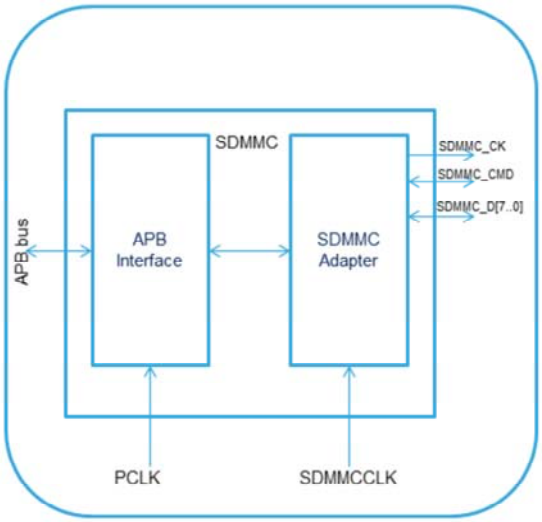
Hello, and welcome to this presentation of the STM32 SDMMC controller module. It covers the main features of the controller which is used to connect the CPU to an SD card, MMC card, or an SDIO device.

Overview



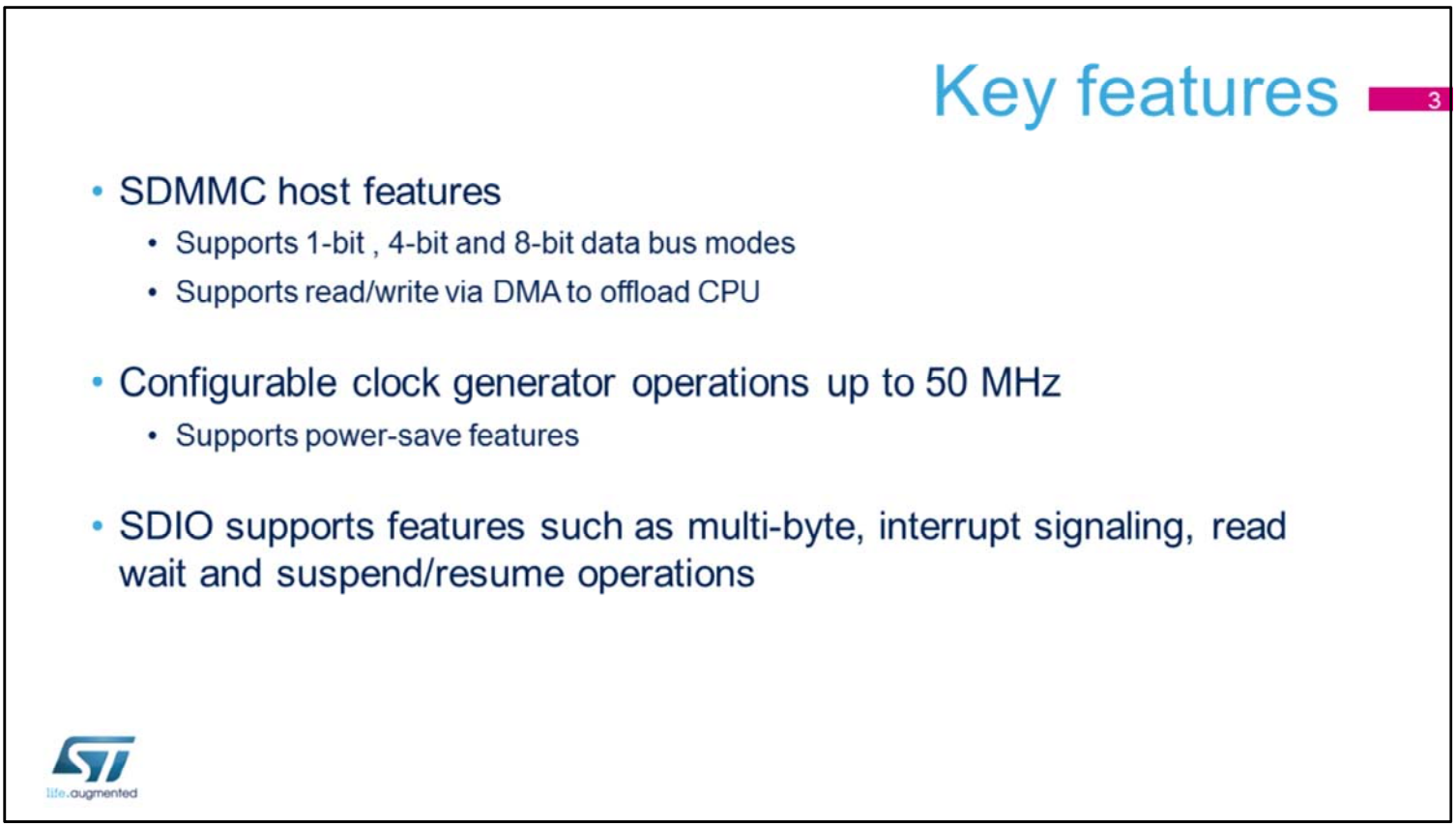
- Provides communication interface with MultiMediaCards (MMC), Secure Digital (SD) memory cards and SD I/O devices (SDIO)
  - Fully configurable
  - Compliant with the SD (2.0), SDIO (2.0) and MMC (4.2) specifications

Application benefits

- Supports both default-speed (< 25 MHz) and high-speed cards and devices (up to 50 MHz)
- Only a few pins needed
- Simple extension of data storage

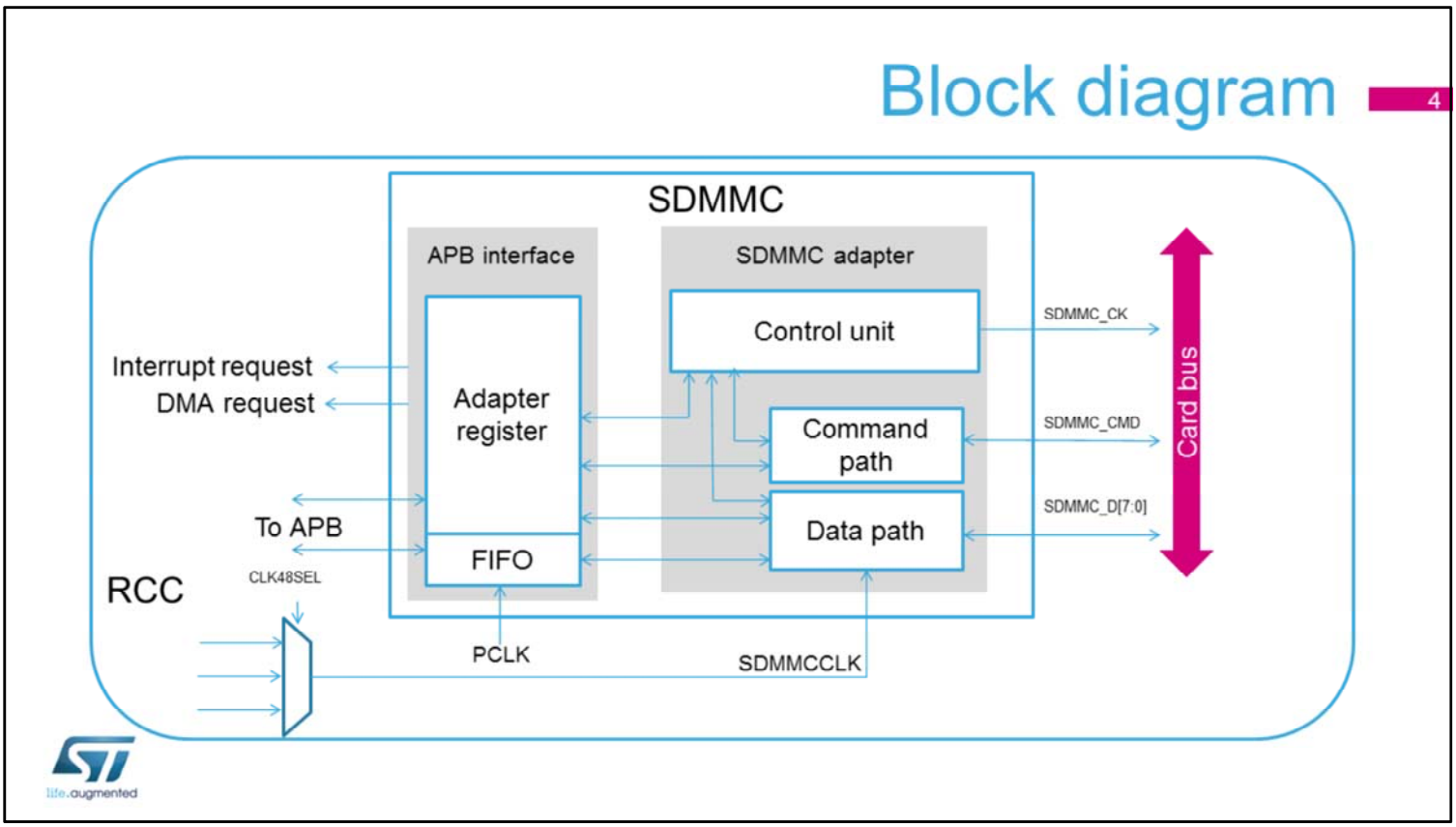
life.augmented

The SDMMC controller integrated inside STM32 products provides a communication interface allowing the microcontroller to communicate with MultiMediaCards, SD memory cards and SDIO devices. This interface is fully configurable, allowing the easy connection of external memories, extending mass storage capability when more memory is needed. Applications benefit from the reduced pin count required to interface with memory cards. Thanks to the SDMMC interface, applications can easily manage high-speed read and write operations in external Flash memories.



