Analog Devices Inc./Maxim Integrated - 73S1209F-44M/F/PD Datasheet

Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Obsolete
Core Processor	80515
Core Size	8-Bit
Speed	24MHz
Connectivity	I²C, SmartCard, UART/USART
Peripherals	LED, POR, WDT
Number of I/O	9
Program Memory Size	32KB (32K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	44-VFQFN Exposed Pad
Supplier Device Package	44-QFN (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/analog-devices/73s1209f-44m-f-pd

Pin Name	Pin (68 QFN)	Pin (44 QFN)	Type	Equivalent Circuit*	Description
VDD	28 42 65	18 27 44	I		General positive power supply pins. All digital IO is referred to this supply voltage. There is an on-chip regulator that uses VDD to provide power for internal circuits (VPD). A 0.1μF capacitor is recommended at each VDD pin.
N/C	2 4 7 8 26 27 39 46	16 17 29			No connect.
GND	9 25 44	7 15	GND		General ground supply pins for all IO and logic circuits.
RESET	66	1	I	Figure 31	Reset input, positive assertion. Resets logic and registers to default condition.

* See the figures in the Equivalent Circuits section.

1.5 Special Function Registers (SFRs)

The 73S1209F utilizes numerous SFRs to communicate with the 73S1209F s many peripherals. This results in the need for more SFR locations outside the direct address IRAM space (0x80 to 0xFF). While some peripherals are mapped to unused IRAM SFR locations, additional SFRs for the smart card and other peripheral functions are mapped to the top of the XRAM data space (0xFC00 to 0xFFFF).

1.5.1 Internal Data Special Function Registers (SFRs)

A map of the Special Function Registers is shown in Table 6.

Table 6: IRAM Special Function Registers Locations

Hex\ Bin	X000	X001	X010	X011	X100	X101	X110	X111	Bin/ Hex
F8									FF
F0	B								F7
E8									EF
E0	A								E7
D8	BRCON								DF
D0	PSW	**KCOL**	**KROW**	**KSCAN**	**KSTAT**	**KSIZE**	**KORDERL**	**KORDERH**	D7
C8	T2CON								CF
C0	IRCON								C7
B8	IEN1	IP1	S0RELH	S1RELH					BF
B0			**FLSHCTL**					**PGADDR**	B7
A8	IEN0	IP0	S0RELL						AF
A0	**USR8**	**UDIR8**							A7
98	S0CON	S0BUF	IEN2	S1CON	S1BUF	S1RELL			9F
90	**USR70**	**UDIR70**	DPS		**ERASE**				97
88	TCON	TMOD	TL0	TL1	TH0	TH1		**MCLKCtl**	8F
80		SP	DPL	DPH	DPL1	DPH1	WDTREL	PCON	87

Only a few addresses are used, the others are not implemented. SFRs specific to the 73S1209F are shown in **bold** print (gray background). Any read access to unimplemented addresses will return undefined data, while most write access will have no effect. However, a few locations are reserved and not user configurable in the 73S1209F. **Writes to the unused SFR locations can affect the operation of the core and therefore must not be written to. This applies to all the SFR areas in both the IRAM and XRAM spaces. In addition, all unused bit locations within valid SFR registers must be left in their default (power on default) states.**

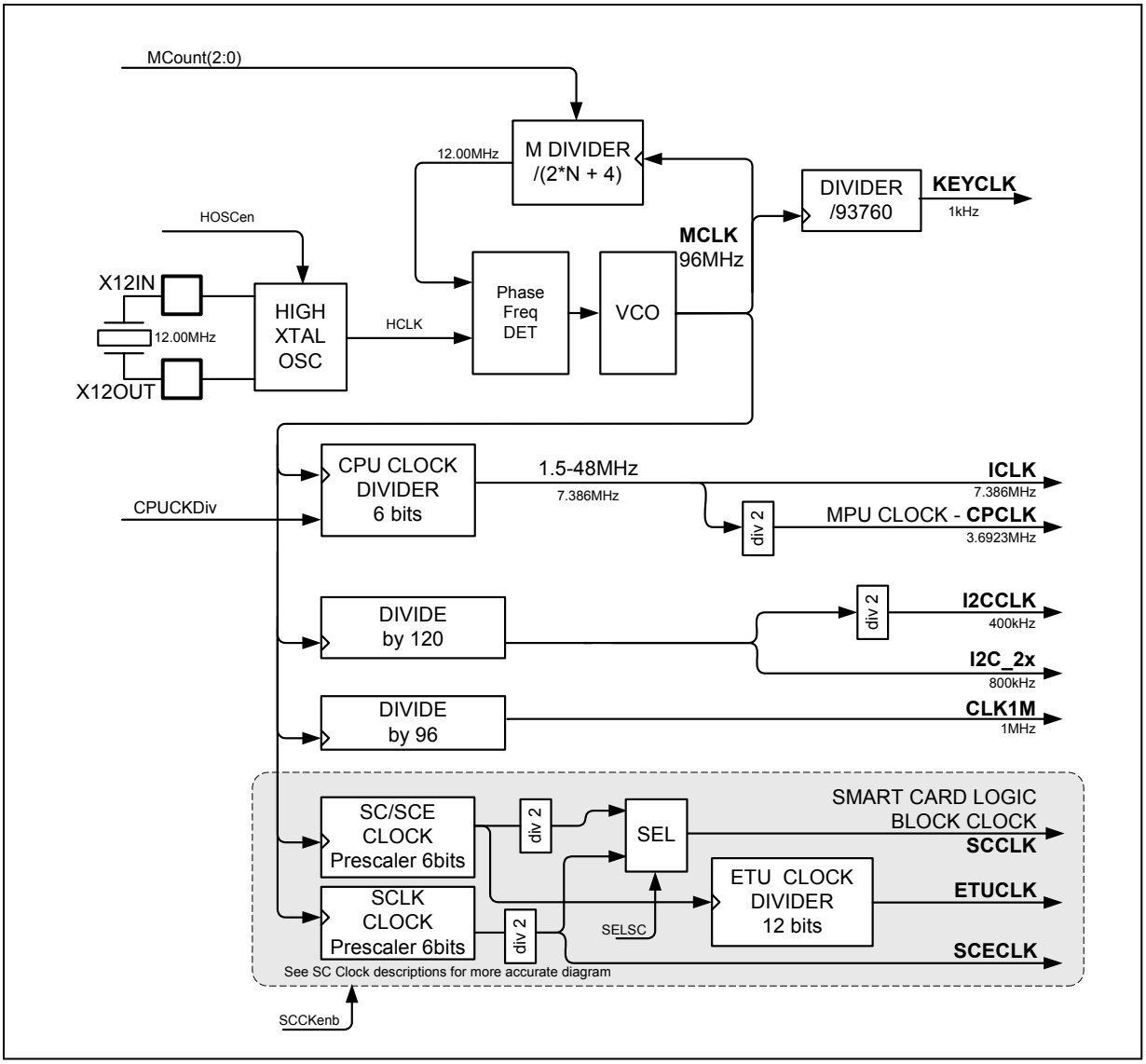


Figure 3: Clock Generation and Control Circuits

1.7.3.1 Interrupt Overview

When an interrupt occurs, the MPU will vector to the predetermined address as shown in Table 32. Once the interrupt service has begun, it can only be interrupted by a higher priority interrupt. The interrupt service is terminated by a return from the RETI instruction. When a RETI is performed, the processor will return to the instruction that would have been next when the interrupt occurred.

