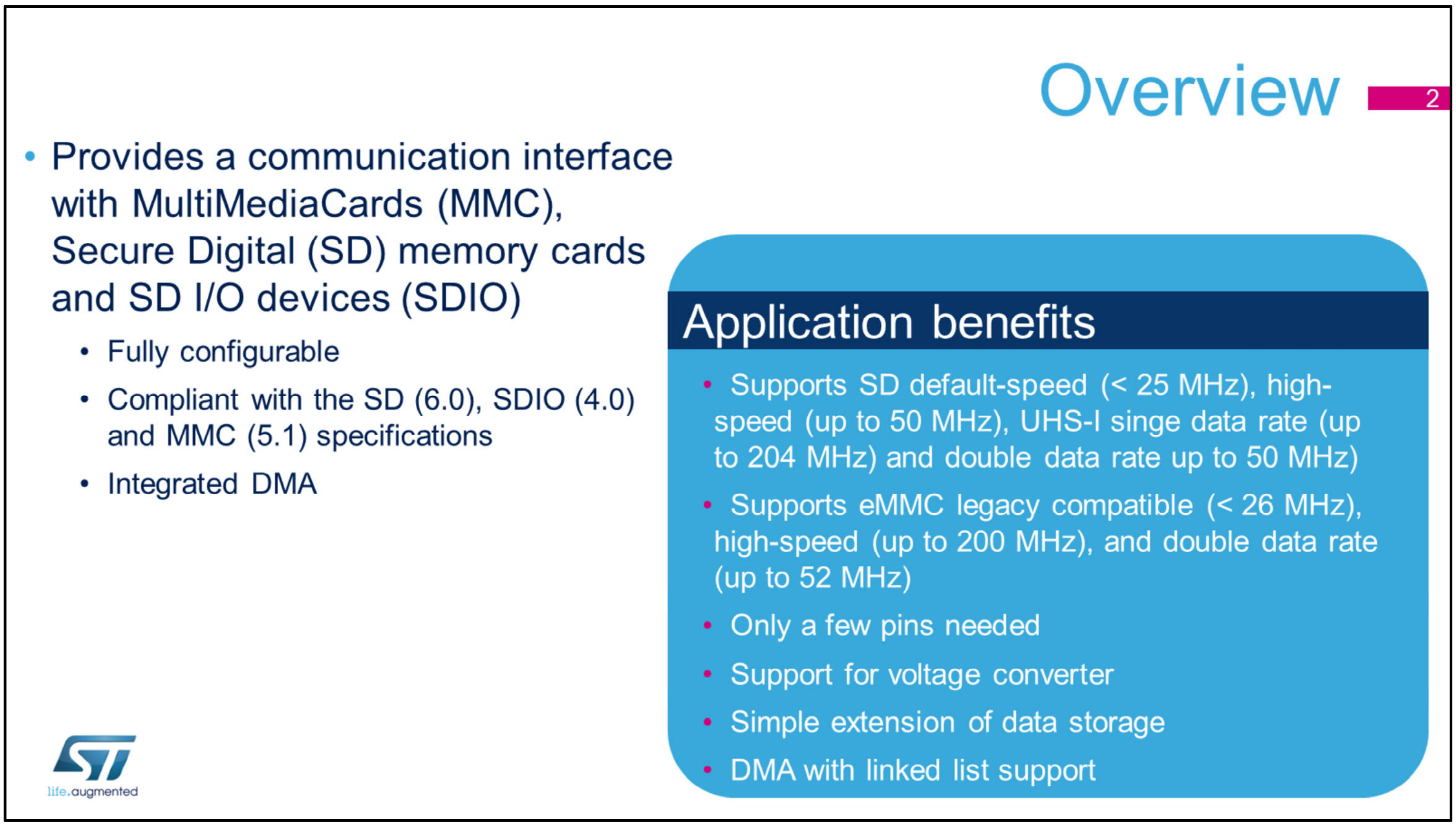
STM32MP1 - SDMMC

SD/SDIO/MMC host interface

Revision 1.0

life.augmented

Hello, and welcome to this presentation of the STM32 SDMMC controller module. It covers the main features of the controller which is used to connect the MPU to an SD card, MMC card, or an SDIO device.



Overview

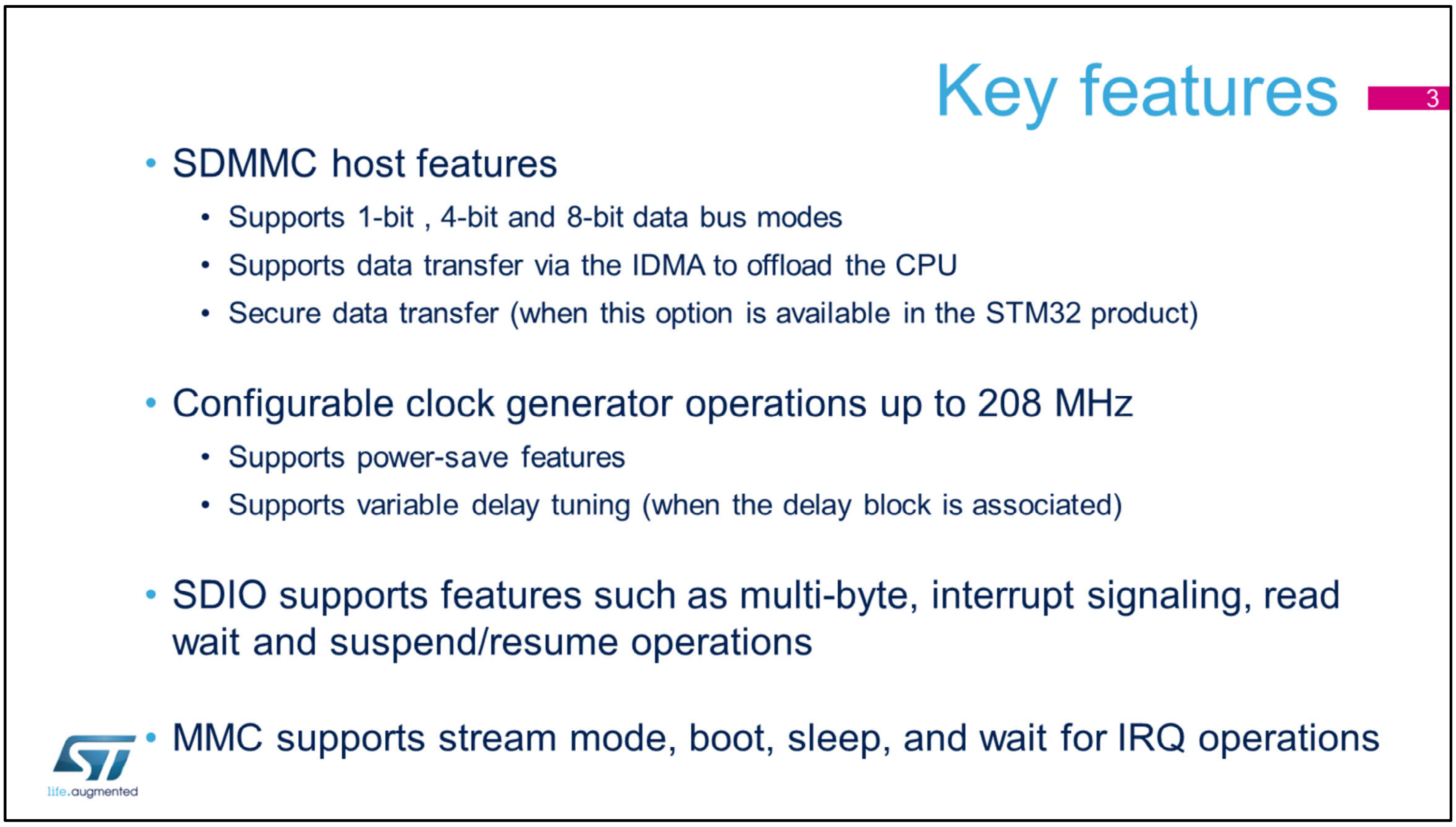
- Provides a communication interface with MultiMediaCards (MMC), Secure Digital (SD) memory cards and SD I/O devices (SDIO)
  - Fully configurable
  - Compliant with the SD (6.0), SDIO (4.0) and MMC (5.1) specifications
  - Integrated DMA

Application benefits

- Supports SD default-speed (< 25 MHz), high-speed (up to 50 MHz), UHS-I singe data rate (up to 204 MHz) and double data rate up to 50 MHz)
- Supports eMMC legacy compatible (< 26 MHz), high-speed (up to 200 MHz), and double data rate (up to 52 MHz)
- Only a few pins needed
- Support for voltage converter
- Simple extension of data storage
- DMA with linked list support

life.augmented

The SDMMC controller provides a communication interface for the microcontroller to communicate with Multi-Media-Cards, SD memory cards and SDIO devices. This interface is fully configurable, allowing the easy connection of external memories thereby extending mass storage capability when more memory is needed. Applications benefit from the reduced pin count required to interface with memory cards. With the SDMMC interface, applications can easily manage high-speed read and write operations in external Flash memories.

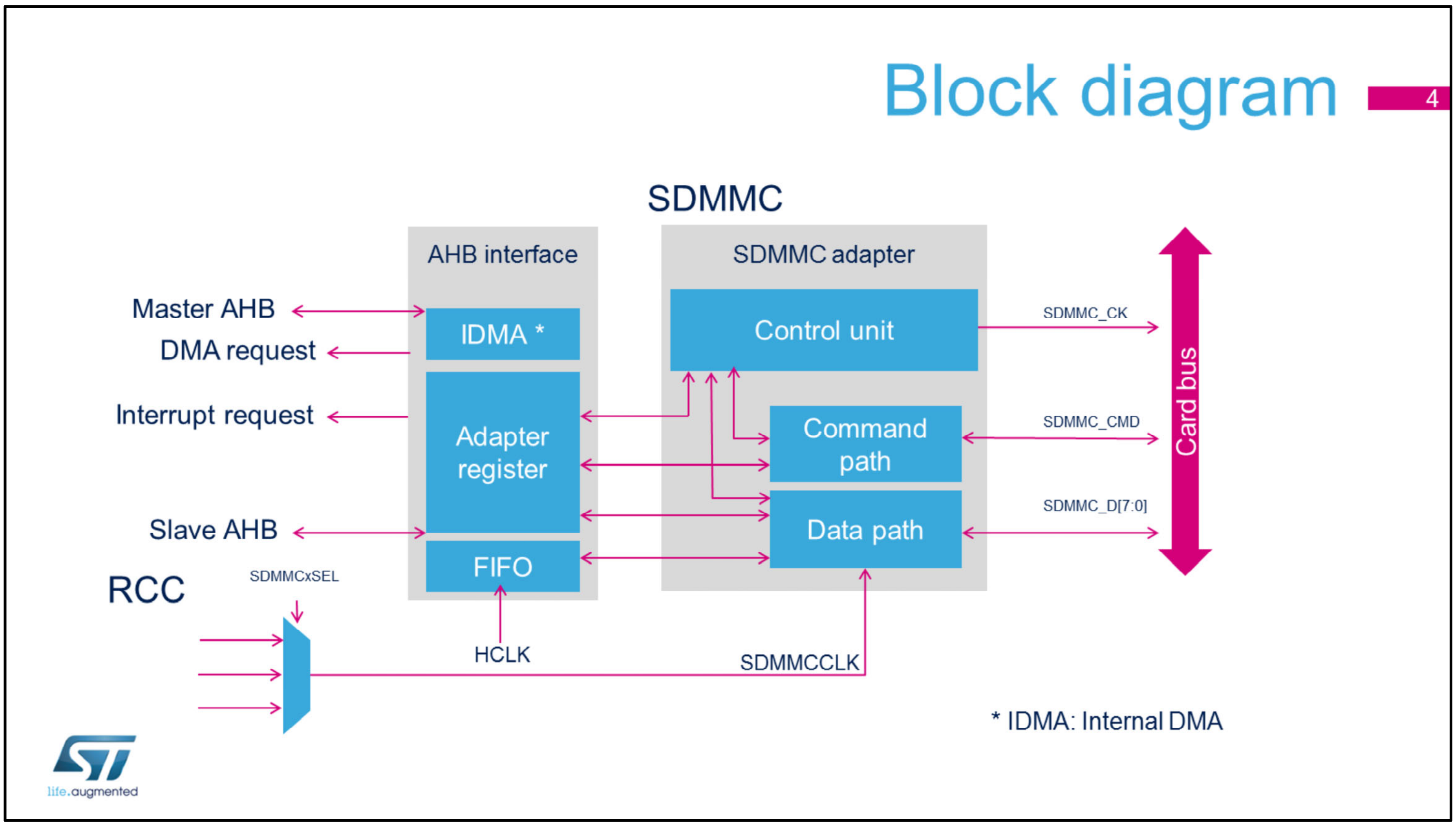


Key features

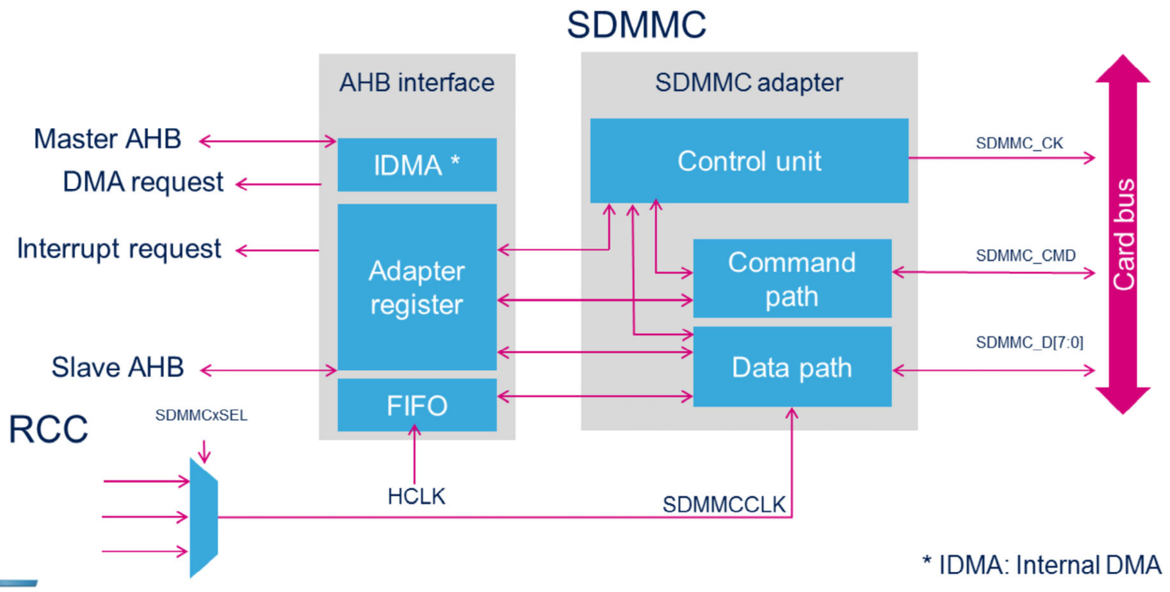
3

- SDMMC host features
  - Supports 1-bit , 4-bit and 8-bit data bus modes
  - Supports data transfer via the IDMA to offload the CPU
  - Secure data transfer (when this option is available in the STM32 product)
- Configurable clock generator operations up to 208 MHz
  - Supports power-save features
  - Supports variable delay tuning (when the delay block is associated)
- SDIO supports features such as multi-byte, interrupt signaling, read wait and suspend/resume operations
- MMC supports stream mode, boot, sleep, and wait for IRQ operations

The SDMMC controller supports data bus widths of 1-bit mode (default), 4-bit mode and 8-bit mode for enhanced data throughput. The SDMMC interface interconnects with the internal DMA (IDMA) to offload the CPU during data read or write transfer periods. The SDMMC clock generator can generate signals up to 400 kHz for the initialization phase and up to 204 MHz for cards supporting High-speed mode.
To enhance power consumption, the SDMMC clock can be disabled when the command and data buses are idle.
The controller can interface with SD I/O modules, with advanced features like read wait, suspend/resume operations, and standard operations like multi-byte transfer and interrupt signaling in 1- and 4-bit modes.
In MMC mode, it supports stream mode, boot and sleep operations.

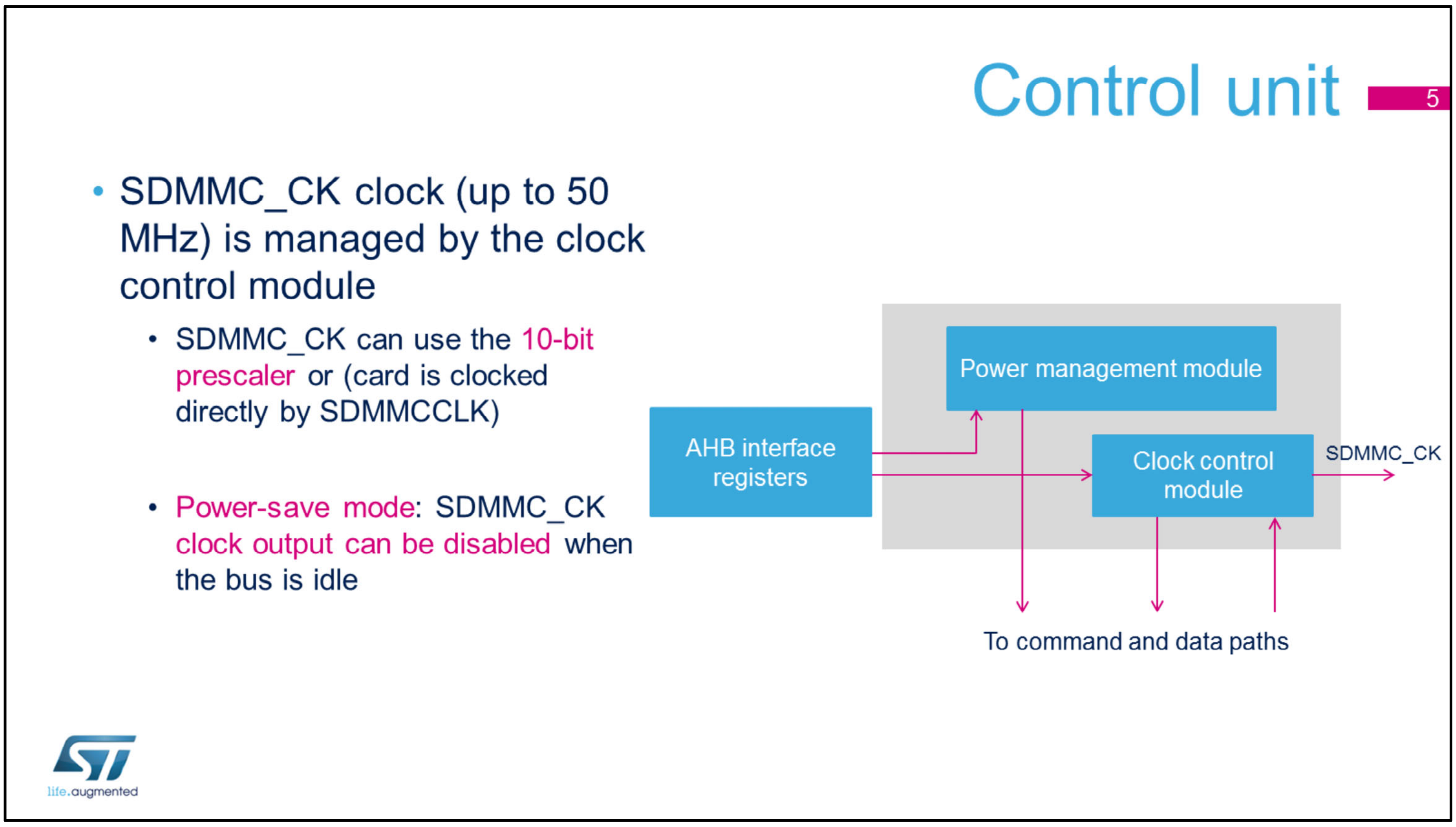


Block diagram



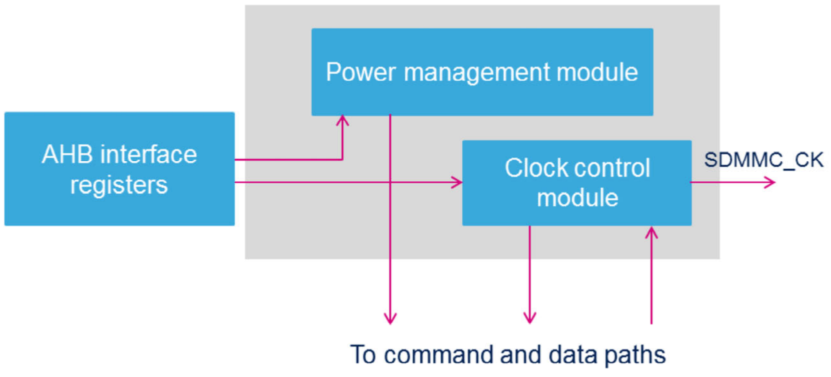
* IDMA: Internal DMA

The SDMMC controller is an SD/MMC bus master that provides all SD/SDIO and MMC functions needed to interface with cards.
It consists of an "SDMMC Adapter" and an "AHB interface".
The “SDMMC adapter” provides functions such as clock generation, command and data transfer, while the “AHB interface” manages the control and status registers, FIFO buffers as well as IDMA with linked list support and interrupt requests.
Two clocks are available for the SDMMC controller, the AHB clock (HCLK) for the “AHB interface” and the SDMMC clock (SDMMCCLK) for the “SDMMC adapter”.



