

M.E.S.' Cusrow Wadia Institute of Technology ,Pune



A Project Presentation on
**MULTIPROTOCOL
TRANSCEIVER**

Project Guide

Prof. A.S.Chandak
Prof. & Head of Dept.
Department of E&TC

Candidates

117019 Kad Saurabh
117022 Kane Rasik

AGENDA

- ▶ Objective and Proposed System
- ▶ Related Work
- ▶ System Elaboration –
 - Hardware Platform
 - Software Platform
 - Results
- ▶ Conclusion
- ▶ Future Scope

INTRODUCTION

OBJECTIVE

- ▶ Design an embedded system which accept information from external source, transmit it to another node on network, and accept information from the node using one of the protocol among the available multiple protocols and represent in the original form. The network can be built using one of the protocols from available multiple protocols and communication standards.

RELEVANCE

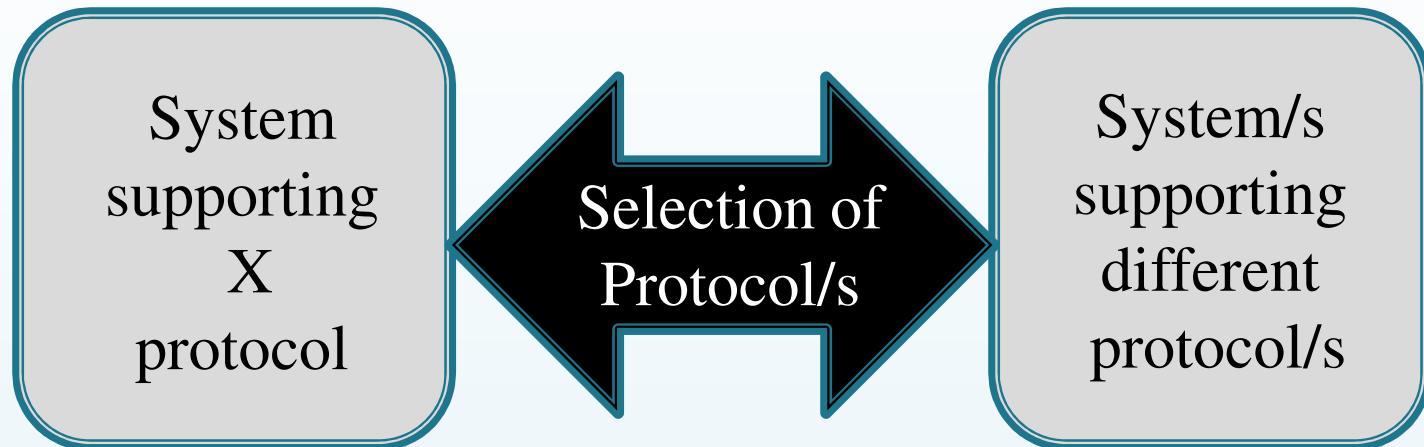


Fig. 1 System Relevance

- ▶ Plays an important role in building network supporting different protocols

LITERATURE REVIEW

EXISTING SYSTEM

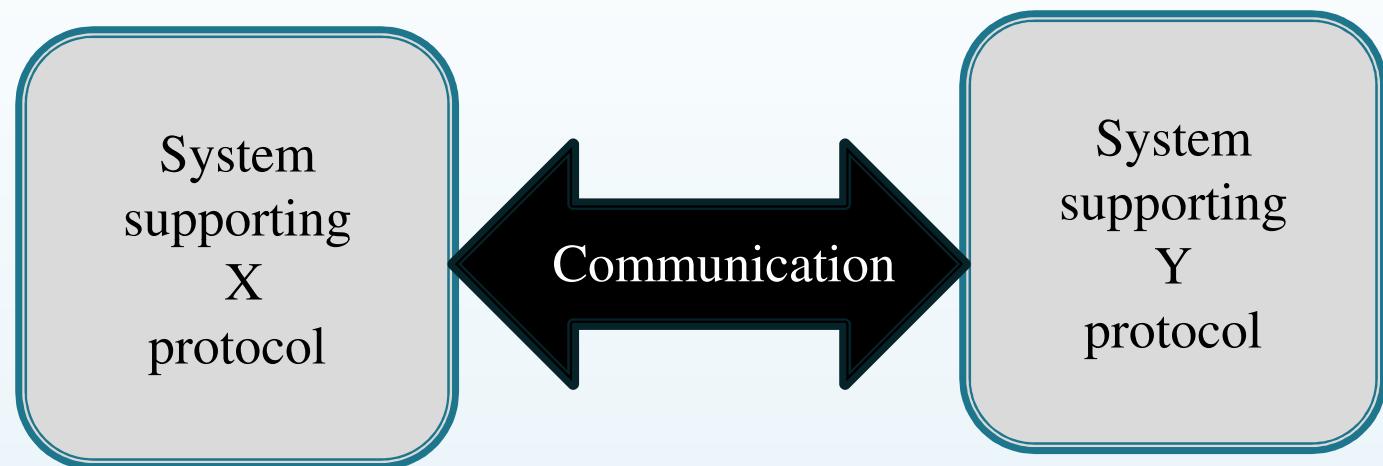


Fig.2 Existing System

PROPOSED SYSTEM

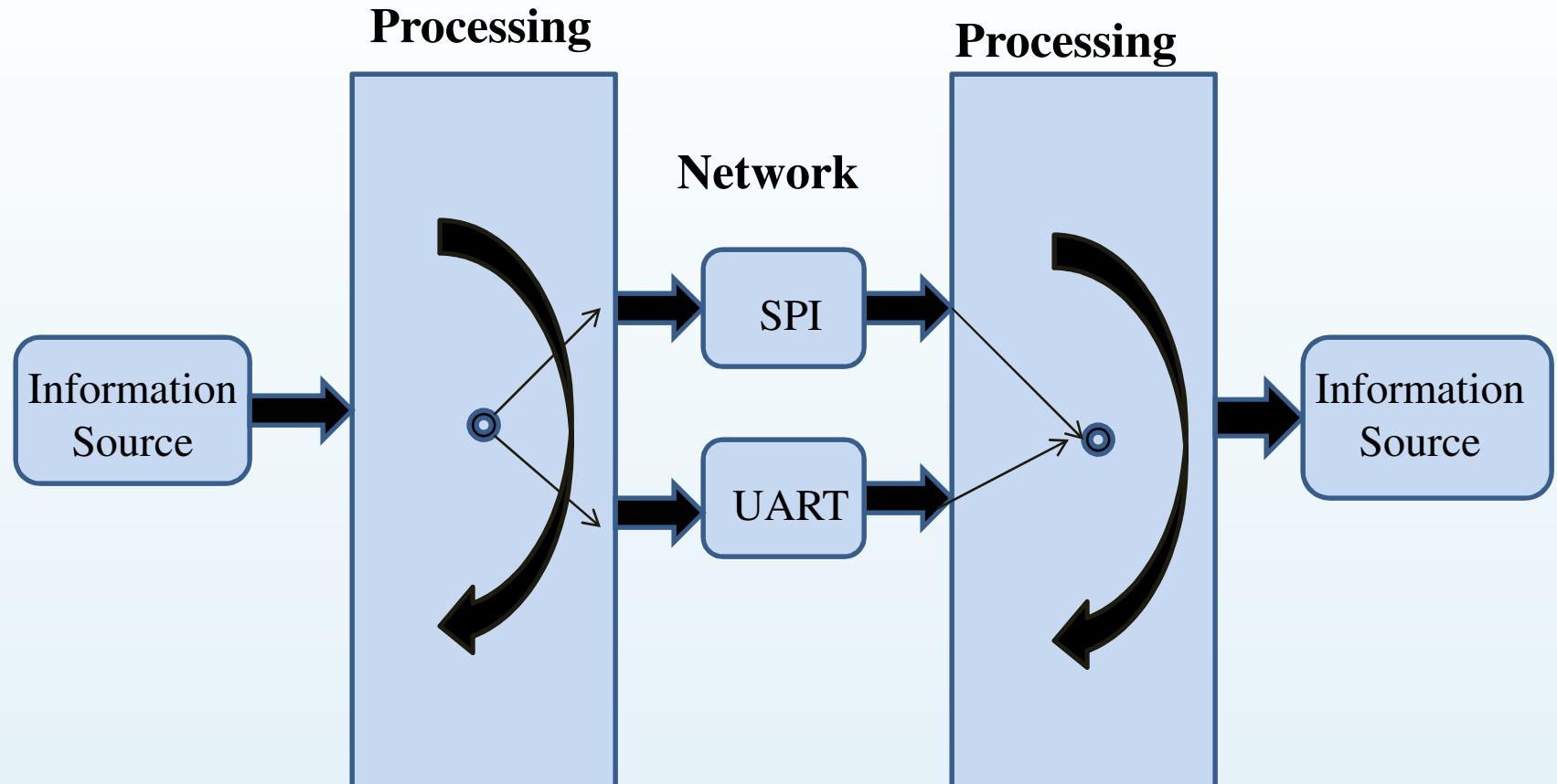


Fig.3 Proposed System

COMPARISON WITH EXISTING SYSTEM

