Date : Mar. 4, 2025

RENESAS TECHNICAL UPDATE

TOYOSU FORESIA, 3-2-24, Toyosu, Koto-ku, Tokyo 135-0061, Japan
Renesas Electronics Corporation

Product Category	MPU/MCU		Document No.	TN-RZ*-A0140A/E	Rev.	1.00
Title	- Typo in User's manual regarding IO buffer drive strength setting register. - PCIe end point REFCLK input architecture restriction. - PCIe Auto equalization Issue. - RSPI Multi-Master mode and Open-drain output mode specification limitations. - I3C initialization setting flow correction. - Addition to xSPI AC specifications.		Information Category	Technical Notification		
Applicable Product	RZ/V2H Group	Lot No. All lots	Reference Document	RZ/V2H Group User’s Manual: Hardware Rev.1.20		

[Title]
Typo in User's manual regarding IO buffer drive strength setting register.

[Phenomenon]
There are Typos in User's manual regarding IO buffer drive strength setting and value after Reset.

[User’s manual Update]
Update the User’s manual due to typo in the setting bit/reset value of the register for IO buffer drive strength setting.

User's Manual
4.2 Pin Function Controller (PFC)
4.2.2.7 IOLH Switching Registers (PFC_IOLH_mn)
Table 4.2-13 Drive Strength Setting Values on Respective Pins and Values after Reset

[From]

Table 4.2-13 Drive Strength Setting Values on Respective Pins and Values after Reset (1/4)

Pin Name	Drive Strength Setting Value IOLH_mn bits = 00b/01b/10b/11b	Value after Reset	Pin Group*2,*3,*4
TMS_SWDIO	×1 / ×2 / ×4 / ×6	01b (×2)	A
TDO	×1 / ×2 / ×4 / ×6	01b (×2)	A
WDTUDFCA	×1 / ×2 / ×4 / ×6	01b (×2)*1	A
WDTUDFCM	×1 / ×2 / ×4 / ×6	01b (×2)*1	A
SCIF_RXD	×1 / ×2 / ×4 / ×6	01b (×2)	A
SCIF_TXD	×1 / ×2 / ×4 / ×6	01b (×2)	A
XSPI0_CKP	×1 / ×2 / ×4 / ×6	11b (×6)	B
XSPI0_CKN	×1 / ×2 / ×4 / ×6	11b (×6)	B
XSPI0_CS0N	×1 / ×2 / ×4 / ×6	11b (×6)	A
XSPI0_DS	×1 / ×2 / ×4 / ×6	11b (×6)	B
XSPI0_IO0	×1 / ×2 / ×4 / ×6	11b (×6)	B
XSPI0_IO1	×1 / ×2 / ×4 / ×6	11b (×6)	B
XSPI0_IO2	×1 / ×2 / ×4 / ×6	11b (×6)	B
XSPI0_IO3	×1 / ×2 / ×4 / ×6	11b (×6)	B
XSPI0_IO4	×1 / ×2 / ×4 / ×6	11b (×6)	B
XSPI0_IO5	×1 / ×2 / ×4 / ×6	11b (×6)	B
XSPI0_IO6	×1 / ×2 / ×4 / ×6	11b (×6)	B
XSPI0_IO7	×1 / ×2 / ×4 / ×6	11b (×6)	B
XSPI0_RESET0N	×1 / ×2 / ×4 / ×6	11b (×6)	A
SD0CLK	×1 / ×2 / ×4 / ×6	11b (×6)	B
SD0CMD	×1 / ×2 / ×4 / ×6	11b (×6)	B
SD0DAT0	×1 / ×2 / ×4 / ×6	11b (×6)	B
SD0DAT1	×1 / ×2 / ×4 / ×6	11b (×6)	B
SD0DAT2	×1 / ×2 / ×4 / ×6	11b (×6)	B
SD0DAT3	×1 / ×2 / ×4 / ×6	11b (×6)	B
SD0DAT4	×1 / ×2 / ×4 / ×6	11b (×6)	B
SD0DAT5	×1 / ×2 / ×4 / ×6	11b (×6)	B
SD0DAT6	×1 / ×2 / ×4 / ×6	11b (×6)	B
SD0DAT7	×1 / ×2 / ×4 / ×6	11b (×6)	B
SD0RSTN	×1 / ×2 / ×4 / ×6	11b (×6)	B
SD1CLK	×1 / ×2 / ×4 / ×6	11b (×6)	B
SD1CMD	×1 / ×2 / ×4 / ×6	11b (×6)	B
SD1DAT0	×1 / ×2 / ×4 / ×6	11b (×6)	B
SD1DAT1	×1 / ×2 / ×4 / ×6	11b (×6)	B
SD1DAT2	×1 / ×2 / ×4 / ×6	11b (×6)	B
SD1DAT3	×1 / ×2 / ×4 / ×6	11b (×6)	B
PCIE0_RSTOUTB	×1 / ×2 / ×4 / ×6	10b (×4)	A
PCIE1_RSTOUTB	×1 / ×2 / ×4 / ×6	10b (×4)	A
ET0_MDIO	×1 / ×2 / ×4 / ×6	10b (×4)	B
ET0_MDC	×1 / ×2 / ×4 / ×6	10b (×4)	B
ET0_TXCTL_TXEN	×1 / ×2 / ×4 / ×6	10b (×4)	B
ET0_TXER	×1 / ×2 / ×4 / ×6	10b (×4)	B

Table 4.2-13 Drive Strength Setting Values on Respective Pins and Values after Reset (2/4)

Pin Name	Drive Strength Setting Value IOLH_mn bits = 00b/01b/10b/11b	Value after Reset	Pin Group*2,*3,*4
ET0_TXC_TXCLK	×1 / ×2 / ×4 / ×6	10b (×4)	B
ET0_TXD0	×1 / ×2 / ×4 / ×6	10b (×4)	B
ET0_TXD1	×1 / ×2 / ×4 / ×6	10b (×4)	B
ET0_TXD2	×1 / ×2 / ×4 / ×6	10b (×4)	B
ET0_TXD3	×1 / ×2 / ×4 / ×6	10b (×4)	B
ET1_MDIO	×1 / ×2 / ×4 / ×6	10b (×4)	B
ET1_MDC	×1 / ×2 / ×4 / ×6	10b (×4)	B
ET1_TXCTL_TXEN	×1 / ×2 / ×4 / ×6	10b (×4)	B
ET1_TXER	×1 / ×2 / ×4 / ×6	10b (×4)	B
ET1_TXC_TXCLK	×1 / ×2 / ×4 / ×6	10b (×4)	B
ET1_TXD0	×1 / ×2 / ×4 / ×6	10b (×4)	B
ET1_TXD1	×1 / ×2 / ×4 / ×6	10b (×4)	B
ET1_TXD2	×1 / ×2 / ×4 / ×6	10b (×4)	B
ET1_TXD3	×1 / ×2 / ×4 / ×6	10b (×4)	B
P00	×1 / ×2 / ×4 / ×6	01b (×2)	A
P01	×1 / ×2 / ×4 / ×6	01b (×2)	A
P02	×1 / ×2 / ×4 / ×6	01b (×2)	A
P03	×1 / ×2 / ×4 / ×6	01b (×2)	A
P04	×1 / ×2 / ×4 / ×6	01b (×2)	A
P05	×1 / ×2 / ×4 / ×6	01b (×2)	A
P06	×1 / ×2 / ×4 / ×6	01b (×2)	A
P07	×1 / ×2 / ×4 / ×6	01b (×2)	A
P10	×1 / ×2 / ×4 / ×6	01b (×2)	A
P11	×1 / ×2 / ×4 / ×6	01b (×2)	A
P12	×1 / ×2 / ×4 / ×6	01b (×2)	A
P13	×1 / ×2 / ×4 / ×6	01b (×2)	A
P14	×1 / ×2 / ×4 / ×6	01b (×2)	A
P15	×1 / ×2 / ×4 / ×6	01b (×2)	A
P20	×1 / ×2 / ×3 / ×4	01b (×2)	D
P21	×1 / ×2 / ×3 / ×4	01b (×2)	D
P30	×1 / ×2 / ×4 / ×6	01b (×2)	A
P31	×1 / ×2 / ×4 / ×6	01b (×2)	A
P32	×1 / ×2 / ×4 / ×6	01b (×2)	A
P33	×1 / ×2 / ×4 / ×6	01b (×2)	A
P34	×1 / ×2 / ×4 / ×6	01b (×2)	A
P35	×1 / ×2 / ×4 / ×6	01b (×2)	A
P36	×1 / ×2 / ×4 / ×6	01b (×2)	A
P37	×1 / ×2 / ×4 / ×6	01b (×2)	A
P40	×1 / ×2 / ×4 / ×6	01b (×2)	A
P41	×1 / ×2 / ×4 / ×6	01b (×2)	A
P42	×1 / ×2 / ×4 / ×6	01b (×2)	A
P43	×1 / ×2 / ×4 / ×6	01b (×2)	A
P44	×1 / ×2 / ×4 / ×6	01b (×2)	A
P45	×1 / ×2 / ×4 / ×6	01b (×2)	A

Table 4.2-13 Drive Strength Setting Values on Respective Pins and Values after Reset (3/4)

Pin Name	Drive Strength Setting Value IOLH_mn bits = 00b/01b/10b/11b	Value after Reset	Pin Group*2,*3,*4
P46	×1 / ×2 / ×4 / ×6	01b (×2)	A
P47	×1 / ×2 / ×4 / ×6	01b (×2)	A
P50	×1 / ×2 / ×4 / ×6	01b (×2)	A
P51	×1 / ×2 / ×4 / ×6	01b (×2)	A
P52	×1 / ×2 / ×4 / ×6	01b (×2)	A
P53	×1 / ×2 / ×4 / ×6	01b (×2)	A
P54	×1 / ×2 / ×4 / ×6	01b (×2)	A
P55	×1 / ×2 / ×4 / ×6	01b (×2)	A
P56	×1 / ×2 / ×4 / ×6	01b (×2)	A
P57	×1 / ×2 / ×4 / ×6	01b (×2)	A
P60	×1 / ×2 / ×4 / ×6	01b (×2)	A
P61	×1 / ×2 / ×4 / ×6	01b (×2)	A
P62	×1 / ×2 / ×4 / ×6	01b (×2)	A
P63	×1 / ×2 / ×4 / ×6	01b (×2)	A
P64	×1 / ×2 / ×4 / ×6	01b (×2)	A
P65	×1 / ×2 / ×4 / ×6	01b (×2)	A
P66	×1 / ×2 / ×4 / ×6	01b (×2)	A
P67	×1 / ×2 / ×4 / ×6	01b (×2)	A
P70	×1 / ×2 / ×4 / ×6	01b (×2)	A
P71	×1 / ×2 / ×4 / ×6	01b (×2)	A
P72	×1 / ×2 / ×4 / ×6	01b (×2)	A
P73	×1 / ×2 / ×4 / ×6	01b (×2)	A
P74	×1 / ×2 / ×4 / ×6	01b (×2)	A
P75	×1 / ×2 / ×4 / ×6	01b (×2)	A
P76	×1 / ×2 / ×4 / ×6	01b (×2)	A
P77	×1 / ×2 / ×4 / ×6	01b (×2)	A
P80	×1 / ×2 / ×4 / ×6	01b (×2)	A
P81	×1 / ×2 / ×4 / ×6	01b (×2)	A
P82	×1 / ×2 / ×4 / ×6	01b (×2)	A
P83	×1 / ×2 / ×4 / ×6	01b (×2)	A
P84	×1 / ×2 / ×4 / ×6	01b (×2)	A
P85	×1 / ×2 / ×4 / ×6	01b (×2)	A
P86	×1 / ×2 / ×4 / ×6	01b (×2)	A
P87	×1 / ×2 / ×4 / ×6	01b (×2)	A
P90	×1 / ×2 / ×4 / ×6	01b (×2)	B
P91	×1 / ×2 / ×4 / ×6	01b (×2)	B
P92	×1 / ×2 / ×4 / ×6	01b (×2)	B
P93	×1 / ×2 / ×4 / ×6	01b (×2)	A
P94	×1 / ×2 / ×4 / ×6	01b (×2)	A
P95	×1 / ×2 / ×4 / ×6	01b (×2)	A
P96	×1 / ×2 / ×4 / ×6	01b (×2)	A
P97	×1 / ×2 / ×4 / ×6	01b (×2)	A
PA0	×1 / ×2 / ×4 / ×6	01b (×2)	A
PA1	×1 / ×2 / ×4 / ×6	01b (×2)	A

Table 4.2-13 Drive Strength Setting Values on Respective Pins and Values after Reset (4/4)

Pin Name	Drive Strength Setting Value IOLH_mn bits = 00b/01b/10b/11b	Value after Reset	Pin Group*2,*3,*4
PA2	×1 / ×2 / ×4 / ×6	01b (×2)	A
PA3	×1 / ×2 / ×4 / ×6	01b (×2)	A
PA4	×1 / ×2 / ×4 / ×6	01b (×2)	A
PA5	×1 / ×2 / ×4 / ×6	01b (×2)	A
PA6	×1 / ×2 / ×4 / ×6	01b (×2)	A
PA7	×1 / ×2 / ×4 / ×6	01b (×2)	A
PB0	×1 / ×2 / ×4 / ×6	01b (×2)	B
PB1	×1 / ×2 / ×4 / ×6	01b (×2)	B
PB2	×1 / ×2 / ×4 / ×6	01b (×2)	B
PB3	×1 / ×2 / ×4 / ×6	01b (×2)	B
PB4	×1 / ×2 / ×4 / ×6	01b (×2)	B
PB5	×1 / ×2 / ×4 / ×6	01b (×2)	B

Note 1. WDTUDFCA and WDTUDFCM are reset in response to the assertion of the ERROR_RESETn signal from the CPG. They are not reset in response to the assertion of the external QRESN pin.

Note 2. Group A output impedance

I/O voltage	00b	01b	10b	11b
3.3 V	150Ω	75Ω	38Ω	25Ω
1.8 V	130Ω	65Ω	33Ω	22Ω

Note 3. Group B output impedance

I/O voltage	00b	01b	10b	11b
3.3 V	65Ω	55Ω	44Ω	33Ω
1.8 V	50Ω	40Ω	33Ω	25Ω

Note 4. Group D output impedance

I/O voltage	00b	01b	10b	11b
1.8 V	110Ω	55Ω	30Ω	20Ω
1.2 V	150Ω	75Ω	38Ω	25Ω

[To]

Table 4.2-13 Drive Strength Setting Values on Respective Pins and Values after Reset (1/4)

Pin Name	Drive Strength Setting Value IOLH_mn bits = 00b/10b/01b/11b	Value after Reset	Pin Group*2,*3,*4
TMS_SWDIO	×1 / ×2 / ×4 / ×6	01b (×4)	A
TDO	×1 / ×2 / ×4 / ×6	01b (×4)	A
WDTUDFCA	×1 / ×2 / ×4 / ×6	01b (×4)*1	A
WDTUDFCM	×1 / ×2 / ×4 / ×6	01b (×4)*1	A
SCIF_RXD	×1 / ×2 / ×4 / ×6	01b (×4)	A
SCIF_TXD	×1 / ×2 / ×4 / ×6	01b (×4)	A
XSPI0_CKP	×1 / ×2 / ×4 / ×6	11b (×6)	B
XSPI0_CKN	×1 / ×2 / ×4 / ×6	11b (×6)	B
XSPI0_CS0N	×1 / ×2 / ×4 / ×6	11b (×6)	A
XSPI0_DS	×1 / ×2 / ×4 / ×6	11b (×6)	B
XSPI0_IO0	×1 / ×2 / ×4 / ×6	11b (×6)	B
XSPI0_IO1	×1 / ×2 / ×4 / ×6	11b (×6)	B
XSPI0_IO2	×1 / ×2 / ×4 / ×6	11b (×6)	B
XSPI0_IO3	×1 / ×2 / ×4 / ×6	11b (×6)	B
XSPI0_IO4	×1 / ×2 / ×4 / ×6	11b (×6)	B
XSPI0_IO5	×1 / ×2 / ×4 / ×6	11b (×6)	B
XSPI0_IO6	×1 / ×2 / ×4 / ×6	11b (×6)	B
XSPI0_IO7	×1 / ×2 / ×4 / ×6	11b (×6)	B
XSPI0_RESET0N	×1 / ×2 / ×4 / ×6	11b (×6)	A
SD0CLK	×1 / ×2 / ×4 / ×6	11b (×6)	B
SD0CMD	×1 / ×2 / ×4 / ×6	11b (×6)	B
SD0DAT0	×1 / ×2 / ×4 / ×6	11b (×6)	B
SD0DAT1	×1 / ×2 / ×4 / ×6	11b (×6)	B
SD0DAT2	×1 / ×2 / ×4 / ×6	11b (×6)	B
SD0DAT3	×1 / ×2 / ×4 / ×6	11b (×6)	B
SD0DAT4	×1 / ×2 / ×4 / ×6	11b (×6)	B
SD0DAT5	×1 / ×2 / ×4 / ×6	11b (×6)	B
SD0DAT6	×1 / ×2 / ×4 / ×6	11b (×6)	B
SD0DAT7	×1 / ×2 / ×4 / ×6	11b (×6)	B
SD0RSTN	×1 / ×2 / ×4 / ×6	11b (×6)	B
SD1CLK	×1 / ×2 / ×4 / ×6	11b (×6)	B
SD1CMD	×1 / ×2 / ×4 / ×6	11b (×6)	B
SD1DAT0	×1 / ×2 / ×4 / ×6	11b (×6)	B
SD1DAT1	×1 / ×2 / ×4 / ×6	11b (×6)	B
SD1DAT2	×1 / ×2 / ×4 / ×6	11b (×6)	B
SD1DAT3	×1 / ×2 / ×4 / ×6	11b (×6)	B
PCIE0_RSTOUTB	×1 / ×2 / ×4 / ×6	10b (×2)	A
PCIE1_RSTOUTB	×1 / ×2 / ×4 / ×6	10b (×2)	A
ET0_MDIO	×1 / ×2 / ×4 / ×6	10b (×2)	B
ET0_MDC	×1 / ×2 / ×4 / ×6	10b (×2)	B
ET0_TXCTL_TXEN	×1 / ×2 / ×4 / ×6	10b (×2)	B
ET0_TXER	×1 / ×2 / ×4 / ×6	10b (×2)	B

Table 4.2-13 Drive Strength Setting Values on Respective Pins and Values after Reset (2/4)

Pin Name	Drive Strength Setting Value IOLH_mn bits = 00b/10b/01b/11b	Value after Reset	Pin Group*2,*3,*4
ET0_TXC_TXCLK	×1 / ×2 / ×4 / ×6	10b (×2)	B
ET0_TXD0	×1 / ×2 / ×4 / ×6	10b (×2)	B
ET0_TXD1	×1 / ×2 / ×4 / ×6	10b (×2)	B
ET0_TXD2	×1 / ×2 / ×4 / ×6	10b (×2)	B
ET0_TXD3	×1 / ×2 / ×4 / ×6	10b (×2)	B
ET1_MDIO	×1 / ×2 / ×4 / ×6	10b (×2)	B
ET1_MDC	×1 / ×2 / ×4 / ×6	10b (×2)	B
ET1_TXCTL_TXEN	×1 / ×2 / ×4 / ×6	10b (×2)	B
ET1_TXER	×1 / ×2 / ×4 / ×6	10b (×2)	B
ET1_TXC_TXCLK	×1 / ×2 / ×4 / ×6	10b (×2)	B
ET1_TXD0	×1 / ×2 / ×4 / ×6	10b (×2)	B
ET1_TXD1	×1 / ×2 / ×4 / ×6	10b (×2)	B
ET1_TXD2	×1 / ×2 / ×4 / ×6	10b (×2)	B
ET1_TXD3	×1 / ×2 / ×4 / ×6	10b (×2)	B
P00	×1 / ×2 / ×4 / ×6	01b (×4)	A
P01	×1 / ×2 / ×4 / ×6	01b (×4)	A
P02	×1 / ×2 / ×4 / ×6	01b (×4)	A
P03	×1 / ×2 / ×4 / ×6	01b (×4)	A
P04	×1 / ×2 / ×4 / ×6	01b (×4)	A
P05	×1 / ×2 / ×4 / ×6	01b (×4)	A
P06	×1 / ×2 / ×4 / ×6	01b (×4)	A
P07	×1 / ×2 / ×4 / ×6	01b (×4)	A
P10	×1 / ×2 / ×4 / ×6	01b (×4)	A
P11	×1 / ×2 / ×4 / ×6	01b (×4)	A
P12	×1 / ×2 / ×4 / ×6	01b (×4)	A
P13	×1 / ×2 / ×4 / ×6	01b (×4)	A
P14	×1 / ×2 / ×4 / ×6	01b (×4)	A
P15	×1 / ×2 / ×4 / ×6	01b (×4)	A
P20	×1 / ×2 / ×3 / ×4	01b (×4)	D
P21	×1 / ×2 / ×3 / ×4	01b (×4)	D
P30	×1 / ×2 / ×4 / ×6	01b (×4)	A
P31	×1 / ×2 / ×4 / ×6	01b (×4)	A
P32	×1 / ×2 / ×4 / ×6	01b (×4)	A
P33	×1 / ×2 / ×4 / ×6	01b (×4)	A
P34	×1 / ×2 / ×4 / ×6	01b (×4)	A
P35	×1 / ×2 / ×4 / ×6	01b (×4)	A
P36	×1 / ×2 / ×4 / ×6	01b (×4)	A
P37	×1 / ×2 / ×4 / ×6	01b (×4)	A
P40	×1 / ×2 / ×4 / ×6	01b (×4)	A
P41	×1 / ×2 / ×4 / ×6	01b (×4)	A
P42	×1 / ×2 / ×4 / ×6	01b (×4)	A
P43	×1 / ×2 / ×4 / ×6	01b (×4)	A
P44	×1 / ×2 / ×4 / ×6	01b (×4)	A
P45	×1 / ×2 / ×4 / ×6	01b (×4)	A

Table 4.2-13 Drive Strength Setting Values on Respective Pins and Values after Reset (3/4)

Pin Name	Drive Strength Setting Value IOLH_mn bits = 00b/10b/01b/11b	Value after Reset	Pin Group*2,*3,*4
P46	×1 / ×2 / ×4 / ×6	01b (×4)	A
P47	×1 / ×2 / ×4 / ×6	01b (×4)	A
P50	×1 / ×2 / ×4 / ×6	01b (×4)	A
P51	×1 / ×2 / ×4 / ×6	01b (×4)	A
P52	×1 / ×2 / ×4 / ×6	01b (×4)	A
P53	×1 / ×2 / ×4 / ×6	01b (×4)	A
P54	×1 / ×2 / ×4 / ×6	01b (×4)	A
P55	×1 / ×2 / ×4 / ×6	01b (×4)	A
P56	×1 / ×2 / ×4 / ×6	01b (×4)	A
P57	×1 / ×2 / ×4 / ×6	01b (×4)	A
P60	×1 / ×2 / ×4 / ×6	01b (×4)	A
P61	×1 / ×2 / ×4 / ×6	01b (×4)	A
P62	×1 / ×2 / ×4 / ×6	01b (×4)	A
P63	×1 / ×2 / ×4 / ×6	01b (×4)	A
P64	×1 / ×2 / ×4 / ×6	01b (×4)	A
P65	×1 / ×2 / ×4 / ×6	01b (×4)	A
P66	×1 / ×2 / ×4 / ×6	01b (×4)	A
P67	×1 / ×2 / ×4 / ×6	01b (×4)	A
P70	×1 / ×2 / ×4 / ×6	01b (×4)	A
P71	×1 / ×2 / ×4 / ×6	01b (×4)	A
P72	×1 / ×2 / ×4 / ×6	01b (×4)	A
P73	×1 / ×2 / ×4 / ×6	01b (×4)	A
P74	×1 / ×2 / ×4 / ×6	01b (×4)	A
P75	×1 / ×2 / ×4 / ×6	01b (×4)	A
P76	×1 / ×2 / ×4 / ×6	01b (×4)	A
P77	×1 / ×2 / ×4 / ×6	01b (×4)	A
P80	×1 / ×2 / ×4 / ×6	01b (×4)	A
P81	×1 / ×2 / ×4 / ×6	01b (×4)	A
P82	×1 / ×2 / ×4 / ×6	01b (×4)	A
P83	×1 / ×2 / ×4 / ×6	01b (×4)	A
P84	×1 / ×2 / ×4 / ×6	01b (×4)	A
P85	×1 / ×2 / ×4 / ×6	01b (×4)	A
P86	×1 / ×2 / ×4 / ×6	01b (×4)	A
P87	×1 / ×2 / ×4 / ×6	01b (×4)	A
P90	×1 / ×2 / ×4 / ×6	01b (×4)	B
P91	×1 / ×2 / ×4 / ×6	01b (×4)	B
P92	×1 / ×2 / ×4 / ×6	01b (×4)	B
P93	×1 / ×2 / ×4 / ×6	01b (×4)	A
P94	×1 / ×2 / ×4 / ×6	01b (×4)	A
P95	×1 / ×2 / ×4 / ×6	01b (×4)	A
P96	×1 / ×2 / ×4 / ×6	01b (×4)	A
P97	×1 / ×2 / ×4 / ×6	01b (×4)	A
PA0	×1 / ×2 / ×4 / ×6	01b (×4)	A
PA1	×1 / ×2 / ×4 / ×6	01b (×4)	A

Table 4.2-13 Drive Strength Setting Values on Respective Pins and Values after Reset (4/4)

Pin Name	Drive Strength Setting Value IOLH_mn bits = 00b/10b/01b/11b	Value after Reset	Pin Group*2,*3,*4
PA2	×1 / ×2 / ×4 / ×6	01b (×4)	A
PA3	×1 / ×2 / ×4 / ×6	01b (×4)	A
PA4	×1 / ×2 / ×4 / ×6	01b (×4)	A
PA5	×1 / ×2 / ×4 / ×6	01b (×4)	A
PA6	×1 / ×2 / ×4 / ×6	01b (×4)	A
PA7	×1 / ×2 / ×4 / ×6	01b (×4)	A
PB0	×1 / ×2 / ×4 / ×6	01b (×4)	B
PB1	×1 / ×2 / ×4 / ×6	01b (×4)	B
PB2	×1 / ×2 / ×4 / ×6	01b (×4)	B
PB3	×1 / ×2 / ×4 / ×6	01b (×4)	B
PB4	×1 / ×2 / ×4 / ×6	01b (×4)	B
PB5	×1 / ×2 / ×4 / ×6	01b (×4)	B

Note 1. WDTUDFCA and WDTUDFCM are reset in response to the assertion of the ERROR_RESETn signal from the CPG. They are not reset in response to the assertion of the external QRESN pin.