When the interrupt condition occurs, the processor will also indicate this by setting a flag bit. This bit is set regardless of whether the interrupt is enabled or disabled. Each interrupt flag is sampled once per machine cycle, then samples are polled by the hardware. If the sample indicates a pending interrupt when the interrupt is enabled, then the interrupt request flag is set. On the next instruction cycle, the interrupt will be acknowledged by hardware forcing an LCALL to the appropriate vector address.

Interrupt response will require a varying amount of time depending on the state of the MPU when the interrupt occurs. If the MPU is performing an interrupt service with equal or greater priority, the new interrupt will not be invoked. In other cases, the response time depends on the current instruction. The fastest possible response to an interrupt is 7 machine cycles. This includes one machine cycle for detecting the interrupt and six cycles to perform the LCALL.

1.7.3.2 Special Function Registers for Interrupts

Interrupt Enable 0 Register (IEN0): 0xA8 ← 0x00

Table 19: The IEN0 Register

MSB LSB

EAL	WDT	–	ES0	ET1	EX1	ET0	EX0

Bit	Symbol	Function
IEN0.7	EAL	EAL = 0 – disable all interrupts.
IEN0.6	WDT	Not used for interrupt control.
IEN0.5	–	
IEN0.4	ES0	ES0 = 0 – disable serial channel 0 interrupt.
IEN0.3	ET1	ET1 = 0 – disable timer 1 overflow interrupt.
IEN0.2	EX1	EX1 = 0 – disable external interrupt 1.
IEN0.1	ET0	ET0 = 0 – disable timer 0 overflow interrupt.
IEN0.0	EX0	EX0 = 0 – disable external interrupt 0.

Timer/Counter Control Register (TCON): 0x88 ← 0x00

Table 43: The TCON Register

MSB LSB

TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

Bit	Symbol	Function
TCON.7	TF1	The Timer 1 overflow flag is set by hardware when Timer 1 overflows. This flag can be cleared by software and is automatically cleared when an interrupt is processed.
TCON.6	TR1	Timer 1 Run control bit. If cleared, Timer 1 stops.
TCON.5	TF0	Timer 0 overflow flag set by hardware when Timer 0 overflows. This flag can be cleared by software and is automatically cleared when an interrupt is processed.
TCON.4	TR0	Timer 0 Run control bit. If cleared, Timer 0 stops.
TCON.3	IE1	External Interrupt 1 edge flag.
TCON.2	IT1	External interrupt 1 type control bit.
TCON.1	IE0	External Interrupt 0 edge flag.
TCON.0	IT0	External Interrupt 0 type control bit.

1.7.6 WD Timer (Software Watchdog Timer)

The software watchdog timer is a 16-bit counter that is incremented once every 24 or 384 clock cycles. After a reset, the watchdog timer is disabled and all registers are set to zero. The watchdog consists of a 16-bit counter (WDT), a reload register (WDTREL), prescalers (by 2 and by 16), and control logic. Once the watchdog starts, it cannot be stopped unless the internal reset signal becomes active.

WD Timer Start Procedure: The WDT is started by setting the SWDT flag. When the WDT register enters the state 0x7CFF, an asynchronous WDTS signal will become active. The signal WDTS sets bit 6 in the IP0 register and requests a reset state. WDTS is cleared either by the reset signal or by changing the state of the WDT timer.

Refreshing the WD Timer: The watchdog timer must be refreshed regularly to prevent the reset request signal from becoming active. This requirement imposes an obligation on the programmer to issue two instructions. The first instruction sets WDT and the second instruction sets SWDT. The maximum delay allowed between setting WDT and SWDT is 12 clock cycles. If this period has expired and SWDT has not been set, WDT is automatically reset, otherwise the watchdog timer is reloaded with the content of the WDTREL register and WDT is automatically reset.

Interrupt Priority 0 Register (IP0): 0xA9 ← 0x00

Table 46: The IP0 Register

MSB							LSB
–	WDTS	IP0.5	IP0.4	IP0.3	IP0.2	IP0.1	IP0.0

Bit	Symbol	Function
IP0.6	WDTS	Watchdog timer status flag. Set when the watchdog timer has expired. The internal reset will be generated, but this bit will not be cleared by the reset. This allows the user program to determine if the watchdog timer caused the reset to occur and respond accordingly. Can be read and cleared by software.

Note: The remaining bits in the IP0 register are not used for watchdog control.

Watchdog Timer Reload Register (WDTREL): 0x86 ← 0x00

Table 47: The WDTREL Register

MSB							LSB
WDPSEL	WDREL6	WDREL5	WDREL4	WDREL3	WDREL2	WDREL1	WDREL0

Bit	Symbol	Function
WDTREL.7	WDPSEL	Prescaler select bit. When set, the watchdog is clocked through an additional divide-by-16 prescaler.
WDTREL.6 to WDTREL.0	WDREL6-0	Seven bit reload value for the high-byte of the watchdog timer. This value is loaded to the WDT when a refresh is triggered by a consecutive setting of bits WDT and SWDT.

