Welcome to the presentation of the STM32H7 DMA request multiplexer (DMAMUX). It covers the main features of this module.
The DMAMUX request multiplexer allows routing a DMA request line between the STM32H7’s peripherals and its DMA controllers. The routing function is ensured by a programmable multi-channel DMA request line multiplexer. Each channel selects a unique DMA request line, unconditionally or synchronously with events, from its DMAMUX synchronization inputs. The DMAMUX may also be used as a DMA request generator from programmable events on its input trigger signals.
DMAMUX main features:

- Up to 16-channel programmable DMA request line multiplexer output
- up to 8-channel DMA request generator
- Up to 107 DMA request lines
- Synchronous operating mode

Per DMA request generator channel:

- DMA request trigger input selector
- DMA request counter
- Event overrun flag for selected DMA request trigger input

Per DMA request line multiplexer channel output:

- Up to 107 input DMA request lines from peripherals
• One DMA request line output
• Synchronization input selector
• DMA request counter
• Event overrun flag for selected synchronization inputs
• One event output for DMA request chaining
The DMAMUX request multiplexer enables routing a DMA request line between the STM32H7’s peripherals and its DMA controllers. The routing function is ensured by a programmable multi-channel DMA request line multiplexer. Each channel selects a unique DMA request line, unconditionally or synchronously with events, from its DMAMUX synchronization inputs. The DMAMUX may also be used as a DMA request generator from programmable events on its input trigger signals. The assignment of DMAMUX request multiplexer inputs to the DMA request lines from peripherals and to the DMAMUX request generator outputs, the assignment of DMAMUX request multiplexer outputs to DMA controller channels, and the assignment of DMAMUX synchronizations and trigger inputs to internal and
external signals depend on the STM32H7’s implementation (refer to the STM32H7 reference manual).
DMAMux operating mode

Unconditionally operating mode

- When in unconditionally operating mode, the connection of one input DMA request to the multiplexer channel's output is selected through:
  - The programmed request ID number in the DMAREQ_ID field of the channel control register (DMAMUX_CxCR)
  - For each peripheral request line, an ID is assigned.
  - DMAREQ_ID = 0x00 corresponds to no DMA request line selected.

- After configuring the DMAMUX channel, the DMA controller channel to which it is routed can then be configured.
  - It is not allowed to configure two different DMAMUX channels to select the same DMA request source.

The DMAMUX request multiplexer enables routing a DMA request line between a peripherals and a DMA channel in unconditionally operating mode. When the multiplex is set, it ensures the actual routing of DMA request/acknowledge control signals.

The connection of one a peripheral request to the multiplexer channel’s output is selected through the programmed request ID in the DMAREQ_ID field of the channel control register (DMAMUX_CxCR)

- For each peripheral request line, an ID is assigned.
- DMAREQ_ID = 0x00 corresponds to no DMA request line selected.

After configuring the DMAMUX channel, the DMA controller channel to which it is routed can then be configured. It is not allowed to configure two different DMAMUX channels to
select the same DMA request source.
Each DMA request line multiplexer can individually be set to synchronous operating mode by setting the synchronization enable (SE) bit in its corresponding multiplexer channel control register (DMAMUX_CxCR). The DMA request router has multiple synchronization inputs. The synchronization inputs are connected in parallel to all multiplexer channels.

When a multiplexer channel is in synchronous operating mode, the effective connection of the selected input DMA request line to the multiplexer channel’s output is conditioned with events on the selected synchronization input and on a built-in DMA request counter. Upon the synchronization event, the selected DMA request line is connected to the multiplexer channel’s output. From this point on, each served DMA request (transition 1-to-0) on the selected DMA request line decrements the DMA request counter.
At its underrun, the DMA request counter is automatically loaded with the value in the NBREQ field of the control register and the DMA request line is disconnected from the multiplexer channel’s output. Thus, the number of DMA requests transferred to the multiplexer channel’s output following a synchronization event is the value in the NBREQ field plus one.
When the DMAMUX channel is configured in synchronous mode, its behavior is as follows.
The request multiplexer input (DMA request from the peripheral) can become active, but it will not be forwarded on the DMAMUX request multiplexer output until the synchronization signal is received. When the sync event is received, the request multiplexer connects its input and output and all the peripheral requests will be forwarded.
Each DMA request forwarded will decrement the request multiplexer counter (user programmed value). When the counter reaches zero, the connection between the DMA controller and the peripheral is cut, waiting for a new synchronization event.
For each underrun of the counter, a request multiplexer line can generate an optional event to synchronize with a second DMAMUX line. The same event can be used in some low-power scenarios to switch the system back to Stop mode.
without CPU intervention. Synchronization mode can be used to automatically synchronize data transfers with a timer for example, or to trigger the transfers on a peripheral event.
A synchronization event (edge) is detected if the state following the edge remains stable for longer than 2 AHB clock periods.

