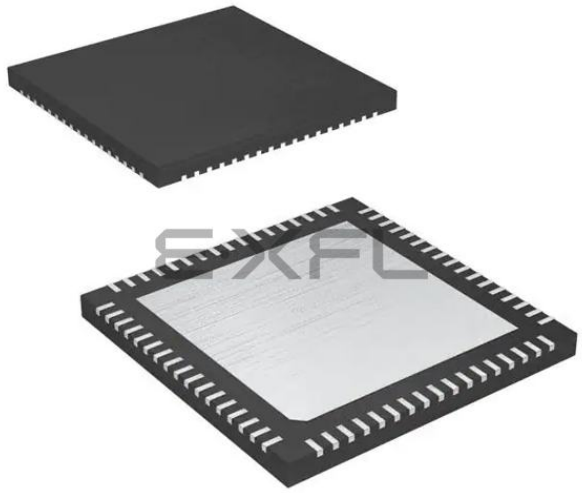
Analog Devices Inc./Maxim Integrated - 73S1215F-68IM/F Datasheet



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, USB
Peripherals	LED, POR, WDT
Number of I/O	9
Program Memory Size	64KB (64K 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/73s1215f-68im-f

Table of Contents

Pin Name	Pin (68 Qfn)	Pin (44 Qfn)	Type	Equivalent Circuit*	Description
LED(3:0) 0 1 2 3	 1 3 2 4	 3 4	IO	Figure 36	Special output drivers, programmable pull-down current to drive LEDs. May also be used as inputs.
RXD	17	11	I	Figure 33	Serial UART Receive data pin.
TXD	18	12	O	Figure 30	Serial UART Transmit data pin.
INT3	51		I	Figure 33	General purpose interrupt input.
INT2	52	32	I	Figure 33	General purpose interrupt input.
SIO	50	31	IO	Figure 29	IO data signal for use with external Smart Card interface circuit such as 73S8024.
SCLK	48	30	O	Figure 30	Clock signal for use with external Smart Card interface circuit.
PRES	64	43	I	Figure 41	Smart Card presence. Active high. Note: the pin has a very weak pull down resistor. In noisy environments, an external pull down may be desired to insure against a false card event.
PRES	56	35	I	Figure 42	Smart Card presence. Active low. Note: the pin has a very weak pull up resistor. In noisy environments, an external pull up may be desired to insure against a false card event.
CLK	57	36	O	Figure 39	Smart card clock signal.
RST	59	38	O	Figure 39	Smart card Reset signal.
IO	63	42	IO	Figure 40	Smart card Data IO signal.
AUX1	62	41	IO	Figure 40	Auxiliary Smart Card IO signal (C4).
AUX2	61	40	IO	Figure 40	Auxiliary Smart Card IO signal (C8).
VCC	60	39	PSO		Smart Card VCC supply voltage output. A 0.47µF capacitor is required and should be located at the smart card connector. The capacitor should be a ceramic type with low ESR.
GND	58	37	GND		Smart Card Ground.
VPC	55	34	PSI		Smart Card LDO regulator power supply source. A 10µF and a 0.1µF capacitor are required at the VPC input. The 10µF capacitor should be a ceramic type with low ESR.
TBUS(3:0) 0 1 2 3	 53 49 47 43		IO		Trace bus signals for ICE.
RXTX	45	28	IO		ICE control.
ERST	40	25	IO		ICE control.
ISBR	68		IO		ICE control.
TCLK	41	26	I		ICE control.

1.2 Hardware Overview

The Teridian 73S1215F single smart card controller integrates all primary functional blocks required to implement a smart card reader. Included on chip are an 8051-compatible microprocessor (MPU) which executes up to one instruction per clock cycle (80515), a fully integrated IS0-7816 compliant smart card interface, expansion smart card interface, full speed USB 2.0 compatible interface, serial interface, I2C interface, 6 x 5 keypad interface, 4 LED drivers, RAM, FLASH memory, a real time clock (RTC), and a variety of I/O pins. Figure 1 shows a functional block diagram of the 73S1215F.

1.3 80515 MPU Core

1.3.1 80515 Overview

The 73S1215F includes an 80515 MPU (8-bit, 8051-compatible) that performs most instructions in one clock cycle. The 80515 architecture eliminates redundant bus states and implements parallel execution of fetch and execution phases. Normally a machine cycle is aligned with a memory fetch, therefore, most of the 1-byte instructions are performed in a single cycle. This leads to an 8x performance (average) improvement (in terms of MIPS) over the Intel 8051 device running at the same clock frequency.

Actual processor clocking speed can be adjusted to the total processing demand of the application (cryptographic calculations, key management, memory management, and I/O management) using the XRAM special function register MPUCKCtl.

Typical smart card, USB, serial, keyboard, I2C, and RTC management functions are available for the MPU as part of the Teridian standard library. A standard ANSI “C” 80515-application programming interface library is available to help reduce design cycle. Refer to the *73S12xxF Software User’s Guide*.

