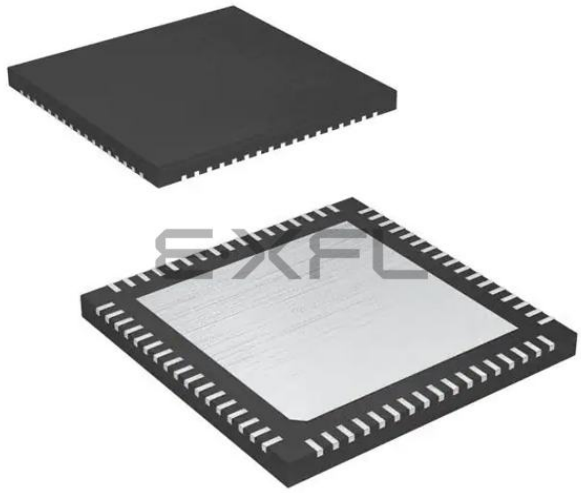
Analog Devices Inc./Maxim Integrated - 73S1209F-68IMR/F/P Datasheet



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	Discontinued at Digi-Key
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	68-VFQFN Exposed Pad
Supplier Device Package	68-QFN (8x8)
Purchase URL	https://www.e-xfl.com/product-detail/analog-devices/73s1209f-68imr-f-p

Internal Data Memory: The Internal data memory provides 256 bytes (0x00 to 0xFF) of data memory. The internal data memory address is always one byte wide and can be accessed by either direct or indirect addressing. The Special Function Registers occupy the upper 128 bytes. **This SFR area is available only by direct addressing. Indirect addressing accesses the upper 128 bytes of Internal RAM.**

The lower 128 bytes contain working registers and bit-addressable memory. The lower 32 bytes form four banks of eight registers (R0-R7). Two bits on the program memory status word (PSW) select which bank is in use. The next 16 bytes form a block of bit-addressable memory space at bit addressees 0x00-0x7F. All of the bytes in the lower 128 bytes are accessible through direct or indirect addressing. Table 4 shows the internal data memory map.

Table 4: Internal Data Memory Map

<table>
<tr><th>Address</th><th>Direct Addressing</th><th>Indirect Addressing</th></tr>
<tr><td>0xFF</td><td rowspan="2">Special Function Registers (SFRs)</td><td rowspan="2">RAM</td></tr>
<tr><td>0x80</td></tr>
<tr><td>0x7F</td><td colspan="2" rowspan="2">Byte-addressable area</td></tr>
<tr><td>0x30</td></tr>
<tr><td>0x2F</td><td colspan="2" rowspan="2">Byte or bit-addressable area</td></tr>
<tr><td>0x20</td></tr>
<tr><td>0x1F</td><td colspan="2" rowspan="2">Register banks R0…R7 (x4)</td></tr>
<tr><td>0x00</td></tr>
</table>

External Data Memory: While the 80515 can address up to 64KB of external data memory in the space from 0x0000 to 0xFFFF, only the memory ranges shown in Figure 2 contain physical memory. The 80515 writes into external data memory when the MPU executes a MOVX @Ri,A or MOVX @DPTR,A instruction. The MPU reads external data memory by executing a MOVX A,@Ri or MOVX A,@DPTR instruction.

There are two types of instructions, differing in whether they provide an eight-bit or sixteen-bit indirect address to the external data RAM.

In the first type (MOVX A,@Ri), the contents of R0 or R1, in the current register bank, provide the eight lower-ordered bits of address. This method allows the user access to the first 256 bytes of the 2KB of external data RAM. In the second type of MOVX instruction (MOVX A,@DPTR), the data pointer generates a sixteen-bit address.

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.**

1.5.3 External Data Special Function Registers (SFRs)

A map of the XRAM Special Function Registers is shown in Table 6. . The smart card registers are listed separately in Table 108.

Table 8: XRAM Special Function Registers Reset Values

Name	Location	Reset Value	Description
DAR	0x FF80	0x00	Device Address Register (I^2C)
WDR	0x FF81	0x00	Write Data Register (I^2C)
SWDR	0x FF82	0x00	Secondary Write Data Register (I^2C)
RDR	0x FF83	0x00	Read Data Register (I^2C)
SRDR	0x FF84	0x00	Secondary Read Data Register (I^2C)
CSR	0x FF85	0x00	Control and Status Register (I^2C)
USRIntCtl1	0x FF90	0x00	External Interrupt Control 1
USRIntCtl2	0x FF91	0x00	External Interrupt Control 2
USRIntCtl3	0x FF92	0x00	External Interrupt Control 3
USRIntCtl4	0x FF93	0x00	External Interrupt Control 4
INT5Ctl	0x FF94	0x00	External Interrupt Control 5
INT6Ctl	0x FF95	0x00	External Interrupt Control 6
MPUCKCtl	0x FFA1	0x0C	MPU Clock Control
ACOMP	0x FFD0	0x00	Analog Compare Register
TRIMPCtl	0x FFD1	0x00	TRIM Pulse Control
FUSECtl	0x FFD2	0x00	FUSE Control
VDDFCtl	0x FFD4	0x00	VDDFault Control
SECReg	0x FFD7	0x00	Security Register
MISCtl0	0x FFF1	0x00	Miscellaneous Control Register 0
MISCtl1	0x FFF2	0x10	Miscellaneous Control Register 1
LEDCtl	0x FFF3	0xFF	LED Control Register

