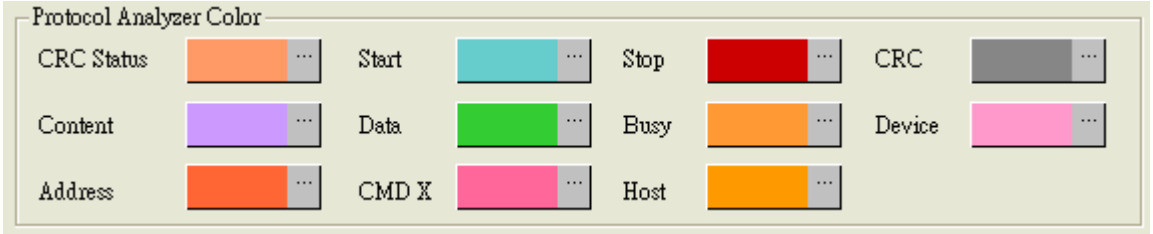
- Decode settings:

  Set the Bus to decode at the CMD line or Data line; if at the Data line, users must set the Block length (2-16KBytes).



- Protocol Analyzer Color:

  Self-set the color of each packet.



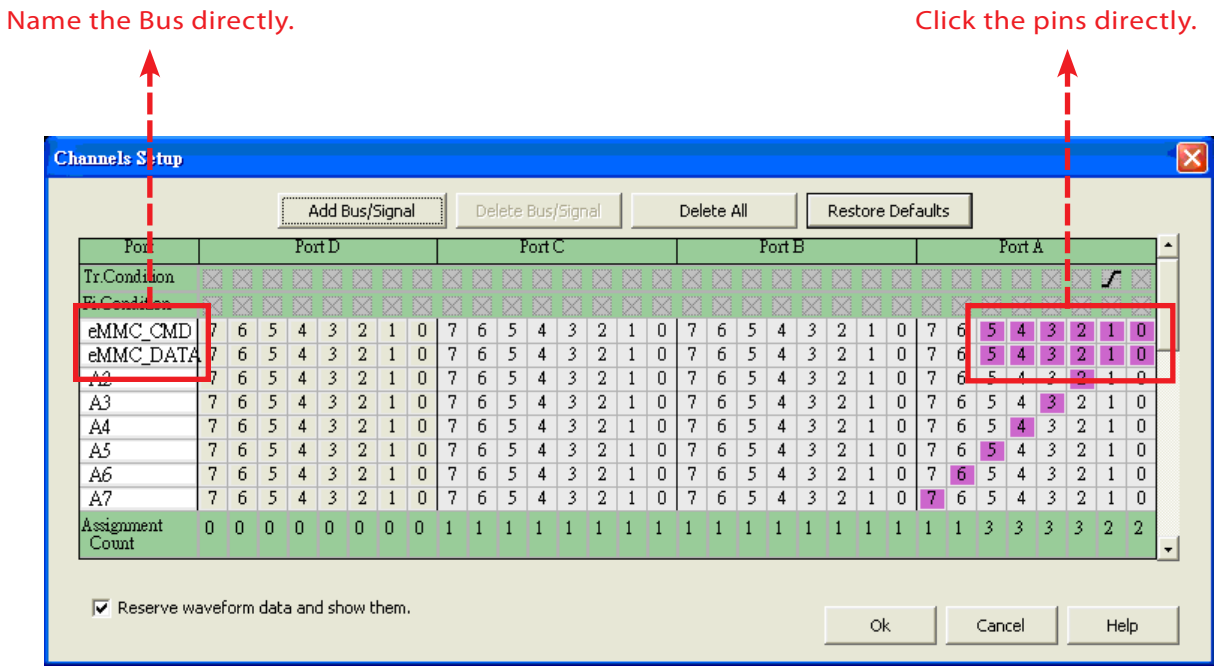
www.zeroplus.com.tw TEL:+886 2-66202225 FAX:+886 2-22234362

If users want to analyze both the Command and the Data, they can group two Buses to decode, one for the Command and one for the Data.

