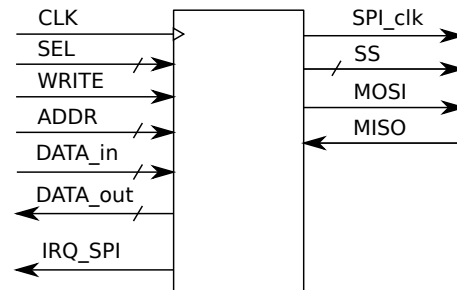


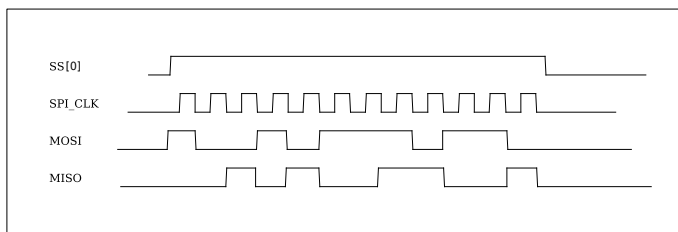
Dossmatik GmbH
Karlstraße 41
04420 Markranstädt
info@dossmatik.de

<http://www.dossmatik.de>

Blockdiagramm



Timingdiagramm



Inhaltsverzeichnis

Einbindung
Registerbeschreibung
C-Codebeispiel
Lieferumfang

Features

- einstellbare Bitlänge
- einfaches paralleles Interface
- 1-32 bit Schiebelänge
- einstellbare Frequenz
- Interruptausgang
- Statusregister
- independent VHDL

Anwendungsbeispiele

- auslesen von externen Sensoren
- Verbindung zu ADC-ICs
- Ansteuern von Schieberegistern

Version: V1.01
Datum: 24.5.2013

1 Einbindung

Die Komponente besteht aus zwei VHDL-Dateien. Die Datei SPI-8bitbus.vhd ist die Busanbindung und kann durch Codeänderungen auf andere kundenspezifische Busse einfach geändert werden. SPI-Master.vhd ist Statemaschine für die SPI Verarbeitung und generiert das Timing für die externen Signale.

```
entity bus8_spi_master is
  generic(
    slavenumber   : integer                := 1;  --max. 8 slaves
    clk_freq      : integer;
    HSEL          : std_logic_vector (4 downto 0) := "00000";
    preset_speed  : std_logic_vector (7 downto 0) := "00000100";  --4
    preset_shiftwidth : std_logic_vector (7 downto 0) := "00001000");--8

  port(
    Clk      : in  std_logic;
    SEL      : in  std_logic_vector (4 downto 0);
    Reset    : in  std_logic;
    WR       : in  std_logic;
    ADDR     : in  std_logic_vector (5 downto 0);
    DATA_in : in  std_logic_vector (7 downto 0);
    DATA_out : out std_logic_vector (7 downto 0);

    -- possibile interrupt source
    spi_irq : out std_logic;

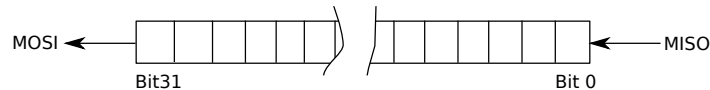
    --Hardwarepins SPI
    SPI_clk : out std_logic;  --SPI Clock
    SS      : out std_logic_vector(slavenumber-1 downto 0);  --SS slave select
    miso    : in  std_logic;  --master in slave out
    mosi    : out std_logic   --master out slave in
  );
end;  --bus8_spi_master
```

generic	Beschreibung
slavenumber	Anzahl der Slaveleitungen, bestimmt die Breite des Vektors für SS
clk_freq	hier muss die Frequenz ohne Einheit angegeben werden Bsp. 50E6 für 50Mhz
HSEL	Selectmuster auf welches reagiert wird
preset_speed	vordefinierter Wert für Register SPI-Speed
preset_shiftwidth	vordefinierter Wert für Register Bitlänge
Signalname	Beschreibung
CLK	Bus Clock zur Datenübertragung, und Ursprung für den internen Takt
SEL	Select für mehrere Busteilnehmer wird mit HSEL aus generic verglichen
WR	Schreibsignal
ADDR	Adressleitung für die Auswahl der Register
DATA_in	Daten zum Schreiben in Slave
DATA_out	die gelesenen Daten aus dem Slave
SPI_irq	Interruptquelle, aktiv nach einer beendeten Übertragung, wird durch ein beliebigen Lesezyklus deaktiviert
SPI_clk	Clock am SPI-Bus
SS	Slave Select Leitung zur Auswahl verschiedener Slaves
MISO	Master in Slave out
MOSI	Master out Slave in