Accumulator (ACC, A): ACC is the accumulator register. Most instructions use the accumulator to hold the operand. The mnemonics for accumulator-specific instructions refer to accumulator as “A”, not ACC.

B Register: The B register is used during multiply and divide instructions. It can also be used as a scratch-pad register to hold temporary data.

Miscellaneous Control Register 1 (MISCtl1): 0xFFF2 ← 0x10

Table 16: The MISCtl1 Register

MSB							LSB
–	–	FRPEN	FLSH66	–	–	–	–

Bit	Symbol	Function
MISCtl1.7	–	
MISCtl1.6	–	
MISCtl1.5	FRPEN	Flash Read Pulse enable (low). If FRPEN=1, the Flash Read signal is passed through with no change. When FRPEN=0, a one-shot circuit that shortens the Flash Read signal is enabled to save power. The Flash Read pulse will shorten to 40 or 66ns (approximate based on the setting of the FLSH66 bit) in duration, regardless of the MPU clock rate. For MPU clock frequencies greater than 10MHz, this bit should be set high.
MISCtl1.4	FLSH66	When high, creates a 66ns Flash read pulse, otherwise creates a 40ns read pulse when FRPEN is set.
MISCtl1.3	–	
MISCtl1.2	–	
MISCtl1.1	–	
MISCtl1.0	–	

Master Clock Control Register (MCLKCtl): 0x8F ← 0x0A

Table 17: The MCLKCtl Register

MSB							LSB
HSOEN	KBEN	SCEN	–	–	MCT.2	MCT.1	MCT.0

Bit	Symbol	Function
MCLKCtl.7	HSOEN*	High-speed oscillator enable. When set = 1, disables the high-speed crystal oscillator and VCO/PLL system. This bit is not changed when the PWRDN bit is set but the oscillator/VCO/PLL is disabled.
MCLKCtl.6	KBEN	1 = Disable the keypad logic clock. This bit is not changed in PWRDN mode but the function is disabled.
MCLKCtl.5	SCEN	1 = Disable the smart card logic clock. This bit is not changed in PWRDN mode but the function is disabled. Interrupt logic for card insertion/removal remains operable even with smart card clock disabled.
MCLKCtl.4	–	
MCLKCtl.3	–	
MCLKCtl.2	MCT.2	This value determines the ratio of the VCO frequency (MCLK) to the high-speed crystal oscillator frequency such that: MCLK=(MCount*2 + 4)*Fxtal. The default value is MCount= 2h such that MCLK = (2*2 + 4)*12.00MHz = 96MHz.
MCLKCtl.1	MCT.1	
MCLKCtl.0	MCT.0	

*Note: The HSOEN bit should never be set under normal circumstances. Power down control should only be initiated via use of the PWRDN bit in MISCtl0.

1.7.4 UART

The 80515 core of the 73S1209F includes two separate UARTs that can be programmed to communicate with a host. The 73S1209F can only connect one UART at a time since there is only one set of TX and Rx pins. The MISCtl0 register is used to select which UART is connected to the TX and RX pins. Each UART has a different set of operating modes that the user can select according to their needs. The UART is a dedicated 2-wire serial interface, which can communicate with an external host processor at up to 115,200 bits/s. The TX and RX pins operate at the V_{DD} supply voltage levels and should never exceed 3.6V. The operation of each pin is as follows:

RX: Serial input data is applied at this pin. Conforming to RS-232 standard, the bytes are input LSB first. The voltage applied at RX must not exceed 3.6V.

TX: This pin is used to output the serial data. The bytes are output LSB first.

The 73S1209F has several UART-related read/write registers. All UART transfers are programmable for parity enable, parity select, 2 stop bits/1 stop bit and XON/XOFF options for variable communication baud rates from 300 to 115200 bps. Table 33 shows the selectable UART operation modes and Table 34 shows how the baud rates are calculated.

