Tony

SPI mode in eMMC Devices?

Tony H.
Partner at Benthic Sciences LLP

I'm looking at possibly using embedded flash (ie a flash chip with a built in controller, such as the SanDisk iNAND - not excluding others, SanDisk just comes to mind).

But I'm getting confused. Most talk about supporting the eMMC spec. Some talk about supporting eSD, or SD.

eMMC up to eMMC 4.2 included a SPI-mode to simplify interfacing to low end microcontrollers. SD up to and including SD3.1 still includes SPI-mode.

But in eMMC 4.3 "The chapter defining SPI mode and all SPI-mode references were removed".

Does anyone know if eMMC 4.3 and later devices have really dropped SPI mode, or do we just not talk about it any more?

I've looked at the data sheets (anyone else noticed how little datasheets contain these days?) and asked the manufacturers (not heard yet) but would much prefer "real" experience!

Any comments much appreciated!

赞 • 评论 (28) • 发布 • 2011 年 10 月 26 日

评论

Viktor S.、Simon Z.和其他 2 位会员赞了

28 条评论 • 跳转到最新评论

Tony

Tony H.
Partner at Benthic Sciences LLP

I knew I should have put Android in the title - bet I'd have got lots of responses! :-)

Peter

Peter B.
Director and Senior Consultant at Software Integrity Ltd

@Tony Hedge • I knew I should have put Android in the title - bet I'd have got lots of responses! :-)

But could any of the Androids have answered your questions?

Graham

Graham C.
European Applications Manager at Atmel Technologies UK Ltd

Does it need to be removable? There are quite a number of serial memory devices around with SPI interface, e.g. http://www.spansion.com/Products/Serial-Flash/Pages/Spansion%20FL.aspx?gclid=CJSg8J_BIqwCFQEd4QodWDAHPA

Tony

Tony H.
Partner at Benthic Sciences LLP

@Graham

Thanks. Let me tell you a bit more. What I'm really trying to find is a chip version of something like a MMC or SD card, so that I can put a few Gb of NAND flash on a board instead of relying on a removable card. It's a harsh environment, hence the hesitation about removable, but not so harsh that if costs were to be ridiculous, I wouldn't consider gluing an SD card into its holder (or going for some sort of rugged removable)!

I want to use a low-end 32 bit micro without a dedicated eMMC interface, just a SPI port. Yes, I could create a eMMC style interface with a bit of GPIO but I would like to avoid doing so if I can. And I don't want to be doing ECC and wear levelling myself - I want the memory devices controller to be doing that sort of stuff.

In short, an SD card with its SPI mode meets all my needs except I want it on a chip.

SanDisk's iNAND, when first introduced, certainly met my needs. It implemented the eMMC standard, which provided a SPI mode. But the eMMC spec has moved on and SPI mode has been dropped from later specs. What I can't find out is whether its really been dropped from actual devices - ie have SanDisk removed old functionality or just left it there as the standard has moved on?

Other manufacturers (eg Toshiba with their eMMC NAND) seem similarly vague about SPI. And some that definitely support SPI don't do ECC and wear-levelling - it is essentially dumbNAND rather than iNAND.

ne that all(?) MMC cards provided SPI mode. I'm not sure we can now, though I think we can still expect all SD cards to - SPI mode still seems to be part of the rather secretive SD spec.

Jonny

Jonny D.
Founder, CEO at GridVortex Systems

@Tony
I am doing exactly that: have to use a Nand flash with 2 mcus, one with a SD card interface and another with only SPI interfaces. We are considering the SD-Nand int ted controllers memories for that.
But it seems that the promise of the iNand family did not catch, and it seems that other SD Nand products are hard to find.
I would be glad to share our experiences with you, as I'm curious about yours.

- Jonny

Tony

Tony H.
Partner at Benthic Sciences LLP

@Jonny

Thanks. We haven't got any further with this. I would still like to be able to do it with a chip rather than with a SD card, but as you point out, such devices do seem a bit scarce. I think for now we are going to have to use a SD card wired into its holder. If we ever get any further, I'll let you know.

