DSPI: relationship between protocol and register interface clocks

Introduction

The main purpose of this technical note is to highlight the relationship between the DSPI_CLK and the PBRIDGE_CLK in the microcontrollers of the SPC58 family, describing the impact in case of missing implementation of right ratio between these two clocks.

1 DSPI clock overview

The microcontrollers of the SPC58 family support more instances of the deserial serial peripheral interface (DSPI) module.

Note: for the chip specific implementation details of the instances of this module, refer to the device configuration chapter of the device's reference manual.

Each DSPI module provides a synchronous serial bus for communication between an MCU and an external peripheral device, supporting an asynchronous clocking scheme for register and protocol interfaces.

Based on the top level clock generation architecture of SPC58 microcontrollers, the clock generation module (MC_CGM) allows developers to select and to divide each clock source.

The register interface is driven by PBRIDGE_CLK while the protocol interface is derived by DSPI_CLKn, n = 0,1.

- PBRIDGE_CLK frequencies that can be selected by the clock sources are:
  - External oscillator/crystal (XOSC)
  - Internal 16 MHz RC oscillator (IRCOSC)
  - PLL0 (PLL0_PHI)
  - PLL1 (PLL1_PHI)

and it is scaled by the divider MC_CGM_SC_DC2 with the condition to be equal at ¼ SYS_CLK.

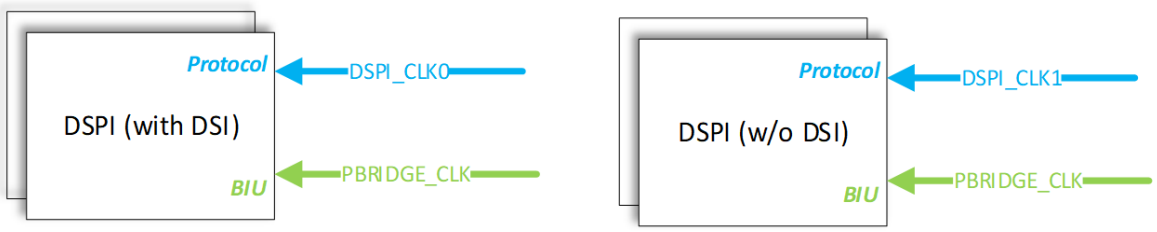
- DSPI_CLKn frequencies that can be selected by the clock sources are:
  - External oscillator/crystal (XOSC)
  - Internal 16 MHz RC oscillator (IRCOSC)
  - PLL0 (PLL0_PHI)

and it is scaled by the divider MC_CGM_AC12_DCn (n = 0,1) limitated to the "**maximum auxiliary level clock frequencies**" (refer to the device's reference manual).

Note: DSPI0_CLK0 only drives DSPI instances supporting DSI feature, while DSPI_CLK1 only drives DSPI instances without DSI feature, as shown in the Figure 1. DSPI clocks.

Note: SPC58 2B line does not support DSI feature and its clock tree implements only one DSPI_CLK (MC_CGM_AC12_DC0).

Figure 1. DSPI clocks



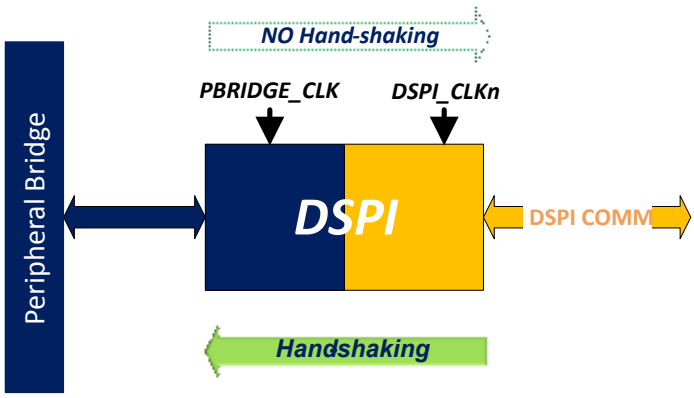
1.1 DSPI clock domains

DSPI module has two clock domains:

- **PBRIDGE_CLK**: bus and register clock, connected with peripheral bridge
- **DSPI_CLKn**: IP clock, used for communication

The DSPI_CLK frequency is always greater than or equal to 1.25 times PBRIDGE_CLK. The ratio between the two clock domains is required to guarantee the rightness of the internal finite state (FSM) of the peripheral. The violation of this ratio can give an unpredictable state, corrupting the communication protocol.