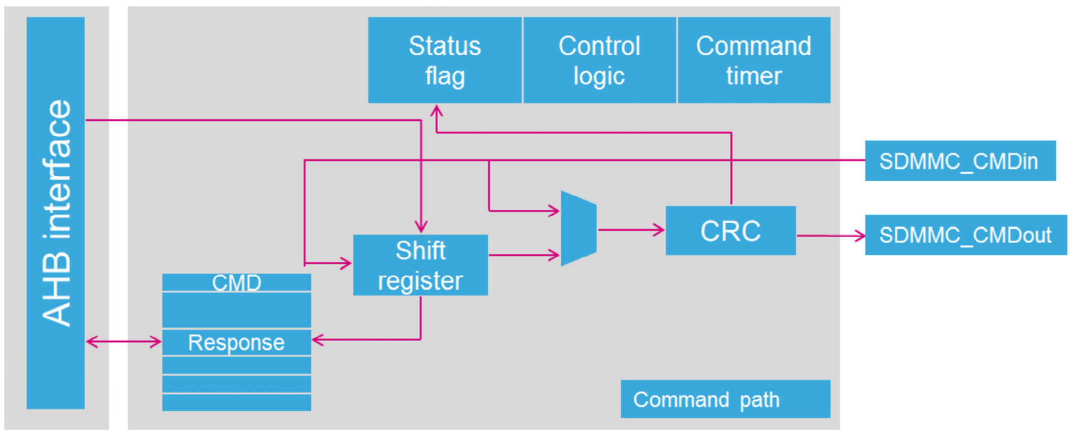
Control unit

- SDMMC_CK clock (up to 50 MHz) is managed by the clock control module
  - SDMMC_CK can use the 10-bit prescaler or (card is clocked directly by SDMMCCLK)
  - Power-save mode: SDMMC_CK clock output can be disabled when the bus is idle

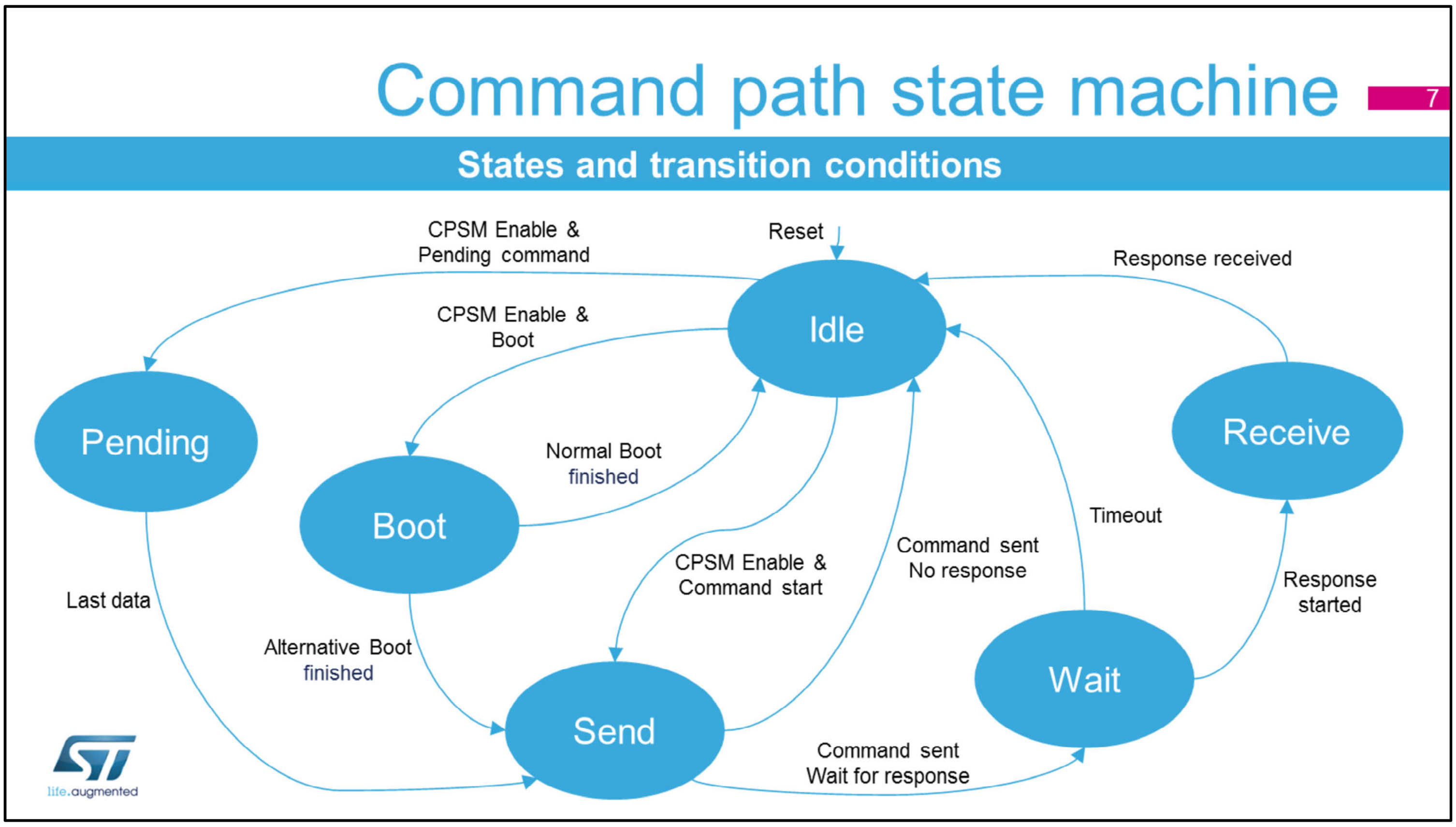


The SDMMC adapter includes a control unit that contains a power management module and a clock control module with the clock divider for the card clock (SDMMC_CK).
The clock control module provides a 10-bit prescaler for SDMMC_CK clock generation, which allows it to generate a clock ranging from SDMMCCLK down to SDMMCCLK/2046.
DDR memory devices are not supported in Divide by 1 mode.
The control unit can disable SDMMC_CK generation when the bus is idle.

Command path

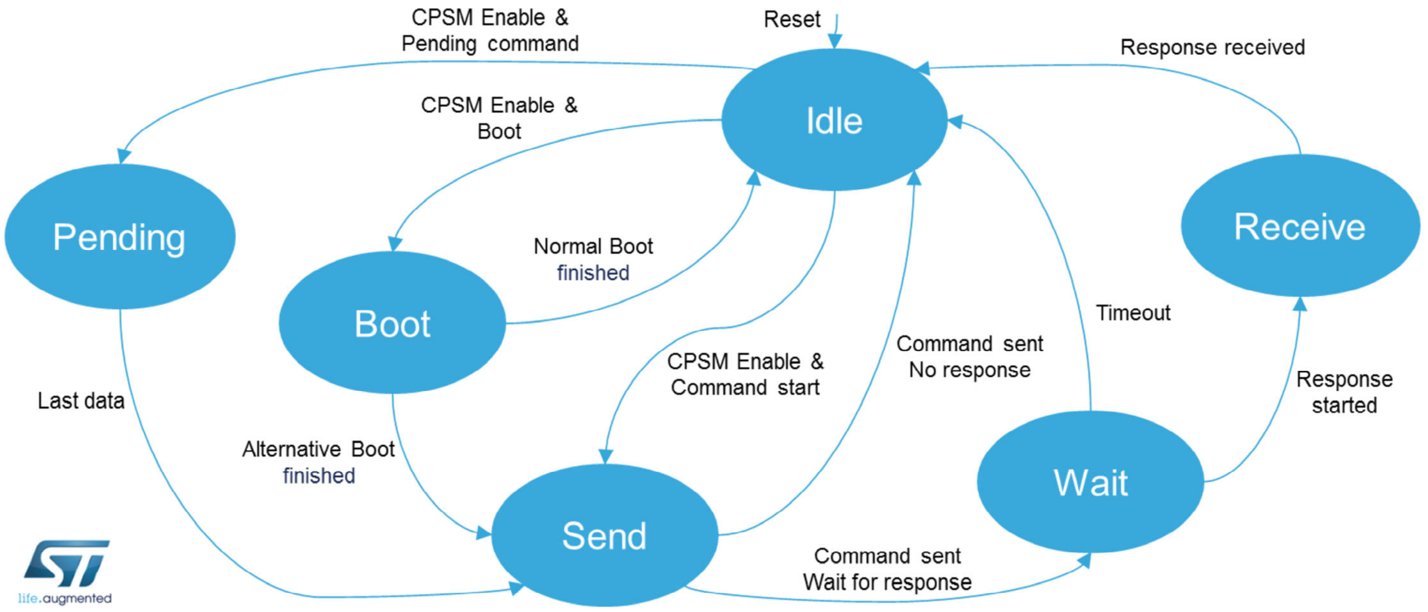


The command path circuit is used to program a command/response sequence.
When enabled, the command path shifts out the command index and argument on the SDMMC_CMD pin. After the last payload bit is sent, a CRC7 is computed and sent on the bus before generating the end bit.
SDMMC_CMDin and SDMMC_CMDout are two modes indicating how the SDMMC_CMD pin is working.
When a response is expected, the command path is configured to SDMMC_CMDin and waits for the device response.



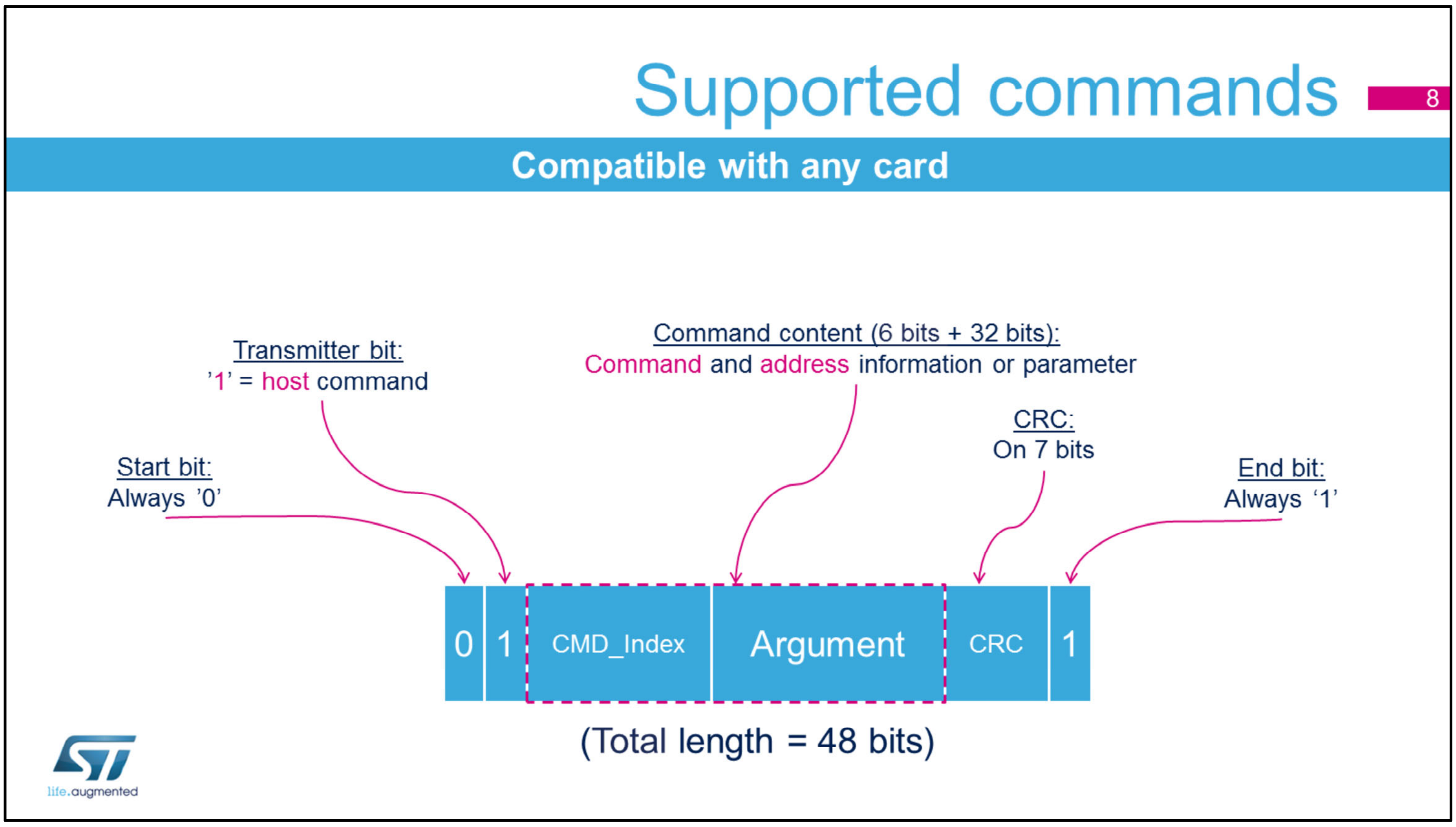
Command path state machine

States and transition conditions



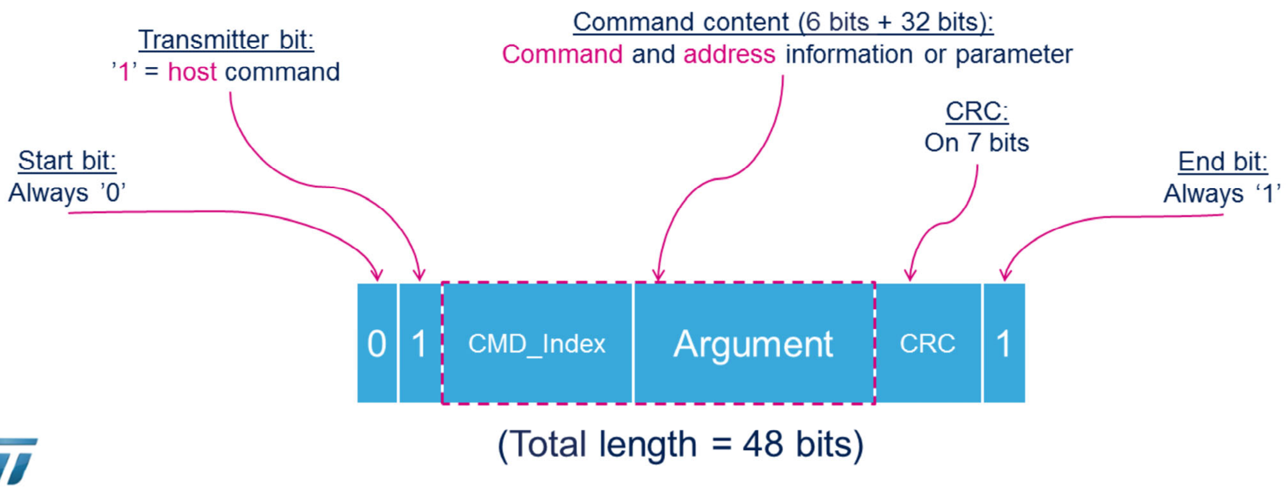
The transmission and reception of commands is controlled by the command path state machine (CPSM). When no command or response is in progress, the command path is in Idle state.
When the CPSM is enabled to send a command, the command path moves to Send state until the last bit of the command is sent, then depending on whether a response is expected or not, the CPSM can return to Idle state when no response is expected or move to Wait state, and wait for a start bit on a command pin (start of the response transmission).
When a response start bit is detected within the allocated time period, the CPSM moves to Receive state. After receiving the last bit of the response, the CPSM verifies the response's integrity using the received CRC, and then returns to Idle state.
The CPSM returns to Idle state after a timeout if a

response start is not detected.
The CPSM can be configured to send a command synchronized with the end of data transfer. When this feature is enabled, the CPSM moves to Pending state and waits for the end of the MMC stream transfer. When the last data signal is triggered by the data path, the CPSM moves to Send state.
The CPSM can be configured to initiate the Boot Normal or Alternative Boot procedure. To trigger this boot phase, the CPSM moves to Boot state. When all data in the Normal Boot has been received, the CPSM terminates the Boot phase and returns to the Idle state. When all data in the Alternative Boot mode has been received, the CPSM moves to the Send state to terminate the Boot phase by sending the CMD0 reset command.

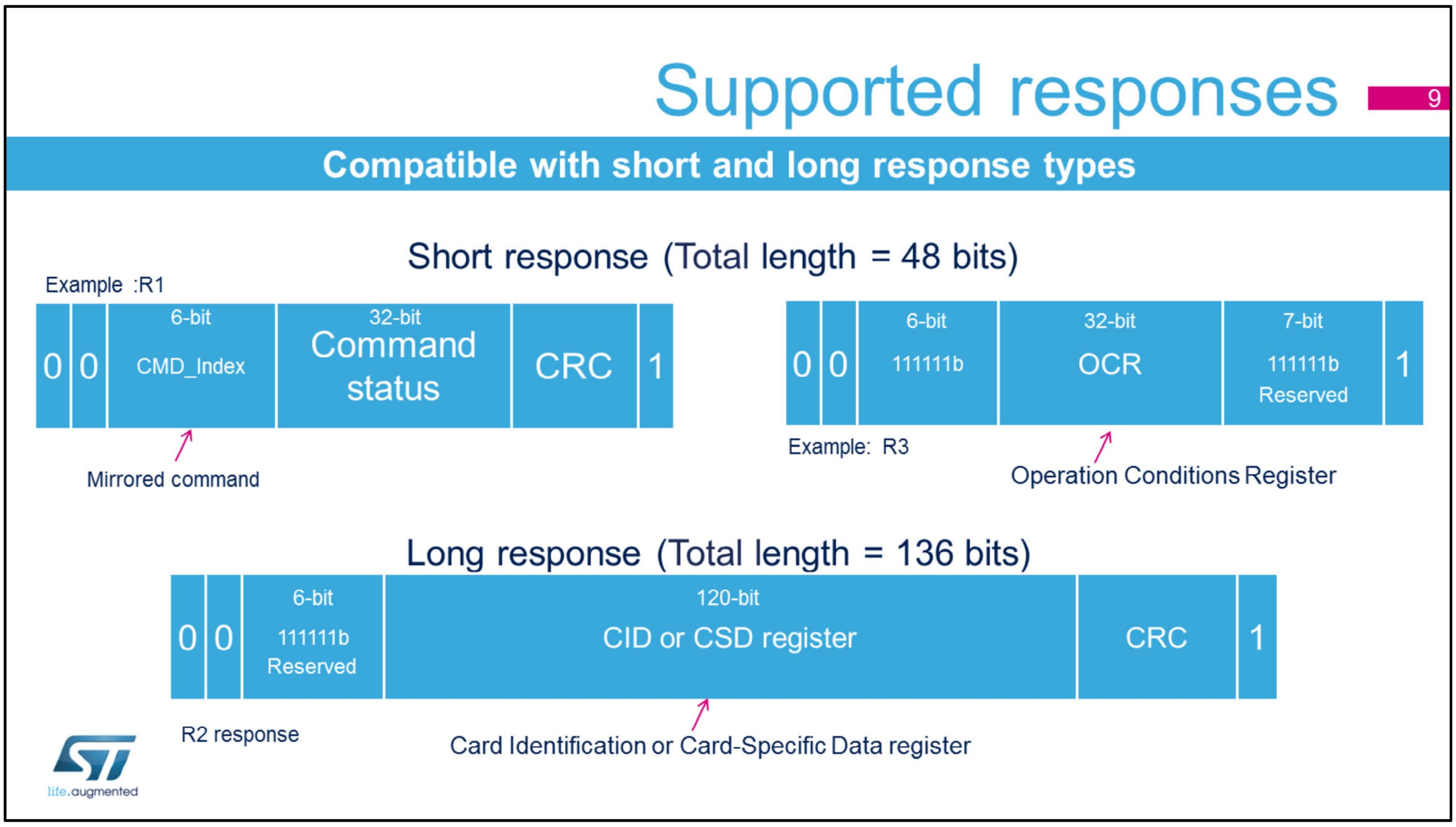


Supported commands

Compatible with any card



The SDMMC controller offers high flexibility for configuring the command indexes and arguments. With a flexible 32-bit register for configuring arguments and an independent 6-bit field for the command index, this architecture ensures that the firmware can address any type of card.
The command path state machine is able to generate all command tokens, with no restrictions on command index or argument. In addition, the start bit, transmitter bit, CRC and end bit fields are automatically generated and sent on the bus.



Supported responses

Compatible with short and long response types

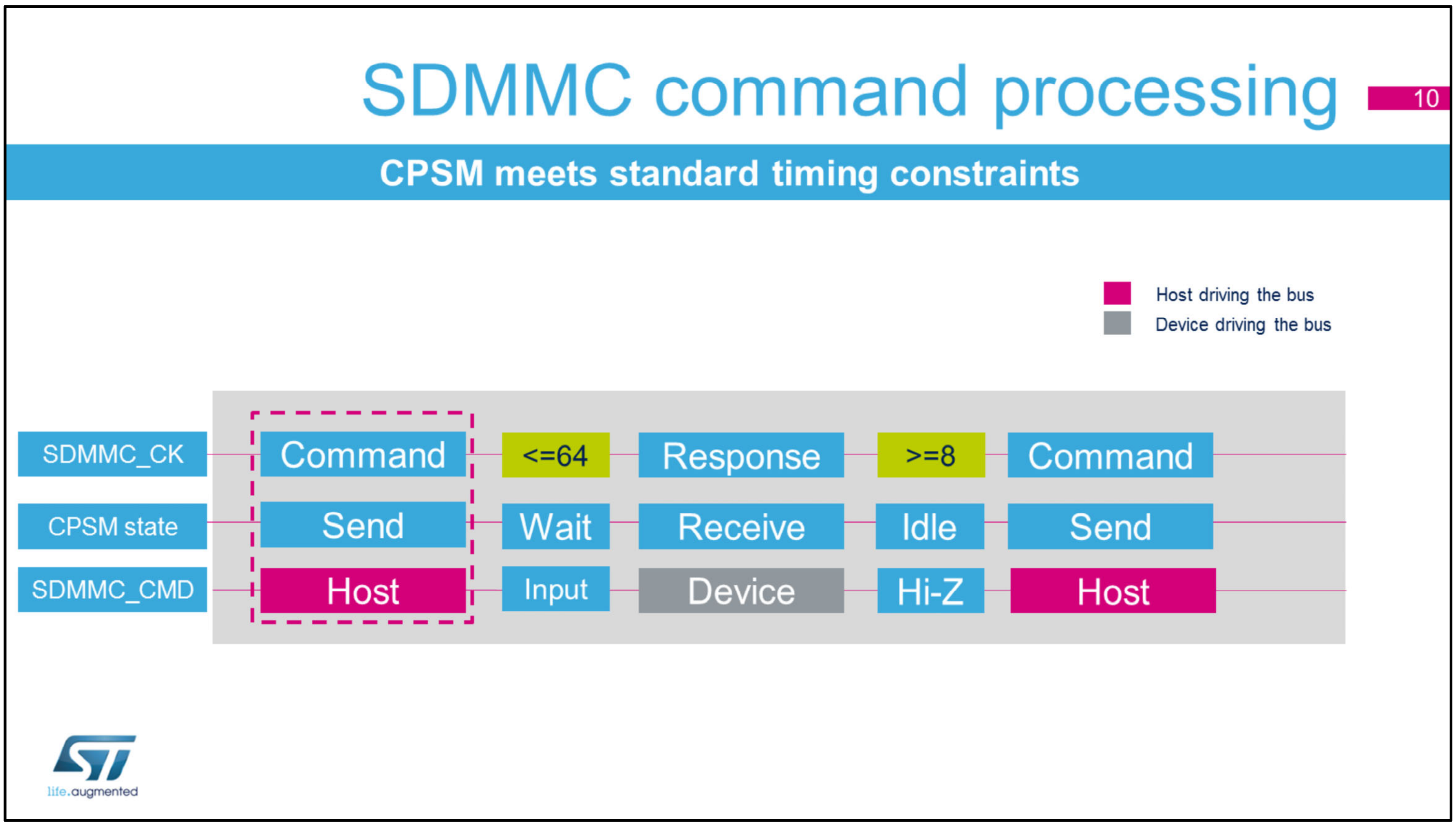
A response is a token that is sent from the card as an answer to the previous command. There are 2 types of responses: short and long.