External Interrupt Control Register (USRIntCtl1) : 0xFF90 ← 0x00

Table 51: The USRIntCtl1 Register

MSB							LSB
–	U1IS.6	U1IS.5	U1IS.4	–	U0IS.2	U0IS.1	U0IS.0

External Interrupt Control Register (USRIntCtl2) : 0xFF91 ← 0x00

Table 52: The USRIntCtl2 Register

MSB							LSB
–	U3IS.6	U3IS.5	U3IS.4	–	U2IS.2	U2IS.1	U2IS.0

External Interrupt Control Register (USRIntCtl3) : 0xFF92 ← 0x00

Table 53: The USRIntCtl3 Register

MSB							LSB
–	U5IS.6	U5IS.5	U5IS.4	–	U4IS.2	U4IS.1	U4IS.0

External Interrupt Control Register (USRIntCtl4) : 0xFF93 ← 0x00

Table 54: The USRIntCtl4 Register

MSB							LSB
–	U7IS.6	U7IS.5	U7IS.4	–	U6IS.2	U6IS.1	U6IS.0

I2C Secondary Write Data Register (SWDR): 0XFF82 ← 0x00

Table 60: The SWDR Register

MSB							LSB
SWDR.7	SWDR.6	SWDR.5	SWDR.4	SWDR.3	SWDR.2	SWDR.1	SWDR.0

Bit	Function
SWDR.7	Second Data byte to be written to the I^2C slave device if bit 0 (I2CLEN) of the Control and Status register (CSR) is set = 1.
SWDR.6	
SWDR.5	
SWDR.4	
SWDR.3	
SWDR.2	
SWDR.1	
SWDR.0	

I2C Read Data Register (RDR): 0XFF83 ← 0x00

Table 61: The RDR Register

MSB							LSB
RDR.7	RDR.6	RDR.5	RDR.4	RDR.3	RDR.2	RDR.1	RDR.0

Bit	Function
RDR.7	Data read from the I^2C slave device.
RDR.6	
RDR.5	
RDR.4	
RDR.3	
RDR.2	
RDR.1	
RDR.0	

External Interrupt Control Register (INT6Ctl): 0xFF95 ← 0x00

Table 64: The INT6Ctl Register

MSB LSB

–	–	VFTIEN	VFTINT	I2CIEN	I2CINT	ANIEN	ANINT

Bit	Symbol	Function
INT6Ctl.7	–	
INT6Ctl.6	–	
INT6Ctl.5	VFTIEN	VDD fault interrupt enable.
INT6Ctl.4	VFTINT	VDD fault interrupt flag.
INT6Ctl.3	I2CIEN	When set = 1, the I^2C interrupt is enabled.
INT6Ctl.2	I2CINT	When set =1, the I^2C transaction has completed. Cleared upon the start of a subsequent I^2C transaction.
INT6Ctl.1	ANIEN	Analog compare interrupt enable.
INT6Ctl.0	ANINT	Analog compare interrupt flag.

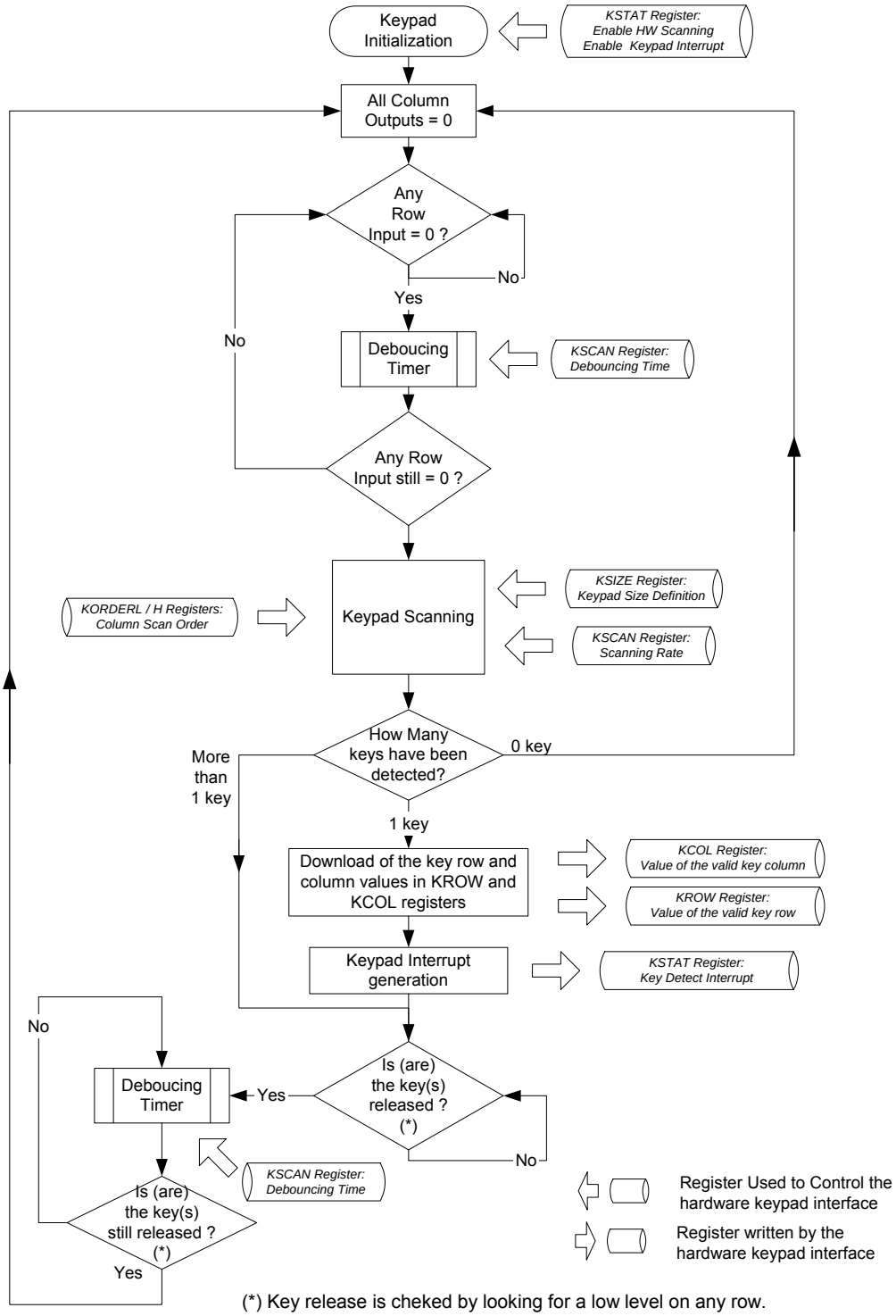


Figure 12: Keypad Interface Flow Chart

on a card insertion / removal to allow power saving modes. Card insertion / removal is generated from the respective card switch detection inputs (whose polarity is programmable).

The built-in ICC Interface has a low dropout regulator (V_{CC} generator) capable of driving 1.8, 3.0 and 5.0V smart cards in accordance with the ISO-7816-3 and EMV4.0 standards. This converter requires a separate 5.0V input supply source designated as VPC. Auxiliary I/O lines C4 and C8 are only provided for the built-in interface. If support for the auxiliary lines is necessary for the external interfaces, they need to be handled manually through the USR GPIO pins. The external 8010 devices directly connect the I/O (SIO) and clock (SCLK) signals and control is handled via the I^2C interface.