Existing system	Proposed system
Multi Protocol facility is not available in a single system	Multi Protocol facility is available in a single system
Expensive	Cheaper
It is simple to construct	It is difficult to construct

SYSTEM ELABORATION

HARDWARE PLATFORM

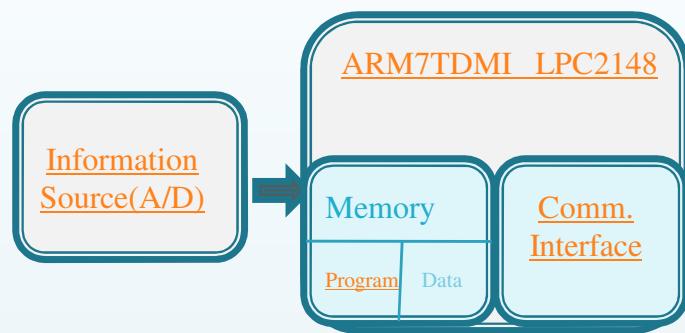
1. Central processing unit :- LPC 2148
2. User interface :- Keypad, LCD Module,
Source(Analog/Digital)
3. Protocols Supported :- UART, SPI
4. Power Supply :- USB, ADAPTER
5. ARM7TDMI Programmer (Parallel port/ISP)

SOFTWARE PLATFORM

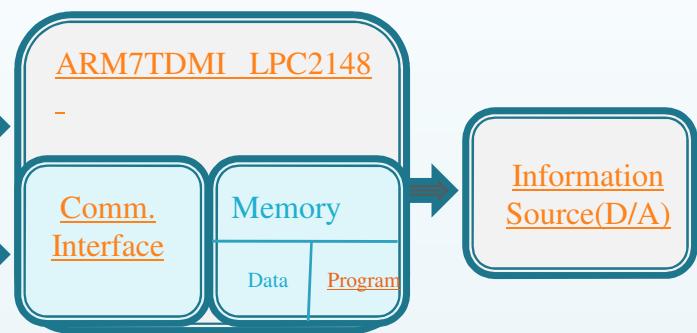
1. IDE for ARM7TDMI
2. ISP Software(e.g. Flash Magic)

BLOCK DIAGARM

Multiprotocol Transmitter



Multiprotocol Receiver



System Block Diagram

PROBLEMS FACED

- ▶ Synchronization
- ▶ Speed of hyper-terminal

ADVANTAGES

- ▶ Systems with different protocols can communicate without hardware and/or software changes
- ▶ Multicasting and/or broadcasting of information over multiple protocols is possible
- ▶ Protocol selection facility
- ▶ User friendly system with small size, low weight
- ▶ Easy provision of hardware plug-ins for each protocol
- ▶ Optimal user interfacing viz. LCD, keyboard etc.

