Analog Devices Inc./Maxim Integrated - 73S1209F-44IM/F 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-44im-f

Figures

Figure 1: IC Functional Block Diagram 7
Figure 2: Memory Map 15
Figure 3: Clock Generation and Control Circuits 24
Figure 4: Oscillator Circuit 26
Figure 5: Power-Down Control 27
Figure 6: Detail of Power-Down Interrupt Logic 28
Figure 7: Power-Down Sequencing 28
Figure 8: External Interrupt Configuration 32
Figure 9: I^2C Write Mode Operation 55
Figure 10: I^2C Read Operation 56
Figure 11: Simplified Keypad Block Diagram 61
Figure 12: Keypad Interface Flow Chart 63
Figure 13: Smart Card Interface Block Diagram 68
Figure 14: External Smart Card Interface Block Diagram 69
Figure 15: Asynchronous Activation Sequence Timing 72
Figure 16: Deactivation Sequence 72
Figure 17: Smart Card CLK and ETU Generation 73
Figure 18: Guard, Block, Wait and ATR Time Definitions 74
Figure 19: Synchronous Activation 76
Figure 20: Example of Sync Mode Operation: Generating/Reading ATR Signals 76
Figure 21: Creation of Synchronous Clock Start/Stop Mode Start Bit in Sync Mode 77
Figure 22: Creation of Synchronous Clock Start/Stop Mode Stop Bit in Sync Mode 77
Figure 23: Operation of 9-bit Mode in Sync Mode 78
Figure 24: 73S1209F Typical PINpad, Smart Card Reader Application Schematic 103
Figure 25: 73S1209F Typical SIM / Smart Card Reader Application Schematic 104
Figure 26: 12 MHz Oscillator Circuit 110
Figure 27: Digital I/O Circuit 110
Figure 28: Digital Output Circuit 111
Figure 29: Digital I/O with Pull Up Circuit 111
Figure 30: Digital I/O with Pull Down Circuit 112
Figure 31: Digital Input Circuit 112
Figure 32: Keypad Row Circuit 113
Figure 33: Keypad Column Circuit 113
Figure 34: LED Circuit 114
Figure 35: Test and Security Pin Circuit 114
Figure 36: Analog Input Circuit 115
Figure 37: Smart Card Output Circuit 115
Figure 38: Smart Card I/O Circuit 116
Figure 39: $\overline{\text{PRES}}$ Input Circuit 116
Figure 40: $\overline{\text{PRES}}$ Input Circuit 116
Figure 41: 73S1209F Pinout 117
Figure 42: 73S1209F Pinout 118
Figure 43: 73S1209F 68 QFN Pinout 119
Figure 44: 73S1209F 44 QFN Pinout 120

Tables

Table 1: 73S1209F Pinout Description 8
Table 2: MPU Data Memory Map 11
Table 3: Flash Special Function Registers 13
Table 4: Internal Data Memory Map 14
Table 5: Security Control Registers 17
Table 6: IRAM Special Function Registers Locations 18
Table 7: IRAM Special Function Registers Reset Values 19
Table 8: XRAM Special Function Registers Reset Values 21
Table 9: PSW Register Flags 22
Table 10: Port Registers 23
Table 11: Frequencies and Mcount Values for MCLK = 96MHz 25
Table 12: The MCLKCtl Register 25
Table 13: The MPUCKCtl Register 26
Table 14: The INT5Ctl Register 29
Table 15: The MISCtl0 Register 29
Table 16: The MISCtl1 Register 30
Table 17: The MCLKCtl Register 30
Table 18: The PCON Register 31
Table 19: The IEN0 Register 33
Table 20: The IEN1 Register 34
Table 21: The IEN2 Register 34
Table 22: The TCON Register 35
Table 23: The T2CON Register 35
Table 24: The IRCON Register 36
Table 25: External MPU Interrupts 36
Table 26: Control Bits for External Interrupts 37
Table 27: Priority Level Groups 37
Table 28: The IP0 Register 37
Table 29: The IP1 Register 38
Table 30: Priority Levels 38
Table 31: Interrupt Polling Sequence 38
Table 32: Interrupt Vectors 38
Table 33: UART Modes 39
Table 34: Baud Rate Generation 39
Table 35: The PCON Register 40
Table 36: The BRCON Register 40
Table 37: The MISCtl0 Register 41
Table 38: The S0CON Register 42
Table 39: The S1CON Register 43
Table 40: The TMOD Register 44
Table 41: TMOD Register Bit Description 44
Table 42: Timers/Counters Mode Description 45
Table 43: The TCON Register 46
Table 44: The IEN0 Register 47
Table 45: The IEN1 Register 47
Table 46: The IP0 Register 48
Table 47: The WDTREL Register 48
Table 48: Direction Registers and Internal Resources for DIO Pin Groups 49
Table 49: UDIR Control Bit 49
Table 50: Selectable Controls Using the UxIS Bits 49
Table 51: The USRIntCtl1 Register 50
Table 52: The USRIntCtl2 Register 50
Table 53: The USRIntCtl3 Register 50
Table 54: The USRIntCtl4 Register 50
Table 55: The ACOMP Register 51
Table 56: The INT6Ctl Register 52

Program Status Word (PSW):

Table 9: PSW Register Flags

MSB							LSB
CV	AC	F0	RS1	RS	OV	–	P

Bit	Symbol	Function
PSW.7	CV	Carry flag.
PSW.6	AC	Auxiliary Carry flag for BCD operations.
PSW.5	F0	General purpose Flag 0 available for user.
PSW.4	RS1	Register bank select control bits. The contents of RS1 and RS0 select the working register bank:
PSW.3	RS0	
PSW.2	OV	Overflow flag.
PSW.1	F1	General purpose Flag 1 available for user.
PSW.0	P	Parity flag, affected by hardware to indicate odd / even number of "one" bits in the Accumulator, i.e. even parity.