Figure 14 shows how multiple 8010 devices can be connected to the 73S1209F.

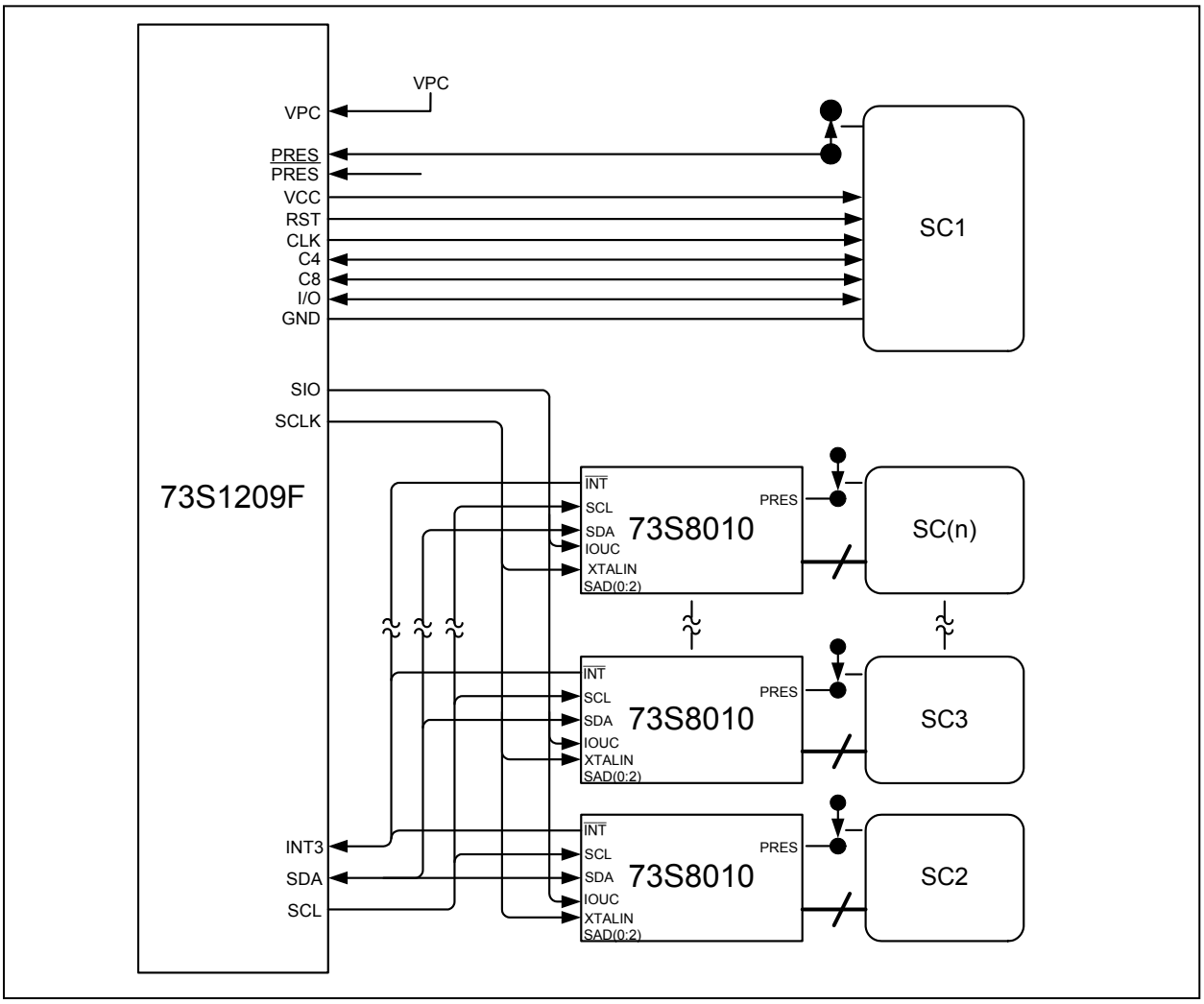


Figure 14: External Smart Card Interface Block Diagram

Block Guard Time register (BGT). Other than the protocol checks described above, the firmware is responsible for all protocol checking and error recovery.

