
AESOP Embedded Forum
Sejong Lee (<http://www.aesop-embedded.org>)

개 요

이 문서는 S3C6400 데이터시트에서
제 29장 SPI를 번역한 자료입니다.

제 29장. SPI

29.1 개관

Serial Peripheral Interface (SPI:직렬 장치 인터페이스)는 직렬 데이터 전송과 연결될 수 있다. SPI는 전송과 수신을 위해 두 개의 8, 16, 32비트 시프트 레지스터를 포함하고 있다. SPI로 전송하는 동안 데이터는 동시에 전송(직렬로 시프트되어 나간다.)되고 수신(직렬로 시프트되어 들어온다.)된다. SPI는 National Semiconductor사의 Microwire와 Motorola의 Serial Peripheral Interface에 대한 프로토콜을 지원한다.

29.2 특징

SPI는 다음의 특징들을 지원한다.

- Full duplex (양방향 통신)
- 8/16/32비트 시프트 레지스터 (TX/RX를 위한)
- 8비트 prescale logic (주파수 조절 로직)
- 3개의 클럭 소스
- 8비트/16비트/32비트 버스 인터페이스
- 모토로라의 SPI 프로토콜 및 내셔널 세미컨덕터의 Microwire
- 두 개의 독립적인 전송과 수신 FIFO, each 16 samples deep by 32-bit wide
- 마스터 모드와 슬레이브 모드
- 전송없는 수신동작
- 마스터 Tx/마스터 RX/슬레이브 Rx의 최대 주파수는 50MHz까지이다.
- Slave Tx의 최대 주파수는 20MHz까지이다.

29.3 신호 설명

다음의 표 리스트는 SPI와 외부 디바이스와의 외부 신호이다. SPI의 모든 포트는 SPI를 사용하지 않을 때 일반적인 입출력 포트로서 사용할 수 있다. 더 많은 정보는 General Purpose I/O 파트를 참고하자.

Table 29-1. 외부 신호 설명

Name	Direction	Description
XspiCLK	Inout	XspiCLK는 데이터를 전송하기 위한 제어시간으로 사용되는 시리얼 클럭이다.
XspiMISO	Inout	마스터 모드에서 이 포트는 입력포트이다. 입력모드는 slave출력 포트로부터 데이터를 얻을 때 사용된다. 슬레이브 모드에서는 이 포트를 통해 마스터로 데이터가 전송된다.
XspiMOSI	Inout	마스터 모드에서 이 포트는 출력포트이다. 이 포트는 마스터 출력 포트로부터 데이터를 전송할 때 사용된다. 슬레이브 모드에서는 이 포트를 통해 마스터로부터 데이터를 수신한다.
XspiCS	Inout	Slave 선택 신호, TX/RX 시퀀스의 모든 데이터는 XspiCS가 low일 때 실행된다.

29.4 동작

S3C6400의 SPI는 S3C6400과 외부장치 사이에서 1비트 직렬 데이터를 전송한다. S3C6400의 SPI는 전송 또는 수신 FIFO를 분리하고 양방향으로 동시에 데이터를 전송하기 위해 CPU 또는 DMA를 지원한다. SPI는 두 개의 채널을 가지는데, TX채널, RX채널이다. TX채널은 Tx FIFO로부터 외부 장치까지의 경로를 갖는다. RX채널은 외부장치로부터 RX FIFO까지 경로를 갖는다.

FIFO에 데이터를 쓰기 위해서는 CPU (또는 DMA)는 반드시 SPI_TX_DATA 레지스터에 데이터를 써야한다. 레지스터상의 데이터는 자동으로 TxFIFO로 이동한다. Rx FIFO로부터 데이터를 읽으려면, CPU (또는 DMA)는 반드시 SPI_RX_DATA 레지스터를 액세스해야 한다. 그리고 데이터는 SPI_RX_DATA 레지스터에 자동으로 보내진다.

