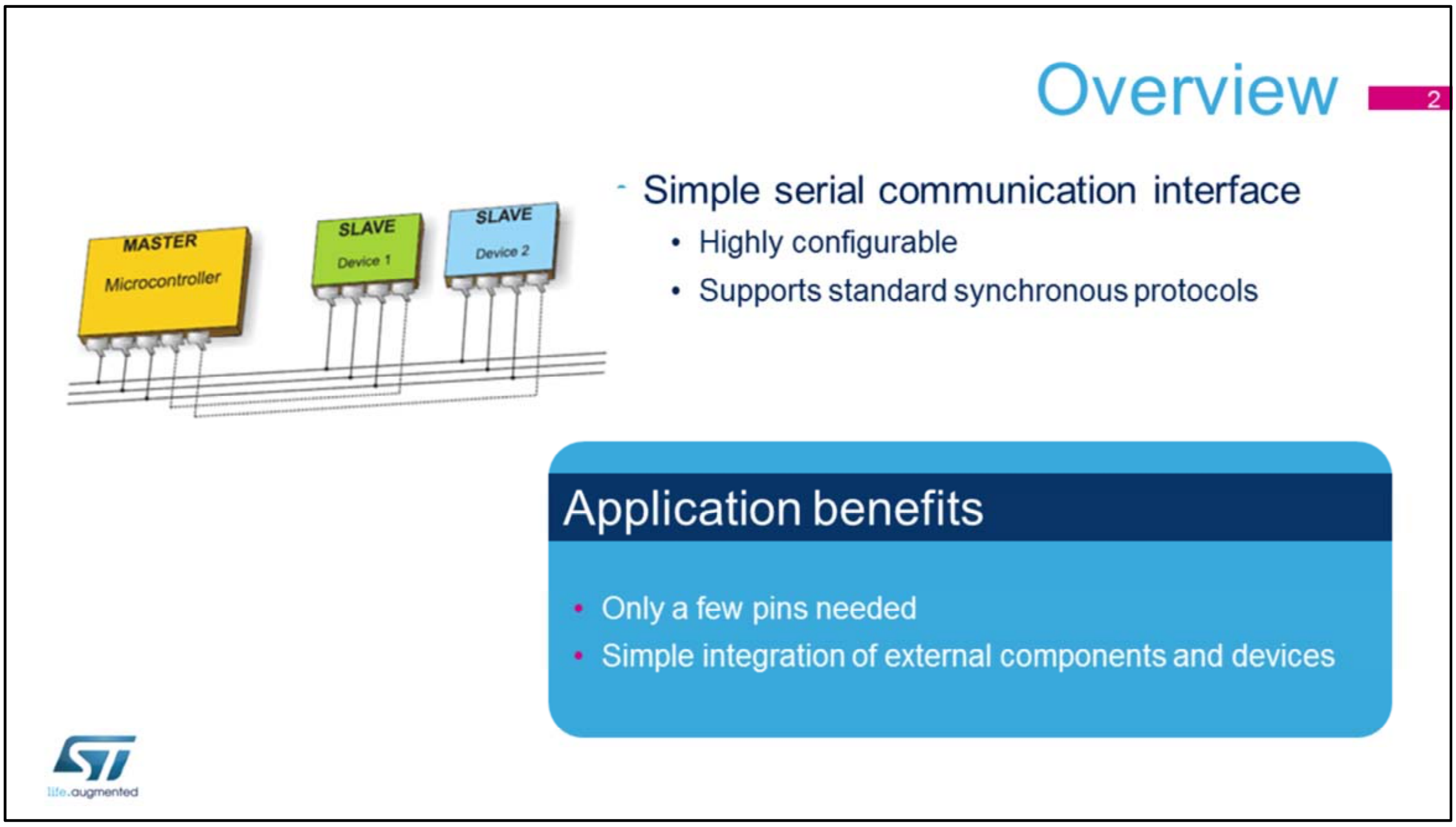
STM32L4 - SPI

Serial Peripheral Interface

Revision 2.0

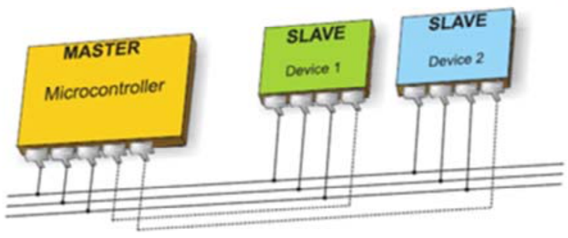
life.augmented

Hello, and welcome to this presentation of the STM32 Serial Peripheral Interface or SPI .



Overview

- Simple serial communication interface
  - Highly configurable
  - Supports standard synchronous protocols

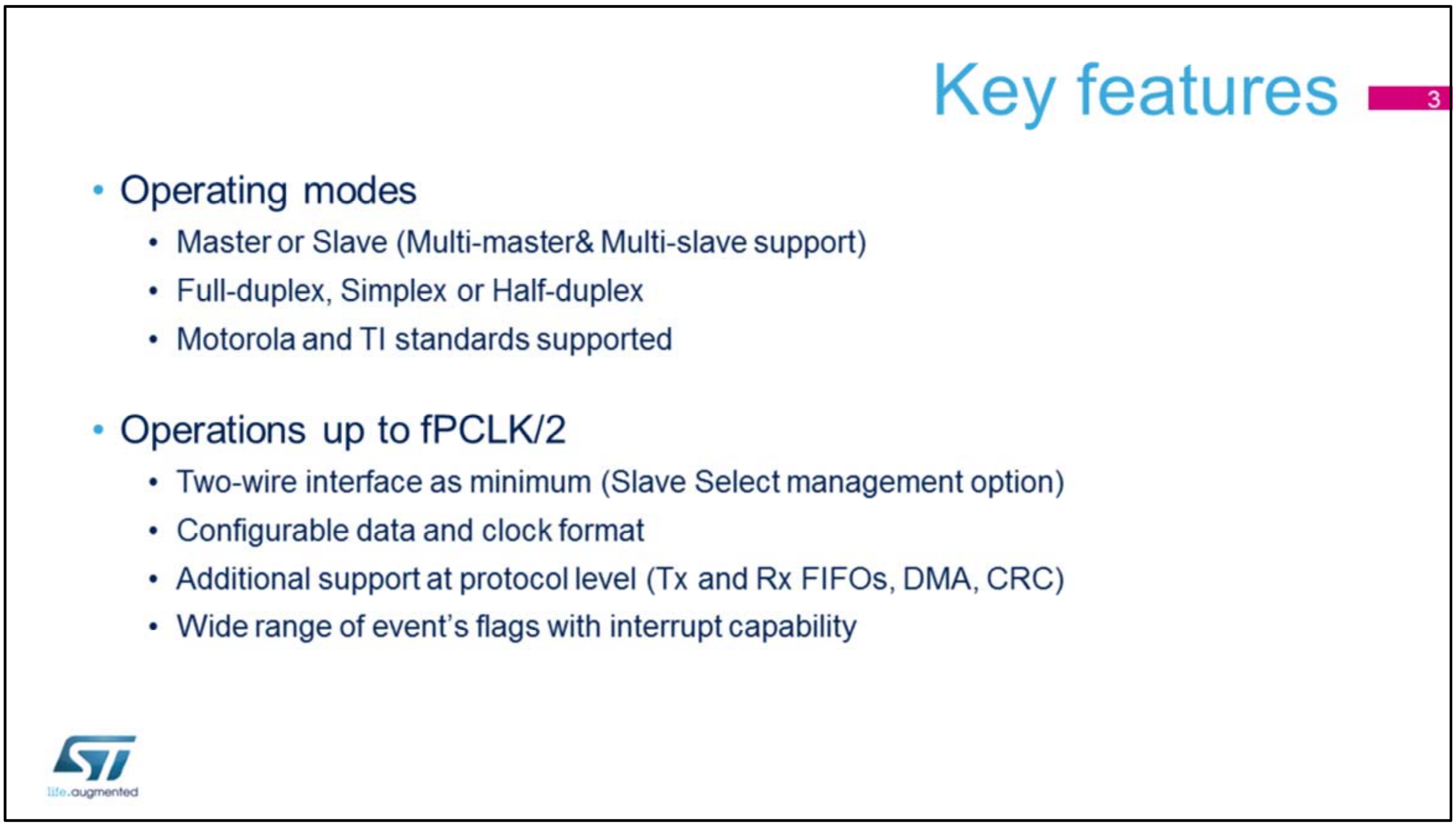


Application benefits

- Only a few pins needed
- Simple integration of external components and devices

life.augmented

The internal Serial Peripheral Interface provides a simple communication interface, allowing the microcontroller to communicate with nearby external devices. This interface is highly configurable to support many synchronous standard communication protocols.
Applications benefit from the simple and direct connection to components which only requires a few pins. Thanks to the highly configurable capabilities of the peripheral, many devices can be simply accommodated in the existing project.



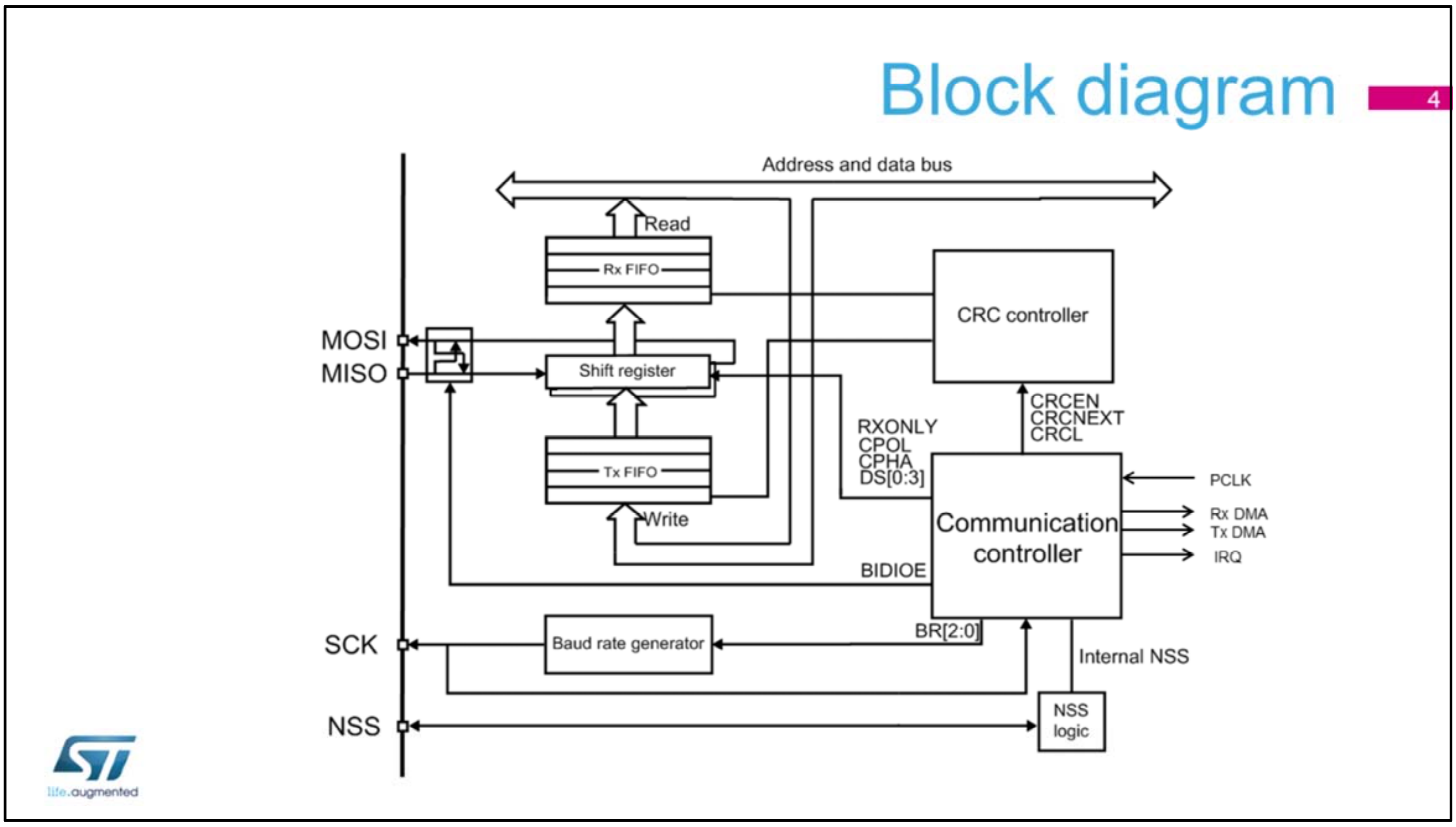
Key features

3

- Operating modes
  - Master or Slave (Multi-master& Multi-slave support)
  - Full-duplex, Simplex or Half-duplex
  - Motorola and TI standards supported
- Operations up to fPCLK/2
  - Two-wire interface as minimum (Slave Select management option)
  - Configurable data and clock format
  - Additional support at protocol level (Tx and Rx FIFOs, DMA, CRC)
  - Wide range of event's flags with interrupt capability