With four 32-bit response registers and no response constraints, the SDMMC interface supports both long and short responses to correctly initialize the card and communicate with it.

Short responses have a total length of 48 bits, and are composed of a mirrored command index, 32-bit command status, start bit, stop bit and CRC7 checksum. When a short response is received, the command status is saved in the SDMMC_RESP1 register, and the mirrored command index, when available, is copied to the SDMMC_RESPCMD register.

Long responses have a total length of 136 bits, and are composed of the 120-bit CID/CSD register content with the start bit, stop bit and CRC7 checksum. When

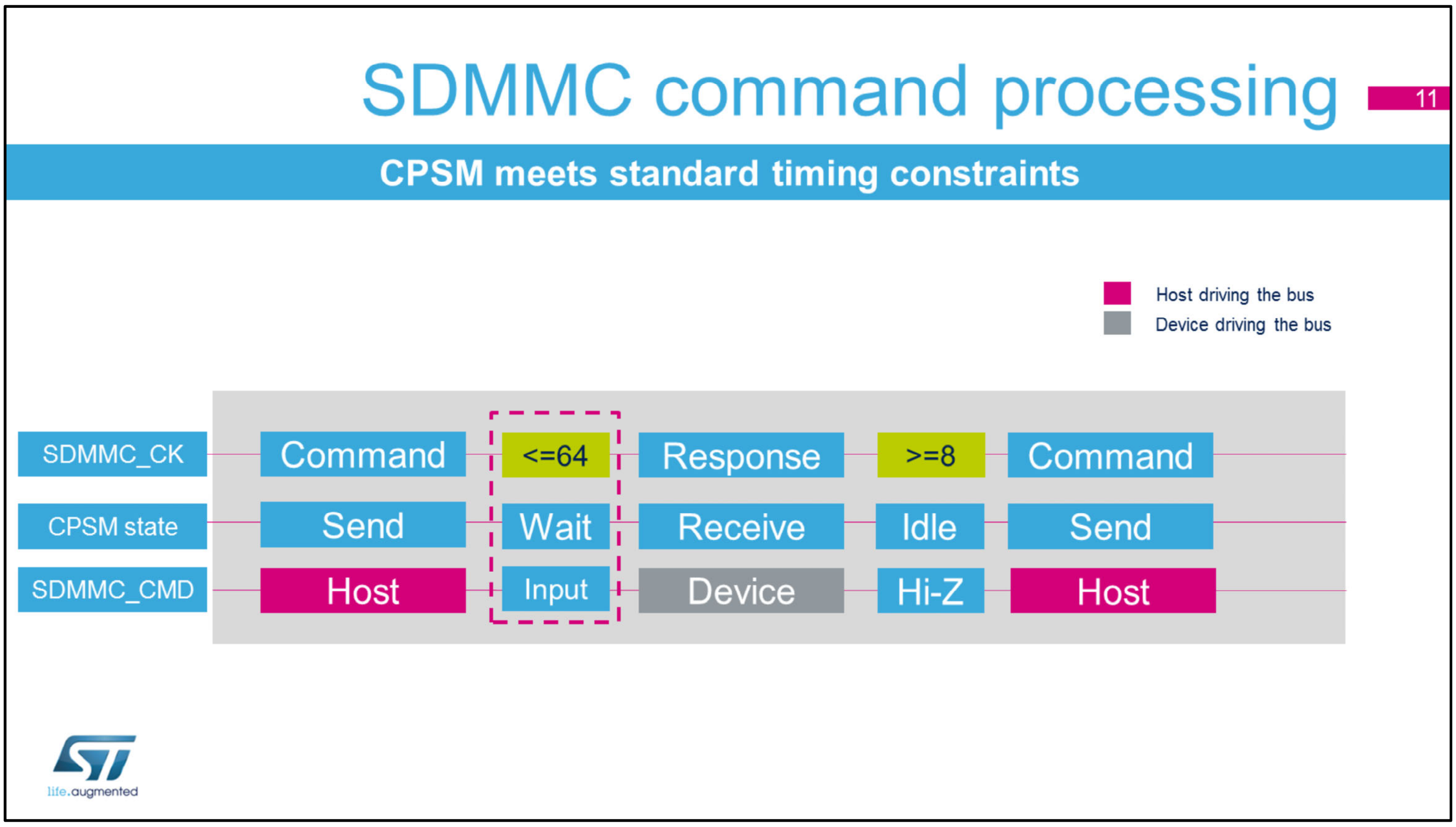
received, the CID/CSD card register is copied to one of the four SDMMC_RESPx registers.
The SDMMC interface also features the automatic detection of a start bit, command index extraction, 32- or 128-bit response extraction and automatic CRC7 verification.



SDMMC command processing

CPSM meets standard timing constraints

Once the SDMMC_ARG and SDMMC_CMD registers are programmed with CMDINDEX, WAITRESP=‘01’ or ‘11’ and CPSMEN =‘1’, the CPSM moves from Idle to Send state and the host starts driving the SDMMC_CMD line to send the command to the card.



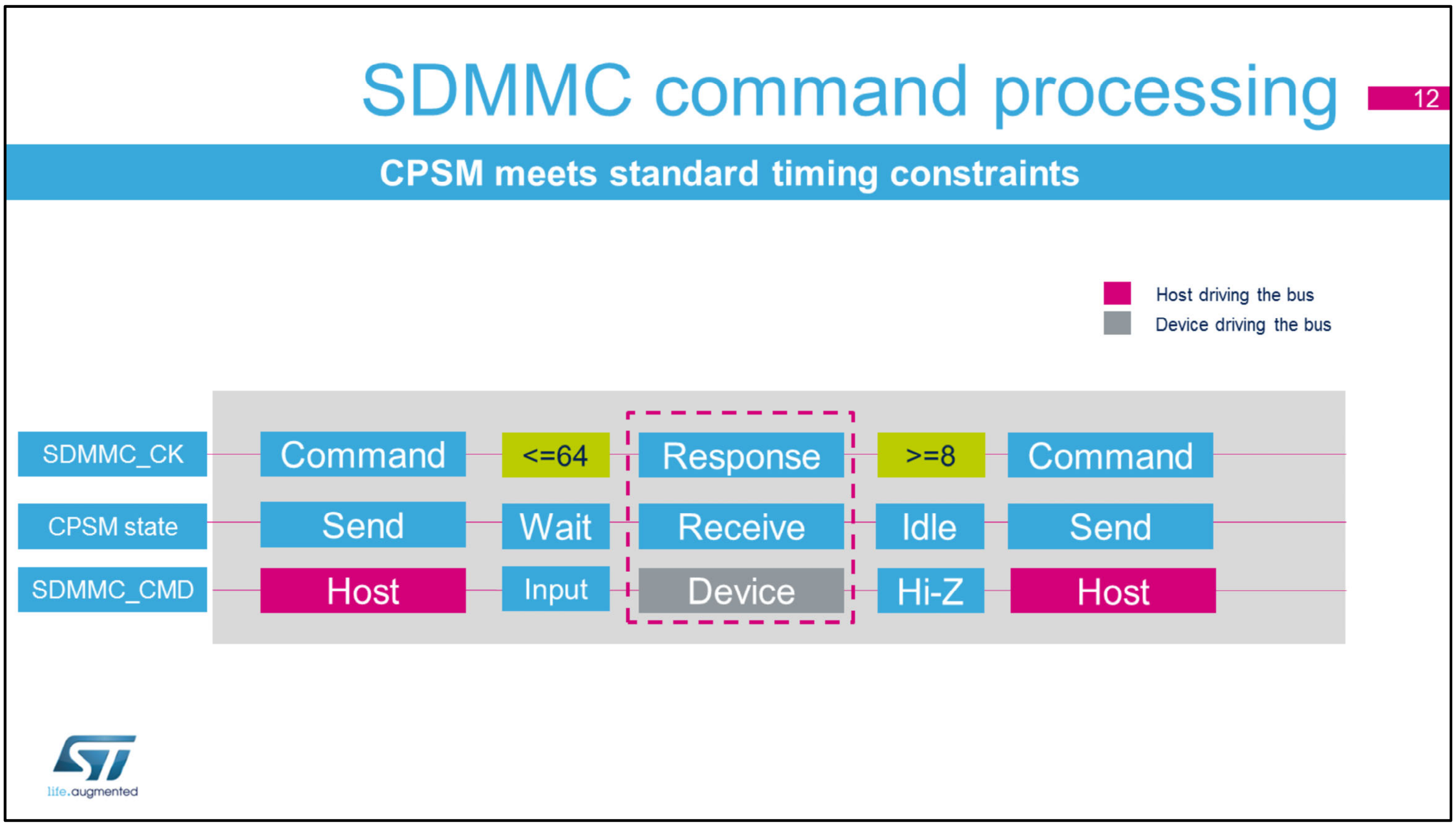
SDMMC command processing

CPSM meets standard timing constraints

Host driving the bus
Device driving the bus
SDMMC_CK
CPSM state
SDMMC_CMD
Command
<=64
Response
>=8
Command
Send
Wait
Receive
Idle
Send
Host
Input
Device
Hi-Z
Host
life.augmented

If the CPSM is programmed to wait for a response (WAITRESP=‘01’, ‘10’ or ‘11’), it enters the Wait state and the command timer starts running. If the card doesn’t respond within the maximum NCR time, the timeout flag is set and the CPSM returns to Idle state.

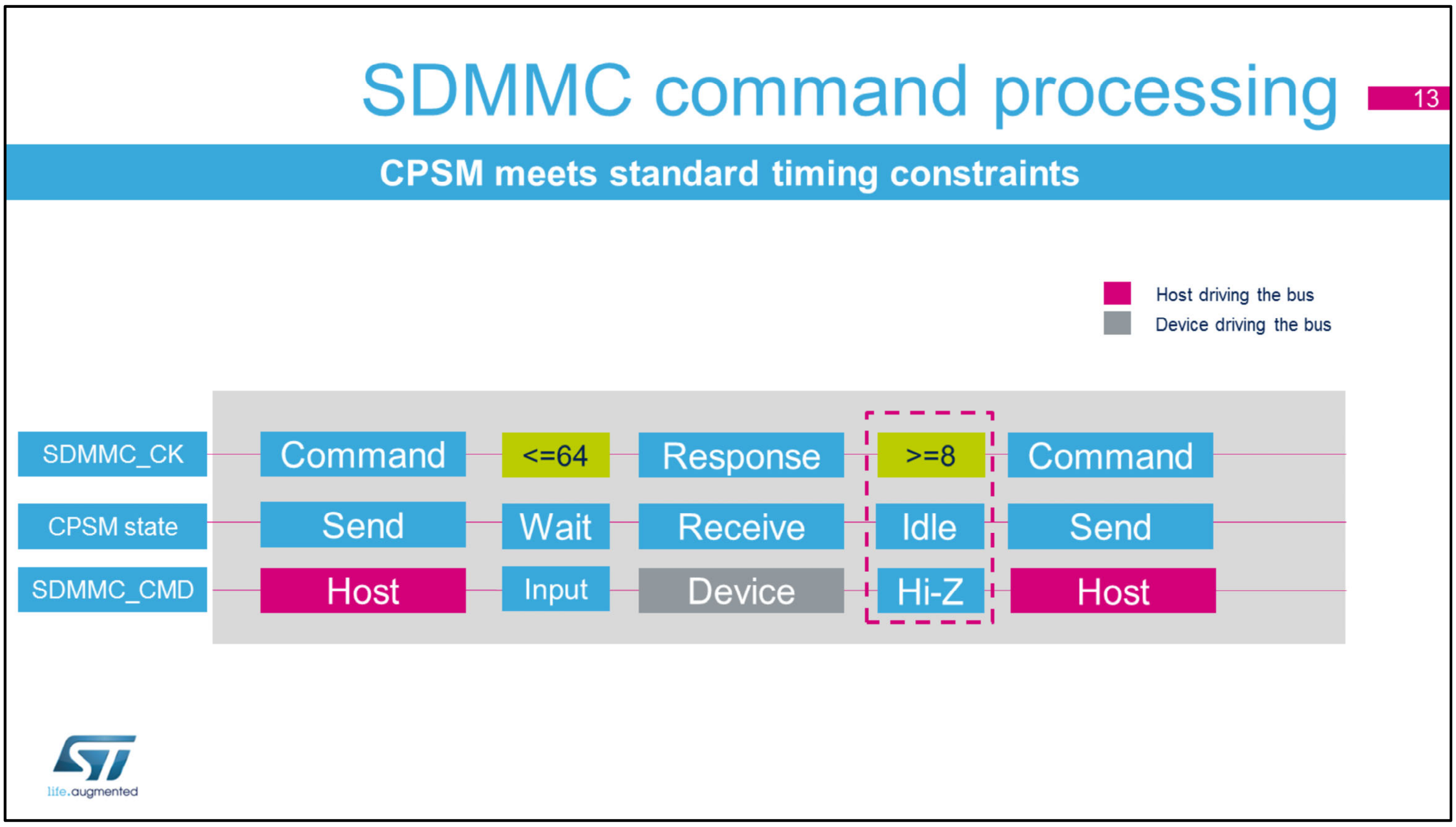
When no response is programmed (WAITRESP=‘00’), the CPSM returns to the Idle state.



SDMMC command processing

CPSM meets standard timing constraints

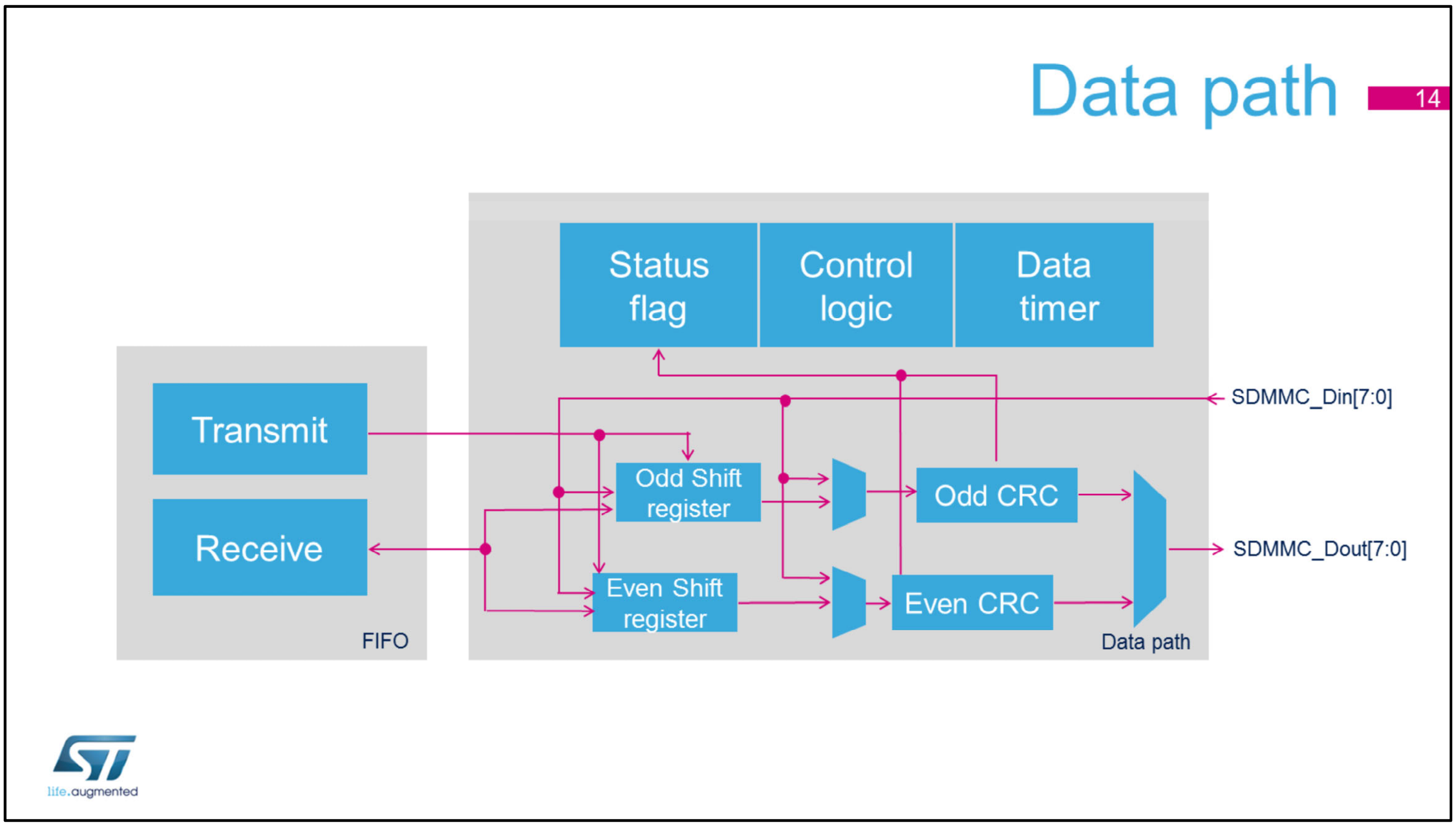
Once a start bit is driven by a device, it is detected on the command line and the CPSM moves to Receive state. When the response is fully received, the received CRC code and the internally-generated checksum code are compared, and the appropriate status flags are set in the SDMMC interface status register. Then the CPSM enters the Idle state.
Note that the CRCFAIL flag will only be set for responses with a CRC and when the CRC check failed.



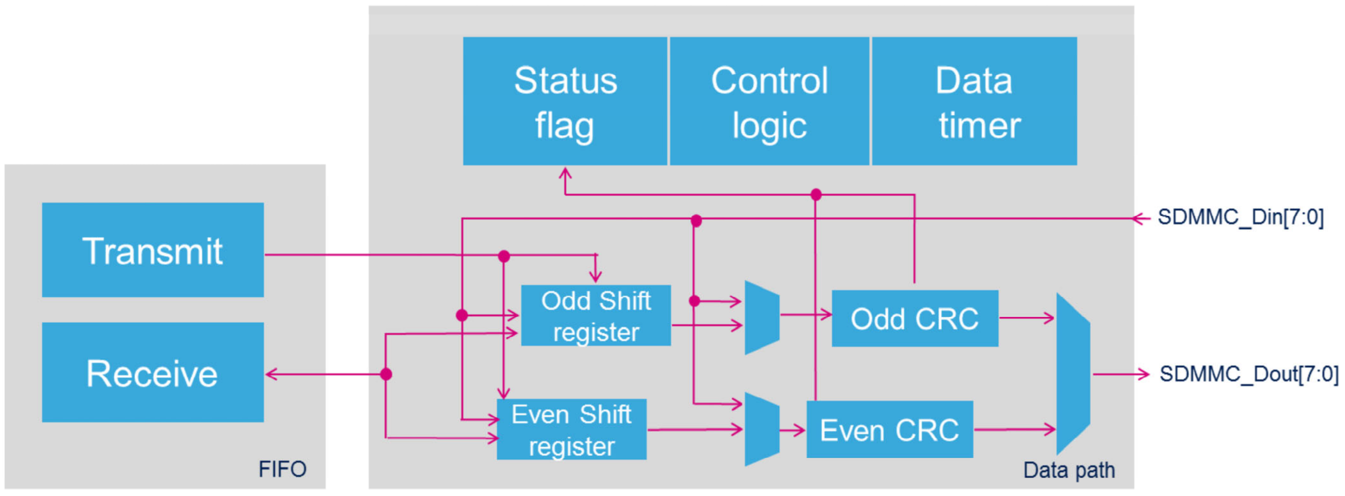
SDMMC command processing

CPSM meets standard timing constraints

After a complete command with a response is received, the CPSM remains in Idle state for at least 8 SDMMC_CK clock periods to meet command-to-command timing (NCC) and response-to-command timing (NRC) constraints.