29.4.1 동작 모드

HS_SPI는 두 개의 모드를 가지고 있는데, 마스터 모드와 슬레이브 모드이다. 마스터 모드에서 HS_SPICLK이 생성되고 외부장치로 전송된다. Slave 선택 신호인 XspiCS#은 low일 때 데이터 유효성을 나타낸다. XspiCS#은 전송 또는 수신시의 패킷 시작전에 low로 되어 있어야 한다.

29.4.2 FIFO 액세스

S3C6400의 SPI는 FIFO 접근을 위해 CPU 액세스와 DMA 액세스를 지원한다. FIFO를 위한 CPU 액세스와 DMA 액세스의 데이터 크기는 8비트 또는 32비트 데이터를 선택할 수 있다. 만약 8비트 데이터 크기를 선택했다면, 유효 비트는 0비트부터 7비트까지이다. CPU 액세스는 보통 trigger threshold에 의해 on/off되고 이것은 사용자 정의된 것이다. 각각의 FIFO의 trigger level은 0바이트부터 64바이트까지 설정된다. SPI_MODE_CFG 레지스터의 TxDMAOn 또는 RxDMAOn 비트는 반드시 DMA 액세스 사용으로 설정되어 있어야 한다. DMA 액세스는 오직 single 전송과 4-busrt 전송만 지원한다. TX FIFO에서 DMA 요청 신호는 FIFO가 꽉 차 있을 동안 high이다. RX FIFO에서 DMA 요청 신호는 만약 FIFO가 비어있지 않다면 high이다.

29.4.3 RX FIFO에서의 TRAILING 바이트

Rx FIFO에서 샘플의 수가 INT모드 또는 DMA 4 burst 모드에서 threshold값 보다 작고 추가적인 데이터 수신에 없을 때, 남은 바이트를 trailing 바이트라 부른다. RX FIFO에서 이 바이트를 제거하기 위해 내부 타이머와 인터럽트 신호가 사용된다. 내부 타이머 값은 APB 버스 클럭에 기반하여 1024 클럭까지 설정할 수 있다. 타이머 값이 0이 될 때, 인터럽트 신호가 발생하고 CPU는 FIFO에 있는 trailing 바이트를 제거할 수 있다.

29.4.4 패킷 수 제어

SPI는 마스터 모드에서 수신되는 패킷의 수를 조절할 수 있다. 만약 어떤수의 패킷을 수신하게 되면, 단지 SFR을 설정한다.(Packet_Count_reg) SPI는 패킷의 수가 사용자가 설정한 것과 같을 때SPICLK 생성을 정지한다. 이 기능을 불러올 때 소프트웨어와 하드웨어 리셋은 필수이다. (소프트웨어 리셋은 특수기능 레지스터를 제외하고 모두 clear한다. 그러나 하드웨어 리셋은 모든 레지스터를 clear한다.)

29.4.5 Chip Select Control (칩 선택 제어)

XspiCS#은 auto control과 manual control을 선택할 수 있다.

29.4.5.1 Manual Control 모드

AUTO_N_MANUAL은 반드시 clear해야 한다.(default value is 0) XspiCS#의 레벨은 NSSOUT 비트에 의해 제어된다.

29.4.5.2 Auto Control 모드

AUTO_N_MANUAL은 반드시 set해야 한다. XspiCS는 패킷과 패킷사이에서 자동적으로 toggle(비트 값이 바뀐다.)된다. XspiCS의 비활성화 기간은 NCS_TIME_COUNT에 의해 제어된다. NSSOUT은 이 시점에서는 사용할 수 없다.

29.4.6 SPI 전송 형식

S3C6400은 데이터 전송을 위해 4가지의 다른 형식을 지원한다. Figure 29-1은 SPICLK에 대한 4가지 파형을 나타낸다.