The SDMMC controller integrated inside STM32 products 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 DMA to offload the CPU during data read or data write transfer periods. The SDMMC clock generator can generate signals up to 400 kHz for the initialization phase and up to 50 MHz for cards supporting High-speed mode.

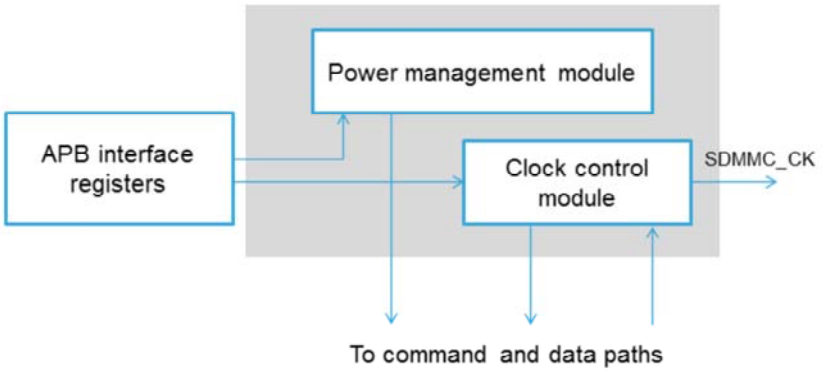
To enhance power consumption, the SDMMC clock can be disabled when the SDMMC command and data buses are idle. The SDMMC controller can interface with SD I/O modules, with advanced features like read wait, suspend/resume operations, and standard operations like multi-byte transfer and interrupt signaling in 1- and 4-bit modes.



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 "APB interface".
The “SDMMC adapter” provides functions such as clock generation, command and data transfer, while the “APB interface” manages the control and status registers, FIFO buffers as well as DMA and interrupt requests.
Two clocks are available for the SDMMC controller, the APB clock (PCLK) for the “APB interface” and the SDMMC clock (SDMMCCLK) for the “SDMMC adapter”.

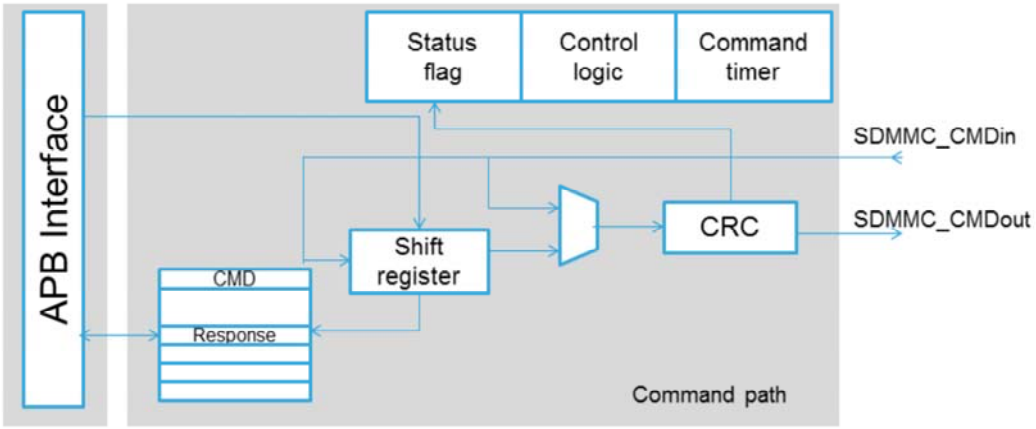
Control unit

- SDMMC_CK clock (up to 50 MHz) is managed by the clock control module
  - SDMMC_CK can use the 8-bit prescaler or the clock bypass mode (card is clocked directly by SDMMCCLK)
  - Powersave mode: SDMMC_CK clock output can be disabled when the bus is idle

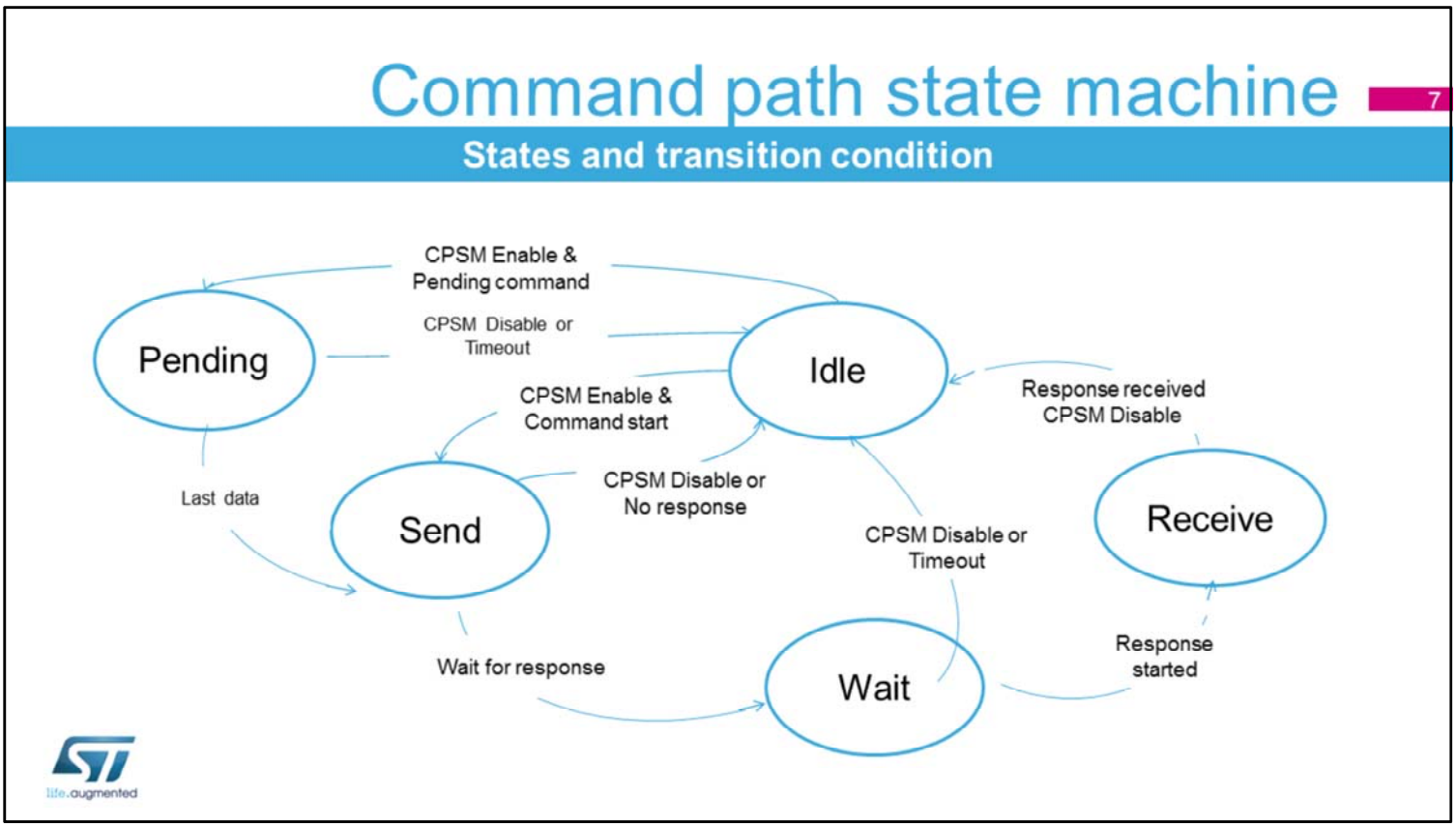


The SDMMC adapter includes a control unit that contains a power management module for power management functions and a clock control module with the clock divider for the card clock (SDMMC_CK).
The clock control module provides an 8-bit prescaler for SDMMC_CK clock generation, which allows it to generate a clock equal to 1/2 SDMMCCLK. It also provides a bypass mode for communications up to 50 MHz.
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.
When a response is expected, the command path is configured to SDMMC_CMDin and waits for the device response.



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.

