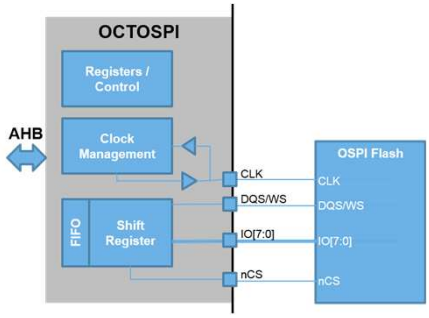
Hello, and welcome to this presentation of the STM32 OctoSPI interface that will present the features of this interface, which is widely used to connect external memories to the microcontroller.

Overview

- Provides communication interface with external serial Flash memories
  - Fully configurable from single to octal
  - Supports memory mapped read & write
  - Supports eXecute In Place (XIP)
  - Support data qualifier & write strobe



Application benefits

- Supports all SPI flash memories Single to Octal
- Only few pins needed
- Simple integration of additional memory to existing project which can be Flash or RAM

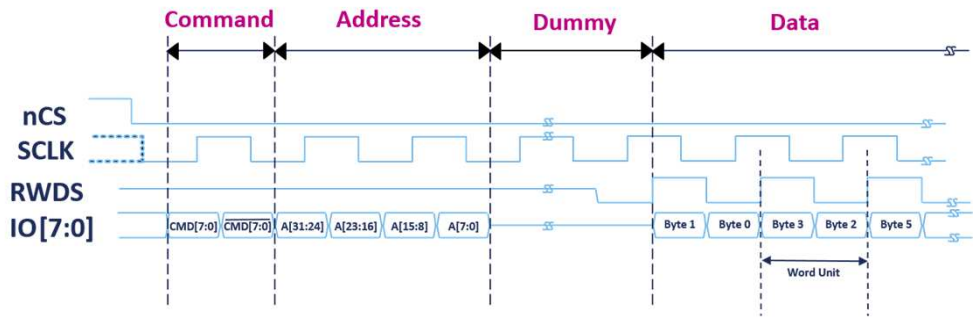
life.augmented

The OctoSPI interface integrated inside STM32 products provides a communication interface allowing the microcontroller to communicate with external single, dual, quad or octal SPI memories. This interface is fully configurable, allowing easy connection of any existing serial memories available today on the market. The external device is memory mapped which allows any system master to access it like any other memory of the system for read and write operations.
Applications will benefit from the easy connection of serial external memory, with only a few pins required. Thanks to the memory mapping feature, external memory could be simply accommodated in the existing project when more memory is needed whether it be Flash or RAM.

Regular Frame Format

Extension of the current QuadSPI

- Same configurable phases as QuadSPI
  - Each phase is individually configurable like QuadSPI IP for read and write operations
  - New 8 lane mode to support Octal Memories
  - New RWDS signal for Read & Write Data Strobe



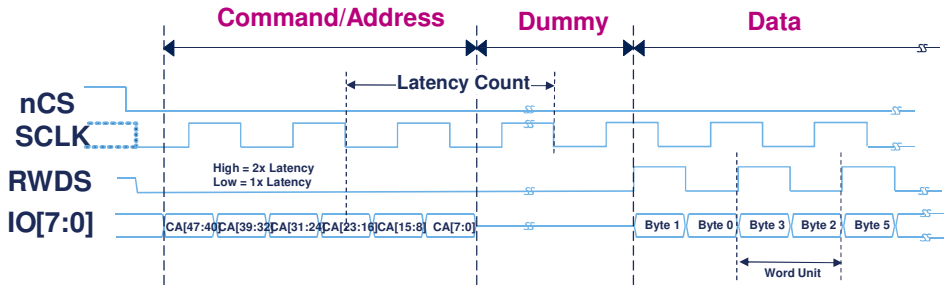
life.augmented

The OctoSPI interface offers high flexibility for frame format configuration to address any serial Flash from single data lane up to 8 data lines. As with regular QuadSPI, the user can enable or disable each of the phases, configure the length of each phase and configure the number of lines used for each phase from 1 to 8. A new signal RWDS acts as either a write strobe during write operations or a read qualifier during read operations.