RS1/RS0	Bank Selected	Location
00	Bank 0	(0x00 – 0x07)
01	Bank 1	(0x08 – 0x0F)
10	Bank 2	(0x10 – 0x17)
11	Bank 3	(0x18 – 0x1F)

Stack Pointer (SP): The stack pointer is a 1-byte register initialized to 0x07 after reset. This register is incremented before PUSH and CALL instructions, causing the stack to begin at location 0x08.

Data Pointer: The data pointer (DPTR) is 2 bytes wide. The lower part is DPL, and the highest is DPH. It can be loaded as a 2-byte register (MOV DPTR,#data16) or as two registers (e.g. MOV DPL,#data8). It is generally used to access external code or data space (e.g. MOVC A,@A+DPTR or MOVX A,@DPTR respectively).

Program Counter: The program counter (PC) is 2 bytes wide initialized to 0x0000 after reset. This register is incremented during the fetching operation code or when operating on data from program memory. Note: The program counter is not mapped to the SFR area.

Port Registers: The I/O ports are controlled by Special Function Registers USR70 and USR8. The contents of the SFR can be observed on corresponding pins on the chip. Writing a 1 to any of the ports (see Table 10) causes the corresponding pin to be at high level (3.3V), and writing a 0 causes the corresponding pin to be held at low level (GND). The data direction registers UDIR70 and UDIR8 define individual pins as input or output pins (see the User (USR) Ports section for details).

MPU Clock Control Register (MPUCKCtl): 0xFFA1 ← 0x0C

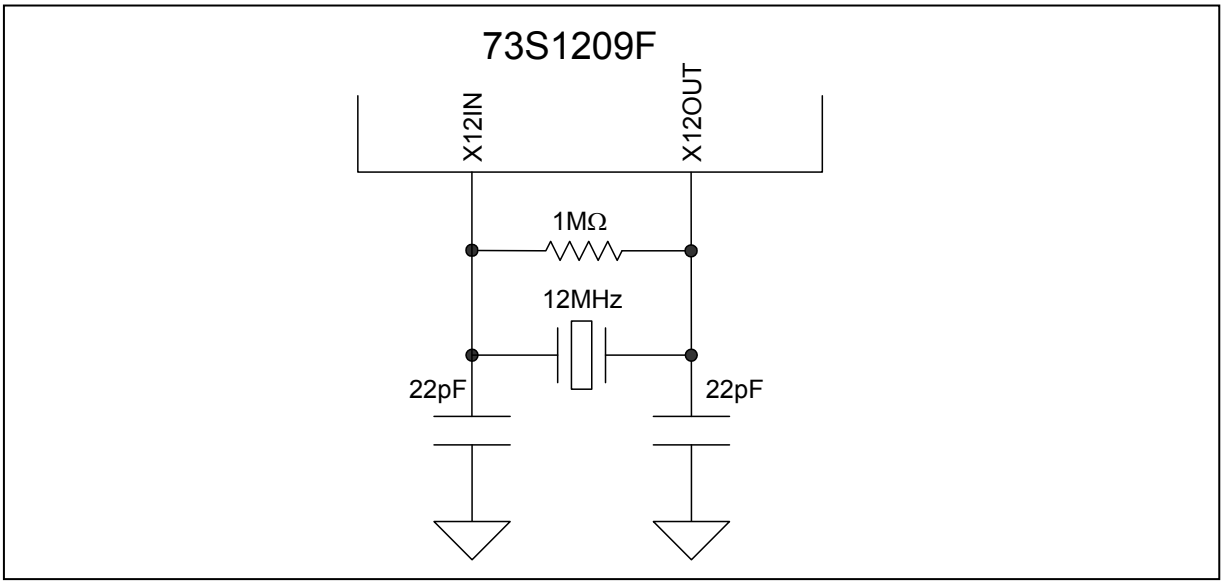
Table 13: The MPUCKCtl Register

MSB							LSB
–	–	MDIV.5	MDIV.4	MDIV.3	MDIV.2	MDIV.1	MDIV.0

Bit	Symbol	Function
MPUCKCtl.7	–	
MPUCKCtl.6	–	
MPUCKCtl.5	MDIV.5	This value determines the ratio of the MPU master clock frequency to the VCO frequency (MCLK) such that MPUClk = MCLK/(2 * (MPUCKDiv(5:0) + 1)). Do not use values of 0 or 1 for MPUCKDiv(n). Default is 0Ch to set CPCLK = 3.6923MHz.
MPUCKCtl.4	MDIV.4	
MPUCKCtl.3	MDIV.3	
MPUCKCtl.2	MDIV.2	
MPUCKCtl.1	MDIV.1	
MPUCKCtl.0	MDIV.0	

The oscillator circuits are designed to connect directly to standard parallel resonant crystal in a Pierce oscillator configuration. Each side of the crystal should include a 22pF capacitor to ground for both oscillator circuits and a 1MΩ resistor is required across the 12MHz crystal.

The CPU clock is available as an output on pin CPUCLK (68-pin version only).



Note: The crystal should be placed as close as possible to the IC, and vias should be avoided.

Figure 4: Oscillator Circuit

Power Control Register 0 (PCON): 0x87 ← 0x00

The SMOD bit used for the baud rate generator is set up via this register.

Table 35: The PCON Register

MSB LSB

SMOD	–	–	–	GF1	GF0	STOP	IDLE

Bit	Symbol	Function
PCON.7	SMOD	If SM0D = 1, the baud rate is doubled.
PCON.6	–	
PCON.5	–	
PCON.4	–	
PCON.3	GF1	General purpose flag 1.
PCON.2	GF0	General purpose flag 1.
PCON.1	STOP	Sets CPU to Stop mode.
PCON.0	IDLE	Sets CPU to Idle mode.

Baud Rate Control Register 0 (BRCON): 0xD8 ← 0x00

The BSEL bit used to enable the baud rate generator is set up via this register.