Data path

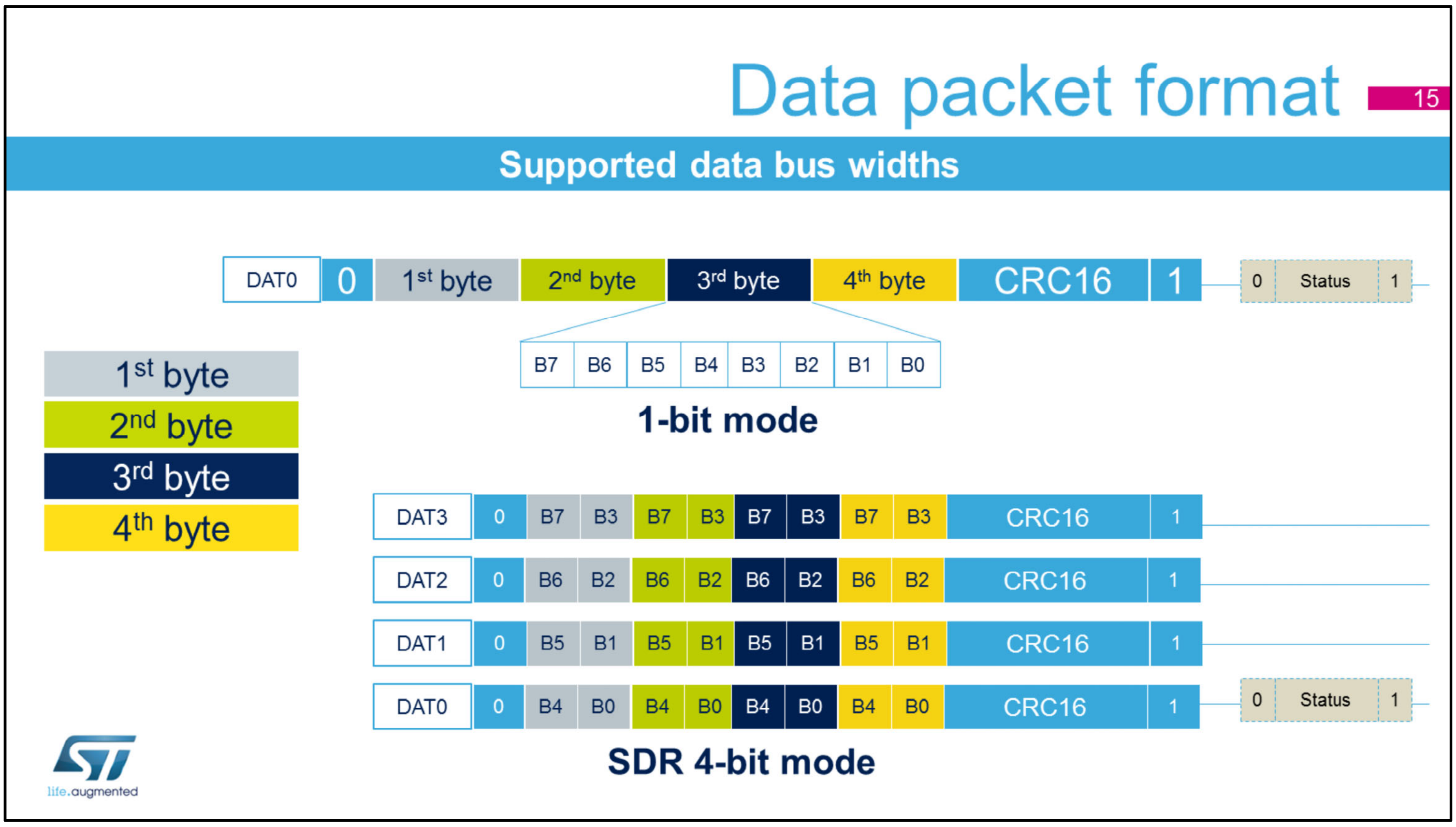


The data path transfers data both to and from the SD/SDIO or MMC card.
In Single data rate (SDR) mode, on each SDMMC_CK clock cycle, the data path can send one, four or eight bits depending on the bus width configuration.
In Double data rate (DDR) mode, on each SDMMC_CK clock cycle, the data path can send two, eight or sixteen bits depending on the bus width configuration.

Transfer logic is clocked by the SDMMCCLK clock. It is divided into two subunits, one for data sent and one for data received with a dedicated control bit and status flags.

The data buffer is not part of the data path. Transmit and receive FIFO logic are mapped in the AHB domain. All signals from the different subunits are resynchronized.

The CRC calculator guarantees data integrity between the card and host. At the end of the data packet, the CRC is calculated automatically.



Data packet format

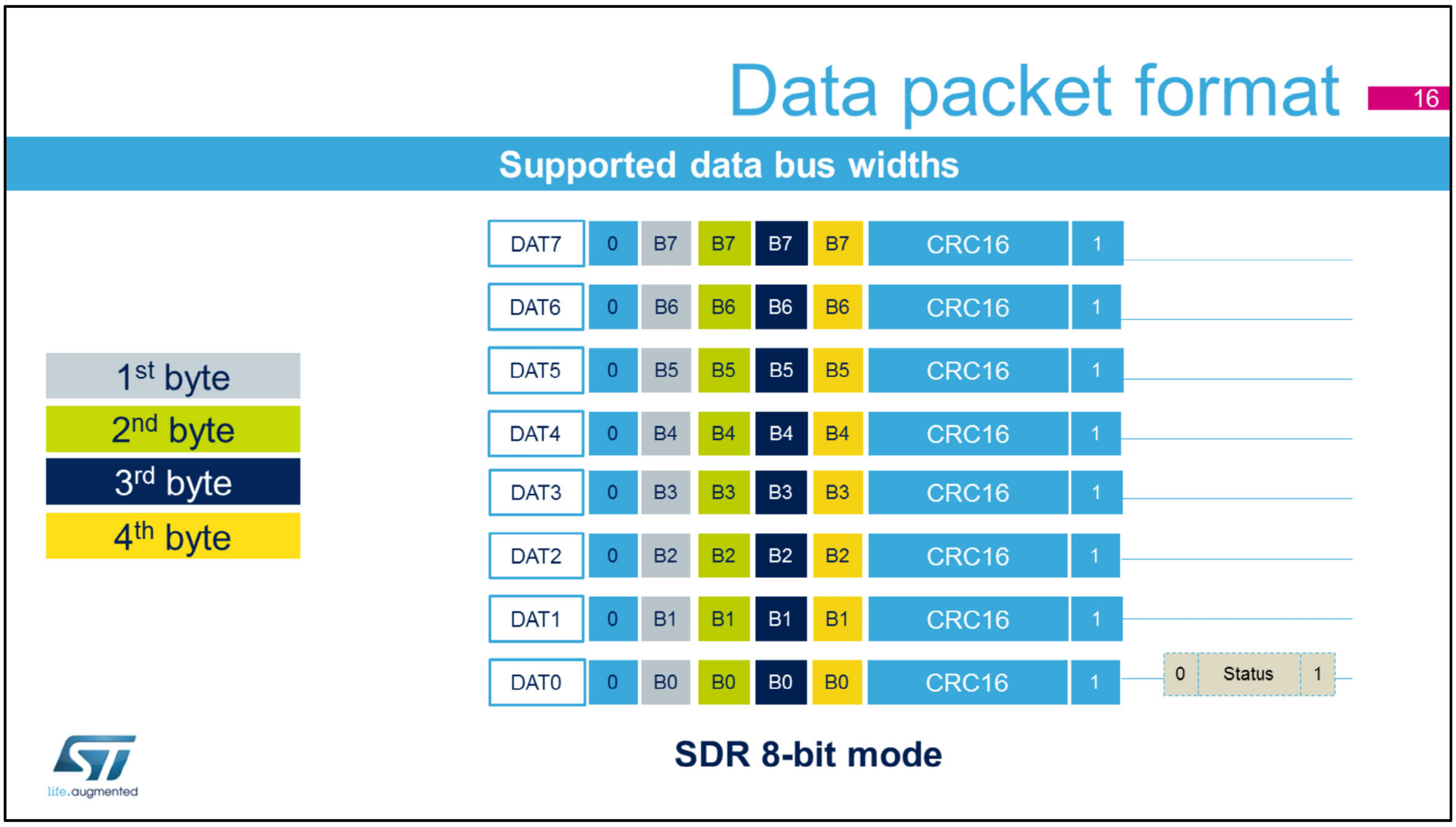
Supported data bus widths

In Single data rate (SDR) mode, depending on the configured data bus width, the data path sends data blocks over one (SDMMC_D0), four (SDMMC_D0 to SDMMC_D3), or eight pins (SDMMC_D0 to SDMMC_D7).

First, a start bit is generated on the bus followed by the data packet with the first to last bytes of the sequence (4th byte in our example). Then, the CRC16 and end bit are appended to the data packet on the bus line.

In a 4-bit data width configuration, each line has its own start bit, end bit and CRC16 checksum.

When the data is sent to the card, the card returns a CRC status on the SDMMC_D0 pin.

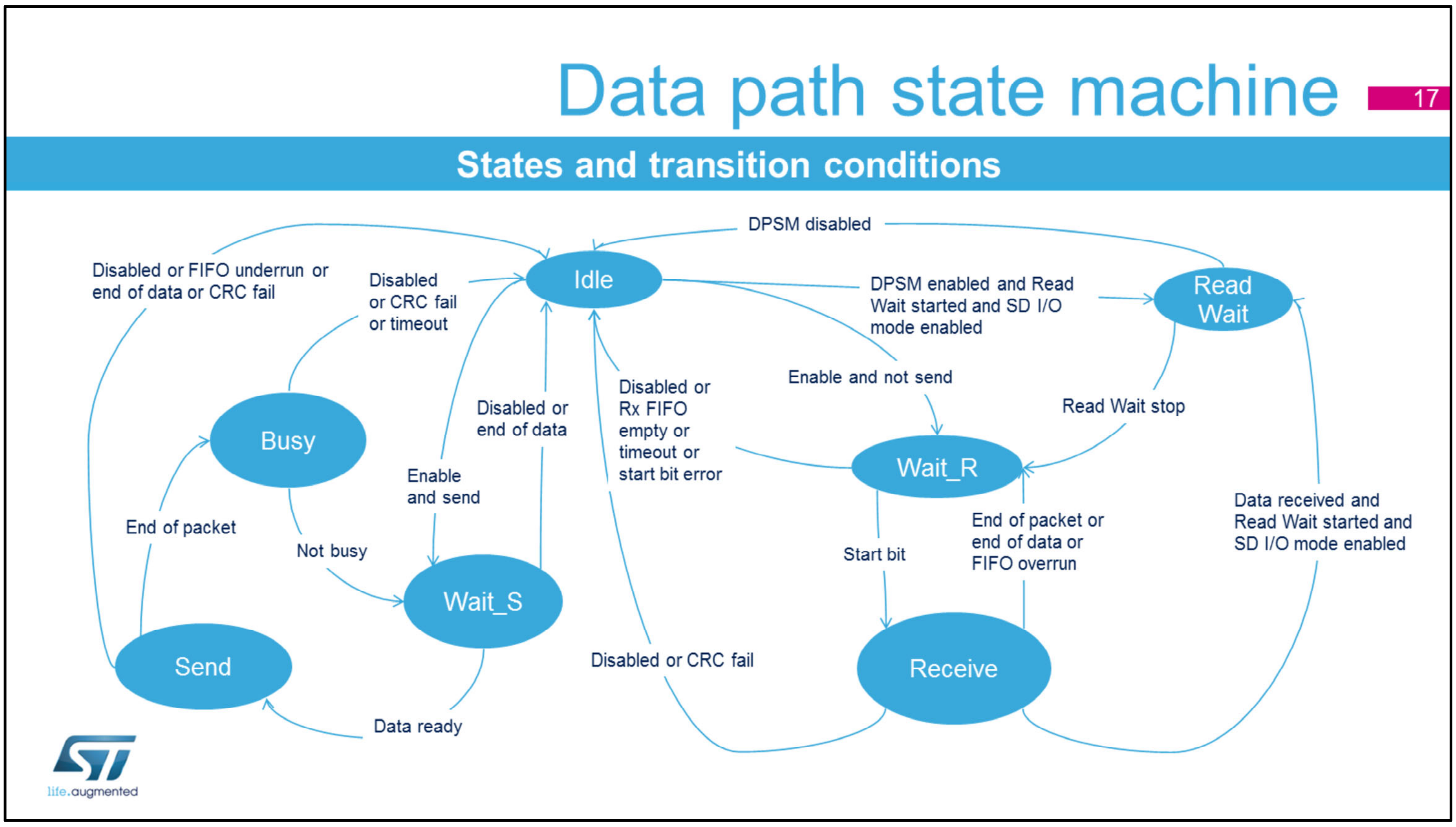


Data packet format

Supported data bus widths

1st byte
2nd byte
3rd byte
4th byte
DAT7 0 B7 B7 B7 B7 CRC16 1
DAT6 0 B6 B6 B6 B6 CRC16 1
DAT5 0 B5 B5 B5 B5 CRC16 1
DAT4 0 B4 B4 B4 B4 CRC16 1
DAT3 0 B3 B3 B3 B3 CRC16 1
DAT2 0 B2 B2 B2 B2 CRC16 1
DAT1 0 B1 B1 B1 B1 CRC16 1
DAT0 0 B0 B0 B0 B0 CRC16 1
0 Status 1
SDR 8-bit mode
life.augmented

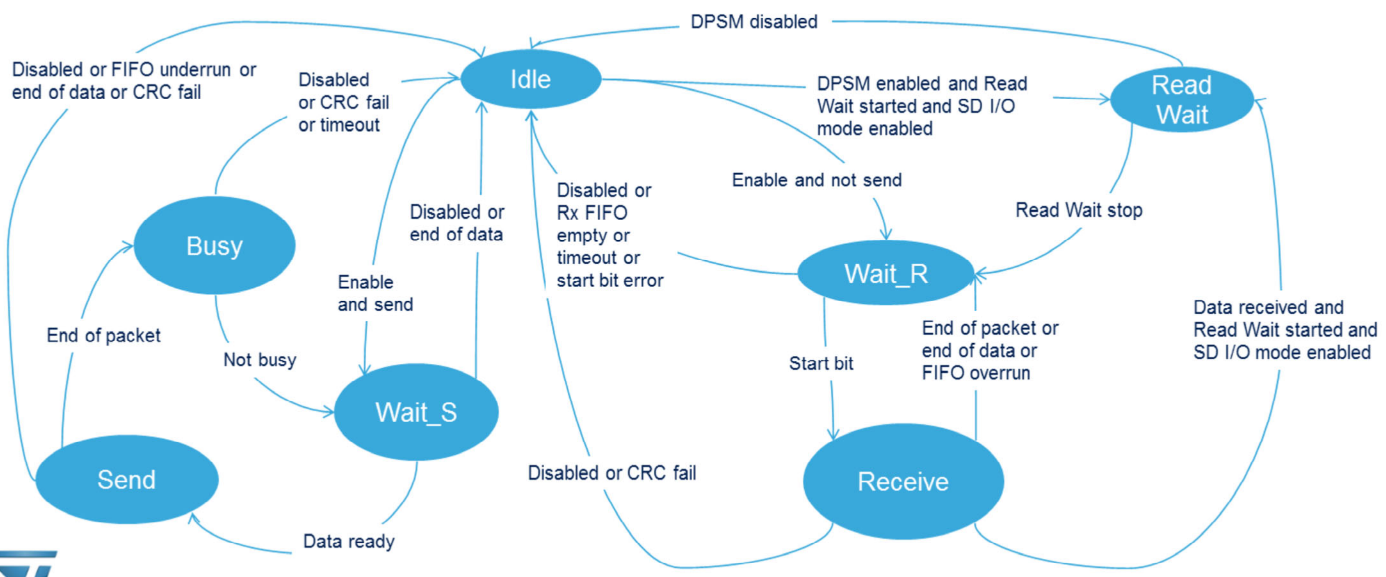
In this example, the four bytes are sent over the SDMMC bus in 8-bit mode. For each SDMMC_CK clock cycle, a byte is shifted out with a start bit, end bit and CRC16 checksum on each data line.
When the data is sent to the card, the card returns a CRC status on the SDMMC_D0 pin.



Data path state machine

17

States and transition conditions



DPSM disabled
Disabled or FIFO underrun or end of data or CRC fail
Disabled or CRC fail or timeout
Idle
DPSM enabled and Read Wait started and SD I/O mode enabled
Read Wait
Busy
Disabled or end of data
Disabled or Rx FIFO empty or timeout or start bit error
Enable and not send
Read Wait stop
Enable and send
Wait_R
End of packet
Not busy
Start bit
End of packet or end of data or FIFO overrun
Data received and Read Wait started and SD I/O mode enabled
Wait_S
Send
Disabled or CRC fail
Receive
Data ready

life.augmented

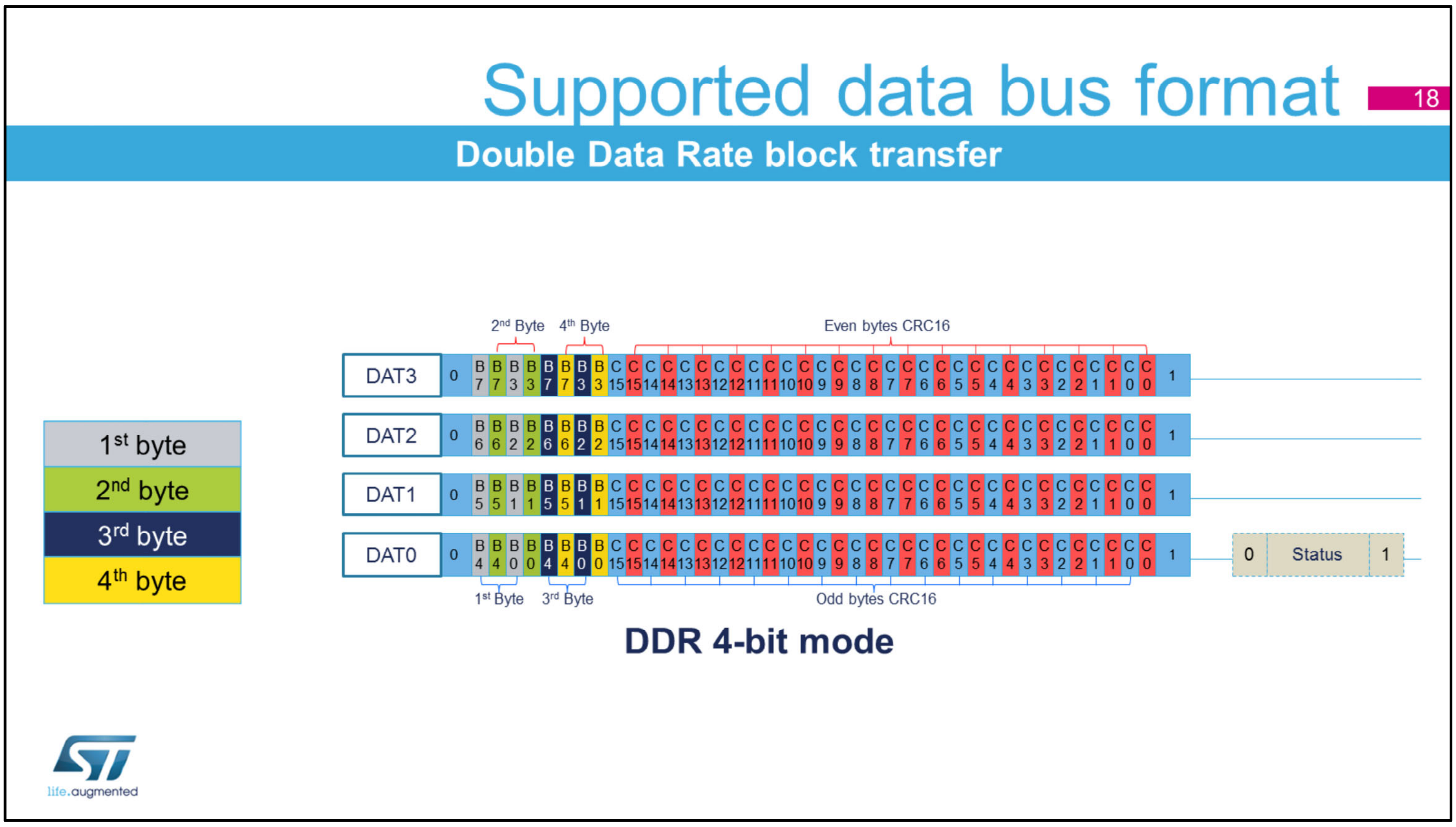
The data path state machine (DPSM) controls the transmission and reception of all data. When the DPSM is in Idle state, the first transition is triggered when the DPSM enable bit and transfer direction are set.
Note: The DPSM enable bit must not be used to transfer data with SD, SDIO and MMC cards.

For data transmission, when enabled, the DPSM moves from Idle to Wait_S state and then to Send state.
While in Wait_S state, the DPSM waits until the data FIFO empty flag is de-asserted.
When data is available in the FIFO buffer, the DPSM moves to the Send state.
In Send state, the DPSM starts sending data to a card according to the data rate, the bus mode and the bus width set in the control register.
At the end of each data packet, the DPSM sends an

internally-generated CRC code and end bit, and moves to the Busy state.
In Busy state, the DPSM waits for the CRC status flag. If it receives a positive CRC status, it moves to Wait_S state when the card is not busy.
From Wait_S state, a new packet transmission can start or the DPSM can return to Idle state when all the data is transmitted or the transfer is disabled.
A negative CRC status from the card or a FIFO underrun error can force the DPSM to return to Idle state, when the card is not busy.

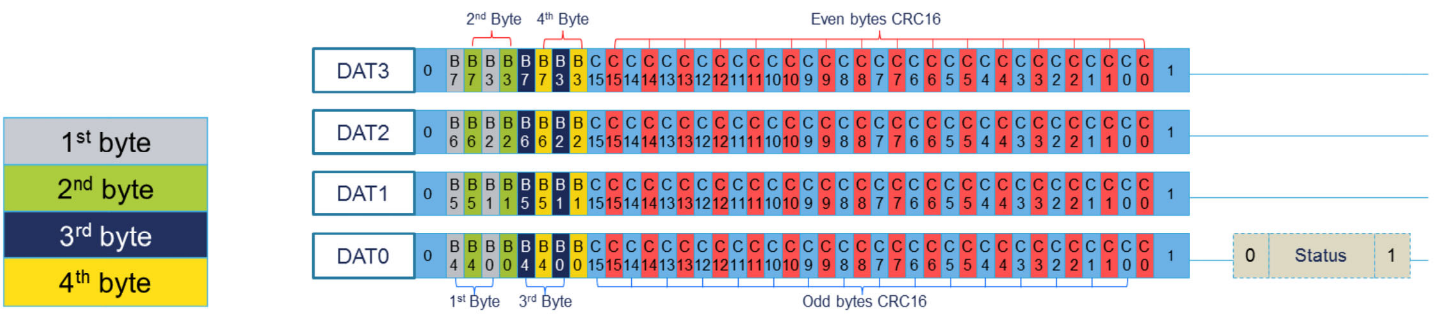
For data reception, the DPSM moves from Idle to Wait_R state. When a start bit is detected on the bus, the DPSM moves to Receive state, where it remains until a full packet is received. As long as the end of data transfer flag and errors are not detected, the DPSM will keep switching between Wait_R and Receive states. If an error or the end of data transfer flag is detected, the DPSM will return to Idle state. If the transfer is disabled, the DPSM moves to the Idle state.
A Read Wait state is an SDIO-specific operation to stall the transfer in order to execute other commands or internal operations. It can be reached from Receive state while a transmission is ongoing or from Idle state. When the firmware requests a read wait stop operation, the DPSM moves to Wait_R state and waits for a start bit from the SDIO device.

For boot acknowledgement, the DPSM moves from Idle to Wait_Ack state. When a positive acknowledgement is detected, the DPSM moves to Wait_R state to receive the Boot data as for data reception. If a boot acknowledgement timeout or a negative boot acknowledgement is received, an Abort command has to be sent by the CPSM where after the DPSM moves to the Idle state.