Table 33: UART Modes

	UART 0	UART 1
Mode 0	N/A	Start bit, 8 data bits, parity, stop bit, variable baud rate (internal baud rate generator)
Mode 1	Start bit, 8 data bits, stop bit, variable baud rate (internal baud rate generator or timer 1)	Start bit, 8 data bits, stop bit, variable baud rate (internal baud rate generator)
Mode 2	Start bit, 8 data bits, parity, stop bit, fixed baud rate 1/32 or 1/64 of f_{CKMPU}	N/A
Mode 3	Start bit, 8 data bits, parity, stop bit, variable baud rate (internal baud rate generator or timer 1)	N/A

Note: Parity of serial data is available through the P flag of the accumulator. Seven-bit serial modes with parity, such as those used by the FLAG protocol, can be simulated by setting and reading bit 7 of 8-bit output data. Seven-bit serial modes without parity can be simulated by setting bit 7 to a constant 1.8-bit serial modes with parity can be simulated by setting and reading the 9th bit, using the control bits S0CON3 and S1CON3 in the S0COn and S1CON SFRs.

Table 34: Baud Rate Generation

	Using Timer 1	Using Internal Baud Rate Generator
Serial Interface 0	$2^{smod} * f_{CKMPU}/ (384 * (256-TH1))$	$2^{smod} * f_{CKMPU}/(64 * (2^{10}-S0REL))$
Serial Interface 1	N/A	$f_{CKMPU}/(32 * (2^{10}-S1REL))$

Note: S0REL (9:0) and S1REL (9:0) are 10-bit values derived by combining bits from the respective timer reload registers SxRELH (bits 1:0) and SxRELL (bits 7:0). TH1 is the high byte of timer 1. The SMOD bit is located in the PCON SFR.

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

Table 44: The IEN0 Register

MSB							LSB
EAL	WDT	ET2	ES0	ET1	EX1	ET0	EX0

Bit	Symbol	Function
IEN0.7	EAL	EAL = 0 – disable all interrupts.
IEN0.6	WDT	Watchdog timer refresh flag. Set to initiate a refresh of the watchdog timer. Must be set directly before SWDT is set to prevent an unintentional refresh of the watchdog timer. WDT is reset by hardware 12 clock cycles after it has been set.
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.

Interrupt Enable 1 Register (IEN1): 0xB8 ← 0x00

Table 45: The IEN1 Register

MSB							LSB
–	SWDT	EX6	EX5	EX4	EX3	EX2	

Bit	Symbol	Function
IEN1.7	–	
IEN1.6	SWDT	Watchdog timer start/refresh flag. Set to activate/refresh the watchdog timer. When directly set after setting WDT, a watchdog timer refresh is performed. Bit SWDT is reset by the hardware 12 clock cycles after it has been set.
IEN1.5	EX6	EX6 = 0 – disable external interrupt 6.
IEN1.4	EX5	EX5 = 0 – disable external interrupt 5.
IEN1.3	EX4	EX4 = 0 – disable external interrupt 4.
IEN1.2	EX3	EX3 = 0 – disable external interrupt 3.
IEN1.1	EX2	EX2 = 0 – disable external interrupt 2.
IEN1.0	–	

Keypad Column Register (KCOL): 0xD1 ← 0x1F

This register contains the value of the column of a key detected as valid by the hardware. In bypass mode, this register firmware writes directly this register to carry out manual scanning.

Table 65: The KCOL Register

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

Bit	Symbol	Function
KCOL.7	–	
KCOL.6	–	
KCOL.5	–	
KCOL.4	COL.4	Drive lines bit mapped to corresponding with pins COL(4:0). When a key is detected, firmware reads this register to determine column. In bypass (S/W keyscan) mode, Firmware writes this register directly. 0x1E = COL(0) low, all others high. 0x0F = COL(4) low, all others high. 0x1F = COL(4:0) all high.
KCOL.3	COL.3	
KCOL.2	COL.2	
KCOL.1	COL.1	
KCOL.0	COL.0	

Keypad Row Register (KROW): 0xD2 ← 0x3F

This register contains the value of the row of a key detected as valid by the hardware. In bypass mode, this register firmware reads directly this register to carry out manual detection.

Table 66: The KROW Register

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

Bit	Symbol	Function
KROW.7	–	
KROW.6	–	
KROW.5	ROW.6	Sense lines bit mapped to correspond with pins ROW(5:0). When key detected, firmware reads this register to determine row. In bypass mode, firmware reads rows and has to determine if there was a key press or not. 0x3E = ROW(0) low, all others high. 0x1F = ROW(5) low, all others high. 0x3F = ROW(5:0) all high.
KROW.4	ROW.4	
KROW.3	ROW.3	
KROW.2	ROW.2	
KROW.1	ROW.1	
KROW.0	ROW.0	

Keypad Scan Time Register (KSCAN): 0xD3 ← 0x00

This register contains the values of scanning time and debouncing time.