Table 36: The BRCON Register

MSB LSB

BSEL	–	–	–	–	–	–	–

Bit	Symbol	Function
BRCON.7	BSEL	If BSEL = 0, the baud rate is derived using timer 1. If BSEL = 1 the baud rate generator circuit is used.
BRCON.6	–	
BRCON.5	–	.
BRCON.4	–	
BRCON.3	–	
BRCON.2	–	
BRCON.1	–	
BRCON.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.

1.7.7 User (USR) Ports

The 73S1209F includes 9 pins of general purpose digital I/O (GPIO). On reset or power-up, all USR pins are inputs until they are configured for the desired direction. The pins are configured and controlled by the USR and UDIR SFRs. Each pin declared as USR can be configured independently as an input or output with the bits of the UDIRn registers. Table 48 lists the direction registers and configurability associated with each group of USR pins. USR pins 0 to 7 are multiple use pins that can be used for general purpose I/O, external interrupts and timer control.

Table 49 shows the configuration for a USR pin through its associated bit in its UDIR register. Values read from and written into the GPIO ports use the data registers USR70 and USR8. Note: After reset, all USR pins are defaulted as inputs and pulled up to VDD until any write to the corresponding UDIR register is performed. This insures all USR pins are set to a known value until set by the firmware. Unused USR pins can be set for output if unused and unconnected to prevent them from floating. Alternatively, unused USR pins can be set for input and tied to ground or V_{DD}.

Table 48: Direction Registers and Internal Resources for DIO Pin Groups

USR Pin Group	Type	Direction Register Name	Direction Register (SFR) Location	Data Register Name	Data Register (SFR) Location
USR_0…USR_7	Multi-use	UDIR70	0x91 [7:0]	USR70	0x90 [7:0]
USR_8	GPIO only	UDIR8	0xA1 [0]	USR8	0xA0 [0]

Table 49: UDIR Control Bit

	UDIR Bit	
	0	1
USR Pin Function	output	input

Four XRAM SFR registers (USRIntCtl1, USRIntCtl2, USRIntCtl3, and USRIntCtl4) control the use of the USR [7:0] pins. Each of the USR [7:0] pins can be configured as GPIO or individually be assigned an internal resource such as an interrupt or a timer/counter control. Each of the four registers contains two 3-bit configuration words named UxIS (where x corresponds to the USR pin). The control resources selectable for the USR pins are listed in Table 74 through Table 78. If more than one input is connected to the same resource, the resources are combined using a logical OR.

Table 50: Selectable Controls Using the UxIS Bits

UxIS Value	Resource Selected for USRx Pin
0	None
1	None
2	T0 (counter0 gate/clock)
3	T1 (counter1 gate/clock)
4	Interrupt 0 rising edge/high level on USRx
5	Interrupt 1 rising edge/high level on USRx
6	Interrupt 0 falling edge/low level on USRx
7	Interrupt 1 falling edge/low level on USRx

Note: x denotes the corresponding USR pin. Interrupt edge or level control is assigned in the IT0 and IT1 bits in the TCON register.

1.7.9 LED Drivers

The 73S1209F provides two dedicated output pins for driving LEDs. The LED driver pins can be configured as current sources that will pull to ground to drive LEDs that are connected to VDD without the need for external current limiting resistors. These pins may be used as general purpose outputs with the programmed pull-down current and a strong (CMOS) pull-up, if enabled. The analog block must be enabled when these outputs are being used to drive the selected output current.

The pins may be used as inputs with consideration of the programmed output current and level. The register bit when read, indicates the state of the pin.

LED Control Register (LEDCtl): 0xFFF3 ← 0xFF

Table 57: The LEDCtl Register

MSB LSB

–	LPUEN	ISET.1	ISET.0	–	–	LEDD 1	LEDD0

Bit	Symbol	Function
LEDCtl.7	–	
LEDCtl.6	LPUEN	0 = Pull-ups are enabled for all of the LED pins.
LEDCtl.5	ISET.1	These two bits control the drive current (to ground) for all of the LED driver pins. Current levels are: 00 = 0ma(off) 01 = 2ma 10 = 4ma 11 = 10ma
LEDCtl.4	ISET.0	
LEDCtl.3	–	
LEDCtl.2	–	
LEDCtl.1	LEDD1	Write data controls output level of pin LED1. Read will report level of pin LED1.
LEDCtl.0	LEDD0	Write data controls output level of pin LED0. Read will report level of pin LED0.

I2C Secondary Read Data Register (SRDR): 0XFF84 ← 0x00

Table 62: The SRDR Register

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

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

I2C Control and Status Register (CSR): 0xFF85 ← 0x00

Table 63: The CSR Register

MSB							LSB
–	–	–	–	–	AKERR	I2CST	I2CLEN

Bit	Symbol	Function
CSR.7	–	
CSR.6	–	
CSR.5	–	
CSR.4	–	
CSR.3	–	
CSR.2	AKERR	Set to 1 if acknowledge bit from Slave Device is not 0. Automatically reset when the new bus transaction is started.
CSR.1	I2CST	Write a 1 to start I^2C transaction. Automatically reset to 0 when the bus transaction is done. This bit should be treated as a “busy” indicator on reading. If it is high, the serial read/write operations are not completed and no new address or data should be written.
CSR.0	I2CLEN	Set to 1 for 2-byte read or write operations. Set to 0 for 1-byte operations.

written into KCOL and KROW registers. The keypad interface uses a 1kHz clock derived from the 12MHz crystal. The clock is enabled by setting bit 6 – KBEN – in the MCLKCtl register (see the Oscillator and Clock Generation section) to carry out scanning and debouncing. The keypad size can be adjusted within the KSIZE register.

Normal scanning is performed by hardware when the SCNEN bit is set to 1 in the KSTAT register. Figure 12 shows the flowchart of how the hardware scanning operates. In order to minimize power, scanning does not occur until a key-press is detected. Once hardware key scanning is enabled, the hardware drives all column outputs low and waits for a low to be detected on one of the inputs. When a low is detected on any row, and before key scanning starts, the hardware checks that the low level is still detected after a debounce time. The debounce time is defined by firmware in the KSCAN register (bits 7:0, DBTIME). Debounce times from 4ms to 256ms in 4ms increments are supported. If a key is not pressed after the debounce time, the hardware will go back to looking for any input to be low. If a key is confirmed to be pressed, key scanning begins.