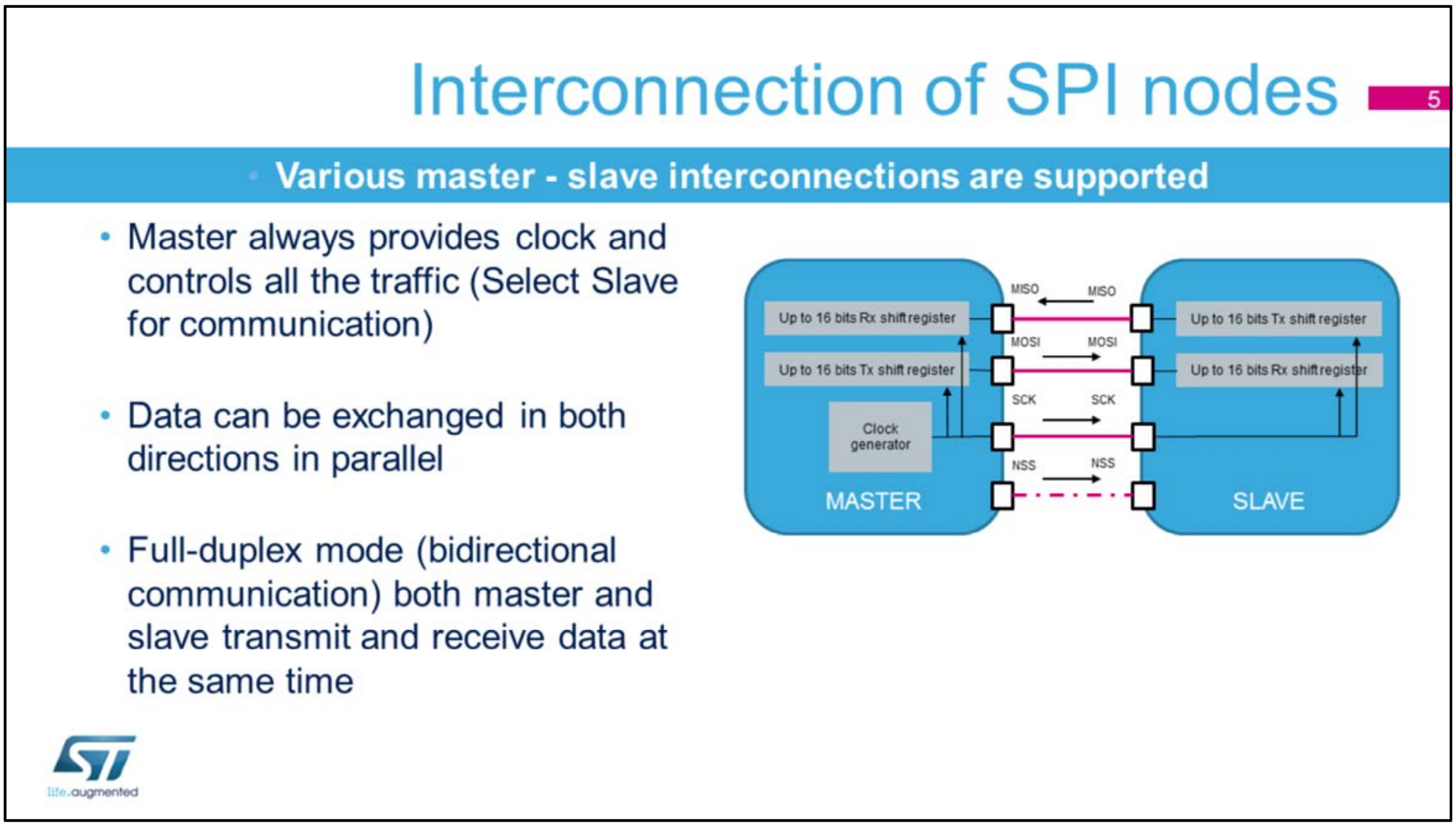
life.augmented

The STM32 SPI offers various operating modes which will be explained in more detail in this presentation.
The communication speed cannot exceed half of the internal bus frequency, and a minimum of two wires is required to provide the serial data flow synchronized by clock signal in a single direction at one time. An optional hardware Slave Select control signal can be added. The data size and transmit shift order are configurable, as well as the clock signal's polarity and phase. At the protocol level, the user can use specific data buffers with an optional automatic cyclic redundancy check or CRC calculation, and transfer data through the direct memory access or DMA .There are a wide range of SPI events that can generate interrupt requests.



Block diagram

The simplified SPI block diagram shows its basic control mechanisms and functions. There are four I/O signals associated with the SPI peripheral. All of the data passes through receive and transmit buffers via their specific interfaces. The control block features are enabled or disabled depending on the configuration.

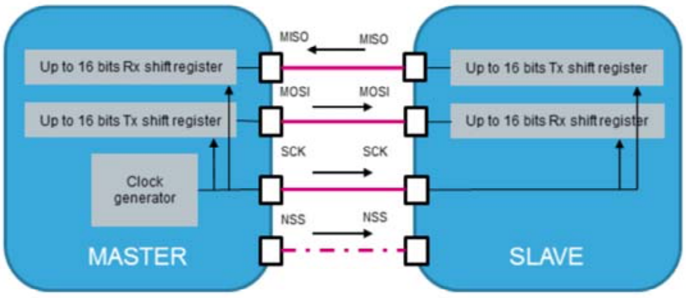


Interconnection of SPI nodes

5

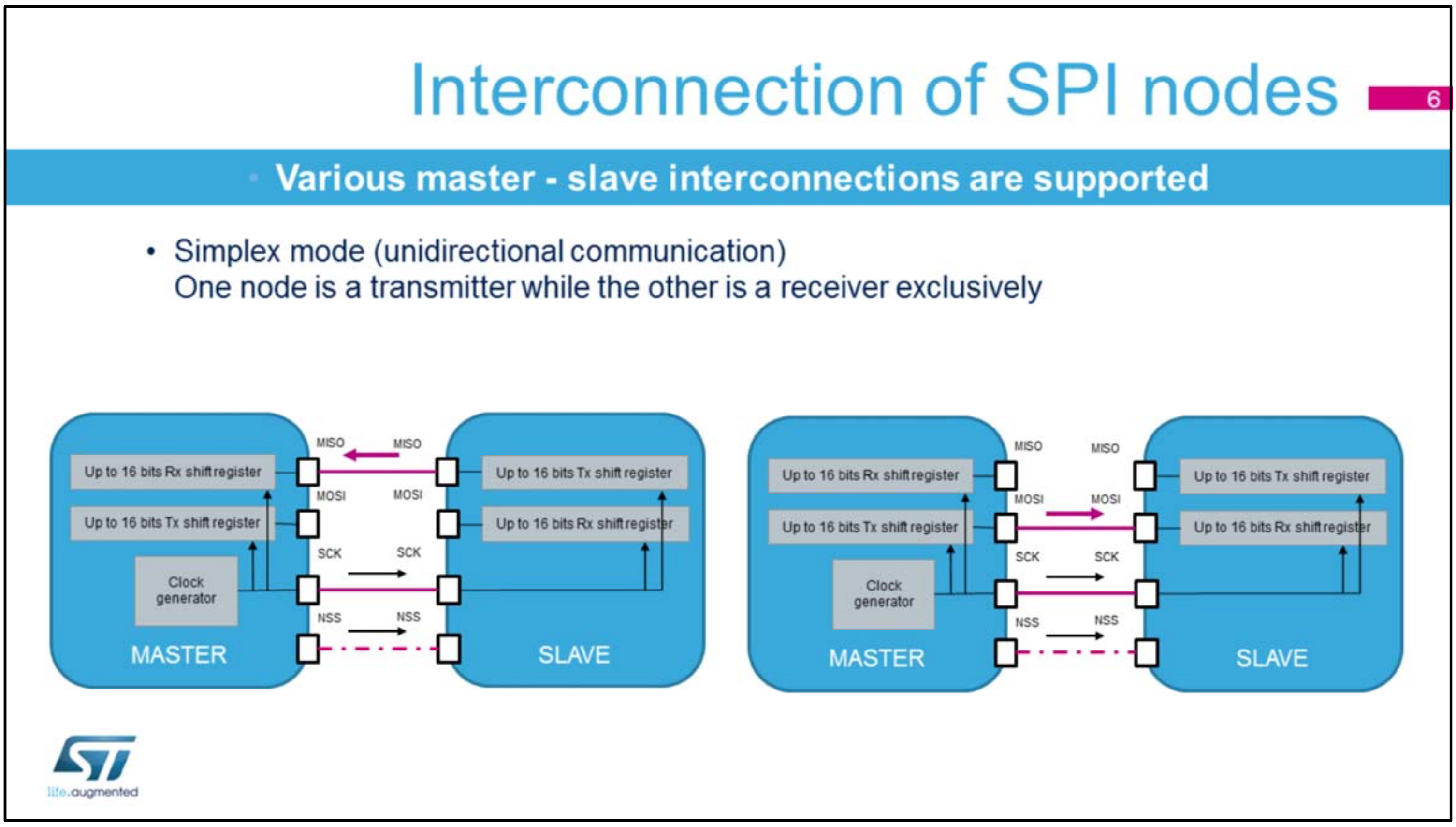
Various master - slave interconnections are supported

- Master always provides clock and controls all the traffic (Select Slave for communication)
- Data can be exchanged in both directions in parallel
- Full-duplex mode (bidirectional communication) both master and slave transmit and receive data at the same time


Up to 16 bits Rx shift register
Up to 16 bits Tx shift register
Clock generator
MASTER
MISO
MOSI
SCK
NSS
MISO
MOSI
SCK
NSS
Up to 16 bits Tx shift register
Up to 16 bits Rx shift register
SLAVE

life.augmented

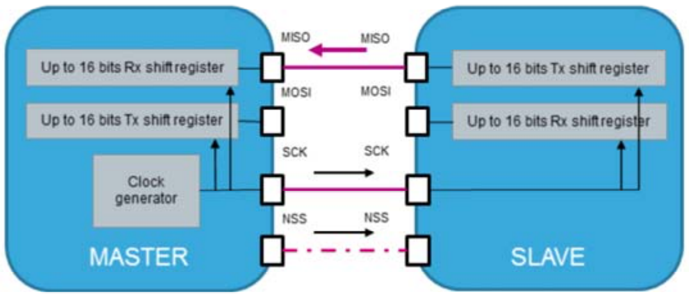
The SPI master always controls the bus traffic and provides the clock signal to the dedicated slave through the SCK line. The master can select the slave it wants to communicate with through the optional NSS signal.
Data stored in the dedicated shift registers can be exchanged synchronously between master and slave through the MOSI – Master Output Slave Input and the MISO - Master Input Slave Output data lines.
When master and slave nodes communicate in Full-duplex mode, both data lines are used and synchronous data flows in both directions at the same time.



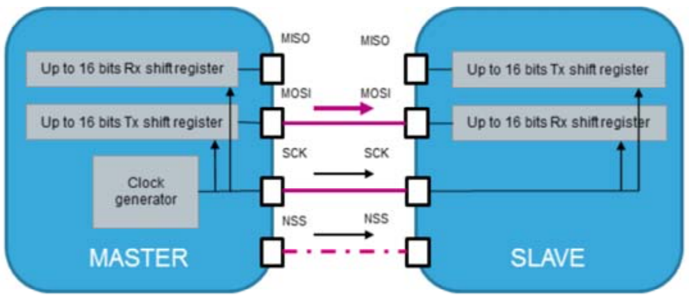
Interconnection of SPI nodes

Various master - slave interconnections are supported

- Simplex mode (unidirectional communication)
  One node is a transmitter while the other is a receiver exclusively

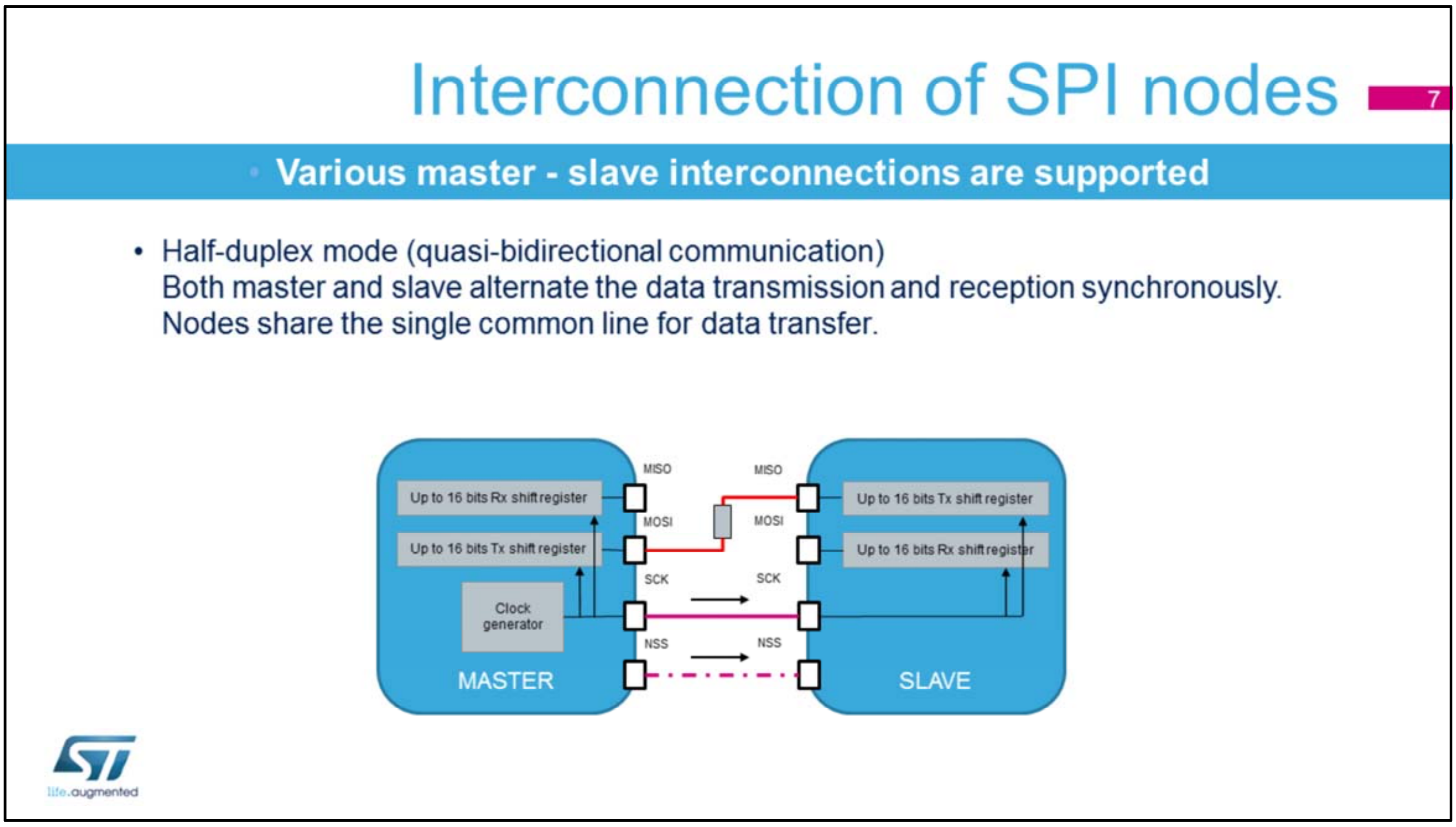


MISO
MISO
MOSI
MOSI
SCK
SCK
NSS
NSS
Up to 16 bits Rx shift register
Up to 16 bits Tx shift register
Clock generator
MASTER
Up to 16 bits Tx shift register
Up to 16 bits Rx shift register
SLAVE



MISO
MISO
MOSI
MOSI
SCK
SCK
NSS
NSS
Up to 16 bits Rx shift register
Up to 16 bits Tx shift register
Clock generator
MASTER
Up to 16 bits Tx shift register
Up to 16 bits Rx shift register
SLAVE

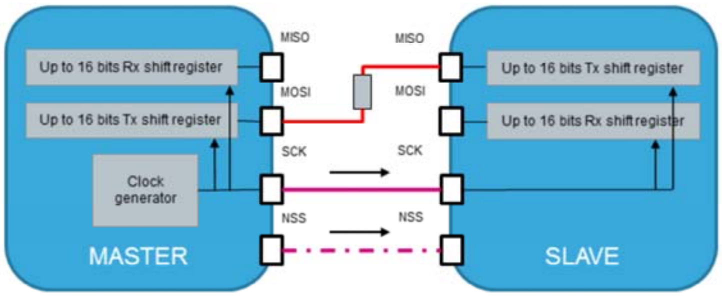
In Simplex mode, one node transmits data while the other one receives it. Data flows in one direction only, and the dedicated data line is used for data transfers exclusively. Unused SPI pins can be used for other purposes.