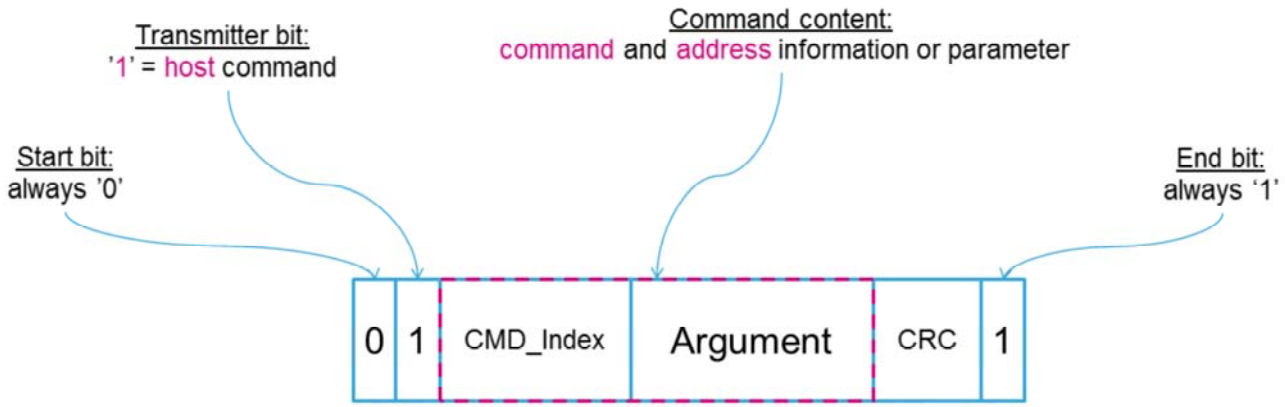
The CPSM returns to Idle state after a timeout if a

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.

Supported commands

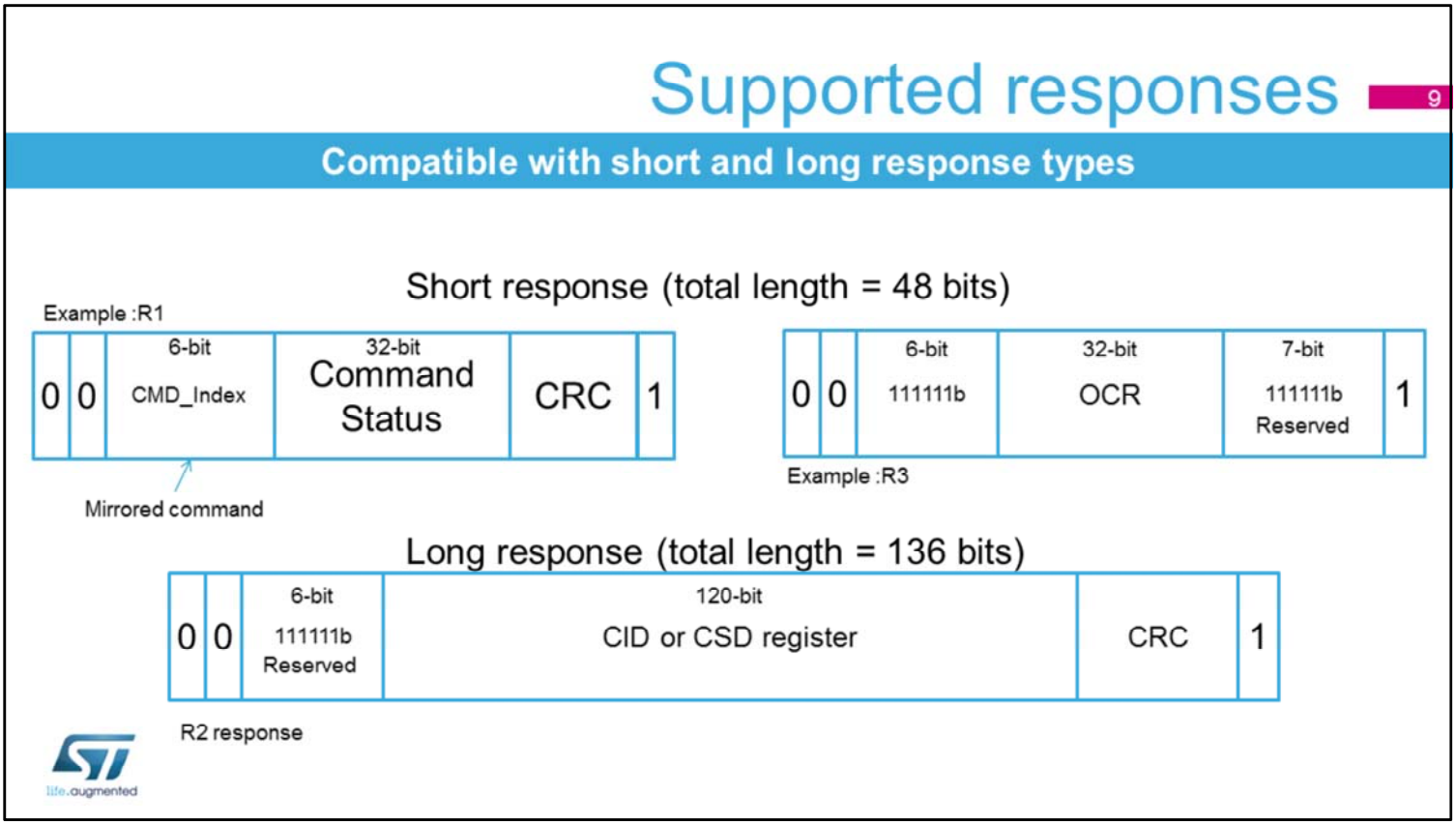
8

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 nor argument. In addition, the start bit, transmitter bit, CRC and end bit fields are automatically generated and sent on the bus.



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 communication with it.]

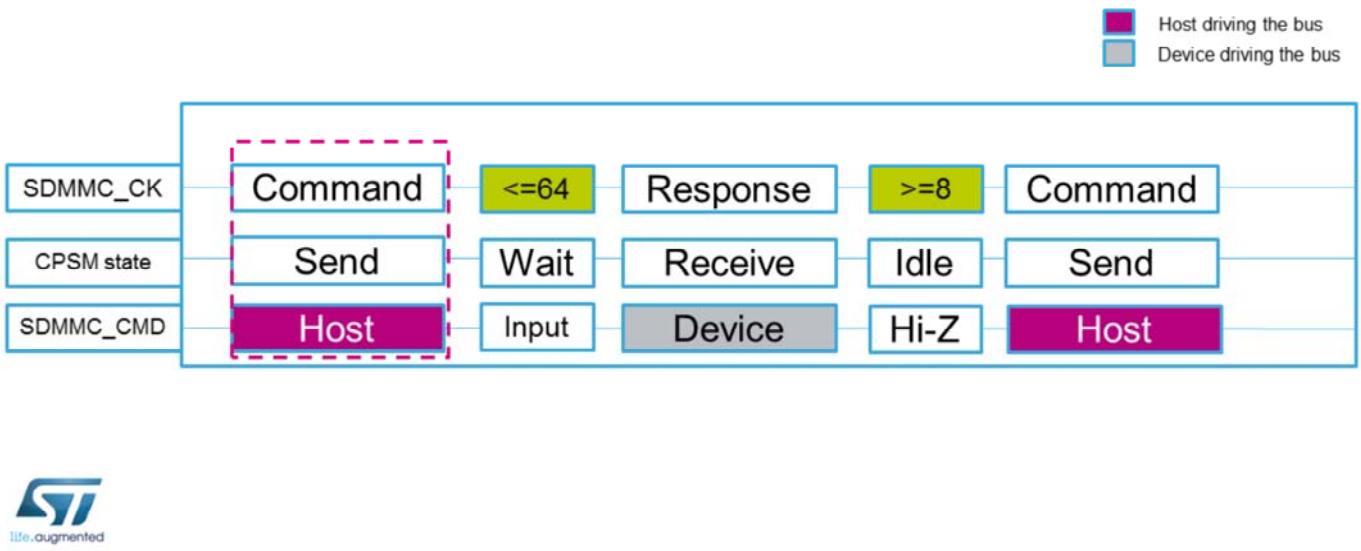
Short responses have a total length of 48 bits, and are composed of a mirrored command index, 32-bit command status, start bit, stop bit and CRC7 checksum. When a short response is received, the command status is saved in the SDMMC_RESP1 register, and the mirrored command index, when available, is copied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