Table 67: The KSCAN Register

MSB							LSB
DBTIME.5	DBTIME.4	DBTIME.3	DBTIME.2	DBTIME.1	DBTIME.0	SCTIME.1	SCTIME.0

Bit	Symbol	Function
KSCAN.7	DBTIME.5	De-bounce time in 4ms increments. 1 = 4ms de-bounce time, 0x3F = 252ms, 0x00 = 256ms. Key presses and key releases are de-bounced by this amount of time.
KSCAN.6	DBTIME.4	
KSCAN.5	DBTIME.3	
KSCAN.4	DBTIME.2	
KSCAN.3	DBTIME.1	
KSCAN.2	DBTIME.0	
KSCAN.1	SCTIME.1	Scan time in ms. 01 = 1ms, 02 = 2ms, 00 = 3ms, 00 = 4ms. Time between checking each key during keypad scanning.
KSCAN.0	SCTIME.0	

Keypad Control/Status Register (KSTAT): 0xD4 ← 0x00

This register is used to control the hardware keypad scanning and detection capabilities, as well as the keypad interrupt control and status.

Table 68: The KSTAT Register

MSB							LSB
–	–	–	–	KEYCLK	HWSCEN	KEYDET	KYDTEN

Bit	Symbol	Function
KSTAT.7	–	
KSTAT.6	–	
KSTAT.5	–	
KSTAT.4	–	
KSTAT.3	KEYCLK	The current state of the keyboard clock can be read from this bit.
KSTAT.2	HWSCEN	Hardware Scan Enable – When set, the hardware will perform automatic key scanning. When cleared, the firmware must perform the key scanning manually (bypass mode).
KSTAT.1	KEYDET	Key Detect – When HWSCEN = 1 this bit is set causing an interrupt that indicates a valid key press was detected and the key location can be read from the Keypad Column and Row registers. When HWSCEN = 0, this bit is an interrupt which indicates a falling edge on any Row input if all Row inputs had been high previously (note: multiple Key Detect interrupts may occur in this case due to the keypad switch bouncing). In all cases, this bit is cleared when read. When HWSCEN = 0 and the keypad interface 1kHz clock is disabled, a key press will still set this bit and cause an interrupt.
KSTAT.0	KYDTEN	Key Detect Enable – When set, the KEYDET bit can cause an interrupt and when cleared the KEYDET cannot cause an interrupt. KEYDET can still get set even if the interrupt is not enabled.

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.

Smart Card V_{CC} Control/Status Register (VccCtl): 0xFE03 ← 0x00

This register is used to control the power up and power down of the integrated smart card interface. It is used to determine whether to apply 5V, 3V, or 1.8V to the smart card. Perform the voltage selection with one write operation, setting both VCCSEL.1 and VCCSEL.0 bits simultaneously. The VDDFLT bit (if enabled) will provide an emergency deactivation of the internal smart card slot. See the VDD Fault Detect Function section for more detail.

Table 76: The VccCtl Register

MSB LSB

VCCSEL.1	VCCSEL.0	VDDFLT	RDYST	VCCOK	–	–	SCPWRDN

Bit	Symbol	Function
VccCtl.7	VCCSEL.1	Setting non-zero value for bits 7,6 will begin activation sequence with target Vcc as given below: State VCCSEL.1 VCCSEL.0 VCC 1 0 0 0V 2 0 1 1.8V 3 1 0 3.0V 4 1 1 5V
VccCtl.6	VCCSEL.0	A card event or VCCOK going low will initiate a deactivation sequence. When the deactivation sequence for RST, CLK and I/O is complete, V_{CC} will be turned off. When this type of deactivation occurs, the bits must be reset before initiating another activation.
VccCtl.5	VDDFLT	If this bit is set = 0, the CMDVCC3B and CMDVCC5B outputs are immediately set = 1 to signal to the companion circuit to begin deactivation when there is a VDD Fault event. If this bit is set = 1 and there is a VDD Fault, the firmware should perform a deactivation sequence and then set CMDVCC3B or CMDVCC5B = 1 to signal the companion circuit to set VCC = 0.
VccCtl.4	RDYST	If this bit is set = 1, the activation sequence will start when bit VCCOK is set = 1. If not set, the deactivation sequence shall start when the VCCTMR times out.
VccCtl.3	VCCOK	(Read only). Indicates that V_{CC} output voltage is stable.
VccCtl.2	–	
VccCtl.1	–	
VccCtl.0	SCPWRDN	This bit controls the power-down mode of the 73S1209F circuit. 1 = power down, 0 = normal operation.

