NXP Semiconductors
Application Note

Document Number: AN5320
Rev. 0, 08/2016

Using LPSPI on KL28Z

1. Introduction

LPSPI is a brand new module. In this application note, the key feature is introduced, and some key points in application is also discussed here. The topic includes both master and slave mode.

2. Key Features

The key features on LPSPI includes:

- FIFO of 4 words
- Support 1, 2, or 4 wires mode
- SOUT and SIN is configurable
- Direct frame size up to 512 bytes
- Work with DMA in low power mode to keep communication

3. LPSPI Master Operation

3.1. Set baud rate

For master operation, the formula to set baud rate is:

```
Baud rate = Function clock / (PRESCALE
*(SCKDIV+2))
```

When LPSPI_TCR[PRESCALE] and LPSPI_CCR[SCKDIVE] is set to be 0, we get the maximum baud rate as Function clock/2.

For example, for 8M function clock, when setting

Contents

1. Introduction 1
2. Key features 1
3. LPSPI master operation 1
   3.1. Set baud rate 1
   3.2. Frame length setting 2
   3.3. Using 4 wire mode 3
   3.4. Select CS 6
   3.5. Work in interrupt mode 6
   3.6. Work in DMA mode 7
   3.7. Work together with DMA in low power mode 7
4. LPSPI slave operation 7
   4.1. Frame length setting 7
   4.2. Baud rate 7
   4.3. Work in interrupt mode and DMA mode 8
   4.4. Work together with DMA in low power mode 8
5. Revision history 9

LPSPI_TCR[PRESCALE] and LPSPI_CCR[SCKDIVE] to be 0, we get the maximum baud rate as 4MHz.

3.2. Frame length setting

On LPSPI, the frame length can be 512 bytes directly by configure LPSPI_TCR[FRAMESZ]. If the frame length is more than 512 bytes, it can be expanded by setting continuing command. To make a frame more than the byte count specified by LPSPI_TCR[FRAMESZ], need to follow the steps below:

- Set LPSPI_TCR[CONT] and LPSPI_TCR[CONTC]
- Send the bytes specified by LPSPI_TCR[FRAMESZ] and repeat this step until all data are sent out
- Clear LPSPI_TCR[CONT] and LPSPI_TCR[CONTC]

To make things simple, here is an example of expanding a frame when frame size is set to 8-bit.

When the frame size is 8bit, and we write TDR four times, we get a waveform, as shown in this figure:

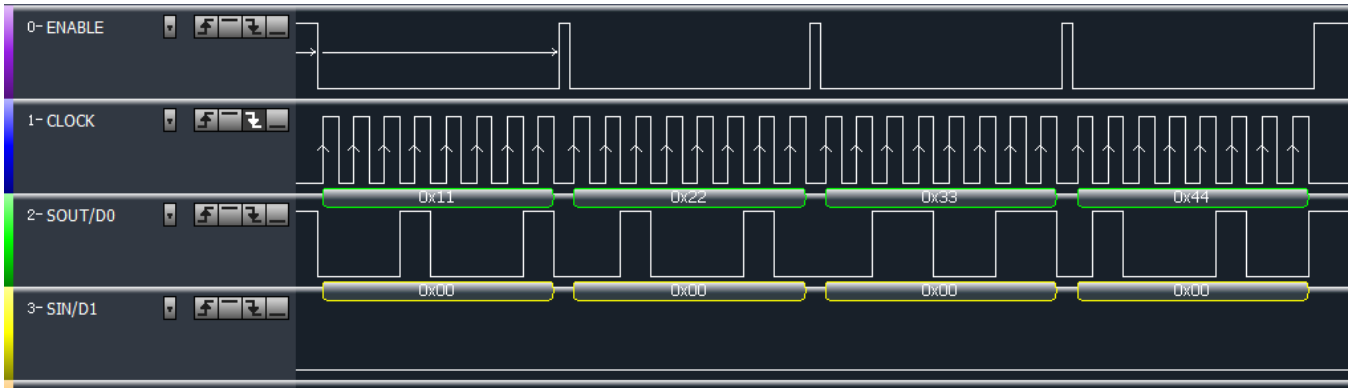

Figure 1. Non continuous mode

Here we can see, the CS is not continous, it is in fact four independent frames.