After writing to the DMAMUX channel control register (DMAMUX_CxCR), synchronization events are masked during 3 HCLK cycles.
**DMAMux operating mode**

**DMA request line multiplexer event generation mode**

- Each DMA request line multiplexer channel can individually be set in Event Generation operating mode.
- Individual enable bit (EGE bit) in the DMAMUX channel control register
- DMAMUX channel generates an event (a pulse) when its DMA request counter is automatically reloaded with the value of the corresponding NBREQ field.
- DMAMUX channel event output can be used as a synchronization event or trigger for another channel
  - This allows the request chaining on different DMA channels

---

When enabled, the multiplexer channel generates an event (a pulse) when its DMA request counter is automatically reloaded with the value of the corresponding NBREQ field. The event generator is enabled by setting the EGE bit in the control register of the corresponding multiplexer channel.
When the DMAMUX channel is in Event Generation mode, it generates an event (a pulse) when its DMA request counter is automatically reloaded. The request counter is decremented with the execution of a DMA request. The DMAMUX channel event output can be used as a synchronization event or trigger for another channel.
### DMAMux operating mode

#### DMA request generator operating mode

- When a request generator channel is enabled, it allows to produce DMA requests following trigger events.
- The outputs of DMA generator channels go to inputs of the DMA request line multiplexer.
- Each generator channel has its individual configuration register:
  - SIG_ID field corresponds to the request trigger input for generator.
  - GNBREQ field corresponds to the number of DMA requests to generate after a trigger event.
  - GPOL field corresponds to the active edge of the trigger input. Trigger events can be rising edge, falling edge or either edge on the trigger input.

On its output, the DMA request generator produces DMA requests following trigger events on DMA request trigger inputs.
The DMA request generator has multiple channels. DMA request trigger inputs are connected in parallel to all channels.
The outputs of DMA generator channels go to inputs of the DMA request line multiplexer.
Each DMA request generator channel (“generator channel” further in this section) has an enable bit.
The DMA request trigger input for generator channel \( x \) is selected through the SIG_ID field of the corresponding generator channel’s control register.
Trigger events on a DMA request trigger input can be rising edge, falling edge or either edge. The active edge is selected through the POL field of the corresponding generator channel’s control register.
When the DMAMUX is used in the DMA request generator mode, it allows performance of DMA transfers on events that are different from a peripheral DMA request, such as:
- External interrupt event,
- Comparators output,
- RTC wakeup,
- And other events (refer to STM32H7 reference manual).

A programmable DMA request counter enables the configuration of the number of requests to generate on a single trigger.
DMA request generator channel

- Upon the trigger event, the corresponding generator channel starts generating DMA requests on its output.
- Each served DMA request, after trigger event, decrements the DMA request counter. At its underrun
  - DMA request generator counter is automatically loaded with the value in the GNBREQ field of the generator control register
  - And the generator channel stops generating DMA requests

Upon the trigger event, the corresponding generator channel starts generating DMA requests on its output. Each served DMA request (transition 1-to-0) decrements a built-in DMA request counter. At its underrun, the DMA request counter is automatically loaded with the value in the GNBREQ field of the corresponding generator channel’s control register and the generator channel stops generating DMA requests. Thus, the number of DMA requests generated after the trigger event is the value in the GNBREQ field plus one.
A trigger event (edge) is detected if the state following the edge remains stable for longer than two AHB clock periods.

After writing to the DMAMUX request generator control register (DMAMUX_RGxCR), trigger events are masked during 3 HCLK cycles.
Overrun and interrupt

- An interrupt can be generated for
  - Synchronization event overrun in each DMA request line multiplexer channel
    - It happens when a new synchronization event occurs while the DMA request counter’s value is lower than the NBREQ field value
    - It sets the synchronization overrun flag OFx in the status register
    - It generates an interrupt if the synchronization overrun interrupt enable bit SOIE is set

- Trigger event overrun in each DMA request generator channel
  - It happens when a new DMA request trigger event occurs while the DMA request counter’s value is lower than the NBREQ field value
  - It sets the trigger event overrun flag OFx in the status register
  - It generates an interrupt if the DMA request trigger event’s overrun interrupt enable bit OIE is set