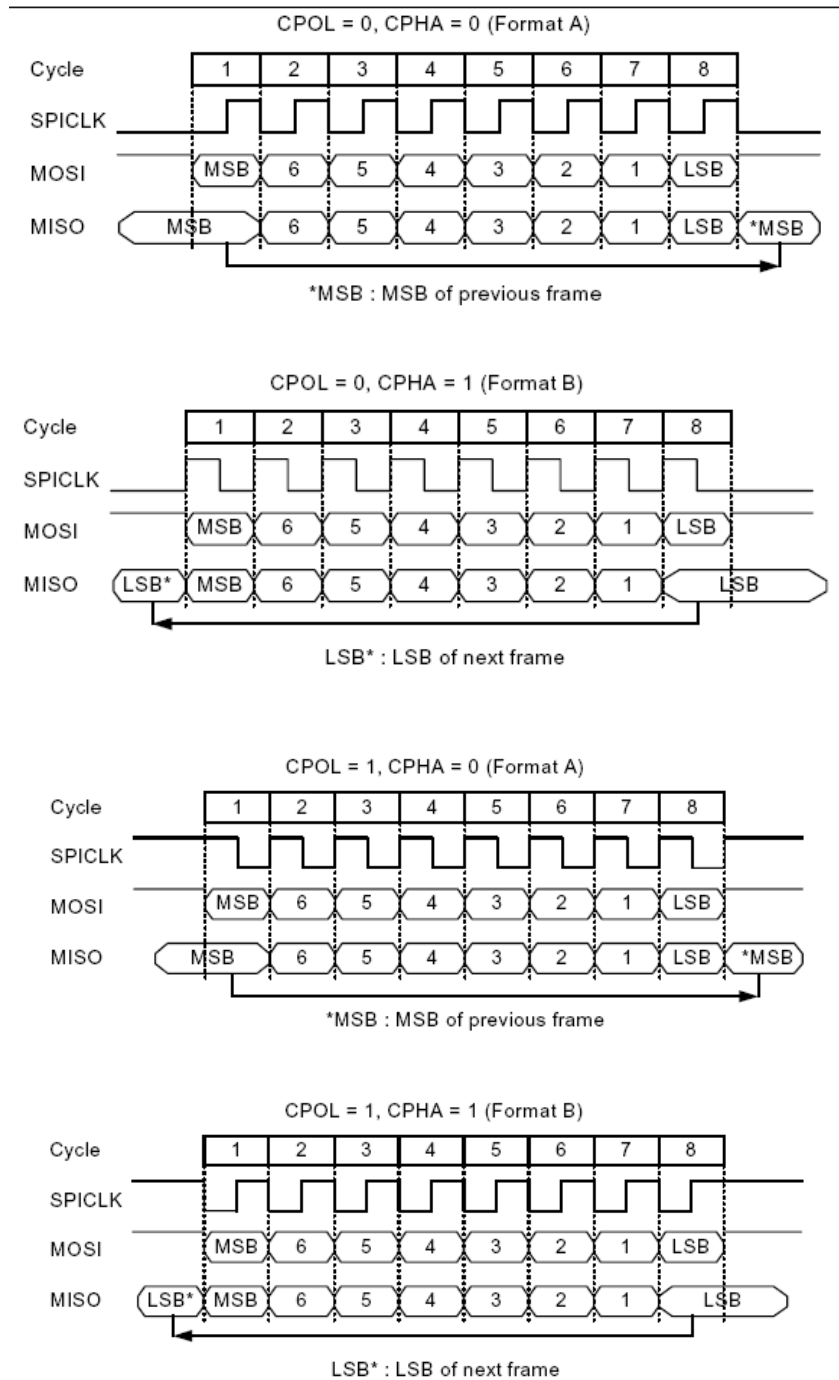


Figure 29-1. SPI Transfer Format

29.5 External loading capacitance (외부에 연결되는 캐패시터)

10pF보다 큰 출력 캐패시터를 가지는 SPI 채널0과 30pF보다 큰 출력 캐패시터를 가지는 채널1은 보증되지 않는다.

채널0에서 출력 캐패시터는 10pF보다 작아야 한다.
채널1에서 출력 캐패시터는 **30pF**보다 작아야 한다.