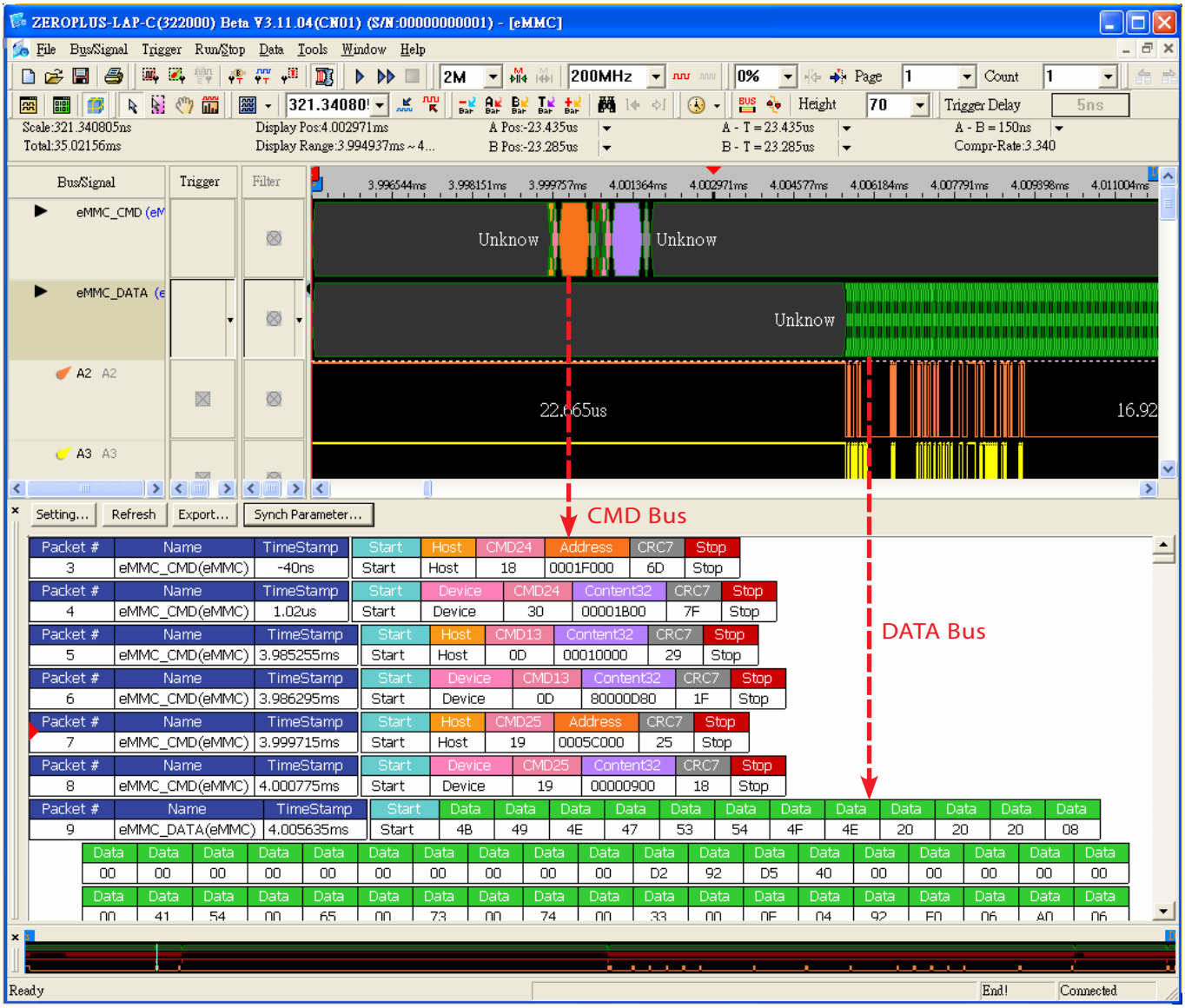
Setting Steps:

Step 1 : Click “Bus/Signal” --> “Channels Setup”.

Step 2 : In the “Channels Setup” window, select A0, A1, A2, A3, A4 and A5 for each Bus. See Fig.8 and Fig.9.