If a new DMA request trigger event occurs while the DMA request counter’s value is lower than the GNBREQ field value, the trigger event overrun flag OFx is set in the status register DMAMUX_RGSR of the corresponding generator channel. The overrun flag OFx is reset by setting the associated clear bit COFx, in the DMAMUX_RGCFR register of the corresponding DMA request line multiplexer channel. Setting the DMA request trigger overrun flag generates an interrupt if the DMA request trigger event’s overrun interrupt enable bit OIE is set in the control register of the corresponding generator channel.
The STM32H7 microcontroller integrates two instances of the DMA request router:

- DMAMUX1 for DMA1 and DMA2 in the D2 domain
- DMAMUX2 for BDMA in the D3 domain

<table>
<thead>
<tr>
<th>Feature</th>
<th>DMAMux1</th>
<th>DMAMux2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of DMA request line multiplexer channels</td>
<td>16</td>
<td>8</td>
</tr>
<tr>
<td>Number of DMA request generator channels</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>Number of peripheral DMA request inputs</td>
<td>107</td>
<td>12</td>
</tr>
<tr>
<td>Number of DMA request trigger inputs</td>
<td>8</td>
<td>32</td>
</tr>
<tr>
<td>Number of synchronization inputs</td>
<td>8</td>
<td>16</td>
</tr>
</tbody>
</table>
This is an example for DMAMUX inputs. For a detailed list of DMAMUX input, synchronization events and request generator triggers, please refer to the STM32H7xx reference manual.

<table>
<thead>
<tr>
<th>RQ ID</th>
<th>Resource</th>
<th>RQ ID</th>
<th>Resource</th>
<th>RQ ID</th>
<th>Resource</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Req. Gen. ch0</td>
<td>40</td>
<td>SPI2_TX</td>
<td>79</td>
<td>UART7_RX</td>
</tr>
<tr>
<td>2</td>
<td>Req. Gen. ch1</td>
<td>41</td>
<td>USART1_RX</td>
<td>80</td>
<td>UART7_TX</td>
</tr>
<tr>
<td>3</td>
<td>Req. Gen. ch2</td>
<td>42</td>
<td>USART1_TX</td>
<td>81</td>
<td>UART8_RX</td>
</tr>
<tr>
<td>4</td>
<td>Req. Gen. ch3</td>
<td>43</td>
<td>USART2_RX</td>
<td>82</td>
<td>UART8_TX</td>
</tr>
<tr>
<td>5</td>
<td>Req. Gen. ch4</td>
<td>44</td>
<td>USART2_TX</td>
<td>83</td>
<td>SPI4_RX</td>
</tr>
<tr>
<td>6</td>
<td>Req. Gen. ch5</td>
<td>45</td>
<td>USART3_RX</td>
<td>84</td>
<td>SPI4_TX</td>
</tr>
<tr>
<td>7</td>
<td>Req. Gen. ch6</td>
<td>46</td>
<td>USART3_TX</td>
<td>85</td>
<td>SPI5_RX</td>
</tr>
<tr>
<td>8</td>
<td>Req. Gen. ch7</td>
<td>47</td>
<td>TIM8_CH1</td>
<td>86</td>
<td>SPI5_TX</td>
</tr>
<tr>
<td>9</td>
<td>ADC1</td>
<td>48</td>
<td>TIM8_CH2</td>
<td>87</td>
<td>SAI1_A</td>
</tr>
<tr>
<td>10</td>
<td>ADC2</td>
<td>49</td>
<td>TIM8_CH3</td>
<td>88</td>
<td>SAI1_B</td>
</tr>
<tr>
<td></td>
<td>...</td>
<td></td>
<td>...</td>
<td></td>
<td>...</td>
</tr>
<tr>
<td>39</td>
<td>SPI2_RX</td>
<td>78</td>
<td>HASH_IN</td>
<td>115</td>
<td>ADC3</td>
</tr>
</tbody>
</table>
An interrupt can be generated for:

- A synchronization event overrun in each DMA request line multiplexer channel
- A trigger event overrun in each DMA request generator channel

In both cases, per-channel individual interrupt enable bits are available.

<table>
<thead>
<tr>
<th>Interrupt event</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOFx</td>
<td>Set when a synchronization event overrun is detected on channel x of the DMA request line multiplexer</td>
</tr>
<tr>
<td>OFx</td>
<td>Set when a Trigger event overrun is detected on channel x of the DMA request generator</td>
</tr>
</tbody>
</table>
Please refer to these trainings linked to this peripheral for more information:

- STM32H7 DMA controller (DMA)
- STM32H7 Basic DMA controller (BDMA)