STX Data Register (STXData): 0xFE07 ← 0x00

Table 80: The STXData Register

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

Bit	Function
STXData.7	Data to be transmitted to smart card. Gets stored in the TX FIFO and then extracted by the hardware and sent to the selected smart card. When the MPU reads this register, the byte pointer is changed to effectively "read out" the data. Thus, two reads will always result in an "empty" FIFO condition. The contents of the FIFO registers are not cleared, but will be overwritten by writes.
STXData.6	
STXData.5	
STXData.4	
STXData.3	
STXData.2	
STXData.1	
STXData.0	

SRX Control/Status Register (SRXCtl): 0xFE08 ← 0x00

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

Table 81: The SRXCtl Register

MSB							LSB
BIT9DAT	–	LASTRX	CRCERR	RXFULL	RXEMTY	RXOVRR	PARITYE

Bit	Symbol	Function
SRXCtl.7	BIT9DAT	Bit 9 Data – When in sync mode and with MODE9/8B set, this bit will contain the data on IO (or SIO) pin that was sampled on the ninth CLK (or SCLK) rising edge. This is used to read data in synchronous 9-bit formats.
SRXCtl.6	–	
SRXCtl.5	LASTRX	Last RX Byte – User sets this bit during the reception of the last byte. When byte is received and this bit is set, logic checks CRC to match 0x1D0F (T=1 mode) or LRC to match 00h (T=1 mode), otherwise a CRC or LRC error is asserted.
SRXCtl.4	CRCERR	(Read only) 1 = CRC (or LRC) error has been detected.
SRXCtl.3	RXFULL	(Read only) RX FIFO is full. Status bit to indicate RX FIFO is full.
SRXCtl.2	RXEMTY	(Read only) RX FIFO is empty. This is only a status bit and does not generate a RX interrupt.
SRXCtl.1	RXOVRR	RX Overrun – (Read Only) Asserted when a receive-over-run condition has occurred. An over-run is defined as a byte was received from the smart card when the RX FIFO was full. Invalid data may be in the receive FIFO. Firmware should take appropriate action. Cleared when read. Additional writes to the RX FIFO are discarded when a RXOVRR occurs until the overrun condition is cleared. Will generate RXERR interrupt.
SRXCtl.0	PARITYE	Parity Error – (Read only) 1 = The logic detected a parity error on incoming data from the smart card. Cleared when read. Will generate RXERR interrupt.

External Smart Card Control Register (SCECtl): 0xFE0B ← 0x00

This register is used to directly set and sample signals of External Smart Card interface. There are three modes of asynchronous operation, an "automatic sequence" mode, and bypass mode. Clock stop per the ISO 7816-3 interface is also supported but firmware must handle the protocol for SIO and SCLK for I^2C clock stop and start. Control for Reset (to make RST signal), activation control, voltage select, etc. should be handled via the I^2C interface when using external 73S73S8010x devices. USR(n) pins shall be used for C4, C8 functions if necessary.

Table 84: The SCECtl Register

MSB LSB

–	–	SIO	SIOD	–	–	SCLKLVL	SCLKOFF

Bit	Symbol	Function
SCECtl.7	–	
SCECtl.6	–	
SCECtl.5	SIO	External Smart Card I/O. Bit when read indicates state of pin SIO for SIOD = 1 (Caution, this signal is not synchronized to the MPU clock), when written, sets state of pin SIO for SIOD = 0. Ignored if not in bypass or sync modes. In sync mode, this bit will contain the value of IO pin on the latest rising edge of SCLK.
SCECtl.4	SIOD	1 = input, 0 = output. External Smart Card I/O Direction control. Ignored if not in bypass or sync modes.
SCECtl.3	–	
SCECtl.2	–	
SCECtl.1	SCLKLVL	Sets the state of SCLK when disabled by SCLKOFF bit. If in bypass mode, this bit directly controls the state of SCLK.
SCECtl.0	SCLKOFF	0 = SCLK enabled, 1 = SCLK disabled. When disabled, SCLK level is determined by SCLKLVL. 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.

ATR Timeout Registers (ATRLsB): 0xFE20 ← 0x00, (ATRMsB): 0xFE1F ← 0x00

Table 104: The ATRLsB Register

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

Table 105: The ATRMsB Register

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

These registers (ATRLsB and ATRLsB) form the ATR timeout (ATRTO [15:0]) parameter. Time in ETU between the leading edge of the first character and leading edge of the last character of the ATR response. Timer is enabled when the RCVATR is set and starts when leading edge of the first start bit is received and disabled when the RCVATR is cleared. An ATR timeout is generated if this time is exceeded.