APPLICATIONS

- ▶ Part of communication radio
- ▶ Can manage multi-source, multi-protocol, multi-destination system information routing

FUTURE SCOPE

- ▶ Support other protocols
- ▶ Secondary memory support
- ▶ Wireless transmission can be possible

CONCLUSION

- ▶ ARM's approach has been to design RISC processor architectures with instruction sets that provide efficient support for various applications, space with an optimal balance between hardware and software implementations. The proposed system needs this key feature of handling the hardware and software implementation of various protocols
- ▶ Thus we select ARM7TDMI family which provides high performance and LPC 2148 microcontroller as it satisfies all system requirements efficiently

REFERENCES

- ▶ ARM System Developer's Guide Designing and Optimizing, System Software by, Andrew N. Sloss, Dominic Symes, Chris Wright
- ▶ LPC2141/42/44/46/48 Product Data Sheet
- ▶ UM10139LPC214x User manual Rev. 3 — 4 October 2010
- ▶ www.wikpedia.com
- ▶ www.nxp.com
- ▶ www.google.com

Thank You.

Your Questions, Please.

Main features of LPC2148 which satisfy the system requirements

▶ **UARTx**

- Contain two UARTs
- Standard transmit and receive data lines
- Provides a full modem control handshake interface
- Fractional baud rate generator for both UARTs
- Auto-CTS/RTS flow-control functions

▶ **SPI serial I/O controller**

- The LPC2141 contain one SPI controller

TRANSMITTER DATA PROCESSING

- 10 bit successive approximation analog to digital converter
- Input multiplexing among 6 pins FOR ADC0.
- Power-down mode.
- Measurement range 0 V to VREF, typically 3 V.
- 10 bit conversion time $\geq 2.44 \mu\text{s}$.
- Burst conversion mode for single or multiple inputs.
- Optional conversion on transition on input pin or Timer Match signal.

RECEIVER DATA PROCESSING

- 10 bit digital to analog converter
- Resistor string architecture
- Buffered output
- Power-down mode
- Selectable speed vs. power, The settling time and o/p current of the DAC can be designed in two combinations,
 1. 1 μ s, 700mA
 1. 2.5 μ s, 350mA

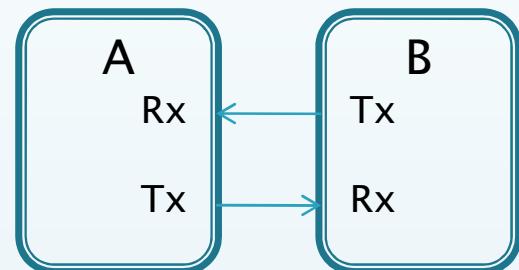
COMMUNICATION PROTOCOLS

A] Universal Asynchronous Receiver Transmitter (UART)

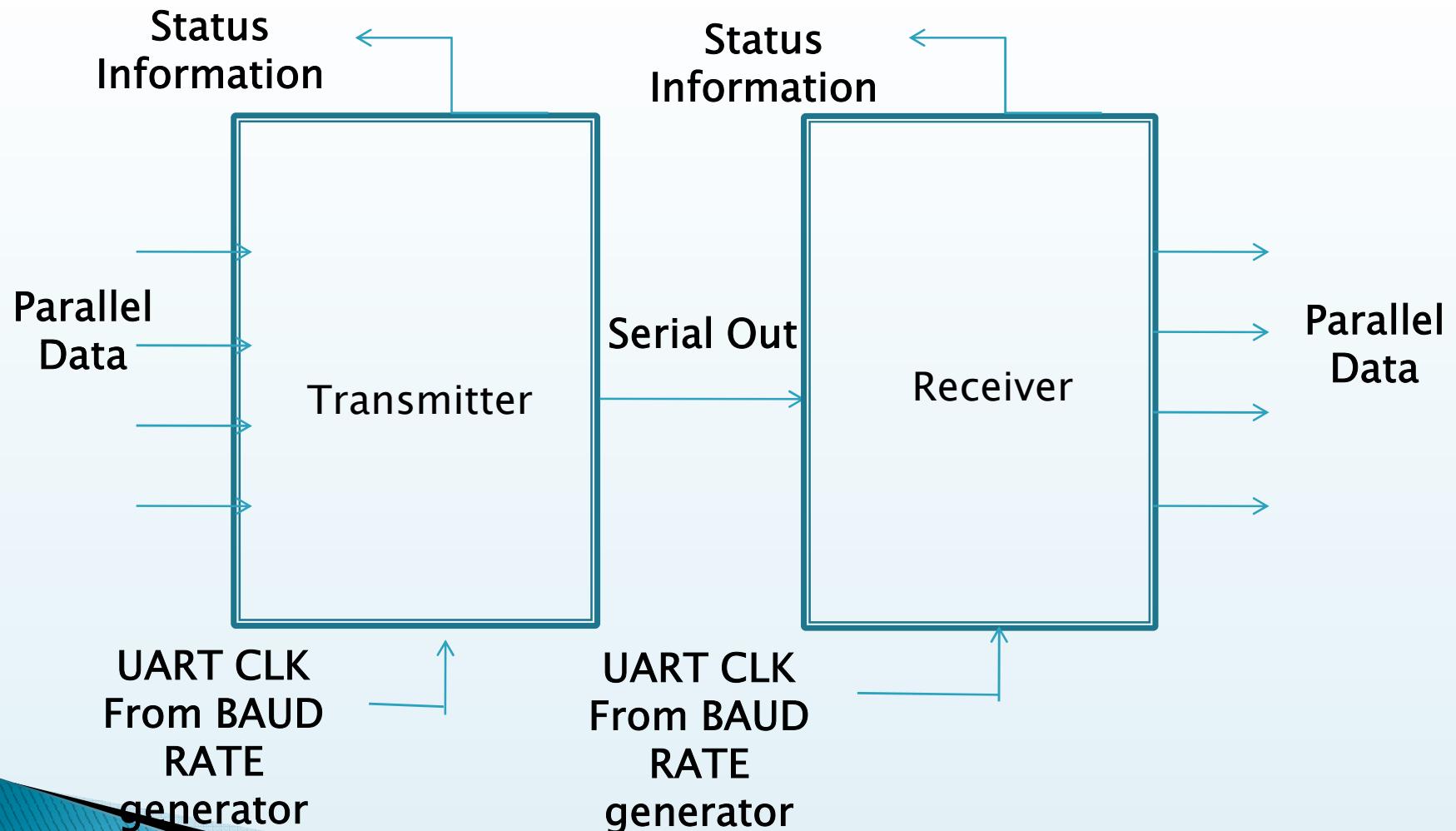
B] Serial Peripheral Interface (SPI)