29.6 특수 기능 레지스터 요약

29.6.1 특수 기능 레지스터 세팅 절차

특수 기능 레지스터는 다음의 절차로 설정되어야 한다. (nCS manual 모드)

1. 전송 타입 설정 (CPOL & CPHA set)
2. Clock configuration register 설정
3. SPI 모드 configuration register 설정
4. SPI INT_EN register 설정
5. Packet Count configuration register if necessary 설정
6. Tx or Rx Channel on 설정
7. NSSOUT low to start Tx or Rx operation 설정
 - A. NSSOUT Bit to low, then start TX data writing 설정
 - B. If auto chip selection bit is set, should not control NSSOUT.

29.6.2 특수 기능 레지스터

Register	Address	R/W	Description	Reset Value
CH_CFG(Ch0)	0x7F00B000	R/W	SPI configuration register	0x0
CH_CFG(Ch1)	0x7F00C000	R/W	SPI configuration register	0x0

CH_CFG	Bit		Description	Initial State
SW_RST	[5]	R/W	Software reset. The following registers and bits are cleared by this bit. Rx/Tx FIFO Data, SPI_STATUS Once reset, this bit must be clear manually. 0: inactive 1: active	1'b0
SLAVE	[4]	R/W	Whether SPI Channel is Master or Slave 0: Master 1: Slave	1'b0
CPOL	[3]	R/W	Determine an active high or active low clock 0: active high 1: active low	1'b0
CPHA	[2]	R/W	Select one of the two fundamentally different transfer format 0: format A 1: format B	1'b0
RX_CH_ON	[1]	R/W	SPI Rx Channel On 0: Channel Off 1: Channel On	1'b0
TX_CH_ON	[0]	R/W	SPI Tx Channel On 0: Channel Off 1: Channel On	1'b0

Register	Address	R/W	Description	Reset Value
CLK_CFG(Ch0)	0x7F00B004	R/W	Clock configuration register	0x0
CLK_CFG(Ch1)	0x7F00C004	R/W	Clock configuration register	0x0

CLK_CFG	Bit		Description	Initial State
SPI_CLKSEL	[10:9]	R/W	Clock source selection to generate SPI clock-out 00 : PCLK 01 : USBCLK 10 : Epll clock 11 : reserved * For using USBCLK source, The USB_SIG_MASK at system controller must be set. *Epll clock is from System Controller and has 4 sources: MOUT _{EPLL} , DOUT _{MPLL} , FIN _{EPLL} , 27MHZ	2'b0
ENCLK	[8]	R/W	Clock enable/disable 0 : disable 1 : enable	1'b0
SPI_SCALER	[7:0]	R/W	SPI clock-out division rate SPI clock-out = Clock source / (2 x (Prescaler value + 1))	8'h0

Register	Address	R/W	Description	Reset Value
MODE_CFG(Ch0)	0x7F00B008	R/W	SPI FIFO control register	0x0
MODE_CFG(Ch1)	0x7F00C008	R/W	SPI FIFO control register	0x0

MODE_CFG	Bit		Description	Initial State
CH_WIDTH	[30:29]	R/W	00 : Byte 01 : Halfword 10 : Word 11 : reserved	2'b0
TRAILING_CNT	[28:19]	R/W	Count value from writing the last data in RX FIFO to flush trailing bytes in FIFO	10'b0
BUS_WIDTH	[18:17]	R/W	00: Byte 01: Halfword 10 : Word 11:reserved	2'b0
RX_RDY_LVL	[16:11]	R/W	Rx FIFO trigger level in INT mode. Trigger level is from 6'h0 to 6'h40. The value means byte number in RX FIFO	6'b0
TX_RDY_LVL	[10:5]	R/W	Tx FIFO trigger level in INT mode. Trigger level is from 6'h0 to 6'h40. The value means byte number in TX FIFO	6'b0
reserved	[4:3]	-	-	-
RX_DMA_SW	[2]	R/W	Rx DMA mode on/off 0 : DMA mode off 1 : DMA mode on	1'b0
TX_DMA_SW	[1]	R/W	Tx DMA mode on/off 0 : DMA mode off 1 : DMA mode on	1'b0
DMA_TYPE	[0]	R/W	DMA transfer type, single or 4 busts. 0 : single 1 : 4 burst DMA transfer size must be set as the same size in DMA as it is in SPI.	1'b0