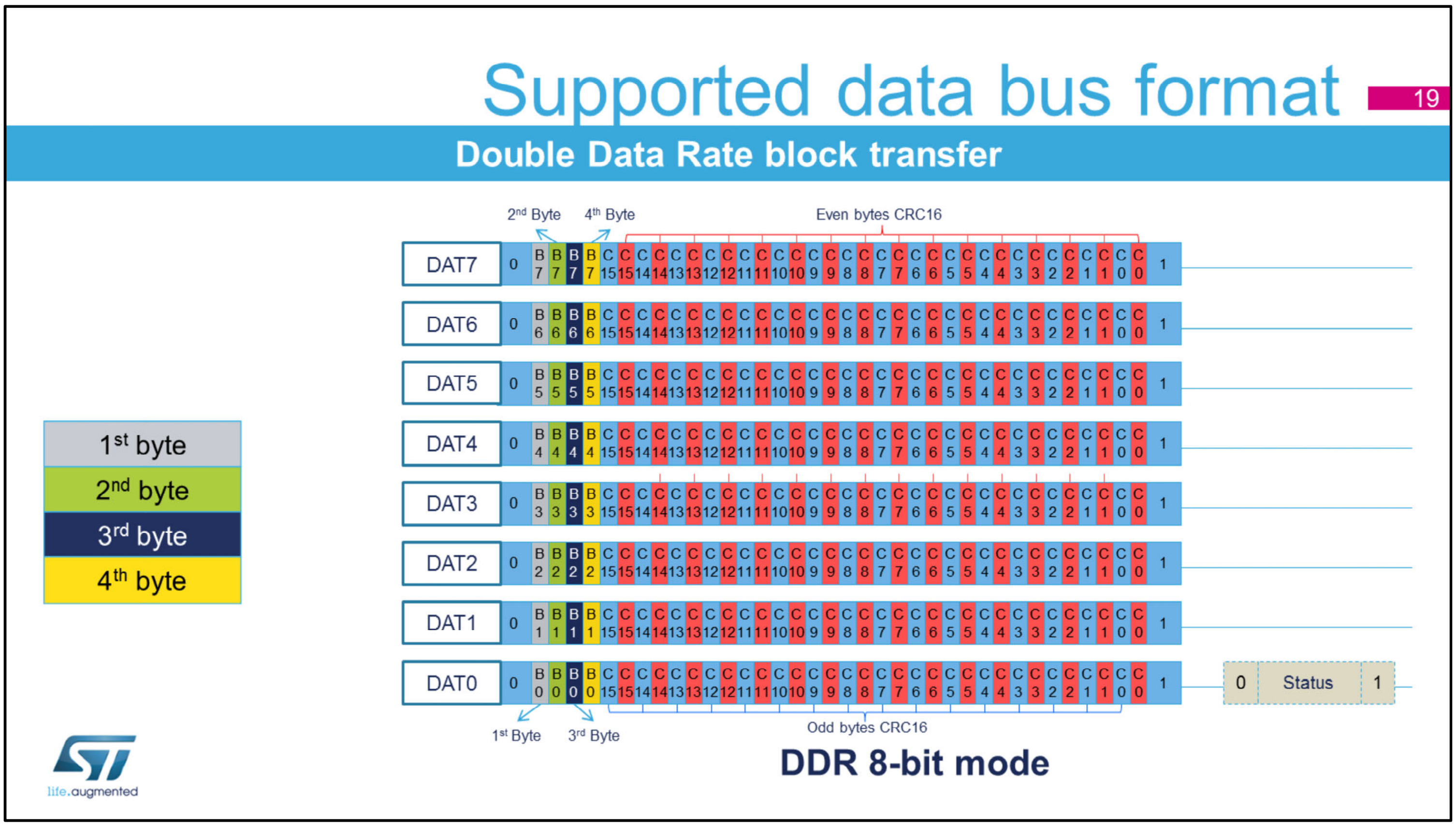
Supported data bus format

Double Data Rate block transfer



DDR 4-bit mode

In Double Data Rate (DDR) mode, depending on the configured data bus width, the data path sends data blocks over four (SDMMC_D0 to SDMMC_D3), or eight pins (SDMMC_D0 to SDMMC_D7).
First, a full cycle start bit is generated on the bus followed by the data packet with the first to last bytes of the sequence, with odd byte data on the falling edge of the clock and even byte data on the rising edge (4th byte in our example). Then, the odd and even CRC16 checksums and full cycle end bit are appended to the data packet on the bus lines.
In a 4-bit data width configuration, each line has its own start bit, end bit and odd and even CRC16 checksums.
When data is sent to the card, the card will return a full cycle CRC status on the SDMMC_D0 pin.
Double Data Rate mode is not available in 1-bit mode.

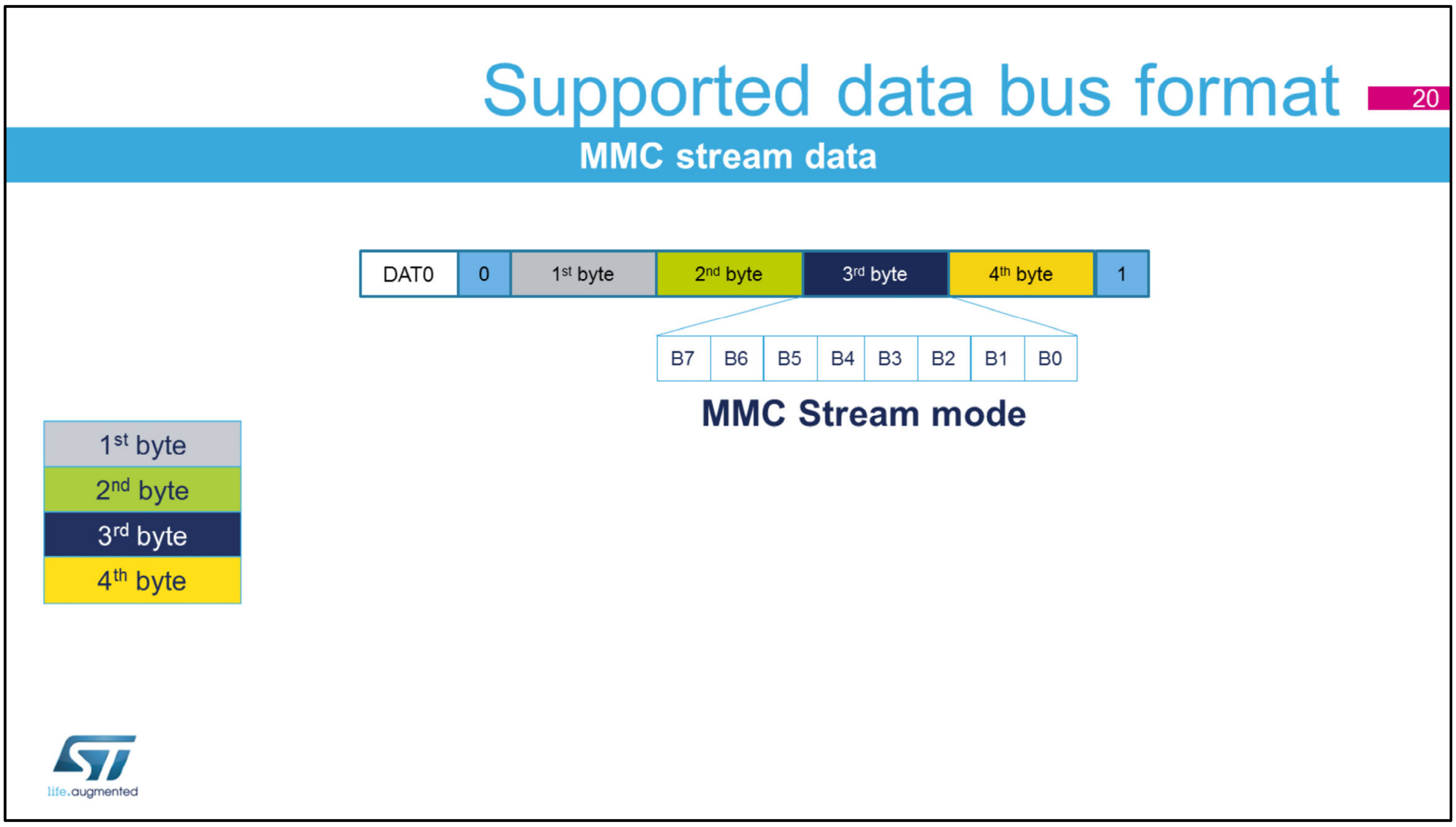


Supported data bus format

Double Data Rate block transfer

In this example, the four bytes are sent over the SDMMC bus in Double Data Rate 8-bit mode. For each SDMMC_CK clock cycle, two bits are shifted out with a block full cycle start bit and end bit and odd and even CRC16 checksums on each data line.

When data is sent to the card, the card will return a full cycle CRC status on the SDMMC_D0 pin.



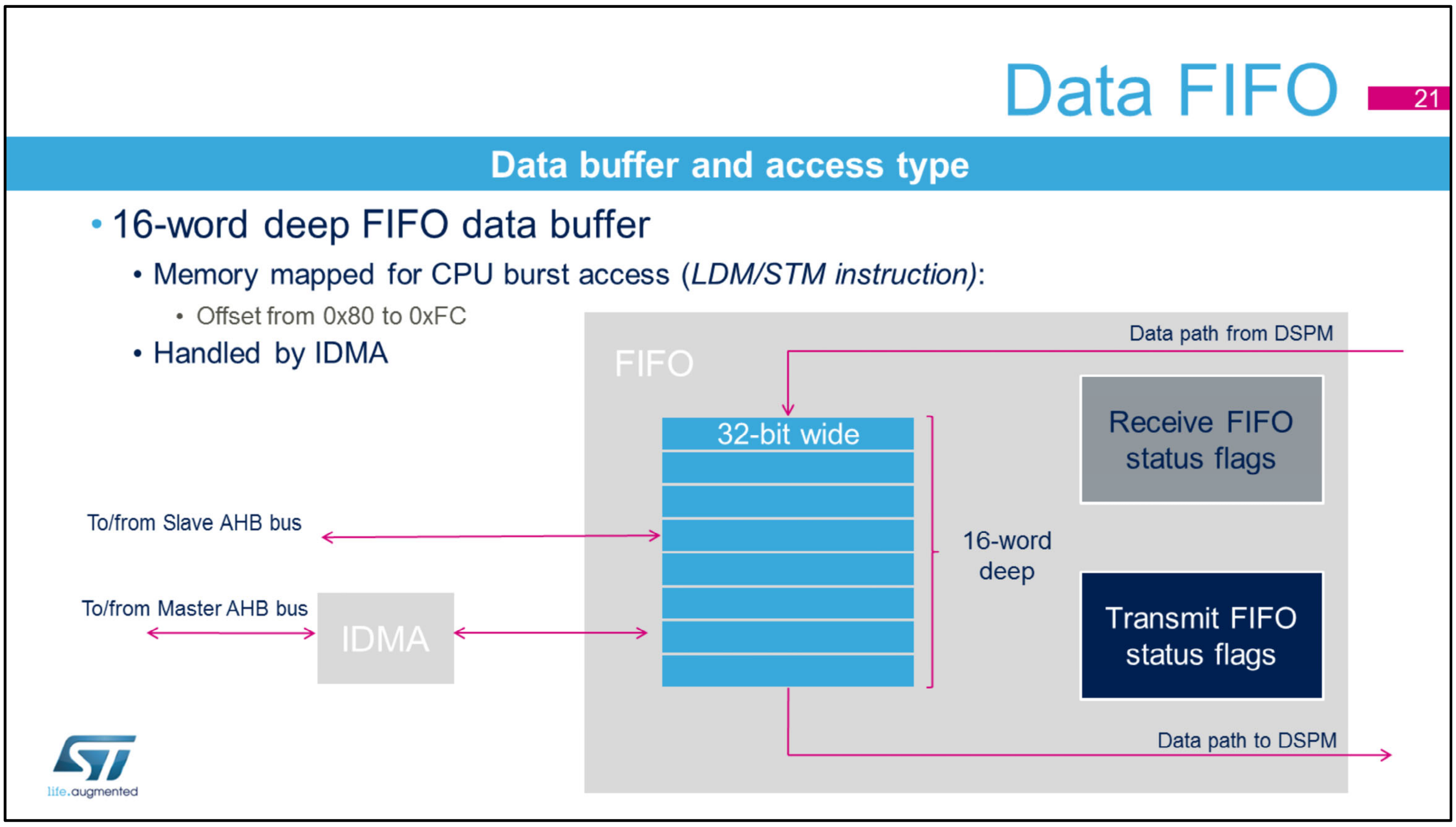
Supported data bus format

MMC stream data

In MMC Stream mode, the data path sends a steam over one pin (SDMMC_D0).

First, a start bit is generated on the bus followed by the data stream with the first to last bytes of the sequence (4th byte in our example). Then, the end bit is appended to the stream on the bus line.

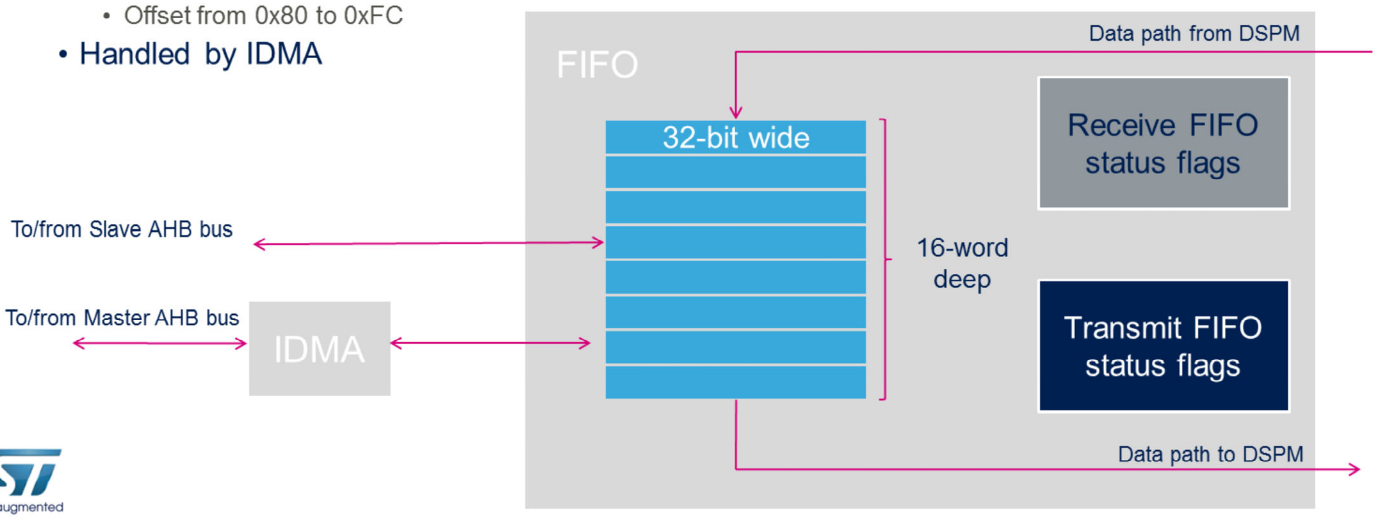
In Stream mode there is no CRC, and the card will not return a CRC status after having received data.



Data FIFO

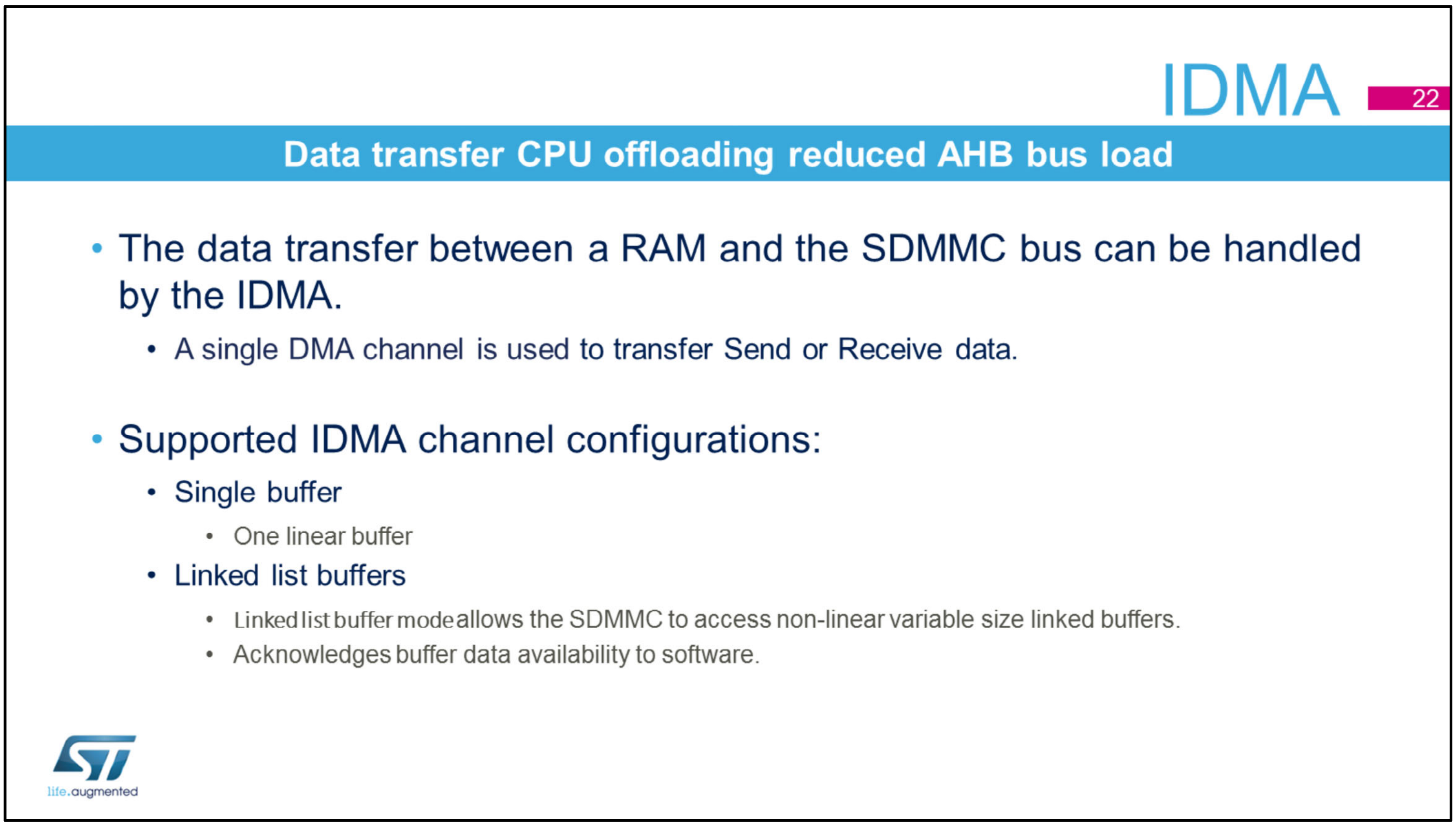
Data buffer and access type

- 16-word deep FIFO data buffer
  - Memory mapped for CPU burst access (*LDM/STM instruction*):
    - Offset from 0x80 to 0xFC
  - Handled by IDMA



A 32-bit wide, 16-word deep FIFO is used to buffer data between the Slave AHB domain and the IDMA on the Master AHB domain.
A single data FIFO is the data source for the data path transmit and receive packets. Depending on the DPSM status, the data path FIFO can be disabled, transmit enabled or receive enabled.

Dedicated receive and transmit FIFO status flags are available to ease firmware implementation.
When enabled, the IDMA transfers data between the FIFO and an external memory, off loading the CPU.



IDMA

22

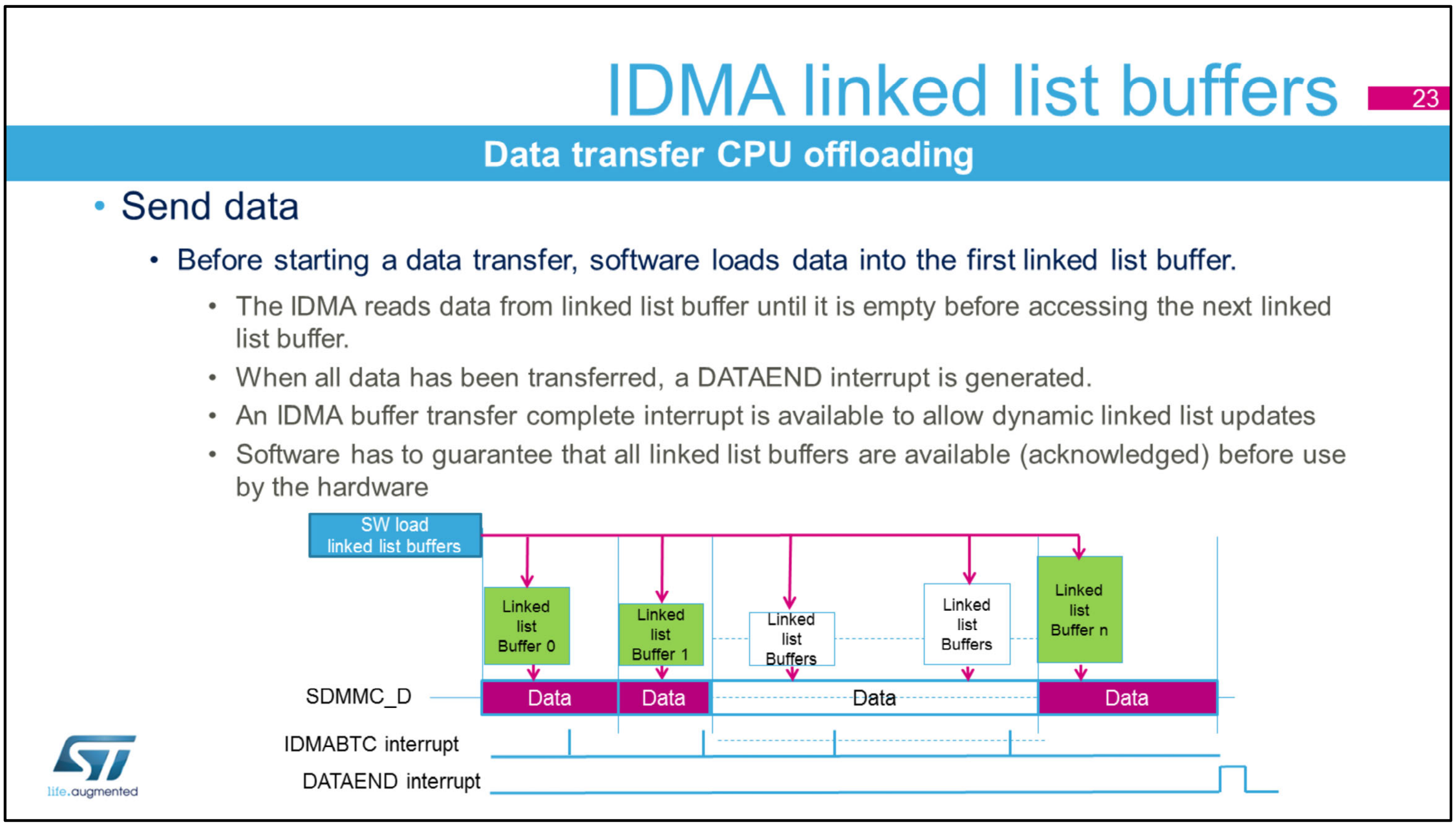
Data transfer CPU offloading reduced AHB bus load

- The data transfer between a RAM and the SDMMC bus can be handled by the IDMA.
  - A single DMA channel is used to transfer Send or Receive data.
- Supported IDMA channel configurations:
  - Single buffer
    - One linear buffer
  - Linked list buffers
    - Linked list buffer mode allows the SDMMC to access non-linear variable size linked buffers.
    - Acknowledges buffer data availability to software.

life.augmented

The integrated DMA inside the SDMMC transfers data between memory and the SDMMC FIFO, allowing to reduce the CPU processing and AHB bus load. Two IDMA operating modes are supported:

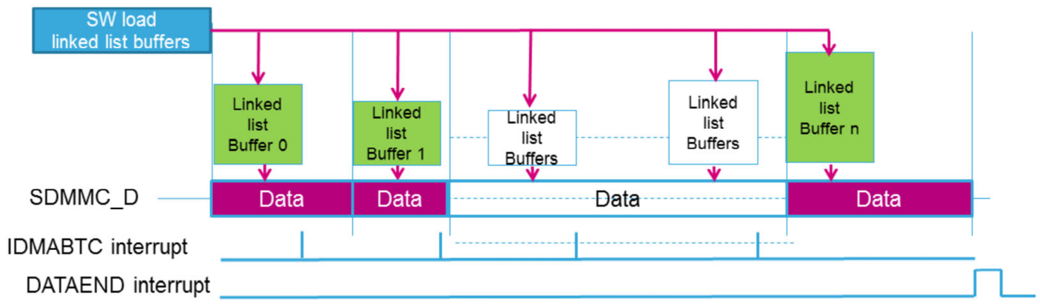
- Single buffer mode, where all transfer data is located in a single linear buffer.
- Linked list buffer mode used to transfer data located in multiple variable size buffers, addressed by a linked list op pointers.