Interconnection of SPI nodes

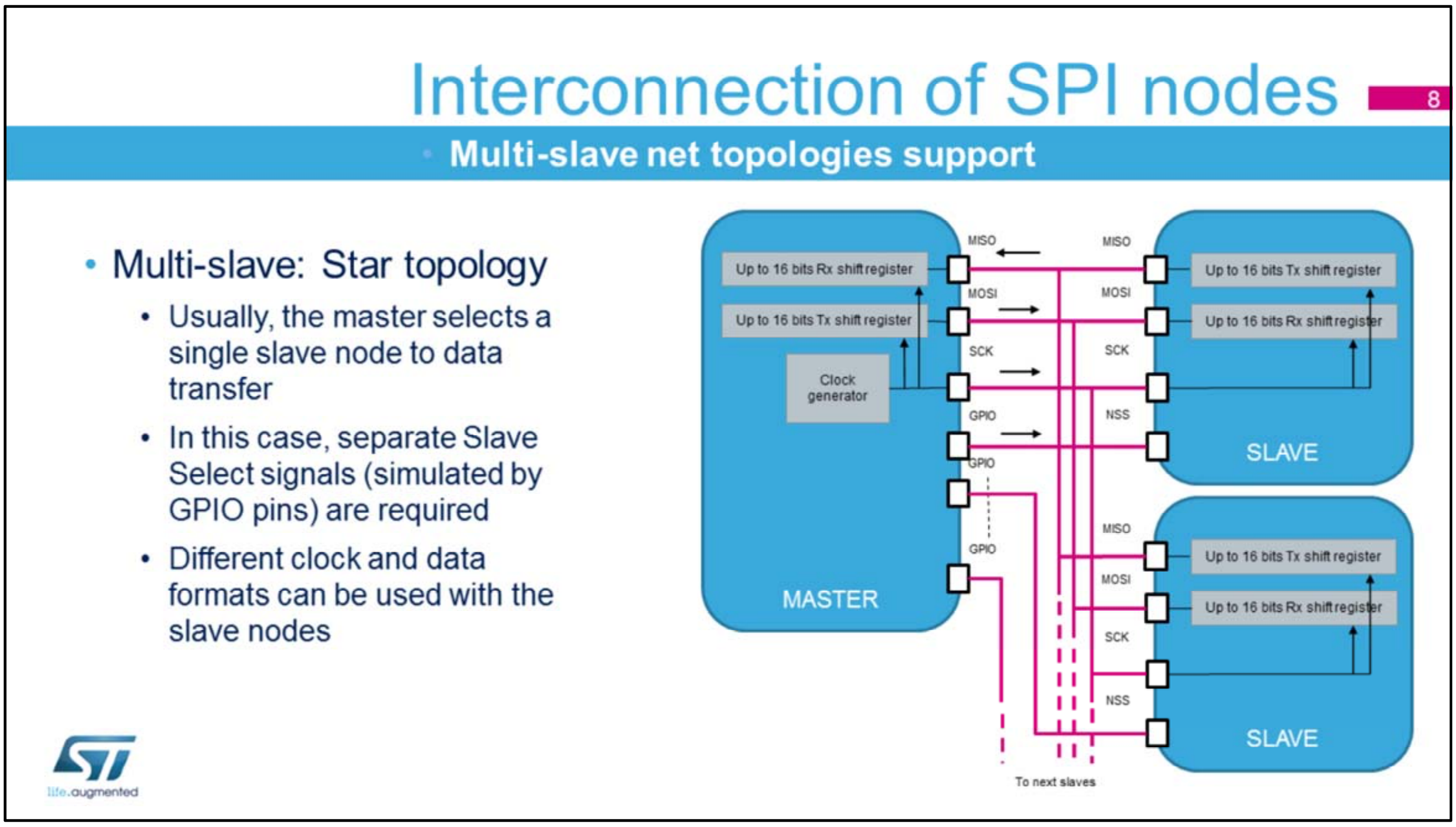
Various master - slave interconnections are supported

- Half-duplex mode (quasi-bidirectional communication)
  Both master and slave alternate the data transmission and reception synchronously.
  Nodes share the single common line for data transfer.


MISO
MOSI
SCK
NSS
Up to 16 bits Rx shift register
Up to 16 bits Tx shift register
Clock generator
MASTER
MISO
MOSI
SCK
NSS
Up to 16 bits Tx shift register
Up to 16 bits Rx shift register
SLAVE

life.augmented

Half-duplex mode integrates the previous two modes by sharing a single line for data exchanges and data flows in only one direction at a single time. There is a cross-connection between the master MOSI and the slave MISO pins in this mode. The master and slave have to alternate their transmitting and receiving roles synchronously when using this common data line. It is common to add a serial resistor on the half-duplex data line to prevent possible temporary short-circuit connections, since the re-configurations of the data directions at master and slave nodes are not usually synchronized.

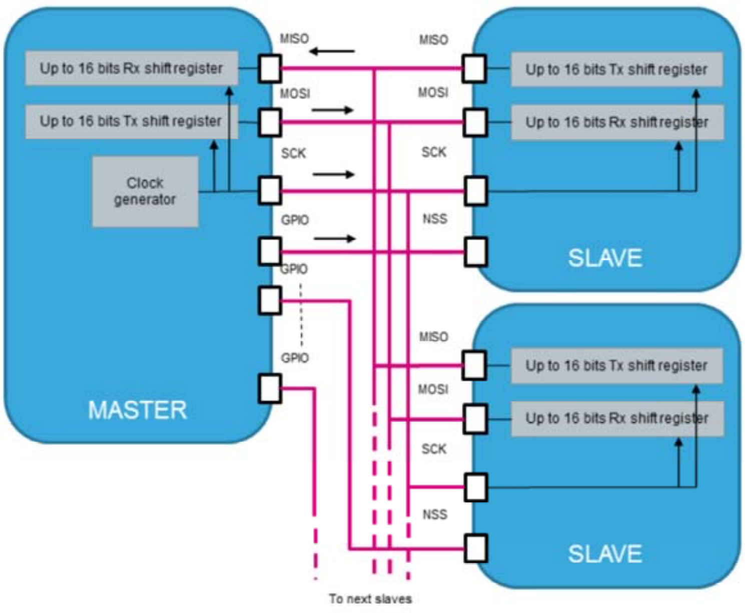


Interconnection of SPI nodes

8

Multi-slave net topologies support

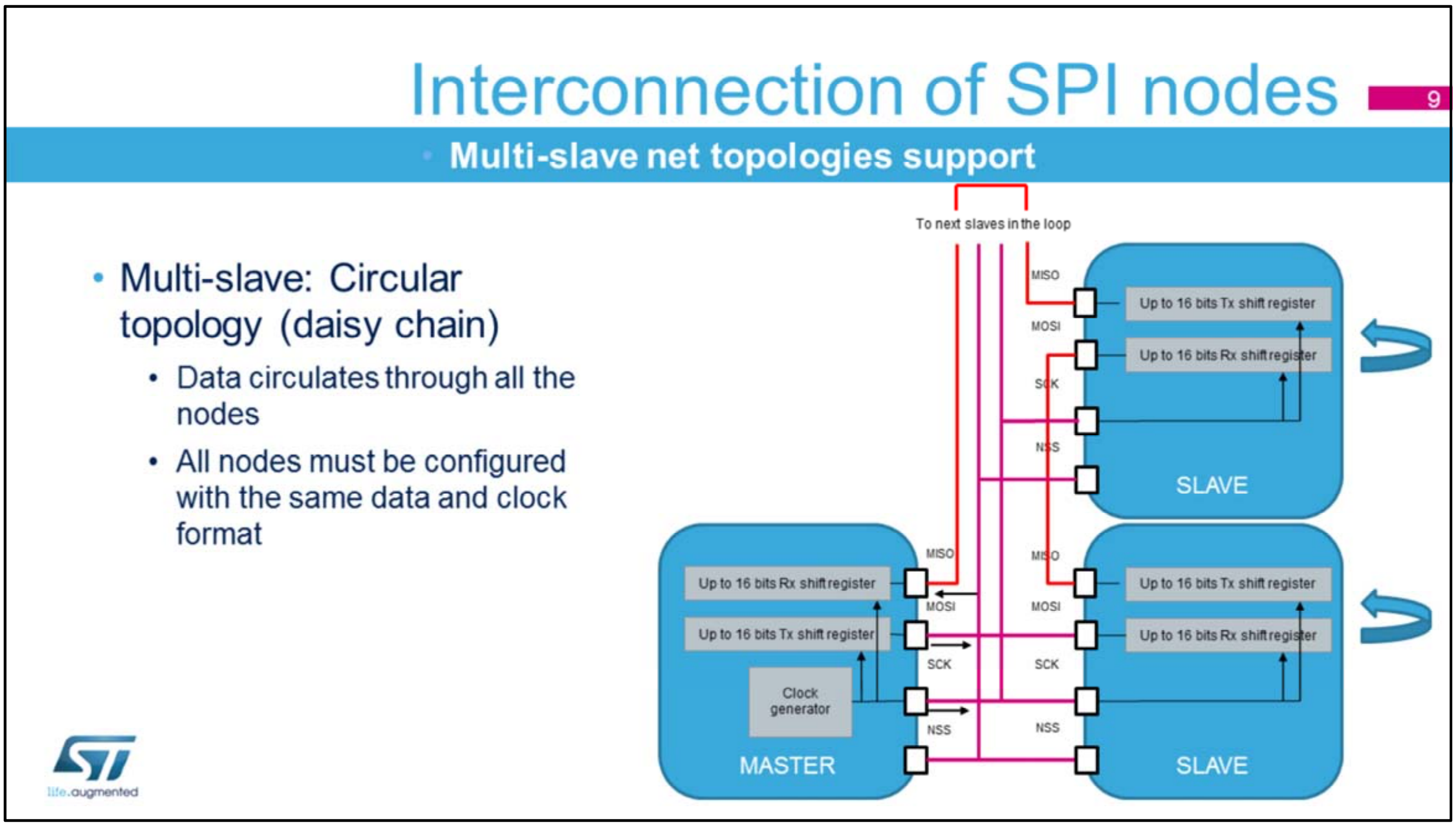
- Multi-slave: Star topology
  - Usually, the master selects a single slave node to data transfer
  - In this case, separate Slave Select signals (simulated by GPIO pins) are required
  - Different clock and data formats can be used with the slave nodes



MISO
MISO
Up to 16 bits Rx shift register
Up to 16 bits Tx shift register
MOSI
MOSI
Up to 16 bits Tx shift register
Up to 16 bits Rx shift register
SCK
SCK
Clock generator
GPIO
NSS
GPIO
SLAVE
MISO
GPIO
Up to 16 bits Tx shift register
MOSI
Up to 16 bits Rx shift register
MASTER
SCK
NSS
SLAVE
To next slaves

life.augmented

When the SPI network includes more than one slave, a star topology is commonly used when all the SCK, MOSI and MISO signals are interconnected.
Usually, the master communicates with one slave at a time, since just one slave can transmit data back to the master through the common MISO pin. In this case, when reading from slaves is required, a separate Slave Select signal from the master has to be provided to each slave node, so the master selects just a single slave for communication session. When separate Slave Select signals are applied, a different SPI data and clock format configuration can be used optionally for the slave nodes.