Note 2. Group A output impedance

I/O voltage	00b	10b	01b	11b
3.3 V	150Ω	75Ω	38Ω	25Ω
1.8 V	130Ω	65Ω	33Ω	22Ω

Note 3. Group B output impedance

I/O voltage	00b	10b	01b	11b
3.3 V	65Ω	55Ω	44Ω	33Ω
1.8 V	50Ω	40Ω	33Ω	25Ω

Note 4. Group D output impedance

I/O voltage	00b	10b	01b	11b
1.8 V	110Ω	55Ω	30Ω	20Ω
1.2 V	150Ω	75Ω	38Ω	25Ω

[Title]
PCIe REFCLK input architecture restriction.

[Phenomenon]
The document does not specify which of the three REFCLK input architecture (Common, SRNS, or SRIS).

[User’s manual Update]
PCIe RELCLK input architecture is limited to Common Clock.

User's Manual
6.6 PCI Express 3.0 Interface (PCIe)
6.6.1 Overview
PCI Express Specification (Compliant with the PCI Express Base Specification 4.0)

[From]

PCI Express Specification (Compliant with the PCI Express Base Specification 4.0)

- PCI Express Gen1 (2.5 GT/s)/Gen2 (5.0 GT/s)/Gen3 (8.0 GT/s)
- Root Complex (RC) / Endpoint (EP) Applications, Type 0/1 Configuration Register
- Lane implementation x4 / x2 × 2ch (when Multilink is selected)
  SYS: PCIe MODE ch1 register LINK_MASTER Bit is selectable
- Data payload: up to 256 bytes; read request size: up to 512 bytes
- Virtual channels are not supported (only VC0 supported)
- Number of outstanding transfers: 1 to 8
- Dynamic control of speed up/down configuration
- Clock Power Management is not supported (P1.CPM, P2.CPM not supported)
- Power Management (ASPM supported not support L1-Substate)
- Error handling/logging (AER supported)
- Replay FIFO with ECC
- Internal Memory without parity
- Number of Support Functions: 2
- Number of DMAC channels: 8

[To]

PCI Express Specification (Compliant with the PCI Express Base Specification 4.0)

- PCI Express Gen1 (2.5 GT/s)/Gen2 (5.0 GT/s)/Gen3 (8.0 GT/s)
- Root Complex (RC) / Endpoint (EP) Applications, Type 0/1 Configuration Register
- Lane implementation x4 / x2 × 2ch (when Multilink is selected)

  SYS: PCIe MODE ch1 register LINK_MASTER Bit is selectable
- Data payload: up to 256 bytes; read request size: up to 512 bytes
- Virtual channels are not supported (only VC0 supported)
- Number of outstanding transfers: 1 to 8
- Dynamic control of speed up/down configuration
- Clock Power Management is not supported (P1.CPM, P2.CPM not supported)
- Power Management (ASPM supported not support L1-Substate)
- Error handling/logging (AER supported)
- Replay FIFO with ECC
- Internal Memory without parity
- Number of Support Functions: 2
- Number of DMAC channels: 8
- Supported Reference clock architecture.
  - Common Clock

User’s Manual (additional document)
6.6 PCI Express 3.0 Interface (PCIe)
6.6.4.1.3 AXI Bridge Register Descriptions
(38) PCIe Core Control 1 Register (PCI_RC_PCCTRL1)

[From]

(38) PCIe Core Control 1 Register (PCI_RC_PCCTRL1)

This register controls power management and LTSSM (Link Training Sequence State Machine) state transitions of the PCI Express core.

Access Size : 32 bits

Offset Address : <PCI0_base> + 0404h
<PCI1_base> + 0404h

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	BLB_RELAX_ORDERING_EN	-	-	-	-	-	UI_ENTER_L1S	-	RETURN_TO_L0	UI_RC_REJECT_ASPML1	Auto PM_Active_State_Nak	UI_ENTER_L2	UI_ENTER_TXL0S
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	RW	R	R	R	RW	R	RW	RW	RW	RW	RW	RW	RW

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	MODE_QUIESCE_GUARANTEE	UI_ENTER_TXMODE_SRIS	MODE_EQ_AUTONOMOUS	MODE_EQ_PHASE23_ENABLE	MODE_RESET_EIEOS_INTERVALL0S	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	RW	RW	RW	RW	RW	R	R	R	R	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 29	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
28	BLB_RELAX_ORDERING_EN	0h	RW	Control of RO bit of Request to be sent 0b: RO bit of Request TLP to be sent is always 0b (default) 1b: A TLP can be sent with the RO bit of the Request TLP to be sent set to 1b.
27 to 25	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
24	-	0h	RW	Reserved These bits are read as 0b. The write value should always be 0b.
23	-	0h	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
22	UI_ENTER_L1S	0h	RW	L1SubState transition permission setting 0b : L1 Substate transition disabled (default) 1b: L1 Substate transition permission (setting prohibited)
21	-	0h	RW	Reserved These bits are read as 0b. The write value should always be 0b.
20	RETURN_TO_L0	0h	RW	RC mode L1, L2 state to L0 state control (usually not used) 0b : Normal operation (default) 1b : Start return operation to L0 state when in L1 or L2 state in RC mode Cleared automatically after confirming PMU_LINKSTATE[0] =1.
19	UI_RC_REJECT_ASPML1	0h	RW	ASPM L1 transition rejection control 0b : Accept ASPM L1 transition request from EP device (default) 1b : Reject ASPM L1 transition request from EP device
18	Auto PM_Active_State_Nak	0h	RW	PM_ActiveState_Nak Message Transmission Mode for ASPM L1 Rejection Set to 1b if you want to reject ASPM L1 in RC. This Bit is automatically cleared when PM_ActiveState_Nak is automatically sent. Note: Auto-sent only once.
17	UI_ENTER_L2	0h	RW	RC mode L2 transition control Set to 1b when transitioning to L2 state in RC mode. When transitioning to the L2 state, the PCIe core must be reset by controlling the Reset register. When returning, this bit must be cleared to 0b after releasing the reset.
16	UI_ENTER_TXL0S	0h	RW	TxL0s transition control 0b: Do not perform ASPM L0s transition (default) 1b: Execute ASPM L0s transition when internal conditions are satisfied

Bit	Bit Name	Initial Value	R/W	Description
15 to 13	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
12	MODE_QUIESCE_GUARANTEE	0h	RW	Symbol6 bit6 Quiesce Guarantee control bit of TS2OS 0b: Set 0b to TS2OS (default) 1b: Set 1b to TS2OS
11	UI_ENTER_TXMODE_SRIS	0h	RW	Setting Clock Tolerance Compensation 0b: SRNS (default) 1b: SRIS (not supported)
10	MODE_EQ_AUTONOMOUS	0h	RW	Gen3 feature: Autonomous Equalization Basically only changeable during the reset period 0b: Do not use Autonomous Mechanism 1b: Use Autonomous Mechanisms
9	MODE_EQ_PHASE23_ENABLE	0h	RW	Gen3 features: Setting whether to execute EQ PHASE2 and EQ PHASE3 in RC mode (MODE_PORT=1). 0b: Do not execute EQ PHASE2/3 1b: Execute EQ PHASE2/3
8	MODE_RESET_EIEOS_INTERVALL0S	0h	RW	Gen3 features: Reset EIEOS Interval information of bit 2, symbol 6 of TS1OS transmitted in Recovery.Equalization state
7 to 4	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
3 to 0	-	All 0	RW	Reserved These bits are read as 0b. The write value should always be 0b.

[To]

(38) PCIe Core Control 1 Register (PCI_RC_PCCTRL1)

This register controls power management and LTSSM (Link Training Sequence State Machine) state transitions of the PCI Express core.

Access Size : 32 bits
Offset Address : <PCI0_base> + 0404h
<PCI1_base> + 0404h
Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	BLB_RELAX_ORDERING_EN	-	-	-	-	-	UI_ENTER_L1S	-	RETURN_TO_L0	UI_RC_REJECT_ASPML1	Auto PM_Active_State_Nak	UI_ENTER_L2	UI_ENTER_TXL0S
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	RW	R	R	R	RW	R	RW	RW	RW	RW	RW	RW	RW

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	MODE_QUIESCE_GUARANTEE	-	MODE_EQ_AUTONOMOUS	MODE_EQ_PHASE23_ENABLE	MODE_RESET_EIEOS_INTERVAL0S	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	RW	RW	RW	RW	RW	R	R	R	R	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 29	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
28	BLB_RELAX_ORDERING_EN	0h	RW	Control of RO bit of Request to be sent 0b: RO bit of Request TLP to be sent is always 0b (default) 1b: A TLP can be sent with the RO bit of the Request TLP to be sent set to 1b.
27 to 25	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
24	-	0h	RW	Reserved These bits are read as 0b. The write value should always be 0b.
23	-	0h	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
22	UI_ENTER_L1S	0h	RW	L1SubState transition permission setting 0b : L1 Substate transition disabled (default) 1b: L1 Substate transition permission (setting prohibited)
21	-	0h	RW	Reserved These bits are read as 0b. The write value should always be 0b.
20	RETURN_TO_L0	0h	RW	RC mode L1, L2 state to L0 state control (usually not used) 0b : Normal operation (default) 1b : Start return operation to L0 state when in L1 or L2 state in RC mode Cleared automatically after confirming PMU_LINKSTATE[0] =1.
19	UI_RC_REJECT_ASPML1	0h	RW	ASPM L1 transition rejection control 0b : Accept ASPM L1 transition request from EP device (default) 1b : Reject ASPM L1 transition request from EP device
18	Auto PM_Active_State_Nak	0h	RW	PM_ActiveState_Nak Message Transmission Mode for ASPM L1 Rejection Set to 1b if you want to reject ASPM L1 in RC. This Bit is automatically cleared when PM_ActiveState_Nak is automatically sent. Note: Auto-sent only once.
17	UI_ENTER_L2	0h	RW	RC mode L2 transition control Set to 1b when transitioning to L2 state in RC mode. When transitioning to the L2 state, the PCIe core must be reset by controlling the Reset register. When returning, this bit must be cleared to 0b after releasing the reset.
16	UI_ENTER_TXL0S	0h	RW	TxL0s transition control 0b: Do not perform ASPM L0s transition (default) 1b: Execute ASPM L0s transition when internal conditions are satisfied

Bit	Bit Name	Initial Value	R/W	Description
15 to 13	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
12	MODE_QUIESCE_GUARANTEE	0h	RW	Symbol6 bit6 Quiesce Guarantee control bit of TS2OS 0b: Set 0b to TS2OS (default) 1b: Set 1b to TS2OS
11	-	0h	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
10	MODE_EQ_AUTONOMOUS	0h	RW	Gen3 feature: Autonomous Equalization Basically only changeable during the reset period 0b: Do not use Autonomous Mechanism 1b: Use Autonomous Mechanisms
9	MODE_EQ_PHASE23_ENABLE	0h	RW	Gen3 features: Setting whether to execute EQ PHASE2 and EQ PHASE3 in RC mode (MODE_PORT=1). 0b: Do not execute EQ PHASE2/3 1b: Execute EQ PHASE2/3
8	MODE_RESET_EIEOS_INTERVALLOS	0h	RW	Gen3 features: Reset EIEOS Interval information of bit 2, symbol 6 of TS1OS transmitted in Recovery.Equalization state
7 to 4	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
3 to 0	-	All 0	RW	Reserved These bits are read as 0b. The write value should always be 0b.

User's Manual
6.6 PCI Express 3.0 Interface (PCIe)
6.6.4.1.4 PCI Express Configuration Register Descriptions (Type1)

[From]

(63) Link Control 3 Register (PCI_RC_LINC3)

Access Size : 32 bits

Offset Address : <PCI0_base> + 61B4h
<PCI1_base> + 61B4h

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Enable Lower SKP OS Generation Vector[6:0]							-	-	-	-	-	-	-	Link Equalization Request Interrupt Enable	Perform Equalization
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	R	R	R	R	R	R	R	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
15 to 9	Enable Lower SKP OS Generation Vector[6:0]	0h	RW	When the Link is in L0 and the bit in this field corresponding to the current Link speed is Set, SKP Ordered Sets are scheduled at the rate defined for SRNS, overriding the rate required based on the clock tolerance architecture. Bit definitions within this field are: Bit[0]: 2.5 GT/s Bit[1]: 5.0 GT/s Bit[2]: 8.0 GT/s Bit[3]: 16.0 GT/s (prohibited) Bit[6:4]: RsvdP Bits in this field are RW if the corresponding bit in the Lower SKP OS Generation Supported Speeds Vector is Set, otherwise they are permitted to be hardwired to 0. Behavior is undefined if a bit is Set in this field and the corresponding bit in the Lower SKP OS Generation Supported Speeds Vector is not Set. The default value of this field is 000 0000b.
8 to 2	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
1	Link Equalization Request Interrupt Enable	0h	RW	Link Equalization Request Interrupt Enable When Set, this bit enables the generation of an interrupt to indicate that the Link Equalization 8.0 GT/s Request bit or the Link Equalization Request 16.0 GT/s bit has been set. This bit is RW for Downstream Ports and for Upstream Ports when Crosslink Supported is 1b. This bit is not applicable and is RsvdP for Upstream Ports when the Crosslink Supported bit is 0b. The default value for this bit is 0b. RC: Default 0b EP: Fixed to 0 because Crosslink is not supported
0	Perform Equalization	0h	RW	Perform Equalization When this bit is 1b and a 1b is written to the Retrain Link bit with the Target Link Speed field set to 8.0 GT/s or higher, the Downstream Port must perform Link Equalization. This bit is RW for Downstream Ports and for Upstream Ports when Crosslink Supported is 1b. This bit is not applicable and is RsvdP for Upstream Ports when the Crosslink Supported bit is 0b. The default value is 0b. RC: Default 0b EP: Fixed to 0 because Crosslink is not supported

[To]

(63) Link Control 3 Register (PCI_RC_LINC3)

Access Size : 32 bits

Offset Address : <PCI0_base> + 61B4h
<PCI1_base> + 61B4h

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	Link Equalization Request Interrupt Enable	Perform Equalization
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	R	R	R	R	R	R	R	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
15 to 9	-	0h	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
8 to 2	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
1	Link Equalization Request Interrupt Enable	0h	RW	Link Equalization Request Interrupt Enable When Set, this bit enables the generation of an interrupt to indicate that the Link Equalization 8.0 GT/s Request bit or the Link Equalization Request 16.0 GT/s bit has been set. This bit is RW for Downstream Ports and for Upstream Ports when Crosslink Supported is 1b. This bit is not applicable and is RsvdP for Upstream Ports when the Crosslink Supported bit is 0b. The default value for this bit is 0b. RC: Default 0b EP: Fixed to 0 because Crosslink is not supported
0	Perform Equalization	0h	RW	Perform Equalization When this bit is 1b and a 1b is written to the Retrain Link bit with the Target Link Speed field set to 8.0 GT/s or higher, the Downstream Port must perform Link Equalization. This bit is RW for Downstream Ports and for Upstream Ports when Crosslink Supported is 1b. This bit is not applicable and is RsvdP for Upstream Ports when the Crosslink Supported bit is 0b. The default value is 0b. RC: Default 0b EP: Fixed to 0 because Crosslink is not supported

User's Manual (additional document)
6.6 PCI Express 3.0 Interface (PCIe)
6.6.4.2.3 AXI Bridge Register Descriptions

[From]

(35) PCIe Core Control 1 Register (PCI_EP_PCCTRL1)

This register controls power management and LTSSM (Link Training Sequence State Machine) state transitions.

Access Size : 32 bits

Offset Address : <PCI0_base> + 0404h
<PCI1_base> + 0404h

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	BLB_RELAX_ORDERING_EN	-	-	-	-	-	UI_ENTER_L1S	-	-	-	-	-	UI_ENTER_TXL0S
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	RW	R	R	R	RW	R	RW	RW	RW	RW	RW	RW	RW

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	MODE_QUIESCE_GUARANTEE	UI_ENTER_TXMODE_SRIS	MODE_EQ_AUTONOMOUS	MODE_EQ_PHASE23_ENABLE	MODE_RESET_EIEOS_INTERVALL0S	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	RW	RW	RW	RW	RW	R	R	R	R	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 29	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
28	BLB_RELAX_ORDERING_EN	0h	RW	Control of RO bit of Request to be sent 0b: RO bit of Request TLP to be sent is always 0b (default) 1b: A TLP can be sent with the RO bit of the Request TLP to be sent set to 1b.
27 to 25	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
24	-	All 0	RW	Reserved Whenever it is read, 0b is read. The write value should always be 0b.
23	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
22	UI_ENTER_L1S	0h	RW	L1SubState transition permission setting 0b : L1 Substate transition disabled (default) 1b: L1 Substate transition permission (setting prohibited)
21 to 17	-	All 0	RW	Reserved Whenever it is read, 0b is read. The write value should always be 0b.
16	UI_ENTER_TXL0S	0h	RW	TxL0s transition control 0b: Do not perform ASPM L0s transition (default) 1b: Execute ASPM L0s transition when internal conditions are satisfied
15 to 13	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
12	MODE_QUIESCE_GUARANTEE	0h	RW	Symbol6 bit6 Quiesce Guarantee control bit of TS2OS 0b: Set 0b to TS2OS (default) 1b: Set 1b to TS2OS
11	UI_ENTER_TXMODE_SRIS	0h	RW	Setting Clock Tolerance Compensation Only changeable during the reset period. 0b: SRNS (default) 1b: SRIS (not supported)
10	MODE_EQ_AUTONOMOUS	0h	RW	Gen3 feature: Autonomous Equalization Basically only changeable during the reset period 0b: Do not use Autonomous Mechanism 1b: Use Autonomous Mechanisms
9	MODE_EQ_PHASE23_ENABLE	0h	RW	Gen3 features: Setting whether to execute EQ PHASE2 and EQ PHASE3 in RC mode (MODE_PORT=1). 0b: Do not execute EQ PHASE2/3 1b: Execute EQ PHASE2/3

[To]

(35) PCIe Core Control 1 Register (PCI_EP_PCCTRL1)

This register controls power management and LTSSM (Link Training Sequence State Machine) state transitions.

Access Size : 32 bits
Offset Address : <PCI0_base> + 0404h
<PCI1_base> + 0404h
Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	BLB_RELAX_ORDERING_EN	-	-	-	-	-	UI_ENTER_L1S	-	-	-	-	-	UI_ENTER_TXL0S
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	RW	R	R	R	RW	R	RW	RW	RW	RW	RW	RW	RW

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	MODE_QUIESCE_GUARANTEE	-	MODE_EQ_AUTONOMOUS	MODE_EQ_PHASE23_ENABLE	MODE_RESET_EIEOS_INTERVAL0S	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	RW	RW	RW	RW	RW	R	R	R	R	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 29	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
28	BLB_RELAX_ORDERING_EN	0h	RW	Control of RO bit of Request to be sent 0b: RO bit of Request TLP to be sent is always 0b (default) 1b: A TLP can be sent with the RO bit of the Request TLP to be sent set to 1b.
27 to 25	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
24	-	All 0	RW	Reserved Whenever it is read, 0b is read. The write value should always be 0b.
23	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
22	UI_ENTER_L1S	0h	RW	L1SubState transition permission setting 0b : L1 Substate transition disabled (default) 1b: L1 Substate transition permission (setting prohibited)
21 to 17	-	All 0	RW	Reserved Whenever it is read, 0b is read. The write value should always be 0b.
16	UI_ENTER_TXL0S	0h	RW	TxL0s transition control 0b: Do not perform ASPM L0s transition (default) 1b: Execute ASPM L0s transition when internal conditions are satisfied
15 to 13	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
12	MODE_QUIESCE_GUARANTEE	0h	RW	Symbol6 bit6 Quiesce Guarantee control bit of TS2OS 0b: Set 0b to TS2OS (default) 1b: Set 1b to TS2OS
11	-	0h	RW	Reserved Whenever it is read, 0b is read. The write value should always be 0b.
10	MODE_EQ_AUTONOMOUS	0h	RW	Gen3 feature: Autonomous Equalization Basically only changeable during the reset period 0b: Do not use Autonomous Mechanism 1b: Use Autonomous Mechanisms
9	MODE_EQ_PHASE23_ENABLE	0h	RW	Gen3 features: Setting whether to execute EQ PHASE2 and EQ PHASE3 in RC mode (MODE_PORT=1). 0b: Do not execute EQ PHASE2/3 1b: Execute EQ PHASE2/3

User's Manual
6.6 PCI Express 3.0 Interface (PCIe)
6.6.4.2.4 PCI Express Configuration Register Descriptions (Type0)

[From]

(68) Link Control 3 Register (Function #0) (PCI_EP_LINC3_F0)

Access Size :	32 bits
Offset Address :	<PCI0_base> + 61B4h <PCI1_base> + 61B4h
Initial Value :	0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Enable Lower SKP OS Generation Vector[6:0]							-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
15 to 9	Enable Lower SKP OS Generation Vector[6:0]	0h	RW	When the Link is in L0 and the bit in this field corresponding to the current Link speed is Set, SKP Ordered Sets are scheduled at the rate defined for SRNS, overriding the rate required based on the clock tolerance architecture. Bit definitions within this field are: Bit[0]: 2.5 GT/s Bit[1]: 5.0 GT/s Bit[2]: 8.0 GT/s Bit[3]: 16.0 GT/s (prohibited) Bit[6:4]: RsvdP Bits in this field are RW if the corresponding bit in the Lower SKP OS Generation Supported Speeds Vector is Set, otherwise they are permitted to be hardwired to 0. Behavior is undefined if a bit is Set in this field and the corresponding bit in the Lower SKP OS Generation Supported Speeds Vector is not Set. The default value of this field is 000 0000b.
8 to 0	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.

[To]

(68) Link Control 3 Register (Function #0) (PCI_EP_LINC3_F0)

Access Size : 32 bits
Offset Address : <PCI0_base> + 61B4h
<PCI1_base> + 61B4h
Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
15 to 9	-	0h	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
8 to 0	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.

RENESAS

[Title]
PCIe Auto equalization Issue.

[Phenomenon]
Auto equalization does not function correctly due to a specification mismatch between Link/PHY (which may result in degraded communication quality and reduced performance).

[User's manual Update]
Add initial settings to increase reception sensitivity as a countermeasure for the Auto equalization issue.

[From]

6.6.6.1.1 Changing the Initial Values of the Registers

Of the internal registers, the initial values of the configuration registers can be changed via the AXI slave interface.

1) AXI Bridge Registers

De-asserting ARESETn following supply of CLK allows access to these registers. In the case of resetting of the PCIe core, write to the registers while the reset signal is being asserted.

2) PCIe Configuration Register

De-asserting ARESETn following supply of CLK allows access to these registers. Of the reset signals of the PCIe core, those related to the configuration register must be de-asserted.

(1) Setting the Initial Values of the Registers

The initial values of the registers listed below are 0. Set appropriate values in the registers before the start of link up.

- Device ID
- Vendor ID
- Class Code (base class/sub-class/programming interface)
- Revision ID
- Subsystem ID
- Subsystem Vendor ID

[To]

6.6.6.1.1 Changing the Initial Values of the Registers

Of the internal registers, the initial values of the configuration registers can be changed via the AXI slave interface.

1) AXI Bridge Registers

De-asserting ARESETn following supply of CLK allows access to these registers. In the case of resetting of the PCIe core, write to the registers while the reset signal is being asserted.

2) PCIe Configuration Register

De-asserting ARESETn following supply of CLK allows access to these registers. Of the reset signals of the PCIe core, those related to the configuration register must be de-asserted.

3) Physical Layer Control/Monitor Registers

De-asserting ARESETn following supply of CLK allows access to these registers. In the case of resetting of the PCIe core, write to the registers while the reset signal is being asserted. It is necessary to set the Permission Registers (<PCIn_base>*1 + 0300h) bit[1] to 1b beforehand to allow access to the PIPE_PHY Register.

Note 1. <PCIn_base>: n = 0 when 4-lane × 1, n = 0, 1 when 2-lane × 2

(1) Setting the Initial Values of the Registers

The initial values of the registers listed below are 0. Set appropriate values in the registers before the start of link up.

- Device ID: Vendor and Device ID Register (<PCIn_base>*1 + 6000h)
- Vendor ID: Vendor and Device ID Register (<PCIn_base>*1 + 6000h)
- Class Code (base class/sub-class/programming interface):
  Revision ID and Class Code Register (<PCIn_base>*1 + 6008h)
- Revision ID: Revision ID and Class Code Register (<PCIn_base>*1 + 6008h)
- Subsystem ID: Subsystem ID (Function #n) (<PCIn_base>*1 + 602Ch)
- Subsystem Vendor ID: Subsystem ID (Function #n) (<PCIn_base>*1 + 602Ch)

Note 1. <PCIn_base>: n = 0 when 4-lane × 1, n = 0, 1 when 2-lane × 2

(2) Setting the Initial Values of the Registers

Make the following settings.