TS Timeout Register (STSTO): 0xFE21 ← 0x00

Table 106: The STSTO Register

MSB							LSB
TST0.7	TST0.6	TST0.5	TST0.4	TST0.3	TST0.1	TST0.2	TST0.0

The TS timeout is the time in ETU between the de-assertion of smart card reset and the leading edge of the TS character in the ATR (when DETTS is set). The timer is started when smart card reset is de-asserted. An ATR timeout is generated if this time is exceeded (MUTE card).

Reset Time Register (RLength): 0xFE22 ← 0x70

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

Table 107: The RLength Register

Time in ETUs that the hardware delays the de-assertion of RST. If set to zero and RSTCRD = 0, the hardware adds no extra delay and the hardware will release RST after VCCOK is asserted during power-up. If set to one, it will delay the release of RST by the time in this register. When the firmware sets the RSTCRD bit, the hardware will assert reset (RST = 0 on pin). When firmware clears the bit, the hardware will release RST after the delay specified in Rlen. If firmware sets the RSTCRD bit prior to instructing the power to be applied to the smart card, the hardware will not release RST after power-up until RLen after the firmware clears the RSTCRD bit. This provides a means to power up the smart card and hold it in reset until the firmware wants to release the RST to the selected smart card. Works with the selected smart card interface.

3.7 DC Characteristics

Symbol	Parameter	Condition	Min	Typ.	Max	Unit
I_{DD}	Supply Current	CPU clock @ 24MHz		20	25	mA
		CPU clock @ 12MHz		14	20	mA
		CPU clock @ 6MHz		10	15	mA
		CPU clock @ 3.69MHz		9	14	mA
		Power down (-40° to 85°C)		8	50	µA
		Power down (25°C)		6	13	µA
I_{PC}	Supply Current	V_{CC} on, ICC=0 I/O, AUX1, AUX2=high, CLK not toggling		450	650	µA
		Power down		1	10	
I_{PCOFF}	V_{PC} supply current when V_{CC} = 0	Smart card deactivated		345		µA

3.8 Voltage / Temperature Fault Detection Circuits

Symbol	Parameter	Condition	Min	Typ.	Max	Unit
V_{PCF}	V_{PC} fault (V_{PC} Voltage supervisor threshold)	V_{PC}<V_{CC}, a transient event		V_{CC} > V_{PC} + 0.3		V
V_{CCF}	VCCOK = 0 (V_{CC} Voltage supervisor threshold)	V_{CC} = 5V			4.6	V
		V_{CC}= 3V			2.7	V
		V_{CC}= 1.8V			1.65	
T_F	Die over temperature fault		115		145	°C
ICCF	Vcc over current fault		110			mA