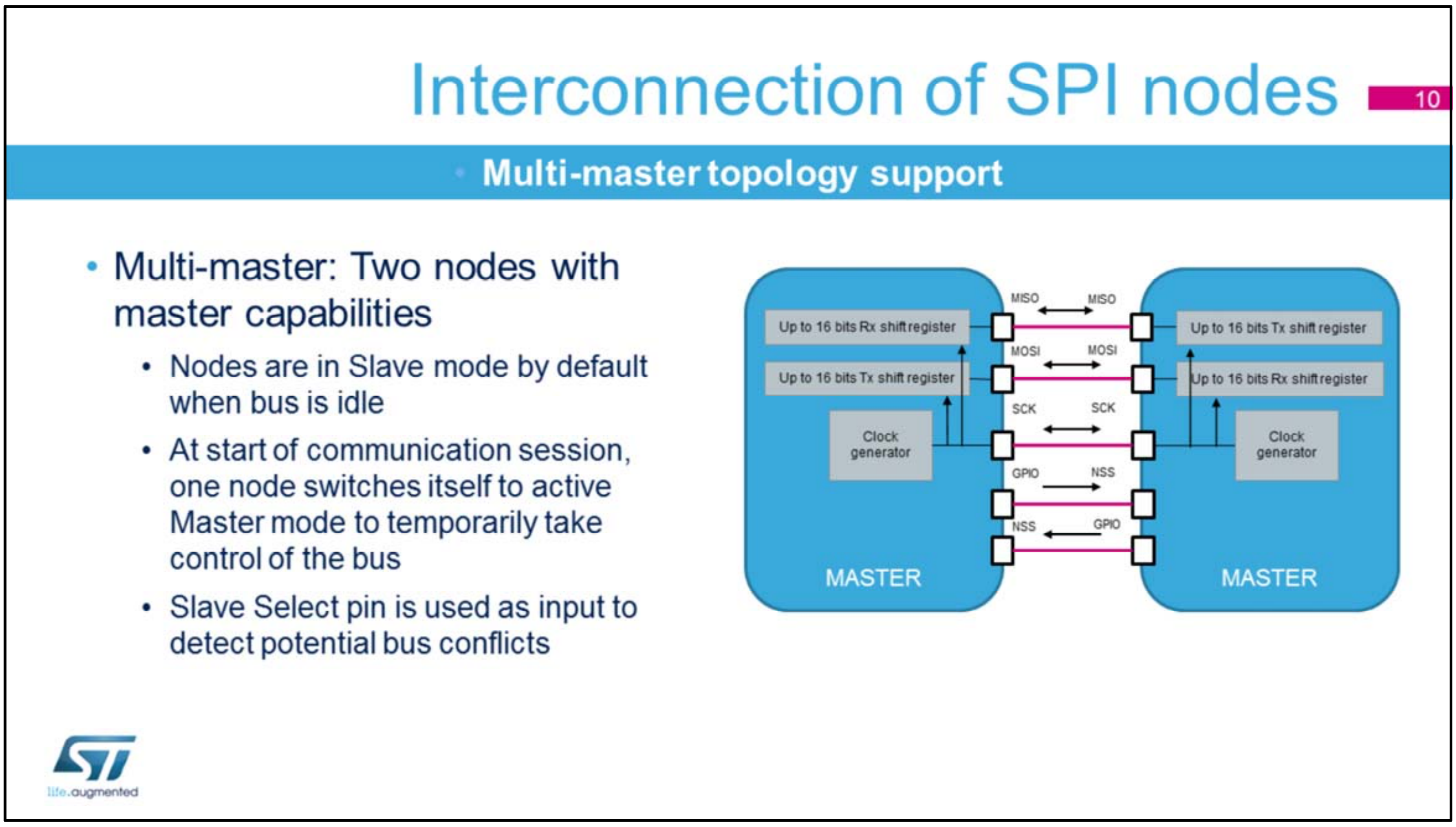
Interconnection of SPI nodes

Multi-slave net topologies support

- Multi-slave: Circular topology (daisy chain)
  - Data circulates through all the nodes
  - All nodes must be configured with the same data and clock format

To next slaves in the loop
MISO
MOSI
SCK
NSS
Up to 16 bits Tx shift register
Up to 16 bits Rx shift register
SLAVE
Up to 16 bits Rx shift register
Up to 16 bits Tx shift register
Clock generator
MASTER
life.augmented

Another multi-slave configuration is the circular topology where all the data inputs and outputs of the nodes are connected together in a closed serial chain.
A common Slave Select signal is used as communication occurs at the same time. This is why all the nodes have to keep the same data and clock format configuration. When the slave SPI nodes are provided by a microcontroller, the internal transmit and receive shift registers are usually physically separated.
So the data transferred between them has to be handled by software in this circular mode while the master node has to provide sufficient time between data to provide these internal transfers.

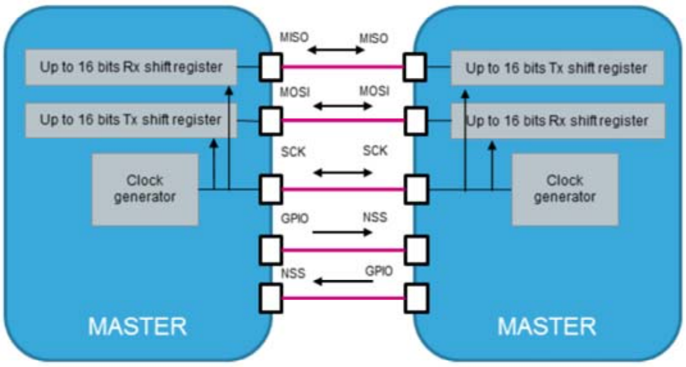


Interconnection of SPI nodes

10

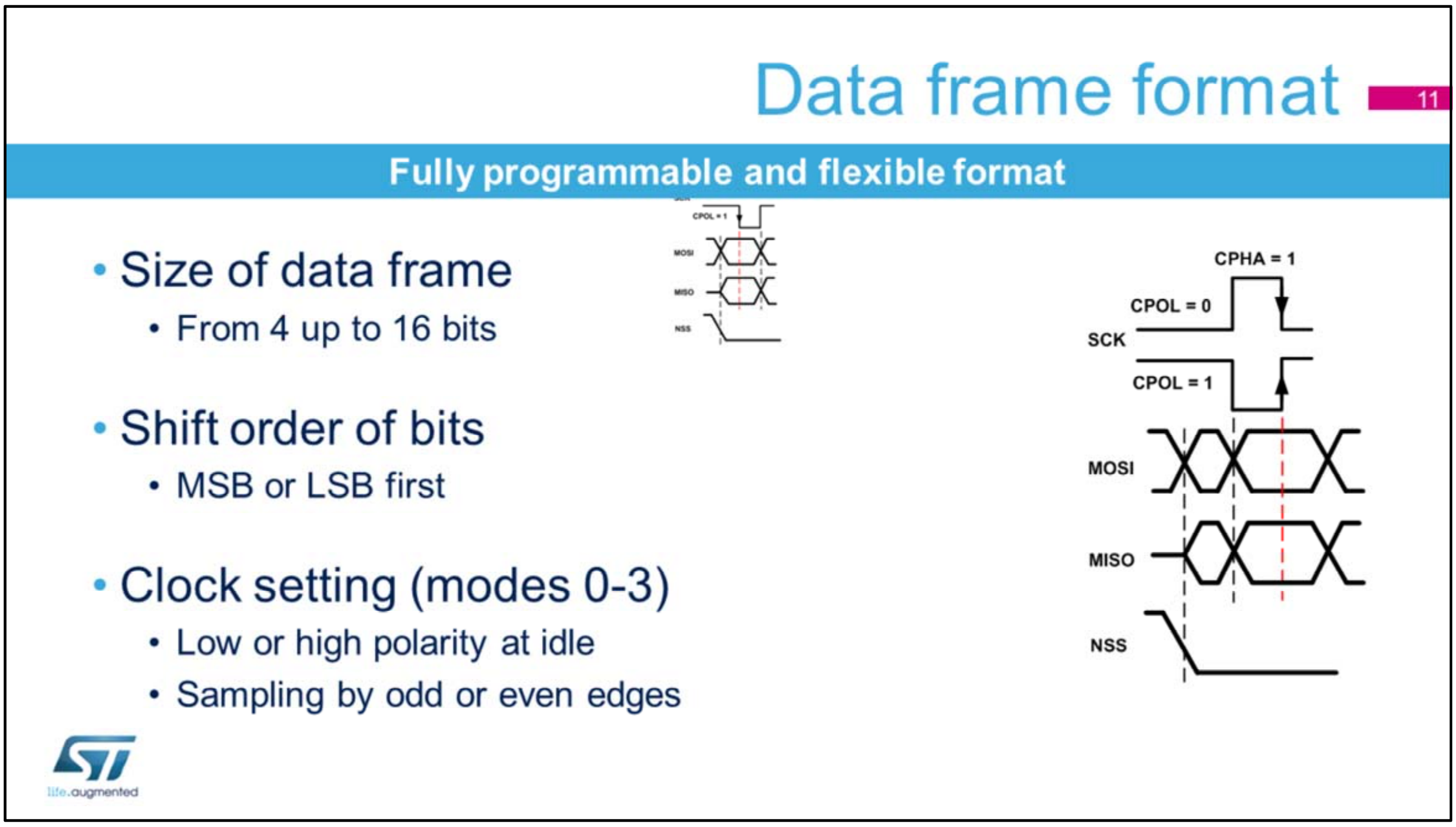
Multi-master topology support

- Multi-master: Two nodes with master capabilities
  - Nodes are in Slave mode by default when bus is idle
  - At start of communication session, one node switches itself to active Master mode to temporarily take control of the bus
  - Slave Select pin is used as input to detect potential bus conflicts


MISO
MISO
Up to 16 bits Rx shift register
Up to 16 bits Tx shift register
MOSI
MOSI
Up to 16 bits Tx shift register
Up to 16 bits Rx shift register
SCK
SCK
Clock generator
Clock generator
GPIO
NSS
NSS
GPIO
MASTER
MASTER

life.augmented

An SPI network can also operate in a Multi-master environment. This mode is used to connect together two master nodes exclusively.
By default, both nodes are in Slave mode as long as the network is idle.
When one node wants to take control of the bus, it switches itself into Master mode and asserts the Slave Select signal on the other node through a GPIO pin. Both Slave Select pins work as a hardware input to detect potential bus collisions between nodes as only one master can use the SPI bus at a single time. After the session is completed, the master node releases the Slave Select signal and returns to passive Slave mode waiting for the start of the next session.

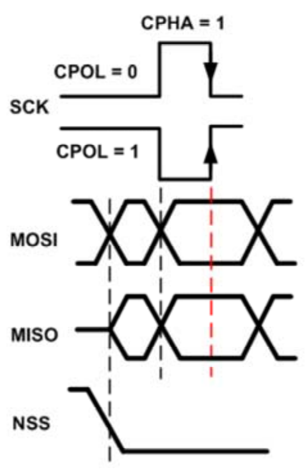


Data frame format

11

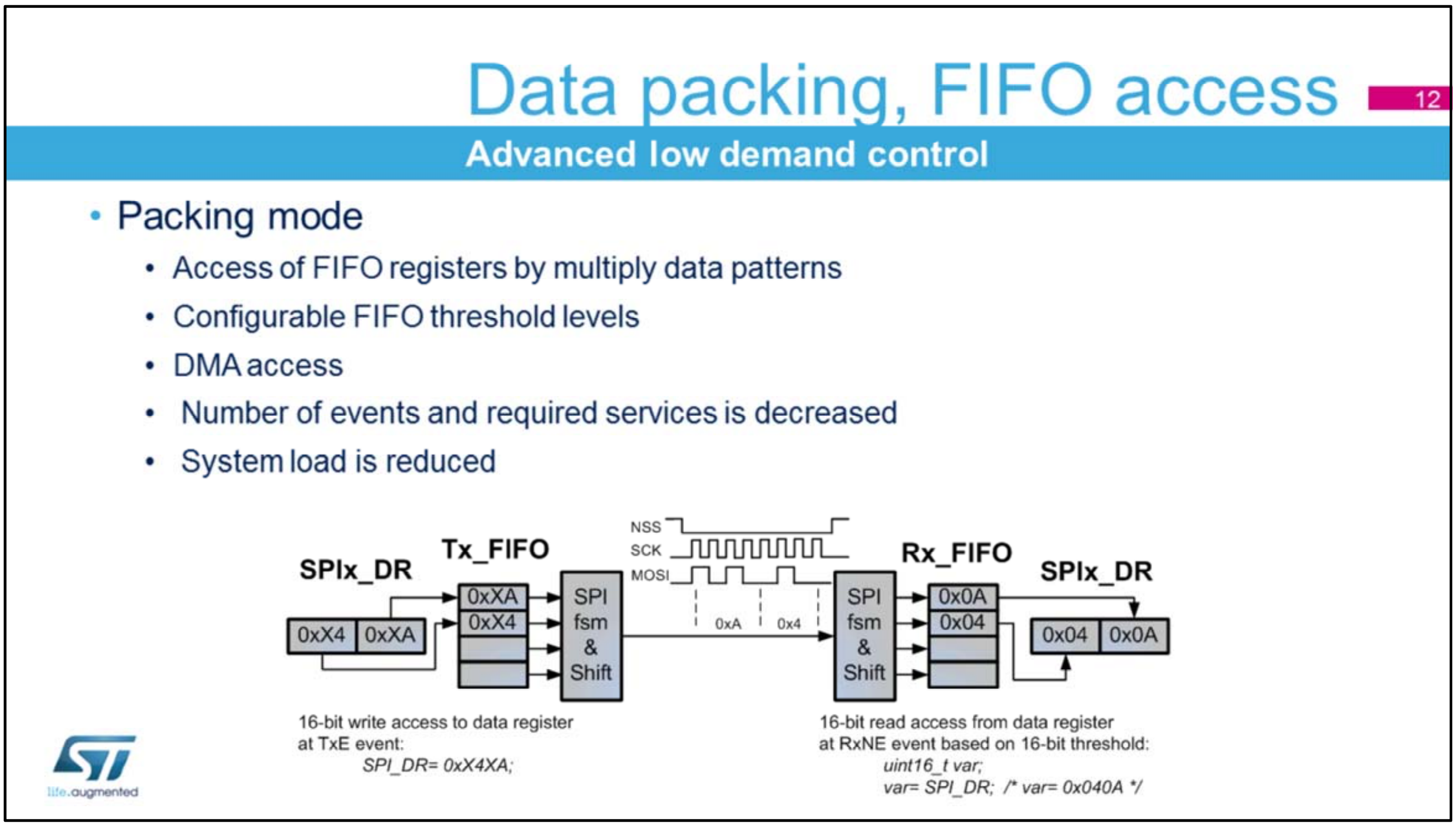
Fully programmable and flexible format

- Size of data frame
  - From 4 up to 16 bits
- Shift order of bits
  - MSB or LSB first
- Clock setting (modes 0-3)
  - Low or high polarity at idle
  - Sampling by odd or even edges



life.augmented

There are a few controls that are used to set up the data format.
Users can define the data frame size and the transmit order of the data bits.
The clock can be set to one of four basic configurations defined in the Motorola SPI specification. The combination of two bits controls the polarity and phase of the clock signal.
When the phase control bit is cleared, data bits are sampled on the odd clock edges, and the even clock edges synchronize the shifting of next bit onto the data line. The opposite occurs when the phase control bit is set.
The clock polarity bit defines the idle and initial state of the clock signal and therefore which clock edge is used for data sampling or shifting, depending on the clock phase setting.