Thanks again.

Tony

Dario

Dario C.
Senior HW developer engineer

@Tony, I had same problem with MMC card in the past, effectively the newest MMC does't have SPI mode, but for my lucky the SD card today have SPI mode and the SD association seems want keep it in the future.
A note about my direct experience is that standard SD, from any manufacter, works well in SPI mode, but I had some prooblem with some microSD.

Tony

Tony H.
Partner at Benthic Sciences LLP

@Dario

Thanks - that is very useful to know. It is what I suspected but had not been able to confirm.

Tony

Chris

Chris N.
Sr. Field Applications Engineer at SanDisk

@Tony,

I know this is a bit late to the party, but hopefully still useful ...

JEDEC's MMC specs are separated into electrical and mechanical, making MMC and eMMC through v4.3 basically different mechanical implementations of the same electrical spec.

For v4.1 and v4.2 the eMMC spec is a 30 page clarification of the much larger MMC electrical spec. v4.1 is mostly legacy from MMCA before JEDEC took over and covers MMC of 2GB or less capacity. The v4.2 big change added capacities above 2GB.

v4.3 folds the eMMC spec delta doc into the MMC spec creating a unified spec for both, while removing SPI and identifing it as “obsolete”.

v4.4 adds some new features including something for security and support for TRIM.

v4.5 is for eMMC only and lacks any reference to MMC or even the word “card”.

Removal of SPI mode with v4.3 is unfortunate for those using low pin count microcontrollers or those without a hardware SD/MMC host interface. A clear case of the latest not being the greatest.

Regarding your assumption, since SPI has always been an optional mode for MMC it was only availability if the manufacturer included it. Since most designers select MMC over SD for cost reasons (SD Assoc license fee being the only significant delta) and many, many designs have smaller or low pin count microcontrollers lacking a MMC host interface, MMC manufacturers wouldn't be addressing the market if they didn't include SPI. After all, SPI mode support is all in the MMC firmware.

To ensure SPI mode support, request MMC v4.2 max and make sure the datasheet clearly states SPI mode support.

To determine the specification version of an MMC, read the Extended CSD register and examine the Extended CSD Revision (EXT_CSD_REV) field. It should be 0x02 or less for v4.2 or less, respectively.

Regarding device retention, perhaps for a high vibration environment, there are sockets with push-push ejector mechanisms. Of course, that doesn't keep the curious individual from removing it.

Chris


Tony

Tony H.
Partner at Benthic Sciences LLP

@Chris

Many thanks - that is extremely helpful and informative. A similar discussion has since started on another group. Once I can remember which group, I'll post a link back to here.

Thanks again.

Tony


Claudio

Claudio C.
Embedded Systems Team Leader at Aerialtronics

Thanks both Chris and Tony for reporting the information among different groups.
One wonders where such information is tucked away that needs someone looking like a guru or insider to carve it out!


Andy

Andy N.
Owner, Antronics Ltd and Embedded Systems Consultant

Note that there are several "low-end 32 bit micros" which do have dedicated interfaces...


Jonny

Jonny D.
Founder, CEO at GridVortex Systems

@Andy: Yes, some "low-end" 32bit micros really have full-blown SD card interfaces, with support for 25MHz to 50MHz data clock. Those parts would really benefit from SD-NAND.
If they do not disappear, that is.

- Jonny


Claudio

Claudio C.
Embedded Systems Team Leader at Aerialtronics

Could anyone elaborate a bit on which parts do actually include this interface?
And on the opposite which do incorporate the necessary peripherals to access nand-flash?
For example a cheap chinese pmp ripoff what is based on? Those 10$ things with 4gigs of memory can't have a Tegra2 inside.

For the sake of simplicity I'm looking for the most cost-effective and least powerfull processors, where the objective is to evaluate migration of embedded designs based off low-cost 32bit MCUs.