▲ **Fig.10: Channels Setup**



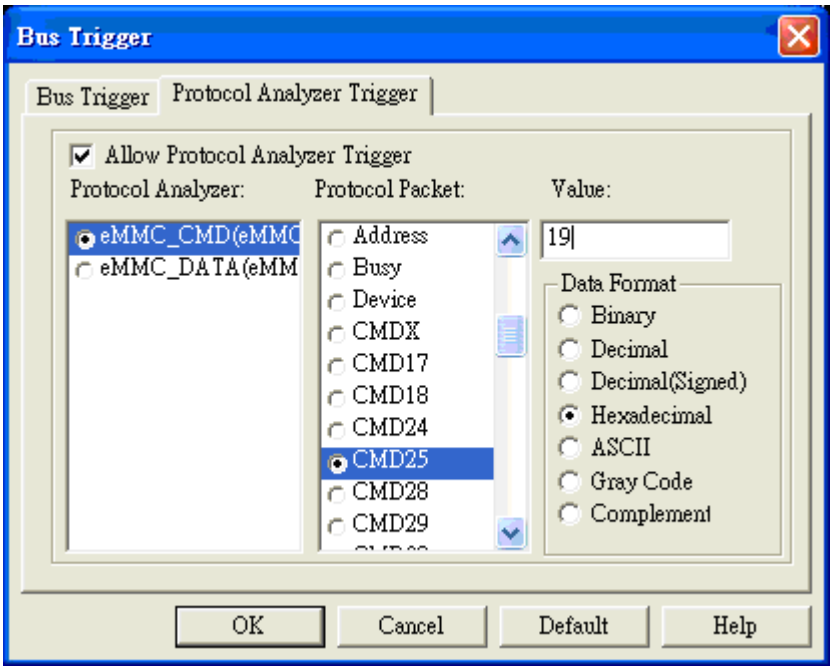
▲ **Fig.11: Analyze the Command and Data at the same time.**

Practical Application: CMD Tracing and Debug Program

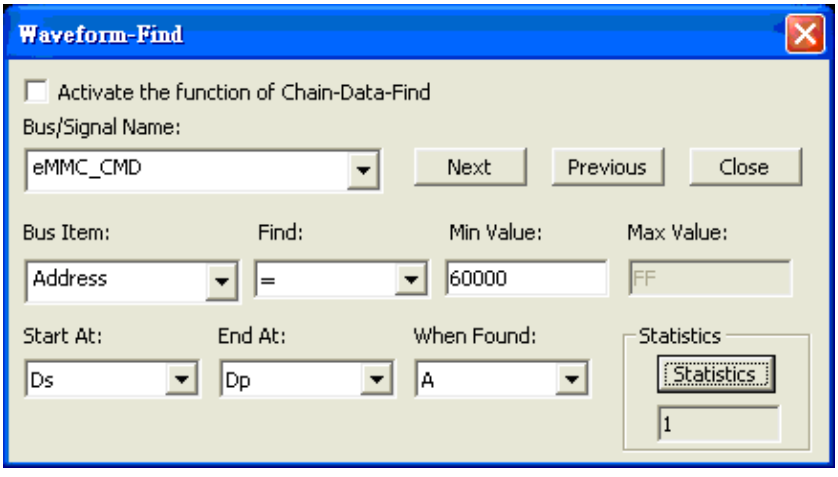
In some eMMC R&D project, the codes are written and compiled, and no problem is found in its basic read/write test. But in some location the data is error in reading/writing. So is there any better method to find the error data except using the software of Hex Editor, Ultra Edit, etc.?

Now you can use Zeroplus Logic Analyzer to trace some CMD, Address or Data, and capture its data to find the bug. See below image, the bug is found through tracing the particular CMD.