Figure 2. DSPI clock domains



1.1.1 DSPI hand-shaking

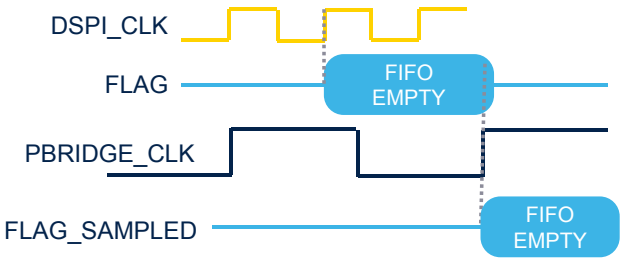
The DSPI only supports ***hand-shaking*** of flags (eg: FIFO-EMPTY, SPEF_CLEAR, DDIF_CLEAR and DPEF_CLEAR, etc…) in the direction from DSPI_CLK domain to PBRIDGE_CLK domain.

*Note: FIFO-EMPTY is never lost and is the best example of **hand-shaking**.*

This means that the flags from the DSPI_CLK domain (**source**) are cleared, once the PBRIDGE_CLK domain (**destination**) samples them.

Note: in this direction there is no limitation between clocks frequencies.

Figure 3. DSPI_CLK(src) to PBRIDGE_CLK(dst) - PBRIDGE_CLK < DSPI_CLK



As shown in Figure 3. DSPI_CLK(src) to PBRIDGE_CLK(dst) - PBRIDGE_CLK < DSPI_CLK, even supposing that PBRIDGE_CLK frequency is slower than DSPI_CLK, the flag is never lost.

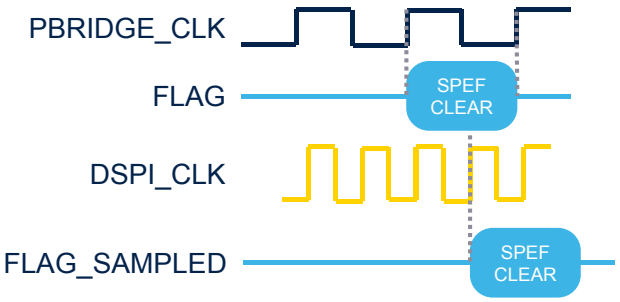
Due to hand-shaking the clearing of the flag, set in the DSPI_CLK domain, is always done after that the flag has been sampled in the PBRIDGE_CLK domain.

However, in the opposite direction, from PBRIDGE _CLK domain to DSPI_CLK domain, the duration of flags is one clock cycle of PBRIDGE_CLK.

This means that the flag is always sampled only if the DSPI_CLK domain (**destination**) is fast enough with respect to the PBRIDGE_CLK domain (**source**).

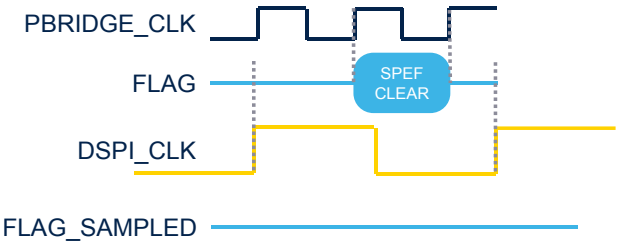
Note: Ratio 1.25 is needed to grant it across all corners, considering jitter, local variations.

Figure 4. PBRIDGE_CLK(src) to DSPI_CLK(dst) - DSPI_CLK > PBRIDGE_CLK



The Figure 4. PBRIDGE_CLK(src) to DSPI_CLK(dst) - DSPI_CLK > PBRIDGE_CLK shows the case where **DSPI_CLK > PBRIDGE_CLK** (including the 25% margin) so that the flag is always sampled.