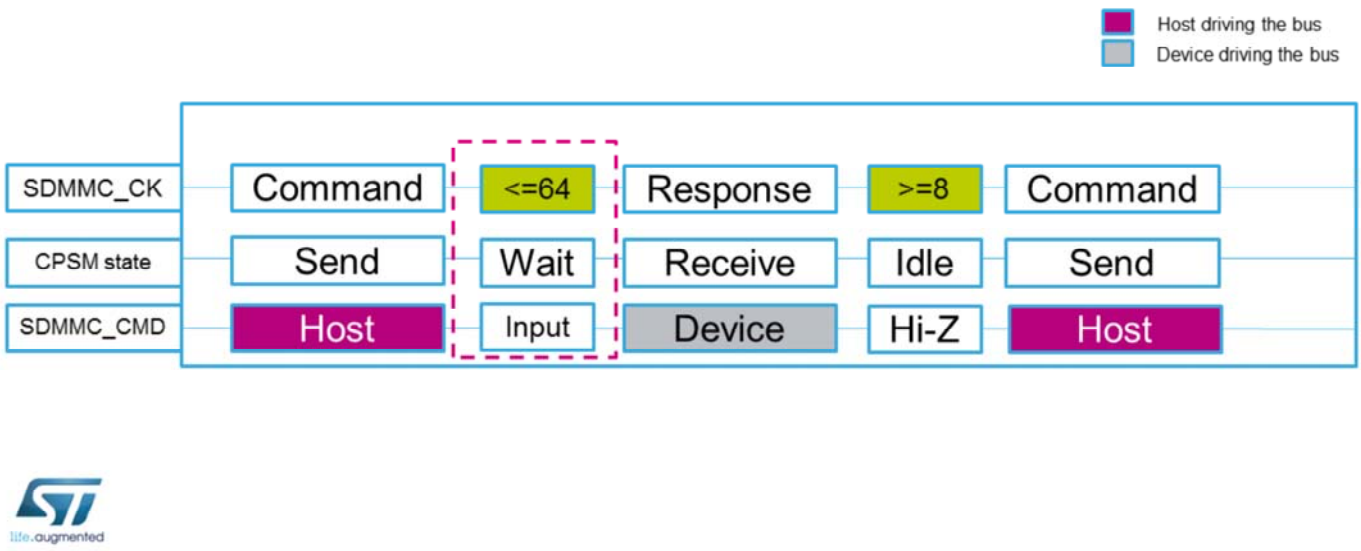
CPSM meets standard timing constraints



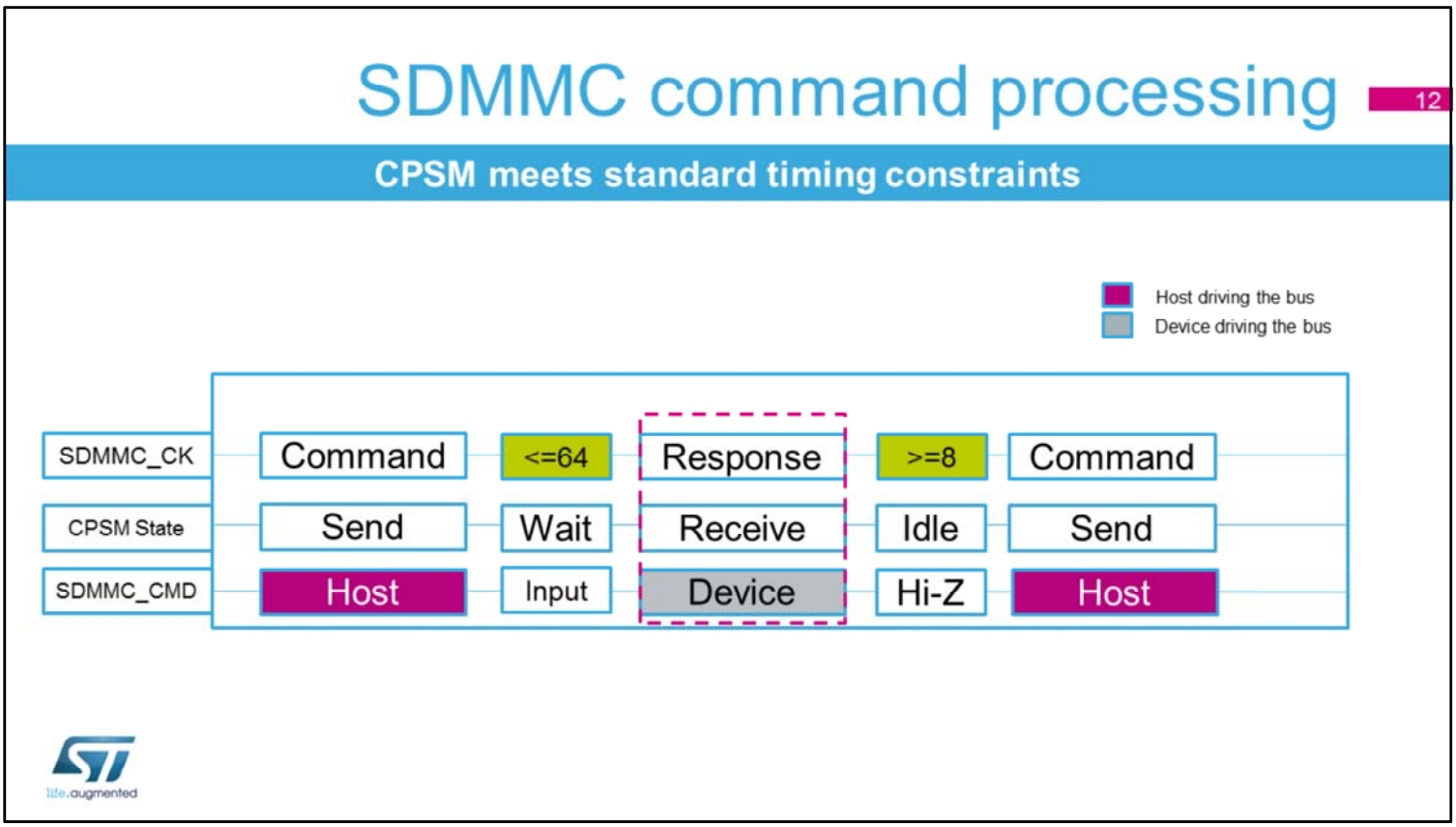
Once the SDMMC_ARG and SDMMC_CMD registers are programmed with CMDINDEX, WAITRESP='01' or '11' and CPSMEN = 1, the CPSM moves from Idle to Send state and the host starts driving the SDMMC_CMD line to send the command to the card.

SDMMC command processing

CPSM meets standard timing constraints



If the CPSM is programmed to wait for a response (WAITRESP=‘01’ or ‘11’), it enters 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.

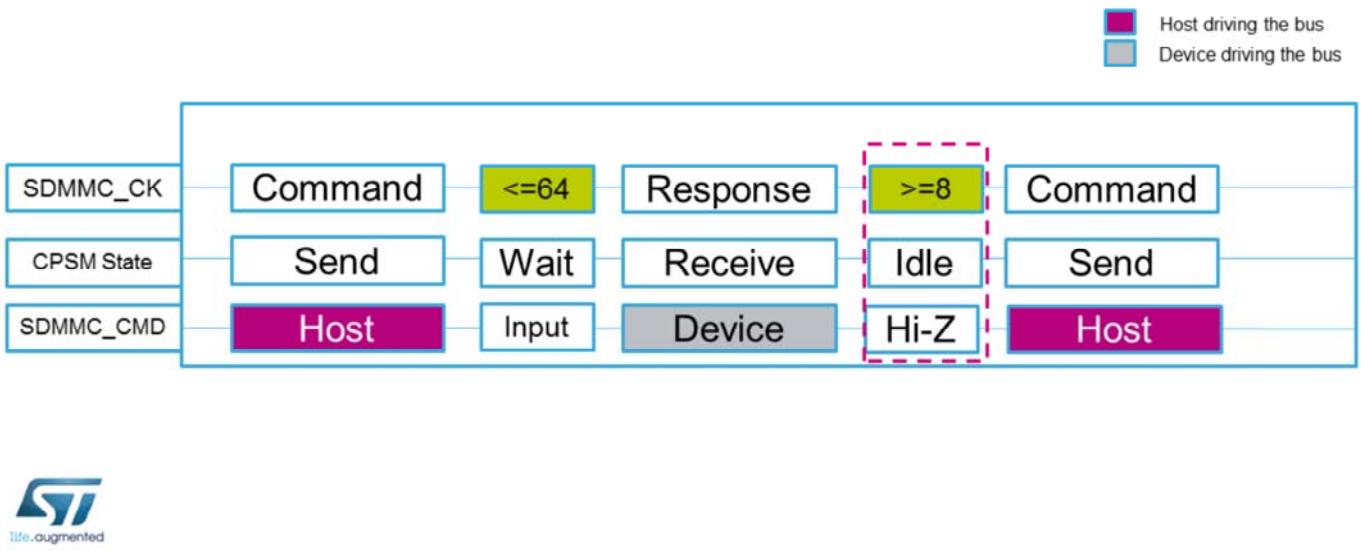


Once a start bit is driven by a device, it is detected on the command line and the CPSM moves to Receive state. When the response is fully received, the received CRC code and the internally-generated checksum code are compared, and the appropriate status flags are set in the SDMMC interface status register.