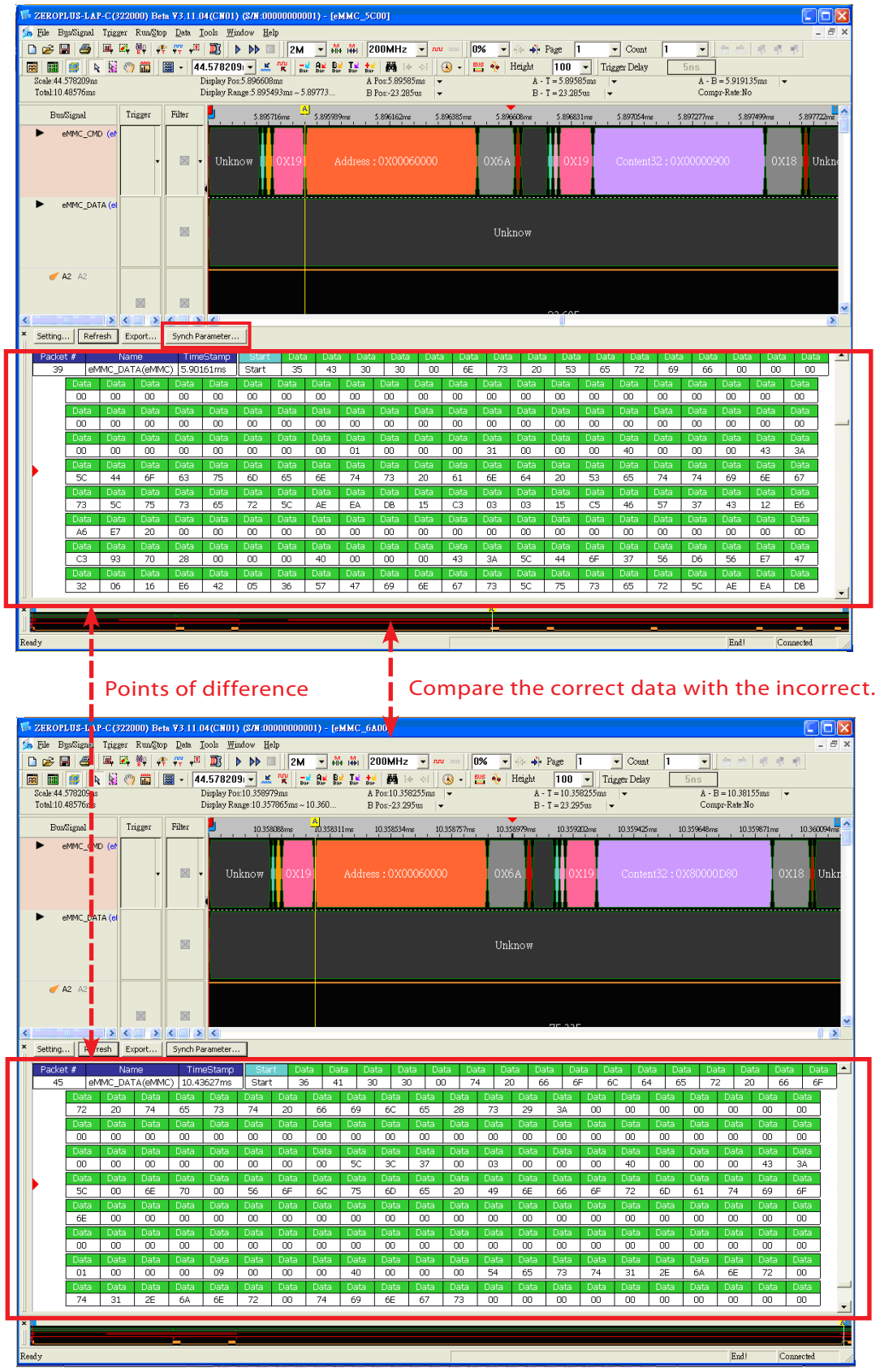
Step 1 : Trace the CMD25(WRITE_MULTIPLE_BLOCK) and capture the written data.



Step 2 : Find the particular location Address 0x60000 to see the data writing status behind that.



Step 3 : Use the 'Bus Packet List' function and set the packet be synchronous with the waveform, then users can analyze directly why the written data is completed different in the same location.



▲ **Fig.12: Find the difference.**

Conclusion

In the past few years, the smart phone, tablet computer and notebook computer have developed very quickly, that requires a large number of traditional multi-chip packaging (MCP) and eMMC. The R&D team has a higher requirement about CPU performance and memory capacity of in-built NAND Flash. Since the first iPhone's hot sell in 2007, more and more mobile phone manufactures began to design the smart phone with in-built memory. The latest iPhone 4 has abandoned the traditional MCP module but adopted the eMMC. So the eMMC is very possible to lead the future trend.

Zeroplus PC-Based Logic Analyzer has supported nearly a hundred Bus modules, its software can auto-decode the captured signal. That helps engineers to quickly analyze and check the signal, optimize the system, quicken the project process and make the product come into the market earlier. They don't need to decode by hand slowly with the oscilloscope. Zeroplus Logic Analyzer is the best tool for you to analyze signal.

Please login our website www.zeroplus.com.tw for more information; or you can register on our website to be our member, we will send technical analysis report to you on irregularly schedule.

Cai Yaowei
Instrument Department/ FAE