Key scanning asserts one of the 5 drive lines (COL 4:0) low and looks for a low on a sense line indicating that a key is pressed at the intersection of the drive/sense line in the keypad. After all sense lines have been checked without a key-press being detected, the next column line is asserted. The time between checking each sense line is the scan time and is defined by firmware in the KSCAN register (bits 0:1 – SCTIME). Scan times from 1ms to 4ms are supported. Scanning order does not affect the scan time. This scanning continues until the entire keypad is scanned. If only one key is pressed, a valid key is detected. Simultaneous key presses are not considered as valid (If two keys are pressed, no key is reported to firmware).

Possible scrambling of the column scan order is provided by means of KORDERL and KORDERH registers that define the order of column scanning. Values in these registers must be updated every time a new keyboard scan order is desired. It is not possible to change the order of scanning the sense lines. The column and row intersection for the detected valid key are stored in the KCOL and KROW registers. When a valid key is detected, an interrupt is generated. Firmware can then read those registers to determine which key had been pressed. After reading the KCOL and KROW registers, the firmware can update the KORDERL / KORDERH registers if a new scan order is needed.

When the SCNEN bit is enabled in the KSTAT register, the KCOL and KROW registers are only updated after a valid key has been identified. The hardware does not wait for the firmware to service the interrupt in order to proceed with the key scanning process. Once the valid key (or invalid key – e.g. two keys pressed) is detected, the hardware waits for the key to be released. Once the key is released, the debounce timer is started. If the key is not still released after the debounce time, the debounce counter starts again. After a key release, all columns will be driven low as before and the process will repeat waiting for any key to be pressed.

When the SCNEN bit is disabled, all drive outputs are set to the value in the KCOL register. If firmware clears the SCNEN bit in the middle of a key scan, the KCOL register contains the last value stored in there which will then be reflected on the output pins.

A bypass mode is provided so that the firmware can do the key scanning manually (SCNEN bit must be cleared). In bypass mode, the firmware writes/reads the Column and Row registers to perform the key scanning.

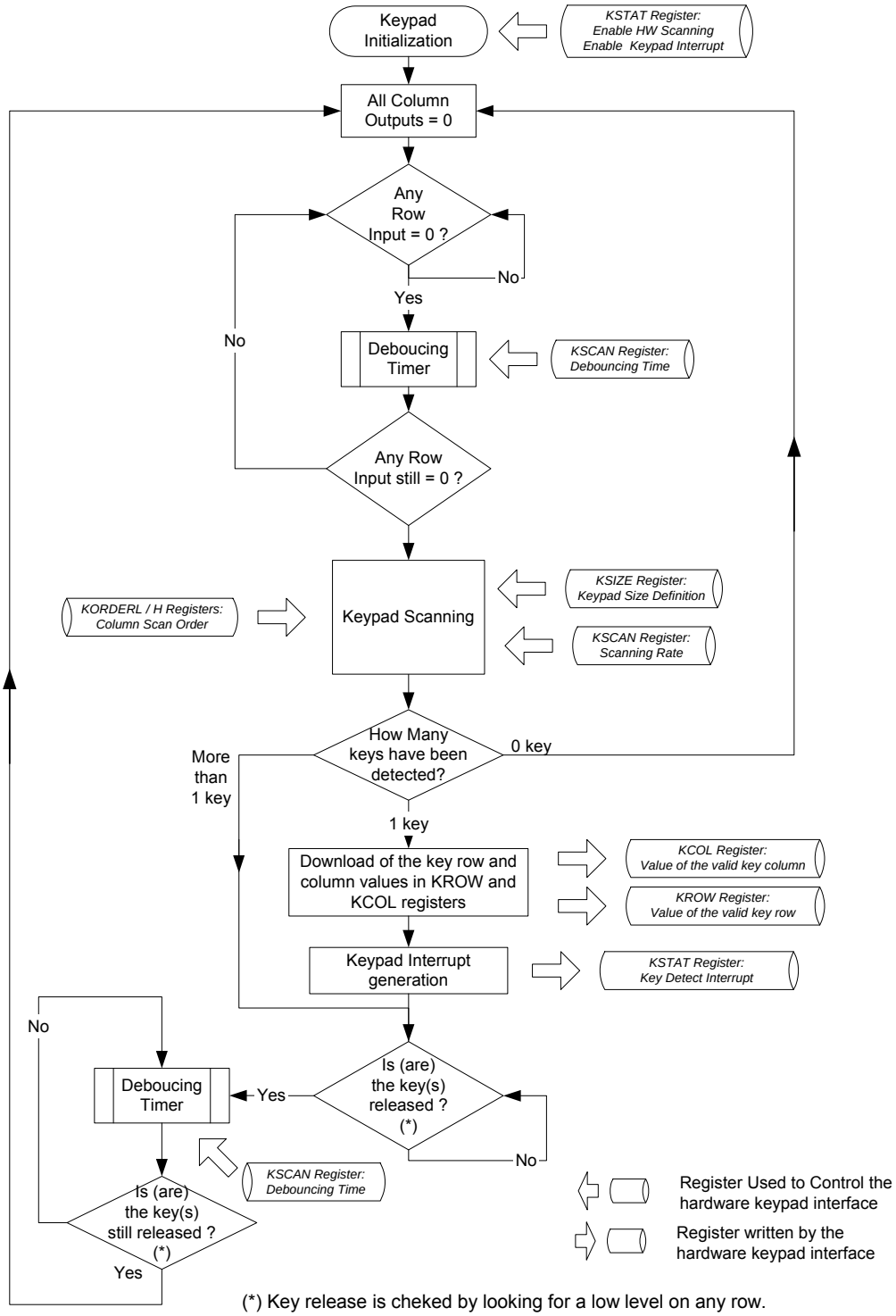


Figure 12: Keypad Interface Flow Chart

Keypad Column MS Scan Order Register (KORDERH): 0xD7 ← 0x00

Table 71: The KORDERH Register

MSB							LSB
–	5COL.2	5COL.1	5COL.0	4COL.2	4COL.1	4COL.0	3COL.2

Bit	Symbol	Function
KORDERH.7	–	
KORDERH.6	5COL.2	Column to scan 5th.
KORDERH.5	5COL.1	
KORDERH.4	5COL.0	
KORDERH.3	4COL.2	Column to scan 4th.
KORDERH.2	4COL.1	
KORDERH.1	4COL.0	
KORDERH.0	3COL.2	Column to scan 3rd (msb).