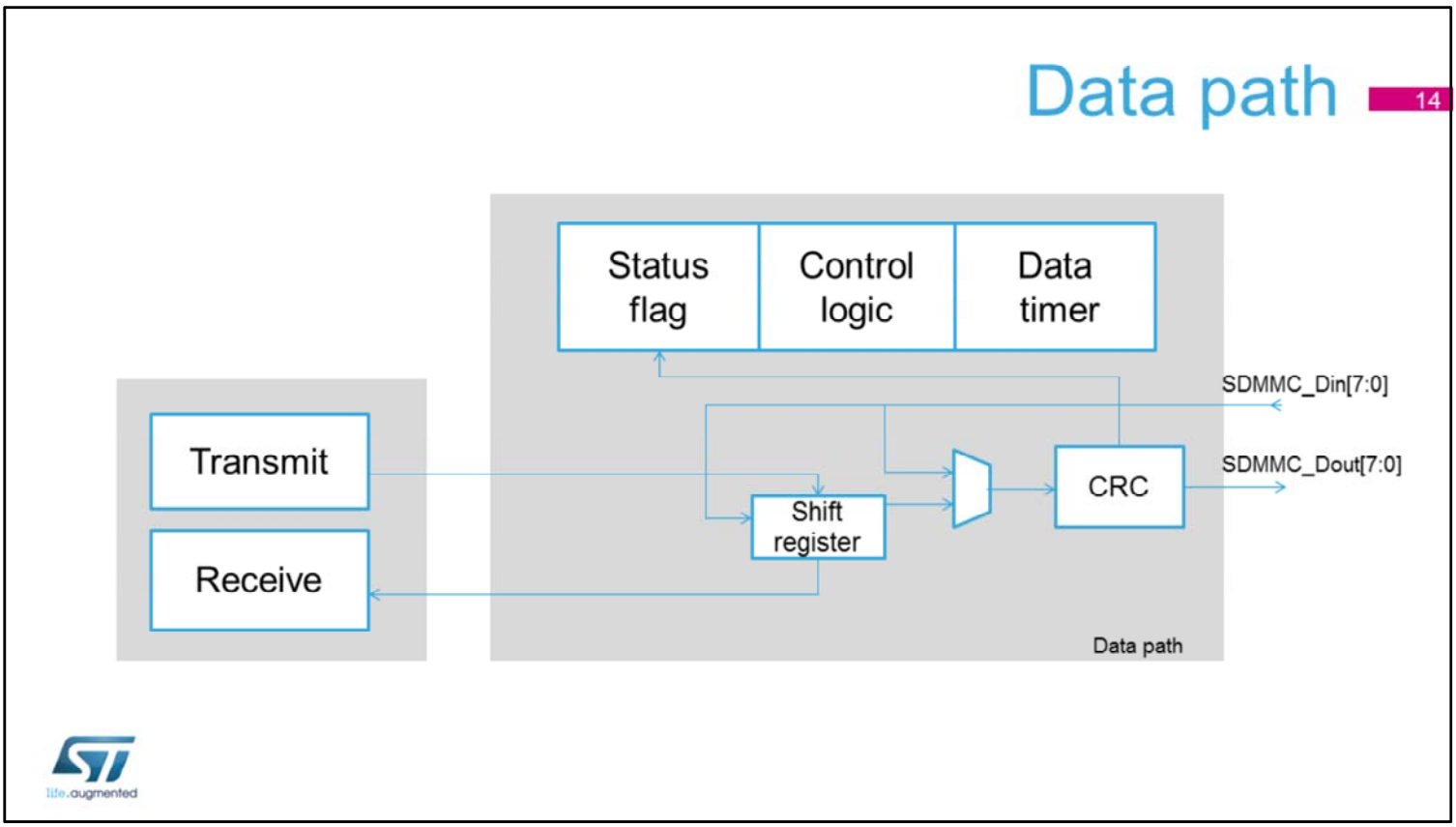
Note that for responses without a CRC, for example in the R3 response format, the SDMMC controller generates a CCRCFAIL flag which means that the command response was received but the CRC check failed.

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 (NRC) timing constraints.



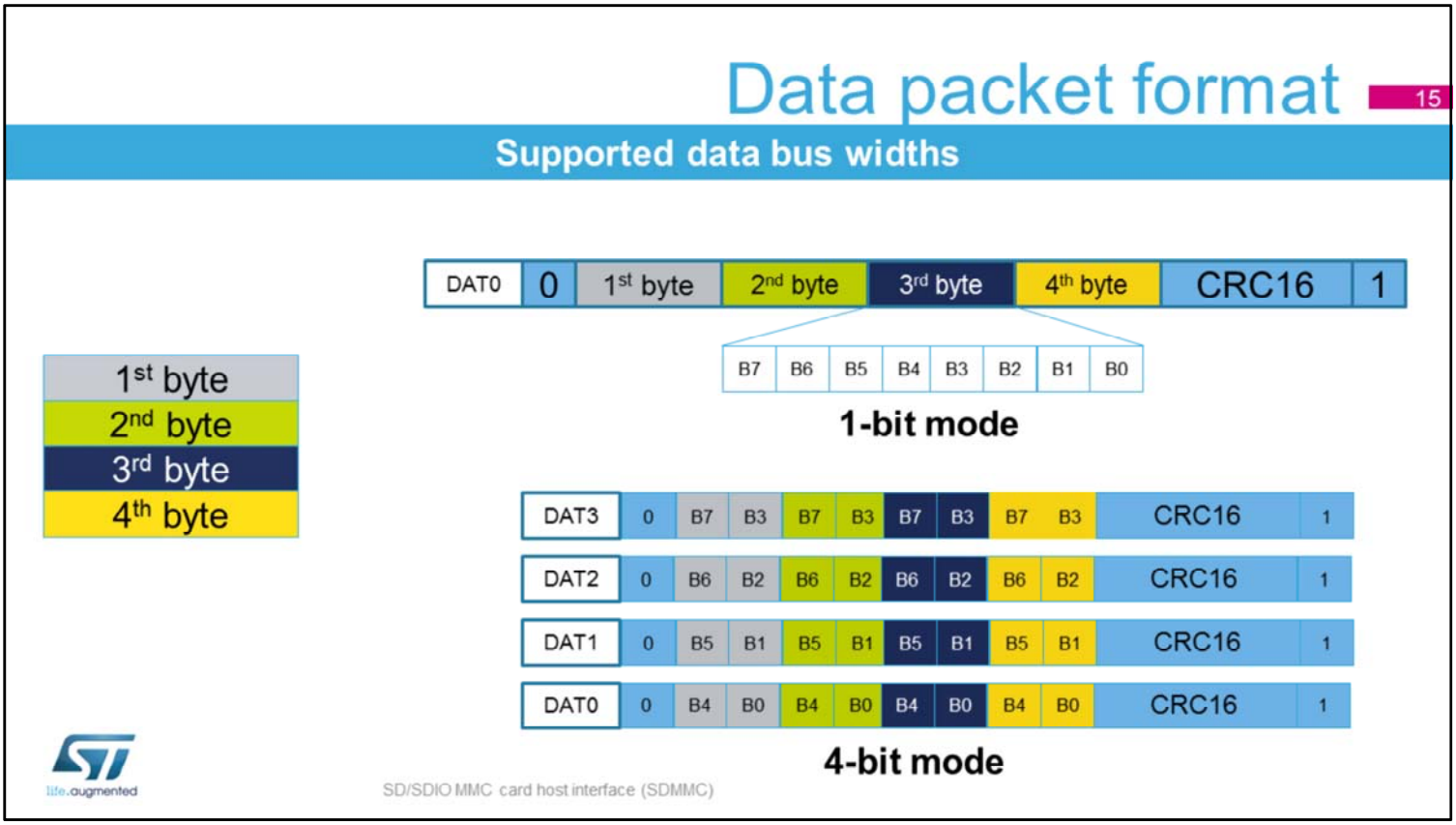
The data path transfers data both to and from the SD/SDIO or MMC card.
On each SDMMC_CK clock cycle, the data path can send one, four or eight bits depending on the bus width configuration.

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 APB 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.



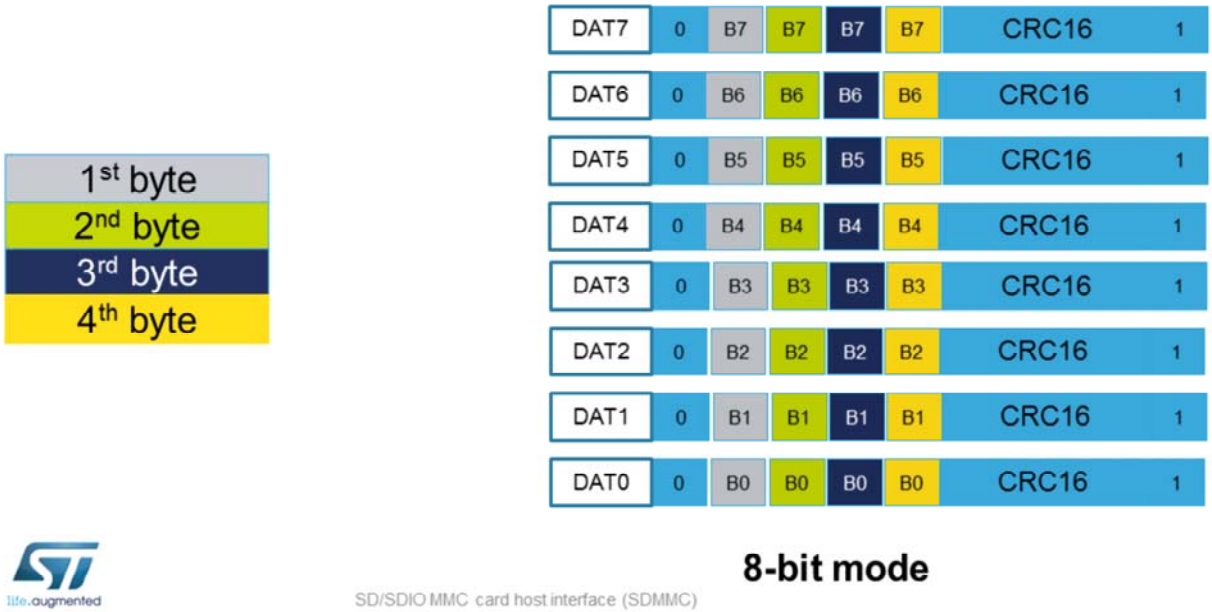
Depending on the configured data bus width, the data path sends data blocks over one (SDMMC_D0), four (SDMMC_D0 to SDMMC_D3), or eight pins (SDMMC_D0 to SDMMC_D7).