UART INTERFACE

- UART: Universal Asynchronous Receiver Transmitter
- Full duplex
- Asynchronous
- No common clock required
- Two wires: Rx, Tx



UART TRANSCEIVER DIAGRAM



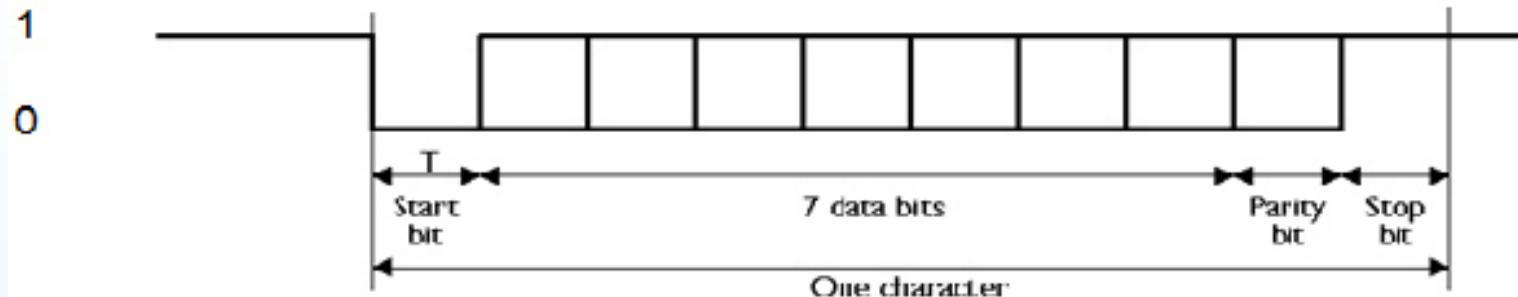
UART TRANSMITTER

- converts data from parallel to serial format
- Inserts start and stop bits
- Calculates and inserts parity bit if required
- Output bit rate is determined by the UART clock

UART RECEIVER

- Synchronises with transmitter
- Samples the input data line at a clock rate
- Reads each bit in middle of bit period
- Removes the start and stop bits
- Optional calculates and checks the parity bit
- Presents the received data value in parallel form

ASYNCHRONOUS SERIAL TRANSMISSION



Serial transmission (least significant bit first)

Example: Letter M = 1001101 (even parity)