(1) Set the Upstream Port 8.0 GT/s Transmitter Preset and Downstream Port 8.0 GT/s Transmitter Preset in the Lane Equalization Control Register Lane #n as follows:

- 4-lane × 1: <PCI0_base> + 61BCh to 0808_0808h (n = 0)
  <PCI0_base> + 61C0h to 0808_0808h (n = 1)
- 2-lane × 2: <PCI0_base> + 61BCh to 0808_0808h (n = 0)
  <PCI1_base> + 61BCh to 0808_0808h (n = 0)

(2) Set XCFGD Setting Resister

- 4-lane × 1: <PCI0_base> + 20D0h to 776E_EEE0h
  <PCI0_base> + 20E0h to 0000_0017h
- 2-lane × 2: <PCI0_base> + 20D0h to 776E_EEE0h
  <PCI0_base> + 20E0h to 0000_0017h

User's Manual
6.6 PCI Express 3.0 Interface (PCIe)
6.6.6.5 PCIe Initialization Procedure
Table 6.6-129 Initialization Procedure (RC)

[From]

Table 6.6-129 Initialization Procedure (RC) (1/2)

Step 1	Set the Root Complex mode by the PCI Device Type setting register. SYS: Write 0000_0001h to the SYS_PCIE_MODE_CH0 register (1024h) (Root Complex mode) SYS: Write 0000_0001h to the SYS_PCIE_MODE_CH1 register (1054h) (Root Complex mode)
Step 2	Set to the reset state. CPG: Write 0004_0000h to the CPG_RST11 register (92Ch) (PCIE_0_ARESETN (TYPE-A))
Step 3	Clock OFF setting CPG: Write 0030_0000h to the CPG_CLKON_12 register (630h) (PCIE_0_ACLK, PCIE_0_CLK_PMU)
Step 4	Set Lane mode (4 lanes or 2 lanes) 4 lanes: SYS: Set bits [9:8] to 01b in the PCIe Mode register (1060h) 2 lanes: SYS: Set bits [9:8] to 11b in the PCIe Mode register (1060h)
Step 5	Release the reset. CPG: Write 0004_0004h to the CPG_RST11 register (92Ch) (PCIE_0_ARESETN (TYPE-A))
Step 6	Set the clock output to "On". CPG: Write 0030_0000h to the CPG_CLKON_12 register (630h) (PCIE_0_ACLK, PCIE_0_CLK_PMU)
Step 7	Set to the PCIe reset state. PCI: Set bits [6:0] = 000_0000b in the PCI_RC_RESET register (310h) (RST_OUT_B, RST_PS_B, RST_LOAD_B, RST_CFG_B, RST_RSM_B, RST_GP_B, RST_B)
Step 8	Release the PCIe reset. PCI: Set bit4 = 1b and bit3 = 1b in the PCI_RC_RESET register (0310h) (RST_LOAD_B, RST_CFG_B)
Step 9*1	Setting of HWINT related registers PCI: Set bit2 = 1b in the PCI_RC_PERM register (0300h) (CFG_HWINIT_EN) (access enable setting) PCI: Write xxxx_xxxxh to the PCI_RC_VID register (6000h) (Device ID, Vendor ID) PCI: Set bit2 = 0b in the PCI_RC_PERM register (300h) (CFG_HWINIT_EN) PCI: Set bit2 = 1b in the PCI_RC_PERM register (300h) (CFG_HWINIT_EN) PCI: Write FFFF_FFDFh to the PCI_RC_RID_CC register (6008h) (Revision ID, Class Code) PCI: Set bit2 = 0b in the PCI_RC_PERM register (300h) (CFG_HWINIT_EN) PCI: Set bit2 = 1b in the PCI_RC_PERM register (300h) (CFG_HWINIT_EN) PCI: Write FFFF_FFFFh to the PCI_RC_BARMSK00L register (60A0h) PCI: Set bit2 = 0b in the PCI_RC_PERM register (300h) (CFG_HWINIT_EN) PCI: Set bit2 = 1b in the PCI_RC_PERM register (300h) (CFG_HWINIT_EN) PCI: Write FFFFF_FFFFh to the PCI_RC_BARMSK00U register (60A4h) PCI: Set bit2 = 0b in the PCI_RC_PERM register (300h) (CFG_HWINIT_EN) PCI: Set bit2 = 1b in the PCI_RC_PERM register (300h) (CFG_HWINIT_EN) PCI: Write 0000_0000h to the PCI_RC_BSIZE00_01 register (60C8h) PCI: Set bit2 = 0b in the PCI_RC_PERM register (0300h) (CFG_HWINIT_EN) (access disable setting)
Step 10	SYS setting (ALLOW_ENTER_L1) SYS: Set bit0 = 1b in the SYS_PCIE_MISC_CH0 register (1020h) SYS: Set bit0 = 1b in the SYS_PCIE_MISC_CH1 register (1050h)

RENESAS

Table 6.6-129 Initialization Procedure (RC) (2/2)

Step 11*2	Interrupt settings PCI: Write 0000_1200h to the PCI_RC_PEIS0 register (0204h) PCI: Set bit30 = 1b, bit29 = 1b, bit12 = 1b, and bit9 = 1b in the PCI_RC_PEIE0 register (0200h) (UI_LINK_WIDTH_CHANGE_DONE EN, UI_LINK_SPEED_CHANGE_DONE EN, RX_DLLP_PM_ENTER_L23 EN, DL_UpDown EN) PCI: Write 0003_0303h to the PCI_RC_PEIS1 register (020ch) PCI: Set bit17 = 1b, bit16 = 1b, bit9 = 1b, bit8 = 1b, bit1 = 1b, and bit0 = 1b in the PCI_RC_PEIE1 register (0208h) (TXB_PARITY_ERR EN, ERR_RPC_REPLAYFIFO_PERR EN, ERR_REPLAY_HIGHER_CORRECTABLE_ERROR EN, ERR_REPLAY_LOWER_CORRECTABLE_ERROR EN, ERR_REPLAY_HIGHER_UNCORRECTABLE_ERROR EN, ERR_REPLAY_LOWER_UNCORRECTABLE_ERROR EN) PCI: Write 0000_0F0Fh to the PCI_RC_AMEIS register (0214h) PCI: Set bits [11:8] = 1111b and bits [3:0] = 1111b in the PCI_RC_AMEIE register (0210h) (Write MSTERR INT EN [3:0], Read MSTERR INT EN [3:0]) PCI: Write 0000_0F03h to the PCI_RC_ASEIS1 register (0224h) PCI: Set bits [11:8] = 1111b, bit1 = 1b, and bit0 = 1b in the PCI_RC_ASEIE1 register (0220h) (Write SLVERR INT EN [3:0], Read SLVERR INT EN [1:0]) PCI: Write 010F_0000h to the PCI_RC_MSGRCVIS register (0124h) PCI: Write 0105_0000h to the PCI_RC_MSGRCVIE register (0120h)
Step 12	Release the reset. PCI: Set bit5 = 1b, bit1 = 1b, and bit0 = 1b in the PCI_RC_RESET register (0310h) (RST_PS_B, RST_GP_B, RST_B)
Step 13	Wait for 500 µs or more
Step 14	Release the reset. PCI: Set bit6 = 1b and bit2 = 1b in the PCI_RC_RESET register (0310h) (RST_OUT_B, RST_RSM_B)

Note 1. Change the setting values according to the operating conditions.

Note 2. Set the corresponding interrupt handler prior to using this function.

[To]

Table 6.6-129 Initialization Procedure (RC) (1/2)

Step 1	Set the Root Complex mode by the PCI Device Type setting register. SYS: Write 0000_0001h to the SYS_PCIE_MODE_CH0 register (1024h) (Root Complex mode) SYS: Write 0000_0001h to the SYS_PCIE_MODE_CH1 register (1054h) (Root Complex mode)
Step 2	Set to the reset state. CPG: Write 0004_0000h to the CPG_RST11 register (92Ch) (PCIE_0_ARESETN (TYPE-A))
Step 3	Clock OFF setting CPG: Write 0030_0000h to the CPG_CLKON_12 register (630h) (PCIE_0_ACLK, PCIE_0_CLK_PMU)
Step 4	Set Lane mode (4 lanes or 2 lanes) 4 lanes: SYS: Set bits [9:8] to 01b in the PCIe Mode register (1060h) 2 lanes: SYS: Set bits [9:8] to 11b in the PCIe Mode register (1060h)
Step 5	Release the reset. CPG: Write 0004_0004h to the CPG_RST11 register (92Ch) (PCIE_0_ARESETN (TYPE-A))
Step 6	Set the clock output to "On". CPG: Write 0030_0000h to the CPG_CLKON_12 register (630h) (PCIE_0_ACLK, PCIE_0_CLK_PMU)
Step 7	Set to the PCIe reset state. PCI: Set bits [6:0] = 000_0000b in the PCI_RC_RESET register (310h) (RST_OUT_B, RST_PS_B, RST_LOAD_B, RST_CFG_B, RST_RSM_B, RST_GP_B, RST_B)
Step 8	Release the PCIe reset. PCI: Set bit4 = 1b and bit3 = 1b in the PCI_RC_RESET register (0310h) (RST_LOAD_B, RST_CFG_B)
Step 9*1	Setting of HWINT and PIPE_PHY related registers PCI: Set bit2 = 1b in the PCI_RC_PERM register (300h) (CFG_HWINIT_EN) (access enable setting) PCI: Set bit1 = 1b in the PCI_RC_PERM register (300h) (PIPE PHY Register Enable) (access enable setting) PCI: Write xxxx_xxxxh to the PCI_RC_VID register (6000h) (Device ID, Vendor ID) PCI: Write FFFF_FFDFh to the PCI_RC_RID_CC register (6008h) (Revision ID, Class Code) PCI: Write FFFF_FFFFh to the PCI_RC_BARMSK00L register (60A0h) PCI: Write FFFFF_FFFFh to the PCI_RC_BARMSK00U register (60A4h) PCI: Write 0000_0000h to the PCI_RC_BSIZE00_01 register (60C8h) PCI: Write 0808_0808h to the PCI_RC_LEQCTL0 register (61BCh)*3 PCI: Write 0808_0808h to the PCI_RC_LEQCTL1 register (61C0h)*3 PCI: Write 776E_EEE0h to the PCI_PHY_XCFGD register (20D0h)*3 PCI: Write 0000_0017h to the PCI_PHY_XCFGD register (20E0h)*3 PCI: Set bit2 = 0b in the PCI_RC_PERM register (300h) (CFG_HWINIT_EN) (access disable setting) PCI: Set bit1 = 0b in the PCI_RC_PERM register (300h) (PIPE PHY Register Enable) (access disable setting)
Step 10	SYS setting (ALLOW_ENTER_L1) SYS: Set bit0 = 1b in the SYS_PCIE_MISC_CH0 register (1020h) SYS: Set bit0 = 1b in the SYS_PCIE_MISC_CH1 register (1050h)

Table 6.6-129 Initialization Procedure (RC) (2/2)

Step 11*2	Interrupt settings PCI: Write 0000_1200h to the PCI_RC_PEIS0 register (0204h) PCI: Set bit30 = 1b, bit29 = 1b, bit12 = 1b, and bit9 = 1b in the PCI_RC_PEIE0 register (0200h) (UI_LINK_WIDTH_CHANGE_DONE EN, UI_LINK_SPEED_CHANGE_DONE EN, RX_DLLP_PM_ENTER_L23 EN, DL_UpDown EN) PCI: Write 0003_0303h to the PCI_RC_PEIS1 register (020ch) PCI: Set bit17 = 1b, bit16 = 1b, bit9 = 1b, bit8 = 1b, bit1 = 1b, and bit0 = 1b in the PCI_RC_PEIE1 register (0208h) (TXB_PARITY_ERR EN, ERR_RPC_REPLAYFIFO_PERR EN, ERR_REPLAY_HIGHER_CORRECTABLE_ERROR EN, ERR_REPLAY_LOWER_CORRECTABLE_ERROR EN, ERR_REPLAY_HIGHER_UNCORRECTABLE_ERROR EN, ERR_REPLAY_LOWER_UNCORRECTABLE_ERROR EN) PCI: Write 0000_0F0Fh to the PCI_RC_AMEIS register (0214h) PCI: Set bits [11:8] = 1111b and bits [3:0] = 1111b in the PCI_RC_AMEIE register (0210h) (Write MSTERR INT EN [3:0], Read MSTERR INT EN [3:0]) PCI: Write 0000_0F03h to the PCI_RC_ASEIS1 register (0224h) PCI: Set bits [11:8] = 1111b, bit1 = 1b, and bit0 = 1b in the PCI_RC_ASEIE1 register (0220h) (Write SLVERR INT EN [3:0], Read SLVERR INT EN [1:0]) PCI: Write 010F_0000h to the PCI_RC_MSGRCVIS register (0124h) PCI: Write 0105_0000h to the PCI_RC_MSGRCVIE register (0120h)
Step 12	Release the reset. PCI: Set bit5 = 1b, bit1 = 1b, and bit0 = 1b in the PCI_RC_RESET register (0310h) (RST_PS_B, RST_GP_B, RST_B)
Step 13	Wait for 500 μs or more
Step 14	Release the reset. PCI: Set bit6 = 1b and bit2 = 1b in the PCI_RC_RESET register (0310h) (RST_OUT_B, RST_RSM_B)

Note 1. Change the setting values according to the operating conditions.

Note 2. Set the corresponding interrupt handler prior to using this function.

Note 3. For details, refer to **6.6.6.1.1 Changing the Initial Values of the Registers**.

User's Manual
6.6 PCI Express 3.0 Interface (PCIe)
6.6.6.5 PCIe Initialization Procedure
Table 6.6-130 Initialization Procedure (EP)

[From]

Table 6.6-130 Initialization Procedure (EP) (1/4)

Step 1	Set the End Point mode by the PCI Device Type setting register. SYS: Write 0000_0000h to the SYS_PCIE_MODE_CH0 register (1024h) (End Point mode) SYS: Write 0000_0000h to the SYS_PCIE_MODE_CH1 register (1054h) (End Point mode)
Step 2	Set to the reset state. CPG: Write 0004_0000h to the CPG_RST11 register (92Ch) (PCIE_0_ARESETN (TYPE-A))
Step 3	Clock OFF setting CPG: Write 0030_0000h to the CPG_CLKON_12 register (630h) (PCIE_0_ACLK, PCIE_0_CLK_PMU)
Step 4	Set Lane mode (4 lanes or 2 lanes) 4 lanes: SYS: Set bits [9:8] to 01b in the PCIe Mode register (1060h) 2 lanes: SYS: Set bits [9:8] to 11b in the PCIe Mode register (1060h)
Step 5	Release the reset. CPG: Write 0004_0004h to the CPG_RST11 register (92Ch) (PCIE_0_ARESETN (TYPE-A))
Step 6	Set the clock output to "On". CPG: Write 0030_0000h to the CPG_CLKON_12 register (630h) (PCIE_0_ACLK, PCIE_0_CLK_PMU)
Step 7	Set to the PCIe reset state. PCI: Set bits [6:0] = 000_0000b in the PCI_EP_RESET register (310h) (RST_OUT_B, RST_PS_B, RST_LOAD_B, RST_CFG_B, RST_RSM_B, RST_GP_B, RST_B)
Step 8	Release the PCIe reset. PCI: Set bit4 = 1b and bit3 = 1b in the PCI_EP_RESET register (0310h) (RST_LOAD_B, RST_CFG_B)

Table 6.6-130 Initialization Procedure (EP) (2/4)

Step 9	Setting of HWINT related registers (Function #0) PCI: Set bit2 = 1b in the PCI_EP_PERM register (0300h) (CFG_HWINIT_EN) (access enable setting) PCI: Write xxxx_xxxxh to the PCI_EP_VID_F0 register (6000h) (Device ID, Vendor ID) PCI: Set bit2 = 0b in the PCI_EP_PERM register (0300h) (CFG_HWINIT_EN) PCI: Set bit2 = 1b in the PCI_EP_PERM register (0300h) (CFG_HWINIT_EN) PCI: Write FFFF_FFFFh to the PCI_EP_RID_CC_F0 register (6008h) (Revision ID, Class Code) PCI: Set bit2 = 0b in the PCI_EP_PERM register (0300h) (CFG_HWINIT_EN) PCI: Set bit2 = 1b in the PCI_EP_PERM register (0300h) (CFG_HWINIT_EN) PCI: Write FFDF_FFFFh to the PCI_EP_SSID_F0 register (602Ch) (Subsystem ID, Subsystem Vendor ID) PCI: Set bit2 = 0b in the PCI_EP_PERM register (0300h) (CFG_HWINIT_EN) PCI: Set bit2 = 1b in the PCI_EP_PERM register (0300h) (CFG_HWINIT_EN) PCI: Write 1FFF_FFFFh to the PCI_EP_BARMSK00L_F0 register (60A0h) PCI: Set bit2 = 0b in the PCI_EP_PERM register (0300h) (CFG_HWINIT_EN) PCI: Set bit2 = 1b in the PCI_EP_PERM register (0300h) (CFG_HWINIT_EN) PCI: Write 0000_0000h to the PCI_EP_BARMSK00U_F0 register (60A4h) PCI: Set bit2 = 0b in the PCI_EP_PERM register (0300h) (CFG_HWINIT_EN) PCI: Set bit2 = 1b in the PCI_EP_PERM register (0300h) (CFG_HWINIT_EN) PCI: Write 0000_0000h to the PCI_EP_BSIZE00_01_F0 register (60C8h) PCI: Set bit2 = 0b in the PCI_EP_PERM register (0300h) (CFG_HWINIT_EN) PCI: Set bit2 = 1b in the PCI_EP_PERM register (0300h) (CFG_HWINIT_EN) PCI: Write 0000_0000h to the PCI_EP_BARMSK01L_F0 register (60A8h) PCI: Set bit2 = 0b in the PCI_EP_PERM register (0300h) (CFG_HWINIT_EN) PCI: Set bit2 = 1b in the PCI_EP_PERM register (0300h) (CFG_HWINIT_EN) PCI: Write 0000_0000h to the PCI_EP_BARMSK01U_F0 register (60ACh) PCI: Set bit2 = 0b in the PCI_EP_PERM register (0300h) (CFG_HWINIT_EN) PCI: Set bit2 = 1b in the PCI_EP_PERM register (0300h) (CFG_HWINIT_EN) PCI: Write 0000_1FFFh to the PCI_EP_BARMSK02L_F0 register (60B0h) PCI: Set bit2 = 0b in the PCI_EP_PERM register (0300h) (CFG_HWINIT_EN) PCI: Set bit2 = 1b in the PCI_EP_PERM register (0300h) (CFG_HWINIT_EN) PCI: Write 0000_0000h to the PCI_EP_BARMSK02U_F0 register (60B4h) PCI: Set bit2 = 0b in the PCI_EP_PERM register (0300h) (CFG_HWINIT_EN) PCI: Set bit2 = 1b in the PCI_EP_PERM register (0300h) (CFG_HWINIT_EN) PCI: Write 0000_0000h to the PCI_EP_BSIZE02_03_F0 register (60CCh) PCI: Set bit2 = 0b in the PCI_EP_PERM register (0300h) (CFG_HWINIT_EN) PCI: Set bit2 = 1b in the PCI_EP_PERM register (0300h) (CFG_HWINIT_EN) PCI: Write 0000_0000h to the PCI_EP_BSIZE04_05_F0 register (60D0h) PCI: Set bit2 = 0b in the PCI_EP_PERM register (0300h) (CFG_HWINIT_EN) PCI: Set bit2 = 1b in the PCI_EP_PERM register (0300h) (CFG_HWINIT_EN) PCI: Write 0000_0000h to the PCI_EC_BSIZE06_F0 register (60D4h) PCI: Set bit2 = 0b in the PCI_EP_PERM register (0300h) (CFG_HWINIT_EN) (access disable setting)

Table 6.6-130 Initialization Procedure (EP) (3/4)

Step 10*1	Setting of HWINT related registers (Function #1) PCI: Set bit2 = 1b in the PCI_EP_PERM register (0300h) (CFG_HWINIT_EN) (access enable setting) PCI: Write xxxx_xxxxh to the PCI_EP_VID_F1 register (7000h) (Device ID, Vendor ID) PCI: Set bit2 = 0b in the PCI_EP_PERM register (0300h) (CFG_HWINIT_EN) PCI: Set bit2 = 1b in the PCI_EP_PERM register (0300h) (CFG _HWINIT_EN) PCI: Write FFFF_FFFFh to the PCI_EP_RID_CC_F1 register (7008h) (Revision ID, Class Code) PCI: Set bit2 = 0b in the PCI_EP_PERM register (0300h) (CFG_HWINIT_EN) PCI: Set bit2 = 1b in the PCI_EP_PERM register (0300h) (CFG_HWINIT_EN) PCI: Write FDDF_FFFFh to the PCI_EP_SSID_F1 register (702Ch) (Subsystem ID, Subsystem Vendor ID) PCI: Set bit2 = 0b in the PCI_EP_PERM register (0300h) (CFG_HWINIT_EN) PCI: Set bit2 = 1b in the PCI_EP_PERM register (0300h) (CFG_HWINIT_EN) PCI: Write 1FFF_FFFFh to the PCI_EP_BARMSK00L_F1 register (70A0h) PCI: Set bit2 = 0b in the PCI_EP_PERM register (0300h) (CFG_HWINIT_EN) PCI: Set bit2 = 1b in the PCI_EP_PERM register (0300h) (CFG_HWINIT_EN) PCI: Write 0000_0000h to the PCI_EP_BARMSK00U_F1 register (70A4h) PCI: Set bit2 = 0b in the PCI_EP_PERM register (0300h) (CFG_HWINIT_EN) PCI: Set bit2 = 1b in the PCI_EP_PERM register (0300h) (CFG_HWINIT_EN) PCI: Write 0000_0000h to the PCI_EP_BSIZE00_01_F1 register (70C8h) PCI: Set bit2 = 0b in the PCI_EP_PERM register (0300h) (CFG_HWINIT_EN) PCI: Set bit2 = 1b in the PCI_EP_PERM register (0300h) (CFG_HWINIT_EN) PCI: Write 0000_0000h to the PCI_EP_BARMSK01L_F1 register (70A8h) PCI: Set bit2 = 0b in the PCI_EP_PERM register (0300h) (CFG_HWINIT_EN) PCI: Set bit2 = 1b in the PCI_EP_PERM register (0300h) (CFG_HWINIT_EN) PCI: Write 0000_0000h to the PCI_EP_BARMSK01U_F1 register (70ACh) PCI: Set bit2 = 0b in the PCI_EP_PERM register (0300h) (CFG_HWINIT_EN) PCI: Set bit2 = 1b in the PCI_EP_PERM register (0300h) (CFG_HWINIT_EN) PCI: Write 0000_1FFFh to the PCI_EP_BARMSK02L_F1 register (70B0h) PCI: Set bit2 = 0b in the PCI_EP_PERM register (0300h) (CFG_HWINIT_EN) PCI: Set bit2 = 1b in the PCI_EP_PERM register (0300h) (CFG_HWINIT_EN) PCI: Write 0000_0000h to the PCI_EP_BARMSK02U_F1 register (70B4h) PCI: Set bit2 = 0b in the PCI_EP_PERM register (0300h) (CFG_HWINIT_EN) PCI: Set bit2 = 1b in the PCI_EP_PERM register (0300h) (CFG_HWINIT_EN) PCI: Write 0000_0000h to the PCI_EP_BSIZE02_03_F1 register (70CCh) PCI: Set bit2 = 0b in the PCI_EP_PERM register (0300h) (CFG_HWINIT_EN) PCI: Set bit2 = 1b in the PCI_EP_PERM register (0300h) (CFG_HWINIT_EN) PCI: Write 0000_0000h to the PCI_EP_BSIZE04_05_F1 register (70D0h) PCI: Set bit2 = 0b in the PCI_EP_PERM register (0300h) (CFG_HWINIT_EN) PCI: Set bit2 = 1b in the PCI_EP_PERM register (0300h) (CFG_HWINIT_EN) PCI: Write 0000_0000h to the PCI_EP_BSIZE06_F1 register (70D4h) PCI: Set bit2 = 0b in the PCI_EP_PERM register (0300h) (CFG_HWINIT_EN) (access disable setting)
Step 11	SYS setting (ALLOW_ENTER_L1) SYS: Set bit0 = 1b in the SYS_PCIE_MISC_CH0 register (1020h) SYS: Set bit0 = 1b in the SYS_PCIE_MISC_CH1 register (1050h)

Table 6.6-130 Initialization Procedure (EP) (4/4)

Step 14*2	Interrupt settings PCI: Write 0000_1200h to the PCI_EP_PEIS0 register (0204h) PCI: Set bit30 = 1b, bit29 = 1b, bit12 = 1b, and bit9 = 1b in the PCI_EP_PEIE0 register (0200h) (UI_LINK_WIDTH_CHANGE_DONE EN, UI_LINK_SPEED_CHANGE_DONE EN, RX_DLLP_PM_ENTER_L23 EN, DL_UpDown EN) PCI: Write 0003_0303h to the PCI_EP_PEIS1 register (020Ch) PCI: Set bit17 = 1b, bit16 = 1b, bit9 = 1b, bit8 = 1b, bit1 = 1b, and bit0 = 1b in the PCI_EP_PEIE1 register (0208h) (TXB_PARITY_ERR EN, ERR_RPC_REPLAYFIFO_PERR EN, ERR_REPLAY_HIGHER_CORRECTABLE_ERROR EN, ERR_REPLAY_LOWER_CORRECTABLE_ERROR EN, ERR_REPLAY_HIGHER_UNCORRECTABLE_ERROR EN, ERR_REPLAY_LOWER_UNCORRECTABLE_ERROR EN) PCI: Write 0000_0F0Fh to the PCI_EP_AMEIS register (0214h) PCI: Set bits [11:8] = 1111b and bits [3:0] = 1111b in the PCI_EP_AMEIE register (0210h) (Write MSTERR INT EN [3:0], Read MSTERR INT EN [3:0]) PCI: Write 0000_0F03h to the PCI_EP_ASEIS1 register (0224h) PCI: Set bits [11:8] = 1111b, bit1 = 1b, and bit0 = 1b in the PCI_EP_ASEIE1 register (0220h) (Write SLVERR INT EN [3:0], Read SLVERR INT EN [1:0]) PCI: Write 010F_0000h to the PCI_EP_MSGRCVIS register (0124h) PCI: Write 010A_0000h to the PCI_EP_MSGRCVIE register (0120h)
Step 15	Release the reset. PCI: Set bit5 = 1b, bit1 = 1b, and bit0 = 1b in the PCI_EP_RESET register (0310h) (RST_PS_B, RST_GP_B, RST_B)
Step 16	Wait for 500 μs or more
Step 17	Release the reset. PCI: Set bit6 = 1b and bit2 = 1b in the PCI_EP_RESET register (0310h) (RST_OUT_B, RST_RSM_B)

Note 1. Change the setting values according to the operating conditions.

Note 2. Set the corresponding interrupt handler prior to using this function.

