NX1 使用 XIP 和读取 FIFO 的注意事项

内容： NX1_OTP 系列 IC 同时执行 XIP 程序与外围功能 FIFO 读取时，FIFO Data 会发生数据错误。

原因： NX1 通过 XIP(e**X**ecute **I**n **P**lace)可以直接执行 SPI Flash 里面的程序，而 XIP 程序内读取 FIFO 或 XIP 程序过程读取 FIFO(例如透过 ISR)，如以下外围功能由于同时占用 FIFO 而发生 FIFO Data 读取错误的情况。

- ADC FIFO
- UART FIFO
- SPI0 FIFO
- SPI1 FIFO

方法： 相关 FIFO Data 的错误读取可透过读入其他寄存器进行回避。以下提供相关 FIFO 寄存器的说明，并搭配实际范例以说明正确读取 FIFO Data 的方法，范例程序请勿透过 XIP 执行。

- ADC FIFO
  - **P_ADC_Data_CH0 (ADC Channel 0 Register)**

Register	Offset		Description	Initial Value
P_ADC_Data_CH0	ADC_BA+0x10		ADC Channel 0 Register	0x0000_0000
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:12]	Reserved			0x0
Bit[15:4]	CH0_DAT	R	ADC's Channel 0 Register	0x0
Bit[3:0]	Reserved			0x0

```
#include "nx1_lib.h"
#include "nx1_adc.h"
U16 getAdcFifo(void) {
    //Step.1 Read register, you can use it or discard.
    //       Use P_ADC->Ctrl for example.
    (void)P_ADC->Ctrl;
    //Step.2 Read ADC FIFO.
    return P_ADC->Data_CH0;
}
```

- UART FIFO
  - **P_UART_Data (UART Data Register)**

Register	Offset		Description	Initial Value
P_UART_Data	UART_BA+0x00		UART Data Register	0x0000_0000
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:0]	Reserved			
Bit[7:0]	UDR	R/W	write to TXD FIFO in WR (THR) read from RXD FIFO in RD (RBR)	0x0

```
#include "nx1_lib.h"
#include "nx1_uart.h"
U8 getUartFifo(void) {
    //Step.1 Read register, you can use it or discard.
    //       Use P_UART->Ctrl for example.
    (void)P_UART->Ctrl;
    //Step.2 Read UART FIFO.
    return P_UART->Data;
}
```

- SPI0 FIFO

  - **P_SPI0_Data (SPI0 Data Register)**

Register	Offset		Description	Initial Value
P_SPI0_Data	SPI_BA+0x10		SPI0 Data Register	0x0000_0000
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:0]	FIFO_D	R/W	Write Data into FIFO Read Data from FIFO	0x0

```
#include "nx1_lib.h"
#include "nx1_spi.h"
U32 getSpi0Fifo(void) {
    //Step.1 Read register, you can use it or discard.
    //       Use P_SPI0->Ctrl for example.
    (void)P_SPI0->Ctrl;
    //Step.2 Read SPI0 FIFO.
    return P_SPI0->Data;
}
```

- SPI1 FIFO

  - **P_SPI1_Data (SPI1 Data Register)**

Register	Offset		Description	Initial Value
P_SPI1_Data	SPI_BA+0x50		SPI1 Data Register	0x0000_0000
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:0]	FIFO_D	R/W	Write Data into FIFO Read Data from FIFO	0x0

```
#include "nx1_lib.h"
#include "nx1_spi.h"
U32 getSpi1Fifo(void) {
    //Step.1 Read register, you can use it or discard.
    //       Use P_SPI1->Ctrl for example.
    (void)P_SPI1->Ctrl;
    //Step.2 Read SPI1 FIFO.
    return P_SPI1->Data;
}
```