External Interrupt Control Register (INT5Ctl): 0xFF94 ← 0x00

Table 72: The INT5Ctl Register

MSB							LSB
PDMUX	–	RTCIEN	RTCINT	USBIEN	USBINT	KPIEN	KPINT

Bit	Symbol	Function
INT5Ctl.7	PDMUX	Power down multiplexer control.
INT5Ctl.6	–	
INT5Ctl.5	RTCIEN	When set =1, enables RTC interrupt.
INT5Ctl.4	RTCINT	When set =1, indicates interrupt from Real Time Clock function. Cleared on read of register.
INT5Ctl.3	USBIEN	USB interrupt enable.
INT5Ctl.2	USBINT	USB interrupt flag.
INT5Ctl.1	KPIEN	Enables Keypad interrupt when set = 1.
INT5Ctl.0	KPINT	This bit indicates the Keypad logic has set Key_Detect bit and a key location may be read. Cleared on read of register.

1.7.12 Emulator Port

The emulator port, consisting of the pins E_RST, E_TCLK and E_RXTX, provides control of the MPU through an external in-circuit emulator. The E_TBUS[3:0] pins, together with the E_ISYNC/BRKRQ, add trace capability to the emulator. The emulator port is compatible with the ADM51 emulators manufactured by Signum Systems.

If code trace capability is needed on this interface, 20pF capacitors (to ground) need to be added to allow the trace function capability to run properly. These capacitors should be attached to the TBUS0:3 and ISBR signals.

1.7.13 Smart Card Interface Function

The 73S1209F integrates one ISO-7816 (T=0, T=1) UART, one complete ICC electrical interface as well as an external smart card interface to allow multiple smart cards to be connected using the Teridian 8010 family of interface devices. Figure 13 shows the simplified block diagram of the card circuitry (UART + interfaces), with detail of dedicated XRAM registers.

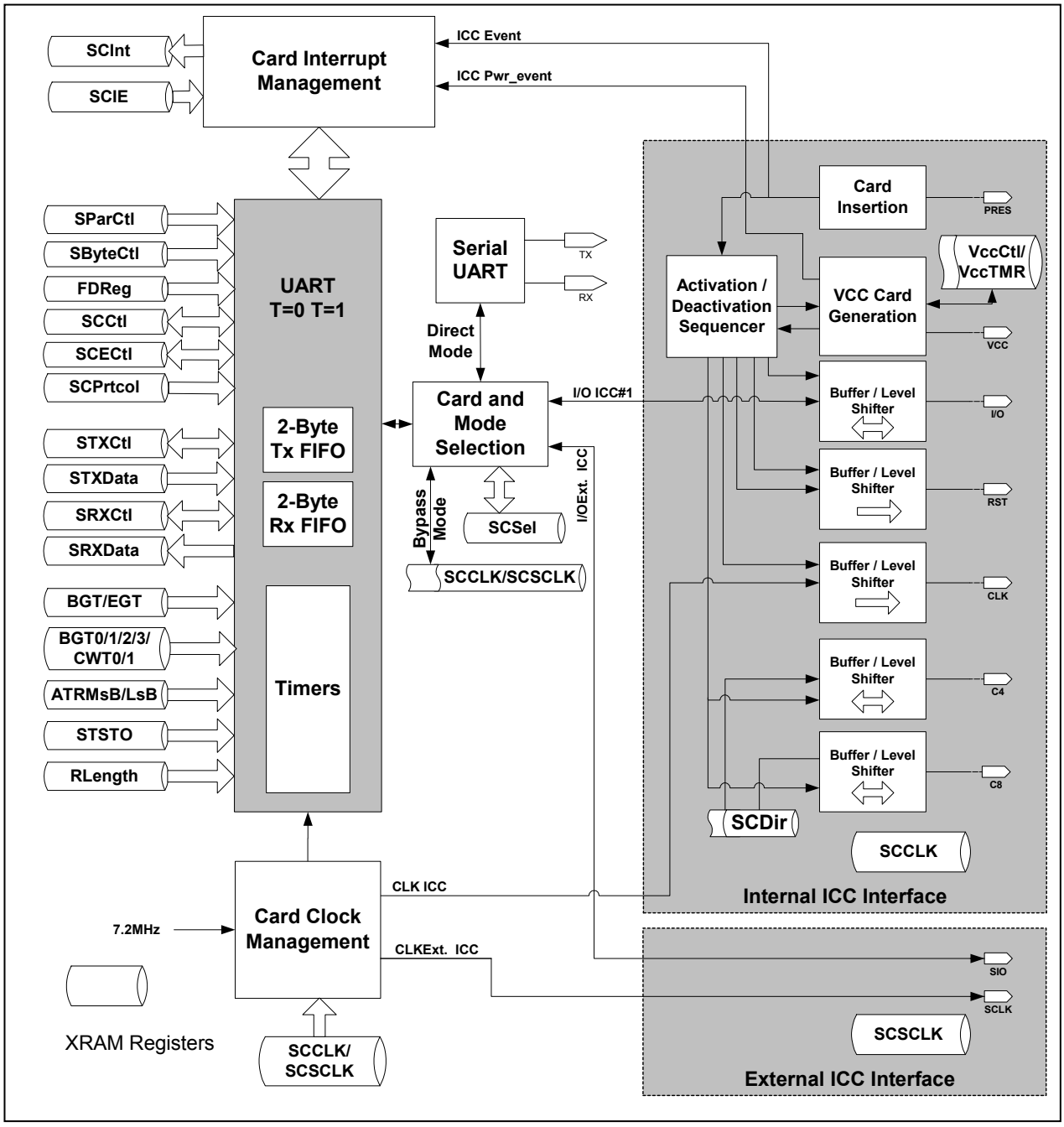
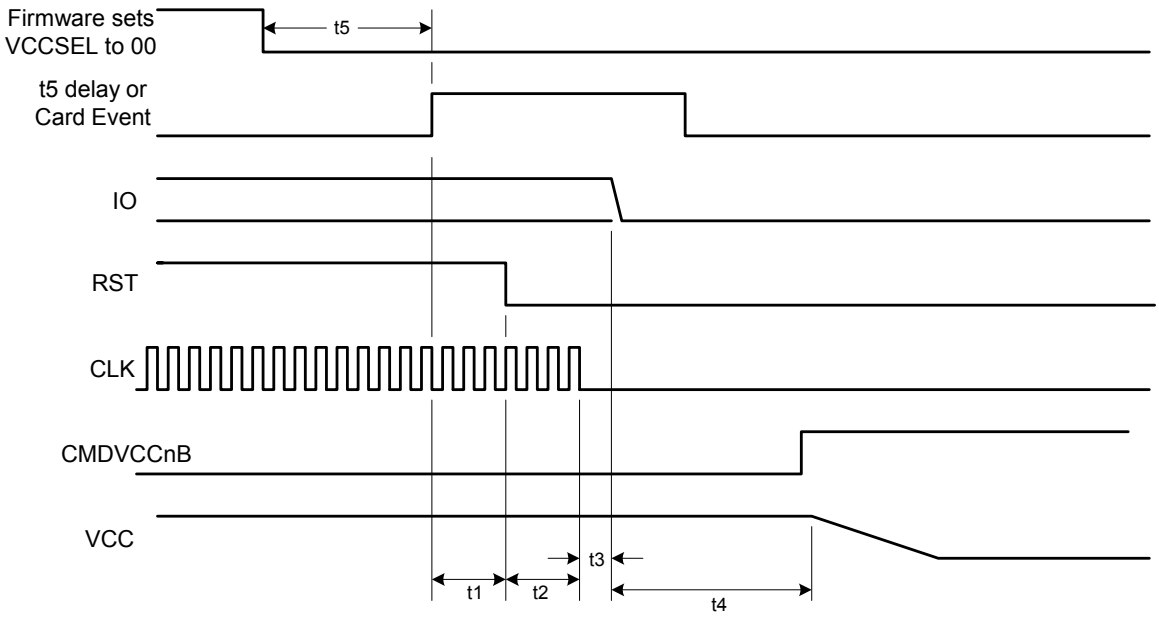