[To]

Table 6.6-130 Initialization Procedure (EP) (1/2)

Step 1	Set the End Point mode by the PCI Device Type setting register. SYS: Write 0000_0000h to the SYS_PCIE_MODE_CH0 register (1024h) (End Point mode) SYS: Write 0000_0000h to the SYS_PCIE_MODE_CH1 register (1054h) (End Point mode)
Step 2	Set to the reset state. CPG: Write 0004_0000h to the CPG_RST11 register (92Ch) (PCIE_0_ARESETN (TYPE-A))
Step 3	Clock OFF setting CPG: Write 0030_0000h to the CPG_CLKON_12 register (630h) (PCIE_0_ACLK, PCIE_0_CLK_PMU)
Step 4	Set Lane mode (4 lanes or 2 lanes) 4 lanes: SYS: Set bits [9:8] to 01b in the PCIe Mode register (1060h) 2 lanes: SYS: Set bits [9:8] to 11b in the PCIe Mode register (1060h)
Step 5	Release the reset. CPG: Write 0004_0004h to the CPG_RST11 register (92Ch) (PCIE_0_ARESETN (TYPE-A))
Step 6	Set the clock output to "On". CPG: Write 0030_0000h to the CPG_CLKON_12 register (630h) (PCIE_0_ACLK, PCIE_0_CLK_PMU)
Step 7	Set to the PCIe reset state. PCI: Set bits [6:0] = 000_0000b in the PCI_EP_RESET register (310h) (RST_OUT_B, RST_PS_B, RST_LOAD_B, RST_CFG_B, RST_RSM_B, RST_GP_B, RST_B)
Step 8	Release the PCIe reset. PCI: Set bit4 = 1b and bit3 = 1b in the PCI_EP_RESET register (0310h) (RST_LOAD_B, RST_CFG_B)
Step 9	Setting of HWINT and PIPE_PHY related registers (Function #0) PCI: Set bit2 = 1b in the PCI_EP_PERM register (300h) (CFG_HWINIT_EN) (access enable setting) PCI: Set bit1 = 1b in the PCI_EP_PERM register (300h) (PIPE PHY Register Enable) (access enable setting) PCI: Write xxxx_xxxxh to the PCI_EP_VID_F0 register (6000h) (Device ID, Vendor ID) PCI: Write FFFF_FFFFh to the PCI_EP_RID_CC_F0 register (6008h) (Revision ID, Class Code) PCI: Write FFDF_FFFFh to the PCI_EP_SSID_F0 register (602Ch) (Subsystem ID, Subsystem Vendor ID) PCI: Write 1FFF_FFFFh to the PCI_EP_BARMSK00L_F0 register (60A0h) PCI: Write 0000_0000h to the PCI_EP_BARMSK00U_F0 register (60A4h) PCI: Write 0000_0000h to the PCI_EP_BSIZE00_01_F0 register (60C8h) PCI: Write 0000_0000h to the PCI_EP_BARMSK01L_F0 register (60A8h) PCI: Write 0000_0000h to the PCI_EP_BARMSK01U_F0 register (60ACh) PCI: Write 0000_1FFFh to the PCI_EP_BARMSK02L_F0 register (60B0h) PCI: Write 0000_0000h to the PCI_EP_BARMSK02U_F0 register (60B4h) PCI: Write 0000_0000h to the PCI_EP_BSIZE02_03_F0 register (60CCh) PCI: Write 0000_0000h to the PCI_EP_BSIZE04_05_F0 register (60D0h) PCI: Write 0000_0000h to the PCI_EP_BSIZE06_F0 register (60D4h) PCI: Write 0808_0808h to the PCI_EP_LEQCTL0_F0 register (61BCh)*3 PCI: Write 0808_0808h to the PCI_EP_LEQCTL1_F0 register (61C0h)*3 PCI: Write 776E_EEE0h to the PCI_PHY_XCFGD register (20D0h)*3 PCI: Write 0000_0017h to the PCI_PHY_XCFGD register (20E0h)*3 PCI: Set bit2 = 0b in the PCI_EP_PERM register (300h) (CFG_HWINIT_EN) (access disable setting) PCI: Set bit1 = 0b in the PCI_EP_PERM register (300h) (PIPE PHY Register Enable) (access disable setting)

Table 6.6-130 Initialization Procedure (EP) (2/2)

Step 10*1	Setting of HWINT related registers (Function #1) PCI: Set bit2 = 1b in the PCI_EP_PERM register (300h) (CFG_HWINIT_EN) (access enable setting) PCI: Write xxxx_xxxxh to the PCI_EP_VID_F1 register (7000h) (Device ID, Vendor ID) PCI: Write FFFF_FFFFh to the PCI_EP_RID_CC_F1 register (7008h) (Revision ID, Class Code) PCI: Write FDDF_FFFFh to the PCI_EP_SSID_F1 register (702Ch) (Subsystem ID, Subsystem Vendor ID) PCI: Write 1FFF_FFFFh to the PCI_EP_BARMSK00L_F1 register (70A0h) PCI: Write 0000_0000h to the PCI_EP_BARMSK00U_F1 register (70A4h) PCI: Write 0000_0000h to the PCI_EP_BSIZE00_01_F1 register (70C8h) PCI: Write 0000_0000h to the PCI_EP_BARMSK01L_F1 register (70A8h) PCI: Write 0000_0000h to the PCI_EP_BARMSK01U_F1 register (70ACh) PCI: Write 0000_1FFFh to the PCI_EP_BARMSK02L_F1 register (70B0h) PCI: Write 0000_0000h to the PCI_EP_BARMSK02U_F1 register (70B4h) PCI: Write 0000_0000h to the PCI_EP_BSIZE02_03_F1 register (70CCh) PCI: Write 0000_0000h to the PCI_EP_BSIZE04_05_F1 register (70D0h) PCI: Write 0000_0000h to the PCI_EP_BSIZE06_F1 register (70D4h) PCI: Set bit2 = 0b in the PCI_EP_PERM register (300h) (CFG_HWINIT_EN) (access disable setting)
Step 11	SYS setting (ALLOW_ENTER_L1) SYS: Set bit0 = 1b in the SYS_PCIE_MISC_CH0 register (1020h) SYS: Set bit0 = 1b in the SYS_PCIE_MISC_CH1 register (1050h)
Step 12*2	Interrupt settings PCI: Write 0000_1200h to the PCI_EP_PEIS0 register (0204h) PCI: Set bit30 = 1b, bit29 = 1b, bit12 = 1b, and bit9 = 1b in the PCI_EP_PEIE0 register (0200h) (UI_LINK_WIDTH_CHANGE_DONE EN, UI_LINK_SPEED_CHANGE_DONE EN, RX_DLLP_PM_ENTER_L23 EN, DL_UpDown EN) PCI: Write 0003_0303h to the PCI_EP_PEIS1 register (020Ch) PCI: Set bit17 = 1b, bit16 = 1b, bit9 = 1b, bit8 = 1b, bit1 = 1b, and bit0 = 1b in the PCI_EP_PEIE1 register (0208h) (TXB_PARITY_ERR EN, ERR_RPC_REPLAYFIFO_PERR EN, ERR_REPLAY_HIGHER_CORRECTABLE_ERROR EN, ERR_REPLAY_LOWER_CORRECTABLE_ERROR EN, ERR_REPLAY_HIGHER_UNCORRECTABLE_ERROR EN, ERR_REPLAY_LOWER_UNCORRECTABLE_ERROR EN) PCI: Write 0000_0F0Fh to the PCI_EP_AMEIS register (0214h) PCI: Set bits [11:8] = 1111b and bits [3:0] = 1111b in the PCI_EP_AMEIE register (0210h) (Write MSTERR INT EN [3:0], Read MSTERR INT EN [3:0]) PCI: Write 0000_0F03h to the PCI_EP_ASEIS1 register (0224h) PCI: Set bits [11:8] = 1111b, bit1 = 1b, and bit0 = 1b in the PCI_EP_ASEIE1 register (0220h) (Write SLVERR INT EN [3:0], Read SLVERR INT EN [1:0]) PCI: Write 010F_0000h to the PCI_EP_MSGRCVIS register (0124h) PCI: Write 010A_0000h to the PCI_EP_MSGRCVIE register (0120h)
Step 13	Release the reset. PCI: Set bit5 = 1b, bit1 = 1b, and bit0 = 1b in the PCI_EP_RESET register (0310h) (RST_PS_B, RST_GP_B, RST_B)
Step 14	Wait for 500 μs or more
Step 15	Release the reset. PCI: Set bit6 = 1b and bit2 = 1b in the PCI_EP_RESET register (0310h) (RST_OUT_B, RST_RSM_B)

Note 1. Change the setting values according to the operating conditions.

Note 2. Set the corresponding interrupt handler prior to using this function.

Note 3. For details, refer to 6.6.6.1.1 Changing the Initial Values of the Registers.

[Title]
RSPI Multi-Master mode and Open-drain output mode specification limitations.

[Phenomenon]
In Multi-Master mode and Open-drain output mode, RSPI SSL1-3 pins do not enter the Open-drain state.

[User's manual Update]
Remove the specifications for Multi-Master mode and Open-drain output mode.

User's Manual
7.5 Serial Peripheral Interface (RSPI)
7.5.1.1 Features
Table 7.5-1 SPI Specifications (1/3)

[From]

Table 7.5-1 SPI Specifications (1/3)

Item	Description
Number of channels	3 channels
Transfer functions	• SPI serial communication (4-wire) and clock synchronous (3-wire) serial communication are possible by using the MOSI (Master Out Slave In), MISO (Master In Slave Out), SSL (Slave Select), and RSPCK (SPI Clock) signals. • Transmit-only operation is available • Receive-only operation is available • Serial communication is possible in master mode and slave mode • RSPCK polarity switching • RSPCK phase switching
Data format	• MSB first or LSB first selectable. • Transfer bit length selectable from 4 to 32 bits • 32 bit × 16 stage FIFO transmit and receive buffers • Up to four frames transferable in one round of transmission or reception (each frame consisting of up to 32 bits) • Byte swap operating function • Transmit/receive data inversion
Bit rate	• In master mode, the on-chip baud rate generator divides the RSPI_n_TCLK to generate RSPCK. A division ratio of 2 to 4096 is settable. • In slave mode, an external input clock is used as a serial clock. The maximum frequency = RSPI_n_TCLK/2 (High width: 1 RSPI_n_TCLK cycle, low width: 1 RSPI_n_TCLK cycle)
Buffer configuration	• Transmit buffer and receive buffer are configured independently.
Error detection	• Mode fault error detection • Underrun error detection • Overrun error detection • Parity error detection

Table 7.5-1 SPI Specifications (2/3)

Item	Description
SSL control function	**[Motorola SPI mode]** • Four SSL pins (SSLn0 to SSLn3) each channel • In single master mode, SSLn0 to SSLn3 pins are output. • In multi-master mode, SSLn0 pin for input, and SSLn1 to SSLn3 pins either for output or unused • In slave mode, SSLn0 pin for input, and SSLn1 to SSLn3 pins unused • Controllable delay from SSL output assertion to RSPCK operation (RSPCK delay) Range: 1 to 8 RSPCK cycles (set in RSPCK-cycle units) • Controllable delay from RSPCK stop to SSL output negation (SSL negation delay) Range: 1 to 8 RSPCK cycles (set in RSPCK-cycle units) • Controllable wait for next-access SSL output assertion (next-access delay) Range: 1 to 8 RSPCK cycles (set in RSPCK-cycle units) • Configurable delay between frames in burst transfer • Function for changing SSL polarity **[TI SSP mode]** • Four SSL pins (SSL0 to SSL3) each channel • In single master mode, SSLn0 to SSLn3 pins are output. • In multi-master mode, SSLn0 pin for input, and SSLn1 to SSLn3 pins either for output or unused • In slave mode, SSLn0 pin for input, and SSLn1 to SSLn3 pins unused • Controllable delay from SSL output assertion to SSL output negation (RSPCK delay) Range: 1 to 8 RSPCK cycles (set in RSPCK-cycle units) • Controllable delay from RSPCK stop to Output disable (SSL negation delay) Range: 1 to 8 RSPCK cycles (set in RSPCK-cycle units) • Controllable wait for next-access SSL output assertion (next-access delay) Range: 1 to 8 RSPCK cycles (set in RSPCK-cycle units) • Configurable delay between frames in burst transfer • Function for changing SSL polarity
Communication Protocol	• Motorola SPI • TI SSP (Synchronous Serial Protocol)
Synchronization bypass function	• Synchronization circuit can be bypassed using bus clock (RSCI_n_PCLK) as operation clock (RSCI_n_TCLK)
Control in master transfer	**[Motorola SPI mode]** • Transfers of up to eight commands each can be executed sequentially in looped execution • For each command, the following can be set: SSL signal value, bit rate, RSPCK polarity and phase, transfer data length, LSB first/MSB first, burst, RSPCK delay, SSL negation delay, next-access delay • Transfers can be initiated by writing to the transmit buffer • MOSI signal value specifiable in SSL negation • RSPCK auto-stop function **[TI SSP mode]** • Transfers of up to eight commands each can be executed sequentially in looped execution • For each command, the following can be set: SSL signal value, bit rate, RSPCK polarity and phase, transfer data length, LSB first/MSB first, burst, RSPCK delay, SSL negation delay (Output disable delay), next- access delay • Transfers can be initiated by writing to the transmit buffer • MOSI signal value specifiable in SSL negation • RSPCK auto-stop function
Interrupt sources	• Maskable Interrupt sources SPI receive buffer full / Receive data ready interrupt SPI transmit buffer empty interrupt SPI communication end interrupt SPI error interrupt (mode fault, overrun, parity error, receive data ready) SPI idle interrupt (SPI idle)

Table 7.5-1 SPI Specifications (3/3)

Item	Description
Others	• Switching between CMOS output and open-drain output • SPI disable (initialization) function • Loopback mode function
Module-stop function	Module-stop state can be set to reduce power consumption*[1]

Note 1. Module stop condition: It depends on the setting higher than this module. This module is not reset, and the clock stops on module stop condition. It contributes to low power consumption, because the clock stops.

[To]

Table 7.5-1 SPI Specifications (1/3)

Item	Description
Number of channels	3 channels
Transfer functions	• SPI serial communication (4-wire) and clock synchronous (3-wire) serial communication are possible by using the MOSI (Master Out Slave In), MISO (Master In Slave Out), SSL (Slave Select), and RSPCK (SPI Clock) signals. • Transmit-only operation is available • Receive-only operation is available • Serial communication is possible in master mode and slave mode • RSPCK polarity switching • RSPCK phase switching
Data format	• MSB first or LSB first selectable. • Transfer bit length selectable from 4 to 32 bits • 32 bit × 16 stage FIFO transmit and receive buffers • Up to four frames transferable in one round of transmission or reception (each frame consisting of up to 32 bits) • Byte swap operating function • Transmit/receive data inversion
Bit rate	• In master mode, the on-chip baud rate generator divides the RSPI_n_TCLK to generate RSPCK. A division ratio of 2 to 4096 is settable. • In slave mode, an external input clock is used as a serial clock. The maximum frequency = RSPI_n_TCLK/2 (High width: 1 RSPI_n_TCLK cycle, low width: 1 RSPI_n_TCLK cycle)
Buffer configuration	• Transmit buffer and receive buffer are configured independently.
Error detection	• Mode fault error detection • Underrun error detection • Overrun error detection • Parity error detection

Table 7.5-1 SPI Specifications (2/3)

Item	Description
SSL control function	**[Motorola SPI mode]** • Four SSL pins (SSLn0 to SSLn3) each channel • In master mode, SSLn0 to SSLn3 pins are output. • In slave mode, SSLn0 pin for input, and SSLn1 to SSLn3 pins unused • Controllable delay from SSL output assertion to RSPCK operation (RSPCK delay) Range: 1 to 8 RSPCK cycles (set in RSPCK-cycle units) • Controllable delay from RSPCK stop to SSL output negation (SSL negation delay) Range: 1 to 8 RSPCK cycles (set in RSPCK-cycle units) • Controllable wait for next-access SSL output assertion (next-access delay) Range: 1 to 8 RSPCK cycles (set in RSPCK-cycle units) • Configurable delay between frames in burst transfer • Function for changing SSL polarity **[TI SSP mode]** • Four SSL pins (SSL0 to SSL3) each channel • In master mode, SSLn0 to SSLn3 pins are output. • In slave mode, SSLn0 pin for input, and SSLn1 to SSLn3 pins unused • Controllable delay from SSL output assertion to SSL output negation (RSPCK delay) Range: 1 to 8 RSPCK cycles (set in RSPCK-cycle units) • Controllable delay from RSPCK stop to Output disable (SSL negation delay) Range: 1 to 8 RSPCK cycles (set in RSPCK-cycle units) • Controllable wait for next-access SSL output assertion (next-access delay) Range: 1 to 8 RSPCK cycles (set in RSPCK-cycle units) • Configurable delay between frames in burst transfer • Function for changing SSL polarity
Communication Protocol	• Motorola SPI • TI SSP (Synchronous Serial Protocol)
Synchronization bypass function	• Synchronization circuit can be bypassed using bus clock (RSCI_n_PCLK) as operation clock (RSCI_n_TCLK)
Control in master transfer	**[Motorola SPI mode]** • Transfers of up to eight commands each can be executed sequentially in looped execution • For each command, the following can be set: SSL signal value, bit rate, RSPCK polarity and phase, transfer data length, LSB first/MSB first, burst, RSPCK delay, SSL negation delay, next-access delay • Transfers can be initiated by writing to the transmit buffer • MOSI signal value specifiable in SSL negation • RSPCK auto-stop function **[TI SSP mode]** • Transfers of up to eight commands each can be executed sequentially in looped execution • For each command, the following can be set: SSL signal value, bit rate, RSPCK polarity and phase, transfer data length, LSB first/MSB first, burst, RSPCK delay, SSL negation delay (Output disable delay), next- access delay • Transfers can be initiated by writing to the transmit buffer • MOSI signal value specifiable in SSL negation • RSPCK auto-stop function
Interrupt sources	• Maskable Interrupt sources SPI receive buffer full / Receive data ready interrupt SPI transmit buffer empty interrupt SPI communication end interrupt SPI error interrupt (mode fault, overrun, parity error, receive data ready) SPI idle interrupt (SPI idle)

Table 7.5-1 SPI Specifications (3/3)

Item	Description
Others	• SPI disable (initialization) function • Loopback mode function
Module-stop function	Module-stop state can be set to reduce power consumption*[1]

Note 1. Module stop condition: It depends on the setting higher than this module. This module is not reset, and the clock stops on module stop condition. It contributes to low power consumption, because the clock stops.

User's Manual
7.5 Serial Peripheral Interface (RSPI)
7.5.1.1 Features

[From]

The SPI automatically switches the SSL0 pin input and output directions. The SSL0 pin is switched to output when single master is set, and it is switched to input when multi-master or slave is set. Furthermore, the SPI automatically switches the input and output directions of pins RSPCK, MOSI, MISO, SSL1 to SSL3 according to the master/slave setting and SPI operation (4-wire)/clock synchronous operation (3-wire) SSL0 input level. (See **7.5.3.2 SPI Pin Control**.)

[To]

The SPI automatically switches the SSL0 pin input and output directions. The SSL0 pin is switched to output when master is set, and it is switched to input when slave is set. Furthermore, the SPI automatically switches the input and output directions of pins RSPCK, MOSI, MISO, SSL1 to SSL3 according to the master/slave setting and SPI operation (4-wire)/clock synchronous operation (3-wire) SSL0 input level. (See **7.5.3.2 SPI Pin Control**.)

User's Manual

7.5 Serial Peripheral Interface (RSPI)

7.5.2.2 Register Description

7.5.2.2.9 SPI Pin Control Register (RSPIm_SPPCR)

[From]

7.5.2.2.9 SPI Pin Control Register (RSPIm_SPPCR)

The SPPCR register is used to set pin mode of the SPI. If SPPCR is modified with the SPCR.SPE bit = 1b, subsequent operation is not guaranteed.

Access Size : 8 bits
Address : <RSPIm_base> + 000Eh
Initial Value : 00h

Bit	7	6	5	4	3	2	1	0
	-	-	MOIFE	MOIFV	-	SPOM	SPLP2	SPLP
Initial Value	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
7, 6	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
5	MOIFE	0h	RW	MOSI Idle Value Fixing Enable 0b: The MOSI output value is the last data of previous transfer. 1b: The MOSI output value is the set MOIFV bit value.
4	MOIFV	0h	RW	MOSI Idle Fixed Value 0b: The fixed value of MOSI idle = 0. 1b: The fixed value of MOSI idle = 1.
3	-	0h	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
2	SPOM	0h	RW	SPI Output Pin Mode 0b: CMOS output 1b: Open-drain output
1	SPLP2	0h	RW	SPI Loopback 2 0b: Normal mode 1b: Loopback mode (data is not inverted for transmission)
0	SPLP	0h	RW	SPI Loopback 0b: Normal mode 1b: Loopback mode (data is inverted for transmission)

SPLP bit (SPI Loopback)

When the SPLP bit is set to 1b, the SPI shuts down the route between the MISO pin and the shift register (when SPCR.MSTR = 1b) or shuts down the route between the MOSI pin and the shift register, inverts the input route value in the shift register, and then connects the route to the output route (when SPCR.MSTR = 0b) (loopback mode).

SPLP2 bit (SPI Loopback 2)

When the SPLP2 bit is set to 1b, the SPI shuts down the route between the MISO pin and the shift register (when SPCR.MSTR = 1b) or shuts down the route between MOSI pin and the shift register and then connects the route to the output route without inverting the input route value in the shift register (when SPCR.MSTR = 0b) (loopback mode). If this bit is set to 1b together with the SPLP bit, setting this bit takes precedence.

SPOM bit (SPI Output Pin Mode)

This bit is used to select CMOS output pin or open drain output pin as SPI’s output pins.

MOIFV bit (MOSI Idle Fixed Value)

This bit is used to select the MOSI pin output value during the SSL negation period (including SSL retention period in burst transfer) when SPCR.MOIFE = 1b in master mode.

[To]

7.5.2.2.9 SPI Pin Control Register (RSPIm_SPPCR)

The SPPCR register is used to set pin mode of the SPI. If SPPCR is modified with the SPCR.SPE bit = 1b, subsequent operation is not guaranteed.

Access Size : 8 bits
Address : <RSPIm_base> + 000Eh
Initial Value : 00h

Bit	7	6	5	4	3	2	1	0
	-	-	MOIFE	MOIFV	-	-	SPLP2	SPLP
Initial Value	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
7, 6	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
5	MOIFE	0h	RW	MOSI Idle Value Fixing Enable 0b: The MOSI output value is the last data of previous transfer. 1b: The MOSI output value is the set MOIFV bit value.
4	MOIFV	0h	RW	MOSI Idle Fixed Value 0b: The fixed value of MOSI idle = 0. 1b: The fixed value of MOSI idle = 1.
3, 2	-	0h	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
1	SPLP2	0h	RW	SPI Loopback 2 0b: Normal mode 1b: Loopback mode (data is not inverted for transmission)
0	SPLP	0h	RW	SPI Loopback 0b: Normal mode 1b: Loopback mode (data is inverted for transmission)

SPLP bit (SPI Loopback)

When the SPLP bit is set to 1b, the SPI shuts down the route between the MISO pin and the shift register (when SPCR.MSTR = 1b) or shuts down the route between the MOSI pin and the shift register, inverts the input route value in the shift register, and then connects the route to the output route (when SPCR.MSTR = 0b) (loopback mode).

SPLP2 bit (SPI Loopback 2)

When the SPLP2 bit is set to 1b, the SPI shuts down the route between the MISO pin and the shift register (when SPCR.MSTR = 1b) or shuts down the route between MOSI pin and the shift register and then connects the route to the output route without inverting the input route value in the shift register (when SPCR.MSTR = 0b) (loopback mode). If this bit is set to 1b together with the SPLP bit, setting this bit takes precedence.

MOIFV bit (MOSI Idle Fixed Value)

This bit is used to select the MOSI pin output value during the SSL negation period (including SSL retention period in burst transfer) when SPCR.MOIFE = 1b in master mode.

User's Manual
7.5 Serial Peripheral Interface (RSPI)
7.5.2.2 Register Description
7.5.2.2.11 SPI Slave Select Polarity Register (RSPIm_SSLP)

[From]

7.5.2.2.11 SPI Slave Select Polarity Register (RSPIm_SSLP)

The SSLP register is used to set the polarity of SSL0 to SSL3 signals of the SPI. If any of these SSLP bits is modified with the SPCR.SPE bit = 1b, subsequent operation is not guaranteed.

Access Size : 8 bits
Address : <RSPIm_base> + 0010h
Initial Value : 00h

Bit	7	6	5	4	3	2	1	0
	-	-	-	-	SSL3P	SSL2P	SSL1P	SSL0P
Initial Value	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
3	SSL3P	0h	RW	SSL3 Signal Polarity Setting 0b: In the Motorola-SPI case, the SSL3 signal is active low (0). In the TI-SSP case, the SSL3 signal is active high (1). 1b: In the Motorola-SPI case, the SSL3 signal is active high (1). In the TI-SSP case, the SSL3 signal is active low (0).
2	SSL2P	0h	RW	SSL2 Signal Polarity Setting 0b: In the Motorola-SPI case, the SSL2 signal is active low (0). In the TI-SSP case, the SSL2 signal is active high (1). 1b: In the Motorola-SPI case, the SSL2 signal is active high (1). In the TI-SSP case, the SSL2 signal is active low (0).
1	SSL1P	0h	RW	SSL1 Signal Polarity Setting 0b: In the Motorola-SPI case, the SSL1 signal is active low (0). In the TI-SSP case, the SSL1 signal is active high (1). 1b: In the Motorola-SPI case, the SSL1 signal is active high (1). In the TI-SSP case, the SSL1 signal is active low (0).
0	SSL0P	0h	RW	SSL0 Signal Polarity Setting 0b: In the Motorola-SPI case, the SSL0 signal is active low (0). In the TI-SSP case, the SSL0 signal is active high (1). 1b: In the Motorola-SPI case, the SSL0 signal is active high (1). In the TI-SSP case, the SSL0 signal is active low (0).

SSLnP bits (SSLn Signal Polarity Setting)

These bits are used to specify the polarity of SSL signals. The set SSLnP bit (n = 3 to 0) values indicate the active polarity of SSLn signals.

SSL0 is different from SSL1, SSL2, and SSL3. In slave or multi-master mode, it functions as an input.

For details, see **7.5.3.3.2 Single master/single slave (this LSI = slave)** and **7.5.3.3.5 Multi-master/multi-slave (with this LSI acting as master)**.

[To]

7.5.2.2.11 SPI Slave Select Polarity Register (RSPIm_SSLP)

The SSLP register is used to set the polarity of SSL0 to SSL3 signals of the SPI. If any of these SSLP bits is modified with the SPCR.SPE bit = 1b, subsequent operation is not guaranteed.

Access Size : 8 bits
Address : <RSPIm_base> + 0010h
Initial Value : 00h

Bit	7	6	5	4	3	2	1	0
	-	-	-	-	SSL3P	SSL2P	SSL1P	SSL0P
Initial Value	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
3	SSL3P	0h	RW	SSL3 Signal Polarity Setting 0b: In the Motorola-SPI case, the SSL3 signal is active low (0). In the TI-SSP case, the SSL3 signal is active high (1). 1b: In the Motorola-SPI case, the SSL3 signal is active high (1). In the TI-SSP case, the SSL3 signal is active low (0).
2	SSL2P	0h	RW	SSL2 Signal Polarity Setting 0b: In the Motorola-SPI case, the SSL2 signal is active low (0). In the TI-SSP case, the SSL2 signal is active high (1). 1b: In the Motorola-SPI case, the SSL2 signal is active high (1). In the TI-SSP case, the SSL2 signal is active low (0).
1	SSL1P	0h	RW	SSL1 Signal Polarity Setting 0b: In the Motorola-SPI case, the SSL1 signal is active low (0). In the TI-SSP case, the SSL1 signal is active high (1). 1b: In the Motorola-SPI case, the SSL1 signal is active high (1). In the TI-SSP case, the SSL1 signal is active low (0).
0	SSL0P	0h	RW	SSL0 Signal Polarity Setting 0b: In the Motorola-SPI case, the SSL0 signal is active low (0). In the TI-SSP case, the SSL0 signal is active high (1). 1b: In the Motorola-SPI case, the SSL0 signal is active high (1). In the TI-SSP case, the SSL0 signal is active low (0).