When the steps above is applied, we get a waveform like Figure 2 shows:

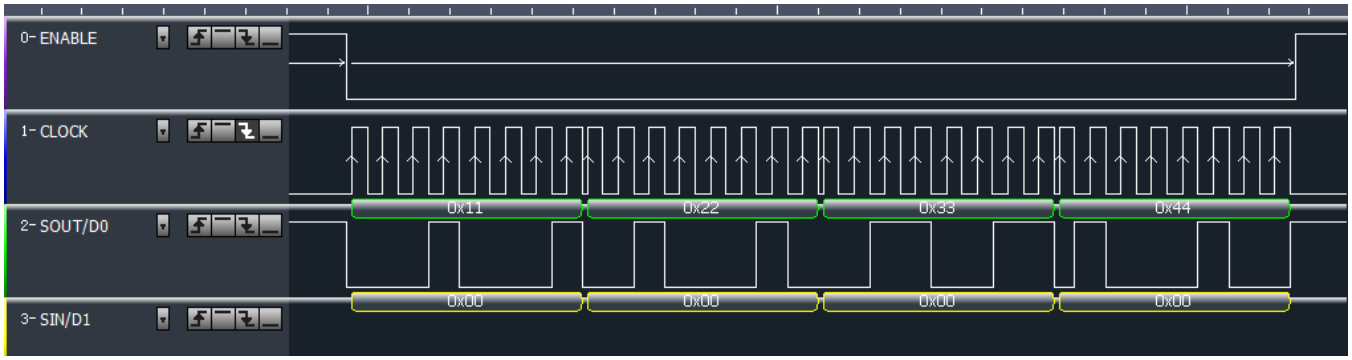

Figure 2. Continuous mode

Now, we can see, the CS is merged into one, all 4 bytes are in one frame.

During the transfer, clear CONTC would make a new frame.

Figure 3 shows insert a new frame after the first byte is sent out.

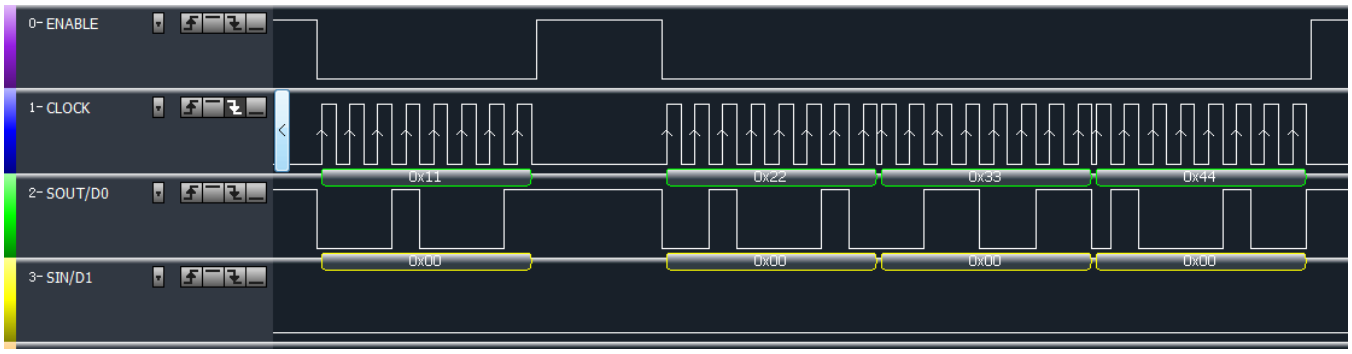

Figure 3. Insert a new frame during a continuous transfer

Write the LPSPI_TCR push the command into FIFO, so, we should not treat LPSPI_TCR as an ordinary register. Instead, we should view it as the entry to send control command.

3.3. Using 4 wire mode

LPSPI support 1 wire, 2 wire and 4 wire mode. This can be configured in LPSPI_TCR[WIDTH]. The signal assignment is as shown in Figure 4.

