

Спецификация

на СФ-блок интерфейса SPI,
режимы «ведущий» и «ведовый»

СОДЕРЖАНИЕ:

1. Основные характеристики
2. Блок-диаграмма
3. Условное графическое обозначение
4. Описание входов/выходов
5. Карта регистров
6. Описание функционирования
7. Система тактирования
8. Верификация

ПРИЛОЖЕНИЯ:

1. Пример синтеза
2. Рекомендации по применению

1. Основные характеристики

- максимальная частота передачи/приема по интерфейсу SPI — $F_{sys}/4$;
- настраиваемая логика передачи/приема (полярность CS, LSB и MSB для MOSI);
- системный интерфейс AXI4-lite, 32p;
- отдельные независимые схемы ведущего и ведомого интерфейсов;
- формирование прерываний по событиям ведущего и ведомого интерфейсов;
- 4-х и 3-х проводной ведущий интерфейс.

2. Блок-диаграмма

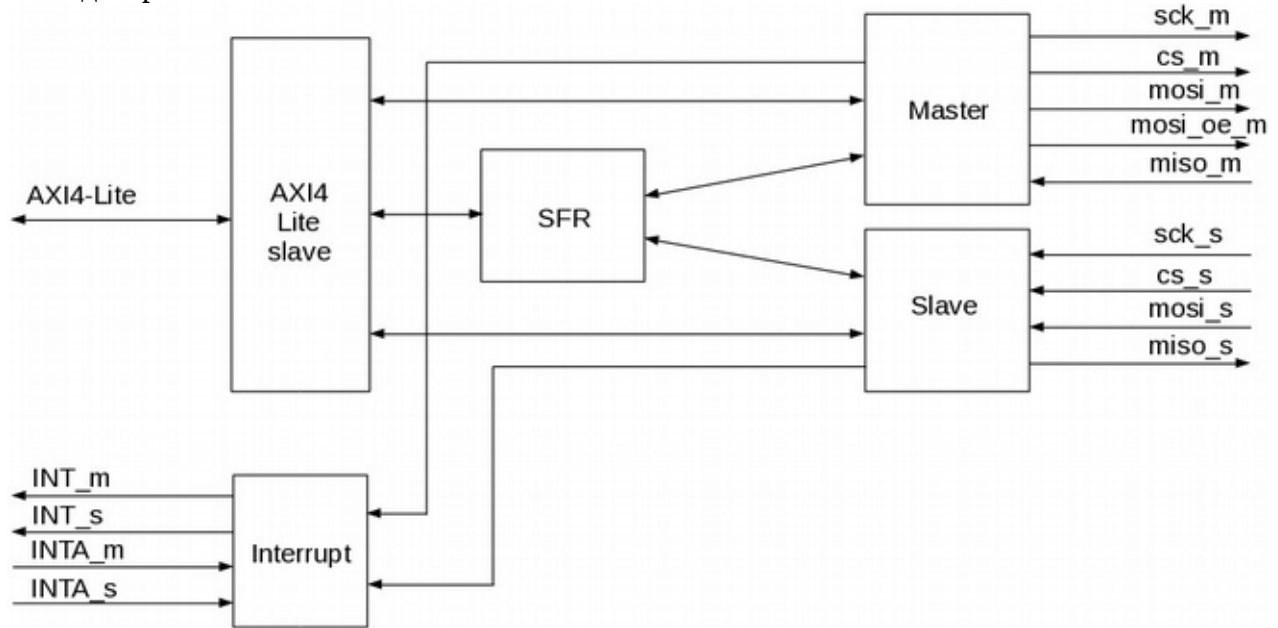


Рисунок 1 — Блок-диаграмма СФ-блока контроллера SPI

AXI4-Lite slave — ведомый контроллер системного интерфейса AXI4-Lite;

SFR — (Special Function Registers) регистры специального назначения;