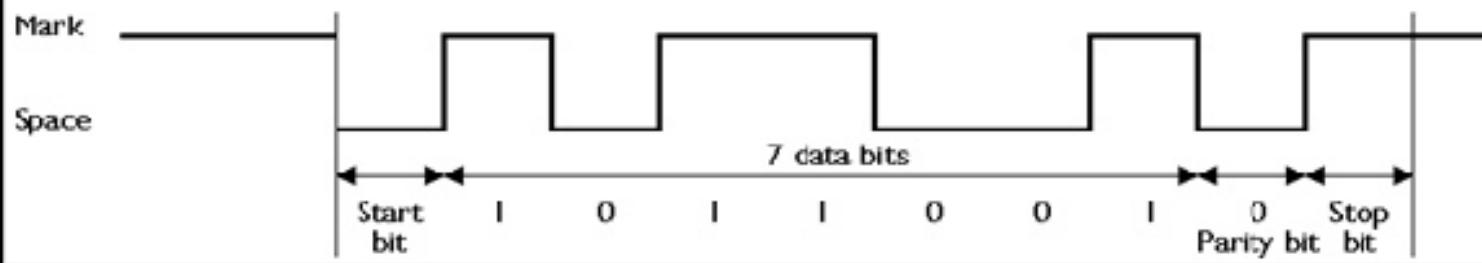


Fig. Asynchronous Serial Transmission

ASYNCHRONOUS SERIAL RECEPTION

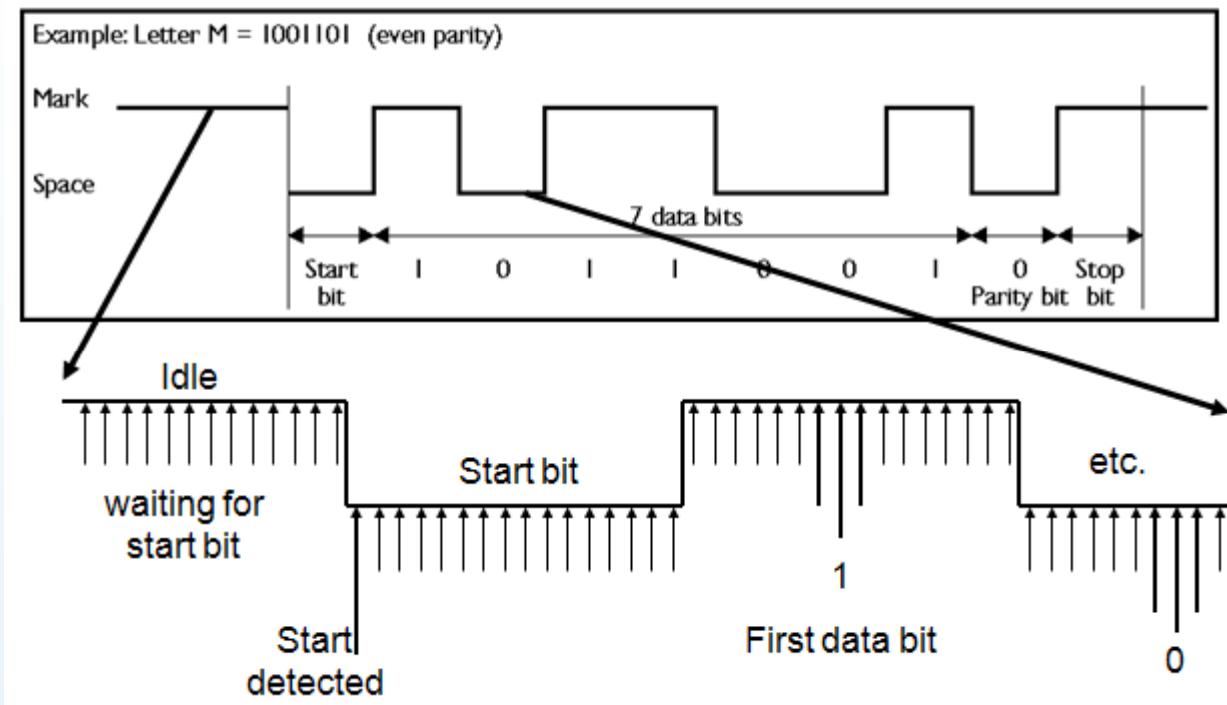


Fig. Asynchronous Serial Reception

SPI INTERFACING

- ▶ The LPC2148 contain one SPI controller

Key Features

- ▶ Compliant with SPI specification
- ▶ Synchronous, Serial, Full Duplex, Communication
- ▶ Combined SPI master and slave
- ▶ Maximum data bit rate of one eighth of the input clock rate

Master Operation

1. Set the SPI clock counter register to the desired clock rate.
2. Set the SPI control register to the master mode.
3. Write the data to transmit to the SPI data register. This write starts the SPI data transfer.
4. Wait for the SPIF bit in the SPI status register to be set to 1. The SPIF bit will be set after the last cycle of the SPI data transfer.
5. Read the SPI status register.
6. Read the received data from the SPI data register (optional).
7. Go to step 3 if more data is required to transmit

Slave Operation

1. Set the SPI control register to the slave mode.
2. Wait for the SPIF bit in the SPI status register to be set to 1. The SPIF bit will be set after the last sampling clock edge of the SPI data transfer.
3. Read the SPI status register.
4. Read the received data from the SPI data register.
5. Go to step 2 if more data is required to transmit