SSLnP bits (SSLn Signal Polarity Setting)

These bits are used to specify the polarity of SSL signals. The set SSLnP bit (n = 3 to 0) values indicate the active polarity of SSLn signals.

SSL0 is different from SSL1, SSL2, and SSL3. In slave mode, it functions as an input.

For details, see **7.5.3.3.2 Master/single slave (this LSI = slave)**.

User's Manual
7.5 Serial Peripheral Interface (RSPI)
7.5.2.2 Register Description
7.5.2.2.14 SPI Command Register n (RSPIm_SPCMDn) (n = 0 to 7)
SSLA[1:0] bits

[From]

SSLA[1:0] bits (SSL Signal Assertion)

These bits are used to control SSL signal assertion for the SPI in master mode to perform serial transfer. The set SSLA[1:0] bits value controls assertion of the SSL3 to SSL0 signals. The signal polarity when the SSL signal is asserted depends on the set value of the SSLP register. When SSLA[1:0] bits are set to 0b in multi-master mode, serial transfer is performed with all SSL signals negated (because SSL0 is input).

To use the SPI in slave mode, set SSLA[1:0] bits to 0b.

[To]

SSLA[1:0] bits (SSL Signal Assertion)

These bits are used to control SSL signal assertion for the SPI in master mode to perform serial transfer. The set SSLA[1:0] bits value controls assertion of the SSL3 to SSL0 signals. The signal polarity when the SSL signal is asserted depends on the set value of the SSLP register.

To use the SPI in slave mode, set SSLA[1:0] bits to 0b.

User’s Manual
7.5 Serial Peripheral Interface (RSPI)
7.5.2.2 Register Description
7.5.2.2.15 SPI Data Control Register (RSPIm_SPDCR)

[From]

7.5.2.2.15 SPI Data Control Register (RSPIm_SPDCR)

The SPDCR register controls the data format.

If the value set in this register is changed while SPCR.SPE = 1b, subsequent operations are not guaranteed.

Access Size : 8, 16 bits
Address : <RSPIm_base> + 0040h
Initial Value : 0000h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	SPFC[3:0]				-	-	-	SINV	SPRDTD	SLSEL[1:0]		BYSW
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
11 to 8	SPFC[3:0]	0h	RW	Frame Count Number of Frames Specification 0000b: 1 frame 0001b: 2 frames 0010b: 3 frames 0011b: 4 frames : 1110b: 15 frames 1111b: 16 frames
7 to 5	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
4	SINV	0h	RW	Serial data invert 0b: Not invert serial data 1b: Invert serial data
3	SPRDTD	0h	RW	SPI Receive Data or Transmit Data Selection 0b: The SPDR reads the receive buffer 1b: The SPDR reads the transmit buffer
2, 1	SLSEL[1:0]	0h	RW	SSL Pin Output Select See **Table 7.5-7**.
0	BYSW	0h	RW	Byte Swap Operating Mode Select 0b: Byte swap is disabled 1b: Byte swap is enabled

BYSW bit (Byte Swap Operating Mode Select)

It is a setting bit, that is to swap a transmit/receive data in byte units. A data after byte swap is different by a data length (setting of SPCMD.SPB[4:0]).

When byte swap, A data length (setting of SPB[4:0]) must be set to 32 bits or 16 bits. Other case of data length (for example, 4 to 15, 17 to 31 bit length), byte swap is not guaranteed. For the arrangement of data before and after swapping data lengths of 32 bits and 16 bits, see **7.5.3.4 Data Format** and **7.5.3.4.5 Byte swap reception**.

When the parity function set to valid, the behavior is not guaranteed.

SLSEL[1:0] bits (SSL Pin Output Select)

The SLSEL[1:0] bits are used to select SSL output or I/O as an SSL pin function in master mode.

Table 7.5-7 SSL Pin Output Selection

SSLn Pin	SLSEL[1:0] = 0b	SLSEL[1:0] = 01b	SLSEL[1:0] = 10b	SLSEL[1:0] = 11b
SSL0	Output or Input	Output or Input	Output or Input	Setting prohibited
SSL1	Output	I/O	Output	
SSL2, SSL3	Output	I/O	I/O	

Note: Input or output of SSL0 is determined by the SPCR.MODFEN bit. For details, see **Table 7.5-8**.

SPRDTD bit (SPI Receive Data or Transmit Data Selection)

The SPRDTD bit is used to select receive buffer or transmit buffer from which the SPI data register (SPDR) value is read.

When the transmit buffer is read, the value written to SPDR the last time is read.

SINV bit (Serial data invert)

The SINV bit is used to invert transmit data and receive data.

When the SINV bit is set to 1b, transmit buffer (SPTX) data is inverted to invert transmit data and receive data, and then the inverted data is stored in the receive buffer (SPRX). The parity bit is the value corresponding to the inverted transmission/ reception data.

SPFC[1:0] bits (Frame Count)

The SPFC[1:0] bits are used for the condition to set the CENDF flag in slave receive only mode. For details on the CENDF flag setting conditions, see **7.5.2.2.18 SPI Status Register (SPSR)**.

The SPFC[1:0] bits are invalid except in the slave receive only mode.

[To]

7.5.2.2.15 SPI Data Control Register (RSPIm_SPDCR)

The SPDCR register controls the data format.

If the value set in this register is changed while SPCR.SPE = 1b, subsequent operations are not guaranteed.

Access Size : 8, 16 bits

Address : <RSPIm_base> + 0040h

Initial Value : 0000h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	SPFC[3:0]				-	-	-	SINV	SPRDTD	-	-	BYSW
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
11 to 8	SPFC[3:0]	0h	RW	Frame Count Number of Frames Specification 0000b: 1 frame 0001b: 2 frames 0010b: 3 frames 0011b: 4 frames : 1110b: 15 frames 1111b: 16 frames
7 to 5	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
4	SINV	0h	RW	Serial data invert 0b: Not invert serial data 1b: Invert serial data
3	SPRDTD	0h	RW	SPI Receive Data or Transmit Data Selection 0b: The SPDR reads the receive buffer 1b: The SPDR reads the transmit buffer
2, 1	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
0	BYSW	0h	RW	Byte Swap Operating Mode Select 0b: Byte swap is disabled 1b: Byte swap is enabled

BYSW bit (Byte Swap Operating Mode Select)

It is a setting bit, that is to swap a transmit/receive data in byte units. A data after byte swap is different by a data length (setting of SPCMD.SPB[4:0]).

When byte swap, A data length (setting of SPB[4:0]) must be set to 32 bits or 16 bits. Other case of data length (for example, 4 to 15, 17 to 31 bit length), byte swap is not guaranteed. For the arrangement of data before and after swapping data lengths of 32 bits and 16 bits, see **7.5.3.4 Data Format** and **7.5.3.4.5 Byte swap reception**.

When the parity function set to valid, the behavior is not guaranteed.

SPRDTD bit (SPI Receive Data or Transmit Data Selection)

The SPRDTD bit is used to select receive buffer or transmit buffer from which the SPI data register (SPDR) value is read.

When the transmit buffer is read, the value written to SPDR the last time is read.

SINV bit (Serial data invert)

The SINV bit is used to invert transmit data and receive data.

When the SINV bit is set to 1b, transmit buffer (SPTX) data is inverted to invert transmit data and receive data, and then the inverted data is stored in the receive buffer (SPRX). The parity bit is the value corresponding to the inverted transmission/ reception data.

SPFC[1:0] bits (Frame Count)

The SPFC[1:0] bits are used for the condition to set the CENDF flag in slave receive only mode. For details on the CENDF flag setting conditions, see **7.5.2.2.18 SPI Status Register (RSPIm_**SPSR).

The SPFC[1:0] bits are invalid except in the slave receive only mode.

User's Manual
7.5 Serial Peripheral Interface (RSPI)
7.5.2.2 Register Description
7.5.2.2.18 SPI Status Register (RSPIm_SPSR)
MODF bit

[From]

MODF bit (Mode Fault Error Flag)

This flag indicates whether a mode fault error or an underrun error is present. The UDRF flag allows you to see which error (mode fault error or underrun error) has occurred.

[Setting (to 1b) condition]

[Multi-master mode]

- The SSL0 pin input level becomes active level while the SPCR.MSTR bit = 1b (master mode) and the SPCR.MODFEN bit = 1b (mode fault error detection enabled), and then the SPI has detected a mode fault error.

[Slave, Motorola-SPI mode]

Any of the following two conditions is met:

- The SSL0 pin is negated before the RSPCK cycles necessary for data transfer end while the SPCR.MSTR bit = 0b (slave mode), SPCR.SPFRF bit = 0b (Motorola-SPI) and the SPCR.MODFEN bit = 1b (mode fault error detection enabled), and then the SPI has detected a mode fault error.
- Serial transfer is started before transmit data output becomes ready while the SPCR.SPE bit = 1b (SPI function enabled), and then the SPI has detected an underrun error.

[Slave, TI-SSP mode]

Any of the following two conditions is met:

- The SSL0 pin is asserted before the RSPCK cycles necessary for data transfer end while the SPCR.MSTR bit = 0b (slave mode), SPCR.SPFRF bit = 1b (TI-SSP) and the SPCR.MODFEN bit = 1b (mode fault error detection enabled), and then the SPI has detected a mode fault error.
- Serial transfer is started before transmit data output becomes ready while the SPCR.SPE bit = 1b (SPI function enabled), and then the SPI has detected an underrun error.

The SSL signal active level depends on the SSLP.SSLiP bits (SSL signal polarity bits).

[Clearing (to 0b) condition]

- When 1b is written to the SPSRC.MODFC bit

[To]

MODF bit (Mode Fault Error Flag)

This flag indicates whether a mode fault error or an underrun error is present. The UDRF flag allows you to see which error (mode fault error or underrun error) has occurred.

[Setting (to 1b) condition]

[Slave, Motorola-SPI mode]

Any of the following two conditions is met:

- The SSL0 pin is negated before the RSPCK cycles necessary for data transfer end while the SPCR.MSTR bit = 0b (slave mode), SPCR.SPFRF bit = 0b (Motorola-SPI) and the SPCR.MODFEN bit = 1b (mode fault error detection enabled), and then the SPI has detected a mode fault error.
- Serial transfer is started before transmit data output becomes ready while the SPCR.SPE bit = 1b (SPI function enabled), and then the SPI has detected an underrun error.

[Slave, TI-SSP mode]

Any of the following two conditions is met:

- The SSL0 pin is asserted before the RSPCK cycles necessary for data transfer end while the SPCR.MSTR bit = 0b (slave mode), SPCR.SPFRF bit = 1b (TI-SSP) and the SPCR.MODFEN bit = 1b (mode fault error detection enabled), and then the SPI has detected a mode fault error.
- Serial transfer is started before transmit data output becomes ready while the SPCR.SPE bit = 1b (SPI function enabled), and then the SPI has detected an underrun error.

The SSL signal active level depends on the SSLP.SSLiP bits (SSL signal polarity bits).

[Clearing (to 0b) condition]

- When 1b is written to the SPSRC.MODFC bit

User's Manual
7.5 Serial Peripheral Interface (RSPI)
7.5.3.1 Overview of SPI Operations

[From]

7.5.3.1 Overview of SPI Operations

The SPI can transfer data in the following five modes.

- Slave mode (SPI operation)
- Single master mode (SPI operation)
- Multi-master mode (SPI operation)
- Slave mode (clock synchronous operation)
- Master mode (clock synchronous operation)

[To]

7.5.3.1 Overview of SPI Operations

The SPI can transfer data in the following five modes.

- Slave mode (SPI operation)
- Master mode (SPI operation)
- Slave mode (clock synchronous operation)
- Master mode (clock synchronous operation)

User's Manual
7.5 Serial Peripheral Interface (RSPI)
7.5.3.1 Overview of SPI Operations
Table 7.5-8 Relationship between SPI Modes and SPCR Settings and Description of Each Mode

[From]

Table 7.5-8 Relationship between SPI Modes and SPCR Settings and Description of Each Mode (1/2)

Mode	Slave (SPI Operation)	Single-Master (SPI Operation)	Multi-Master (SPI Operation)	Slave (Clock Synchronous Operation)	Master (Clock Synchronous Operation)
MSTR bit setting	0	1	1	0	1
MODFEN bit setting	0 or 1	0	1	0	0
SPMS bit setting	0	0	0	1	1
SPFRF bit setting	valid	valid	valid	In-valid	In-valid
RSPCK signal	Input	Output	Output/Hi-Z	Input	Output
MOSI signal	Input	Output	Output/Hi-Z	Input	Output
MISO signal	Output/Hi-Z	Input	Input	Output	Input
SSL0 signal	Input	Output	Input	Hi-Z (not used)	Hi-Z (not used)
SSL1 to SSL3 signals	Hi-Z (not used)	Output	Output/Hi-Z	Hi-Z (not used)	Hi-Z (not used)
Output pin mode	CMOS/open drain	CMOS/open drain	CMOS/open drain	CMOS/open drain	CMOS/open drain
SSL polarity modification function	Supported	Supported	Supported	—	—
Transfer rate	Up to RSPI_n_TCLK/2	Up to RSPI_n_TCLK/2	Up to RSPI_n_TCLK/2	Up to RSPI_n_TCLK/2	Up to RSPI_n_TCLK/2
Clock source	RSPCK input	On-chip baud rate generator	On-chip baud rate generator	RSPCK input	On-chip baud rate generator
Clock polarity	2 types				
Clock phase	2 types*[1]	2 types*[5]	2 types*[5]	One (CPHA = 1b)	2 types
First transfer bit	MSB/LSB				
Transfer data length	4 to 32 bits				
Burst transfer	Enabled (CPHA = 1b)	Enabled (CPHA = 0b, 1b)	Enabled (CPHA = 0b, 1b)	—	—
RSPCK delay control	Not supported	Supported	Supported	Not supported	Supported
SSL negation delay control	Not supported*[6]	Supported	Supported	Not supported	Supported
Next-access delay control	Not supported	Supported	Supported	Not supported	Supported

Table 7.5-8 Relationship between SPI Modes and SPCR Settings and Description of Each Mode (2/2)

Mode	Slave (SPI Operation)	Single-Master (SPI Operation)	Multi-Master (SPI Operation)	Slave (Clock Synchronous Operation)	Master (Clock Synchronous Operation)
Transfer activation method	SSL input active or RSPCK oscillation	Transmit buffer empty interrupt request or transmit buffer write when SPTEF = 1b	Transmit buffer empty interrupt request or transmit buffer write when SPTEF = 1b	RSPCK oscillation	Transmit buffer empty interrupt request or transmit buffer write when SPTEF = 1b
Sequence control	Not supported	Supported	Supported	Not supported	Supported
Transmission buffer empty detection	Supported*4	Supported	Supported	Supported*4	Supported
Reception buffer full detection	Supported*1				
Overrun error detection	Supported*1	Supported*1,*3	Supported*1,*3	Supported*1	Supported*1 *3
Parity error detection	Supported*1,*2				
Mode fault error detection	Supported (MODFEN = 1b)	Not supported	Supported	Not supported	Not supported
Underrun error detection	Supported*4	Not supported	Not supported	Supported*4	Not supported

Note 1. When SPI is transmit-master mode or transmit-slave mode (see **Table 7.5-5**), none of receive buffer full error, overrun error, and parity error is detected.

Note 2. When the SPCR.SPPE bit is 0b, parity error is not detected.

Note 3. When the SPCR.SCKASE bit is 1b, overrun error is not detected.

Note 4. When SPI is receive only slave mode (see **Table 7.5-5**), none of transmit buffer empty and underrun error is detected.

Note 5. CPHA = 0b is invalid in TI SSP mode. (Even if it is set, the operation is the same as when CPHA = 1b.)

Note 6. Available only in TI SSP mode.

[To]

Table 7.5-7 Relationship between SPI Modes and SPCR Settings and Description of Each Mode (1/2)

Mode	Slave (SPI Operation)	Master (SPI Operation)	Slave (Clock Synchronous Operation)	Master (Clock Synchronous Operation)
MSTR bit setting	0	1	0	1
MODFEN bit setting	0 or 1	0	0	0
SPMS bit setting	0	0	1	1
SPFRF bit setting	valid	valid	In-valid	In-valid
RSPCK signal	Input	Output	Input	Output
MOSI signal	Input	Output	Input	Output
MISO signal	Output/Hi-Z	Input	Output	Input
SSL0 signal	Input	Output	Hi-Z (not used)	Hi-Z (not used)
SSL1 to SSL3 signals	Hi-Z (not used)	Output	Hi-Z (not used)	Hi-Z (not used)
SSL polarity modification function	Supported	Supported	—	—
Transfer rate	Up to RSPI_n_TCLK/2	Up to RSPI_n_TCLK/2	Up to RSPI_n_TCLK/2	Up to RSPI_n_TCLK/2
Clock source	RSPCK input	On-chip baud rate generator	RSPCK input	On-chip baud rate generator
Clock polarity	2 types			
Clock phase	2 types*1	2 types*5	One (CPHA = 1b)	2 types
First transfer bit	MSB/LSB			
Transfer data length	4 to 32 bits			
Burst transfer	Enabled (CPHA = 1b)	Enabled (CPHA = 0b, 1b)	—	—
RSPCK delay control	Not supported	Supported	Not supported	Supported
SSL negation delay control	Not supported*6	Supported	Not supported	Supported
Next-access delay control	Not supported	Supported	Not supported	Supported

Table 7.5-7 Relationship between SPI Modes and SPCR Settings and Description of Each Mode (2/2)

Mode	Slave (SPI Operation)	Master (SPI Operation)	Slave (Clock Synchronous Operation)	Master (Clock Synchronous Operation)
Transfer activation method	SSL input active or RSPCK oscillation	Transmit buffer empty interrupt request or transmit buffer write when SPTEF = 1b	RSPCK oscillation	Transmit buffer empty interrupt request or transmit buffer write when SPTEF = 1b
Sequence control	Not supported	Supported	Not supported	Supported
Transmission buffer empty detection	Supported*4	Supported	Supported*4	Supported
Reception buffer full detection	Supported*1			
Overrun error detection	Supported*1	Supported*1,*3	Supported*1	Supported*1 *3
Parity error detection	Supported*1,*2			
Mode fault error detection	Supported (MODFEN = 1b)	Not supported	Not supported	Not supported
Underrun error detection	Supported*4	Not supported	Supported*4	Not supported

Note 1. When SPI is transmit-master mode or transmit-slave mode (see **Table 7.5-5**), none of receive buffer full error, overrun error, and parity error is detected.

Note 2. When the SPCR.SPPE bit is 0b, parity error is not detected.

Note 3. When the SPCR.SCKASE bit is 1b, overrun error is not detected.

Note 4. When SPI is receive only slave mode (see **Table 7.5-5**), none of transmit buffer empty and underrun error is detected.

Note 5. CPHA = 0b is invalid in TI SSP mode. (Even if it is set, the operation is the same as when CPHA = 1b.)

Note 6. Available only in TI SSP mode.

User's Manual
7.5 Serial Peripheral Interface (RSPI)
7.5.3.2 SPI Pin Control

[From]

7.5.3.2 SPI Pin Control

The SPI automatically switches pin directions and output modes according to the settings of the SPCR.MSTR, SPCR.MODFEN and SPCR.SPMS bits and the SPPCR.SPOM bit.

When the SPPCR.SPOM bit is set to 0b, each output becomes CMOS output, when the SPOM bit is set to 1b, each output becomes open drain output.

Table 7.5-9 shows the relationship between pin state and set values of each bit.

For details of OE function, see **7.5.3.5 Transfer Format (Frame Format)**.

[To]

7.5.3.2 SPI Pin Control

The SPI automatically switches pin directions and output modes according to the settings of the SPCR.MSTR, SPCR.MODFEN and SPCR.SPMS bits.

Table 7.5-8 shows the relationship between pin state and set values of each bit.

For details of OE function, see **7.5.3.5 Transfer Format (Frame Format)**.

User's Manual
7.5 Serial Peripheral Interface (RSPI)
7.5.3.2 SPI Pin Control
Table 7.5-9 Relationship between Pin States and Set Control Bit Values

[From]

Table 7.5-9 Relationship between Pin States and Set Control Bit Values

Mode	Pin	Pin State*1	
		SPOM = 0	SPOM = 1
Single-master mode (SPI operation) (MSTR = 1b, MODFEN = 0b, SPMS = 0b)	RSPCK	CMOS output	Open-drain output
	SSL0 to SSL3	CMOS output	Open-drain output
	MOSI	CMOS output	Open-drain output
	MISO	Input	Input
Multi-master mode (SPI operation) (MSTR = 1b, MODFEN = 1b, SPMS = 0b)	RSPCK*2	CMOS output/Hi-Z	Open-drain output/Hi-Z
	SSL0	Input	Input
	SSL1 to SSL3*2	CMOS output/Hi-Z	Open-drain output/Hi-Z
	MOSI*2	CMOS output/Hi-Z	Open-drain output/Hi-Z
	MISO	Input	Input
Slave mode (SPI operation) (MSTR = 0b, SPMS = 0b)	RSPCK	Input	Input
	SSL0	Input	Input
	SSL1 to SSL3*4	Hi-Z (not used)	Hi-Z (not used)
	MOSI	Input	Input
	MISO*3	CMOS output/Hi-Z	Open-drain output/Hi-Z
Master mode (Clock synchronous operation) (MSTR = 1b, MODFEN = 0b, SPMS = 1b)	RSPCK	CMOS output	Open-drain output
	SSL0 to SSL3*4	Hi-Z (not used)	Hi-Z (not used)
	MOSI	CMOS output	Open-drain output
	MISO	Input	Input
Slave mode (Clock synchronous operation) (MSTR = 0b, SPMS = 1b)	RSPCK	Input	Input
	SSL0 to SSL3*4	Hi-Z (not used)	Hi-Z (not used)
	MOSI	Input	Input
	MISO	CMOS output	Open-drain output

Note 1. The set SPI value is not applied to multi-function pins for which the SPI function is not selected.

Note 2. When SSL0 is at the active level, the pin state is Hi-Z. Motorola-SPI: When SSL0 is active level, the pin state becomes Hi-Z.
TI-SSP: The terminal status becomes Hi-Z until SSL0 is asserted after SPCR.SPE = 1b and communication completed.

Note 3. Motorola-SPI: When SSL0 is inactive level or when SPCR.SPE = 0b, the pin state becomes Hi-Z.
TI-SSP: When SSL0 is except the communication period or when SPCR.SPE = 0b (assertion after SPE = 1b and communication completed), the pin status changes to Hi-Z.

Note 4. These pins are available for use as I/O port pins.

The SPI in single master mode (SPI operation) or multi-master mode (SPI operation) determines the MOSI signal value in the SSL negation period (including the SSL hold period in burst transfer) according to the settings of the SPPCR.MOIFE and SPPCR.MOIFV bits, as listed in **Table 7.5-10**.

[To]

Table 7.5-8 Relationship between Pin States and Set Control Bit Values

Mode	Pin	Pin State*1
Master mode (SPI operation) (MSTR = 1b, MODFEN = 0b, SPMS = 0b)	RSPCK	Output
	SSL0 to SSL3	Output
	MOSI	Output
	MISO	Input
Slave mode (SPI operation) (MSTR = 0b, SPMS = 0b)	RSPCK	Input
	SSL0	Input
	SSL1 to SSL3*3	Not used
	MOSI	Input
	MISO*2	Output/Hi-Z
Master mode (Clock synchronous operation) (MSTR = 1b, MODFEN = 0b, SPMS = 1b)	RSPCK	Output
	SSL0 to SSL3*3	Not used
	MOSI	Output
	MISO	Input
Slave mode (Clock synchronous operation) (MSTR = 0b, SPMS = 1b)	RSPCK	Input
	SSL0 to SSL3*3	Not used
	MOSI	Input
	MISO	Output

Note 1. The set SPI value is not applied to multi-function pins for which the SPI function is not selected.

Note 2. Motorola-SPI: When SSL0 is inactive level or when SPCR.SPE = 0b, the pin state becomes Hi-Z.
TI-SSP: When SSL0 is except the communication period or when SPCR.SPE = 0b (assertion after SPE = 1b and communication completed), the pin status changes to Hi-Z.

Note 3. These pins are available for use as I/O port pins.

The SPI in master mode (SPI operation) determines the MOSI signal value in the SSL negation period (including the SSL hold period in burst transfer) according to the settings of the SPPCR.MOIFE and SPPCR.MOIFV bits, as listed in **Table 7.5-9**.

User's Manual
7.5 Serial Peripheral Interface (RSPI)
7.5.3.3 SPI System Configuration Examples

[From]

7.5.3.3 SPI System Configuration Examples

This configuration example describes that 0 level of SSL signals is active level. When connecting and using in a multi-slave or multi-master mode, the transfer format of the connected device should be unified to either Motorola-SPI or TISSP.

[To]

7.5.3.3 SPI System Configuration Examples

This configuration example describes that 0 level of SSL signals is active level. When connecting and using in a multi-slave mode, the transfer format of the connected device should be unified to either Motorola-SPI or TISSP.

User's Manual
7.5 Serial Peripheral Interface (RSPI)
7.5.3.3.1 Single master/single slave (this LSI = master)

[From]

7.5.3.3.1 Single master/single slave (this LSI = master)

Figure 7.5-7 shows an example of single master/single slave SPI system configuration where this LSI is used as a master. In the single master/single slave configuration, the SSL0 to SSL3 output signals of this LSI (master) are not used.

SSL input signals of the SPI slave are fixed to 0 level to always select the SPI slave. When SPCMD.CPHA = 0b, some slave devices cannot fix SSL signals to active level in the relevant transfer format. If the SSL signal level cannot be fixed, connect the SSL output of this LSI to the SSL input of the slave device.

This LSI (master) always drives the RSPCK and MOSI pins. The SPI slave always drives the MISO pin.

