On the sink side of a PPIB connection, for a (local) peripheral to be able to receive this event, the corresponding PPIB channel is configured as a producer, publishing to the same DPPI channel as the (local) peripheral subscribes to, using the PPIB.PUBLISH_RECEIVE[n] register, with n the PPIB channel number.

In a PPI system, several peripherals can publish to the same DPPI channel on the source side of a PPIB connection. Similarly, several peripherals can subscribe to the same DPPI channel on the sink side of a PPIB connection. This allows multiple connection options between peripherals in different PPI systems, same as DPPI allows in a local PPI system: one-to-one, one-to-many, many-to-one and many-to-many. However, when multiple peripherals can publish to the same DPPI channel on the source side of a PPIB connection, there is a risk of overflow. See Handshake and overflow on page 118.

6.3.2 Handshake and overflow

The two PPIB instances in a PPIB connection need a handshake to transfer a peripheral event.

This is handled by a Handshake module in the PPIB. If a handshake fails because an earlier event has not been processed completely, the new event won't be sent. Instead, bit `i` in OVERFLOW.SEND register on the source side will be set, with `i` the correspoding PPIB channel number.

6.3.3 Connection examples

This section contains examples on how two connect two PPI systems using PPIB.

The following example shows how to create a PPIB connection between the TIMER10 compare event in the RADIO PD and the SAADC start task in PERI PD. PPIB11 in RADIO PD is hardwired to PPIB21 in PERI PD, which allows the PPI systems in the two separate power domains to connect. DPPI channel 0 is used by both power domains. Note that it is only necessary to use the same DPPI channel within the power domain; different DPPI channels can be used across power domains. An example of this is given further down in this section.

```
   // RADIO PD
   NRF_TIMER10->PUBLISH_COMPARE[0] = (0<<TIMER_PUBLISH_COMPARE_CHIDX_Pos) |
TIMER_PUBLISH_COMPARE_EN_Msk;
   NRF_PPIB11->SUBSCRIBE_SEND[0] = (0<<PPIB_SUBSCRIBE_SEND_CHIDX_Pos) |
PPIB_SUBSCRIBE_SEND_EN_Msk;
   NRF_DPPIC10->CHENSET = DPPIC_CHENSET_CH0_Msk;

   // PERI PD
   NRF_SAADC->SUBSCRIBE_START = (0<<SAADC_SUBSCRIBE_START_CHIDX_Pos) |
SAADC_SUBSCRIBE_START_EN_Msk;
   NRF_PPIB21->PUBLISH_RECEIVE[0] = (0<<PPIB_PUBLISH_RECEIVE_CHIDX_Pos) |
PPIB_PUBLISH_RECEIVE_EN_Msk;
   NRF_DPPIC20->CHENSET = DPPIC_CHENSET_CH0_Msk;
```

The following example shows how to create a PPIB connection between the TIMER10 compare event in the RADIO PD and the COMP start task in LP PD. The two PPI systems must be connected through PPIB instances PPIB21 and PPIB22 in PERI PD. These PPIB instances are not connected to any peripheral, only to the PPIB instances in RADIO and LP power domains. PERI PD acts as a central system that connects the two systems by means of local PPIB and DPPIC instances. This allows scaling to larger PPI systems, since multiple PPI systems can be interconnected through a central PPI system. DPPI channel 0 is used for internal RADIO PD connections, channel 1 is used by LP PD, and channel 5 is used by PERI PD. It is