UART0 BLOCK DIAGRAM

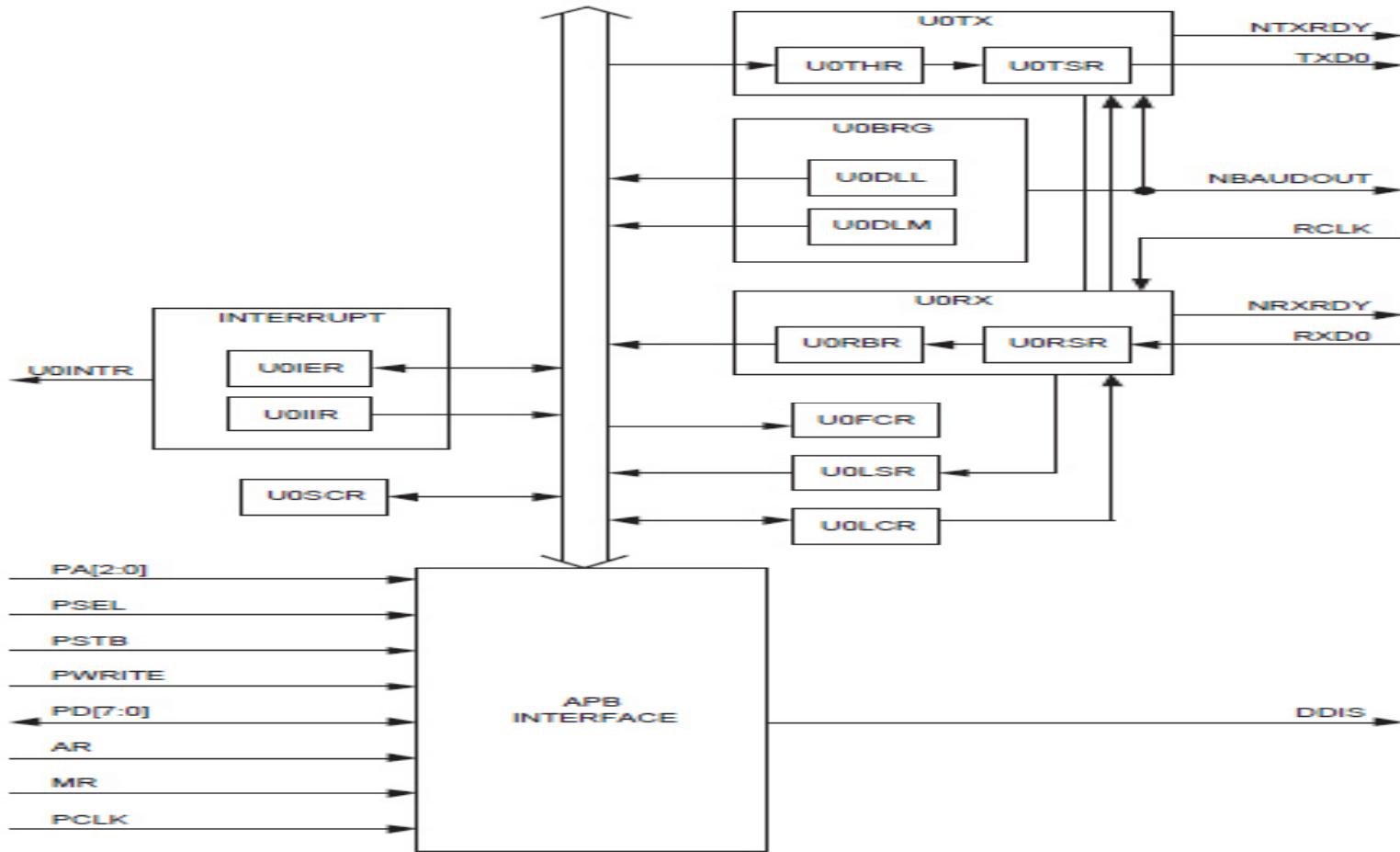
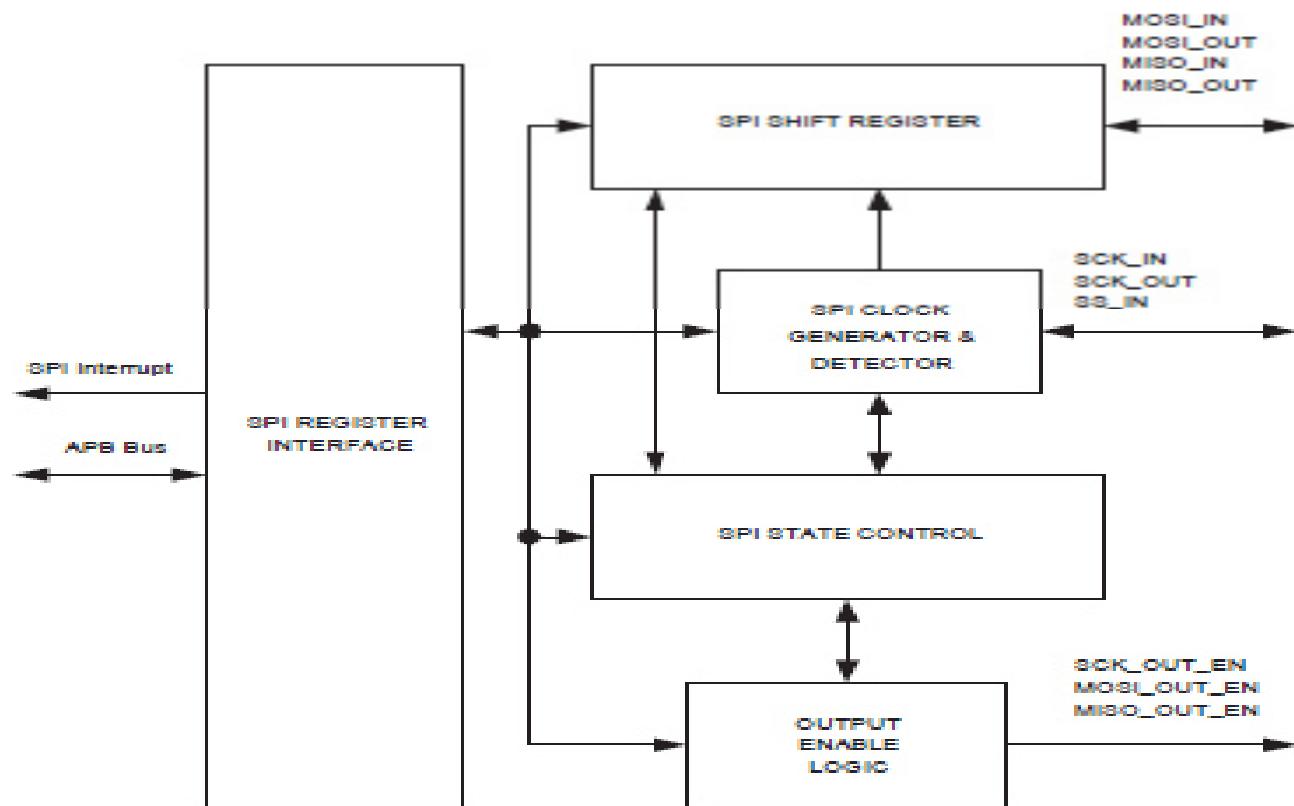


Fig 25. UART0 block diagram

Algorithm for UART

- ▶ Set the U0LCR
- ▶ Set the APBDIV register
- ▶ Set U0DLL and U0DLM register
- ▶ Set U0FCR register
- ▶ Check the status register
- ▶ Set U0FCR register
- ▶ Go to step 5 if more data is required to transmit.

SPI BLOCK DIAGRAM



Algorithm for SPI

- ▶ 1. Set the SPI clock counter register
- ▶ 2. Set the SPI control register
- ▶ 3. Write the data to transmit SPI DATA REGISTER to the SPI data register
- ▶ 4. Wait for the SPIF bit in the SPI status register to be set to 1
- ▶ 5. Read the SPI status register.
- ▶ 6. Read the received data from the SPI data register (optional).
- ▶ 7. Save the data which is read in above step
- ▶ 8. Go to step 3 if more data is required to transmit.