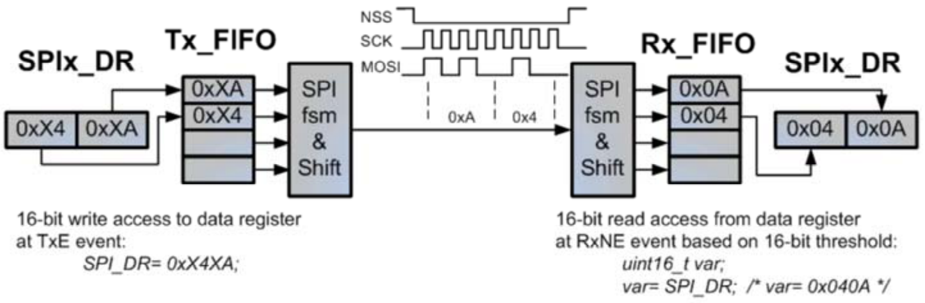


Data packing, FIFO access

12

Advanced low demand control

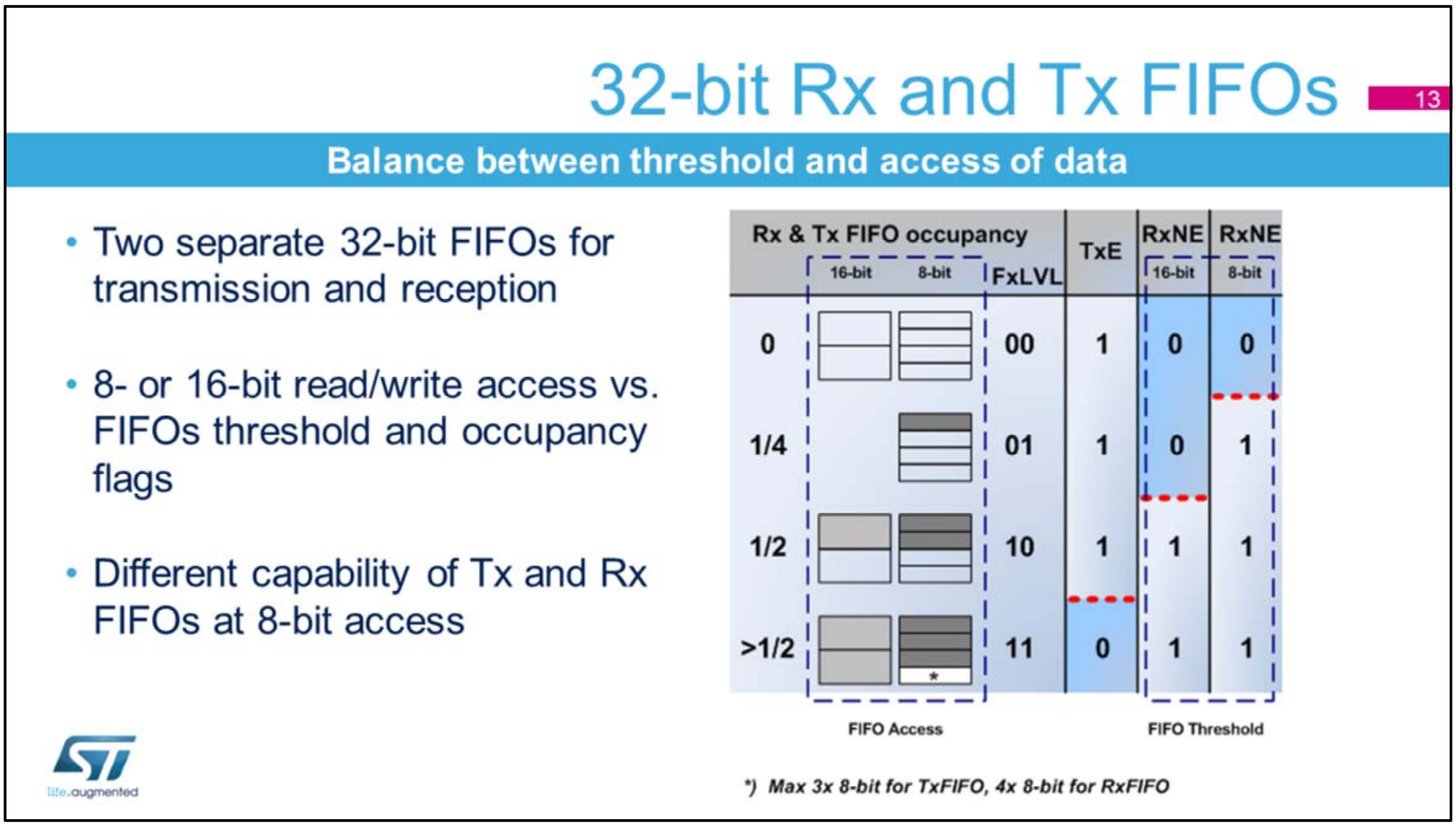
- Packing mode
  - Access of FIFO registers by multiply data patterns
  - Configurable FIFO threshold levels
  - DMA access
  - Number of events and required services is decreased
  - System load is reduced



16-bit write access to data register at TxE event:
SPI_DR= 0xX4XA;

16-bit read access from data register at RxNE event based on 16-bit threshold:
uint16_t var;
var= SPI_DR; / var= 0x040A */*

When communication speeds are fast and data frames short, it can be a demanding task to ensure correct data flows when the clock signal becomes continuous and the Full-duplex mode is used. Slave nodes have to properly service all the transactions sent by the master to prevent any data overrun or underrun conditions.
When the data frame size fits into a byte, Packing mode can be used. Then multiple data patterns can be written or read in a single access to the FIFO registers. Together with the proper setting of the FIFO threshold event, the number of events to service will decrease to better control the data flow. When the DMA is used additionally, overall loading on the system is significantly reduced.
In the figure shown, you can see the principle of how two short 4 bit data frames can be written and read by a single 16-bit access in the dedicated FIFO registers. The read or write data accesses are performed just by raising a single event.



32-bit Rx and Tx FIFOs

13

Balance between threshold and access of data

- Two separate 32-bit FIFOs for transmission and reception
- 8- or 16-bit read/write access vs. FIFOs threshold and occupancy flags
- Different capability of Tx and Rx FIFOs at 8-bit access

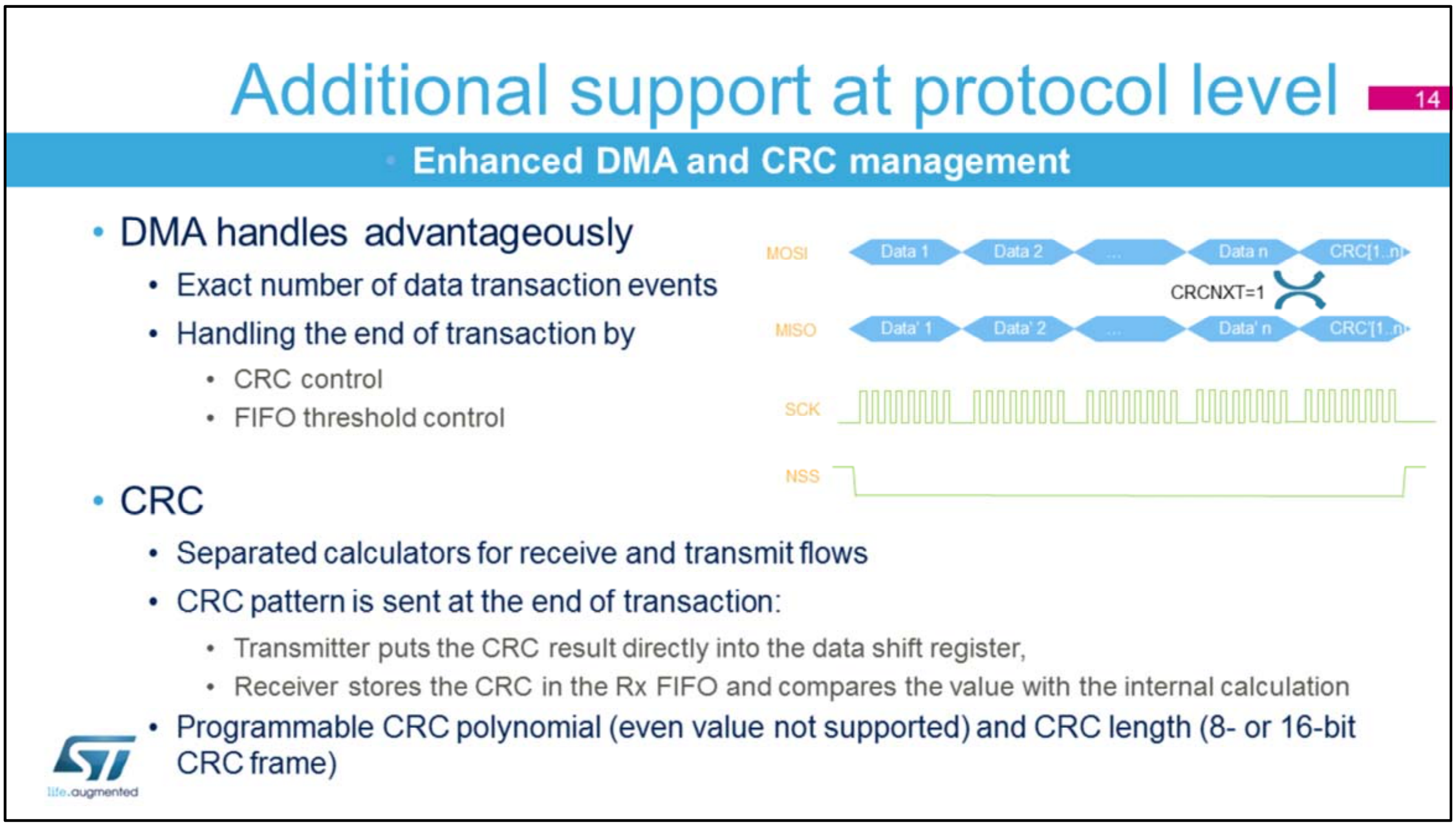
Rx & Tx FIFO occupancy	FIFO Access 16-bit	FIFO Access 8-bit	FxLVL	TxE	RxNE 16-bit	RxNE 8-bit
0			00	1	0	0
1/4			01	1	0	1
1/2			10	1	1	1
>1/2		*	11	0	1	1

FIFO Access — FIFO Threshold

*) Max 3x 8-bit for TxFIFO, 4x 8-bit for RxFIFO

life.augmented

The SPI peripheral features two 32-bit FIFOs to handle the data flow. The FIFOs can be accessed by using either 8-bit or 16-bit data access instructions.
During reception, the events generated from the FIFO depend on the threshold setting.
This table gives an overview of how the event flag behavior changes depending on the configuration. It is important to keep the FIFO access balanced with the threshold setting so the data consistency is not lost. During transmission, the FIFO occupancy depends on the data access.
The system can never predict the next access to the transmission FIFO, so the FIFO capability is not fully used when an 8-bit write access is applied to fill the second half of the FIFO. In this case, the TxE flag is cleared as a consequence, even though the Tx FIFO is not fully occupied.



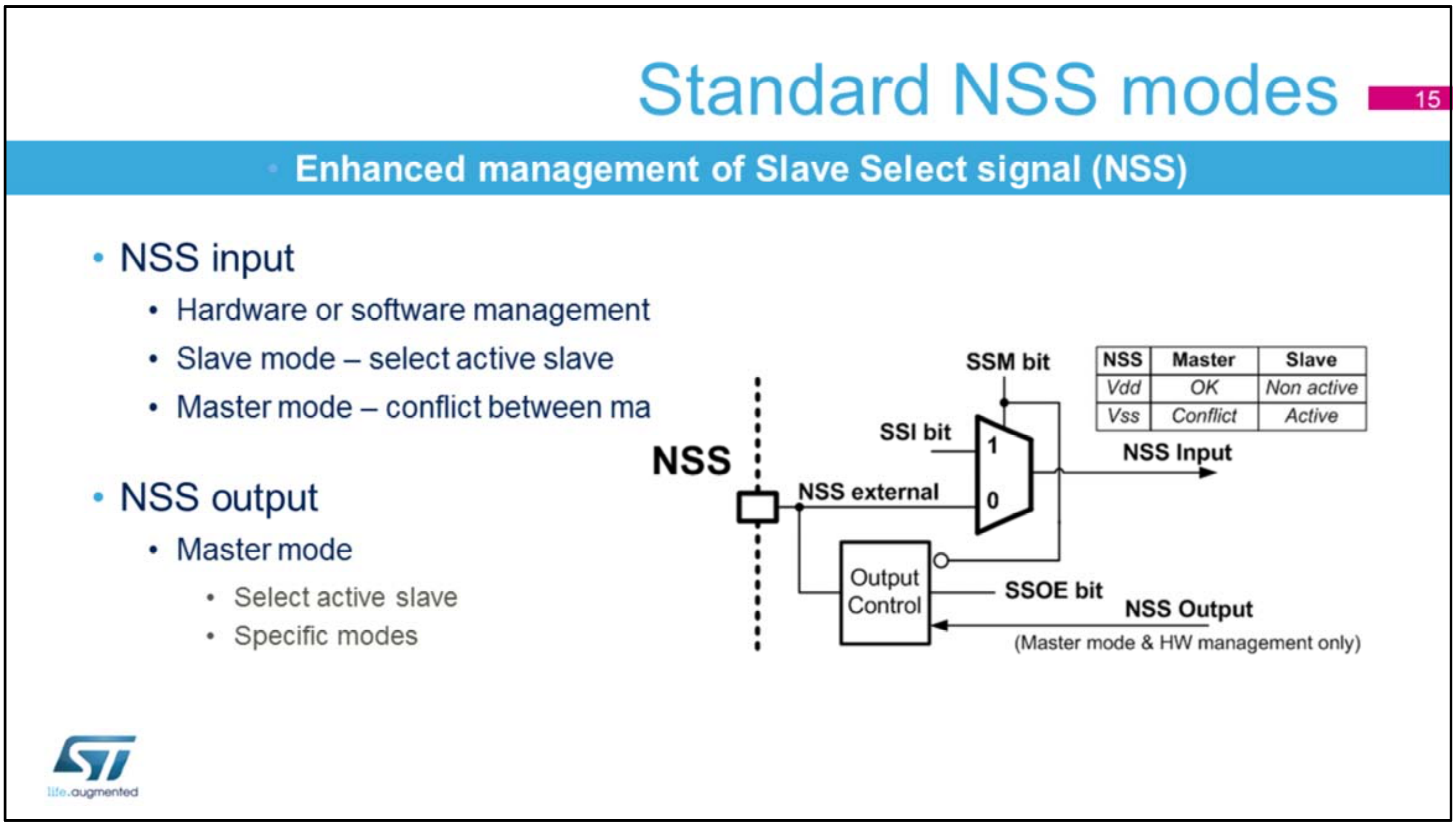
Additional support at protocol level

14

Enhanced DMA and CRC management

- DMA handles advantageously
  - Exact number of data transaction events
  - Handling the end of transaction by
    - CRC control
    - FIFO threshold control
- CRC
  - Separated calculators for receive and transmit flows
  - CRC pattern is sent at the end of transaction:
    - Transmitter puts the CRC result directly into the data shift register,
    - Receiver stores the CRC in the Rx FIFO and compares the value with the internal calculation
  - Programmable CRC polynomial (even value not supported) and CRC length (8- or 16-bit CRC frame)

During protocol-level communication, the DMA can be used favorably to apply the CRC patterns or change the FIFO threshold setting correctly after an exact number of data is transferred. For threshold control, the last odd data frame is correctly applied in Packed mode when the number of frames is not aligned with the packet size. If the CRC is enabled, separated CRC calculators are used for the transmitter and receiver. The result of the CRC calculation is applied at the end of the transfer either automatically by the DMA or by software. Results from the transmitter CRC calculation register are loaded directly into the shift register, and the received CRC value is stored in the FIFO and compared with the result of the receiver CRC calculation register. The CRC polynomial used for the calculation is programmable, and the length of the CRC pattern can be set to either 8- or 16-bit frames.