PCS[2] / DATA[2]	Peripheral Chip Select or data pin 2 during quad-data transfers. Input in slave mode, output in master mode, input in quad-data receive transfers, output in quad-data transmit transfers.	I/O
PCS[3] / DATA[3]	Peripheral Chip Select or data pin 3 during quad-data transfers. Input in slave mode, output in master mode, input in quad-data receive transfers, output in quad-data transmit transfers.	I/O
SOUT / DATA[0]	Serial Data Output. Can be configured as serial data input signal. Used as data pin 0 in quad-data and dual-data transfers.	I/O
SIN / DATA[1]	Serial Data Input. Can be configured as serial data output signal. Used as data pin 1 in quad-data and dual-data transfers.	I/O

Figure 4. Signal assignment for 2 wire/4wire mode

The typical application for 4-bit transfer is to connect a SPI flash which support multi-IO. Figure 4 shows the timing for 4 x I/O Read Mode Sequence on MX25R512F.

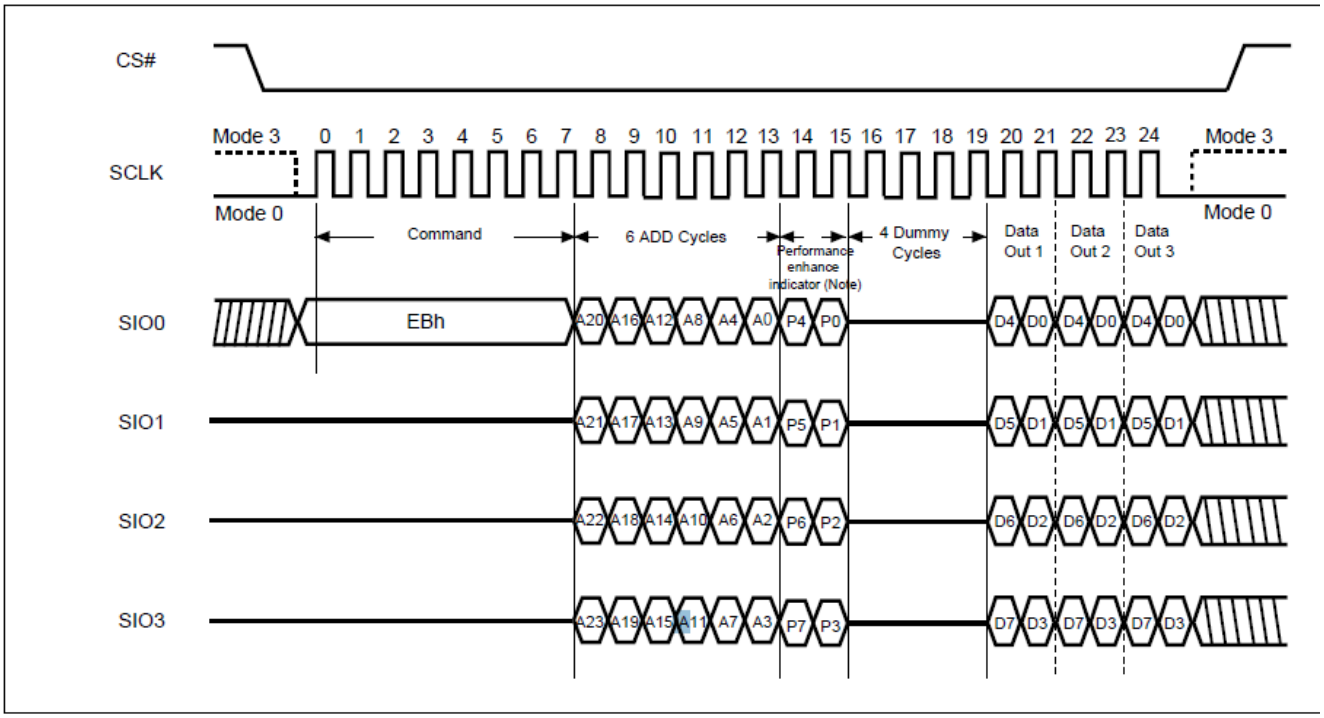

Figure 5. 4 x I/O Read Mode Sequence

From Figure 5 we can see, there are 3 kinds of 4-bit data operation:

- Sending
- Receive
- Dummy

3.3.1. 4-bit sending operation

Figure 6 shows the timing when LPSPI works in 4-bit mode and sending data out, for each byte (eight bits), 4 bits are shifted out on clock edge. After two clocks, one byte is transferred.