Hyperbus Frame Format

Extension of the current QuadSPI

- New protocol
  - Single Command/Address Phase
  - Variable latency
  - New RWDS signal for Read & Write Data Strobe



life.augmented

The OctoSPI supports the new “Hyperbus” mode which combines the command and the addresses in a single initial phase. As with the regular frame format, Hyperbus mode also uses a read qualifier and a write strobe during the data operations. The OctoSPI supports variable or fixed external memory latency as defined by the Hyperbus protocol specification.

Key Features

- Three operating modes
  - Indirect
  - Status-polling
  - Memory-mapped
- Optimized operations up to 60MHz
  - Single data rate (SDR) support
  - Dual data rate (DDR) support

life.augmented

The OctoSPI integrated inside STM32 products offers three operating modes which will be later explained in this presentation. Communication with external memories supports single or dual data rate operation.

Operating Modes

- **Flexible operating mode to reduce CPU load**
  - Indirect mode
    - All the operations are performed through registers (classical SPI)
  - Status polling mode
    - Automatic periodical read of the Flash status registers and interrupt generation on match
  - Memory mapped mode
    - External Flash seen as internal for read operations

life.augmented

The OctoSPI supports three different modes of operation :

- Indirect mode, where it behaves as a classical SPI interface and all operations are performed through registers,
- Status polling mode, where the Flash status registers are read periodically with interrupt generation
- Memory mapped mode, where external memory is seen as if it is internal memory for read operations.

Indirect Operating Mode

Classical SPI interface

- Same usage as a classical communication IP
  - The data is transferred by writing or reading data register
  - Number of bytes is specified in the data length register
- Management of data FIFO with
  - Interrupts flag (Transfer Complete Flag)
  - DMA support
- When does transaction start ?
  - At the writing of the instruction if only instruction is needed
  - At the writing of the address if only instruction & address are needed
  - At the writing of the data when data phase are needed

life.augmented

In indirect operating mode, the OctoSPI behaves like a classical SPI interface. Transferred data goes through the data register via FIFO. Data exchange is driven by software or by the DMA, using related interrupt flags in the OctoSPI status registers.
Each command is launched by the writing of an instruction, address or data depending on the instruction context.

Status Poling Operating Mode

Reduced software overhead

- Specific mode for polling a Status Register
  - Programmable register length : 8/16/24/32-bit
  - Repeated the read operation at a defined rate
- Mask the response and generate an interrupt in case of match
  - Programmable mask (PSMKR register)
  - The masked value is compared bit per bit with the match register (PSMAR)
  - The result of the comparison can be ANDed or ORed.
  - Interrupt is generated on success (Stop on Match Flag)
- Automatic stop when a match occurs

life.augmented

A specific mode as been implemented in the OctoSPI interface to autonomously poll status registers in the external Flash. The OctoSPI interface can be configured to periodically read a register in the external Flash. The returned data can be masked to select the bits to be evaluated. The selected bits are compared with their required values stored in the match register. The result of the comparison can be treated in two ways:
- In ANDed mode, if all the selected bits are matching, an interrupt is generated.
- In Ored mode, if one of the selected bits is matching, an interrupt is generated.
When a match occurs, the OctoSPI interface can stop automatically.

Memory Mapped Mode

- **Simple extension of memory into the project**
- **Low power management**

- Prefetch for XiP
- External Flash seen as internal with wait states
  - Read & Write operations are automatically generated on AHB access
  - Frame & opcode defined during IP configuration as for indirect mode
- nCS is held low and clock is stopped to stall the OctoSPI bus and relaunch sequential read if needed
- Timeout counter to release nCS High for low power

life.augmented

The OctoSPI also provides a memory mapped mode. The main application benefit introduced by this mode is the simple integration of an external memory extension with no difference between read or write accesses of internal or externally connected memory, except the number of wait states.

This mode is suitable for both read and write operations and external memories, whether it be RAM or Flash they are seen as internal memory with wait states included to compensate for lower speed of external memory. The maximum size supported by this mode is limited to 256 MB.
A prefetch buffer supports the local execution, therefore the code could be executed directly from the external memory without the need to download it into the internal

RAM.
This mode supports also SIOO mode which is supported by some Flash memories, which allows the controller to send instructions once only and remove the instruction phase for the following accesses.