Figure 5. PBRIDGE_CLK(src) to DSPI_CLK(dst) - DSPI_CLK < PBRIDGE_CLK



As shown in Figure 5. PBRIDGE_CLK(src) to DSPI_CLK(dst) - DSPI_CLK < PBRIDGE_CLK, supposing that **DSPI_CLK < PBRIDGE_CLK**, DSPI_CLK domain (**destination**) may miss the flag.

2 Effects of violation - ratio of DSPI clock freq and PBRIDGE clock freq < 1.25 times

In case of violation of ratio between clocks the DSPI behaviour could be unpredictable and the failure shall depend on the application but violating the errata, the internal state machine (FSM) of the peripheral could be corrupted.

Note: the FSM can stuck without trigger the eDMA transfer as well.

As countermeasure, specific safety mechanisms (SMs) can reduce the risk of each violation.

Note: the reset and reinitialization of the DSPI peripheral is the preferred way to restore from such kind of violations.

The three use cases, described in the following paragraphs, totally cover the violation impact inside the peripheral:

- Serialization and de-serialization (DSI module) communication is impacted as well as CSI mode. In the status register the DSI data received with active bits could be not valid
- SPI parity error is impacted
- DSI parity error is impacted

2.1 dDDIF CLEAR issue

Assuming that software application implements a DSI transmission.

The DSPI module is configured to stop DSI frame transmissions when DSI data with active bits has been received, that is:

```
DSPI_DSICR0.DMS = 0x1 /* Data Match Stop*/
```

This event is reported by the DSPI_SR[DDIF] flag in the status register by setting this bit to 0x1:

```
DSPI_SR.DDIF == 0x1
```

The SW enables at this event the request of interrupt:

```
DSPI_RSER.DDIF_RE = 0x1 /* DSI data received with active bits Request Enable */
DSPI_RSER.DDIF_DIRS = 0x0 /* Select interrupt request if DDIF flag is set */
```

or the DMA request:

```
DSPI_RSER.DDIF_RE = 0x1 /* DSI data received with active bits Request Enable */
DSPI_RSER.DDIF_DIRS = 0x1 /* Select DMA request if DDIF flag is set */
```

To resume the SPI transmission it is necessary that SW clears the DSPI_SR[DDIF] flag (**w1c**), or waits that the DMA generates the acknowledgement.

Failure impact

If the clearing of DDIF flag (done in PBRIDGE_CLK domain) is not rightly sampled (in the DSPI_CLK domain), the pulse is missed and the transmission may not resume.

Issue detection

SW can check if the transmission is resuming by clearing the flag and the data transfer is happening, if data transmission is not happening, we have pulse missed issue.

For DMA case we can see the liveness of DMA transmission and detect the pulse miss issue.

2.1.1 SPEF CLEAR issue

Assuming that SW implements a SPI transmission.

The DSPI module is configured to stop SPI operation if a parity error occurred in the received SPI frame:

```
DSPI_MCR.PES = 0x1 /* Parity Error Stop in a received SPI frame.*/
```

This event is reported by the DSPI_SR[SPEF] flag in the status register by setting this bit to 0x1:

```
DSPI_SR.SPEF == 0x1 /* Parity error has occurred.*/
```

To resume the SPI transmission the SW shall clear the DSPI_SR[SPEF] flag (**w1c**).

Failure impact

If the clearing of the SPEF flag (done in the PBRIDGE_CLK domain) is not rightly sampled (in the DSPI_CLK domain), the pulse is missed and the transmission may not resume.

Issue detection

SW can check if transmission is resuming by clearing the SPEF flag and the data transfer is happening, if data transmission is not happening, we have pulse missed issue.

2.1.2 DPEF CLEAR issue

Assuming that SW implements a DSI transmission.

The DSPI module is configured to stop SPI operation if a parity error occurred in the received DSI frame:

```
DSPI_DSICR0.PES = 0x1 /* Parity Error Stop in a received DSI frame. */
```

This event is reported by the DSPI_SR[DPEF] flag in the status register by setting this bit to 0x1:

```
DSPI_SR.DPEF == 0x1 /* Parity error has occurred.*/
```

To resume the DSI transmission the SW shall clear the DSPI_SR[DPEF] flag (w1c).