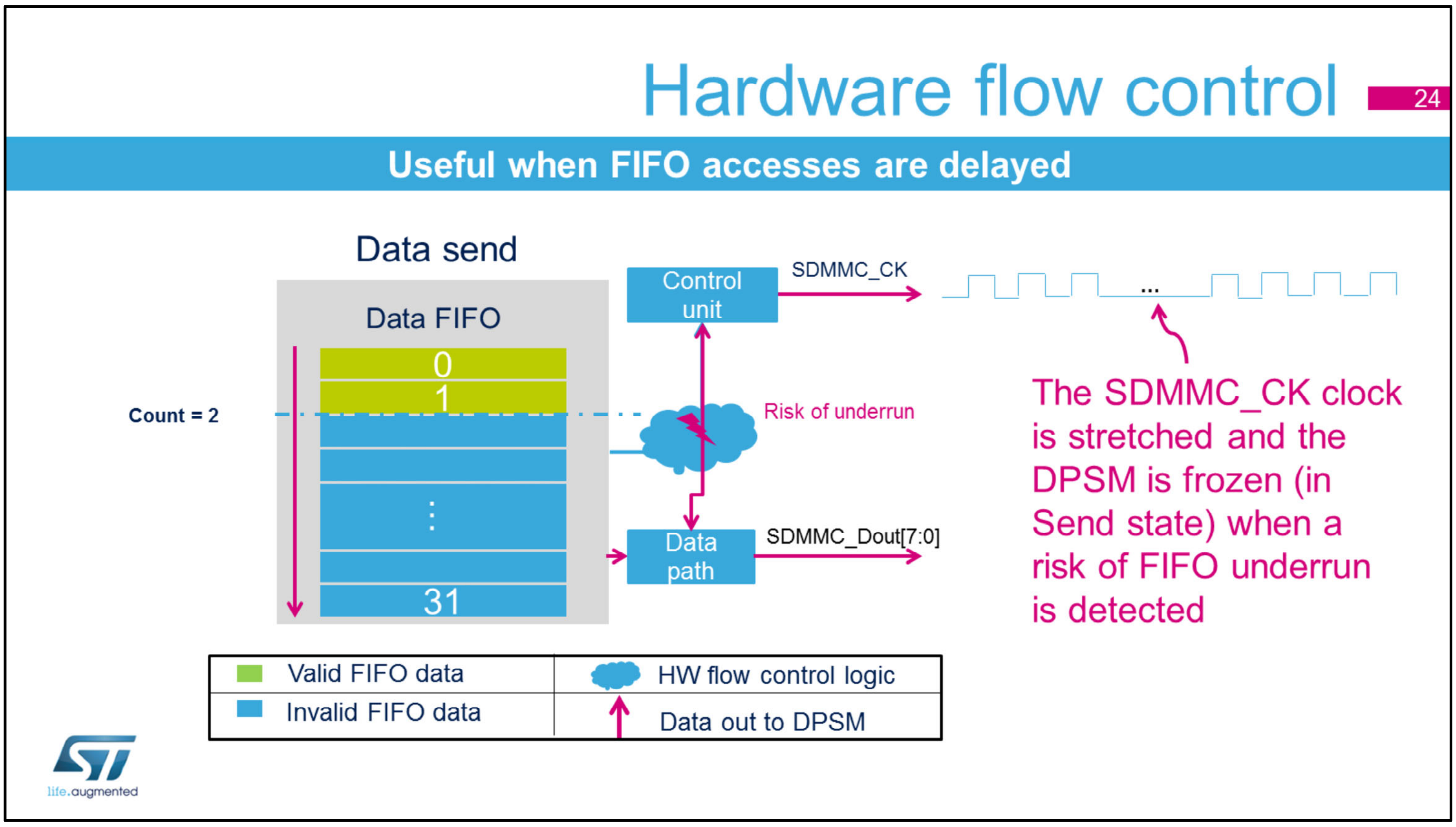
IDMA linked list buffers

Data transfer CPU offloading

- Send data
  - Before starting a data transfer, software loads data into the first linked list buffer.
    - The IDMA reads data from linked list buffer until it is empty before accessing the next linked list buffer.
    - When all data has been transferred, a DATAEND interrupt is generated.
    - An IDMA buffer transfer complete interrupt is available to allow dynamic linked list updates
    - Software has to guarantee that all linked list buffers are available (acknowledged) before use by the hardware



The IDMA linked list buffer mode allows firmware to update one buffer, while the SDMMC transfers the data of other buffers. Each linked list buffer has a base address and buffer size. Every time the IDMA reaches the end of one buffer, an IDMA Buffer Transfer Complete interrupt may be generated when a subsequent linked list buffer is requested. Each linked list buffer provides an acknowledgment for data availability. At the end of the SDMMC transfer, a DATAEND interrupt is generated.



Hardware flow control

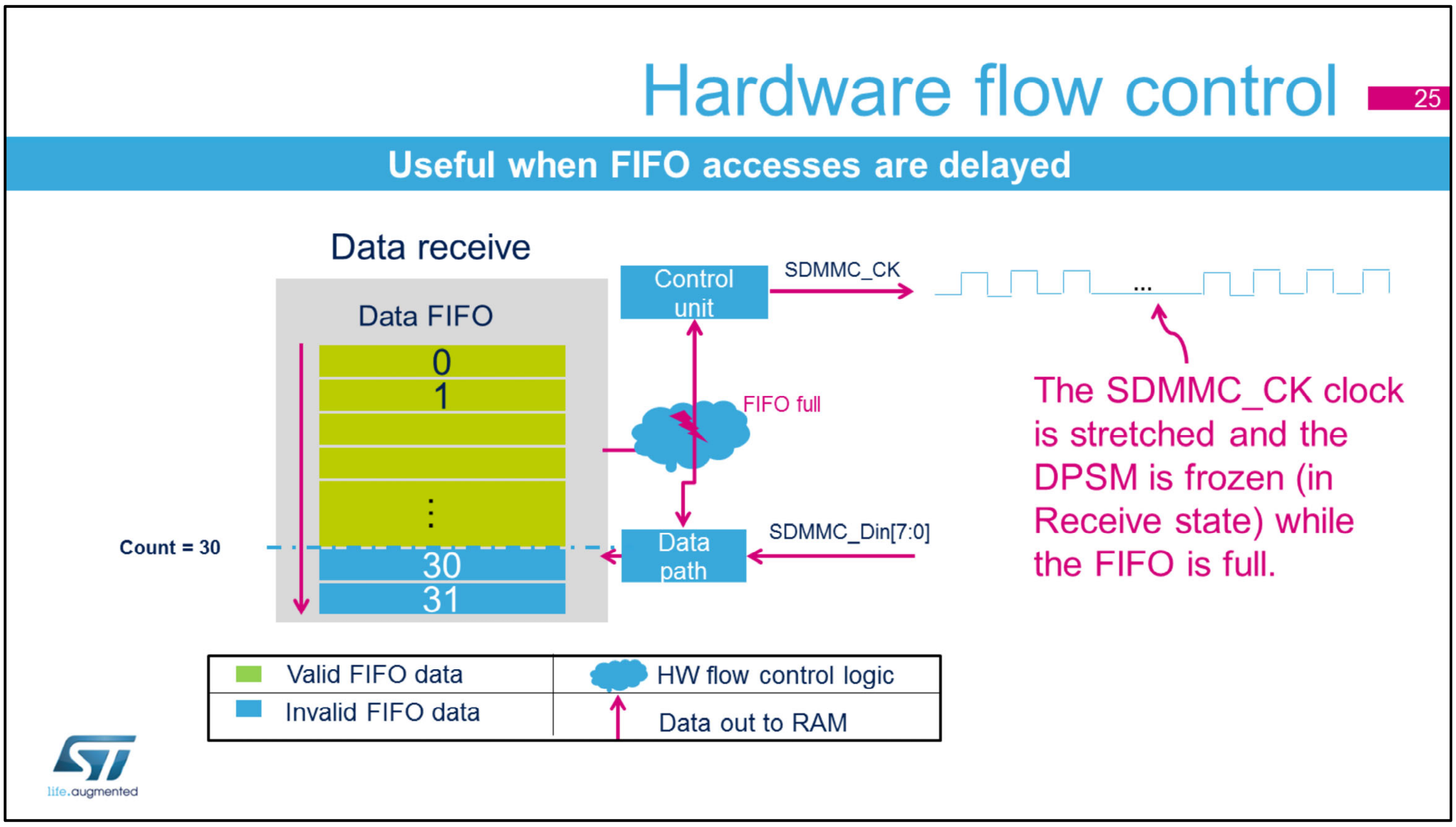
Useful when FIFO accesses are delayed

The SDMMC_CK clock is stretched and the DPSM is frozen (in Send state) when a risk of FIFO underrun is detected

🟩 Valid FIFO data	HW flow control logic
🟦 Invalid FIFO data	↑ Data out to DPSM

The hardware flow control function is used to avoid FIFO underrun (when DPSM is in Send mode) and overrun errors (when DPSM is in Receive mode). The hardware flow control logic stops the SDMMC_CK pin signals and freezes the DPSM when a risk of underrun/overrun is detected.

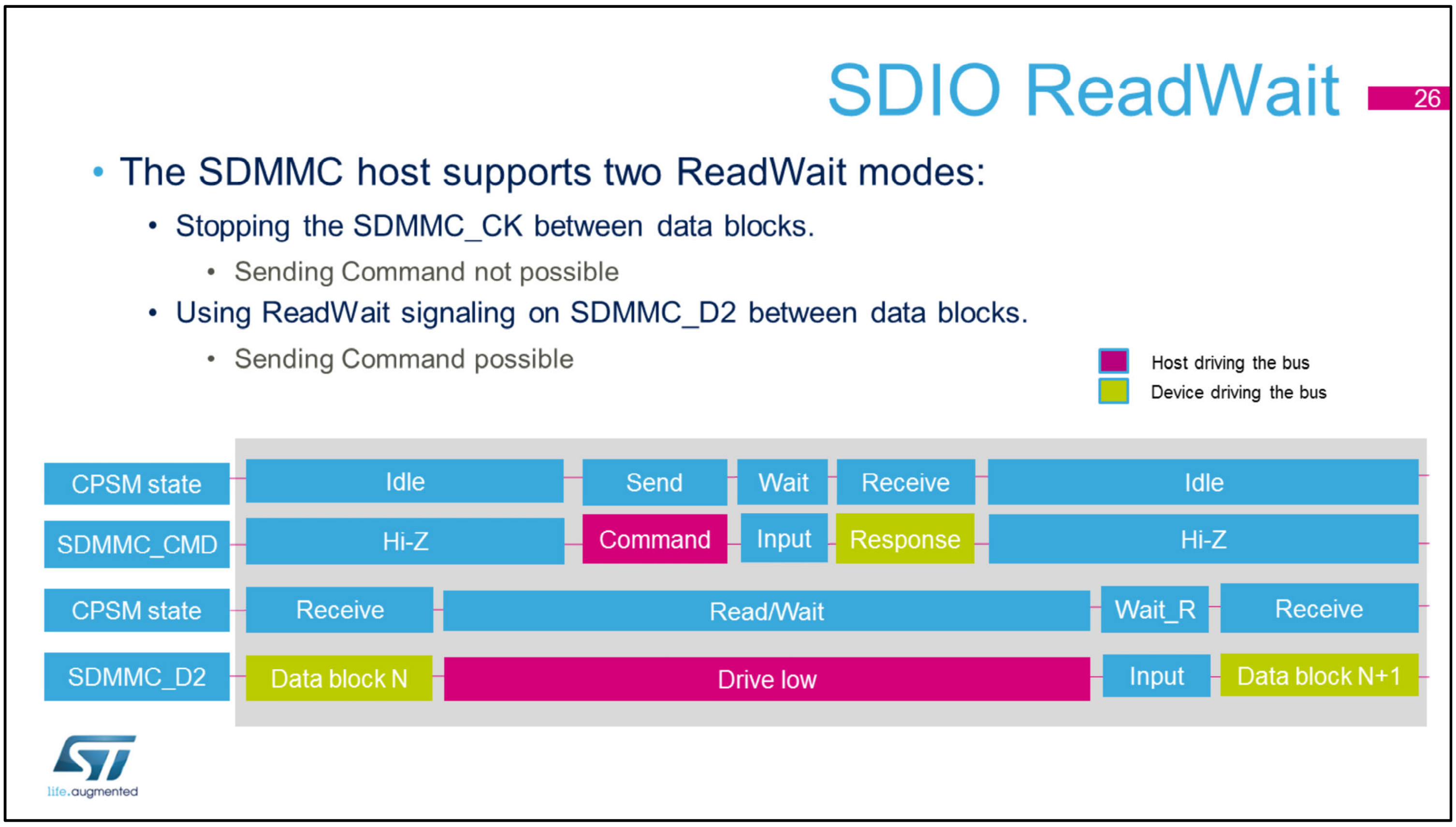
Hardware flow control must not be used with a variable delay, i.e. SDR104.
In Send state, the SDMMC_CK pin clock signal is stretched and the DPSM is frozen to prevent any FIFO underruns. The clock and DPSM are restarted when the FIFO is half full or all the last transfer data is available in the FIFO.



Hardware flow control

Useful when FIFO accesses are delayed

In Receive state, the SDMMC_CK clock is stretched and the DPSM is frozen in Receive state while the FIFO is full (risk of overrun). The clock and DPSM are restarted when the FIFO becomes half empty.

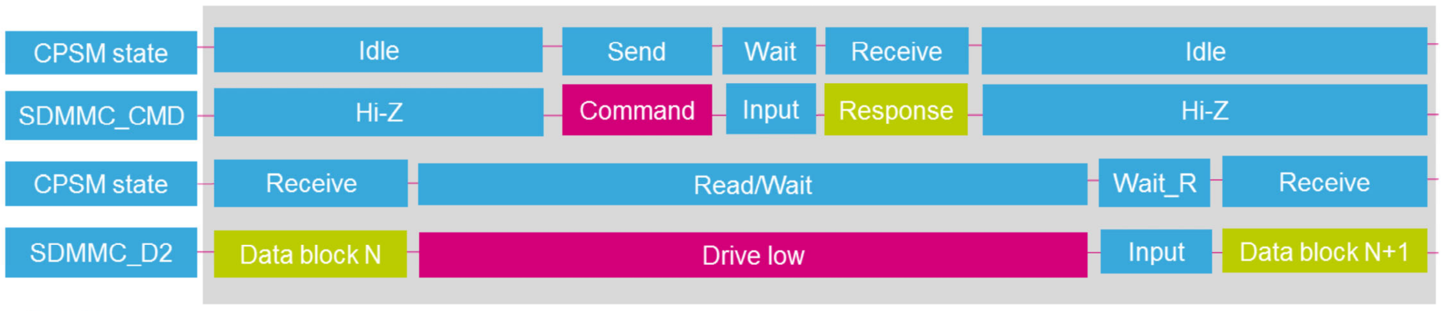


SDIO ReadWait

26

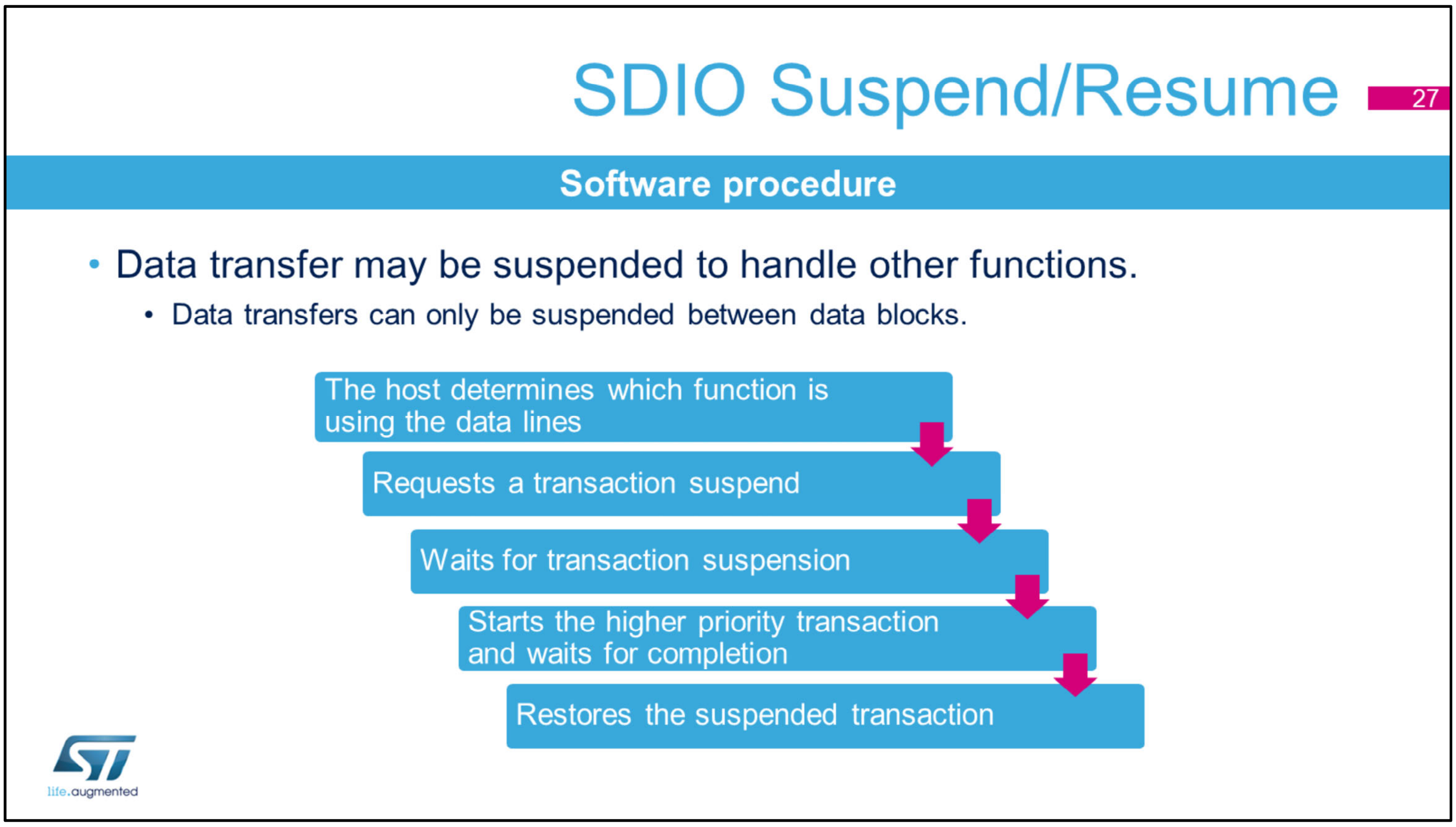
- The SDMMC host supports two ReadWait modes:
  - Stopping the SDMMC_CK between data blocks.
    - Sending Command not possible
  - Using ReadWait signaling on SDMMC_D2 between data blocks.
    - Sending Command possible

🟥 Host driving the bus
🟩 Device driving the bus



Concept: The Read Wait operation is an SDIO-specific operation that allows the host to temporarily stall the data transfer between data blocks, for better managing the data buffer or for sending commands to other functions of the SDIO device.
The SDMMC controller supports two Read Wait modes: either by stopping the SDMMC_CK or using SDMMC_D2 signaling.
The advantage of SDMMC_D2 signaling is you are still able to communicate with the card while in Read Wait mode.
When the RWSTART bit is set, after the data block is fully transferred and the CRC code is correct, the DPSM moves to ReadWait state.
The DPSM remains in ReadWait state until terminated by writing 1 to the RWSTOP bit. (RWSTOP bit is auto cleared by hardware once the ReadWait phase is

terminated.)
If the CRC code fails any further, all data transfers are stopped and the DPSM remains in the Receive state. An Abort command needs to be sent by the CPSM which moves the DPSM to Idle state. In this case, the ReadWait state will NOT be entered.
Read Wait mode must be used with a variable delay, i.e. SDR104 to handle FIFO flow control.