Interrupts and DMA

Interrupt event	Description
Timeout	Set when timeout occurs.
Status match	Set in automatic polling mode when the masked received data matches the corresponding bits in the match register.
FIFO threshold	Set in indirect mode when the FIFO threshold has been reached.
Transfer complete	Set in indirect mode when the programmed number of data has been transferred or in any mode when the transfer has been aborted.
Transfer error	Set in indirect mode when an invalid address is being accessed.

- DMA requests can be generated in indirect mode when THE FIFO threshold is reached.

life.augmented

The OctoSPI has 5 interrupt sources: Timeout, Status match when the masked received data matches the corresponding bits in the match register in automatic polling mode, FIFO Threshold, Transfer complete and Transfer error.
DMA requests can be generated in indirect mode when the FIFO threshold has been reached.

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 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.

life.augmented

The OctoSPI is active in Run, Sleep, Low-power run and Low-power sleep mode. An OctoSPI interrupt can cause the device to exit Sleep or Low-power sleep mode. In Stop1 or Stop2 mode, the OctoSPI is frozen, and its registers content is maintained. In Standby or Shutdown mode, the OctoSPI is powered-down and it must be reinitialized afterward.

OCTOSPI in STM32L4P5xx/Q5xx references compared to STM32L4Rxxx/Sxxx references

L4+ Product line OCTOSPI feature support summary

OCTOSPI features		STM32L4Rxxx/Sxxx	STM32L4P5xx/Q5xx
Automatic Status Polling mode		N	Y
Refresh counter for Self-Refresh (1)		N	Y
CSBOUND to support RBX (2)		N	Y
Multiplexed mode		N	Y
Single ended clock for 3V HyperBus™ mode		Y(3)	Y
Differential clock for 1V8 HyperBus™ mode		N	Y
Octal Micron memories support		N	Y
Support of QSPI and OSPI PSRAMs		N	Y
Sequential Read performances	CPU@60MHz OctoSPI@60MHz DDR	64MB/s	96 MB/s
	CPU@120MHz OctoSPI@60MHz DDR	106 MB/s	119 MB/s

(1) Self-Refresh: PSRAM are internally based on DRAM, it must be regularly refreshed (typically every 4 or 8µs). So the host must stop the transfers and release the nCS for so that the memory will refresh its internal capacitors.
(2) Row Boundary Crossing: PSRAM require to change some internal loading when changing a page. So the OctoSPI must stop the reads/writes at the exact address before the raw boundary crossing.
(3) Only HyperFlash memories are supported

life.augmented

The STM32L4Rxxx/Sxxx references provide several new important features compared to STM32L4P5xx/Q5xx :

- Automatic status polling mode : the OctoSPI automatically polls the memory status register and indicates a change to the processor via an interrupt.
- Refresh counter : most PSRAM memories require a recurrent period of time where no access is performed and Chip Select is released, in order to make their internal self-refresh. Thanks to the refresh counter, OctoSPI guarantees this without requiring SW management.
- Row Boundary crossing : PSRAM memories require specific timings when the page currently accessed is changed. The OctoSPI automatically stops in such case.
- Multiplexed mode : This mode is explained and detailed

in next slide.

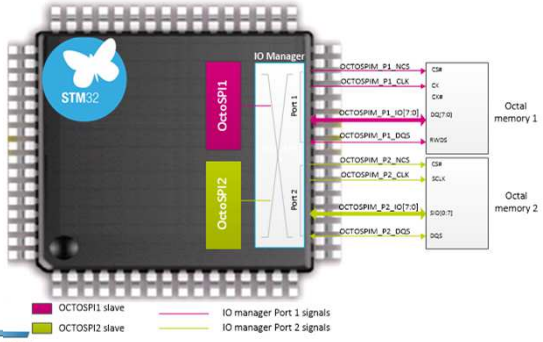
- Single ended clock for 3V mode : all memory protocols are supported
- Differential clock for 1V8 Hyperbus mode : the OctoSPI provides the differential clock in such case. The NCLK signal must be routed to pin thanks to IO multiplexing.
- Octal Micron memories support : now this type of memory - in which the data byte order is inverted - are supported
- Support of QSPI and OSPI PSRAMs : now this type of memory is supported, for both “standard” and Hyperbus protocols
- Performance : the throughput on Octal-SPI bus has been improved, thanks to the above features and sequencing optimization in the OctoSPI.