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.

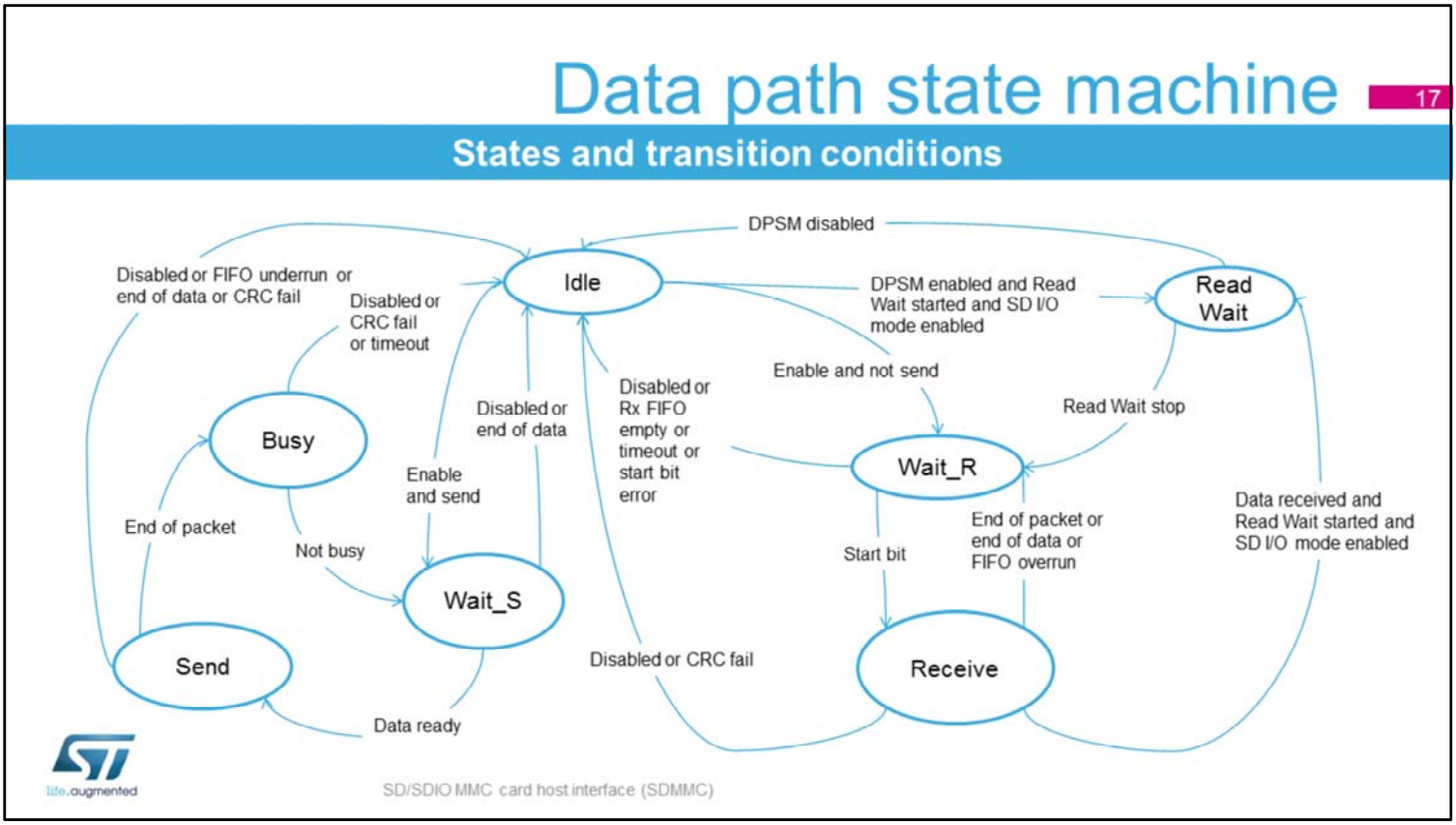
Data packet format

Supported data bus widths



8-bit mode

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.



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.

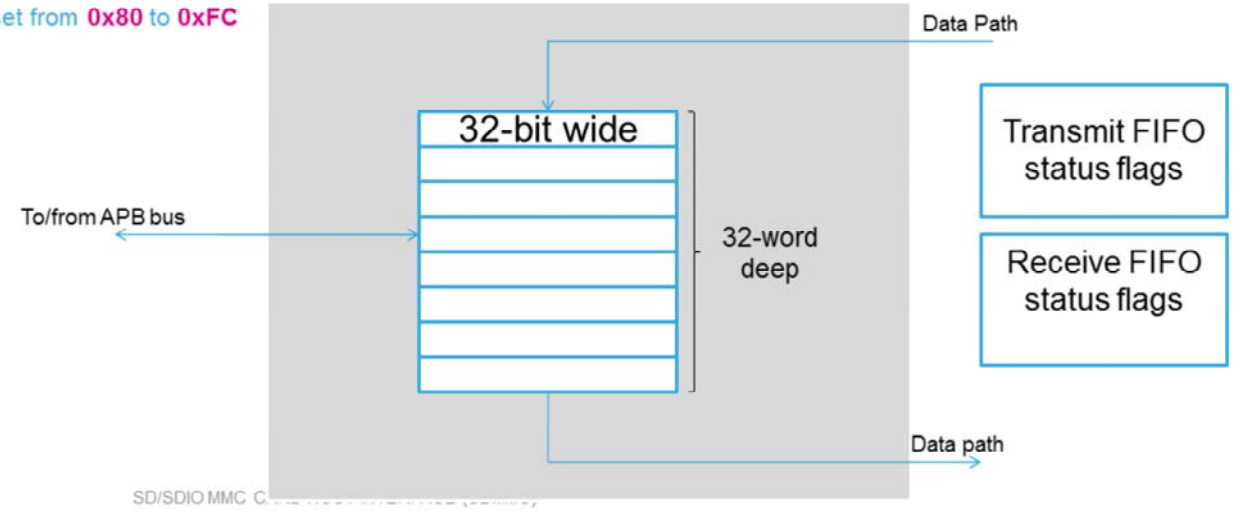
For data transmission, when enabled, the DPSM moves from Idle to Wait_S and then to Send state.
While in the Wait_S state, the DPSM waits until the data FIFO empty flag is de-asserted.
When data is available in the FIFO buffer, the DPSM moves to the Send state.
In Send state, the DPSM starts sending data to a card according to the bus width set in the control register.
At the end of 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 if the SDMMC_D0 pin is not low (meaning that 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.
A negative CRC status from the card or a FIFO underrun error can force the DPSM to return to Idle state.
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.
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.

Data FIFO

Data buffer and access type

- 32-word deep FIFO data buffer memory mapped for CPU burst access (*LDM/STM instruction)*:
  - Offset from **0x80** to **0xFC**

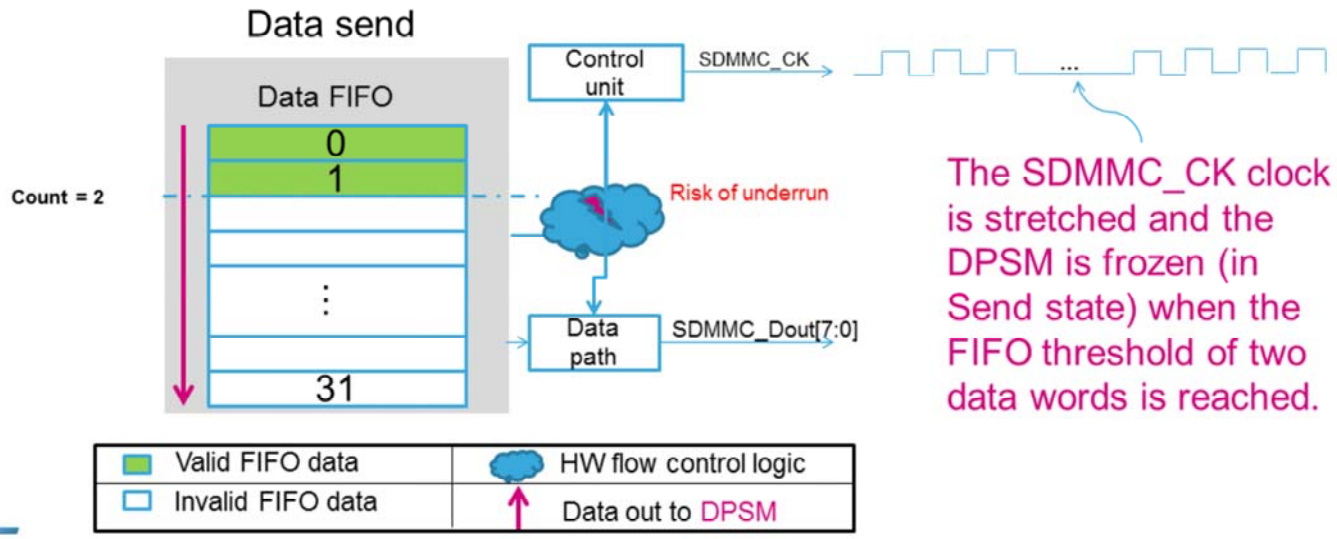


The FIFO is a 32-bit wide, 32-word deep data buffer mapped on the APB domain.
A data FIFO packet is the data source for the data path transmit and receive packets. Depending on the DPSM status, the data path FIFO can be disabled, transmit enabled or receive enabled.