Figure 13: Smart Card Interface Block Diagram

Card interrupts are managed through two dedicated registers SCIE (Interrupt Enable to define which interrupt is enabled) and SCInt (Interrupt status). They allow the firmware to determine the source of an interrupt, that can be a card insertion / removal, card power fault, or a transmission (TX) or reception (RX) event / fault. It should be noted that even when card clock is disabled, an ICC interrupt can be generated

Figure 15: Asynchronous Activation Sequence Timing



t1: Time after either a "card event" occurs or firmware sets the VCCSela and VCCSelb bits to 0 (see t5, VCCOff_tmr) occurs until RST is asserted low.

t2: Time after RST goes low until CLK stops.

t3: Time after CLK stops until IO goes low.

t4: Time after IO goes low until VCC is powered down.

t5: Delayed VCC off time (in ETUs per VCCOff_tmr bits). Only in effect due to firmware deactivation.

Figure 16: Deactivation Sequence

1.7.13.3 Data Reception/Transmission

When a 12Mhz crystal is used, the smart card UART will generate a 3.69Mhz (default) clock to both smart card interfaces. This will allow approximately 9600bps (1/ETU) communication during ATR (ISO 7816 default). As part of the PPS negotiation between the smart card and the reader, the firmware may determine that the smart card parameters F & D may be changed. After this negotiation, the firmware may change the ETU by writing to the SFR FDReg to adjust the ETU and CLK. The firmware may also change the smart card clock frequency by writing to the SFR SCCLK (SCECLK for external interface). Independent clock frequency control is provided to each smart card interface. Clock stop high or Clock stop low is supported in asynchronous mode. Figure 17 shows the ETU and CLK control circuits. The firmware determines when clock stop is supported by the smart card and when it is appropriate to go into that mode (and when to come out of it). The smart card UART is clocked by the same clock that is provided to the selected smart card. The transition between smart card clocks is handled in hardware to eliminate any glitches for the UART during switchover. The external smart card clock is not affected when switching the UART to communicate with the internal smart card.

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.

V_{CC} Stable Timer Register (VccTmr): 0xFE04 ← 0x0F

A programmable timer is provided to set the time from activation start (setting the VCCSEL.1 and VCCSEL.0 bits to non-zero) to when VCC_OK is evaluated. VCC_OK must be true at the end of this timers programmed interval (tto in Figure 15) in order for the activation sequence to continue. If VCC_OK is not true and the end of the interval (tto), the Card Event interrupt will be set, and a deactivation sequence shall begin including clearing of the VCCSEL bits.

