Hello, and welcome to this presentation of the STM32 Quad-SPI memory interface. It covers the main features of this interface, which is widely used for connecting external memories to the microcontroller.
The Quad-SPI memory interface integrated inside STM32 products provides a communication interface, allowing the microcontroller to communicate with external SPI and Quad-SPI memories. This interface is fully configurable, allowing easy connection of any existing serial memories available today on the market. Applications can benefit from the easy connection of external serial memories, requiring only few pins. Thanks to the memory mapping feature, external memories can be simply accommodated in the existing project whenever more memory space is needed.
The Quad-SPI memory interface integrated inside STM32 products offers three operating modes and is optimized for communication with external memories with support for dual flash mode, allowing to access 8 bits in single reading cycle. It supports both single- and dual-data rate operation.
The Quad-SPI memory interface supports the connection of one or two external memories. This means that data can be transferred over a 4- or 8-bit data bus in between the memory and the microcontroller. It gives the user flexibility to choose between the number of pins required for connection (6 for a single and 10 for a double connection) and the performance of the data transfer (4 bits for a single or 8 bits for a double connection).
The Quad-SPI memory interface operates in three different modes:
1. Indirect mode, where it behaves as classical SPI interface and all operations are performed through registers,
2. Status-polling mode, where the Flash status registers are read periodically with interrupt generation,
3. Memory-mapped mode, where the external memory is seen as an internal memory for read operations.
The Quad-SPI memory interface offers high flexibility in frame format configuration. This flexibility allows addressing any serial Flash memory. Users can enable or disable each of the five phases and configure the length of each phase as well as the number of lines used for each phase.
The Quad-SPI memory interface used in indirect operating mode behaves like a classical SPI interface. Transferred data goes through the data register with FIFO. Data exchanges are driven by software or by DMA, using related interrupt flags in the Quad-SPI status registers. Each command is launched by writing the instruction, address or data, depending on the instruction context.
A specific mode has been implemented in the Quad-SPI interface to autonomously poll the status registers in the external Flash memory. The Quad-SPI interface can also be configured to periodically read a register in the external Flash memory. 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 Quad-SPI interface can stop automatically.
The Quad-SPI memory interface also has a Memory-mapped mode. The main application benefit introduced by this mode is the simple integration of an external memory extension thanks to their being no difference between the read accesses of internal or externally-connected memories, except the number of wait states. This mode is only suitable for read operations and the external Flash memory is seen as internal one with wait states included to compensate for the lower speed of the external memory. The maximum size supported by this mode is limited to 256 Mbytes.

The prefetch buffer supports execution in place, therefore code can be executed directly from the external memory without having to download it into the internal RAM.

This mode also supports SIOO mode (Send Instruction Only Once) supported by certain Flash memories, which
allows the controller to send an instruction only once and to remove the instruction phase for following accesses.
Delayed data sampling allows users to compensate for the delay of the signals due to constraints on the PCB layout optimization. It allows applications to shift the data sampling time by an additional $\frac{1}{2}$ clock cycle when operating in SDR mode. In DDR mode, the output data can be shifted by $\frac{1}{2}$ system clock cycle to relax hold constraints.
The Quad-SPI memory interface 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 is reached.
The Quad-SPI memory interface is active in Run, Sleep, Low-power run and Low-power sleep modes. A Quad-SPI interrupt can cause the device to exit Sleep or Low-power sleep modes. In Stop 0, Stop1 or Stop2 mode, the Quad-SPI is frozen and its registers content is kept. In Standby or Shutdown mode, the Quad-SPI is powered-down and it must be reinitialized afterwards.
To measure the performance of the Quad-SPI memory interface, we use a simple benchmark reading a 10-Kbyte table from the Quad-SPI Flash memory, internal Flash memory and internal SRAM. The code itself is executed from the internal Flash memory. In Single Data Rate mode at 80 MHz, the internal Flash memory is 40% faster than the external Quad-SPI Flash memory. But in Double Data Rate mode at 48 MHz, the Quad-SPI takes advantage of its prefetch mechanism and is a little bit faster than the internal Flash memory! The STM32L4’s power consumption is higher when using the Quad-SPI Flash memory because of I/O toggling.
Combining the execution time and the power consumption during the benchmark, we can obtain an overall picture of the energy budget. The DDR mode is the most optimized in terms of energy consumption for a given task.
Even if the Quad-SPI interface is optimized for data transfers, it’s possible to execute code from the external Quad-SPI Flash memory. The CoreMark results are better when using DDR mode. The consumption is equivalent to the data fetch benchmark under the same conditions.
Wearable applications require low-power management functions together with a high-quality user interface. This can be achieved using the STM32L4’s Quad-SPI interface to store in an external Flash memory all the graphical content needed including background images, high resolution icons, or fonts to support multiple languages. Additional audio data for ringtones can also benefit from the large space offered by an external Flash memory. The low pin-count needed to drive such devices allows for a highly optimized system integration.
You can refer to the Peripherals training slide related to RCC, interrupts, DMA and GPIO for additional information.
For more details, please refer to following resources

- AN4760 - Quad-SPI (QSPI) interface on STM32 microcontrollers

For more details, please refer to application note AN4760, Quad-SPI interface on STM32 microcontrollers. Thank you