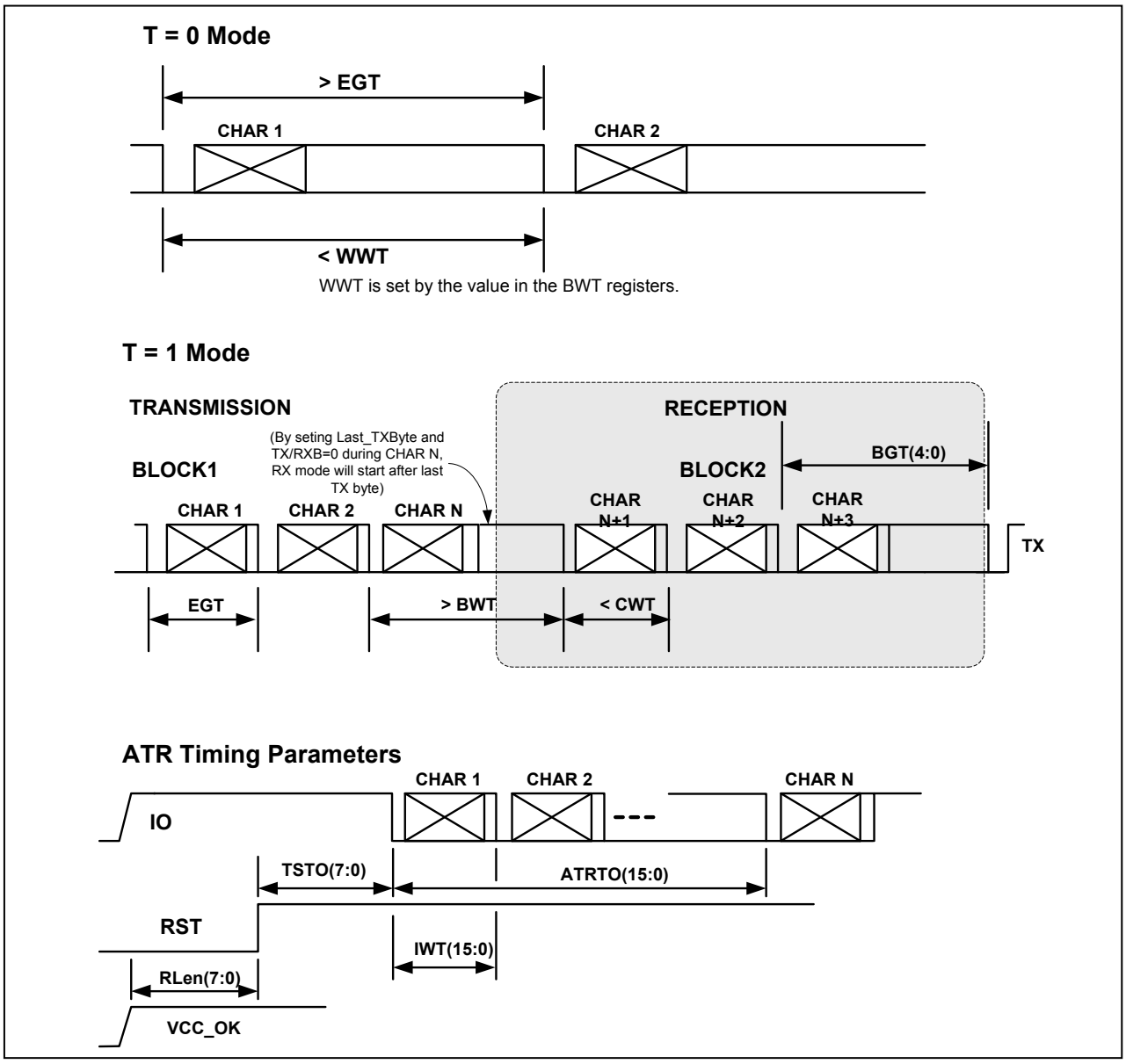


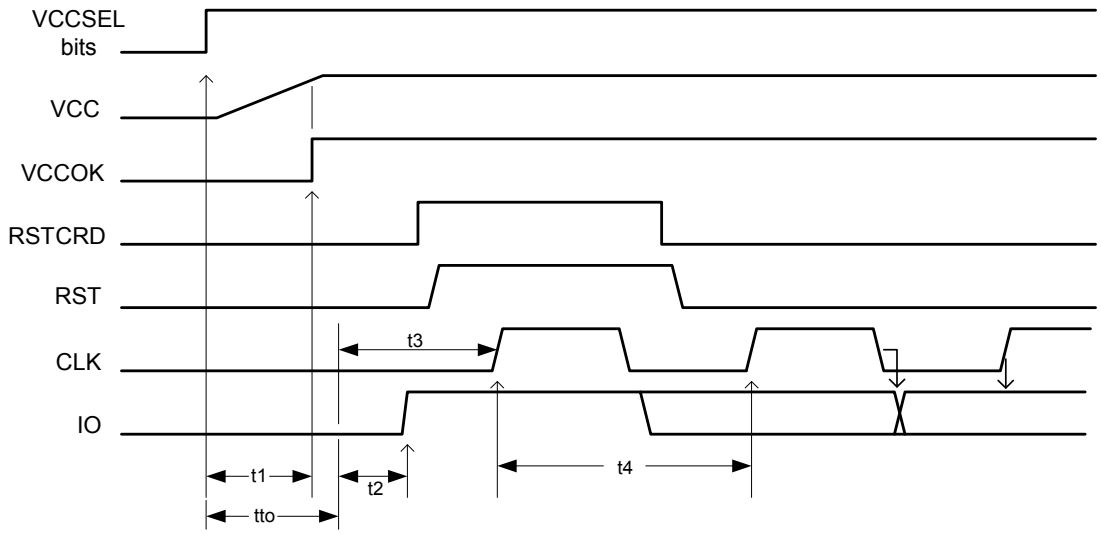
Figure 18: Guard, Block, Wait and ATR Time Definitions

1.7.13.4 Bypass Mode

It is possible to bypass the smart card UART in order for the firmware to support non-T=0/T=1 smart cards. This is called Bypass mode. In this mode the embedded firmware will communicate directly with the selected smart card and drive I/O during transmit and read I/O during receive in order to communicate with the smart card. In this mode, ATR processing is under firmware control. The firmware must sequence the interface signals as required. Firmware must perform TS processing, parity checking, break generation and CRC/LRC calculation (if required).

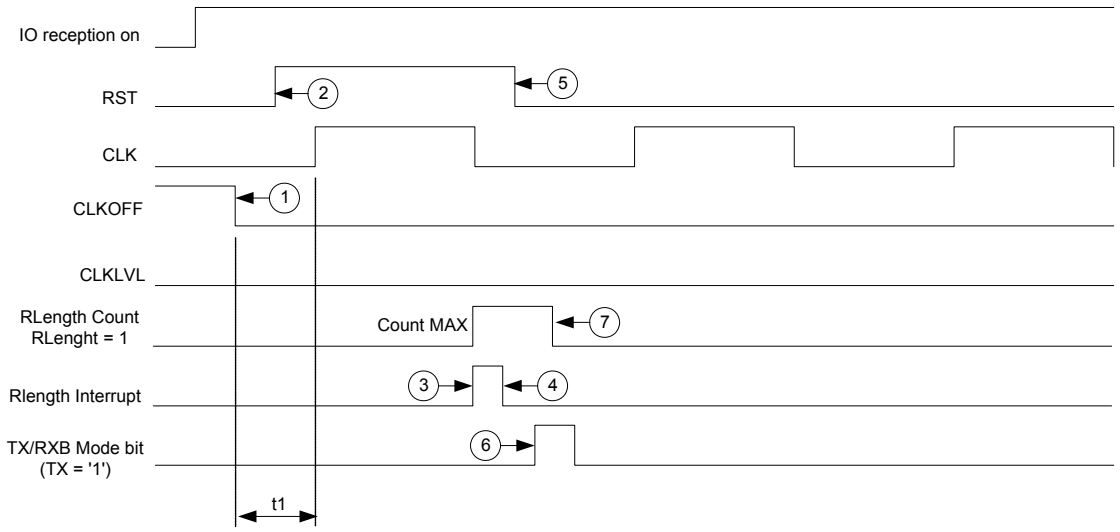
1.7.13.5 Synchronous Operation Mode

The 73S1209F supports synchronous operation. When sync mode is selected for either interface, the CLK signal is generated by the ETU counter. The values in FDReg, SCCLK, and SCECLK must be set to obtain the desired sync CLK rate. There is only one ETU counter and therefore, in sync mode, the interface must



t1: The time from setting VCCSEL bits until VCCOK = 1.
tto: The time from setting VCCSEL bits until VCCTMR times out. At t1 (if RDYST = 1) or tto (if RDYST = 0), activation starts. It is suggested to have RDYST = 0 and use the VCCTMR interrupt to let MPU know when sequence is starting.
t2: time from start of activation (no external indication) until IO goes into reception mode (= 1). This is approximately 4 SCCLK (or SCECLK) clock cycles.
t3: minimum one half of ETU period.
t4: ETU period.
Note that in Sync mode, IO as input is sampled on the rising edge of CLK. IO changes on the falling edge of CLK, either from the card or from the 73S1209F. The RST signal to the card is directly controlled by the RSTCRD bit (non-inverted) via the MPU and is shown as an example of a possible RST pattern.