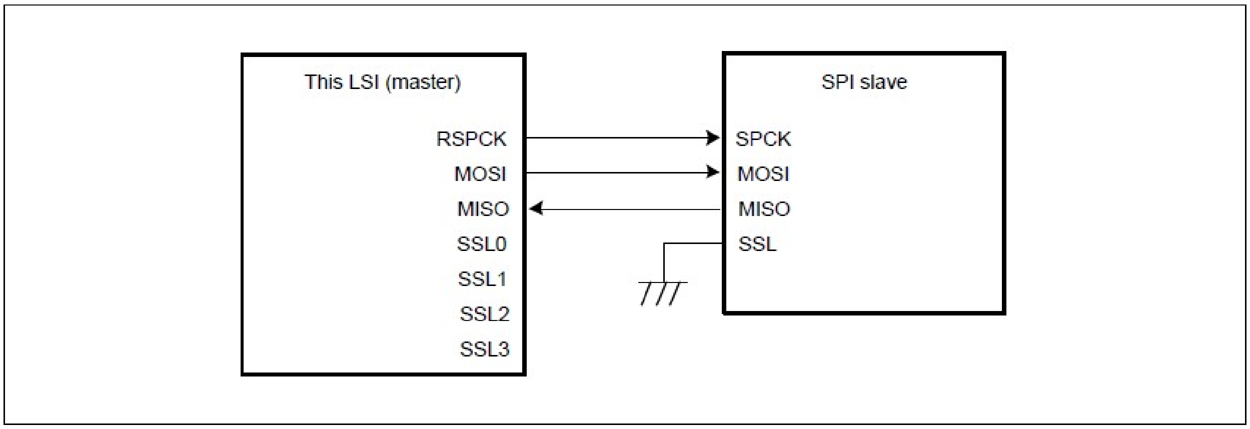


Figure 7.5-7 Single-Master/Single-Slave Configuration Example (This LSI = Master)

[To]

7.5.3.3.1 Master/single slave (this LSI = master)

Figure 7.5-7 shows an example of master/single slave SPI system configuration where this LSI is used as a master. In the master/single slave configuration, the SSL0 to SSL3 output signals of this LSI (master) are not used.

SSL input signals of the SPI slave are fixed to 0 level to always select the SPI slave. When SPCMD.CPHA = 0b, some slave devices cannot fix SSL signals to active level in the relevant transfer format. If the SSL signal level cannot be fixed, connect the SSL output of this LSI to the SSL input of the slave device.

This LSI (master) always drives the RSPCK and MOSI pins. The SPI slave always drives the MISO pin.

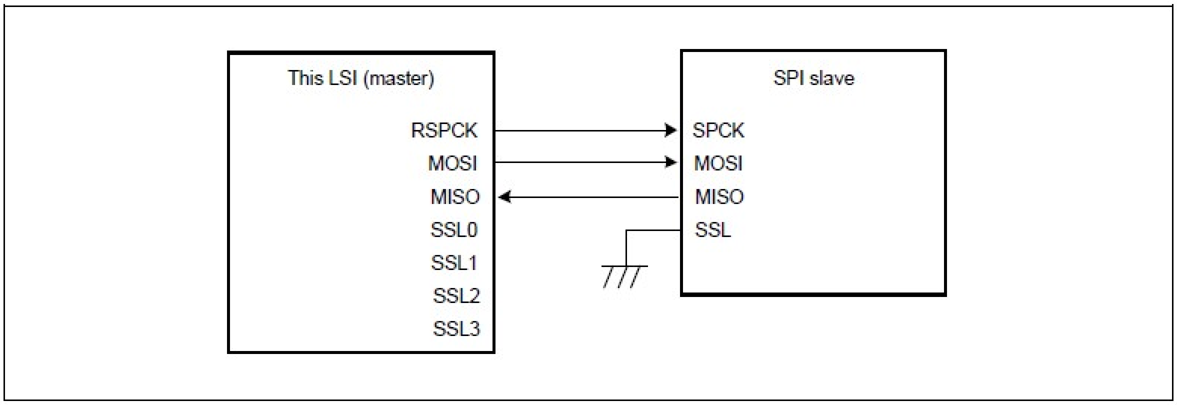


Figure 7.5-7 Master/Single-Slave Configuration Example (This LSI = Master)

User's Manual
7.5 Serial Peripheral Interface (RSPI)
7.5.3.3.2 Single master/single slave (this LSI = slave)

[From]

7.5.3.3.2 Single master/single slave (this LSI = slave)

Figure 7.5-8 shows an example of single master/single slave SPI system configuration where this LSI is used as a slave. When this LSI is used as a slave, the SSL0 pin is used as SSL input. The SPI master always drives the RSPCK and MOSI pins. This LSI (slave) always drives the MISO pin. When the SSL0 level is inactive, the pin state becomes Hi-Z.

In the single slave configuration with the SPCMD.CPHA bit set to 1b, set the SPCR.SPFRF bit to 0b, and set the SPMS bit to 0b. There is the SSL0 input level of this LSI (slave) is fixed to 0 so that this LSI (slave) can be always selected and serial transfer can also be performed (**Figure 7.5-9**). However, the communication end interrupt does not output when SSL0 input was fixed as **Figure 7.5-9**.

[To]

7.5.3.3.2 Master/single slave (this LSI = slave)

Figure 7.5-8 shows an example of master/single slave SPI system configuration where this LSI is used as a slave. When this LSI is used as a slave, the SSL0 pin is used as SSL input. The SPI master always drives the RSPCK and MOSI pins. This LSI (slave) always drives the MISO pin. When the SSL0 level is inactive, the pin state becomes Hi-Z.

In the single slave configuration with the SPCMD.CPHA bit set to 1b, set the SPCR.SPFRF bit to 0b, and set the SPMS bit to 0b. There is the SSL0 input level of this LSI (slave) is fixed to 0 so that this LSI (slave) can be always selected and serial transfer can also be performed (**Figure 7.5-9**). However, the communication end interrupt does not output when SSL0 input was fixed as **Figure 7.5-9**.

User's Manual
7.5 Serial Peripheral Interface (RSPI)
Figure 7.5-8 Single-Master/Single-Slave Configuration Example (This LSI = Slave, CPHA = 0b)

[From]

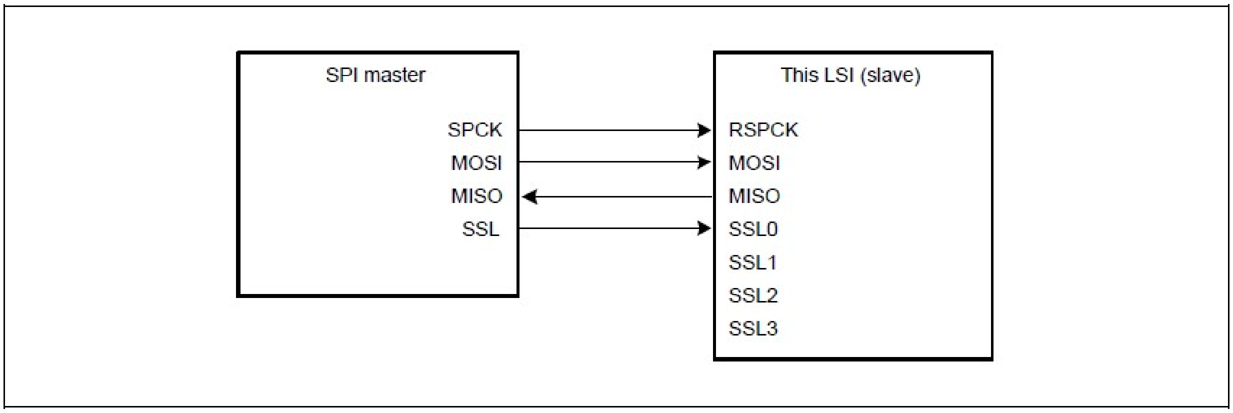


Figure 7.5-8 Single-Master/Single-Slave Configuration Example (This LSI = Slave, CPHA = 0b)

[To]

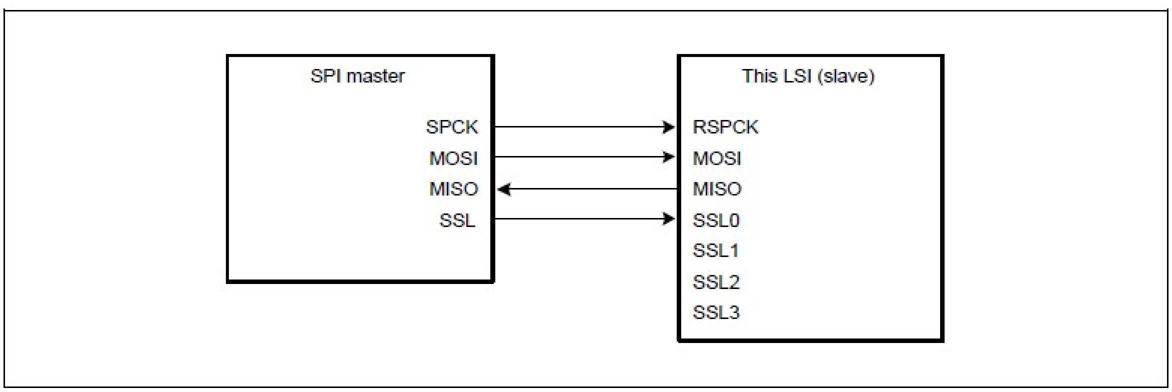


Figure 7.5-8 Master/Single-Slave Configuration Example (This LSI = Slave, CPHA = 0b)

User's Manual
7.5 Serial Peripheral Interface (RSPI)
Figure 7.5-9 Single-Master/Single-Slave Configuration Example (This LSI = Slave, CPHA = 1b)

[From]

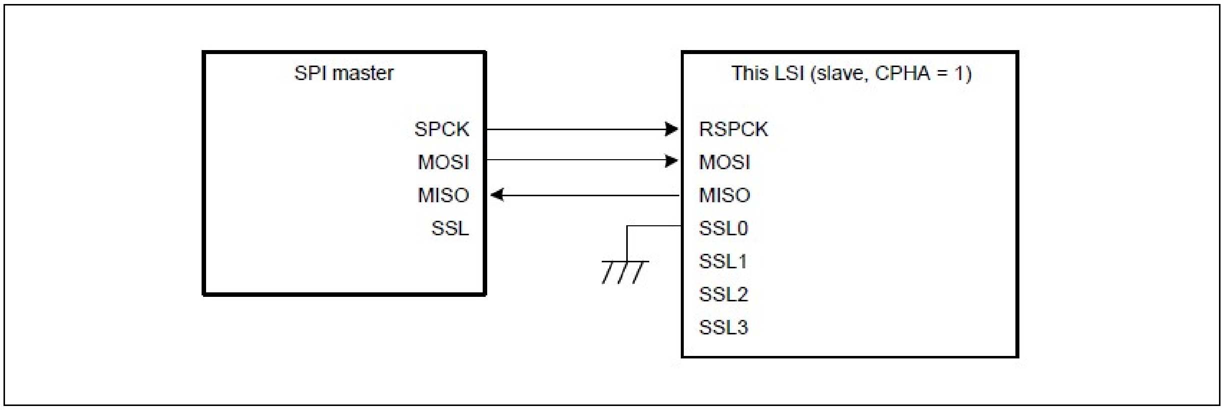


Figure 7.5-9 Single-Master/Single-Slave Configuration Example (This LSI = Slave, CPHA = 1b)

[To]

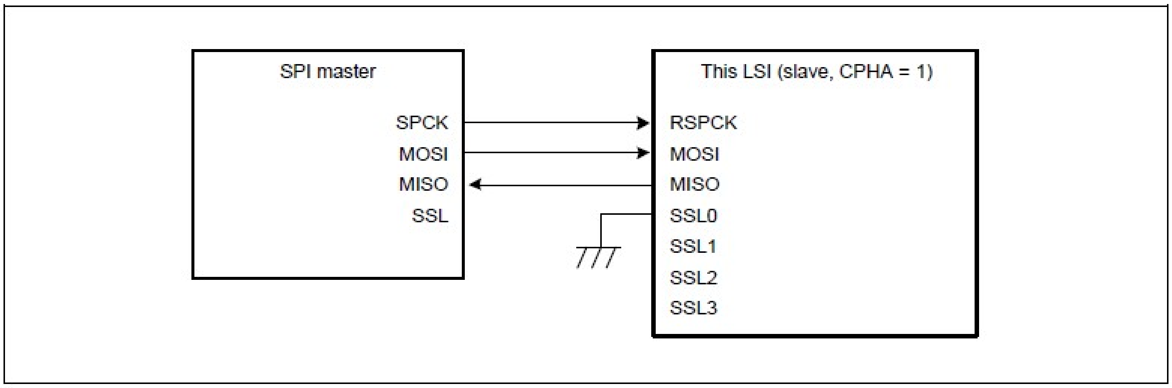


Figure 7.5-9 Master/Single-Slave Configuration Example (This LSI = Slave, CPHA = 1b)

User's Manual
7.5 Serial Peripheral Interface (RSPI)
7.5.3.3.3 Single master/multi-slave (this LSI = master)

[From]

7.5.3.3.3 Single master/multi-slave (this LSI = master)

Figure 7.5-10 shows an example of single master/multi-slave SPI system configuration where this LSI is used as a master. In the example in **Figure 7.5-10**, the SPI system is comprised of this LSI (master) and four slaves (SPI slave 0 to SPI slave 3).

[To]

7.5.3.3.3 Master/multi-slave (this LSI = master)

Figure 7.5-10 shows an example of master/multi-slave SPI system configuration where this LSI is used as a master. In the example in **Figure 7.5-10**, the SPI system is comprised of this LSI (master) and four slaves (SPI slave 0 to SPI slave 3).

User's Manual
7.5 Serial Peripheral Interface (RSPI)
Figure 7.5-10 Single-Master/Multi-Slave Configuration Example (This LSI = Master)

[From]

Figure 7.5-10 Single-Master/Multi-Slave Configuration Example (This LSI = Master)

[To]

Figure 7.5-10 Master/Multi-Slave Configuration Example (This LSI = Master)

User's Manual
7.5 Serial Peripheral Interface (RSPI)
7.5.3.3.4 Single master/multi-slave (with this LSI acting as slave)

[From]

7.5.3.3.4 Single master/multi-slave (with this LSI acting as slave)

Figure 7.5-11 shows an example of single master/multi-slave SPI system configuration where this LSI is used as a slave. In the example in **Figure 7.5-11**, the SPI system consists of the SPI master and two these LSIs (slave X, slave Y).

The RSPCK output and MOSI output pins of the SPI master are connected to the RSPCK input and MOSI input pins of the LSIs (slave X, slave Y). The MISO output pin of the LSIs (slave X, slave Y) is connected to the MISO input pin of the SPI master. The SSLX output and SSLY output pins of the SPI master are connected to the SSL0 input pin of the LSIs (slave X, slave Y).

The SPI master always drives the RSPCK, MOSI, SSLX, and SSLY pins. The slave X or slave Y (this LSI) where 0 level is input to the SSL0 input pin drives the MISO pin.

[To]

7.5.3.3.4 Master/multi-slave (with this LSI acting as slave)

Figure 7.5-11 shows an example of master/multi-slave SPI system configuration where this LSI is used as a slave. In the example in **Figure 7.5-11**, the SPI system consists of the SPI master and two these LSIs (slave X, slave Y).

The RSPCK output and MOSI output pins of the SPI master are connected to the RSPCK input and MOSI input pins of the LSIs (slave X, slave Y). The MISO output pin of the LSIs (slave X, slave Y) is connected to the MISO input pin of the SPI master. The SSLX output and SSLY output pins of the SPI master are connected to the SSL0 input pin of the LSIs (slave X, slave Y).

The SPI master always drives the RSPCK, MOSI, SSLX, and SSLY pins. The slave X or slave Y (this LSI) where 0 level is input to the SSL0 input pin drives the MISO pin.

User's Manual
7.5 Serial Peripheral Interface (RSPI)
Figure 7.5-11 Single-Master/Multi-Slave Configuration Example (This LSI = Slave)

[From]

Figure 7.5-11 Single-Master/Multi-Slave Configuration Example (This LSI = Slave)

[To]

Figure 7.5-11 Master/Multi-Slave Configuration Example (This LSI = Slave)

User's Manual
7.5 Serial Peripheral Interface (RSPI)
7.5.3.3.5 Multi-master/multi-slave (with this LSI acting as master)

[From]

7.5.3.3.5 Multi-master/multi-slave (with this LSI acting as master)

Figure 7.5-12 shows an example of multi-master/multi-slave SPI system configuration where this LSI is used as a master. In the example in **Figure 7.5-12**, the SPI system consists of two these LSIs (master X, master Y) and two SPI slaves (SPI slave 1, SPI slave 2).

The RSPCK output and MOSI output pins of these LSIs (master X, master Y) are connected to the RSPCK input and MOSI input pins of SPI slave 1 and SPI slave 2. The MISO output pin of SPI slave 1 and SPI slave 2 is connected to the MISO input pin of these LSIs (master X, master Y). The Port Y (general port) output pin of this LSI (master X) is connected to the SSL0 input pin of this LSI (master Y). The Port X (general port) output pin of this LSI (master Y) is connected to the SSL0 input pin of this LSI (master X). The SSL1 output and SSL2 output pins of these LSIs (master X, master Y) are connected to the SSL input pin of SPI slave 1 and SPI slave 2. In this configuration example, SSL3 output pins of this LSI are not used because the system can be configured only with SSL0 input pin and SSL1 output and SSL2 output pins for connecting slaves.

While the SSL0 input level is 1, this LSI drives the RSPCK, MOSI, SSL1, and SSL2 pins. While the SSL0 input level is 0, this LSI detects a mode fault error and changes the RSPCK, MOSI, SSL1, SSL2 pin levels to Hi-Z to release the SPI bus mastership for another master. SPI slave 1 or SPI slave 2 where 0 level is input to the SSL input pin drives the MISO pin.

[To]
This section has deleted

User's Manual
7.5 Serial Peripheral Interface (RSPI)
Table 7.5-10 Abnormal Transfer Occurrence Conditions and Error Detection Function of SPI

[From]

Table 7.5-11 Abnormal Transfer Occurrence Conditions and Error Detection Function of SPI

	Transfer Occurrence Condition	SPI Operation	Error Detection
1	SPDR is written when while no empty stages in the transmit FIFO.	• Transmit buffer data is retained. • No write data is present.	None
2	SPDR is read while no data stored in receive FIFO.	Received data and previously received serial data are read.	None
3	Serial transfer starts in transmit slave mode or transmit-only slave mode, before transmit data output is ready.	• Serial transfer is suspended. • No transmit data or receive data is present. • Driving of the MISO output signal is stopped. • SPI function is disabled.	Underrun error
4	Serial transfer ends when data is stored in the receive FIFO for the number of FIFO stages.	• Receive FIFO data is retained. • No serial receive data is present.	Overrun error
5	Incorrect parity bit has been received with the parity function enabled in following mode. • Transmit/receive-only master mode • Transmit/receive slave mode • Receive-only slave mode	The parity error flag is asserted.	Parity error
6	The SSL0 input signal is asserted when the serial transfer is idle state in multi-master mode.	• Driving of the RSPCK, MOSI, SSL1 to SSL3 output signals is stopped. • SPI function is disabled.	Mode fault error
7	The SSL0 input signal is asserted during serial transfer in multi-master mode.	• Serial transfer is suspended. • Missing transmit/receive data. • Driving of the RSPCK, MOSI, SSL1 to SSL3 output signals is stopped. • SPI function is disabled.	Mode fault error
8	[In the Motorola-SPI case] The SSL0 input signal is negated during serial transfer in slave mode.	• Serial transfer is suspended. • No transmit data or receive data is present. • Driving of the MISO output signal is stopped. • SPI function is disabled.	Mode fault error
9	[In the TI-SSP case] The SSL0 input signal is asserted during serial transfer in slave mode.	• Serial transfer is suspended. • No transmit data or receive data is present. • Driving of the MISO output signal is stopped. • SPI function is disabled.	Mode fault error

[To]

Table 7.5-10 Abnormal Transfer Occurrence Conditions and Error Detection Function of SPI

	Transfer Occurrence Condition	SPI Operation	Error Detection
1	SPDR is written when while no empty stages in the transmit FIFO.	• Transmit buffer data is retained. • No write data is present.	None
2	SPDR is read while no data stored in receive FIFO.	Received data and previously received serial data are read.	None
3	Serial transfer starts in transmit slave mode or transmit-only slave mode, before transmit data output is ready.	• Serial transfer is suspended. • No transmit data or receive data is present. • Driving of the MISO output signal is stopped. • SPI function is disabled.	Underrun error
4	Serial transfer ends when data is stored in the receive FIFO for the number of FIFO stages.	• Receive FIFO data is retained. • No serial receive data is present.	Overrun error
5	Incorrect parity bit has been received with the parity function enabled in following mode. • Transmit/receive-only master mode • Transmit/receive slave mode • Receive-only slave mode	The parity error flag is asserted.	Parity error
6	[In the Motorola-SPI case] The SSL0 input signal is negated during serial transfer in slave mode.	• Serial transfer is suspended. • No transmit data or receive data is present. • Driving of the MISO output signal is stopped. • SPI function is disabled.	Mode fault error
7	[In the TI-SSP case] The SSL0 input signal is asserted during serial transfer in slave mode.	• Serial transfer is suspended. • No transmit data or receive data is present. • Driving of the MISO output signal is stopped. • SPI function is disabled.	Mode fault error

User's Manual
7.5 Serial Peripheral Interface (RSPI)
7.5.3.10.3 Mode fault error

[From]

7.5.3.10.3 Mode fault error

When the SPCR.MSTR = 1b, SPCR.SPMS = 0b, and SPCR.MODFEN = 1b, the SPI operates in multi-master mode. When an active level is input to the SSL0 input signal of the SPI in multi-master mode, the SPI detects a mode fault error regardless of serial transfer status and sets the MODF flag in the SPSR register to 1. When a mode fault error is detected, the SPI copies the value of pointer to the SPCMD register to the SPSSR.SPECM[2:0] bits. The SSL0 signal active level depends on the SSL0P bit in the SSLP register.

While SPCR.MSTR = 0b, the SPI operates in slave mode. When SPCR.SPMS = 0b and SPCR.MODFEN = 1b in slave mode, if the SSL0 input signal is negated during the serial transfer period (from valid data drive start to final valid data latch), the SPI detects a mode fault error while any of the following 2 conditions is met.

(1) In the Motorola-SPI case

When the SSL0 input signal is negated while serial data transfer.

(2) In the TI-SSP case

When the SSL0 input signal is asserted while serial data transfer. However, during burst transfer, no error is detected even if the SSL0 input signal is asserted during the last bit of frame.

When the SPI detects a mode fault error, it stops driving output signals and clears the SPCR.SPE bit.

When the SPE bit is cleared, the SPI function is disabled (as described in **7.5.3.12 SPI Initialization**). In a multi-master configuration, the mastership can be released by stopping driving output signals and disabling the SPI function by using a mode fault error.

Whether a mode fault error is present can be checked by reading SPSR or by reading an SPI error interrupt and SPSR. To detect a mode fault error without using an SPI error interrupt, poll SPSR. When the SPI is used in master mode, the pointer value to SPCMD when an error is present can be checked by reading the SPSSR.SPECM[2:0] bits.

While the MODF flag = 1b, the SPI ignores writing 1b to the SPE bit. To enable the SPI function after a mode fault error is detected, clear the MODF flag to 0b without fail.

[To]

7.5.3.10.3 Mode fault error

When the SPCR.MSTR = 0b, the SPI operates in slave mode. When SPCR.SPMS = 0b and SPCR.MODFEN = 1b in slave mode, if the SSL0 input signal is negated during the serial transfer period (from valid data drive start to final valid data latch), the SPI detects a mode fault error while any of the following 2 conditions is met.

(1) In the Motorola-SPI case

When the SSL0 input signal is negated while serial data transfer.

(2) In the TI-SSP case

When the SSL0 input signal is asserted while serial data transfer. However, during burst transfer, no error is detected even if the SSL0 input signal is asserted during the last bit of frame.

When the SPI detects a mode fault error, it stops driving output signals and clears the SPCR.SPE bit.

When the SPE bit is cleared, the SPI function is disabled (as described in **7.5.3.12 SPI Initialization**).

Whether a mode fault error is present can be checked by reading SPSR or by reading an SPI error interrupt and SPSR. To detect a mode fault error without using an SPI error interrupt, poll SPSR.

While the MODF flag = 1b, the SPI ignores writing 1b to the SPE bit. To enable the SPI function after a mode fault error is detected, clear the MODF flag to 0b without fail.

User's Manual
7.5 Serial Peripheral Interface (RSPI)
7.5.3.13.1 Master mode operation

[From]

7.5.3.13.1 Master mode operation

Single master mode operation and multi-master mode operation are different from each other only in mode fault error detection (see **7.5.3.10 Error Detection**). The SPI in single master mode (SPI) detects no mode fault error.

The SPI in multi-master mode detects a mode fault error. The following describes operations common to single master mode and multi-master mode.

(1) Starting a serial transfer

When data is written to the SPDR register while the next transfer data is not set in the transmit FIFO, the SPI updates the transmit buffer (SPTXn, n = 0 to 15) data in SPDR.

While the shift register is empty, the SPI copies transmit buffer data to the shift register to start serial transfer. After the SPI copies transmit data to the shift register, it changes the shift register status to full. Upon completion of serial transfer, the SPI changes the shift register status to empty. The shift register status cannot be monitored.

For details about the SPI transfer format, see **7.5.3.5 Transfer Format (Frame Format)**. The SSL output signal polarity depends on the set SSLP register value.

[To]

7.5.3.13.1 Master mode operation

(1) Starting a serial transfer

When data is written to the SPDR register while the next transfer data is not set in the transmit FIFO, the SPI updates the transmit buffer (SPTXn, n = 0 to 15) data in SPDR.

While the shift register is empty, the SPI copies transmit buffer data to the shift register to start serial transfer. After the SPI copies transmit data to the shift register, it changes the shift register status to full. Upon completion of serial transfer, the SPI changes the shift register status to empty. The shift register status cannot be monitored.

For details about the SPI transfer format, see **7.5.3.5 Transfer Format (Frame Format)**. The SSL output signal polarity depends on the set SSLP register value.