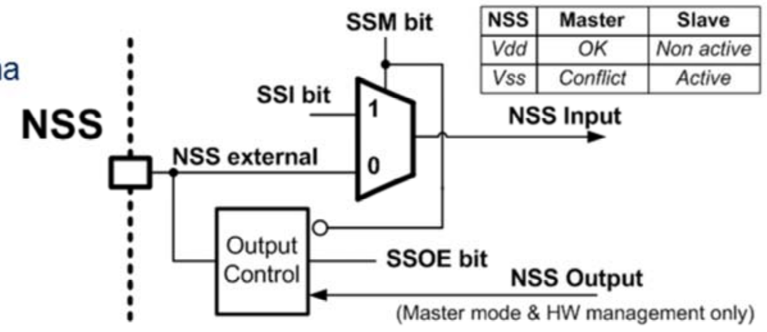


Standard NSS modes

15

Enhanced management of Slave Select signal (NSS)

- NSS input
  - Hardware or software management
  - Slave mode – select active slave
  - Master mode – conflict between ma
- NSS output
  - Master mode
    - Select active slave
    - Specific modes

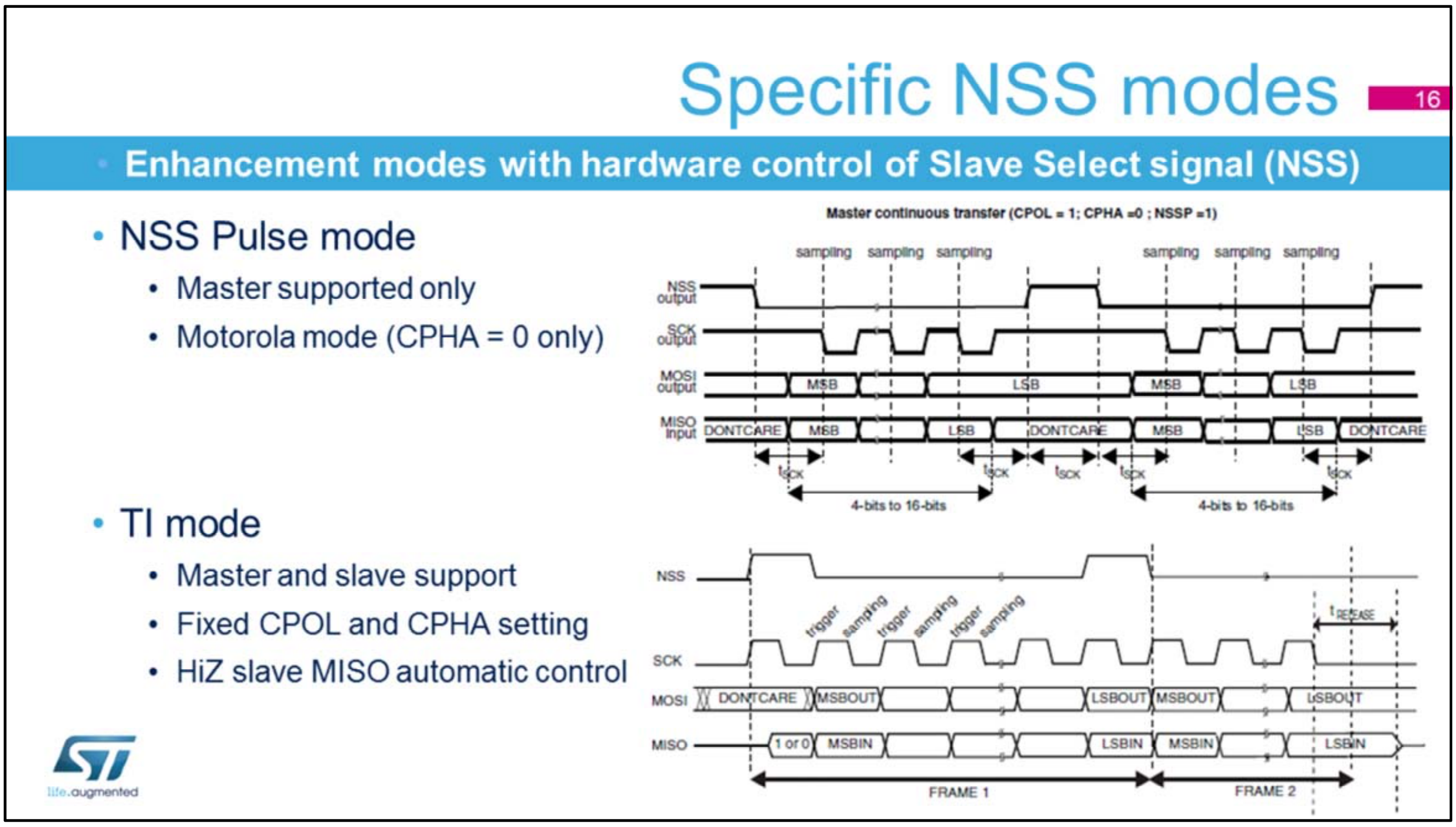


NSS	Master	Slave
Vdd	*OK*	*Non active*
Vss	*Conflict*	*Active*

life.augmented

The Slave Select signal (NSS) is commonly used by the Master node to select the Slave node for communication. It is generally used in a multi-slave topology but it can also be used to synchronize the data flow in single master-slave pair. The Slave Select signal can operate either as an input or as an output.

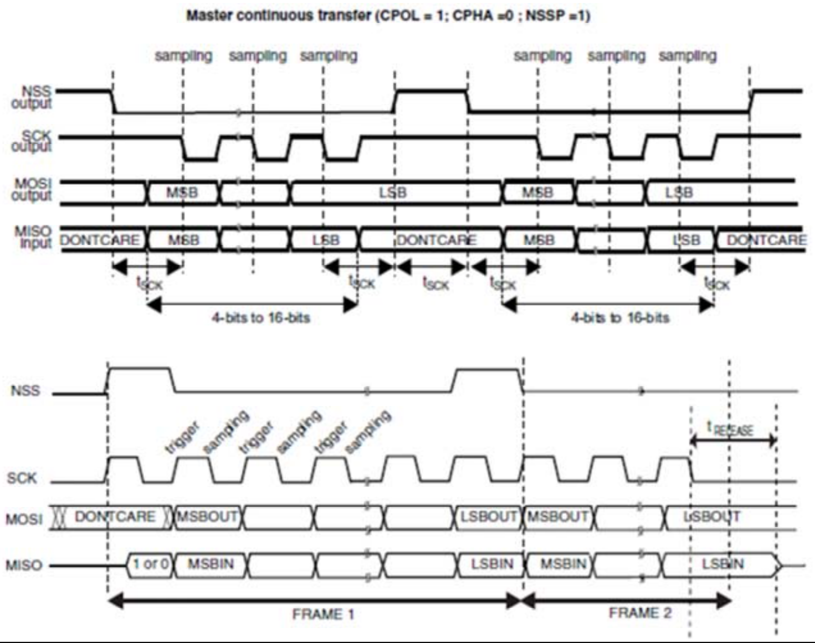
The NSS input can be managed externally by hardware or internally by software depending on the SSM and SSI bits. As a slave, it always works as an input and enables the slave for communication. As a master input, it indicates potential conflicts between masters in a multi-master system. As a master output, it is managed by hardware in a standard or specific control mode. Generally, in a standard mode, the master Slave Select outputs can be easily replaced by GPIOs under software control.



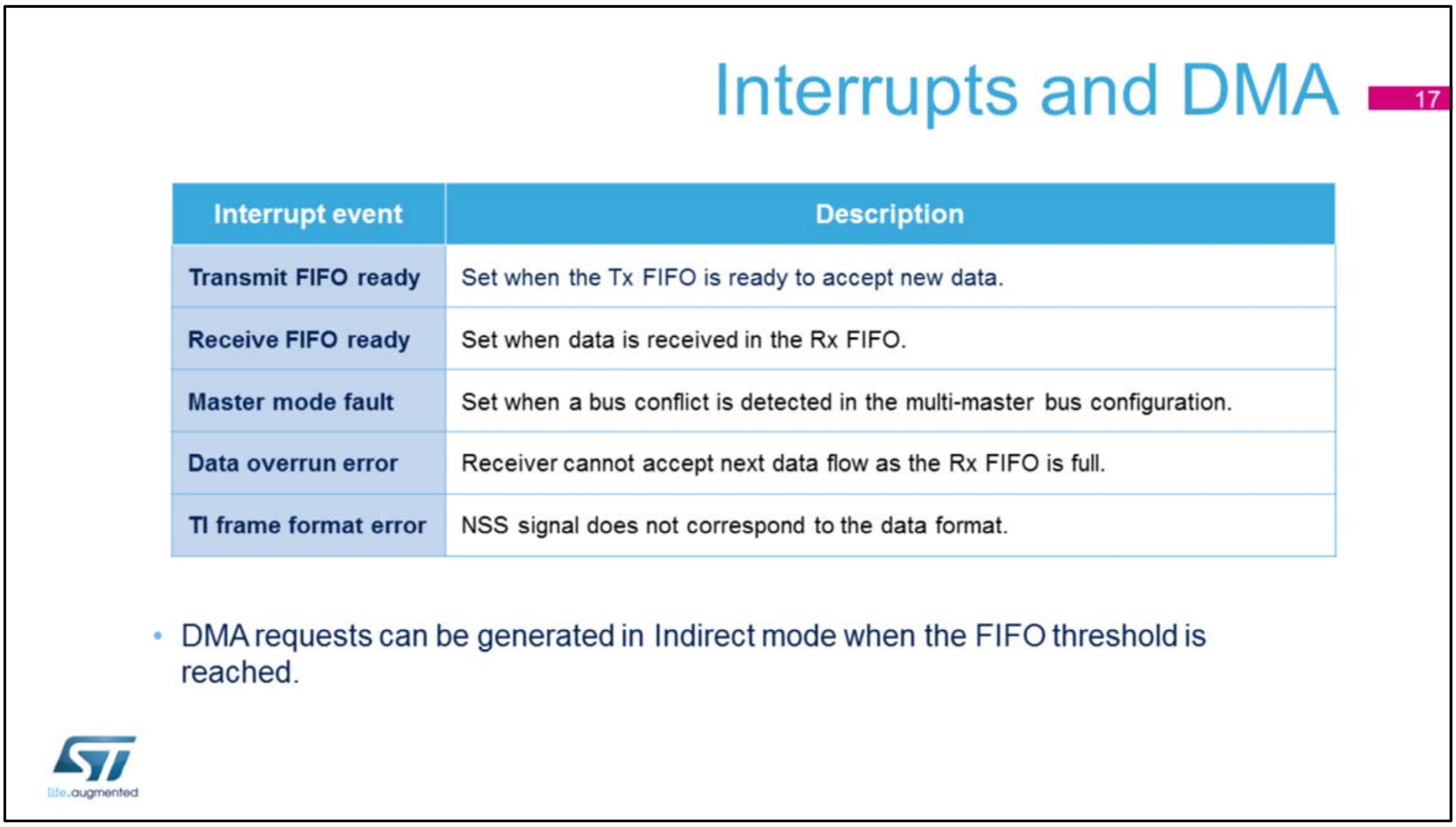
Specific NSS modes

Enhancement modes with hardware control of Slave Select signal (NSS)

- NSS Pulse mode
  - Master supported only
  - Motorola mode (CPHA = 0 only)
- TI mode
  - Master and slave support
  - Fixed CPOL and CPHA setting
  - HiZ slave MISO automatic control



There are a few enhanced modes when the Slave Select signal is under specific hardware control.
The Slave Select signal can operate in a pulse mode where the master generates pulses on the NSS output signal between data frames for a duration of one SPI clock period when there is a continuous transfer of data. The data is then interleaved by two SPI clock periods. The clock phase is fixed in this mode. Another enhanced mode is the TI mode where the data flow is synchronized by the NSS pulses, provided by the master, on the last bit of data.
The clock polarity and phase configuration is fixed and the slave data output is automatically switched into high impedance when the bus traffic stops and on a specific configurable timeout.