****채널 전송 크기는 버스 전송 사이즈보다 같거나 작아야 한다.**

Register	Address	R/W	Description	Reset Value
CS_REG(Ch0)	0x7F00B00C	R/W	Slave selection signal control register	0x1
CS_REG(Ch1)	0x7F00C00C	R/W	Slave selection signal control register	0x1

CS_REG	Bit		Description	Initial State
NCS_TIME_COUNT	[9:4]	R/W	NSSOUT inactive time = $((nCS_time_count+3)/2) \times SPICLKout$	6'b0
reserved	[3:2]	-	reserved	-
AUTO_N_MANUAL	[1]	R/W	Chip select toggle manual or auto selection 0: manual 1: Auto	1'b0
NSSOUT	[0]	R/W	Slave selection signal (manual only) 0: active 1: inactive	1'b1

Register	Address	R/W	Description	Reset Value
SPI_INT_EN(Ch0)	0x7F00B010	R/W	SPI Interrupt Enable register	0x0
SPI_INT_EN(Ch1)	0x7F00C010	R/W	SPI Interrupt Enable register	0x0

SPI_INT_EN	Bit		Description	Initial State
INT_EN_TRAILING	[6]	R/W	Interrupt Enable for trailing count to be zero 0: Disable 1:Enable	1'b0
INT_EN_RX_OVERRUN	[5]	R/W	Interrupt Enable for RxOverrun 0: Disable 1:Enable	1'b0
INT_EN_RX_UNDERRUN	[4]	R/W	Interrupt Enable for RxUnderrun 0: Disable 1:Enable	1'b0
INT_EN_TX_OVERRUN	[3]	R/W	Interrupt Enable for TxOverrun 0: Disable 1:Enable	1'b0
INT_EN_TX_UNDERRUN	[2]	R/W	Interrupt Enable for TxUnderrun. In slave mode, this bit must be clear first after turning on slave TX path. 0: Disable 1:Enable	1'b0
INT_EN_RX_FIFO_RDY	[1]	R/W	Interrupt Enable for RxFifoRdy(INT mode) 0: Disable 1:Enable	1'b0
INT_EN_TX_FIFO_RDY	[0]	R/W	Interrupt Enable for TxFifoRdy(INT mode) 0: Disable 1:Enable	1'b0

Register	Address	R/W	Description	Reset Value
SPI_STATUS(Ch0)	0x7F00B014	R	SPI status register	0x0
SPI_STATUS(Ch1)	0x7F00C014	R	SPI status register	0x0

SPI_STATUS	Bit		Description	Initial State
TX_DONE	[21]	R	Indication of transfer done in Shift register 0 : all case except blow case 1 : when tx fifo and shift register are empty	1'b0
TRAILING_BYTE	[20]	R	Indication that trailing count is zero	1'b0
RX_FIFO_LVL	[19:13]	R	Data level in RX FIFO 0 ~ 7'h40 byte	7'b0
TX_FIFO_LVL	[12:6]	R	Data level in TX FIFO 0 ~ 7'h40 byte	7'b0
RX_OVERRUN	[5]	R	Rx Fifo overrun error 0: no error, 1: overrun error	1'b0
RX_UNDERRUN	[4]	R	Rx Fifo underrun error 0: no error, 1: underrun error	1'b0
TX_OVERRUN	[3]	R	Tx Fifo overrun error 0: no error, 1: overrun error	1'b0
TX_UNDERRUN	[2]	R	Tx Fifo underrun error 0: no error, 1: underrun error	1'b0
RX_FIFO_RDY	[1]	R	0 : data in FIFO less than trigger level 1 : data in FIFO more than trigger level	1'b0
TX_FIFO_RDY	[0]	R	0 : data in FIFO more than trigger level 1 : data in FIFO less than trigger level	1'b0

Register	Address	R/W	Description	Reset Value
SPI_TX_DATA(Ch0)	0x7F00B018	W	SPI TX DATA register	0x0
SPI_TX_DATA(Ch1)	0x7F00C018	W	SPI TX DATA register	0x0

