Hello, and welcome to this presentation of the STM32 Quad-SPI memory interface. It covers the features of this interface, which is widely used for connecting external memories to the microcontroller.
The Quad-SPI memory interface integrated inside STM32G4 microcontrollers provides a communication interface, allowing the microcontroller to communicate with external SPI and Quad-SPI memories. The Quad-SPI memory interface supports the connection to one or two external memories. This means that data can be transferred over a 4-bit 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 two external memories can be managed separately with dedicated chip select. They can also use the same chip select and then behave as an 8-bit data memory. In such case the two memory
references must be the same.
The Quad-SPI memory interface offers three operating modes. It is optimized for communication with external memories with support for Dual-Flash mode, allowing to access 8 bits in a single reading cycle. It also supports both single- and dual-data rate operation.
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 it to address 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, using related interrupt flags in the Quad-SPI status registers, or by the DMA controller.

Each command for access to external memory 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 to consider this external memory as an internal memory from a access standpoint. This mode is only suitable for read operations. The external Flash memory is seen as internal one with more wait states due to the lower speed of the external memory. The maximum size supported by this mode is limited to 256 Mbytes.

The internal pre-fetch buffer allows optimized code execution in place: the 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 some 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 a $\frac{1}{4}$ system clock cycle to relax hold constraints. 

* - depends on part numbers
The Quad-SPI memory interface is able to generate interrupts from 5 internal events: 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 and Sleep modes. A Quad-SPI interrupt can cause the device to exit Sleep mode. In Stop mode, the Quad-SPI memory interface is frozen and its registers content is kept. In Standby mode, the Quad-SPI is powered-down and it must be reinitialized afterwards.

<table>
<thead>
<tr>
<th>Mode</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Run</td>
<td>Active.</td>
</tr>
<tr>
<td>Sleep</td>
<td>Active. Peripheral interrupts cause the device to exit Sleep mode.</td>
</tr>
<tr>
<td>Stop</td>
<td>Frozen. Peripheral registers content is kept.</td>
</tr>
<tr>
<td>Standby</td>
<td>Powered down. The peripheral must be reinitialized after exiting Standby mode.</td>
</tr>
</tbody>
</table>
Wearable applications require low-power management functions together with a high-quality user interface. This can be achieved using the STM32G4’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.
Related peripherals

- Please refer to the following trainings related to this peripheral:
  - RCC (Quad-SPI clock control, Quad-SPI enable/reset)
  - Interrupts (Quad-SPI interrupt mapping)
  - DMA (Quad-SPI data transfer)
  - GPIO (Quad-SPI input/output pins)

You can refer to the training slides related to the RCC, interrupts, DMA and GPIOs for additional information.
The Quad-SPI is also implemented in some STM32F4 and STM32L4 devices, however, the STM32F4, STM32F7 and STM32G4 offer Dual-Flash support and higher I/O speeds.

<table>
<thead>
<tr>
<th>QUADSPI features</th>
<th>STM32L4</th>
<th>STM32F0</th>
<th>STM32F1</th>
<th>STM32F2</th>
<th>STM32F3</th>
<th>STM32F4</th>
<th>STM32F7</th>
<th>STM32G4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of instances</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Maximum speed</td>
<td>48 MHz</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>90 MHz</td>
<td>100 MHz</td>
<td>133 MHz</td>
</tr>
<tr>
<td>Dual Flash</td>
<td>N</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Software compatibility</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
</tr>
</tbody>
</table>

The Quad-SPI is also implemented in some STM32F4 and STM32L4 devices, however, the STM32F4, STM32F7 and STM32G4 offer Dual-Flash support and higher I/O speeds.
For more details and additional information, please refer to the following:

- Application note AN4760: Quad-SPI (QSPI) interface on STM32 microcontrollers

For more details, please refer to the following documentation available on our website.