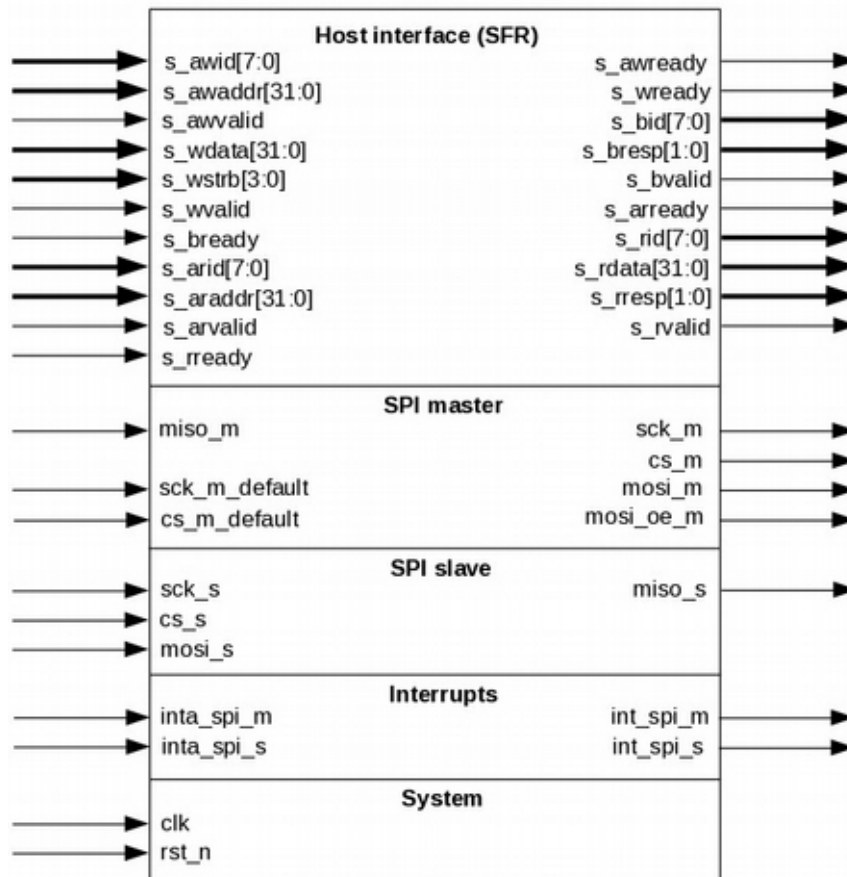
Interrupts — блок формирования сигналов системных прерываний;

Master — блок формирования сигналов ведущего SPI интерфейса;

Slave — блок формирования сигналов ведомого SPI интерфейса.

Описание всех входных и выходных сигналов приведено в п.4 - «Описание входов/выходов»

3. Условное графическое обозначение



4. Описание входов/выходов

Таблица 1 — Описание входов/выходов СФ-блока

Именованье сигнала	Направление	Описание	Примечание
s_awid[7:0]	вх	Идентификатор адреса записи	Не является обязательными сигналами AXI4-Lite, но необходимы при подключении СФ-блока к интерфейсу AXI4
s_awaddr[31:0]	вх	Адрес записи	
s_awvalid	вх	Строб адреса записи, показывающий момент валидности адреса записи на шине	
s_awready	вых	Готовность ведомого устройства на шине принять адрес записи	
s_wdata[31:0]	вх	Данные записи	
s_wstrb[3:0]	вх	Стробы данных записи. Показывают над какими байтами шины данных необходимо выполнять операцию записи	
s_wvalid	вх	Строб данных записи, показывающий момент валидности данных записи на шине	
s_wready	вых	Готовность ведомого устройства на шине принять данные записи	