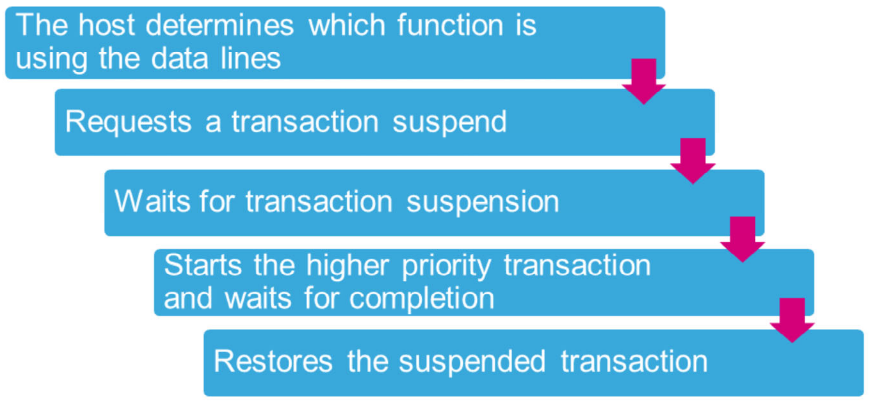


SDIO Suspend/Resume

27

Software procedure

- Data transfer may be suspended to handle other functions.
  - Data transfers can only be suspended between data blocks.

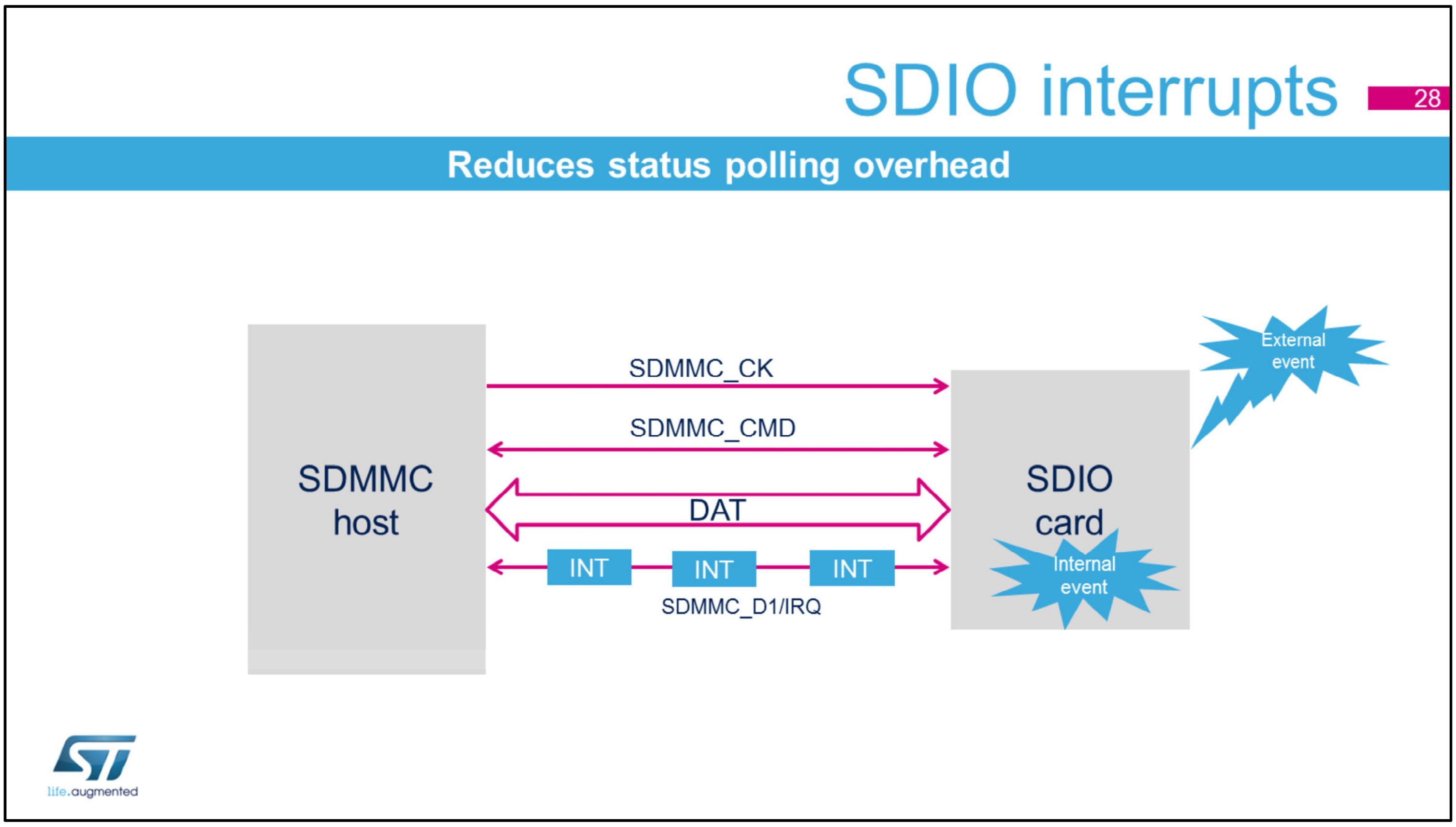

The host determines which function is using the data lines
Requests a transaction suspend
Waits for transaction suspension
Starts the higher priority transaction and waits for completion
Restores the suspended transaction

life.augmented

With multi-function cards, there are multiple devices that share the access to the SD bus. When the function supports Suspend/Resume, the host can temporarily halt data transfers to perform other internal operations or to communicate with other functions and then resume the suspended transaction.
If a card supports the suspend/resume feature, the host can temporarily halt a data transfer operation to one function or memory in order to free the bus for a higher priority transfer to a different function or memory.
A command with the CMDSUSPEND bit set must be sent to indicate to the SDMMC that the current command is a Suspend command.

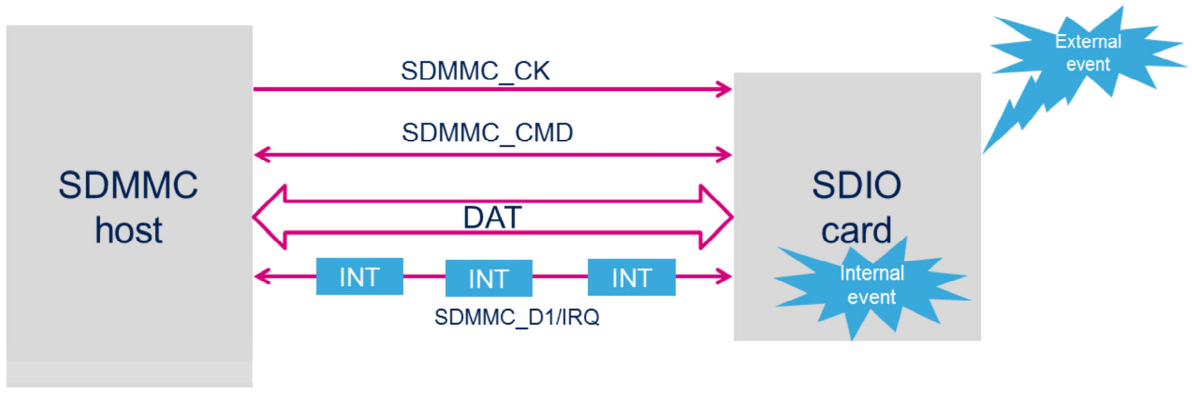
If a Suspend request is accepted, the DPSM will wait in Wait_R state, as the function is only suspended after a complete data block. When IDMA mode is used, it empties the reception FIFO. If the application reads the FIFO, it has to empty the reception FIFO before setting the DTHOLD bit to Idle state. When the FIFO is empty and the DTHOLD bit is set, the DPSM moves to Idle state.

Only then can the firmware start communication with a higher priority portion of the card.
In order to restore a suspended transaction, the firmware needs to reconfigure the DPSM to read the remaining data before requesting a function resume.
This function is no longer supported in SDIO version 4.00 or more recent.



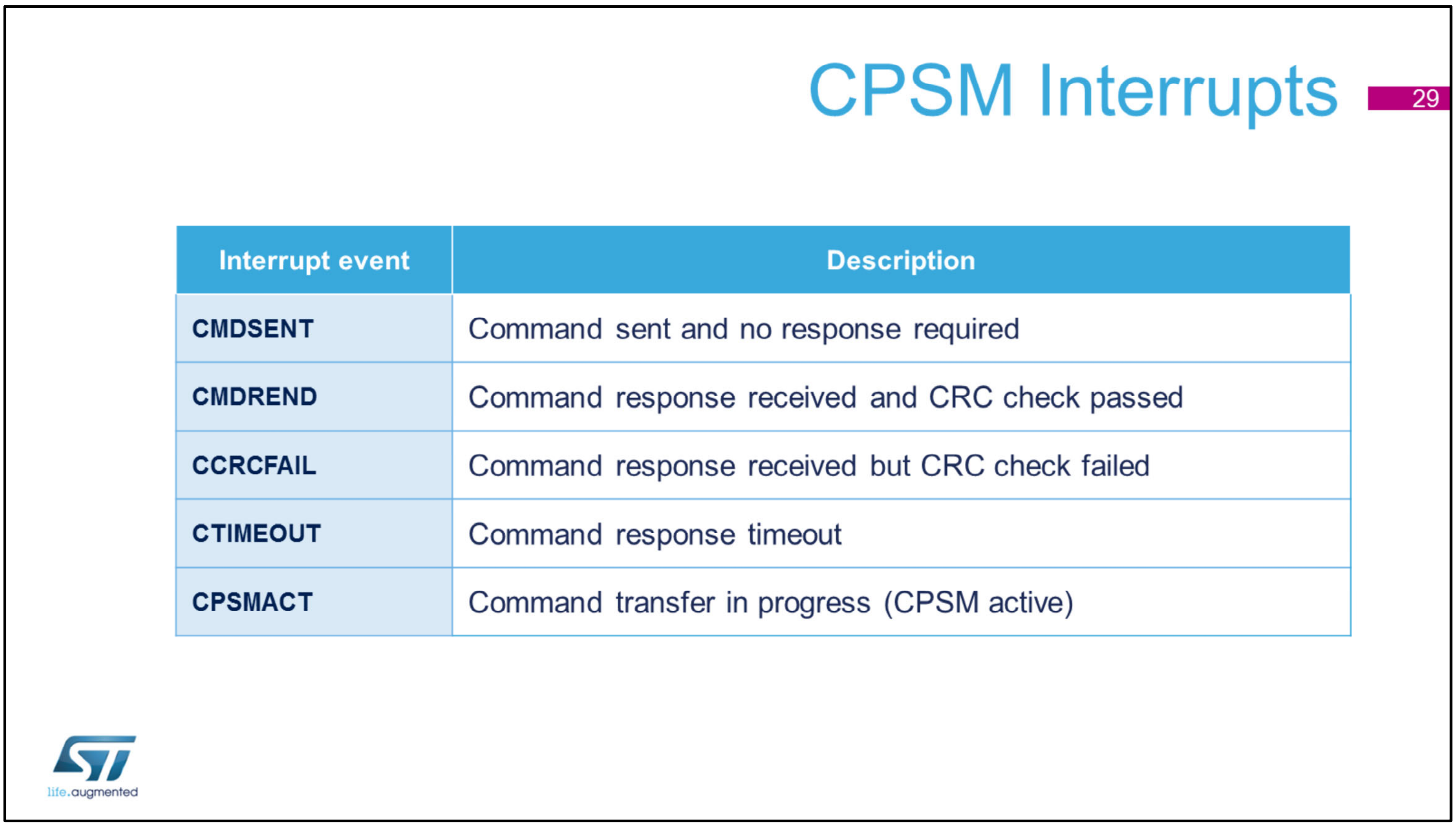
SDIO interrupts

Reduces status polling overhead



The interrupt concept is used to inform the host of changes in the card status using the SDMMC_D1/IRQ pin in 1-bit or in 4-bit data bus mode.
SDIO interrupts are sent from the card to the SDMMC host when the card detects an external event.
Interrupts are only sent outside the data transfer periods.
The SDMMC host detects interrupts sent on the SDMMC_D1 pin once the SDIOEN configuration bit in the data control register is enabled.

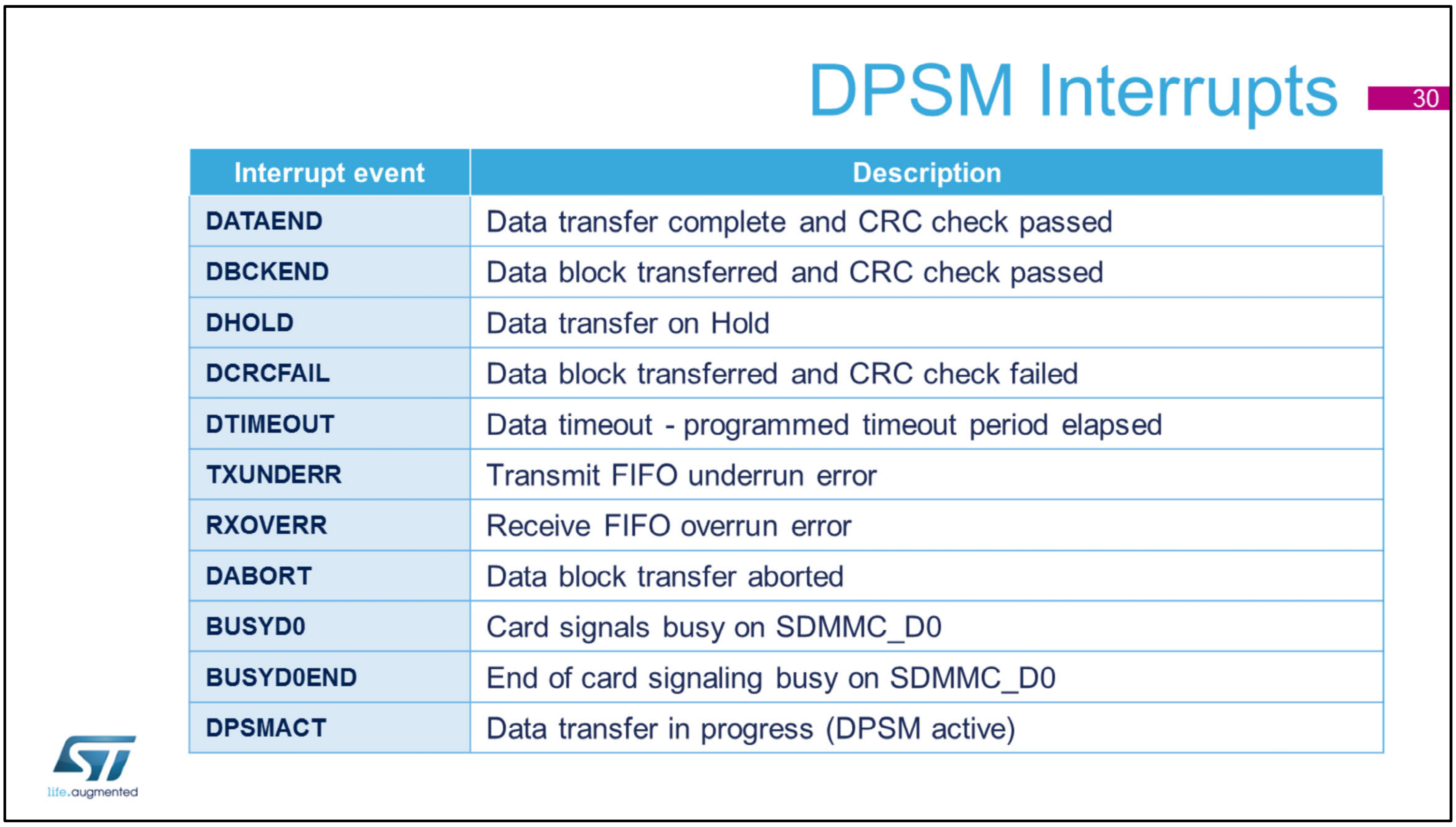
While the DPSM remains in Idle state and in Busy state between data blocks for DS, HS, SDR12 and SDR25 speed or after the last data block in all speed modes, all low levels on the SDMMC_D1 pin are detected as interrupts from the card to the host.



CPSM Interrupts

Interrupt event	Description
CMDSENT	Command sent and no response required
CMDREND	Command response received and CRC check passed
CCRCFAIL	Command response received but CRC check failed
CTIMEOUT	Command response timeout
CPSMACT	Command transfer in progress (CPSM active)

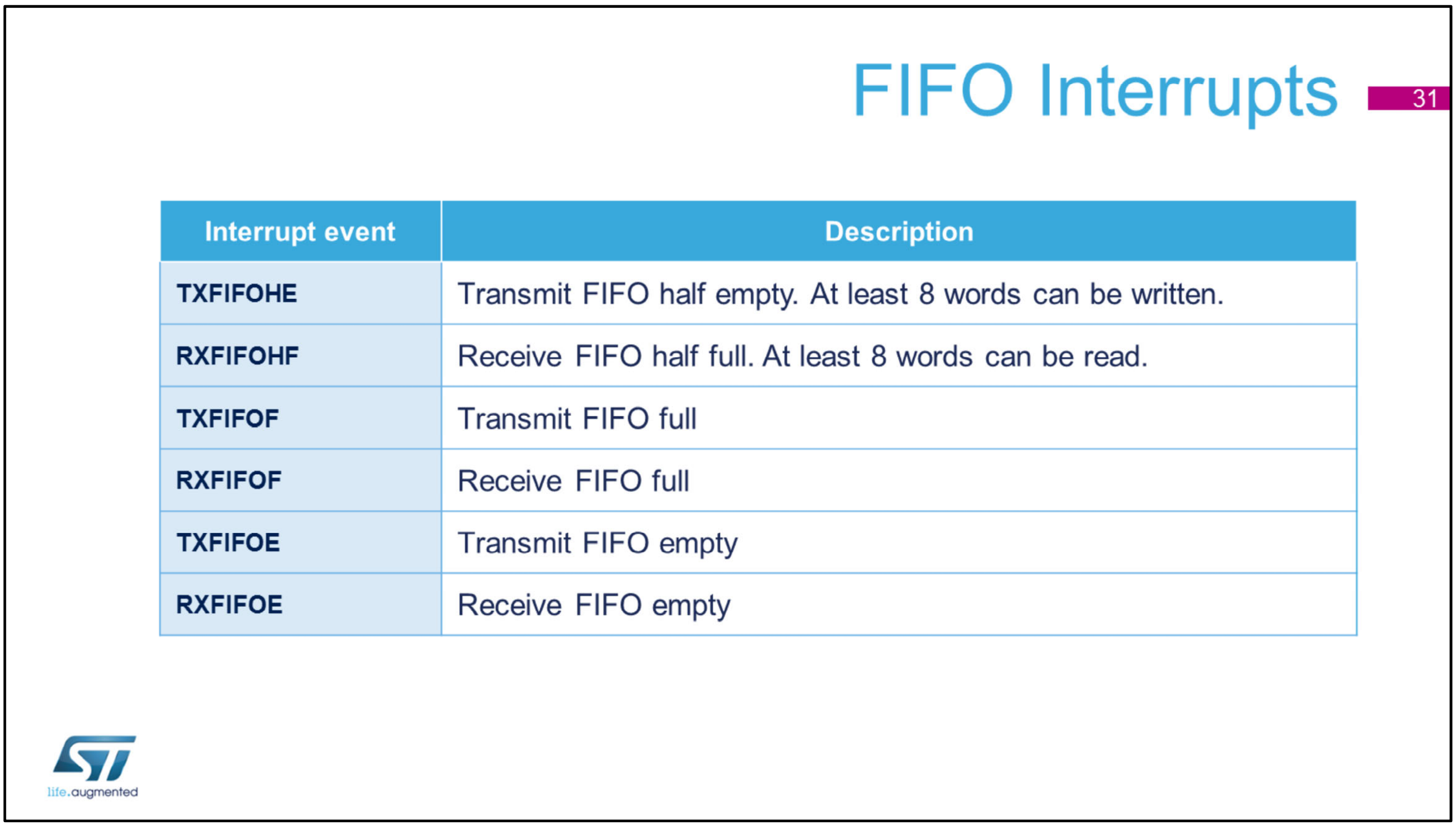
Here is an overview of DPSM interrupt events related to data transfers.
Correct transfers of all data is signaled by DATAEND.
Correct transfer of a data block is signaled by DBCKEND, used with the ReadWait feature
Data transfer errors are signaled with DCRCFAIL, DTIMEOUT, TXUNDERR, and RXOVERR.
Aborting an ongoing data transfer or after a transfer error is signaled with DABORT.
Data send transfer busy indication is signaled with BUSYD0 and BUSYD0END, used with the R1b command, SD voltage switch feature, and SDMMC Sleep feature.
DPSMACT signals when a data transfer is in progress.