Figure 19: Synchronous Activation



1. Clear CLKOFF after Card is in reception mode.
2. Set RST bit.
3. Interrupt is generated when Rlength counter is MAX.
4. Read and clear Interrupt.
5. Clear RST bit.
6. Toggle TX/RXB to reset bit counter.
7. Reload RLength Counter.

t1. CLK wll start at least 1/2 ETU after CLKOFF is set low when CLKLVL = 0

Figure 20: Example of Sync Mode Operation: Generating/Reading ATR Signals

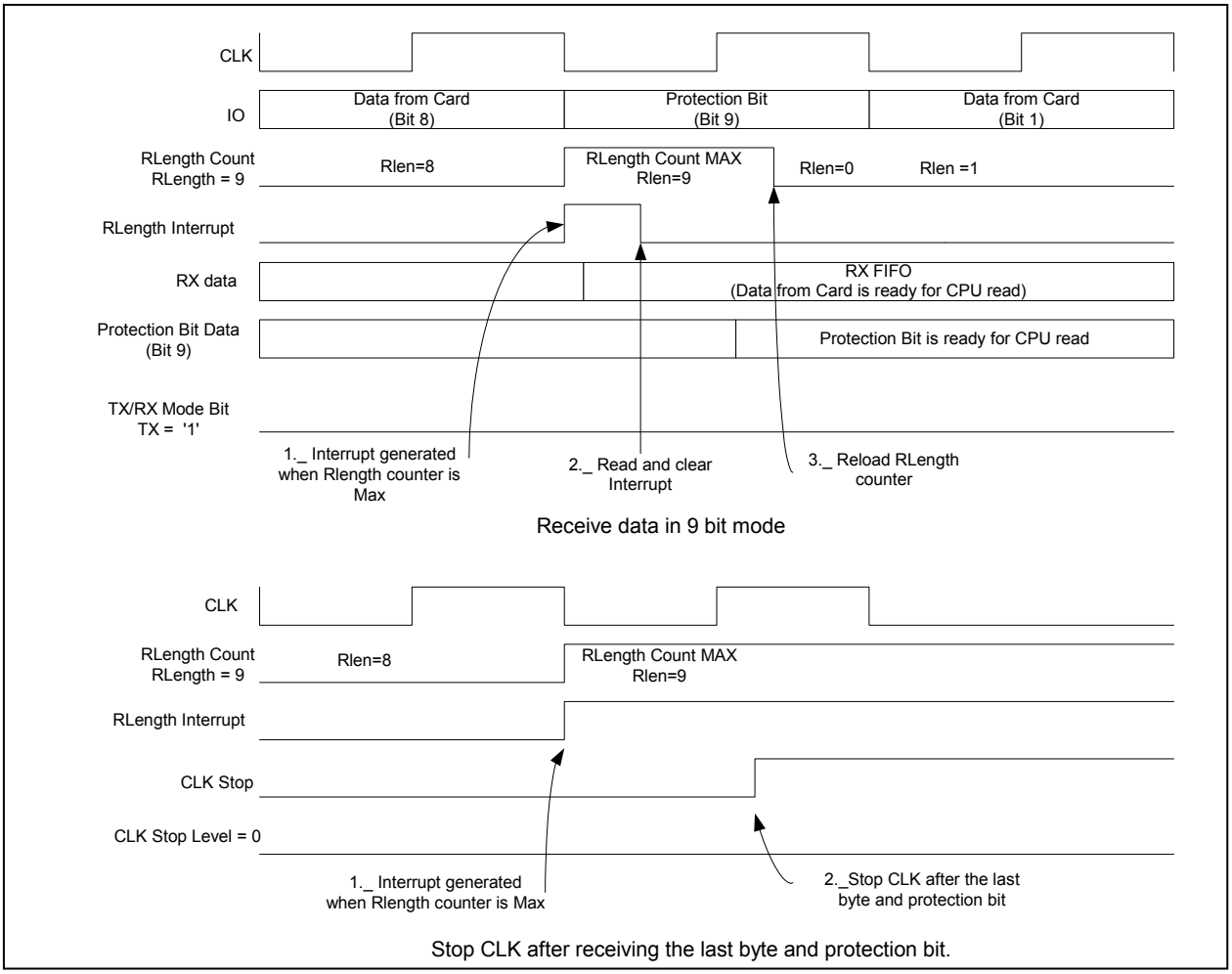


Figure 23: Operation of 9-bit Mode in Sync Mode

Synchronous card operation is broken down into three primary types. These are commonly referred to as 2-wire, 3-wire and I2C synchronous cards. Each card type requires different control and timing and therefore requires different algorithms to access. Teridian has created an application note to provide detailed algorithms for each card type. Refer to the application note titled *73S12xxF Synchronous Card Design Application Note*.

Smart Card Interrupt Register (SCInt): 0xFE01 ← 0x00

When the smart card interrupt is asserted, the firmware can read this register to determine the actual cause of the interrupt. The bits are cleared when this register is read. Each interrupt can be disabled by the Smart Card Interrupt Enable register. Error processing must be handled by the firmware. This register relates to the interface that is active – see the SCSel register (above).