SPI_TX_DATA	Bit		Description	Initial State
TX_DATA	[31:0]	W	This field contains the data to be transmitted over the SPI channel.	32'b0

Register	Address	R/W	Description	Reset Value
SPI_RX_DATA(Ch0)	0x7F00B01C	R	SPI RX DATA register	0x0
SPI_RX_DATA(Ch1)	0x7F00C01C	R	SPI RX DATA register	0x0

SPI_RX_DATA	Bit		Description	Initial State
RX_DATA	[31:0]	R	This field contains the data to be received over the SPI channel.	32'b0

Register	Address	R/W	Description	Reset Value
PACKET_CNT_REG(Ch0)	0x7F00B020	R/W	Count how many data master gets	0x0
PACKET_CNT_REG(Ch1)	0x7F00C020	R/W	Count how many data master gets	0x0

PACKET_CNT_REG	Bit		Description	Initial State
PACKET_CNT_EN	[16]	R/W	Enable bit for packet count 0: Disable 1:Enable	1'b0
COUNT_VALUE	[15:0]	R/W	Packet count value	16'b0

Register	Address	R/W	Description	Reset Value
PENDING_CLR_REG(Ch0)	0x7F00B024	R/W	Status pending clear register	0x0
PENDING_CLR_REG(Ch1)	0x7F00C024	R/W	Status pending clear register	0x0

PENDING_CLR_REG	Bit	R/W	Description	Initial State
TX_UNDERRUN_CLR	[4]	R/W	TX underrun pending clear bit 0: non-clear 1:clear	1'b0
TX_OVERRUN_CLR	[3]	R/W	TX overrun pending clear bit 0: non-clear 1:clear	1'b0
RX_UNDERRUN_CLR	[2]	R/W	RX underrun pending clear bit 0: non-clear 1:clear	1'b0
RX_OVERRUN_CLR	[1]	R/W	RX overrun pending clear bit 0: non-clear 1:clear	1'b0
TRAILING_CLR	[0]	R/W	Trailing pending clear bit 0: non-clear 1:clear	1'b0

Register	Address	R/W	Description	Reset Value
SWAP_CFG(Ch0)	0x7F00B028	R/W	SWAP config register	0x0
SWAP_CFG (Ch1)	0x7F00C028	R/W	SWAP config register	0x0

SWAP_CFG	Bit		Description	Initial State
RX_HWORD_SWAP	[7]	R/W	0: off 1: swap	1'b0
RX_BYTE_SWAP	[6]	R/W	0: off 1: swap	1'b0
RX_BIT_SWAP	[5]	R/W	0: off 1: swap	1'b0
RX_SWAP_EN	[4]	R/W	Swap enable 0 : normal 1 : swap	1'b0
TX_HWORD_SWAP	[3]	R/W	0: off 1: swap	1'b0
TX_BYTE_SWAP	[2]	R/W	0: off 1: swap	1'b0
TX_BIT_SWAP	[1]	R/W	0: off 1: swap	1'b0
TX_SWAP_EN	[0]	R/W	Swap enable 0 : normal 1 : swap	1'b0

** Data size must be larger than swap size.

Register	Address	R/W	Description	Reset Value
FB_CLK_SEL (Ch0)	0x7F00B02C	R/W	Feedback clock selecting register.	0x3
FB_CLK_SEL (Ch1)	0x7F00C02C	R/W	Feedback clock selecting register.	0x3

FB_CLK_SEL	Bit		Description	Initial State
SPICLKOUT_DELAY	[2]	R/W	0 : no additional delay 1 : 2.7ns delay (base on typical)	1'b0
FB_CLK_SEL	[1:0]	R/W	00 : 0nS additional delay 01 : 3nS additional delay 10 : 6nS additional delay 11 : 9nS additional delay * Delay based on typical condition.	2'b3

Revision History

Date	Editor	Version	Descriptions
2008-07-30	Sejong Lee	1.0	최초 작성