Colin

Colin B.
Firmware Engineer III at Harman International Industries

@Claudio Carbone: The STM32F103RET6 is an ARM Cortex-M3 microcontroller with both SDIO and external flash memory interfaces. It costs less than $9.00 in single unit quantity from Mouser and probably half that in volume. Avnet has it for $5.91 (single unit qty). Depends on your product, but I consider this a very reasonable price for the features and performance in this microcontroller.


Jonny

Jonny D.
Founder, CEO at GridVortex Systems

@Claudio Carbone:
You can have it in some ARM7 uControllers like the LPC2368, and also in some Cortex-M3, like the LPC1785.

- Jonny


Claudio

Claudio C.
Embedded Systems Team Leader at Aerialtronics

Thank you both Coling and Jonny, valuable resources.
I have at least another question though:
when I went reading about the LPC MCUs, some ar labeled "external memory controller" and some "SD/MMC card interface".

Now in my ignorance I could understand that the "external memory controller" is supposed to allow use of external ram modules to enhance the quantity of memory available to the application.

BUT being nand a kind of external memory too, it could also mean support for external ROM.

Also when I read "SD/MMC card interface" I take it to mean support for cards in the physical sense: removable SD/MMC cards.
That I think is different from supporting NAND chips?

It's confusing me, but that's just because I don't know enough about this.

I also just read about SPI-NOR in a different sentence than SLC-NAND:
http://www.embeddedartists.com/products/kits/lpc3250_kit.php
which seems to imply that access to these different kind of memories, require different peripherals on the MCU?

Ok so now the question may become: where can I get a basic level comprehensive informative documentation on these different kinds of memories and their requirements?


Colin

Colin B.
Firmware Engineer III at Harman International Industries

@Claudio Carbone:
The STM32 external memory interface, called Flexible Static Memory Controller (FSMC), supports many different types of memory: PC Card/Compact Flash, SRAM, PSRAM, NOR and NAND. The FSMC can also be used to efficiently interface other peripherals such as LCDs.

The NXP LPC1785 supports RAM, ROM, and flash memory, but has support for dynamic RAM (SDRAM) as well.

The SD/MMC card interface is *totally* different because the SD/MMC interface is an application-specific protocol between your host microcontroller and the SD flash controller chip inside the card. This SD flash controller handles all the specifics of NAND flash access, etc.


Jonny

Jonny D.
Founder, CEO at GridVortex Systems

@Claudio Carbone:

In LPCxxxx cpus, as @Colin said, you have a standard parallel external memory bus, but each chip has specific EMC (external memory controller) features.

The LPC23xx has a static memory small EMC bus, that is a ARM PrimeCell peripheral (meaning it's a ARM provided RTL), that is a AMBA parallel memory controller for static RAM, ROM and parallel FLASH. This memory controller (for the LPC23xx) is a simple 8-bit bus, 128KB maximum size, with 2 separate chip selects.

In the LPC24xx, the EMC bus is a full 32-bit wide, 16MB static and dynamic memory controller, that has up to 8 chip selects, and can support static memory like parallel Flash and static RAM, but also single data rate SDRAM.

These memory controllers are for very high-speed parallel memory, and do not control NAND Flash. The memories in the EMC bus can have up to 144MBytes/s (2 wait states, 32bits) of continuous data transfer speed.

Some selected cpus, like the LPC2368, LPC24xx, LPC18xx, and the new LPC43xx, also have SD-Card hardware controllers. These are full SD-Card interfaces, with 4 high-speed serial data lines. The SD-Card hardware interface controller can handle up to 50MHz on the serial bus, and can achieve up to 200Mbps, or 25MBytes/s of short burst transfer speeds, if the cards support SDHC with a high speed class.
The SD-card controllers might be used to control high-speed NAND Flash, if you use a SD-Nand memory, which is a NAND Flash with a SD-Card-type integrated controller. These memories are seen by the SD controller as if they are a SD card, and the built-in controller will take care of all NAND Flash controller needs.