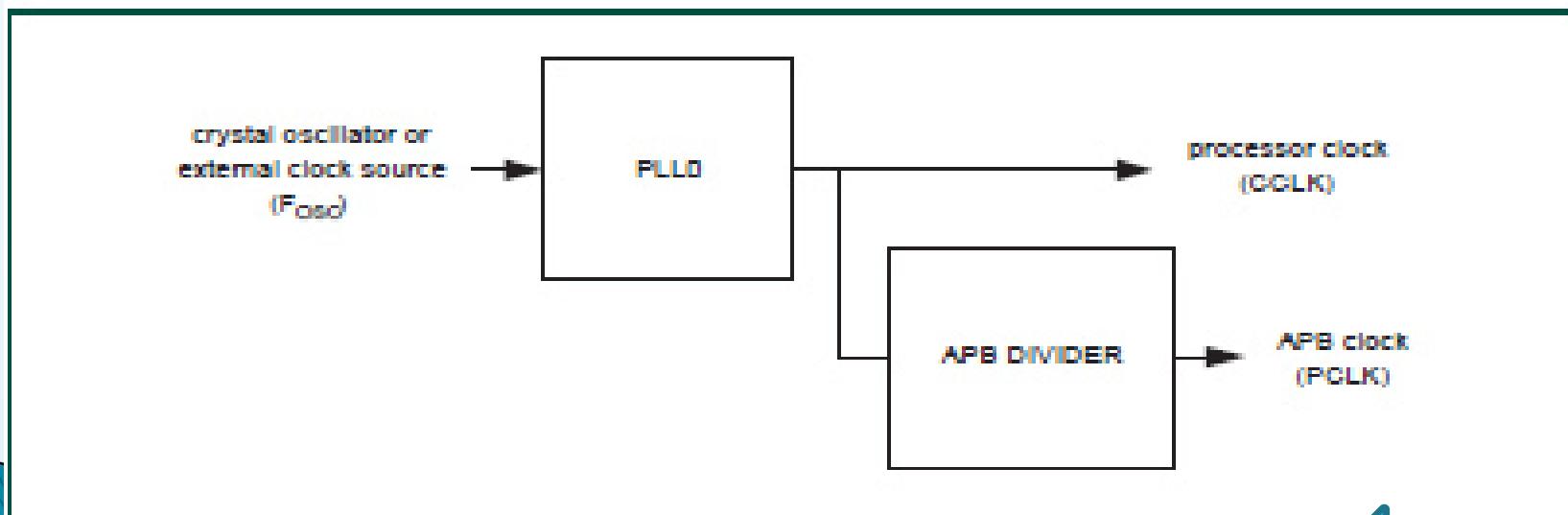


UART0 Line Control Register

Bit	Symbol	Value	Description	Reset value
1:0	Word Length Select	00	5 bit character length	0
		01	6 bit character length	0
		10	7 bit character length	0
		11	8 bit character length	0
2	Stop Bit Select	0	1 stop bit.	0
		1	2 stop bits (1.5 if UOLCR[1:0]=00).	0
3	Parity Enable	0	Disable parity generation and checking.	0
		1	Enable parity generation and checking.	0
5:4	Parity Select	00	Odd parity. Number of 1s in the transmitted character and the attached parity bit will be odd.	0
		01	Even Parity. Number of 1s in the transmitted character and the attached parity bit will be even.	0
		10	Forced "1" stick parity.	0
		11	Forced "0" stick parity.	0
6	Break Control	0	Disable break transmission.	0
		1	Enable break transmission. Output pin UART0 TXD is forced to logic 0 when UOLCR[6] is active high.	0
7	Divisor Latch Access Bit (DLAB)	0	Disable access to Divisor Latches.	0
		1	Enable access to Divisor Latches.	0

APBDIV Register

Bit	Symbol	Value	Description	Reset value
1:0	APBDIV	00	APB bus clock is one fourth of the processor clock.	00
		01	APB bus clock is the same as the processor clock.	
		10	APB bus clock is one half of the processor clock.	
		11	Reserved. If this value is written to the APBDIV register, it has no effect (the previous setting is retained).	
7:2	-	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA



UART0 Divisor Latch LSB/MSB Register

Bit	Symbol	Description	Reset value
7:0	DLL	The UART0 Divisor Latch LSB Register, along with the U0DLM register, determines the baud rate of the UART0.	0x01

Table 165: UART0 Divisor Latch MSB register (U0DLM - address 0xE000 C004, when DLAB = 1) bit description

Bit	Symbol	Description	Reset value
7:0	DLM	The UART0 Divisor Latch MSB Register, along with the U0DLL register, determines the baud rate of the UART0.	0x00

UART0 FIFO Control Register

Bit	Symbol	Value	Description	Reset value
0	FIFO Enable	0	UART0 FIFOs are disabled. Must not be used in the application.	0
		1	Active high enable for both UART0 Rx and TX FIFOs and U0FCR[7:1] access. This bit must be set for proper UART0 operation. Any transition on this bit will automatically clear the UART0 FIFOs.	
1	RX FIFO Reset	0	No impact on either of UART0 FIFOs.	0
		1	Writing a logic 1 to U0FCR[1] will clear all bytes in UART0 Rx FIFO and reset the pointer logic. This bit is self-clearing.	
2	TX FIFO Reset	0	No impact on either of UART0 FIFOs.	0
		1	Writing a logic 1 to U0FCR[2] will clear all bytes in UART0 TX FIFO and reset the pointer logic. This bit is self-clearing.	
5:3	-	0	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA
7:6	RX Trigger Level	00	These two bits determine how many receiver UART0 FIFO characters must be written before an interrupt is activated.	0
		01	Trigger level 0 (1 character or 0x01)	
		10	Trigger level 1 (4 characters or 0x04)	
		11	Trigger level 2 (8 characters or 0x08)	
			Trigger level 3 (14 characters or 0x0E)	

