SBI - Sandbox input from FPGA

SD022 | Posted on April 2, 2021 | Updated on June 30, 2025

Benoît STEINMANN
Software Team Leader
imperix

Table of Contents

The Sandbox Input from FPGA (SBI) block reads the value of the SBI registers in the FPGA. It is used to transfer data from user-made code within the FPGA to the CPU. To transfer data from the CPU to the user-made code within the FPGA, the SBO block should be used.

Information on FPGA edition is available on:

- Editing the FPGA firmware (sandbox) (PN116)

Usage examples of the SBI block are available on:

- FPGA-based hysteresis current control (TN120)
- FPGA-based SPI communication IP for A/D converter (TN130)

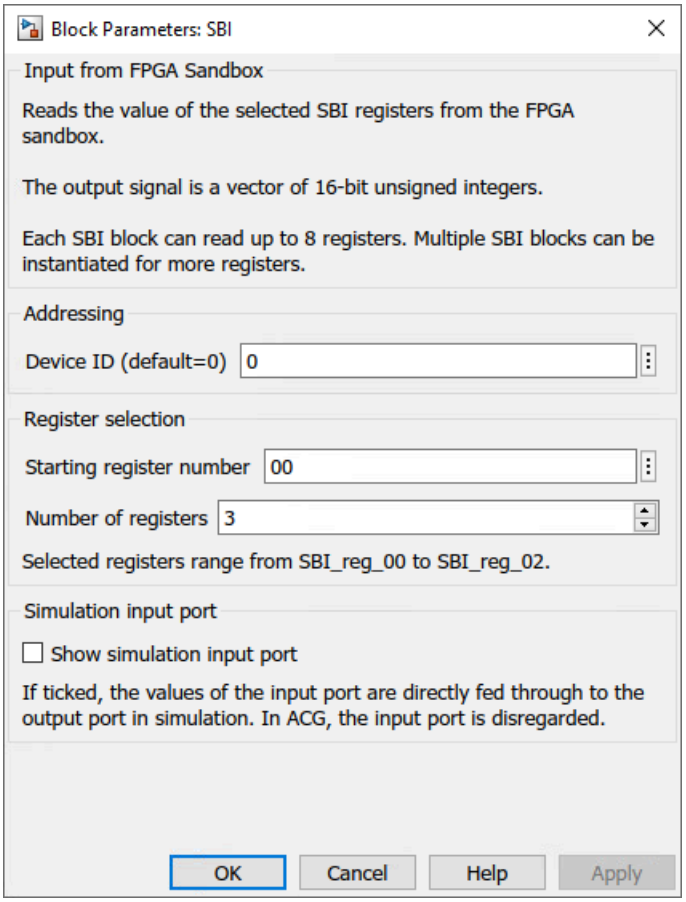
Simulink block

Signal specification

The output returns a vector of the 16-bit unsigned integers representing the SBI register values. Up to 8 registers can be read from a single SBI block. Multiple SBI blocks can be used to read more registers.

Parameters

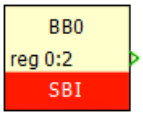
- `Device ID` selects which B-Box/B-Board to address when used in a multi-device configuration.
- `Starting register number` and `Number of registers` defines the range of registers to read.



PLECS block

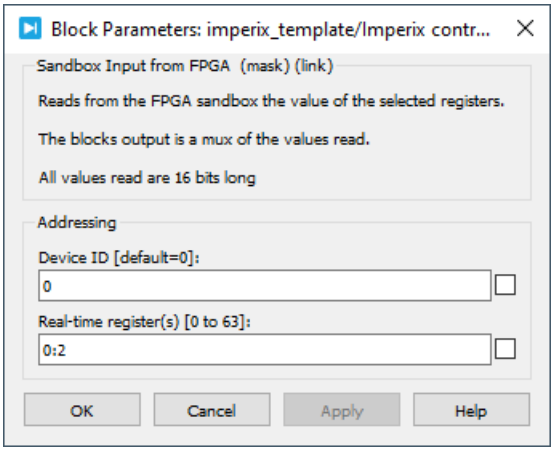
Signal specification

The output returns a vector of the 16-bit unsigned integers representing the SBI register values. Up to 8 registers can be read from a single SBI block. Multiple SBI blocks can be used to read more registers.



Parameters

- `Device ID` selects which B-Box/B-Board to address when used in a multi-device configuration.
- `Starting register number` and `Number of registers` defines the range of registers to read.



C++ functions

`Sbi_ReadDirectly` **— Read during init phase**

```
uint16_t Sbi_ReadDirectly(unsigned int address, unsigned int device=0);Code language: C++ (cpp)
```

Returns the SBI register value.

It can only be called in `UserInit()`.

Parameters

- `address` : address of the targeted register (0 to 63)
- `device` : the id of the addressed device (optional, used in multi-device configuration only)

`Sbi_Read` **— Read during run-time**

```
uint16_t Sbi_Read(unsigned int address, unsigned int device=0);Code language: C++ (cpp)
```

Returns the SBI register value, has to be called in the interrupt.

For this function to work the addressed register must be set as *real-time* using `Sbi_ConfigureAsRealTime()`, otherwise it returns 0.

Parameters

- `address` : address of the targeted register (0 to 63)
- `device` : the id of the addressed device (optional, used in multi-device configuration only)

`Sbi_ConfigureAsRealTime` **— Configure as readable during run-time**

```
void Sbi_ConfigureAsRealTime(unsigned int address, unsigned int device=0);Code language: C++ (cpp)
```

Tags an SBI register as *real-time*, meaning that its value is transferred before each interrupts and can then be retrieved using `Sbi_Read()`.

It has to be called in `UserInit()`.

Parameters

- `address` : address of the targeted register (0 to 63)
- `device` : the id of the addressed device (optional, used in multi-device configuration only)