The LPC18xx and LPC43xx have better EMC controllers, that can handle up to 200MHz bus

speeds (800MBytes/s transfer speeds), with bus buffers and low power DRAM operation.

Yet another memory controller is the SPI Flash Interface, that maps Quad serial NOR Flash into the internal memory bus, allowing you to DMA from/to an external QuadSPI Flash memory, and allow code execution directly from serial flash (XiP - Execute In Place). This bus can handle up to 120MHz SPI clocks, for up to theoretical 480Mbps, or about 15MIPS to 30MIPS of execution performance in XiP mode.

There is no NAND flash memory controller in these processors.

- Jonny

Chris

Chris N.
Sr. Field Applications Engineer at SanDisk

@Claudio,

Regarding discrete flash memory devices, that are soldered to the board in a "memory down" design, I found this paper from Micron informative:

Micron Technical Note NAND Flash 101:
An Introduction to NAND Flash and How to Design It In to Your Next Product
http://www.micron.com/~/media/Documents/Products/Technical%20Note/NAND%20Flash/145tn2919_nand_101.ashx
* Gives a very good basic description of SLC NAND with comparison to NOR, plus some info on MLC NAND

* Note the increased pin count required for NOR

* This doc is 6 years old (though it has some updates in 2010) so the details in Table 11: MLC vs. SLC are valid for NAND with geometries above 60nm which are going EOL but out of date for current MLC/SLC generation, specifically Endurance. While density is going up and cost per bit comes down, endurance is decreasing and significantly more intense ECC is required. Specifically at sub-50nm MLC is closer to 3k write/erase cycles and SLC is closer to 30k cycles. As the technology gets smaller, the ECC requirements increase and the write/erase cycles decreases. Note how MLC needs many more bits ECC than SLC.

Why is this important, especially for a microcontroller design? This added ECC burden is a CPU timeline leech because ECC needs to be calculated by your microcontroller for every read and write operation. And the micro needs to manage bad blocks. And it needs to perform wear leveling. File systems like JFFS2 and UBIFS do this, if you are running an OS that has them. This is not as much a concern for NOR, not nearly.

You can see why an SD card (or USB or MO-297A) has an increasing advantage as NAND geometries shrink, even for faster micros. While it has the very same NAND die discussed above, it also has a controller of its own to manage the NAND and a relatively low pin count interface, very low with SPI. The SD controller can be an 8-bit or 32-bit controller, it has its own firmware to manage the interface protocol and offloads all the NAND management operations from the micro. It does all the wear leveling and bad block management. It also typically has dedicated ECC hardware to accelerate these operations. It will also have read and write buffers to isolate the SD interface from NAND operations as much as possible, but even so it can put up a BUSY signal. I wonder what percentage of designers realize their PIC or 8051 has an SD card with a 32-bit controller.

The Micron Tech Note mentioned above closely matches a 2006 EETimes article by Jim Cooke of Micron. Google "NAND Flash 101" to find it. Mr. Cooke also wrote a follow up entitled "NAND 201: the continued evolution of NAND Flash" early last year. To get this you will have to register an account with EETimes. Excellent articles. The 201 article mentions 8LC NAND indicating 8 distinct charge levels on the NAND cell to identify 3-bits of data. Some vendors refer to this as TLC. MLC has four charge levels for 2-bits of data. You can imagine that TLC will have enhanced reliability and endurance issues. And we all know what happens when the heat gets turned up in an industrial application.

The 4MB SPI-NOR you mention sounds like the same type of memory that server motherboards use for the BIOS image. Much smaller and since data centers never turn off servers unless they must, it doesn't get used too much. Boot time is not an big issue here.

Claudio

Claudio C.
Embedded Systems Team Leader at Aerialtronics

Thank you Jonny, Colin and Chris.
I hope Tony, the OP, isn't getting too upset about me hijacking his thread.
This has been very useful!