s_bid[7:0]	вых	Идентификатор подтверждения записи	Не является обязательными сигналами AXI4-Lite, но необходимы при подключении СФ-блока к интерфейсу AXI4
s_bresp[1:0]	вых	Статус транзакции записи	
s_bvalid	вых	Строб статуса транзакции записи	
s_bready	вх	Готовность ведущего устройства на шине принять данные статуса записи	
s_arid[7:0]	вх	Идентификатор адреса чтения	Не является обязательными сигналами AXI4-Lite, но необходимы при подключении СФ-блока к интерфейсу AXI4
s_araddr[31:0]	вх	Адрес чтения	
s_arvalid	вх	Строб адреса чтения, показывающий момент валидности адреса чтения на шине	
s_arready	вых	Готовность ведомого устройства на шине принять адрес чтения	
s_rid[7:0]	вых	Идентификатор чтения	Не является обязательными сигналами AXI4-Lite, но необходимы при подключении СФ-блока к интерфейсу AXI4
s_rdata[31:0]	вых	Данные чтения	
s_rresp[1:0]	вых	Статус транзакции чтения	
s_rvalid	вых	Стробы данных чтения	
s_rready	вх	Готовность ведомого устройства принять данные чтения	
sck_m	вых	Синхросигнал сопровождения данных	
cs_m	вых	Строб активной транзакции	
mosi_m	вых	Выходная линия данных	
miso_m	вх	Входная линия данных	
mosi_oe_m	вых	Сигнал управления двунаправленной линией передачи данных в режиме 3-х проводного SPI интерфейса	
sck_s	вх	Синхросигнал сопровождения данных	
cs_s	вх	Строб активной транзакции	Активный уровень «0»
mosi_s	вх	Входная линия данных	
miso_s	вых	Выходная линия данных	
int_m	вых	Линия сигнала прерывания по событиям SPI в режиме ведущего	
int_s	вых	Линия сигнала прерывания по событиям SPI в режиме ведомого	
inta_m	вх	Подтверждение сигнала прерывания int_m	
inta_s	вх	Подтверждение сигнала прерывания int_s	
clk	вх	Синхросигнал шины AXI	
rst_n	вх	Сигнал начальной инициализации	Активный уровень «0»
sck_m_default	вх	Уровень, который будет выставлен на выходной линии SCK_M по включению	

		питания или после сигнала инициализации RST_N	
cs_m_default	вх	Уровень, который будет выставлен на выходной линии CS_M по включению питания или после сигнала инициализации RST_N	

5. Карта регистров

Таблица 2 — Карта регистров СФ-блока SPI

Смещение	Имя регистра	Режим	Разрядность	Описание	Значение по-умолчанию
0x000	rg_clk_div_m	Зп/Чт	16	Коэффициент деления частоты сигнала CLK для формирования сигнала SCK ведущего SPI интерфейса	0x0000
0x004	rg_ctrl_m	Зп/Чт	8	Регистр управления режимами ведущего SPI интерфейса	0x00
0x008	rg_data_out_m	Зп	32	Выходной регистр данных	0x00000000
0x00C	rg_data_in_m	Чт	32	Входной регистр данных	0x00000000
0x010	reserved				
0x014	rg_ctrl_s	Зп/Чт	8	Регистр управления режимами ведомого SPI интерфейса	0x00
0x018	rg_data_in_s	Чт	32	Входной регистр данных	0x00000000
0x01C	rg_data_out_s	Зп	32	Выходной регистр данных	0x00000000

Описание бит регистров управления:

rg_ctrl_m

№ бита	Описание
0	LSB/MSB («1»-LSB, «0»-MSB)
1	reserved
2	SCK_RISE/SCK_FALL («1»-данные записываются в ведомое устройство по переднему фронту сигнала SCK, «0»--данные записываются в ведомое устройство по rplytve фронту сигнала SCK)
3	CS_VALUE (значение сигнала CS)
4	
5	NUM_BYTES (количество байт для обмена. Изменяется в пределах от 1 до 4)
6	
7	MOSI_OE_VALUE (значение сигнала mosi_oe_m)

rg_ctrl_s

№ бита	Описание
0	LSB/MSB («1»-LSB, «0»-MSB)
1	reserved
2	SCK_RISE/SCK_FALL («1»-данные записываются в ведомое устройство по переднему фронту сигнала SCK, «0»--данные записываются в ведомое устройство по рflytve фронту сигнала SCK)
3	reserved
4	
5	NUM_BYTES (количество байт для обмена. Изменяется в пределах от 1 до 4)
6	
7	reserved

6. Описание функционирования

СФ-блок SPI_M_S является ведомым устройством на шине AXI4-Lite и работает как формирователь интерфейса SPI как в режиме ведущего, так и в режиме ведомого как по отдельности, так и одновременно. Для каждого режима формируются отдельные сигналы системных прерываний. Реализованы режим 4-х и 3-х проводного SPI. Временные параметры и функциональная нагрузка сигналов (направление передачи данных LSB, MSB по MOSI и MISO; полярность SCK и CS; фаза SCK) настраиваются через соответствующие регистры управления.