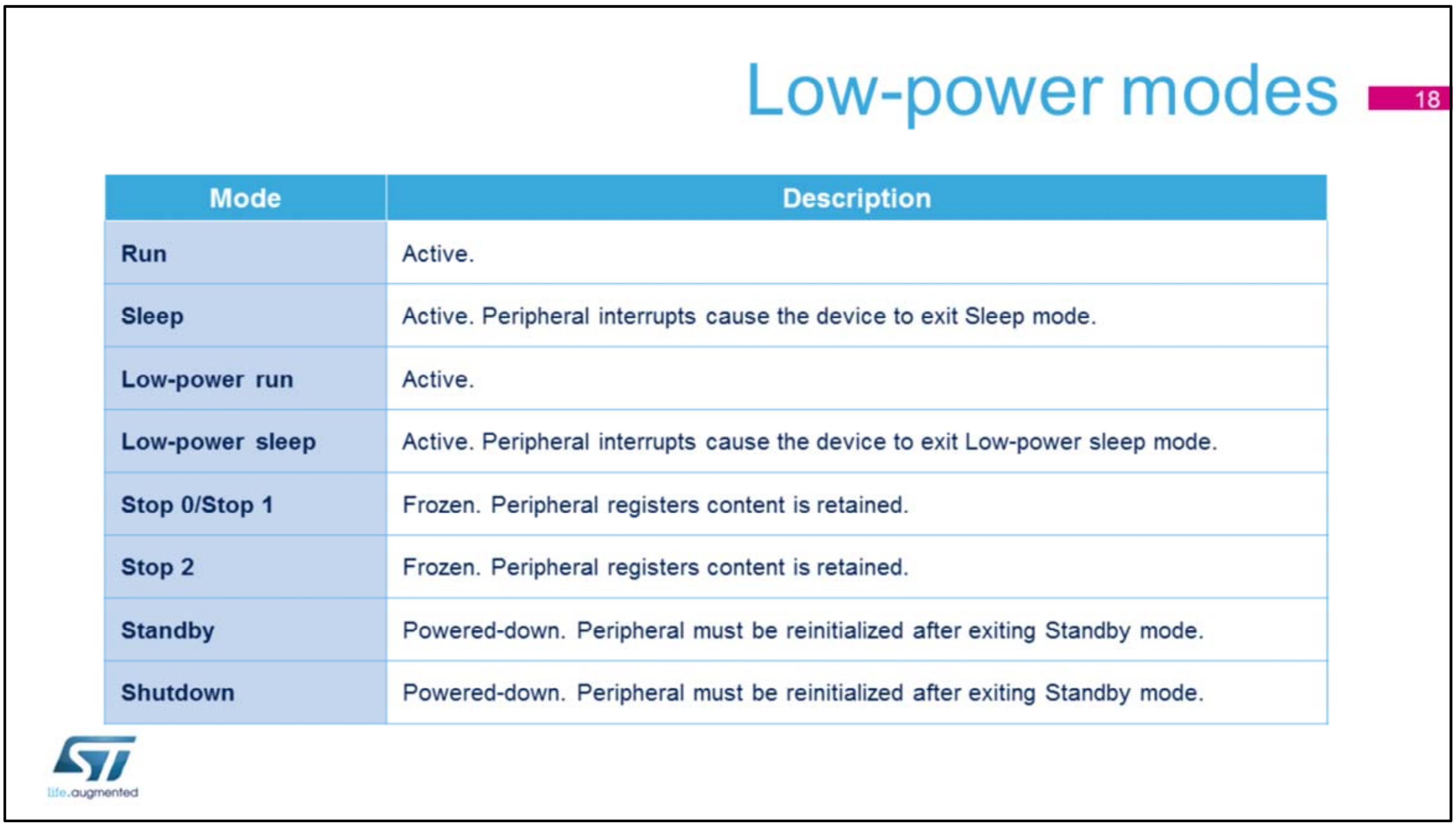
Interrupts and DMA

Interrupt event	Description
Transmit FIFO ready	Set when the Tx FIFO is ready to accept new data.
Receive FIFO ready	Set when data is received in the Rx FIFO.
Master mode fault	Set when a bus conflict is detected in the multi-master bus configuration.
Data overrun error	Receiver cannot accept next data flow as the Rx FIFO is full.
TI frame format error	NSS signal does not correspond to the data format.

- DMA requests can be generated in Indirect mode when the FIFO threshold is reached.

life.augmented

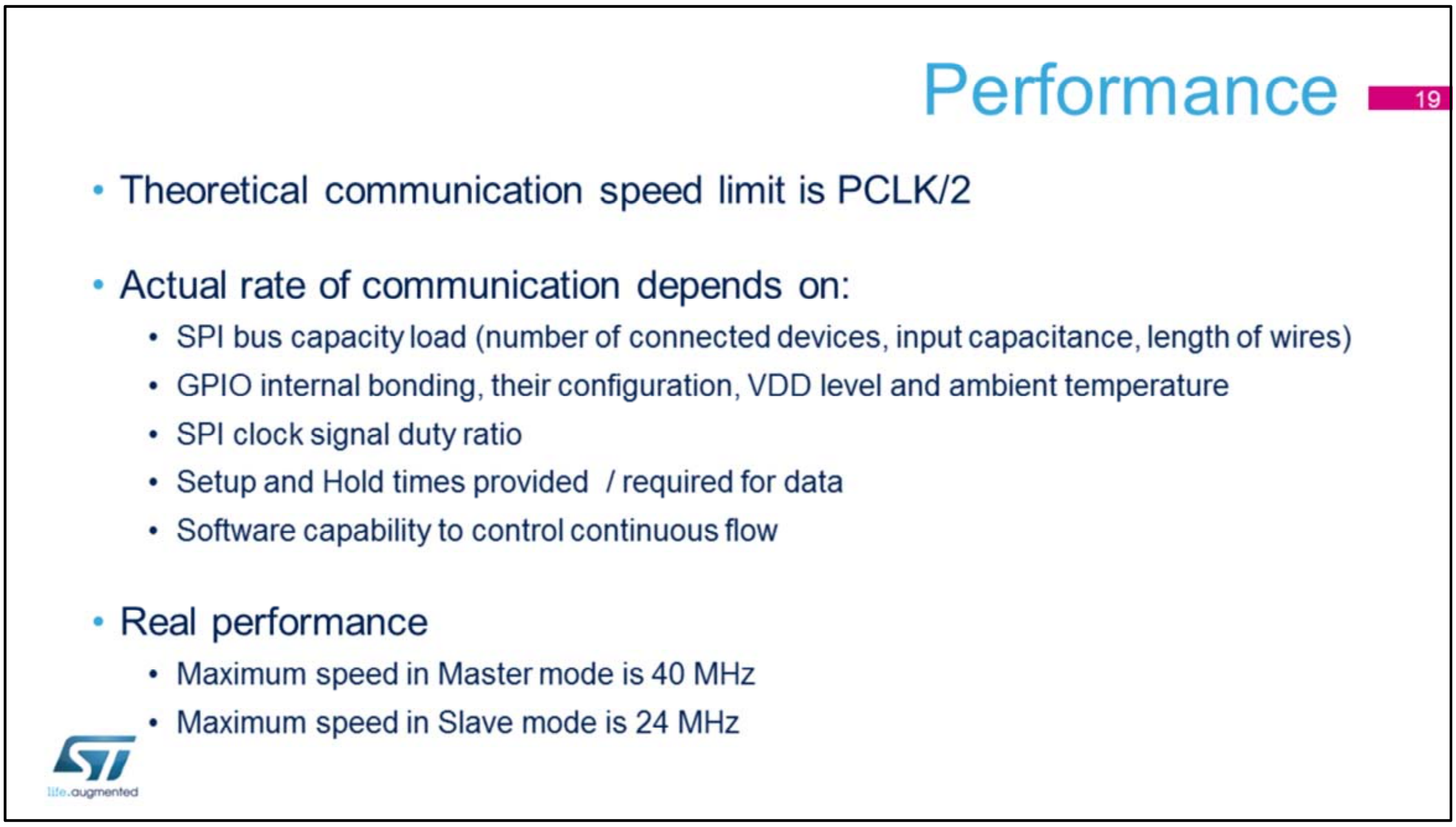
Here is an overview of the SPI interrupt events.
There are FIFO and error detection events to handle the data flow.
DMA requests are triggered internally by FIFO threshold events.



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 retained.
Stop 2	Frozen. Peripheral registers content is retained.
Standby	Powered-down. Peripheral must be reinitialized after exiting Standby mode.
Shutdown	Powered-down. Peripheral must be reinitialized after exiting Standby mode.

Here is an overview of the SPI status in specific low-power modes.
The device is not able to perform any communications in Stop, Standby or Shutdown mode. It is important to ensure that all SPI traffic is completed before the peripheral enters Stop or Power-down mode.

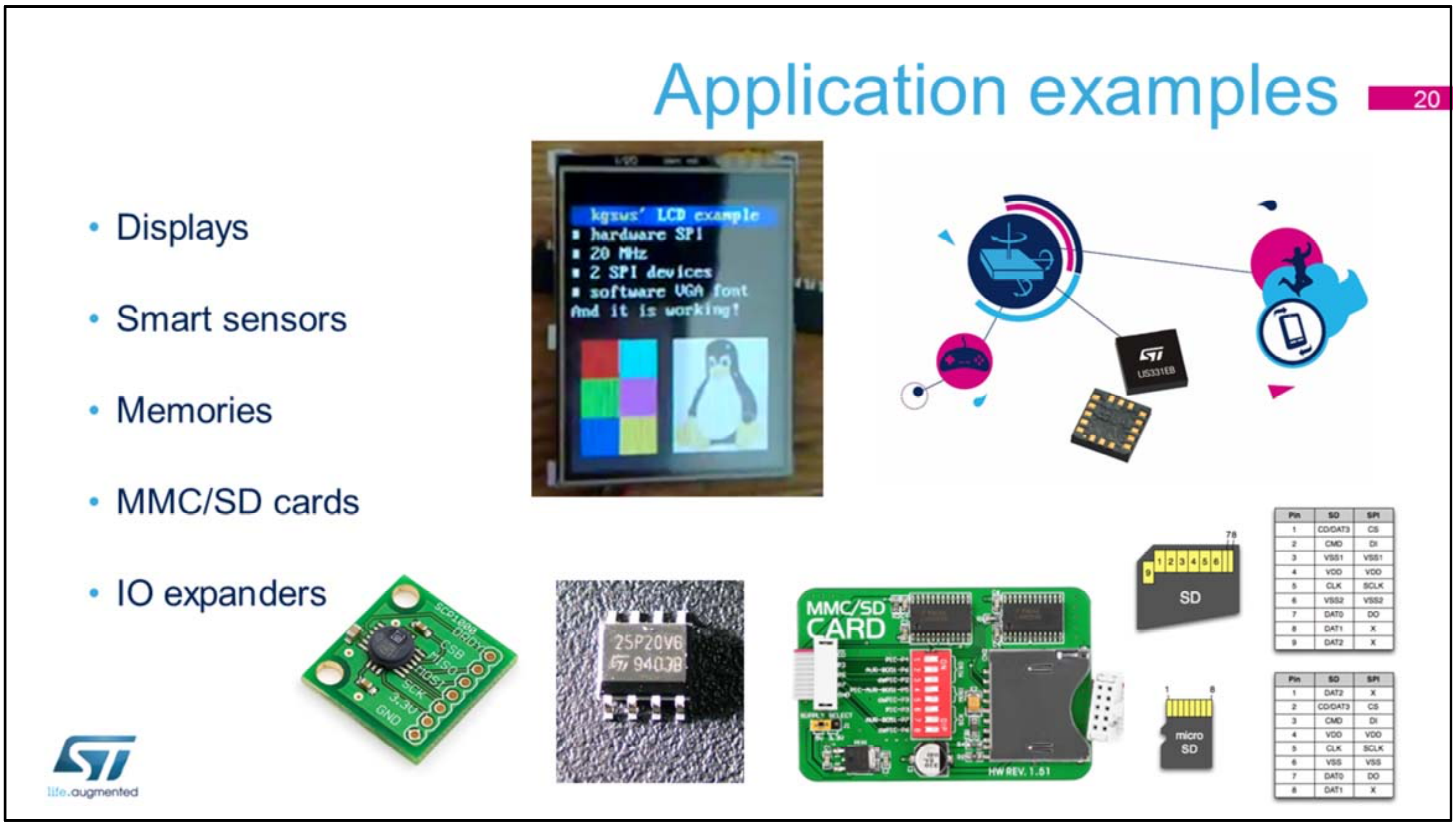


Performance

- Theoretical communication speed limit is PCLK/2
- Actual rate of communication depends on:
  - SPI bus capacity load (number of connected devices, input capacitance, length of wires)
  - GPIO internal bonding, their configuration, VDD level and ambient temperature
  - SPI clock signal duty ratio
  - Setup and Hold times provided / required for data
  - Software capability to control continuous flow
- Real performance
  - Maximum speed in Master mode is 40 MHz
  - Maximum speed in Slave mode is 24 MHz

life.augmented

The SPI performance depends mainly on the internal clock applied to the peripheral.
The peripheral clock should be at least the double of the maximum achievable communication frequency. The actual rate of communication can be decreased by many application factors.
The user has to consider SPI bus loads such as the number of nodes, the connection distance, input capacitance, as well as GPIO settings. Fast GPIO mode should be applied on the data and clock signals. Lower power supply voltages and extreme ambient temperatures also slow down the GPIO edges. Sometimes slower data Hold or Setup time requirements have to be respected between the nodes.
Managing a fast data flow can be demanding for application software due to frequent servicing of exceptions. The DMA capacity has to be considered as well when the system uses more DMA channels, frequent interrupt services or executes non-interruptible instructions (e.g. LDMIA).