Table 77: The VccTmr Register

MSB LSB

OFFTMR.3	OFFTMR.2	OFFTMR.1	OFFTMR.0	VCCTMR.3	VCCTMR.2	VCCTMR.1	VCCTMR.0

Bit	Symbol	Function
VccTmr.7	OFFTMR.3	VCC Off Timer – The bits set the delay (in number of ETUs) for deactivation after the VCCSEL.1 and VCC SEL.0 have been set to 0. The time value is a count of the 32768Hz clock and is given by tto = OFFTMR(7:4) * 30.5µs. This delay does not affect emergency deactivations due to VDD Fault or card events. A value of 0000 results in no additional delay.
VccTmr.6	OFFTMR.2	
VccTmr.5	OFFTMR.1	
VccTmr.4	OFFTMR.0	
VccTmr.3	VCCTMR.3	VCC Timer – VCCOK must be true at the time set by the value in these bits in order for the activation sequence to continue. If not, the VCCSEL bits will be cleared. The time value is a count of the 32768Hz clock and is given by tto = VCCTMR(3:0) * 30.5µs. A value of 0000 results in no timeout, not zero time, and activation requires that RDYST is set and RDY goes high.
VccTmr.2	VCCTMR.2	
VccTmr.1	VCCTMR.1	
VccTmr.0	VCCTMR.0	

Byte Control Register (SByteCtl): 0xFE12 ← 0x2C

This register controls the processing of characters and the detection of the TS byte. When receiving, a Break is asserted at 10.5 ETU after the beginning of the start bit. Break from the card is sampled at 11 ETU.

Table 90: The SByteCtl Register

MSB LSB

–	DETTS	DIRTS	BRKDUR.1	BRKDUR.0	–	–	–

Bit	Symbol	Function
SByteCtl.7	–	
SByteCtl.6	DETTS	Detect TS Byte – 1 = Next Byte is TS, 0 = Next byte is not TS. When set, the hardware will treat the next character received as the TS and determine if direct or indirect convention is being used. Direct convention is the default used if firmware does not set this bit prior to transmission of TS by the smart card to the firmware. The hardware will check parity and generate a break as defined by the DISPAR and BRKGEN bits in the parity control register. This bit is cleared by hardware after TS is received. TS is decoded before being stored in the receive FIFO.
SByteCtl.5	DIRTS	Direct Mode TS Select – 1 = direct mode, 0 = indirect mode. Set/cleared by hardware when TS is processed indicating either direct/indirect mode of operation. When switching between smart cards, the firmware should write the bit appropriately since this register is not unique to an individual smart card (firmware should keep track of this bit).
SByteCtl.4	BRKDUR.1	Break Duration Select – 00 = 1 ETU, 01 = 1.5 ETU, 10 = 2 ETU, 11 = reserved. Determines the length of a Break signal which is generated when detecting a parity error on a character reception in T=0 mode.
SByteCtl.3	BRKDUR.0	
SByteCtl.2	–	
SByteCtl.1	–	
SByteCtl.0	–	

FD Control Register (FDReg): 0xFE13 ← 0x11

Table 91: The FDReg Register

MSB							LSB
FVAL.3	FVAL.2	FVAL.1	FVAL.0	DVAL.3	DVAL.2	DVAL.1	DVAL.0

Bit	Symbol	Function
FDReg.7	FVAL.3	Refer to Table 93 above. This value is converted per the table to set the divide ratio used to generate the baud rate (ETU). Default, also used for ATR, is 0001 (Fi = 372). This value is used by the selected interface.
FDReg.6	FVAL.2	
FDReg.5	FVAL.1	
FDReg.4	FVAL.0	
FDReg.3	DVAL.3	Refer to Table 93 above. This value is used to set the divide ratio used to generate the smart card CLK. Default, also used for ATR, is 0001 (Di = 1).
FDReg.2	DVAL.2	
FDReg.1	DVAL.1	
FDReg.0	DVAL.0	

This register uses the transmission factors F and D to set the ETU (baud) rate. The values in this register are mapped to the ISO 7816 conversion factors as described below. The CLK signal for each interface is created by dividing a high-frequency, intermediate signal (MSCLK) by 2. The ETU baud rate is created by dividing MSCLK by 2 times the Fi/Di ratio specified by the codes below. For example, if FI = 0001 and DI = 0001, the ratio of Fi/Di is 372/1. Thus the ETU divider is configured to divide by 2 * 372 = 744. The maximum supported F/D ratio is 4096.

Table 92: Divider Ratios Provided by the ETU Counter

FI (code)	0000	0001	0010	0011	0100	0101	0110	0111
Fi (ratio)	372	372	558	744	1116	1488	1860	1860⊕
FCLK max	4	5	6	8	12	16	20	20⊕

FI(code)	1000	1001	1010	1011	1100	1101	1110	1111
Fi(ratio)	512⊕	512	768	1024	1536	2048	2048⊕	2048⊕
FCLK max	5⊕	5	7.5	10	15	20	20⊕	20⊕

DI(code)	0000	0001	0010	0011	0100	0101	0110	0111
Di(ratio)	1⊕	1	2	4	8	16	32	32⊕

DI(code)	1000	1001	1010	1011	1100	1101	1110	1111
Di(ratio)	12	20	16⊕	16⊕	16⊕	16⊕	16⊕	16⊕

Note: values marked with ⊕ are not included in the ISO definition and arbitrary values have been assigned.

The values given below are used by the ETU divider to create the ETU clock. The entries that are not shaded will result in precise CLK/ETU per ISO requirements. Shaded areas are not precise but are within 1% of the target value.

Shaded locations indicate functions that are not provided in sync mode.