Dedicated receive and transmit FIFO status flags are available to ease firmware implementation. When the data path is disabled, all FIFO flags are de-asserted.

Hardware flow control

- Useful when FIFO accesses are delayed

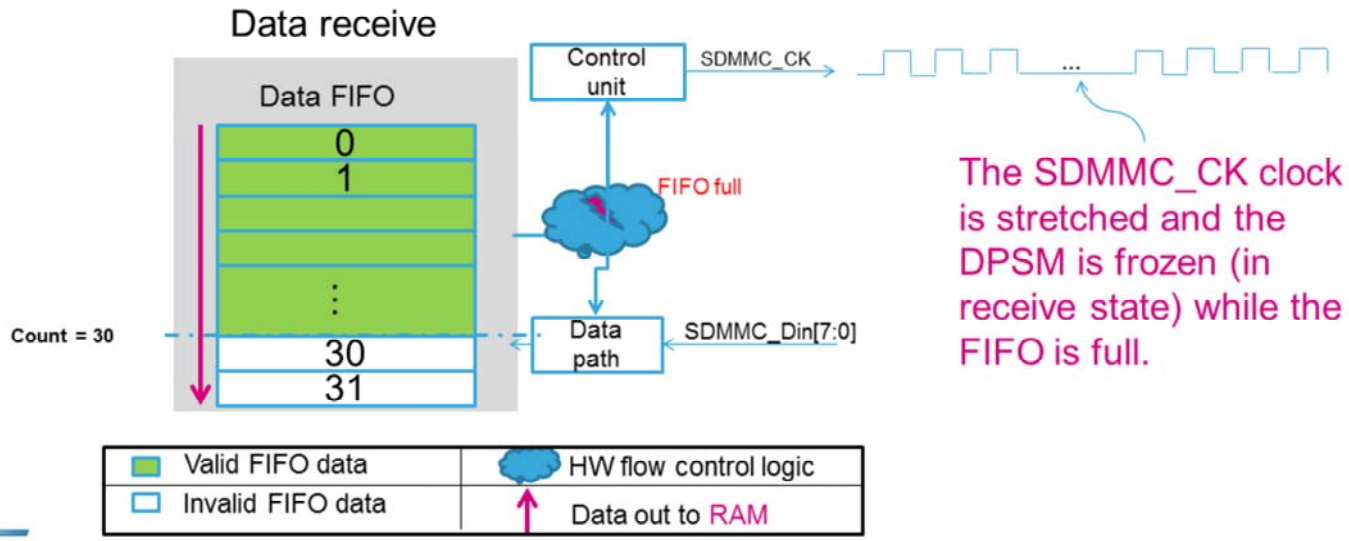


The hardware flow control function is used to avoid FIFO underrun (when DPSM is in TX mode) and overrun (when DPSM is in RX mode) errors. The hardware flow control logic stops the SDMMC_CK pin signals and freezes the DPSM when a risk of underrun/ overrun is detected.

In Send state, the SDMMC_CK pin clock signal is stretched and the DPSM is frozen when the FIFO threshold of two data words is reached.

Hardware flow control

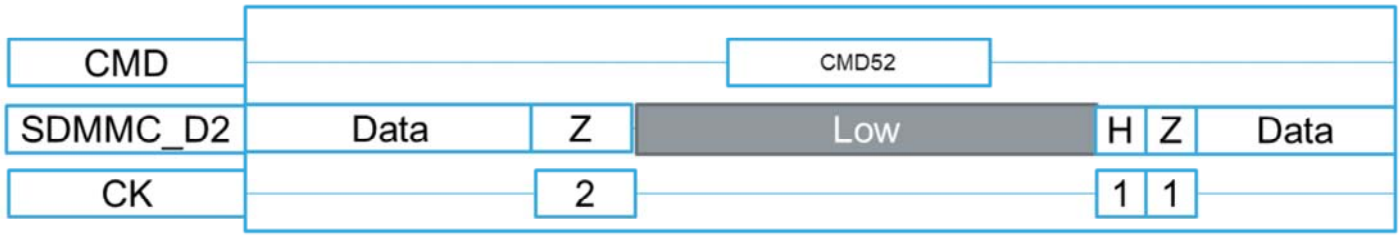
- Useful when FIFO accesses are delayed



In Receive state, the SDMMC_CK clock is stretched and the DPSM is frozen in Receive state while the FIFO is full (threshold is 30 words). The clock and DPSM are restarted when the FIFO Full flag is de-asserted.

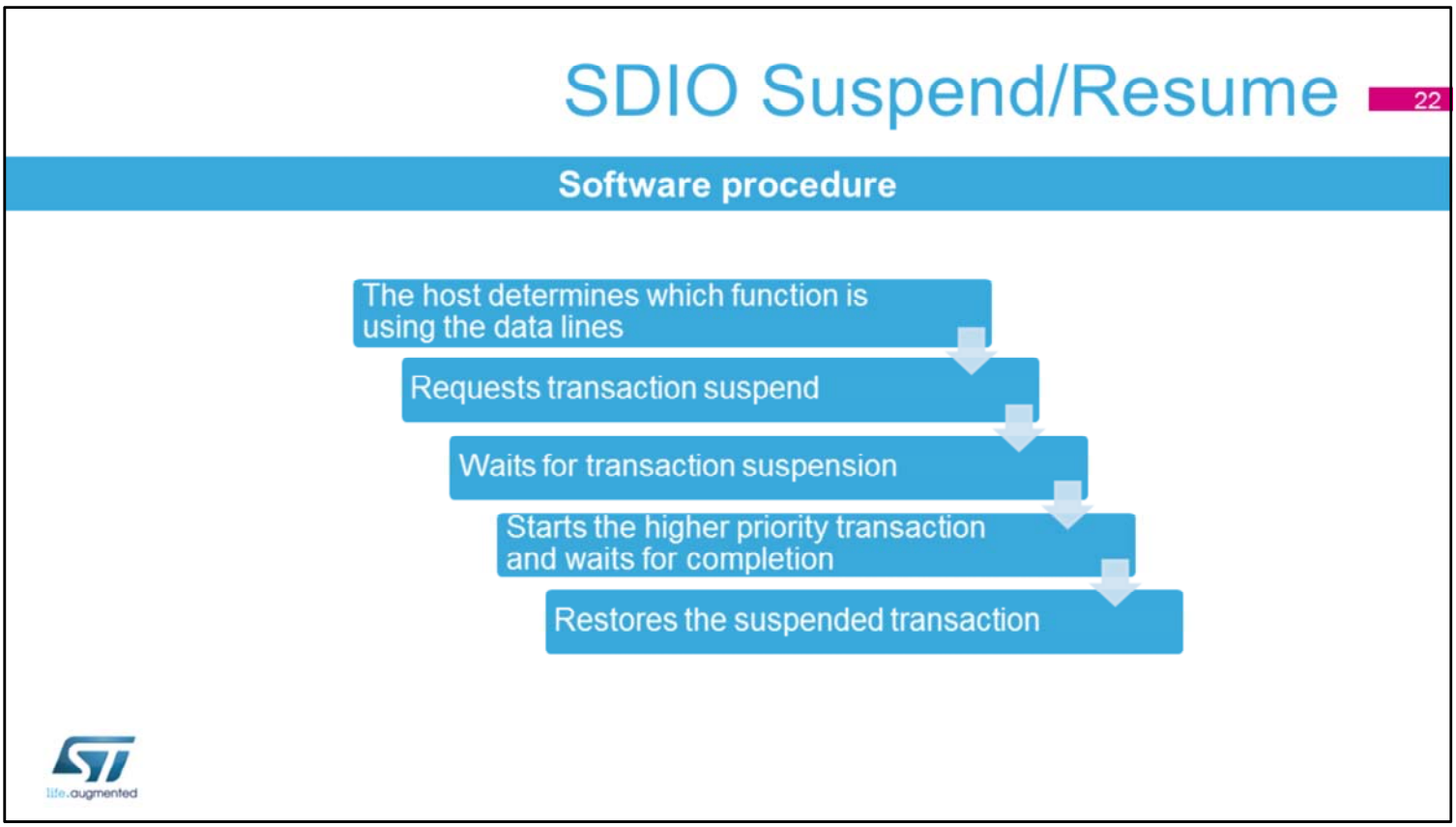
SDIO Read Wait

- The SDMMC host supports two Read Wait modes:
  - Stopping the SDMMC_CK
  - Using SDMMC_D2



Concept: The Read Wait operation is an SDIO-specific operation that allows the host to temporarily stall the data transfer while emptying its data buffer or 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.