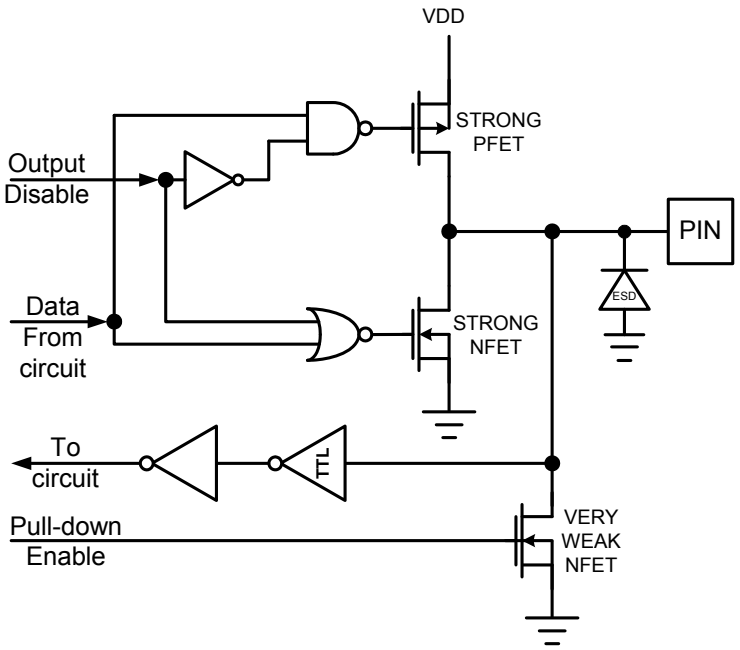


Figure 30: Digital I/O with Pull Down Circuit

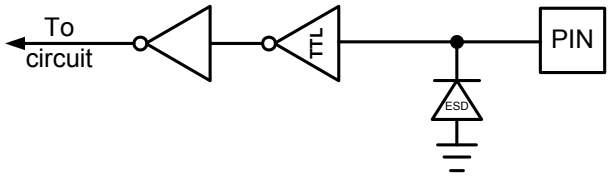


Figure 31: Digital Input Circuit

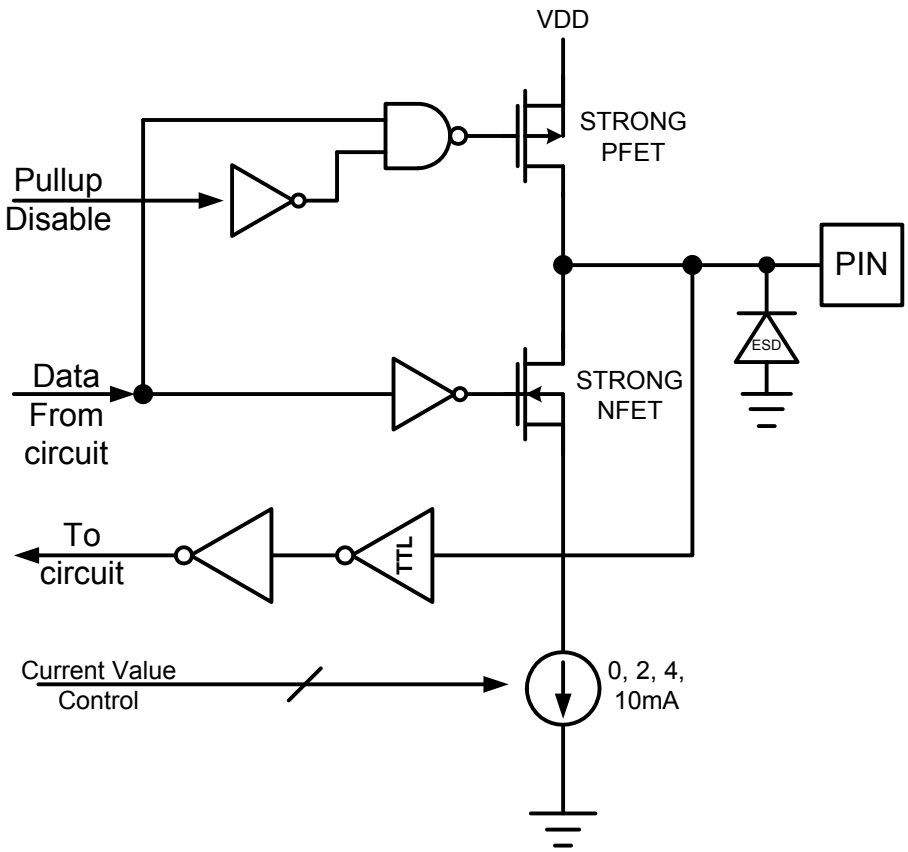


Figure 34: LED Circuit

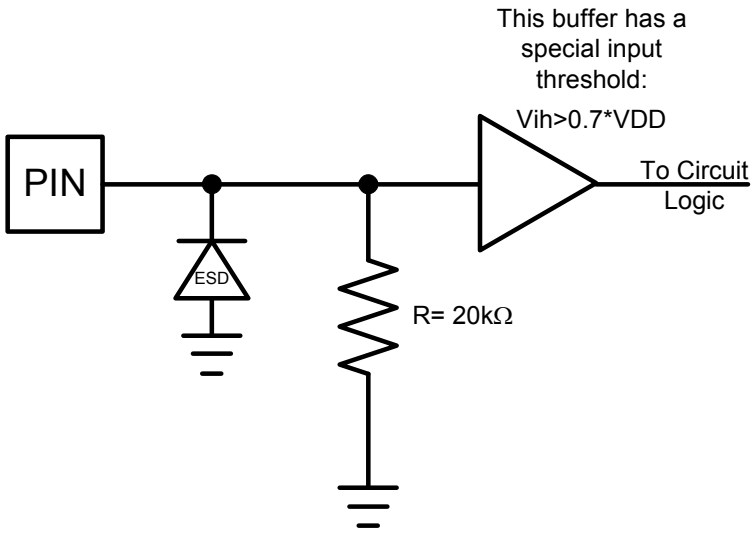


Figure 35: Test and Security Pin Circuit

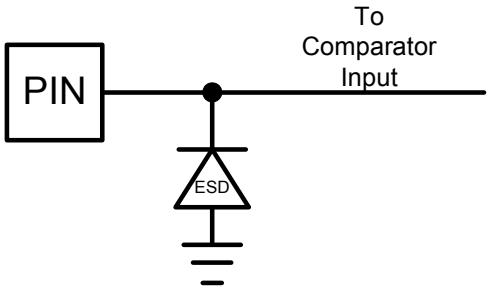


Figure 36: Analog Input Circuit

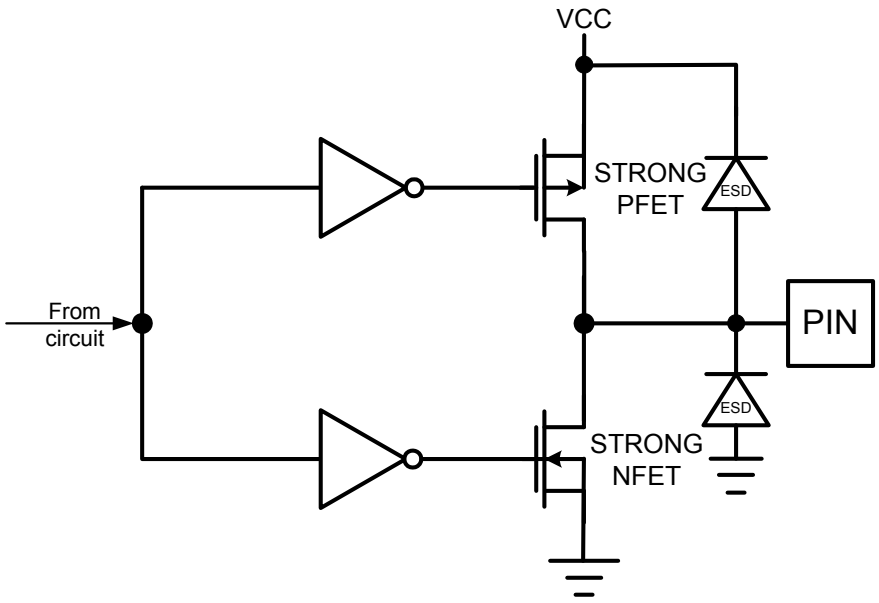


Figure 37: Smart Card Output Circuit

Revision History

Revision	Date	Description
1.0	4/4/2007	First publication.
1.1	11/7/2007	In Table 1, added the Type column and the Equivalent Circuit references. In Section 1.4, updated program security description to remove pre-boot and 32-cycle references. In Section 1.7.1, changed "Mcount is configured in the MCLKCtl register must be bound between a value of 1 to 7. The possible crystal or external clock are shown in Table 12." to "Mcount is configured in the MCLKCtl register must be bound between a value of 1 to 7. The possible crystal or external clock frequencies for getting MCLK = 96MHz are shown in Table 11." In the BRCON description, changed "If BSEL = 1, the baud rate is derived using timer 1." to "If BSEL = 0, the baud rate is derived using timer 1." In Section 1.7.12, removed the following from the emulator port description: "The signals of the emulator port have weak pull-ups. Adding resistor footprints for signals E_RST, E_TCLK and E_RXTX on the PCB is recommended. If necessary, adding 10KΩ pull-up resistors on E_TCLK and E_RXTX and a 3KΩ on E_RST will help the emulator operate normally if a problem arises." Added Section 4, Equivalent Circuits. In Ordering Information, removed the leaded part numbers.