UART0 Line Status Register

Bit	Symbol	Value	Description	Reset value
0	Receiver Data Ready (RDR)	0	U0LSR0 is set when the U0RBR holds an unread character and is cleared when the UART0 RBR FIFO is empty.	0
		1	U0RBR is empty.	
		1	U0RBR contains valid data.	
1	Overrun Error (OE)	0	The overrun error condition is set as soon as it occurs. An U0LSR read clears U0LSR1. U0LSR1 is set when UART0 RSR has a new character assembled and the UART0 RBR FIFO is full. In this case, the UART0 RBR FIFO will not be overwritten and the character in the UART0 RSR will be lost.	0
		1	Overrun error status is inactive.	
		1	Overrun error status is active.	
2	Parity Error (PE)	0	When the parity bit of a received character is in the wrong state, a parity error occurs. An U0LSR read clears U0LSR[2]. Time of parity error detection is dependent on U0FCR[0].	0
		1	Note: A parity error is associated with the character at the top of the UART0 RBR FIFO.	
		0	Parity error status is inactive.	
		1	Parity error status is active.	
3	Framing Error (FE)	0	When the stop bit of a received character is a logic 0, a framing error occurs. An U0LSR read clears U0LSR[3]. The time of the framing error detection is dependent on U0FCR0. Upon detection of a framing error, the Rx will attempt to resynchronize to the data and assume that the bad stop bit is actually an early start bit. However, it cannot be assumed that the next received byte will be correct even if there is no Framing Error.	0
		1	Note: A framing error is associated with the character at the top of the UART0 RBR FIFO.	
		0	Framing error status is inactive.	
		1	Framing error status is active.	

4	Break Interrupt (BI)	When RXD0 is held in the spacing state (all 0's) for one full character transmission (start, data, parity, stop), a break interrupt occurs. Once the break condition has been detected, the receiver goes idle until RXD0 goes to marking state (all 1's). An U0LSR read clears this status bit. The time of break detection is dependent on U0FCR[0]. Note: The break interrupt is associated with the character at the top of the UART0 RBR FIFO.	0
		0 Break interrupt status is inactive.	
		1 Break interrupt status is active.	
5	Transmitter Holding Register Empty (THRE))	THRE is set immediately upon detection of an empty UART0 THR and is cleared on a U0THR write.	1
		0 U0THR contains valid data.	
		1 U0THR is empty.	
6	Transmitter Empty (TEMT)	TEMT is set when both U0THR and U0TSR are empty; TEMT is cleared when either the U0TSR or the U0THR contain valid data.	1
		0 U0THR and/or the U0TSR contains valid data.	
		1 U0THR and the U0TSR are empty.	
7	Error in RX FIFO (RXFE)	U0LSR[7] is set when a character with a Rx error such as framing error, parity error or break interrupt, is loaded into the U0RBR. This bit is cleared when the U0LSR register is read and there are no subsequent errors in the UART0 FIFO.	0
		0 U0RBR contains no UART0 RX errors or U0FCR[0]=0.	
		1 UART0 RBR contains at least one UART0 RX error.	

SPI Clock Counter Register

Bit	Symbol	Description	Reset value
7:0	Counter	SPI0 Clock counter setting.	0x00

SPI Control Register

Bit	Symbol	Value	Description	Reset value
1:0	-		Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA
2	BitEnable	0	The SPI controller sends and receives 8 bits of data per transfer.	0
		1	The SPI controller sends and receives the number of bits selected by bits 11:8.	
3	CPHA		Clock phase control determines the relationship between the data and the clock on SPI transfers, and controls when a slave transfer is defined as starting and ending.	0
		0	Data is sampled on the first clock edge of SCK. A transfer starts and ends with activation and deactivation of the SSEL signal.	
		1	Data is sampled on the second clock edge of the SCK. A transfer starts with the first clock edge, and ends with the last sampling edge when the SSEL signal is active.	
4	CPOL		Clock polarity control.	0
		0	SCK is active high.	
5	MSTR		SCK is active low.	
		0	Master mode select.	0
		1	The SPI operates in Slave mode.	
			The SPI operates in Master mode.	

CONTD...

Bit	Symbol	Value	Description	Reset value
6	LSBF		LSB First controls which direction each byte is shifted when transferred.	0
		0	SPI data is transferred MSB (bit 7) first.	
		1	SPI data is transferred LSB (bit 0) first.	
7	SPIE		Serial peripheral interrupt enable.	0
		0	SPI interrupts are inhibited.	
		1	A hardware interrupt is generated each time the SPIF or WCOL bits are activated.	
11:8	BITS		When bit 2 of this register is 1, this field controls the number of bits per transfer.	0000
		1000	8 bits per transfer	
		1001	9 bits per transfer	
		1010	10 bits per transfer	
		1011	11 bits per transfer	
		1100	12 bits per transfer	
		1101	13 bits per transfer	
		1110	14 bits per transfer	
		1111	15 bits per transfer	
		0000	16 bits per transfer	
15:12	-		Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

SPI Data Register

Bit	Symbol	Description	Reset value
7:0	DataLow	SPI Bi-directional data port.	0x00
15:8	DataHigh	If bit 2 of the SPCR is 1 and bits 11:8 are other than 1000, some or all of these bits contain the additional transmit and receive bits. When less than 16 bits are selected, the more significant among these bits read as zeroes.	0x00

SPI Status Register

Bit	Symbol	Description	Reset value
2:0	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA
3	ABRT	Slave abort. When 1, this bit indicates that a slave abort has occurred. This bit is cleared by reading this register.	0
4	MODF	Mode fault. when 1, this bit indicates that a Mode fault error has occurred. This bit is cleared by reading this register, then writing the SPI control register.	0
5	ROVR	Read overrun. When 1, this bit indicates that a read overrun has occurred. This bit is cleared by reading this register.	0
6	WCOL	Write collision. When 1, this bit indicates that a write collision has occurred. This bit is cleared by reading this register, then accessing the SPI data register.	0
7	SPIF	SPI transfer complete flag. When 1, this bit indicates when a SPI data transfer is complete. When a master, this bit is set at the end of the last cycle of the transfer. When a slave, this bit is set on the last data sampling edge of the SCK. This bit is cleared by first reading this register, then accessing the SPI data register. Note: this is not the SPI interrupt flag. This flag is found in the SPINT register.	0