Table 108: Smart Card SFR Table

Name	Address	b7	b6	b5	b4	b3	b2	b1	b0
SCSel	FE00					SelSC(1:0)		BYPASS	
SCInt	FE01	WAITTO/ RLIEN	CRDEVT	VCCTMR	RXDAVI	TXEVNT	TXSENT	TXERR	RXERR
SCIE	FE02	WTOI/ RLIEN	CDEVNT	VTMREN	RXDAEN	TXEVEN	TXSNTEN	TXERR	RXERR
VccCtl	FE03	VCCSEL.1	VCCSEL.0	VDDFLT	RDYST	VCCOK			SCPWRDN
VccTmr	FE04	OFFTMR(3:0)				VCCTMR(3:0)			
CRDCtl	FE05	DEBOUN	CDETEN			DETPOL	PUENB	PDEN	CARDIN
STXCtl	FE06	SYCKST		TXFULL	TXEMTY	TXUNDR	LASTTX	TX/RXB	BREAKD
STXData	FE07	TXDATA(7:0)							
SRXCtl	FE08	BIT9DAT		LASTRX	CRCERR	RXFULL	RXEMTY	RXOVRR	PARITYE
SRXData	FE09	RXDATA(7:0)							
SCCtl	FE0A	RSTCRD		IO	IOD	C8	C4	CLKLVL	CLKOFF
SCECtl	FE0B			SIO	SIOD			SCLKLVL	SCLKOFF
SCDIR	FE0C					C8D	C4D		
SPrtcol	FE0D	I2CMODE	MOD9/8B	SCESYN	0	TMODE	CRCEN	CRCMS	RCVATR
SCCLK	FE0F			ICLKFS(5:0)					
SCECLK	FE10			ECLKFS(5:0)					
SParCtl	FE11		DISPAR	BRKGEN	BRKDET	RTRAN	DISCRX	INSPE	FORCPE
SByteCtl	FE12		DETTS	DIRTS	BRKDUR (1:0)				
FDReg	FE13	FVAL(3:0)				DVAL (3:0)			
CRCMsB	FE14	CRC(15:8)							
CRCLsB	FE15	CRC(7:0)							
BGT	FE16	EGT8				BGT(4:0)			
EGT	FE17	EGT(7:0)							
BWTB3	FE18					BWT(27:24)			
BWTB2	FE19	BWT(23:16)							
BWTB1	FE1A	BWT(15:8)							
BWTB0	FE1B	BWT(7:0)							
CWTB1	FE1C	CWT(15:8)							
CWTB0	FE1D	CWT(7:0)							
ATRMsB	FE1F	ATRTO(15:8)							
ATRLsB	FE20	ATRTO(7:0)							
STSTO	FE21	TSTO(7:0)							
RLength	FE22	RLen(7:0)							

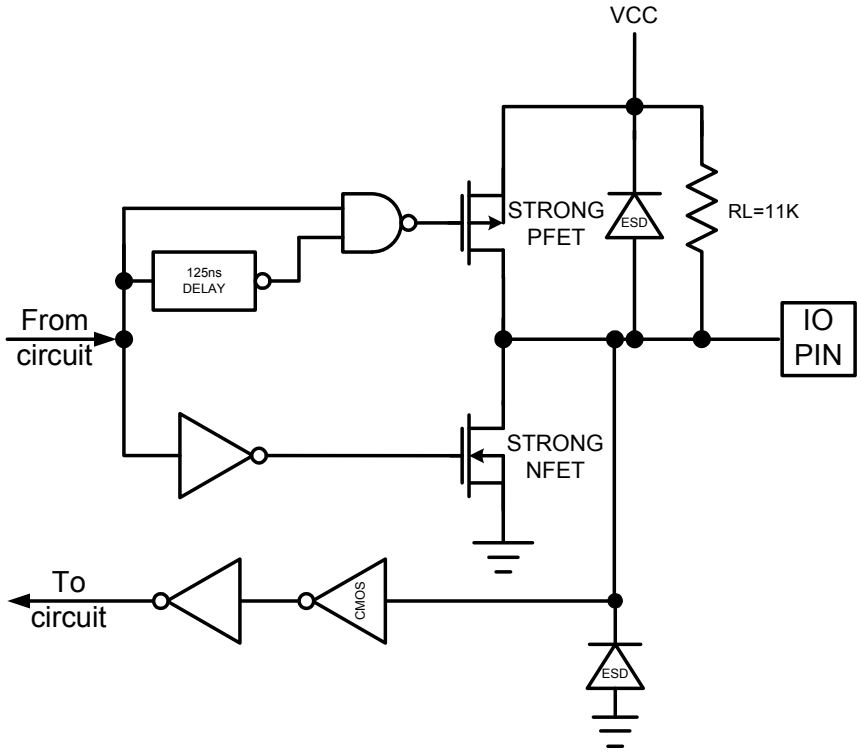


Figure 38: Smart Card I/O Circuit

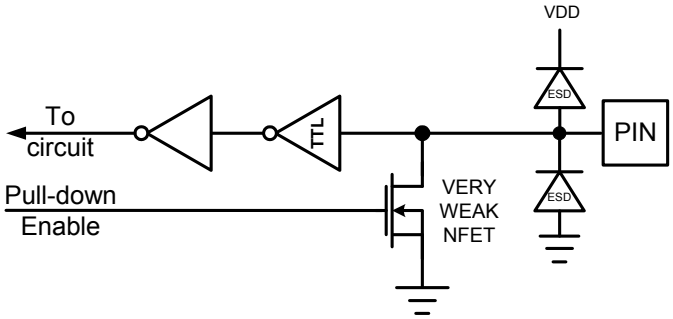


Figure 39: PRES Input Circuit

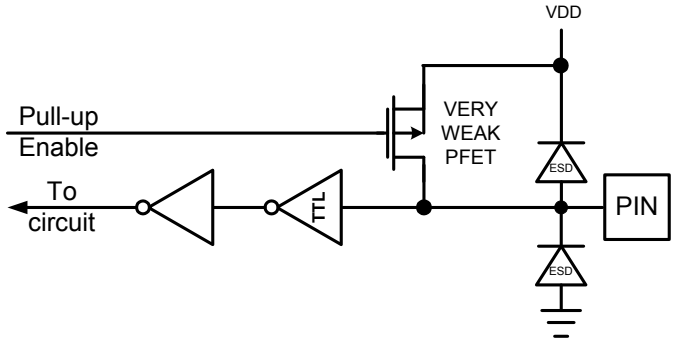


Figure 40: $\overline{\text{PRES}}$ Input Circuit