OCTOSPI Multiplexed mode

- Multiplexed mode feature enables to connect 2 Octal memories using the same interface
  - This reduces the used pins, only 12 pins instead of 22 pins
  - This reduces PCB design complexity and cost

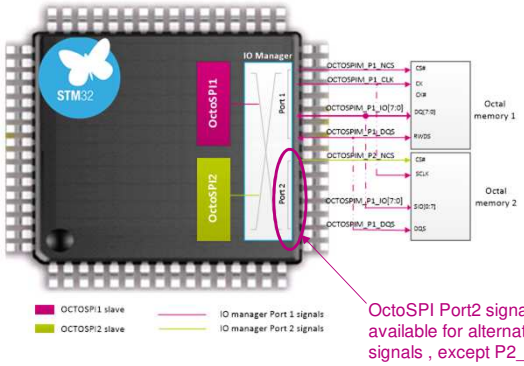
2 OctoSPI ports configuration

- OCTOSPIM_P1_NCS
- OCTOSPIM_P1_CLK
- OCTOSPIM_P1_DQS
- OCTOSPIM_P1_IO[7:0]
- OCTOSPIM_P2_NCS
- OCTOSPIM_P2_CLK
- OCTOSPIM_P2_DQS
- OCTOSPIM_P2_IO[7:0]



Multiplexed mode: requires Only 1 port and NCS of the second port

- OCTOSPIM_P1_NCS
- OCTOSPIM_P1_CLK
- OCTOSPIM_P1_DQS
- OCTOSPIM_P1_IO[7:0]
- OCTOSPIM_P2_NCS



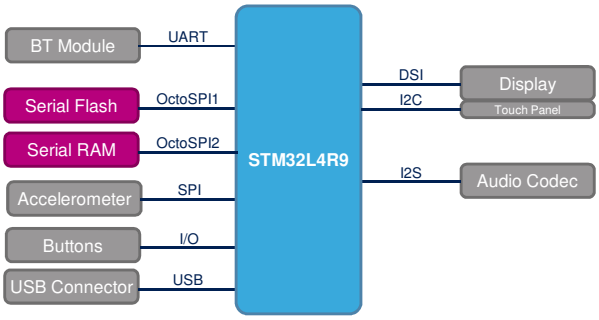
life.augmented

Multiplexed mode is a major feature supported in the STM32L4P5xx/Q5xx series.
This mode enables the communication with two external memories sharing a single Octal-SPI bus (Port1 on the right-hand picture) , simplifying PCB footprint and design.
Note that the two memories do not need to follow the same protocol : For example one memory can work in Hyperbus mode and the second one in “standard” octal bus mode.
To enable such configuration, only an extra Chip Select pin is needed to select the second memory on the bus. This also allows the release of the unused port pins for other functions.
In this mode, an internal hardware arbiter located in the IO Manager block (white block in the picture) selects alternatively the OCTOSPI1 or OCTOSPI2 depending on

their transfer requests.
This arbiter embeds time counters to limit the maximum transaction duration for each OctoSPI. This tuning of the sharing of the Octal-SPI bus bandwidth avoids the starvation of one of the OctoSPI ports.
Once the initial set-up of the time counters is done, the arbiter operation does not require any software management.

Application Examples

- Wearable applications including connectivity and HMI:



- External OctoSPI can store graphical (icons, fonts…etc…) and audio data required for HMI.

life.augmented

Wearable applications are requiring low-power management together with high quality HMI. This can be achieved using the STM32L4 OctalSPI interface to store in an external Flash all the graphical content needed like background images, high resolution icons, or fonts to support multiple languages. Additional audio data for ringtone can also benefit from the large space offered by the external Flash. The low pin count needed to drive such devices allows a very optimized system integration.

Related Peripherals

- Refer to these peripherals trainings linked to this peripheral:
  - RCC (OctoSPI clock control, OctoSPI enable/reset)
  - Interrupts (OctoSPI interrupt mapping)
  - DMA (OctoSPI data transfer)
  - GPIO (OctoSPI input/output pins)

life.augmented

You can refer to peripheral training slides related to RCC, interrupts, DMA and GPIO for additional information.

References

- For more details, please refer to following sources
  - AN5050: Octal SPI interface (OctoSPI) on STM32 MCUs

life.augmented

For more details, please have a look into the application note AN5050 about the Octal SPI interface.