Application examples

- Displays
- Smart sensors
- Memories
- MMC/SD cards
- IO expanders

```
kgzws' LCD example
▪ hardware SPI
▪ 20 MHz
▪ 2 SPI devices
▪ software VGA font
And it is working!
```

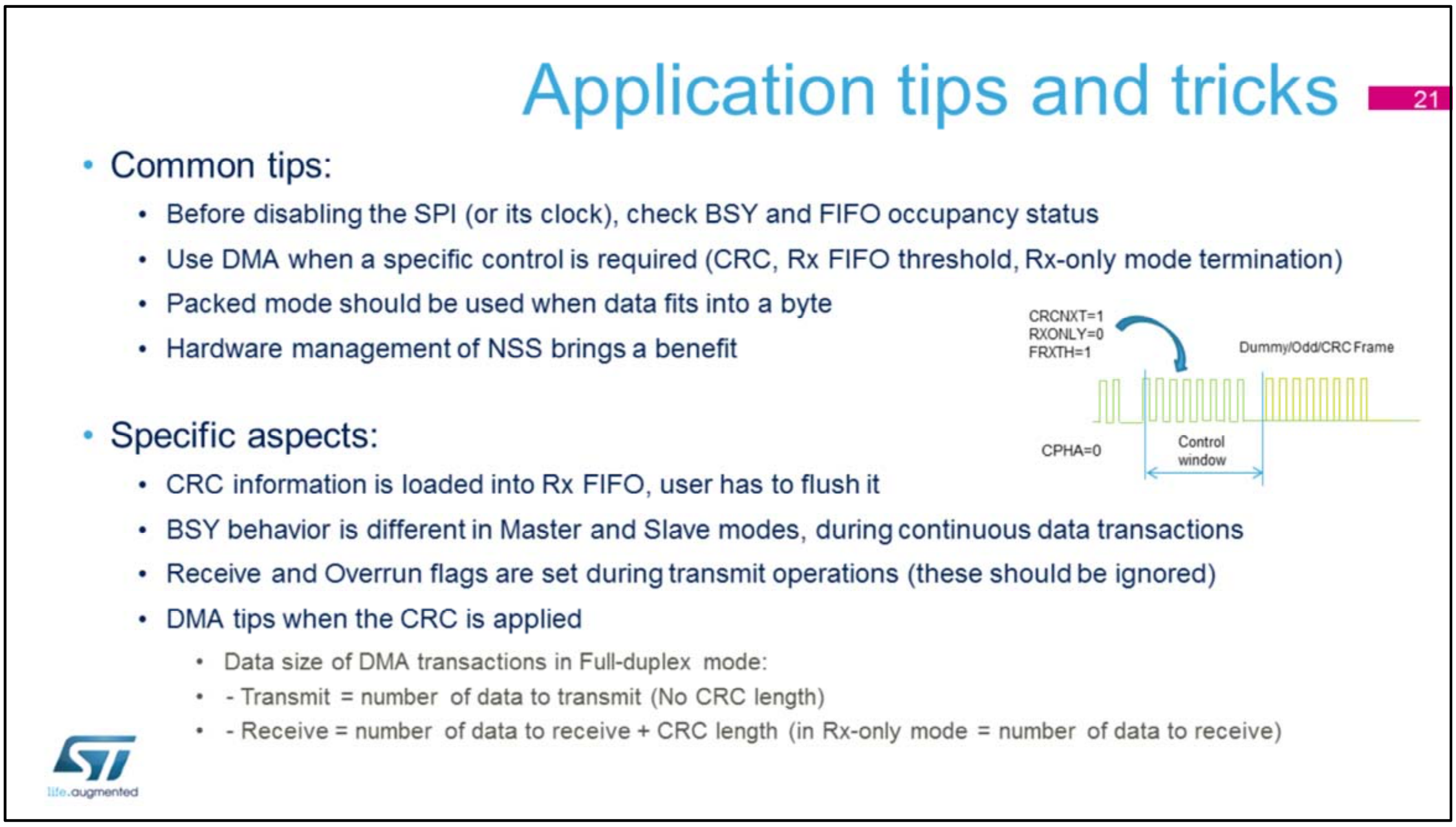
US331EB
SCP1000
DRDY
CSB
MISO
MOSI
SCK
3.3V
GND
25P20V6
94038
MMC/SD CARD
HW REV. 1.61
SD
micro SD

Pin	SD	SPI
1	CD/DAT3	CS
2	CMD	DI
3	VSS1	VSS1
4	VDD	VDD
5	CLK	SCLK
6	VSS2	VSS2
7	DAT0	DO
8	DAT1	X
9	DAT2	X

Pin	SD	SPI
1	DAT2	X
2	CD/DAT3	CS
3	CMD	DI
4	VDD	VDD
5	CLK	SCLK
6	VSS	VSS
7	DAT0	DO
8	DAT1	X

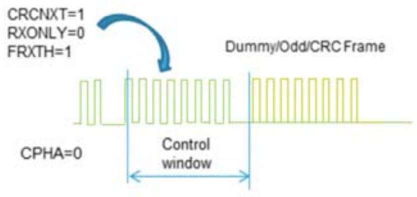
life.augmented

The SPI can be used in a wide range of applications where a simple data transfer is required without the need for a complex communication protocol. Secured transfers are also supported when used with smartcards.



Application tips and tricks

- Common tips:
  - Before disabling the SPI (or its clock), check BSY and FIFO occupancy status
  - Use DMA when a specific control is required (CRC, Rx FIFO threshold, Rx-only mode termination)
  - Packed mode should be used when data fits into a byte
  - Hardware management of NSS brings a benefit



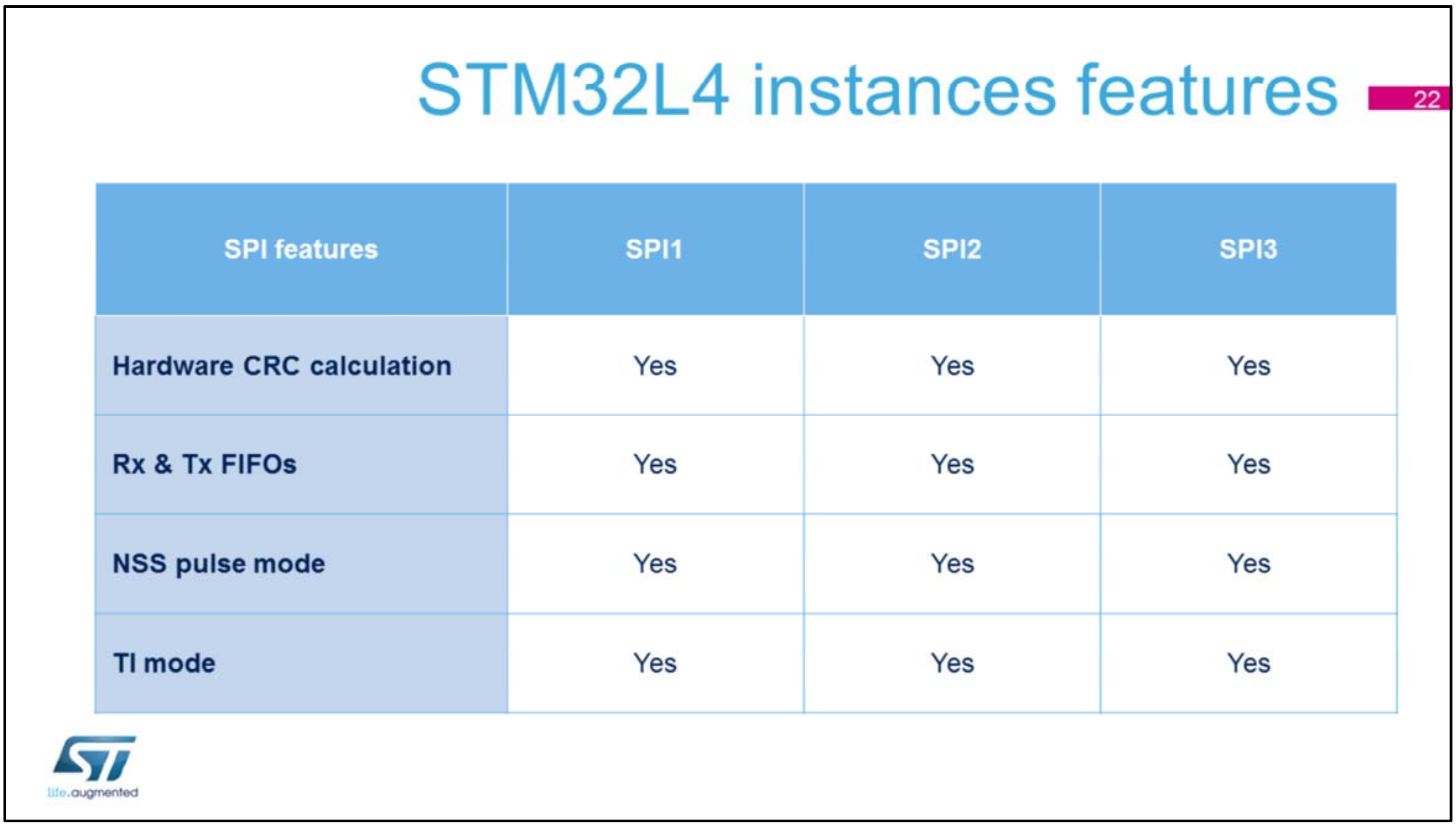
- Specific aspects:
  - CRC information is loaded into Rx FIFO, user has to flush it
  - BSY behavior is different in Master and Slave modes, during continuous data transactions
  - Receive and Overrun flags are set during transmit operations (these should be ignored)
  - DMA tips when the CRC is applied
    - Data size of DMA transactions in Full-duplex mode:
    - - Transmit = number of data to transmit (No CRC length)
    - - Receive = number of data to receive + CRC length (in Rx-only mode = number of data to receive)

life.augmented

Here are some helpful tips:
The user should be aware that traffic on the bus may still be ongoing even if the DMA transaction is completed or the transmit FIFO becomes empty.
That is why the user has to carefully check the peripheral status and follow the suggested procedures before disabling the SPI or placing it in Stop mode.
Use the DMA if you have a specific control that requires CRC handling, or receive FIFO threshold control (when the number of data is not aligned in Packet mode), or to receive an exact amount of data in Receive-only mode.
Such controls have to be applied during a specific time window and exclusively within a frame transaction so all the following transactions are handled properly.
The use of the DMA and data packing can increase the system's overall performance. These features can help when data frames are short and a fast, continuous communication flow is required.

Hardware management of Slave Select signal is not quite necessary in a single-master/single-slave pair, but it can help synchronize the data flow and prevent conflicts in a Multi-master system.

There are some additional specific aspects which should be taken into account when designing an SPI network.
The receiver always loads CRC information into the receive FIFO. The user has to account for this in the buffer and flush it.
The Busy flag should not be used for any data handling but to check for ongoing traffic. The BSY bit stays set between data frames during the master continuous data transactions. It always drops low for at least one SPI clock cycle between data frames in Slave mode, no matter if the communication is continuous or not.
When the node transmits data only, the receive flow stays active. The user should ignore all receive and associated overrun events in this case.
The data size to be processed by the DMA when including the CRC is dependent on the mode. The data size has to be configured differently when receiving or transmitting data in Full-duplex mode than in Receive-only mode.

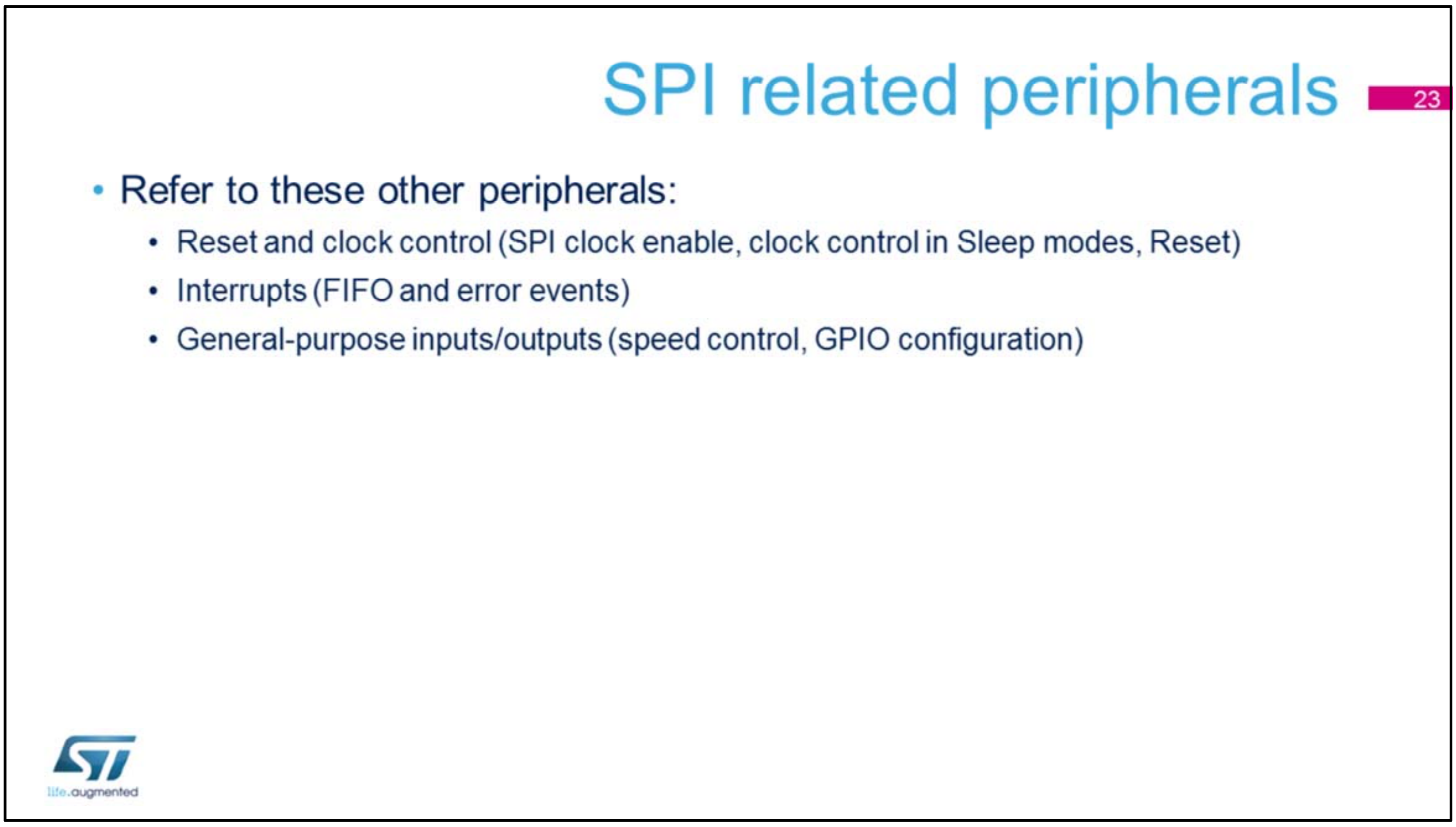


STM32L4 instances features

SPI features	SPI1	SPI2	SPI3
Hardware CRC calculation	Yes	Yes	Yes
Rx & Tx FIFOs	Yes	Yes	Yes
NSS pulse mode	Yes	Yes	Yes
TI mode	Yes	Yes	Yes

life.augmented

There are three SPI instances within the STM32L4, and each support all the features we have discussed.



SPI related peripherals

23

- Refer to these other peripherals:
  - Reset and clock control (SPI clock enable, clock control in Sleep modes, Reset)
  - Interrupts (FIFO and error events)
  - General-purpose inputs/outputs (speed control, GPIO configuration)

life.augmented

This is a list of peripherals related to the SPI. Please refer to these peripheral trainings for more information if needed.

References

- For more details, please refer to following resources
  - AN4286 - SPI protocol used in the STM32 bootloader
  - AN3364 - Migration and compatibility guidelines for STM32 microcontroller applications
  - Web (connection examples, available monitoring tools)

life.augmented

There are some dedicated SPI application notes. To learn more about the SPI generally, there are many web pages that discuss SPI topics and SPI bus monitoring tools. Many digital oscilloscopes support direct decoding of the SPI bus.

Thank you.