is sleeping will have additional access latency, as the system needs to request and provide sufficient resources for PPI handling. Examples of such events are the following:

- A GPIO toggling and generating an event through GPIOTE
- Events generated by GRTC while all other clocks are stopped

To improve PPI latency, the Constant Latency mode can be used, see Sub-power modes on page 67.

For peripherals in different power domains, additional access latency will apply. Events that are generated while the generating or receiving power domains are sleeping will have additional access latency, as the system needs to request and provide sufficient resources for PPI handling.

6.2 DPPI — Distributed programmable peripheral interconnect

The distributed programmable peripheral interconnect (DPPI) enables peripherals to interact autonomously with each other through tasks and events, without CPU intervention. DPPI allows precise synchronization between peripherals when real-time application constraints exist, and eliminates the need for CPU involvement to implement behavior which can be predefined using the DPPI.

Note: For more information on tasks, events, publish, subscribe, interrupts, and other concepts, see Peripheral interface on page 213.

The main features of DPPI are the following:

- Peripheral tasks can subscribe to channels
- Peripheral events can be published on channels
- Publish/subscribe pattern enabling multiple connection options that include the following:
  - One-to-one
  - One-to-many
  - Many-to-one
  - Many-to-many

The DPPI consists of several PPIBus modules. These modules are connected to a fixed number of DPPI channels and a DPPI controller (DPPIC).