Failure impact

If the clearing of the DPEF flag (done in the PBRIDGE_CLK domain) is not rightly sampled (in the DSPI_CLK domain), the pulse is missed and the transmission may not resume.

Issue detection

SW can check if transmission is resuming by clearing the DPEF flag and the data transfer is happening, if data transmission is not happening, we have pulse missed issue.

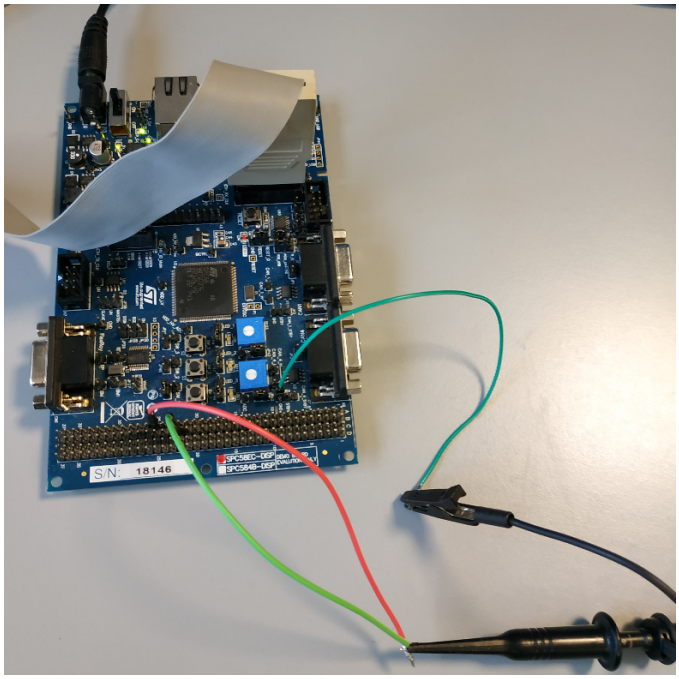
2.2 DDIF clear example

In this example the DSPI 4 is configured in DSI mode.

The SPC58EC-DISP board has been used for testing and the SOUT (PF[0]) is wired to the SIN (PF[1]) and to the Lecroy oscilloscope 610Zi.

Note: related DSPI signals are programmed in the SIUL2 as documented in the device's reference manual.