Figure 6. LPSPI work in 4-bit sending mode

To implement 4-bit sending operation, the following steps are necessary specially on KL28Z-TWR board:

- LPSPI_TCR_WIDTH should be set to be 2
- PTB3 should be set to LPSPI1_CS3
- PTE6 should be set to LPSPI1_CS2
- LPSPI_CFGR1[PCSCFG] should be set to 1 to disable PCS[3] and PCS[2], to make them to be D3 and D2
- LPSPI_TCR[RXMSK] should be set to 1 to disable receiving
- Write to LPSPI_TDR to begin a transfer for one byte

3.3.2. 4-bit receiving operation

To implement 4-bit receiving operation, in addition to the step 1 to step 4 in Section 3.3.1, “4-bit sending operation”, the following steps are necessary:

- Clear LPSPI_TCR[RXMSK] to enable receiving
- Set LPSPI_TCR[TXMSK] to begin a reading transfer
- Read LPSPI_RDR to get the data

3.3.3. 4-bit dummy operation

In addition to sending and receiving, dummy operation is also necessary. For 4-bit dummy operation, in addition to the step 1 to step 4 in section 3.3.1, the following steps are necessary:

- Set LPSPI_TCR[RXMSK] to disable receiving
- Set LPSPI_TCR[TXMSK] to begin a dummy transfer
- Dummy transfer finishes after LPSPI_TCR[TXMSK] goes back to 0

3.3.4. 4-bit dummy operation in continuous mode

If use dummy operation in continuous mode, need to pay attention, LPSPI would not stop after the frame length specified in TCR is sent out. So, for 4-bit dummy operation, continuous mode is not recommended. So, to implement 4-bit operation, need to work in non-continuous mode and implement CS by GPIO mode.

3.3.5. Tristate in 4-bit operation

As in 4-bit operation, if we try to read by LPSPI, the data line in fact is bi-directional. So we need to set LPSPI into tristate by setting LPSPI_CFGR1[OUTCFG], when working in 4-bit mode.

3.4. Select CS

For a LPSPI, we can select different chip select, this is implemented by setting LPSPI_TCR[PCS].

For detailed configuration, please refer to Figure 7.

25–24 PCS	Peripheral Chip Select Configures the peripheral chip select used for the transfer. This field is only updated between frames. 00 Transfer using LPSPI_PCS[0] 01 Transfer using LPSPI_PCS[1] 10 Transfer using LPSPI_PCS[2] 11 Transfer using LPSPI_PCS[3]

Figure 7. Signal assignment for 2 wire/4wire mode

3.5. Work in interrupt mode

To make LPSPI working in interrupt mode, need to follow the steps below:

- Initialize LPSPI
- Configure the LPSPI_IER[RDIE], LPSPI_IER[TDIE], other control bit in LPSPI_IER is optional depending on current application requirement
- Enable the IRQ in NVIC
- In ISR, read LPSPI_RDR when LPSPI_SR[RDF] is set
- In ISR, write LPSPI_TDR when LPSPI_SR[TDF] is set
- Disable LPSPI when a frame is transferred

3.6. Work in DMA mode

To make LPSPI work with DMA, the following steps are necessary:

- LPSPI_DER[TDDE] and LPSPI_DER[RDDE] should be set
- DMA MUX for TX should be set for LPSPI TX
- DMA MUX for RX should be set for LPSPI RX
- DMA_TCD_CSR[DREQ] should be set to disable DMA after transfer

3.7. Work together with DMA in low power mode

Another key feature supported on LPSPI is, that it can work with DMA in lower mode. Usually VLPS mode is used here. By this way, we can keep SPI communication and go into low power mode to reduce power consumption.

To implement this application, the following steps are necessary:

- Initialize LPSPI with LPSPI_CR[DOZEN] enabled
- Initialize DMA with DMA_EARS set for the channel used to enable asynchronous DMA request
- Enable DMA
- Go to VLPS mode
- Wake up after LPSPI transfer ends by DMA interrupt

4. LPSPI Slave Operation

4.1. Frame length setting

When frame size is less or equal to 32, LPSPI_SR[RDF] is set when the bits specified is received. For example, for receiving:

- When frame size is 8, LPSPI_SR[RDF] is set when each byte is received. So, in this configuration, when 32 bit is transferred, LPSPI_SR[RDF] is set four times, LPSPI[RDR] still need to be read four times.
- When frame size is 32, LPSPI_SR[RDF] is not set when the first 8 bit arrives. After 32 bit transfer finishes, LPSPI_SR[RDF] is set only once, and LPSPI[RDR] is also read only once.