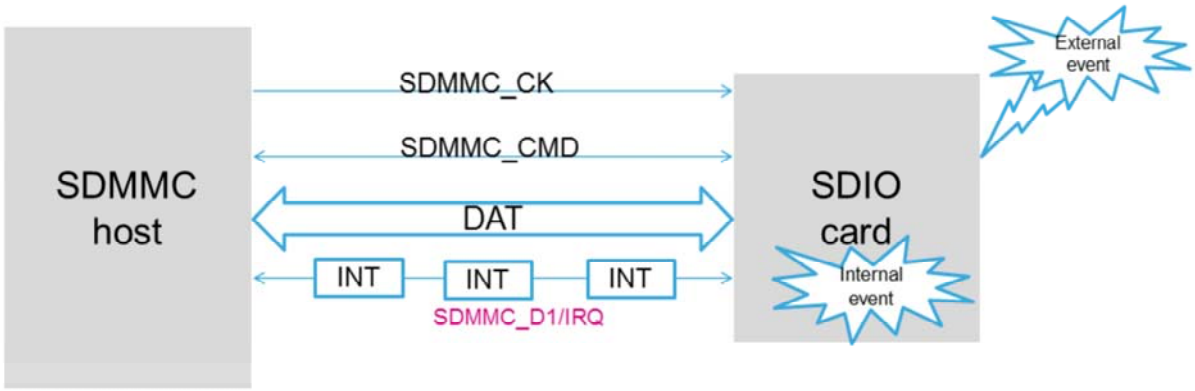


Concept: With multi-function cards, there are multiple devices that share the access to the SD bus. When the function supports Suspend/Resume, the host can temporarily halt data transfers to perform other internal operations or to communicate with other functions and then resume the suspended transaction.
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.
When the SDMMC_CMD bit is set to ‘11’, the CPSM knows that the current command is a Suspend command.
If a Suspend request is accepted, the DPSM will wait as the function sends a complete packet and application empties the reception FIFO before going to the 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.

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.
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, all low levels on the SDMMC_D1 pin are detected as interrupts from the card to the host.

Interrupts and DMA

Interrupt event	Description
CCRCFAIL	Command response received but CRC check failed
CTIMEOUT	Command response timeout after Timeout period of 64 SDMMC_CK
CMDSENT	Command sent and no response required
CMDREND	Command response received and CRC check passed
CMDACT	Command transfer in progress (CPSM active)

Here is an overview of interrupt events. This slide shows events related to the command path state machine.

Interrupts and DMA

Interrupt event	Description
DCRCFAIL	Data block sent/received and CRC check failed
DTIMEOUT	Data timeout - programmed timeout period elapsed
TXUNDERR	Transmit FIFO underrun error
RXOVERR	Receive FIFO overrun error
DBCKEND	Data block sent/received and CRC check passed
TXACT	Data transmit in progress (DPSM active)
RXACT	Data receive in progress (DPSM active)

This is the list of flags for the data path state machine with events related to the transfer direction and transfer status.

Interrupts and DMA

Interrupt event	Description
TXFIFOHE	Transmit FIFO half empty. At least 8 words can be written into the FIFO
RXFIFOHF	Receive FIFO half full. There are at least 8 words in the FIFO
TXFIFOF	Transmit FIFO full
RXFIFOF	Receive FIFO full
TXFIFOE	Transmit FIFO empty
RXFIFOE	Receive FIFO empty
TXDAVL	Data available in transmit FIFO
RXDAVL	Data available in receive FIFO

life.augmented

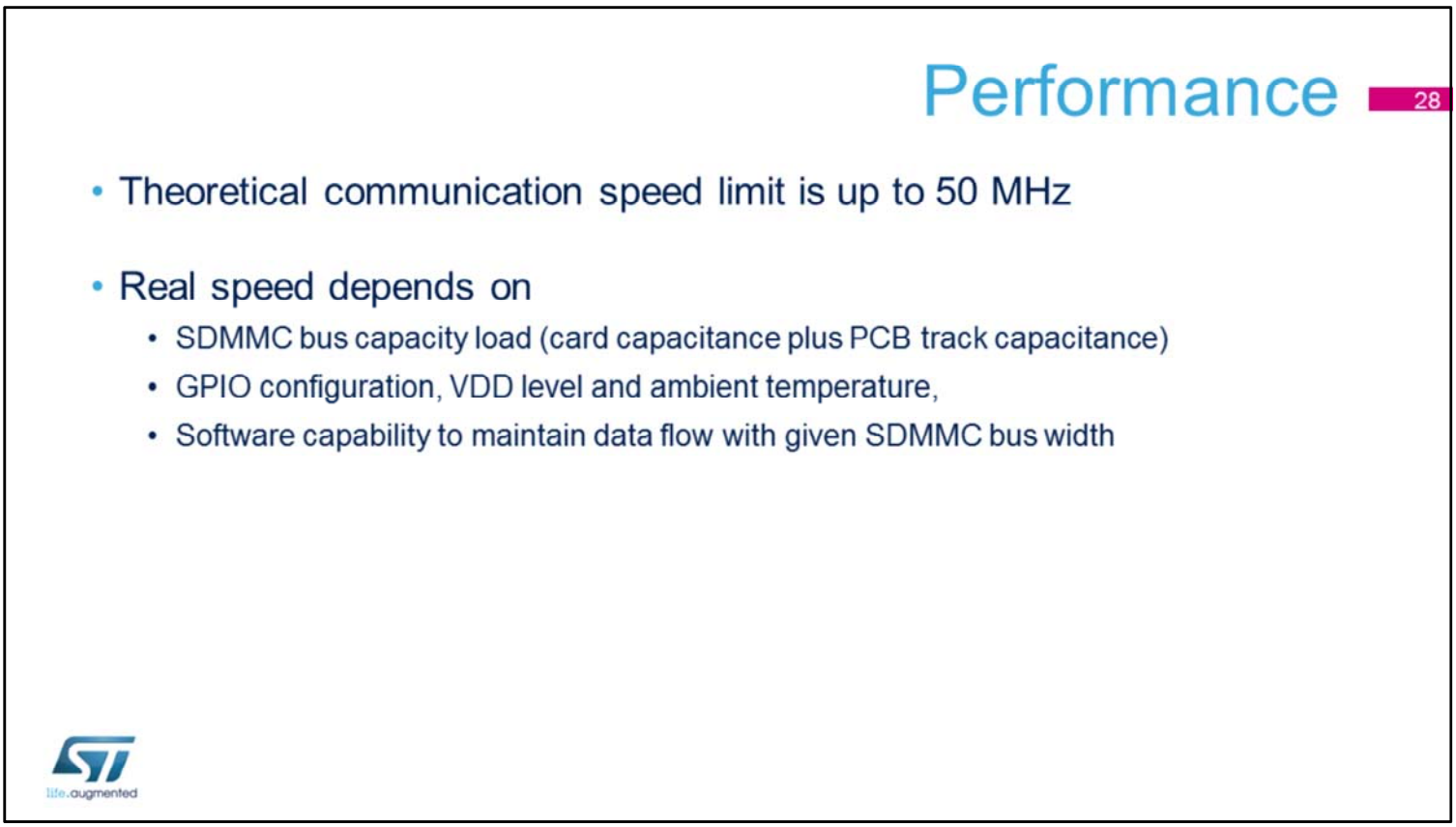
- DMA requests available on data Transmit and Receive transfers.

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.

Low-power modes

Mode	Description
Run	Active.
Sleep	Active. Peripheral interrupts cause the device to exit Sleep mode.
Low-power run	Active.
Low-power sleep	Active. Peripheral interrupts cause the device to exit Low-power sleep mode.
Stop 0/Stop 1	Frozen. Peripheral registers content is kept.
Stop 2	Frozen. Peripheral registers content is kept.
Standby	Powered-down. The peripheral must be reinitialized after exiting Standby mode.
Shutdown	Powered-down. The peripheral must be reinitialized after exiting Shutdown 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 Stop mode and lower. It is important to ensure that all transmissions are completed before the SDMMC controller is disabled or the system is switched down to stop.



Performance depends mainly on the SDMMC bus width and clock configuration. The SDMMC interface can generate clock signals up to 50 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.

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
  - Direct memory access controller (DMA)
  - 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.