Table 74: The SCInt Register

MSB LSB

WAITTO	CRDEVT	VCCTMRI	RXDAV	TXEVT	TXSENT	TXERR	RXERR

Bit	Symbol	Function
SCInt.7	WAITTO	Wait Timeout – An ATR or card wait timeout has occurred. In sync mode, this interrupt is asserted when the RLen counter (it advances on falling edges of CLK/ETU) reaches the loaded (max) value. This bit is cleared when the SCInt register is read. When running in Synchronous Clock Stop Mode, this bit becomes RLenINT interrupt (set when the Rlen counter reaches the terminal count).
SCInt.6	CRDEVT	Card Event – A card event is signaled via pin DETCARD either when the Card was inserted or removed (read the CRDCtl register to determine card presence) or there was a fault condition in the interface circuitry. This bit is functional even if the smart card logic clock is disabled and when the PWRDN bit is set. This bit is cleared when the SCInt register is read.
SCInt.5	VCCTMRI	VCC Timer – This bit is set when the VCCTMR times out. This bit is cleared when the SCInt register is read.
SCInt.4	RXDAV	Rx Data Available – Data was received from the smart card because the Rx FIFO is not empty. In bypass mode, this interrupt is generated on a falling edge of the smart card I/O line. After receiving this interrupt in bypass mode, firmware should disable it until the firmware has received the entire byte and is waiting for the next start delimiter. This bit is cleared when there is no RX data available in the RX FIFO.
SCInt.3	TXEVNT	TX Event – Set whenever the TXEMTY or TXFULL bits are set in the SRXCtl SFR. This bit is cleared when the STXCtl register is read.
SCInt.2	TXSENT	TX Sent – Set whenever the ISO UART has successfully transmitted a byte to the smart card. Also set when a CRC/LRC byte is sent in T=1 mode. Will not be set in T=0 when a break is detected at the end of a byte (when break detection is enabled). This bit is cleared when the SCInt register is read.
SCInt.1	TXERR	TX Error – An error was detected during the transmission of data to the smart card as indicated by either BREAKD or TXUNDR bit being set in the STXCtl SFR. Additional information can be found in that register description. This bit is cleared when the STXCtl register is read.
SCInt.0	RXERR	RX Error – An error was detected during the reception of data from the smart card. Additional information can be found in the SRXCtl register. This interrupt will be asserted for RXOVRR, or RX Parity error events. This bit is cleared when the SRXCtl register is read.

Smart Card Interrupt Enable Register (SCIE): 0xFE02 ← 0x00

When set to a 1, the respective condition can cause a smart card interrupt. When set to a 0, the respective condition cannot cause an interrupt. When disabled, the respective bit in the Smart Card Interrupt register can still be set, but it will not interrupt the MPU.

Table 75: The SCIE Register

MSB LSB

WTOIEN	CDEVEN	VTMREN	RXDAEN	TXEVEN	TXSNTEN	TXEREN	RXEREN

Bit	Symbol	Function
SCIE.7	WTOIEN	Wait Timeout Interrupt Enable – Enable for ATR or Wait Timeout Interrupt. In sync mode, function is RLIEN (RLen = max.) interrupt enable.
SCIE.6	CDEVEN	Card Event Interrupt Enable.
SCIE.5	VTMREN	VCC Timer Interrupt Enable.
SCIE.4	RXDAEN	Rx Data Available Interrupt Enable.
SCIE.3	TXEVEN	TX Event Interrupt Enable.
SCIE.2	TXSNTEN	TX Sent Interrupt Enable.
SCIE.1	TXEREN	TX Error Interrupt Enable.
SCIE.0	RXEREN	RX Error Interrupt Enable.

TX Control/Status Register (STXCtl): 0xFE06 ← 0x00

This register is used to control transmission of data to the smart card. Some control and some status bits are in this register.

Table 79: The STXCtl Register

MSB							LSB
SYCKST	–	TXFULL	TXEMTY	TXUNDR	LASTTX	TX/RXB	BREAKD

Bit	Symbol	Function
STXCtl.7	I2CMODE	I2C Mode – When in sync mode and this bit is set, and when the RLen count value = max or 0, the source of the smart card data for IO pin (or SIO pin) will be connected to the IO bit in SCCtl (or SCECtl) register rather than the TX FIFO. See the description for the Protocol Mode Register for more detail.
STXCtl.6	–	
STXCtl.5	TXFULL	TX FIFO is full. Additional writes may corrupt the contents of the FIFO. This bit it will remain set as long as the TX FIFO is full. Generates TX_Event interrupt upon going full.
STXCtl.4	TXEMTY	1 = TX FIFO is empty, 0 = TX FIFO is not empty. If there is data in the TX FIFO, the circuit will transmit it to the smart card if in transmit mode. In T=1 mode, if the LASTTX bit is set and the hardware is configured to transmit the CRC/LRC, the TXEMTY will not be set until the CRC/LRC is transmitted. In T=0, if the LASTTX bit is set, TXEMTY will be set after the last word has been successfully transmitted to the smart card. Generates TXEVNT interrupt upon going empty.
STXCtl.3	TXUNDR	TX Underrrun – (Read only) Asserted when a transmit under-run condition has occurred. An under-run condition is defined as an empty TX FIFO when the last data word has been successfully transmitted to the smart card and the LASTTX bit was not set. No special processing is performed by the hardware if this condition occurs. Cleared when read by firmware. This bit generates TXERR interrupt.
STXCtl.2	LASTTX	Last TX Byte – Set by firmware (in both T=0 and T=1) when the last byte in the current message has been written into the transmit FIFO. In T=1 mode, the CRC/LRC will be appended to the message. Should be set after the last byte has been written into the transmit FIFO. Should be cleared by firmware before writing first byte of next message into the transmit FIFO. Used in T=0 to determine when to set TXEMTY.
STXCtl.1	TX/RXB	1 = Transmit mode, 0 = Receive mode. Configures the hardware to be receiving from or transmitting to the smart card. Determines which counters should be enabled. This bit should be set to receive mode prior to switching to another interface. Setting and resetting this bit shall initialize the CRC logic. If LASTTX is set, this bit can be reset to RX mode and UART logic will automatically change mode to RX when TX operation is completed (TX_Empty =1).
STXCtl.0	BREAKD	Break Detected – (Read only) 1 = A break has been detected on the I/O line indicating that the smart card detected a parity error. Cleared when read. This bit generates TXERR interrupt.

Smart Card Control Register (SCCtl): 0xFE0A ← 0x21

This register is used to monitor reception of data from the smart card.