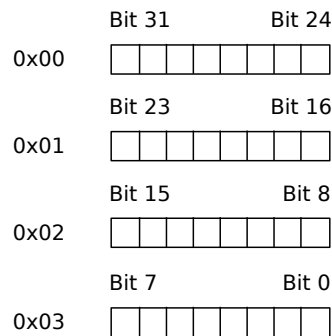
2 Registerbeschreibung

2.1 Sende-/Empfangsregister 0x00...0x03

In das Sende-/Empfangsregister wird das Bitmuster geschrieben, das gesendet werden soll. Nach der Übertragung steht das empfangene Bitmuster im Register. Das höchstwertige Bit ist auf dem Port MOSI und in das niederwertigste Bit wird der Wert von dem Port MISO geschrieben.



Das Register ist 32 bit breit und wird über die Adressen 0x00...0x03 angesprochen.



2.2 Register SPI-Speed 0x10

Die Busfrequenz muß höher als die SPI Frequenz sein, der generic clk_freq wird ein Frequenzteiler ermittelt, der die interne state-Maschine steuert. Hier kann der Frequenz nochmal geteilt werden, um niedrigere Frequenzen zu erhalten.

2.3 Register SPI-Status 0x11

Bit0 zeigt den Status an, ob aktuell eine Transmission erfolgt. Das Register kann nur gelesen werden und ist für pollende Statusabfragen gedacht.

2.4 Register SS (Slave Select) 0x12

Der SPI Master kann über mehrere Select-Leitungen mit mehreren Slaves kommunizieren. Vor der Übertragung wird die entsprechende Slave-Select Leitung auf High Signal aktiviert. Mit dem Schreiben in das Register SS wird die Übertragung aktiviert.

2.5 Register Bitlänge 0x13

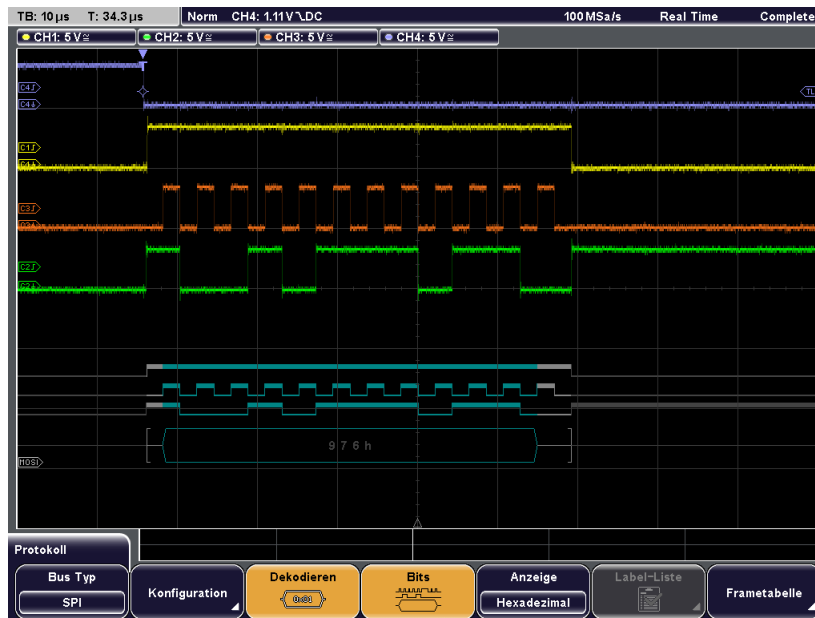
In dem Register steht die Schiebeweite. Mit jeder Aktivierung des Sendevorganges wird das Sende-/Empfangsregister um den Wert des Registers geschoben.

3 C-Codebeispiel

```

*(volatile char*) SPI_BASE      = 0x97; // Sendemuster
*(volatile char*) SPI_BASE+0x01 = 0x60; // Sendemuster
*((volatile char*) SPI_BASE+0x10 = 0x4; // SPI Speed
*((volatile char*) SPI_BASE+0x13 = 12; // Schiebelänge
*((volatile char*) SPI_BASE+0x12 = 3; // setze CS und starte

```



4 Lieferumfang

Der SPI-Master wird in VHDL ausgeliefert. Eine Testbench ermöglicht den einfachen Einstieg in die Komponente.