DPSM Interrupts

Interrupt event	Description
DATAEND	Data transfer complete and CRC check passed
DBCKEND	Data block transferred and CRC check passed
DHOLD	Data transfer on Hold
DCRCFAIL	Data block transferred and CRC check failed
DTIMEOUT	Data timeout - programmed timeout period elapsed
TXUNDERR	Transmit FIFO underrun error
RXOVERR	Receive FIFO overrun error
DABORT	Data block transfer aborted
BUSYD0	Card signals busy on SDMMC_D0
BUSYD0END	End of card signaling busy on SDMMC_D0
DPSMACT	Data transfer in progress (DPSM active)

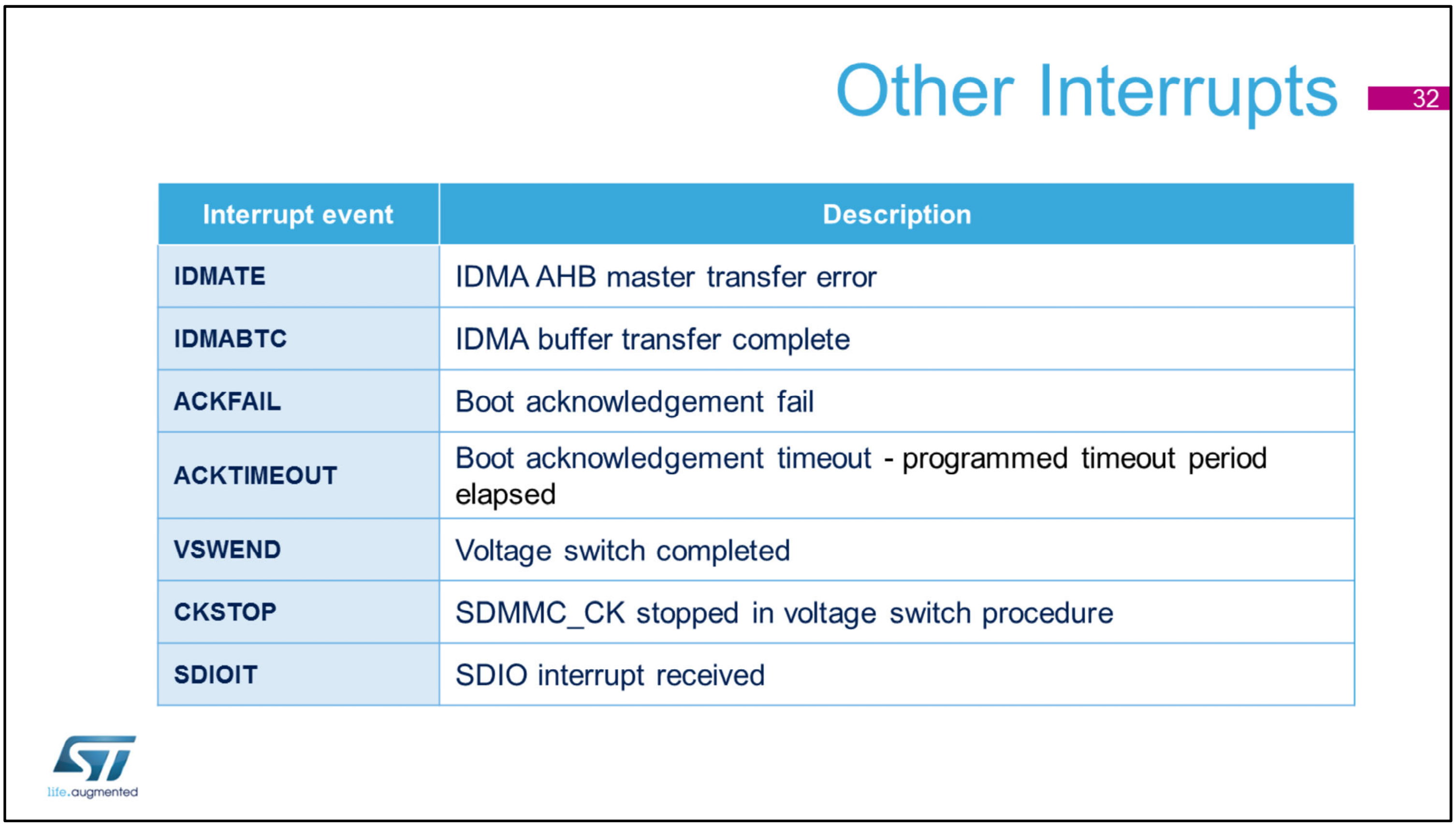
Here is an overview of DPSM interrupt events related to data transfers.
Correct transfers of all data is signaled by DATAEND .
Correct transfer of a data block is signaled by DBCKEND, used with readwait feature
Data transfer errors are signaled with DCRCFAIL, DTIMEOUT, TXUNDERR, and RXOVERR.
Aborting an ongoing data transfer or after a transfer error is signaled with DABORT.
Data send transfer busy indication is signaled with BUSYD0 and BUSYD0END, used with R1b command, SD Voltage switch feature, and SDMMC Sleep feature.
DPSMACT signals when the a data transfer is in progress.



FIFO Interrupts

Interrupt event	Description
TXFIFOHE	Transmit FIFO half empty. At least 8 words can be written.
RXFIFOHF	Receive FIFO half full. At least 8 words can be read.
TXFIFOF	Transmit FIFO full
RXFIFOF	Receive FIFO full
TXFIFOE	Transmit FIFO empty
RXFIFOE	Receive FIFO empty

Here is the list of flags available for FIFO management in Interrupt and Polling modes.
DMA requests are internally generated when triggered by FIFO threshold events.

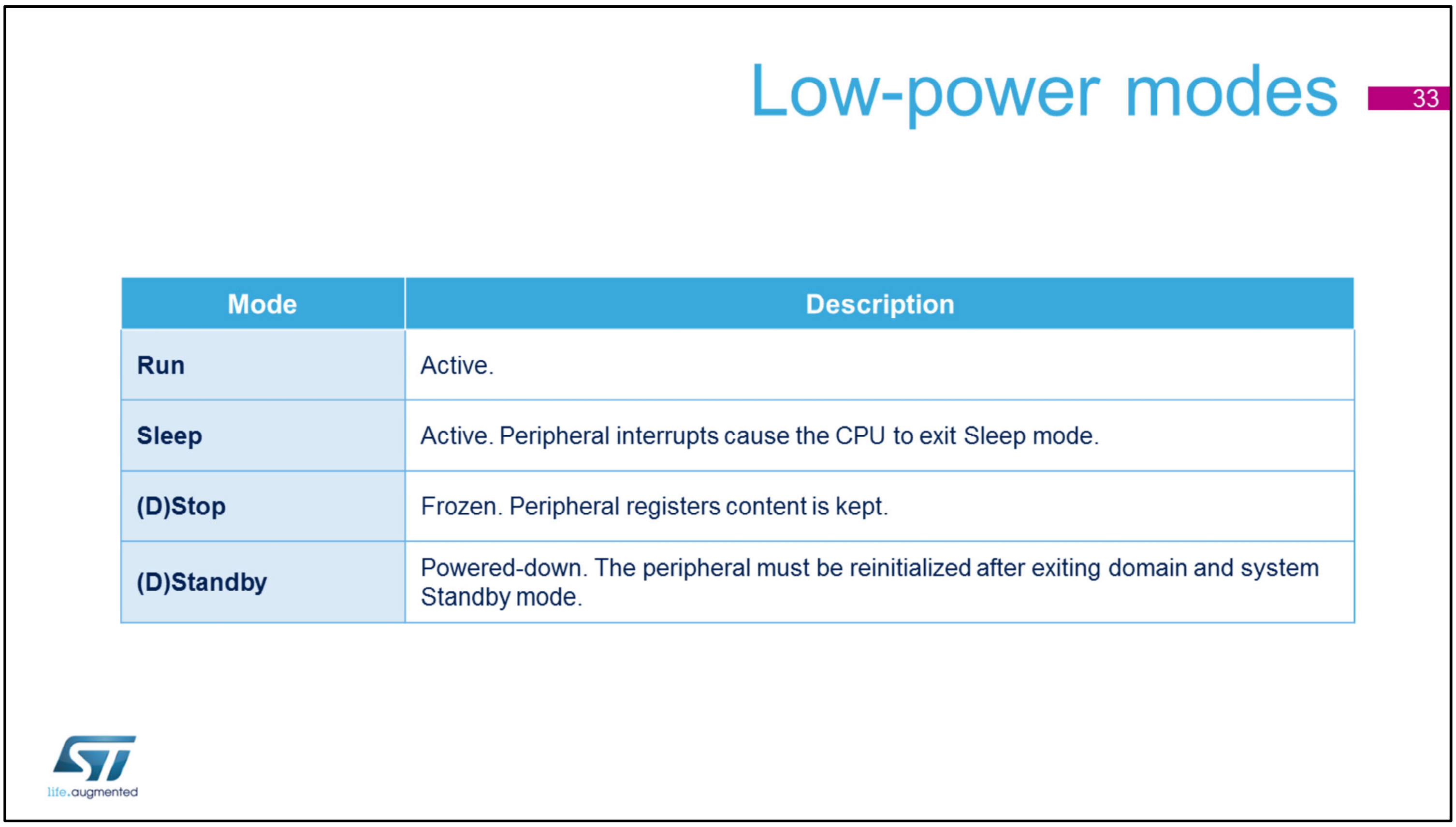


Other Interrupts

Interrupt event	Description
IDMATE	IDMA AHB master transfer error
IDMABTC	IDMA buffer transfer complete
ACKFAIL	Boot acknowledgement fail
ACKTIMEOUT	Boot acknowledgement timeout - programmed timeout period elapsed
VSWEND	Voltage switch completed
CKSTOP	SDMMC_CK stopped in voltage switch procedure
SDIOIT	SDIO interrupt received

life.augmented

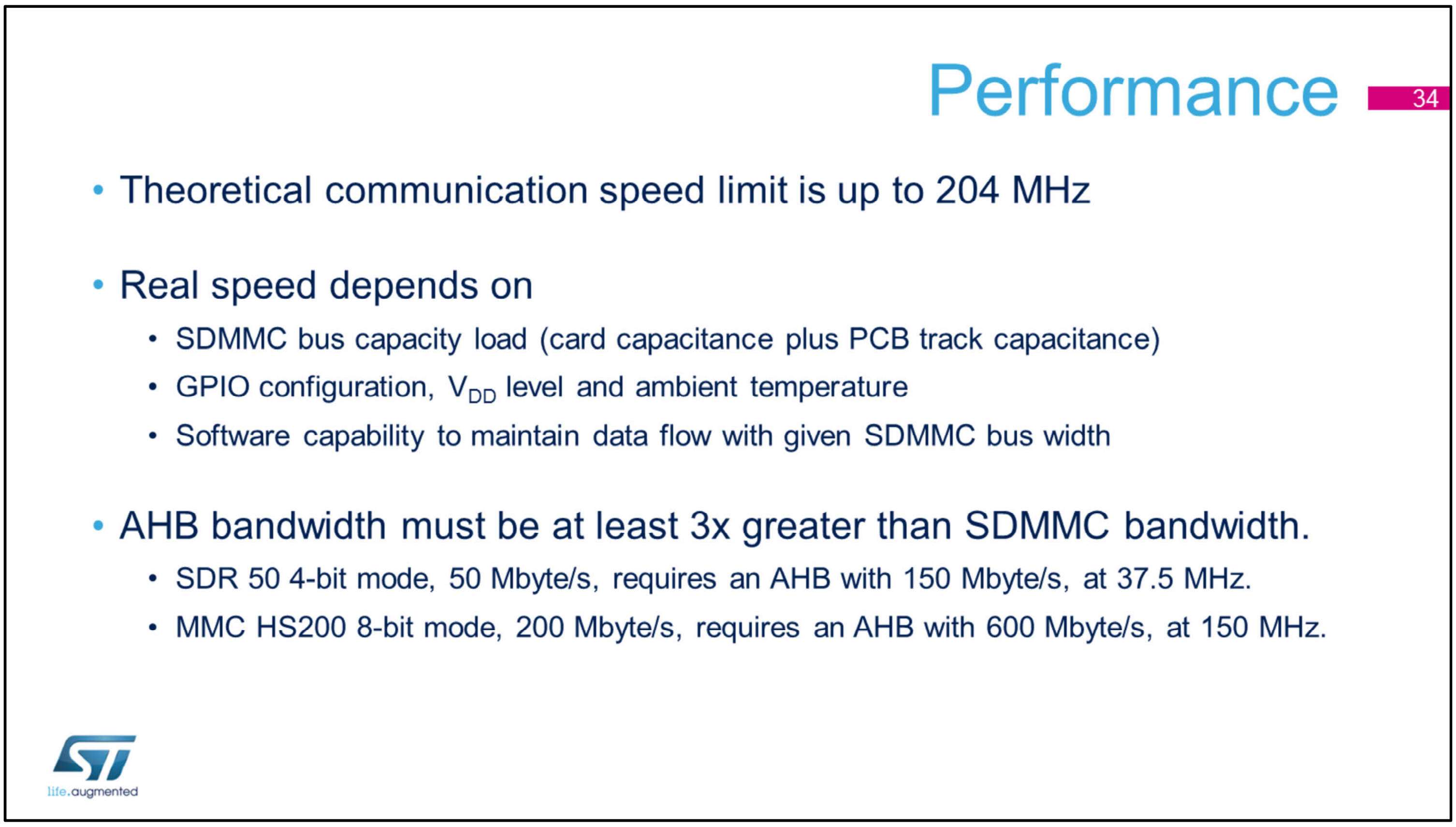
Here is an overview of interrupt events related to IDMA, Boot, Voltage switch and SDIO interrupt.
An IDMA master AHB transfer error is signaled by IDMATE.
Correct transfer of a complete data buffer is signaled by IDMABTC, used with Double Buffer mode.
Boot acknowledgement errors are signaled with ACKFAIL and ACKTIMEOUT.
Voltage switch progress is signaled with VSWEND and CKSTOP.
The SDIO interrupt is signaled with SDIOIT.



Low-power modes

Mode	Description
Run	Active.
Sleep	Active. Peripheral interrupts cause the CPU to exit Sleep mode.
(D)Stop	Frozen. Peripheral registers content is kept.
(D)Standby	Powered-down. The peripheral must be reinitialized after exiting domain and system Standby mode.

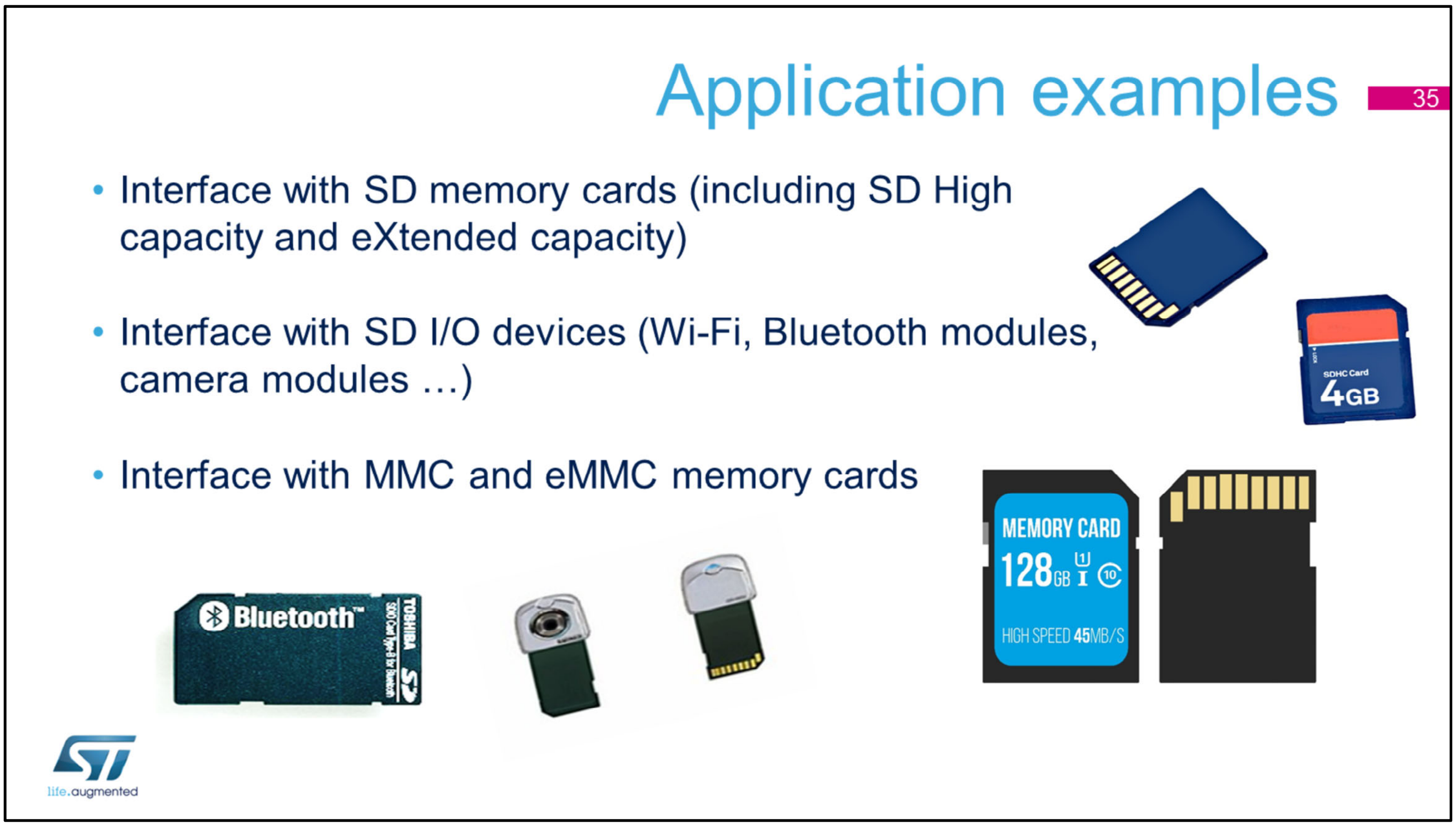
Here is an overview of the peripheral status at specific low-power configuration modes. The device is not able to perform any communication in domain or system Stop mode and lower. It is important to ensure that all transmissions are completed before the SDMMC controller is disabled or the domain or system is switched down to Stop or Standby mode.



Performance

- Theoretical communication speed limit is up to 204 MHz
- Real speed depends on
  - SDMMC bus capacity load (card capacitance plus PCB track capacitance)
  - GPIO configuration, V_{DD} level and ambient temperature
  - Software capability to maintain data flow with given SDMMC bus width
- AHB bandwidth must be at least 3x greater than SDMMC bandwidth.
  - SDR 50 4-bit mode, 50 Mbyte/s, requires an AHB with 150 Mbyte/s, at 37.5 MHz.
  - MMC HS200 8-bit mode, 200 Mbyte/s, requires an AHB with 600 Mbyte/s, at 150 MHz.

Performance depends mainly on the SDMMC bus width and clock configuration. The SDMMC interface can generate clock signals up to 204 MHz. But real speed can be decreased by the application and depends on several factors. The SDMMC bus capacitance has to be considered, as PCB track and card input capacity can play a significant role. GPIO settings also have an effect. Fast GPIO mode should be applied on command, data and clock signals. Lower power supply voltages and extreme ambient temperatures slow down the edges. And in some cases, the application can’t always manage fast data flows, especially due to overly frequent exception servicing or long times spent in interrupt handlers.
The AHB bandwidth must be at least 3x greater than the SDMMC bandwidth.

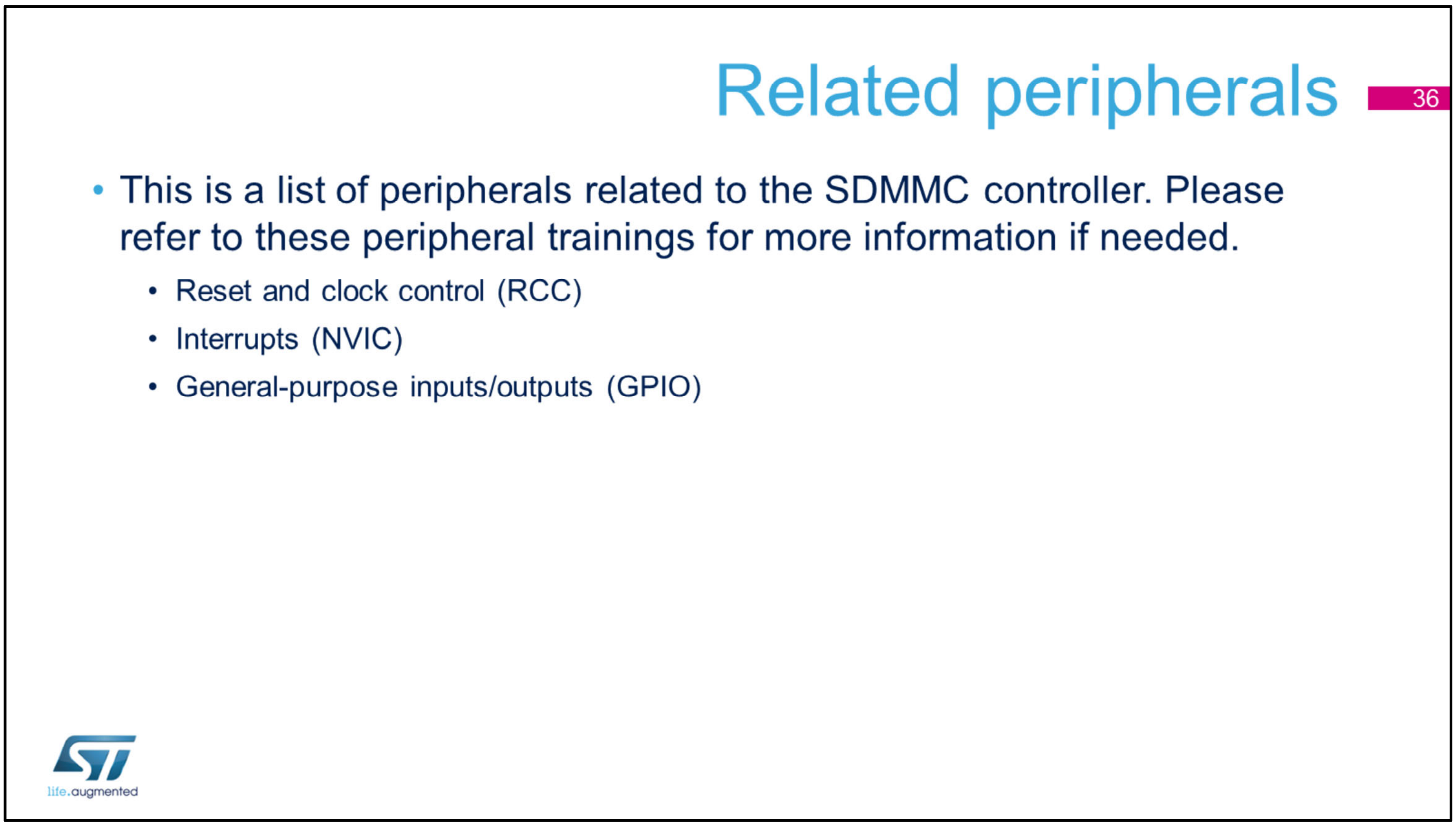


Application examples

- Interface with SD memory cards (including SD High capacity and eXtended capacity)
- Interface with SD I/O devices (Wi-Fi, Bluetooth modules, camera modules …)
- Interface with MMC and eMMC memory cards

life.augmented

The SDMMC interface can be used in a wide range of applications where a low pin count is needed to interface with removable or permanent mass storage data memories.
The SDMMC controller can be used to extend device connectivity when using external SDIO devices (for example, Bluetooth SDIO modules).



Related peripherals

- This is a list of peripherals related to the SDMMC controller. Please refer to these peripheral trainings for more information if needed.
  - Reset and clock control (RCC)
  - Interrupts (NVIC)
  - General-purpose inputs/outputs (GPIO)

Here is a list of peripherals related to the STM32 SDMMC interface. Users should be familiar with all the relationships between these peripherals to correctly configure and use the SDMMC controller.