Table 83: The SCCtl Register

MSB							LSB
RSTCRD	–	IO	IOD	C8	C4	CLKLVL	CLKOFF

Bit	Symbol	Function
SCCtl.7	RSTCRD	1 = Asserts the RST (set RST = 0) to the smart card interface, 0 = De-assert the RST (set RST = 1) to the smart card interface. Can be used to extend RST to the smart card. Refer to the RLength register description. This bit is operational in all modes and can be used to extend RST during activation or perform a "Warm Reset" as required. In auto-sequence mode, this bit should be set = 0 to allow the sequencer to de-assert RST per the RLength parameters. In sync mode (see the SPrtcol register) the sense of this bit is non-inverted, if set =1 , RST = 1, if set = 0, RST = 0. Rlen has no effect on Reset in sync mode.
SCCtl.6	–	
SCCtl.5	IO	Smart Card I/O. Read is state of I/O signal (Caution, this signal is not synchronized to the MPU clock). In Bypass mode, write value is state of signal on I/O. In sync mode, this bit will contain the value of I/O pin on the latest rising edge of CLK.
SCCtl.4	IOD	Smart Card I/O Direction control Bypass mode or sync mode. 1 = input (default), 0 = output.
SCCtl.3	C8	Smart Card C8. When C8 is an output, the value written to this bit will appear on the C8 line. The value read when C8 is an output is the value stored in the register. When C8 is an input, the value read is the value on the C8 pin (Caution, this signal is not synchronized to the MPU clock). When C8 is an input, the value written will be stored in the register but not presented to the C8 pin.
SCCtl.2	C4	Smart Card C4. When C4 is an output, the value written to this bit will appear on the C4 line. The value read when C4 is an output is the value stored in the register. When C4 is an input, the value read is the value on the C4 pin (Caution, this signal is not synchronized to the MPU clock). When C4 is an input, the value written will be stored in the register but not presented to the C4 pin.
SCCtl.1	CLKLVL	1 = High, 0 = Low. If CLKOFF is set = 1, the CLK to smart card will be at the logic level indicated by this bit. If in bypass mode, this bit directly controls the state of CLK.
SCCtl.0	CLKOFF	0 = CLK is enabled. 1 = CLK is not enabled. When asserted, the CLK will stop at the level selected by CLKLVL. This bit has no effect if in bypass mode.

CRC MS Value Registers (CRCMsB): 0xFE14 ← 0xFF, (CRCLsB): 0xFE15 ← 0xFF

Table 94: The CRCMsB Register

MSB							LSB
CRC.15	CRC.14	CRC.13	CRC.12	CRC.11	CRC.10	CRC.9	CRC.8

Table 95: The CRCLsB Register

MSB							LSB
CRC.7	CRC.6	CRC.5	CRC.4	CRC.3	CRC.2	CRC.1	CRC.0

The 16-bit CRC value forms the TX CRC word in TX mode (write value) and the RX CRC in RX mode (read value). The initial value of CRC to be used when generating a CRC to be transmitted at the end of a message (after the last TX byte is sent) when enabled in T=1 mode. Should be reloaded at the beginning of every message to be transmitted. When using CRC, the both CRC registers should be initialized to FF. When using LRC the CRCLsB Value register should be loaded to 00. When receiving a message, the firmware should load this with the initial value and then read this register to get the final value at the end of the message. These registers need to be reloaded for each new message to be received. When in LRC mode, bits (7:0) are used and bits (15:8) are undefined. During LRC/CRC checking and generation, this register is updated with the current value and can be read to aid in debugging. This information will be transmitted to the smart card using the timing specified by the Guard Time register. When checking CRC/LRC on an incoming message (CRC/LRC is checked against the data and CRC/LRC), the firmware reads the final value after the message has been received and determines if an error occurred (= 0x1D0F (CRC_ no error, else error; = 0 (LRC) no error, else error). When a message is received, the CRC/LRC is stored in the FIFO. The polynomial used to generate and check CRC is $x^{16} + x^{12} + x^5 + 1$. When in indirect convention, the CRC is generated prior to the conversion into indirect convention. When in indirect convention, the CRC is checked after the conversion out of indirect convention. For a given message, the CRC generated (and readable from this register) will be the same whether indirect or direct convention is used to transmit the data to the smart card. The CRCLsB / CRCMsB registers will be updated with CRC/LRC whenever bits are being received or transmitted from/to the smart card (even if CRCEN is not set and in mode T1). They are available to the firmware to use if desired.

Block Wait Time Registers (BWTB0): 0xFE1B ← 0x00, (BWTB1): 0xFE1A ← 0x00, (BWTB2): 0xFE19 ← 0x00, (BWTB3): 0xFE18 ← 0x00

Table 98: The BWTB0 Register

MSB							LSB
BWT.7	BWT.6	BWT.5	BWT.4	BWT.3	BWT.1	BWT.2	BWT.0

Table 99: The BWTB1 Register

MSB							LSB
BWT.15	BWT.14	BWT.13	BWT.12	BWT.11	BWT.10	BWT.9	BWT.8

Table 100: The BWTB2 Register

MSB							LSB
BWT.23	BWT.22	BWT.21	BWT.20	BWT.19	BWT.18	BWT.17	BWT.16

Table 101: The BWTB3 Register

MSB							LSB
–	–	–	–	BWT.27	BWT.26	BWT.25	BWT.24

These registers (BWTB0, BWTB1, BWTB2, BWTB3) are used to set the Block Waiting Time(27:0) (BWT). All of these parameters define the maximum time the 73S1209F will have to wait for a character from the smart card. These registers serve a dual purpose. When T=1, these registers are used to set up the block wait time. The block wait time defines the time in ETUs between the beginning of the last character sent to smart card and the start bit of the first character received from smart card. It can be used to detect an unresponsive card and should be loaded by firmware prior to writing the last TX byte. When T = 0, these registers are used to set up the work wait time. The work wait time is defined as the time between the leading edge of two consecutive characters being sent to or from the card. If a timeout occurs, an interrupt is generated to the firmware. The firmware can then take appropriate action. A Wait Time Extension (WTX) is supported with the 28-bit BWT.

Character Wait Time Registers (CWTB0): 0xFE1D ← 0x00, (CWTB1): 0xFE1C ← 0x00

Table 102: The CWTB0 Register

MSB							LSB
CWT.7	CWT.6	CWT.5	CWT.4	CWT.3	CWT.1	CWT.2	CWT.0

Table 103: The CWTB1 Register

MSB							LSB
CWT.15	CWT.14	CWT.13	CWT.12	CWT.11	CWT.10	CWT.9	CWT.8

These registers (CWTB0, CWTB1) are used to hold the Character Wait Time(15:0) (CWT) or Initial Waiting Time(15:0) (IWT) depending on the situation. Both the IWT and the CWT measure the time in ETUs between the leading edge of the start of the current character received from the smart card and the leading edge of the start of the next character received from the smart card. The only difference is the mode in which the card is operating. When T=1 these registers are used to configure the CWT and these registers configure the IWT when the ATR is being received. These registers should be loaded prior to receiving characters from the smart card. Firmware must manage which time is stored in the register. If a timeout occurs, an interrupt is generated to the firmware. The firmware can then take appropriate action.