User's Manual
7.5 Serial Peripheral Interface (RSPI)
Figure 7.5-63 Example of Initialization Flowchart in Master Mode (SPI Operation)

[From]

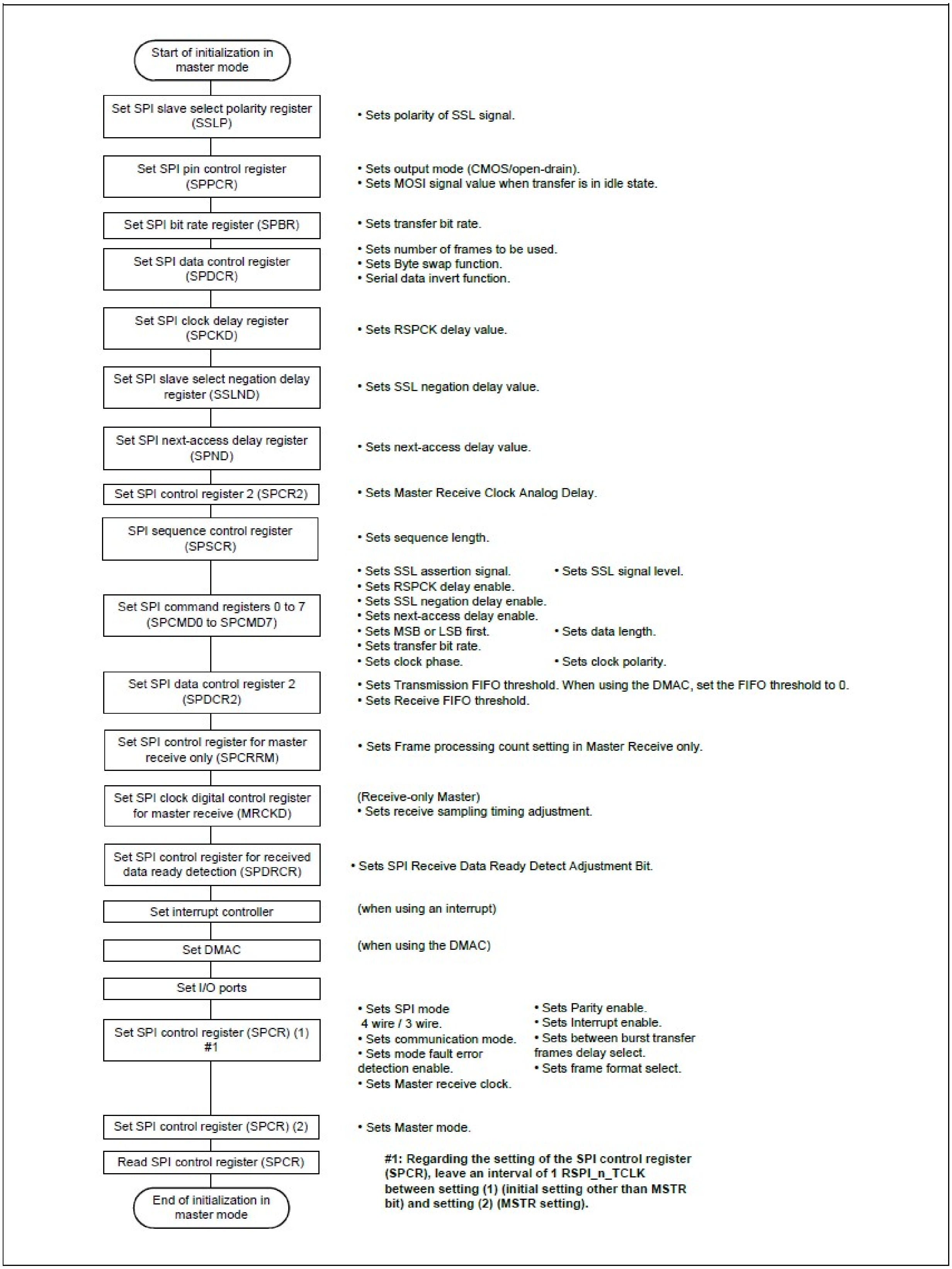


Figure 7.5-64 Example of Initialization Flowchart in Master Mode (SPI Operation)

[To]

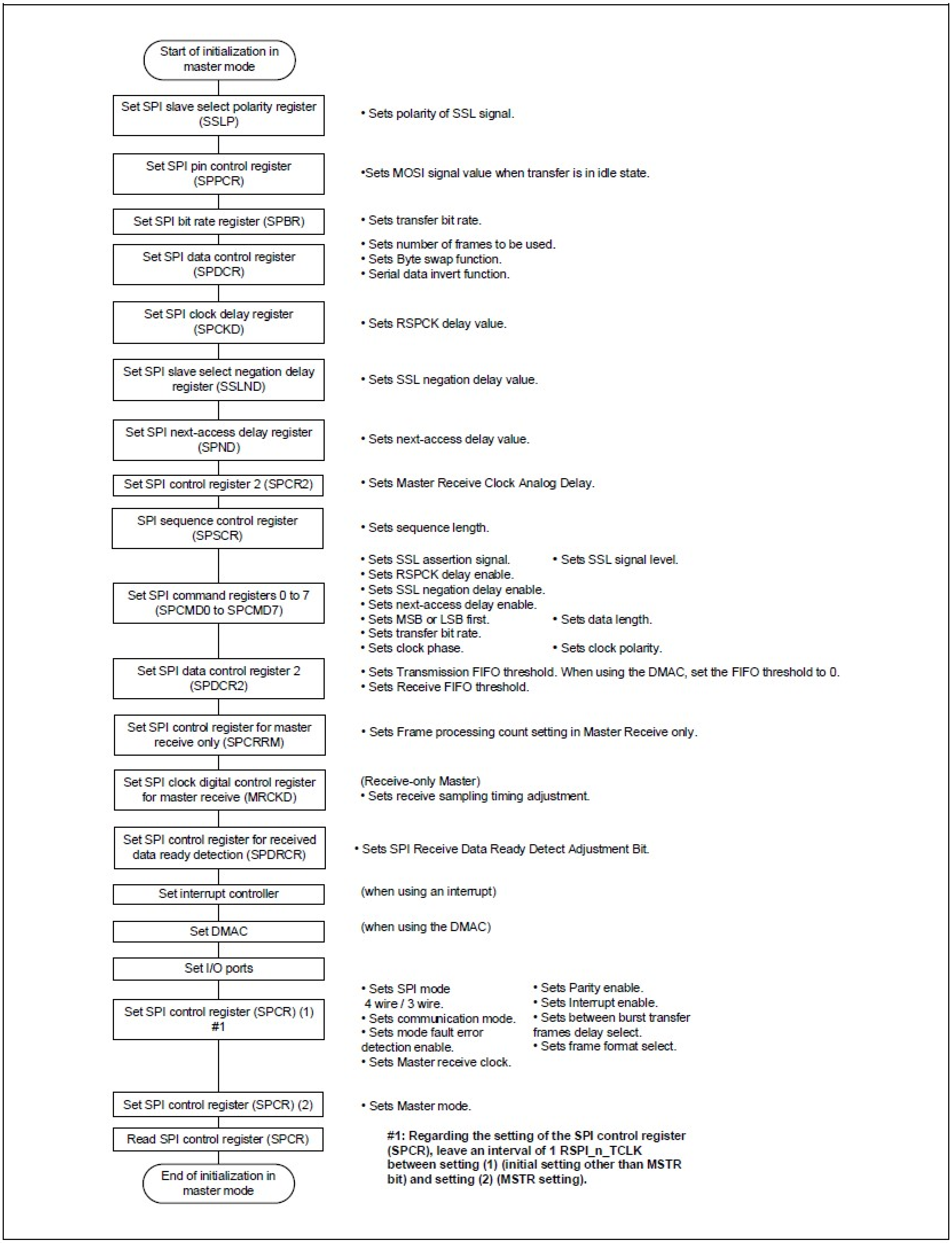


Figure 7.5-63 Example of Initialization Flowchart in Master Mode (SPI Operation)

User's Manual
7.5 Serial Peripheral Interface (RSPI)
Figure 7.5-69 Example of Initialization Flow in Slave Mode

[From]

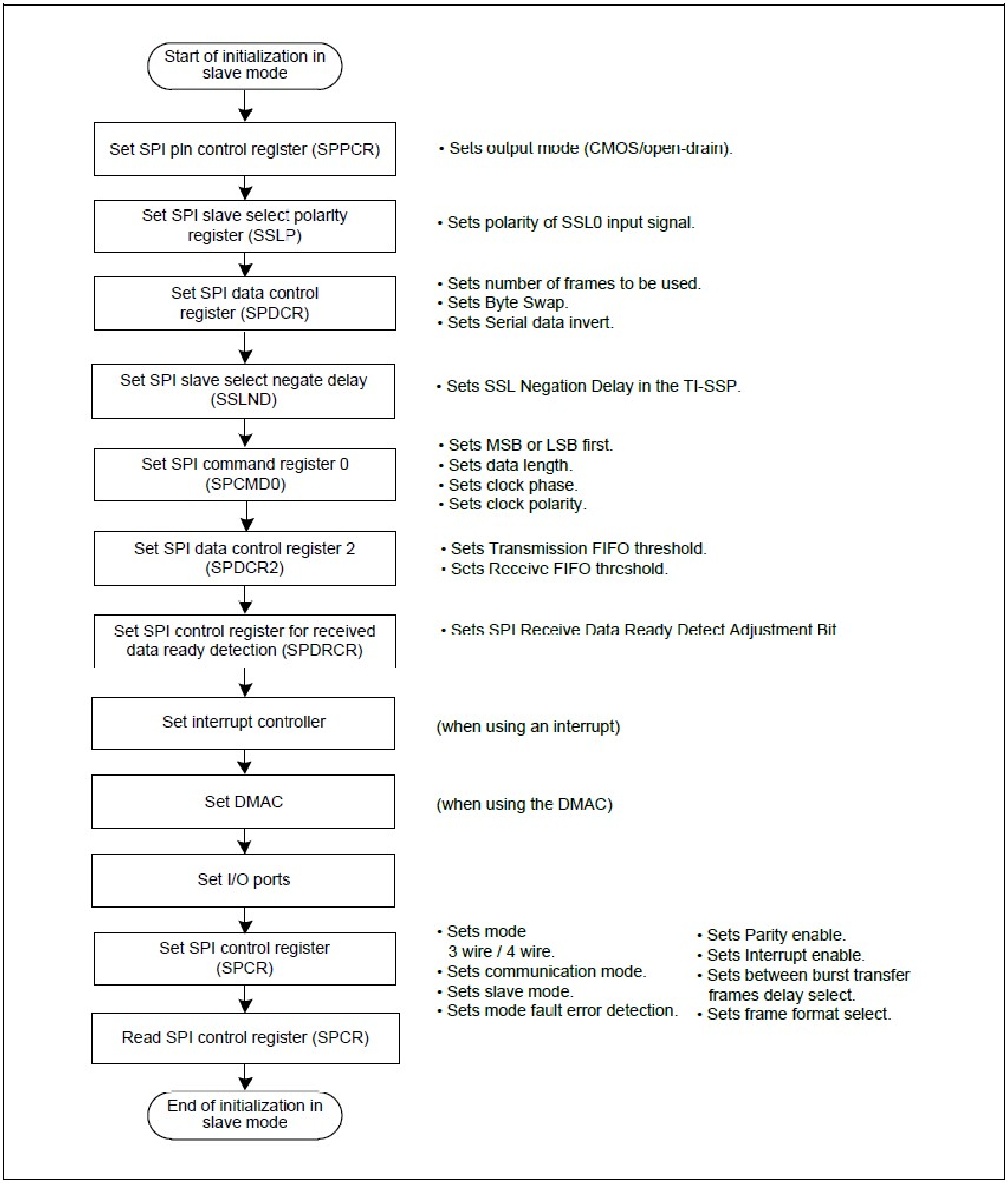


Figure 7.5-69 Example of Initialization Flow in Slave Mode

[To]

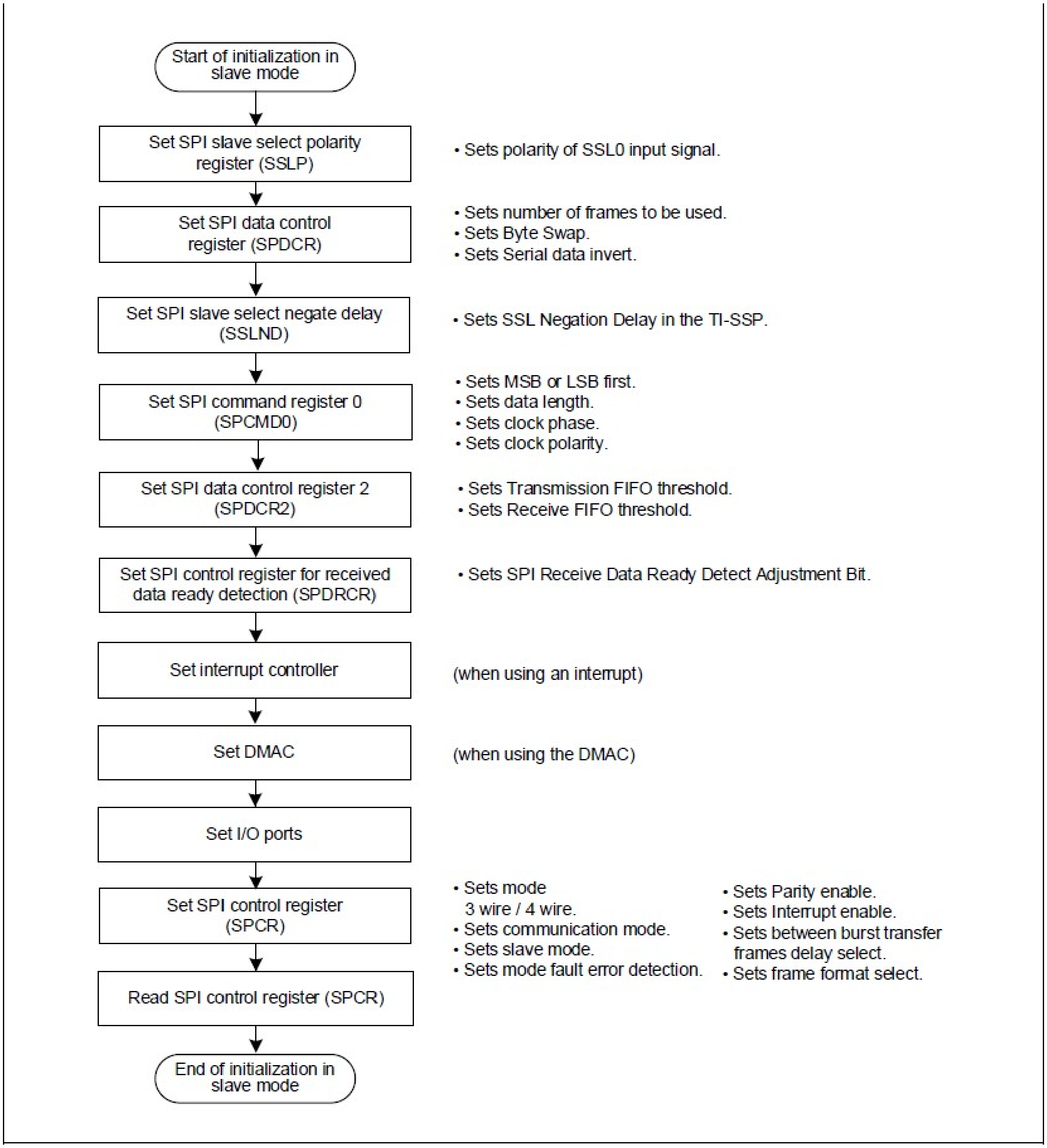


Figure 7.5-68 Example of Initialization Flow in Slave Mode

User's Manual
7.5 Serial Peripheral Interface (RSPI)
Figure 7.5-77 Example of Initialization Flowchart in Master Mode (Clock Synchronous Operation)

[From]

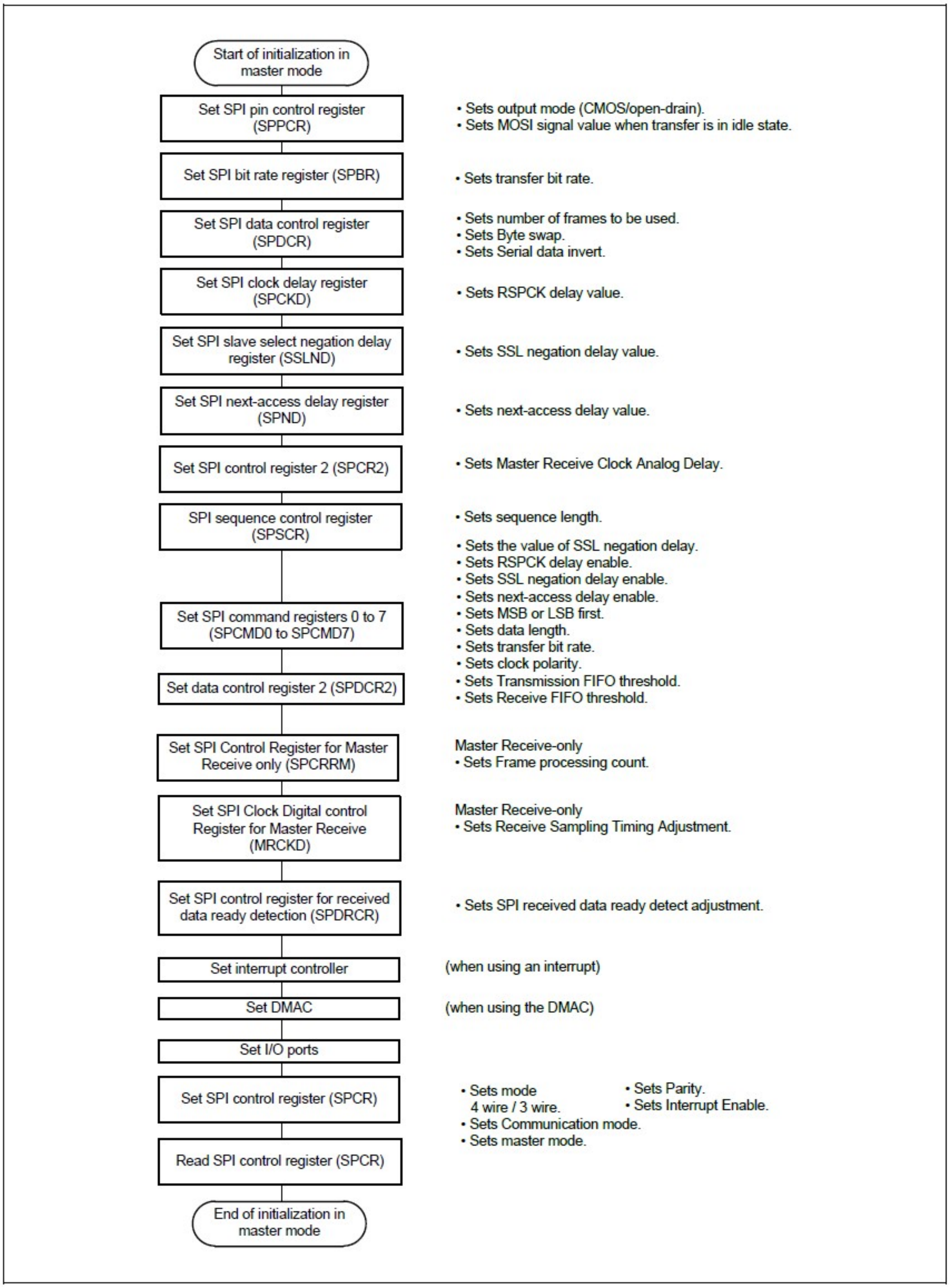


Figure 7.5-77 Example of Initialization Flowchart in Master Mode (Clock Synchronous Operation)

[To]

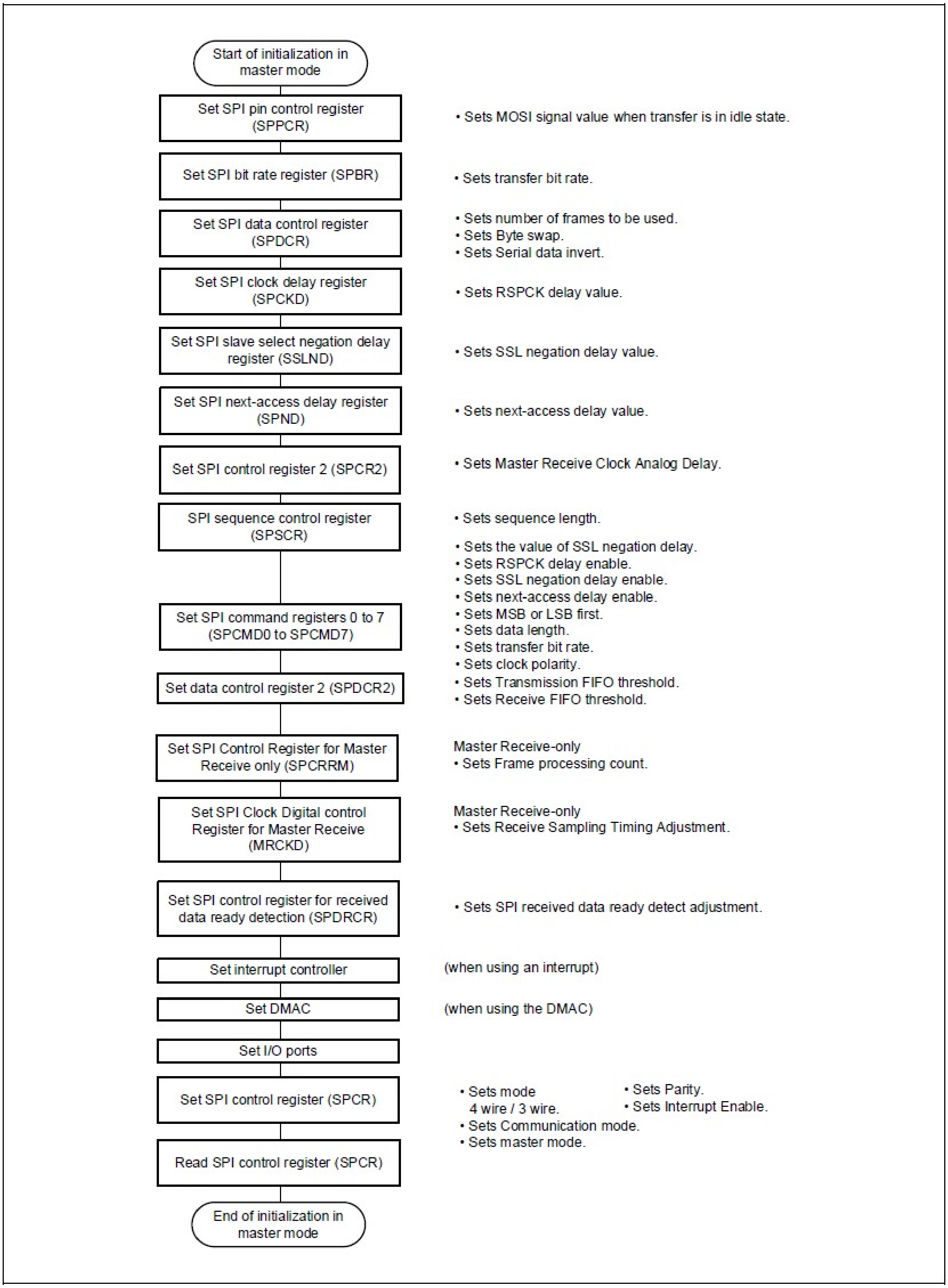


Figure 7.5-76 Example of Initialization Flowchart in Master Mode (Clock Synchronous Operation)

User's Manual
7.5 Serial Peripheral Interface (RSPI)
Figure 7.5-78 Example of Initialization Flow in Slave Mode

[From]

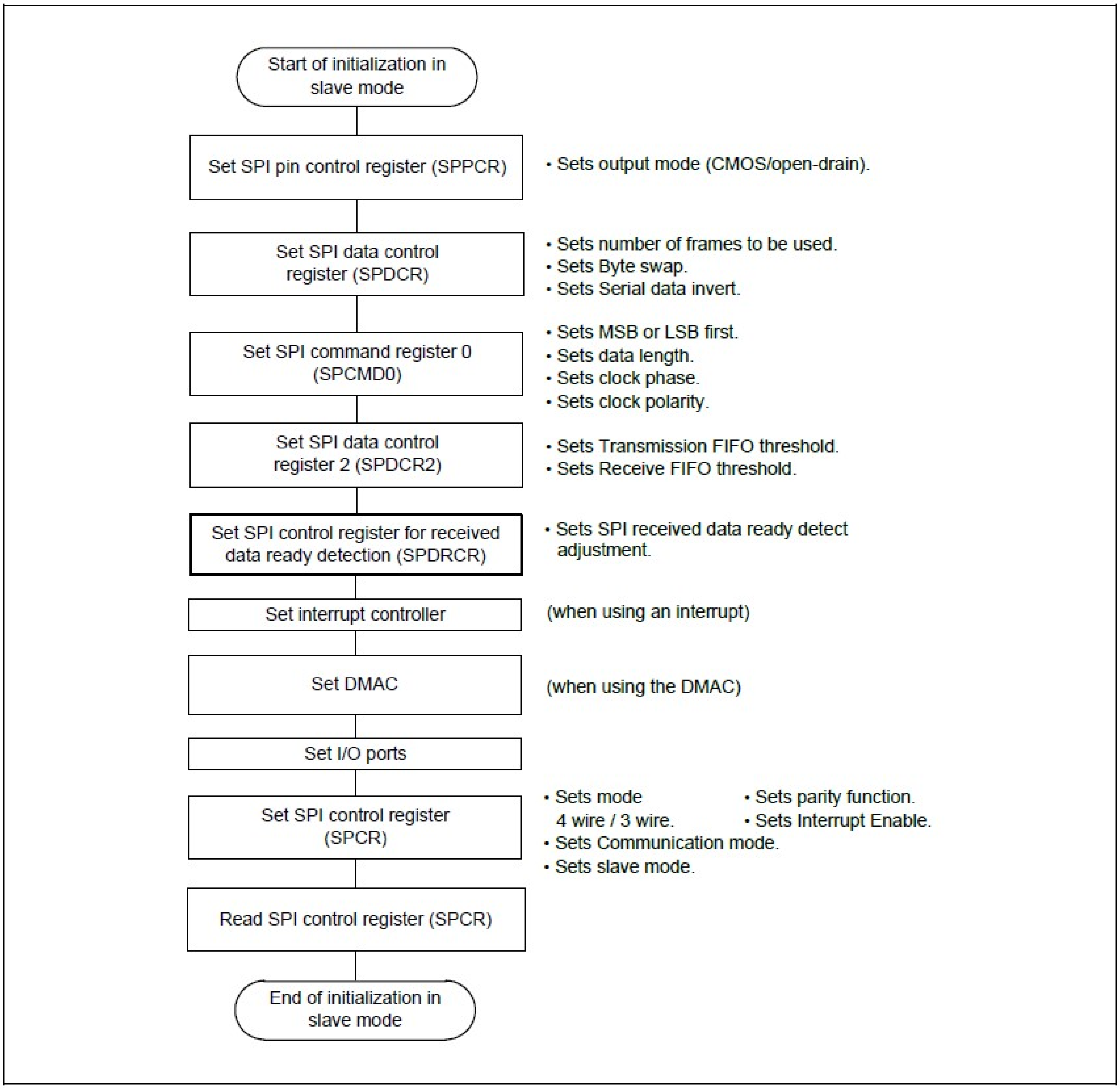


Figure 7.5-78 Example of Initialization Flow in Slave Mode

[To]