For transmit, it is the same.

4.2. Baud rate

Baud rate is only configurable for master mode. For slave, the baud rate configuration is not necessary, and the maximum baud rate supported for slave is function clock/2.

4.3. Work in interrupt mode and DMA mode

These steps are just like what is in master mode, please refer to section 3.5 and 3.6 for more information.

4.4. Work together with DMA in low power mode

Just like the LPSPI master, LPSPI slave can also work with DMA and keep in low power mode to save power consumption.

To implement this application, the steps are:

- Initialize LPSPI with LPSPI_CR[DOZEN] enabled
- Initialize DMA with DMA_EARS set for the channel used
- Enable DMA
- Go to VLPS mode
- Wake up when get a frame
- Enable DMA again in the interrupt if necessary

For tips to optimize the current on KL28Z-TWR board, please refer to the discussion in AN5301.

Table 1 and Figure 8 shows the comparison for LPSPI slave working in interrupt mode and DMA mode, in the test VLPS is applied when CPU is not working. In DMA mode, it keeps communication by DMA and get an interrupt after a frame is received. It shows that by using DMA + LPSPI, it improves the current by up to 44.6%. In this test, SIRC (8M Hz) is selected as the clock source.

Table 1. Improvement made by using DMA mode in VLPS

Bytes in one frame	Mode	Current(mA)	Improvement
8	INT	2.73	-
8	DMA	1.98	27.5%
128	INT	3.27	-
128	DMA	1.81	44.6%

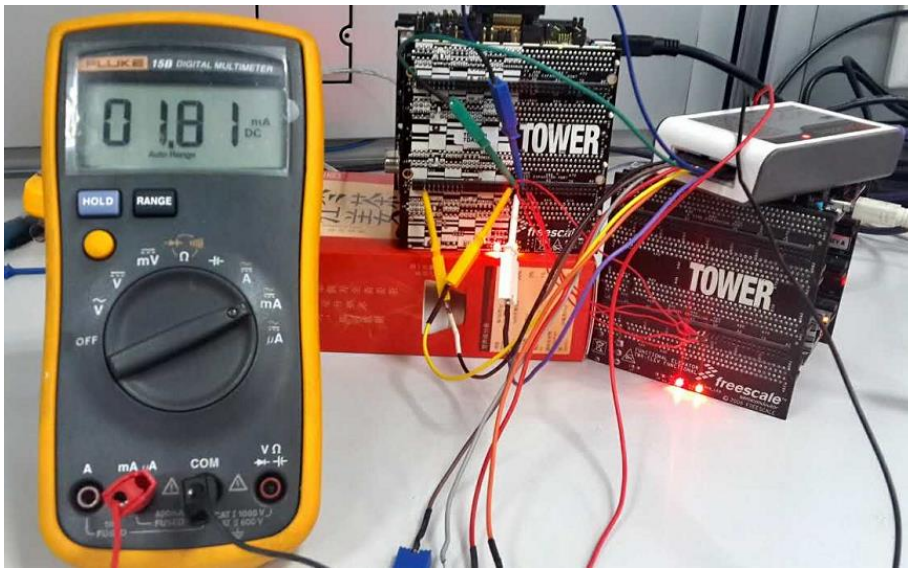

Figure 8. Improvement made by using DMA mode in VLPS

5. Revision History

Table 2. Revision history

Revision number	Date	Substantive changes
1.0	06/2016	Initial release

How to Reach Us:

Home Page:
nxp.com

Web Support:
nxp.com/support

Information in this document is provided solely to enable system and software implementers to use NXP products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document. NXP reserves the right to make changes without further notice to any products herein.

NXP makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does NXP assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters that may be provided in NXP data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including “typicals,” must be validated for each customer application by customer’s technical experts. NXP does not convey any license under its patent rights nor the rights of others. NXP sells products pursuant to standard terms and conditions of sale, which can be found at the following address: nxp.com/SalesTermsandConditions.

Document Number: AN5320
Rev. 0
08/2016