@Chris
from your very exaustive info, it seems I'm missing something.
If SLC and MLC are more or less in the thousands cycles, I don't get how can we have SSDs, PMPs, smartphones and all the rest: if my smartphone really had 3000 cycles, I may well be at half its life in less than a year!
Not to mention SSDs...
Sure this gets into the realm of statisticians (which is far from my realm) and their wear leveling algos, but 3000 cycles seems awfully few.
But this is all very OT (I'm just endlessly curious).

Back on topic: how do chinese get away with their PMP knockoffs? I doubt a 20$ gadget has a full operating system managing a complete file system... if so, how could they offer 4GB or more if the only big enough chips are nand?
http://www.ebay.it/itm/4-GB-1-8-MP4-media-player-with-recording-game-radio-calendar-/230765263575?pt=LH_DefaultDomain_0&hash=item35baae7ad7
at 13$ including S&H this can't possibly have a full nand, could it?
So what is the solution to get 4GB without a discrete SD and without managing the NAND?

It seems there are other options including some kind of controller of which we haven't spoken yet. Or maybe it's just me not getting it right.


Chris

Chris N.
Sr. Field Applications Engineer at SanDisk

@Claudio,

Many devices don't write to the actual NAND as much as you might think. Temp data is kept in DRAM. Embedded Linux systems use a RAMdisk for temp file to avoid hitting the flash more than necessary, or they should. SSDs as small as 32GB can have a DRAM cache on the controller to collect writes before they hit the NAND. The faster ones definitely do.

Media players and your smartphone are read mostly applications from the microSD point of view. And how often does your phone or something like an iTouch die before the user wants to upgrade to the latest generation? (I've been fighting the call of the Galaxy S II myself and the Galaxy S I have is just fine.)

Consider a very simplified look at this, how often do you write the full capacity of the microSD in you phone? Probably not even once per day. (per week?) If you did this 10 times per day which would logically require an erase of the full capacity 10 times per day, your device with MLC NAND should last about 3000/10 or 300 days. Seems extreme so MLC is fine for this kind of application. (Different story for the storage of the boot ROM code. Not sure what most cell phones are using for this now, but I'd hope SLC, probably serial.)

Regarding the player you referenced, I would think the media player chip is the NAND controller, providing a two chip solution with a 32Gbit NAND chip. And the media player chip likely an integrated boot ROM that allows further booting from the NAND.


Claudio

Claudio C.
Embedded Systems Team Leader at Aerialtronics

Thank you again Chris.
Coming from a very low-level segment of the embedded ecosphere, this was all outside my reach.
I could start experimenting with an LPC but all the basic dev-boards don't have this capability. I could try with one of these
http://www.embeddedartists.com/products/oem/lpc2478_oem.php
but it seems a bit extreme to get a start on the tech. Also expansion would be difficult with a SODIMM like connector.

The LPCXpresso would be the right choice, but they got nothing beyond the 1769.


George

George I.
General Partner at New Technology Embedded GP

Is there a Cortex A9 SoC that can be plugged into the microSD Slot?


Juergen

Juergen P.
Sales and Marketing Manager at ExMark

Coming, from a different angle, have you lookd at the Hyperstone FLASH controller micros?
http://www.hyperstone.com/en/Solutions-Solid-State-Disk-Hyperstone-USA-167,2749.html


Claudio

Claudio C.
Embedded Systems Team Leader at Aerialtronics

@Juergen
an SSD controller is entirely out of place here.
Eventually the Hyperstone S6 could be a candidate, surely the A2 is not.


Claudio

Claudio P.
Senior HW/SW Engineer

I'm glad to see I'm not the only one having trouble interfacing a large memory with a simple SPI. I also am in a very harsh environment and I can't trust the SD mechanical solution.
Tony, have you found a solution you may suggest?
Thank you
Claudio

Paul

Paul M.
Embedded Soft & Hardware Engineer

Hi

If you goes for a SD or micro SD card provide then the possibility to switch the power off on, like above is already explain, some card like the one of sandisk have such problems but by switch them off and on, initialize it again to switch back to data-transfer mode then they works in SPI mode very well. They have also a internal fat controller who replace damage records it make it in any way more safe.