1.2	12/16/2008	In Table 1, added more description to the SCL, SDA, PRES, PRESB, VCC, VPC, SEC, TEST and VDD pins. In Section 1.3.2, changed "FLSH_ERASE" to "ERASE" and "FLSH_PGADR" to "PGADDR". Added "The PGADDR register denotes the page address for page erase. The page size is 512 (200h) bytes and there are 128 pages within the flash memory. The PGADDR denotes the upper seven bits of the flash memory address such that bit 7:1 of the PGADDR corresponds to bit 15:9 of the flash memory address. Bit 0 of the PGADDR is not used and is ignored." In the description of the PGADDR register, added "Note: the page address is shifted left by one bit (see detailed description above)." In Table 3, change "FLSH_PGADR" to "PGADDR". In Table 3, changed "FLSHCRL" to "FLSHCTL". In Table 3, added "Note: the page address is shifted left by one bit (see detailed description above)." to the PGADDR description. In Table 5, moved the TRIMPCtl bit description to FUSECtl and moved the FUSECtl bit description to TRIMPCtl. In Table 6, changed "PGADR" to "PGADDR". In Table 7, added PGADDR. In Table 11, removed the Mcount 7 row. In Section 1.7.13.5, deleted "The ETU clock is held in reset condition until the activation sequence begins (either by VCCOK = 1 or VCCTMR timeout) and will go high ½ the ETU period thereafter." In Section 1.7.13.5 (number 3), deleted "If CLKOFF/SCLKOFF is high and SYCKST is set=1(STXCtl, b7=1), Rlen=max will stop the clock at the selected (CLKLVL or SCLKLVL) level." In Section 1.7.13.5, deleted number 9.