Figure 6. Board setup

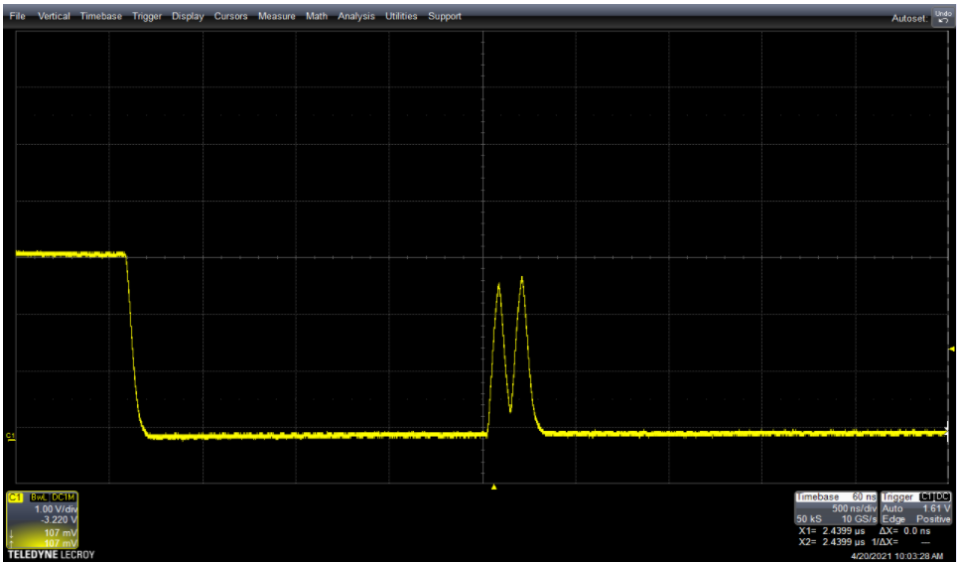


The ASDR0 is only used by application software to write the 32 LSB of the data to be serialized.

```
DSPI_4.CTAR0.R = 0xF9020020;
DSPI_4.CTAR1.R = 0xF9020080;
DSPI_4.DSICR0.R = 0x00100001;
DSPI_4.DSICR1.R = 0x1E010001;
DSPI_4.MCR.R = 0xD0011000;
DSPI_4.TCR.B.SPI_TCNT = 0x00000000;
DSPI_4.MCR.B.MDIS= 0;
/* Module : Enable*/
DSPI_4.DSICR0.B.TXSS = 1;
DSPI_4.DSICR0.B.DCONT = 1;
DSPI_4.DSICR0.B.FMSZ4 = 1;
DSPI_4.DSICR0.B.TSBC = 0;
/* data match enabled */
DSPI_4.DSICR0.B.DMS = 1;
/* Enable at this event the request of interrupt (no DMA) */
DSPI_4.RSER.B.DDIF_RE = 1;
DSPI_4.DPIR0.R = MATCH_USER_VALUE;
DSPI_4.DIMR0.R = MATCH_USER_VALUE;
/* Transmit the data */
DSPI_4.ASDR0.R = USER_VALUE;
```

By ISR or in polling mode, the software checks if the DSPI_4.SR.B.DDIF is set to 1 and it is cleared by writing it on matching. No eDMA is used in this test.

Figure 7. DSI SOUT output



The Figure 7. DSI SOUT output shows the DSI output from the SOUT pin, this reflects the value programmed in the DSPI_4.ASDR0.R register. In this case, the DSPI_CLK is 80 MHz while the PBDRIDGE_CLCK is 40 MHz.

When the match stop condition is found the DSPI_4.SR.B.DDIF is set to 1 by the hardware.

The software clears this status and it restarts the transmission in a loop and another match is expected in the next cycles.

When the clock relatioship is respected, after clearing the DSPI_4.SR.B.DDIF, the transmission restarts after writing the DSPI_4.ASDR0.R register and the oscilloscope dumps the output again.

In other clock settings where the relationship is violated, e.g. DSPI_CLK is 20 MHz and the PBDRIDGE_CLCK is 40 MHz, it has been verified that, after the first match stop condition, the DSPI_4.SR.B.DDIF is cleared by the software (so from 1 it becomes 0) but the transmission stops working and no expected waveform is measured by the oscilloscope although the application successfully updates the DSPI_4.ASDR0.R register.

After resetting the peripheral, from MC_RGM.PRST3.DSPI_4_RST, and reinit the DSPI the transmission restarts and the proper signal is measured by the oscilloscope.

Note: tests have been performed in normal temperature and voltage conditions and respecting the datasheet values for this MCU.

3 Conclusion

This document aims to clarify the issues that can be found when violating the relationship between the PBRIDGE and DSPI clocks.

The DSPI_CLK frequency shall always be greater than or equal to 1.25 times PBRIDGE_CLK.

The techincal note also shows some cases and a real example of the device behavior in case of the DSPI_CLK is lesser than the PBRIDGE_CLK and how the peripheral internal state machine stuck. Application should always take care about this limitation, in automotive context usually expected safety mechanisms are implemented to check if the peripheral is working fine under different conditions. Application cannot only rely on these mechanisms if the clocks relationship is completely violated.

Appendix A Other information

A.1 Reference documents

Table 1. **Reference documents**

Document name	ID	Title
RM0407	028117	SPC58 C Line - 32 bit Power Architecture automotive MCU Dual z4 cores 180 MHz, 4 MBytes Flash, HSM, ASIL-B
DS11620	029264	SPC58 C Line - 32 bit Power Architecture automotive MCU Dual z4 cores 180 MHz, 4 MBytes Flash, HSM, ASIL-B

A.2 Acronyms

Table 2. **Acronyms**

Acronyms	Meaning
CGM	Clock generator module
DSI	Deserial serial interface
RGM	Reset generation module
SIUL2	System integration unit lite 2

Revision history

Table 3. **Document revision history**

Date	Revision	Changes
10-Jun-2021	1	Initial release.

Contents

List of tables

List of figures