Перед началом обмена выполняется настройка регистров управления и записи данных в регистр выходных данных, в случае режима ведущего SPI интерфейса. Обмен запускается при переключении сигнала CS_M в активное состояние. В случае ведомого интерфейса ожидается установка внешнего сигнала CS_S в активное состояние. Далее выполняется прием/передача данных в/из регистров данных в соответствие с количеством байт определенным регистром управления. После этого выставляется активный уровень на выходной линии прерывания INT_x. Прерывание будет стоять до тех пока не придет системный сигнал подтверждения прерывания INTA_x.

7. Система тактирования

Весь СФ-блок функционирует с использованием внешнего сигнала синхронизации CLK. Формирование выходного сигнала SCK_M происходит путем тактирования полученными с использованием коэффициента деления rg_clk_div_m.

8. Верификация

Блок-диаграмма тестового окружения

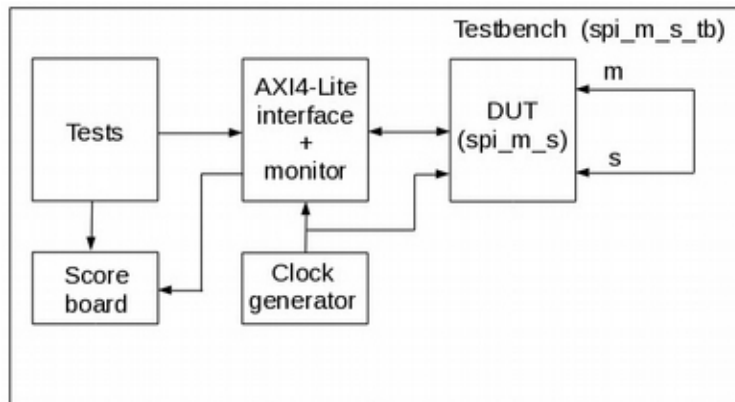


Рисунок 2 — Блок-диаграмма тестового окружения

Реализованные тесты:

1. SPI_s -> SPI_m, 1 byte, sck — rise, sck_default=1, LSB
SPI_m -> SPI_s, 1 byte, sck — rise, sck_default=1, LSB
2. SPI_s -> SPI_m, 1 byte, sck — fall, sck_default=1, MSB
SPI_m -> SPI_s, 1 byte, sck — fall, sck_default=1, MSB
3. SPI_s -> SPI_m, 3 bytes, sck — rise, sck_default=0, LSB
SPI_m -> SPI_s, 3 bytes, sck — rise, sck_default=0, LSB
4. SPI_s -> SPI_m, 2 bytes, sck — rise, sck_default=0, LSB
SPI_m -> SPI_s, 2 bytes, sck — rise, sck_default=0, LSB

Пример отчета:

```
run all
Test 1.1 : SPI_M_S (SPI_s -> SPI_m, 1 byte, sck_rise, sck_defaul=1, lsb) completed successfully!
Read transaction successfully done
transaction_ok=1
Test 1.1 : SPI_M_S (SPI_m -> SPI_s, 1 byte, sck_rise, sck_defaul=1, lsb) completed successfully!
Test 1.2 : SPI_M_S (SPI_s -> SPI_m, 1 byte, sck_fall, sck_defaul=1, msb) completed successfully!
Read transaction successfully done
transaction_ok=1
Test 1.2 : SPI_M_S (SPI_m -> SPI_s, 1 byte, sck_fall, sck_defaul=1, msb) completed successfully!
Test 1.3 : SPI_M_S (SPI_s -> SPI_m, 3 byte, sck_rise, sck_defaul=0, lsb) completed successfully!
Read transaction successfully done
transaction_ok=1
Test 1.3 : SPI_M_S (SPI_m -> SPI_s, 3 byte, sck_rise, sck_defaul=0, lsb) completed successfully!
Test 1.4 : SPI_M_S (SPI_s -> SPI_m, 2 byte, sck_rise, sck_defaul=0, lsb) completed successfully!
Read transaction successfully done
transaction_ok=1
Test 1.4 : SPI_M_S (SPI_m -> SPI_s, 2 byte, sck_rise, sck_defaul=0, lsb) completed successfully!
$finish called at time : 15610 ns
```