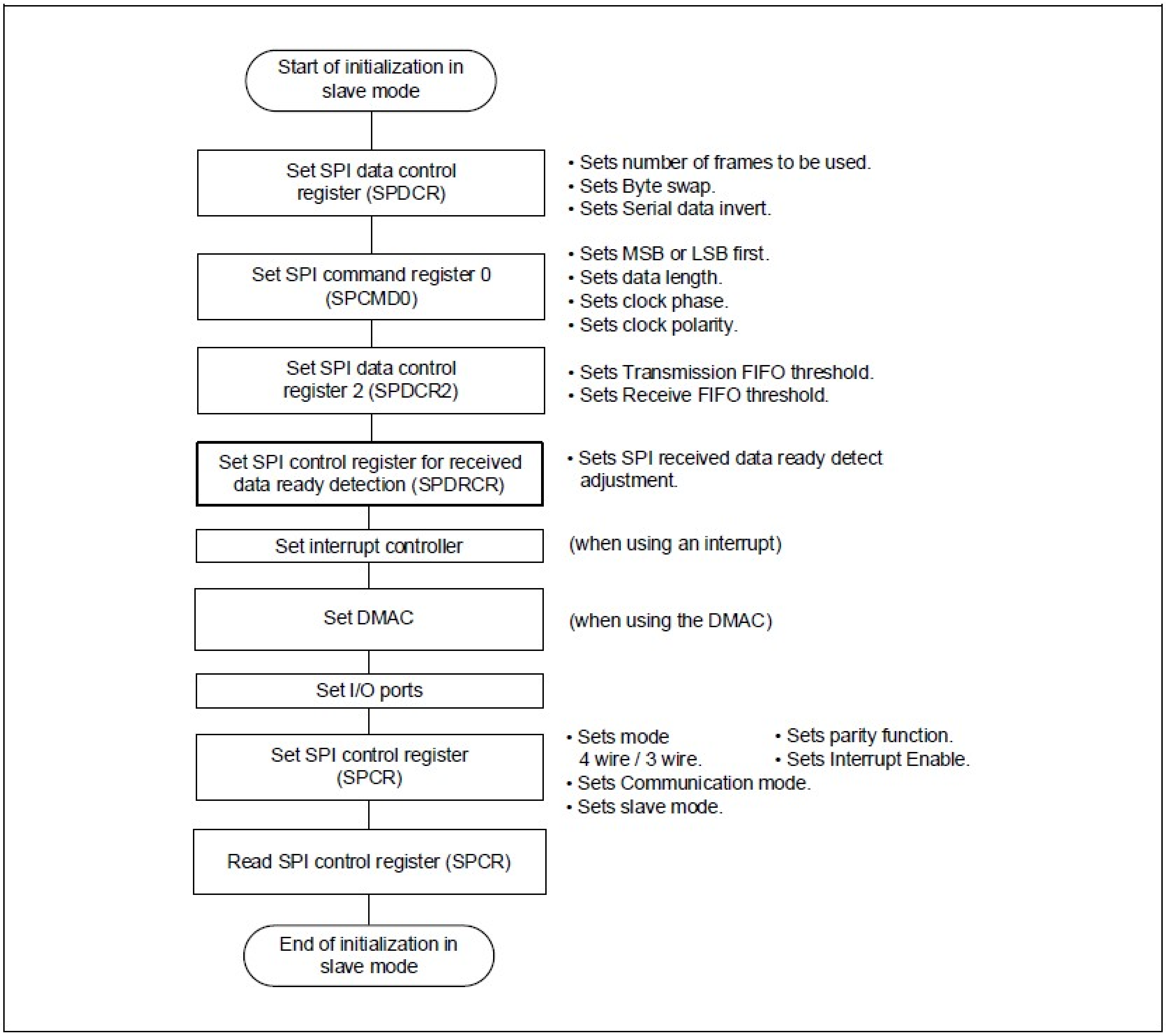


Figure 7.5-77 Example of Initialization Flow in Slave Mode

User's Manual
7.5 Serial Peripheral Interface (RSPI)
Table 7.5-17 Communication End Event Generating Conditions (Receive Only Slave Mode)

[From]

Table 7.5-17 Communication End Event Generating Conditions (Receive Only Slave Mode) (1/2)

Register Name	Bit Name
SPCKD	SCKDL[2:0]
SSLND	SLNDL[2:0]
SPND	SPNDL[2:0]
MRCKD	ARST[2:0]
SPCR	BPEN
	MSTR
	TXMD[1:0]
	SPFRF
	SPMS
	MODFEN
	BFDS
	SCKASE
	PTE
	SPOE
	SPPE
	SPSCKSEL
SPCRRM	RMFM[4:0]
SPDRCR	SPDRC[7:0]
SPPCR	MOIFE
	MOIFV
	SPOM
	SPLP2
	SPLP
SPCR2	SPSCKDL[2:0]
SSLP	SSL3P
	SSL2P
	SSL1P
	SSL0P
SPBR	SPR[7:0]
SPSCR	SPSLN[2:0]
SPCMD0*[1]	SSLA[1:0]
	SPB[4:0]
	SCKDEN
	SLNDEN
	SPNDEN
	LSBF
	SSLKP
	BRDV[1:0]
	CPOL
	CPHA

Table 7.5-17 Communication End Event Generating Conditions (Receive Only Slave Mode) (2/2)

Register Name	Bit Name
SPDCR	SPFC[1:0]
	SINV
	SPRDTD
	SLSEL[1:0]
	BYSW
SPDCR2	TTRG[1:0]
	RTRG[1:0]
SPFCR	SPFRST

Note 1. Rewriting prohibited in slave mode. In master mode, this is possible only when there is no next transfer data in the transmit FIFO.

[To]

Table 7.5-16 Communication End Event Generating Conditions (Receive Only Slave Mode) (1/2)

Register Name	Bit Name
SPCKD	SCKDL[2:0]
SSLND	SLNDL[2:0]
SPND	SPNDL[2:0]
MRCKD	ARST[2:0]
SPCR	BPEN
	MSTR
	TXMD[1:0]
	SPFRF
	SPMS
	MODFEN
	BFDS
	SCKASE
	PTE
	SPOE
	SPPE
	SPSCKSEL
SPCRRM	RMFM[4:0]
SPDRCR	SPDRC[7:0]
SPPCR	MOIFE
	MOIFV
	SPLP2
	SPLP
SPCR2	SPSCKDL[2:0]
SSLP	SSL3P
	SSL2P
	SSL1P
	SSL0P
SPBR	SPR[7:0]
SPSCR	SPSLN[2:0]
SPCMD0*[1]	SSLA[1:0]
	SPB[4:0]
	SCKDEN
	SLNDEN
	SPNDEN
	LSBF
	SSLKP
	BRDV[1:0]
	CPOL
	CPHA

Table 7.5-16 Communication End Event Generating Conditions (Receive Only Slave Mode) (2/2)

Register Name	Bit Name
SPDCR	SPFC[1:0]
	SINV
	SPRDTD
	BYSW
SPDCR2	TTRG[1:0]
	RTRG[1:0]
SPFCR	SPFRST

Note 1. Rewriting prohibited in slave mode. In master mode, this is possible only when there is no next transfer data in the transmit FIFO.

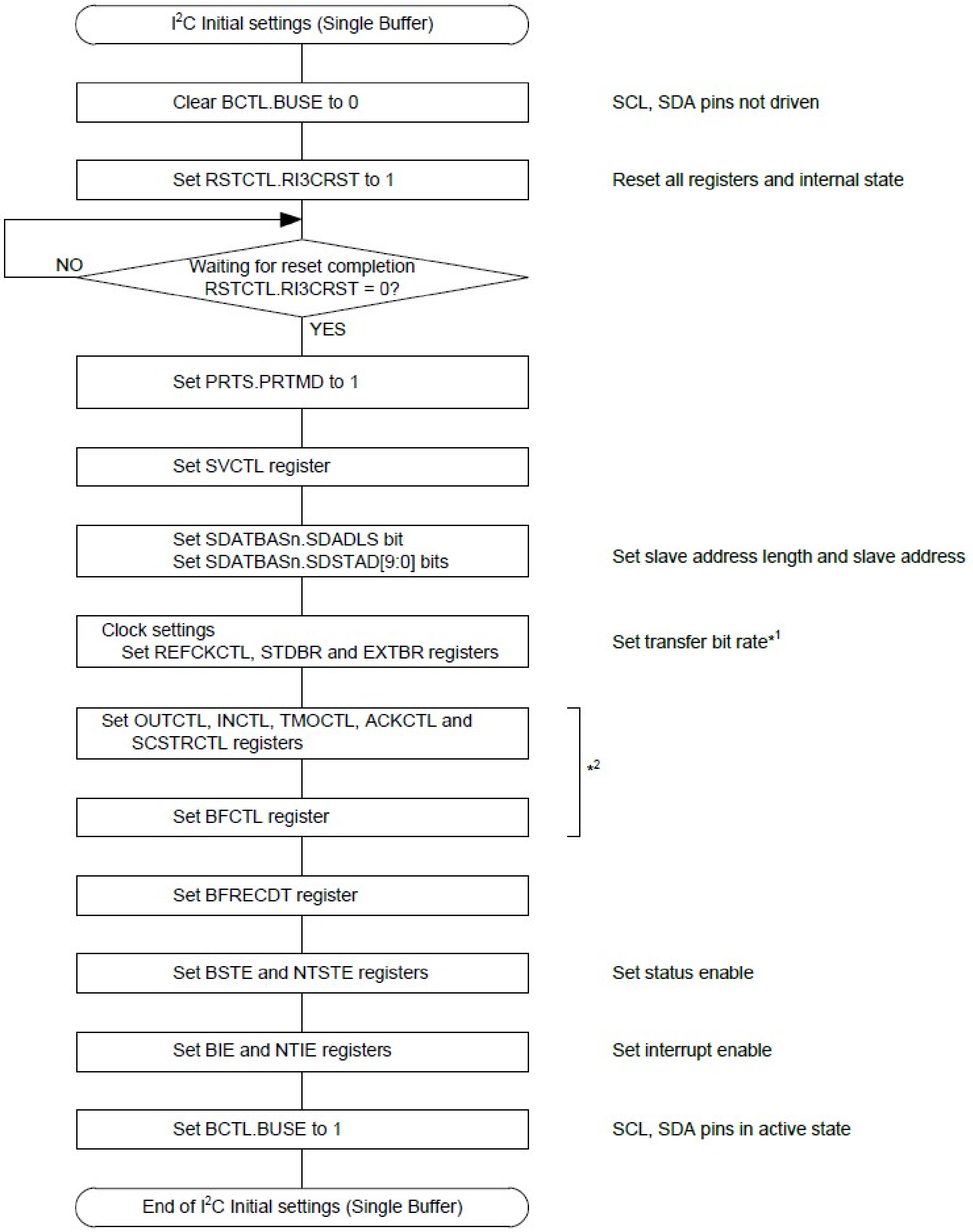
[Title]
I3C initialization setting flow correction.

[Phenomenon]
In the current initialization flow, the initial state of the communication I/F pins may become unstable.

[User's manual Update]
Add an internal reset assert to the initialization flow to mask the unstable period of the I/F pins.

User's Manual
7.8 I3C Bus Interface (I3C)
7.8.4.3.1 Initial Setting Flow
Figure 7.8-111 Example of I2C Initialization Flowchart (Single Buffer Transfer)

[From]



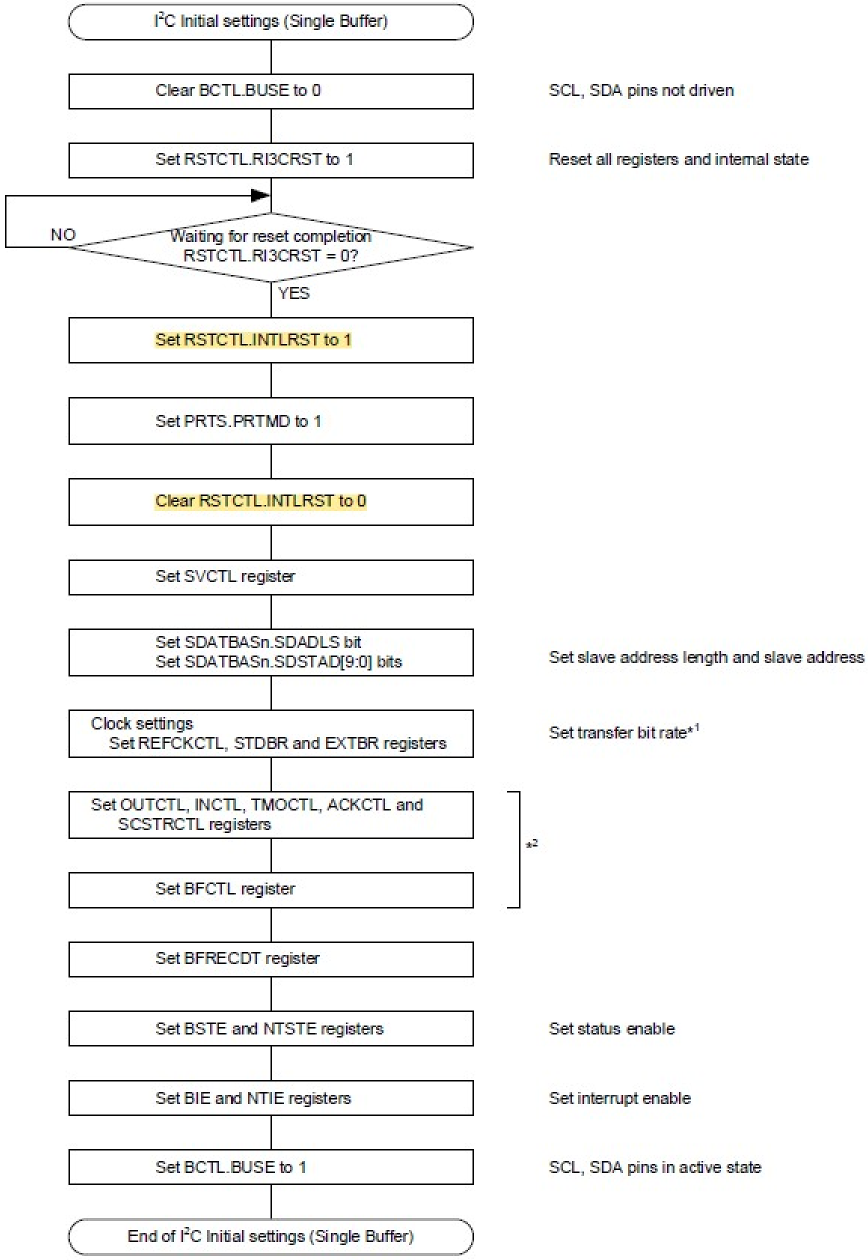
Note: n = 0 to 2

Note 1. When the I3C is used only in slave mode, set the STDBR register to a value longer than the data setup time.

Note 2. Set these registers as necessary.

Figure 7.8-111 Example of I^{2}C Initialization Flowchart (Single Buffer Transfer)

[To]



Note: n = 0 to 2

Note 1. When the I3C is used only in slave mode, set the STDBR register to a value longer than the data setup time.

Note 2. Set these registers as necessary.

Figure 7.8-111 Example of I²C Initialization Flowchart (Single Buffer Transfer)

User's Manual
7.8 I3C Bus Interface (I3C)
7.8.4.3.1 Initial Setting Flow
Figure 7.8-112 Example of I3C initialization flowchart

[From]

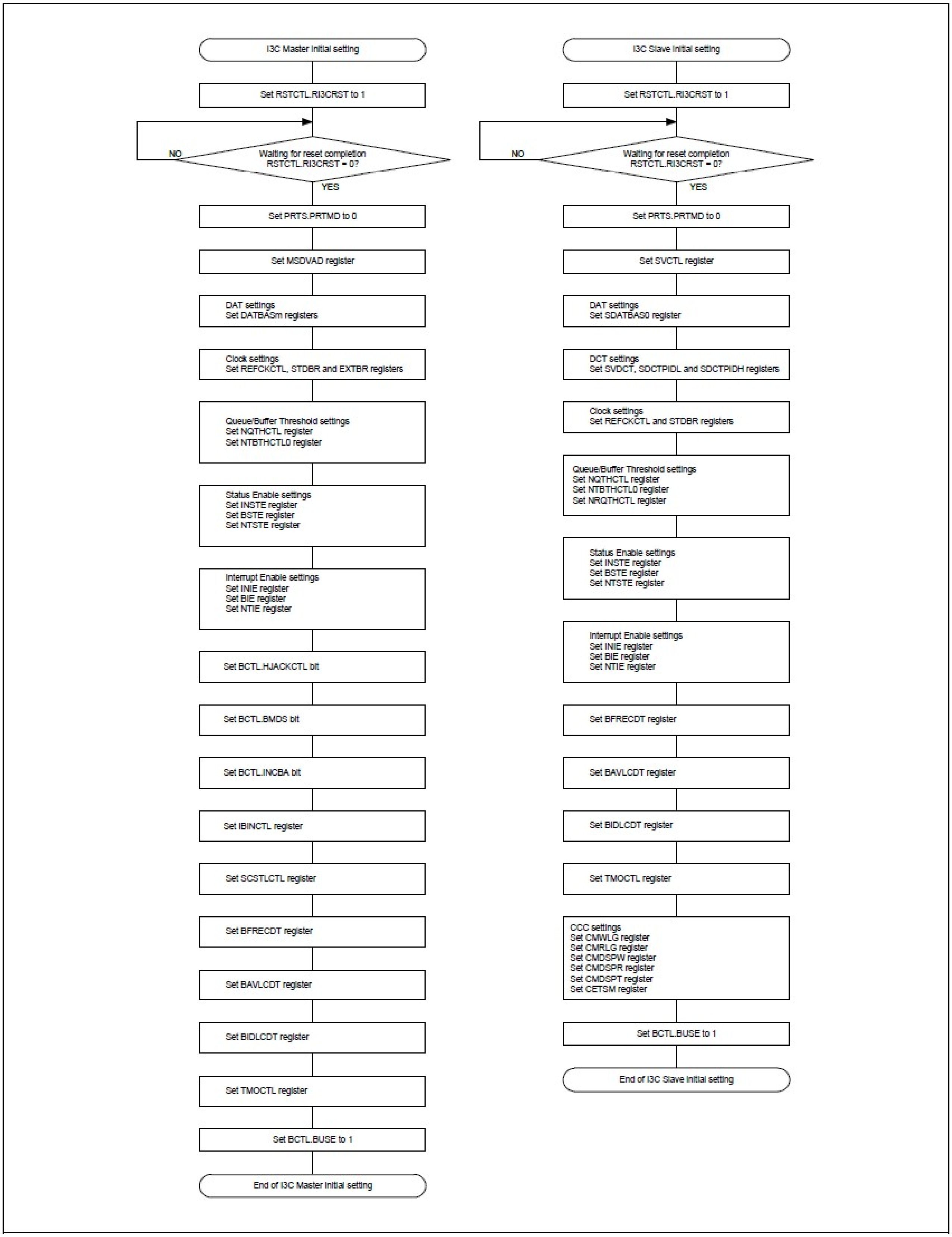


Figure 7.8-112 Example of I3C initialization flowchart

[To]

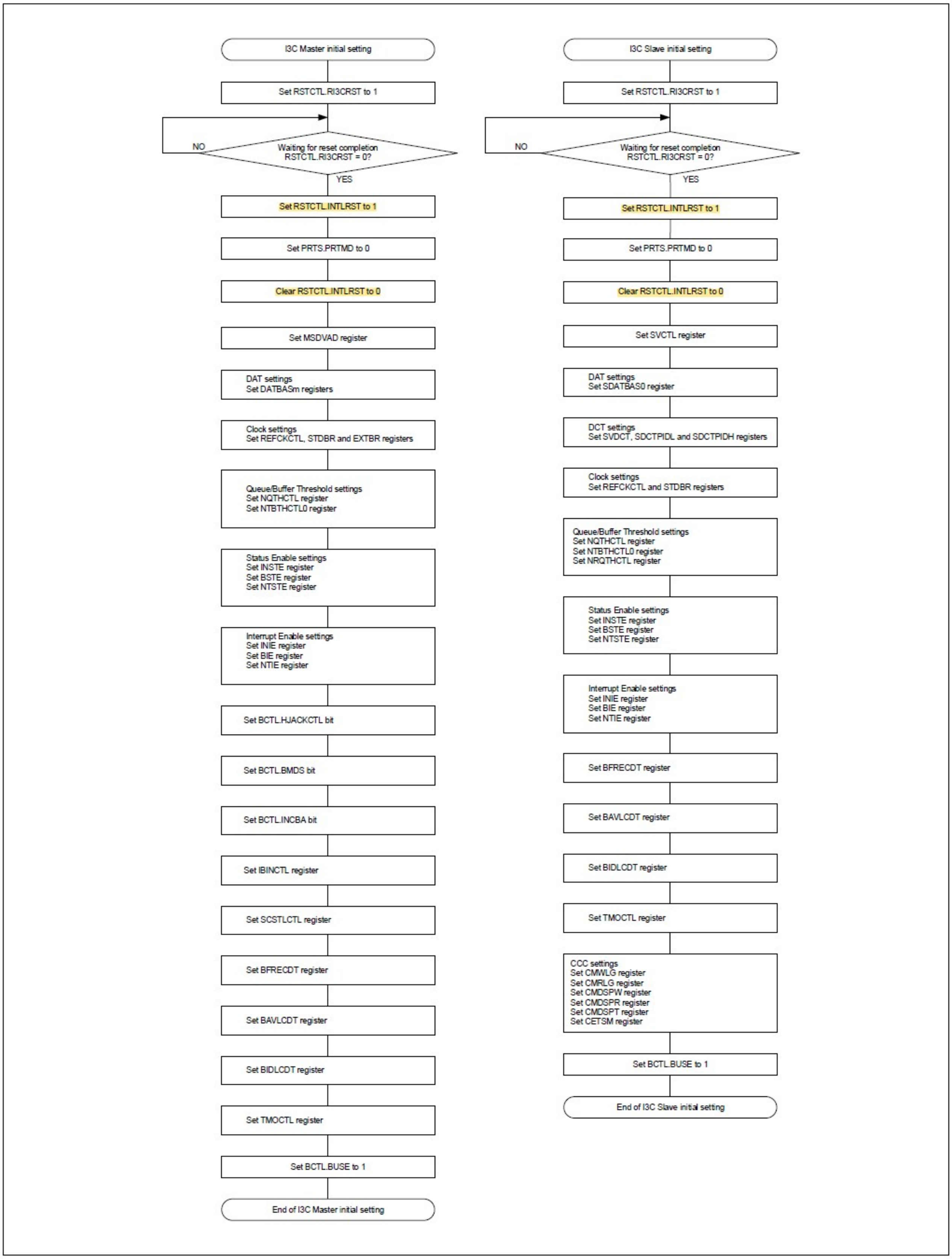


Figure 7.8-112 Example of I3C initialization flowchart

[Title]
Addition to xSPI AC specifications.

[Phenomenon]
There is a possibility of malfunction due to capturing the DS signal (intermediate voltage) immediately after the CS signal falls.

[User's manual Update]
Define the maximum value of t_{CSLDSL} as the corresponding specification. If this AC specification is not met and JESD251 Profile 1.0 memory or Profile 2.0 memory is used, the internal pull-down of the IOBUFF must be enabled before starting the access.

User's Manual
10.1 Electrical Characteristics
10.1.5.12 xSPI Timing

[From]

Table 10.1-16 xSPI Timing (2/2)

Parameter	Symbol	1.8V		3.3V		Unit	Figure
		Min.	Max.	Min.	Max.		
DS low to CS high	t_{DSLCSH}	6.0 / 8.0*1	—	10.6	—	ns	**Figure 10.1-24**
CS high to DS Tri-state	t_{CSHDST}	0.0	t_{PERIOD}	0.0	t_{PERIOD}	ns	
CS low to DS low	t_{CSLDSL}	0.0	—	0.0	—	ns	
DS Tri-state to CS low	t_{DSTCSL}	0.0	—	0.0	—	ns	

Note: CK: XSPI0_CKP (XSPI0_CKN)
DS: XSPI0_DS
CS: XSPI0_CS0N, XSPI0_CS1N

Note 1. Specification at 133 MHz / Specification at 100 MHz

Note 2. These are values when the OEN assertion is extended in the Output Enable Asserting extension bit (COMCFG.OEASTEX = 1b).

Note 3. These are the values when the CS assertion is extended in the CS asserting extension bit (LIOCFGCSn.CSASTEX = 1b).

Note 4. The standard value for xSPI266 is 0.8 ns.

[To]

Table 10.1-17 xSPI Timing (2/2)

Parameter	Symbol	1.8V		3.3V		Unit	Figure
		Min.	Max.	Min.	Max.		
DS low to CS high	t_{DSLCSH}	6.0 / 8.0*1	—	10.6	—	ns	**Figure 10.1-26**
CS high to DS Tri-state	t_{CSHDST}	0.0	t_{PERIOD}	0.0	t_{PERIOD}	ns	
CS low to DS low*5	t_{CSLDSL}	0.0	12.5*6	0.0	17.4*6	ns	
DS Tri-state to CS low	t_{DSTCSL}	0.0	—	0.0	—	ns	

Note: CK: XSPI0_CKP (XSPI0_CKN)
DS: XSPI0_DS
CS: XSPI0_CS0N, XSPI0_CS1N

Note 1. Specification at 133 MHz / Specification at 100 MHz

Note 2. These are values when the OEN assertion is extended in the Output Enable Asserting extension bit (COMCFG.OEASTEX = 1b).

Note 3. These are the values when the CS assertion is extended in the CS asserting extension bit (LIOCFGCSn.CSASTEX = 1b).

Note 4. The standard value for xSPI266 is 0.8 ns.

Note 5. If the DS is high during the command & modifier phase when using JESD251 Profile 2.0 memory, the time from CS low to DS high must also meet this specification.

Note 6. When using JESD251 Profile 1.0 memory or JESD251 Profile 2.0 memory with LIOCFGCSn.LATEMD set to 0, this constraint does not apply if the internal pull-down resistor of the DS pin is enabled.