1.3.2 Memory Organization

The 80515 MPU core incorporates the Harvard architecture with separate code and data spaces. Memory organization in the 80515 is similar to that of the industry standard 8051. There are three memory areas: Program memory (Flash), external data memory (XRAM), and internal data memory (IRAM). Data bus address space is allocated to on-chip memory as shown Table 2.

Table 2: MPU Data Memory Map

Address (hex)	Memory Technology	Memory Type	Typical Usage	Memory Size (bytes)
0000-FFFF	Flash Memory	Non-volatile	Program and non-volatile data	64KB
0000-07FF	Static RAM	Volatile	MPU data XRAM	2KB
FC00-FFFF	External SFR	Volatile	Peripheral control	1KB

Note: The IRAM is part of the core and is addressed differently.

Program Memory: The 80515 can address up to 64KB of program memory space from 0x0000 to 0xFFFF. Program memory is read when the MPU fetches instructions or performs a MOVC operation. After reset, the MPU starts program execution from location 0x0000. The lower part of the program memory includes reset and interrupt vectors. The interrupt vectors are spaced at 8-byte intervals, starting from 0x0003 (Reset is located at 0x0000).

Flash Memory: The program memory consists of flash memory. The flash memory is intended to primarily contain MPU program code. Flash erasure is initiated by writing a specific data pattern to specific SFR registers in the proper sequence. These special pattern/sequence requirements prevent inadvertent erasure of the flash memory.

Program Status Word (PSW):

Table 9: PSW Register Flags

MSB							LSB
CV	AC	F0	RS1	RS	OV	–	P

Table 10: PSW Bit Functions

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 11) 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).

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

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

Table 19: 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.

1.7.3.1 Interrupt Overview

When an interrupt occurs, the MPU will vector to the predetermined address as shown in Table 33. 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 REIT 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 20: 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.

Interrupt Request Register (IRCON): 0xC0 ← 0x00

Table 25: The IRCON Register

MSB							LSB
–	–	EX6	IEX5	IEX4	IEX3	IEX2	–

Bit	Symbol	Function
IRCON.7	–	
IRCON.6	–	
IRCON.5	IEX6	External interrupt 6 flag.
IRCON.4	IEX5	External interrupt 5 flag.
IRCON.3	IEX4	External interrupt 4 flag.
IRCON.2	IEX3	External interrupt 3 flag.
IRCON.1	IEX2	External interrupt 2 flag.
IRCON.0	–	

1.7.3.3 External Interrupts

The external interrupts (external to the CPU core) are connected as shown in Table 26. Interrupts with multiple sources are OR'ed together and individual interrupt source control is provided in XRAM SFRs to mask the individual interrupt sources and provide the corresponding interrupt flags. Multifunction USR [7:0] pins control Interrupts 0 and 1. Dedicated external interrupt pins INT2 and INT3 control interrupts 2 and 3. The polarity of interrupts 2 and 3 is programmable in the MPU. Interrupts 4, 5 and 6 have multiple peripheral sources and are multiplexed to one of these three interrupts. The peripheral functions will be described in subsequent sections. Generic 80515 MPU literature states that interrupts 4 through 6 are defined as rising edge sensitive. Thus, the hardware signals attached to interrupts 4, 5 and 6 are converted to rising edge level by the hardware.

SFR (special function register) enable bits must be set to permit any of these interrupts to occur. Likewise, each interrupt has its own flag bit that is set by the interrupt hardware and is reset automatically by the MPU interrupt handler.

Table 26: External MPU Interrupts

External Interrupt	Connection	Polarity	Flag Reset
0	USR I/O High Priority	see USRIntCtlx	Automatic
1	USR I/O Low Priority	see USRIntCtlx	Automatic
2	External Interrupt Pin INT2	Edge selectable	Automatic
3	External Interrupt Pin INT3	Edge selectable	Automatic
4	Smart Card Interrupts	N/A	Automatic
5	USB, RTC and Keypad	N/A	Automatic
6	I^2C, V_{DD}_Fault, Analog Comp	N/A	Automatic

Note: Interrupts 4, 5 and 6 have multiple interrupt sources and the flag bits are cleared upon reading of the corresponding register. To prevent any interrupts from being ignored, the register containing multiple interrupt flags should be stored temporary to allow each interrupt flag to be tested separately to see which interrupt(s) is/are pending.

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

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

Table 36: 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 37: 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	–	

1.7.8 Real-Time Clock with Hardware Watchdog (RTC)

Figure 9 shows the block diagram of the Real Time Clock. The RTC block uses the 32768Hz oscillator signal and divider logic to produce 0.5-second time marks. The time marks are used to create interrupts at intervals from 0.5 seconds to 8 seconds as selected by RTC Interval (RTCINV(2:0)). The 32768Hz oscillator can be disabled but is intended to operate at all times and in all power consumption modes. If a 32kHz crystal is not provided, the 32kHz oscillator should be disabled and the RTC will operate from MCLK (96MHz) divided by 2930 (refer to the oscillator and clock generation section). The clock generated by the high speed oscillator will not yield exactly 32768 Hz, but a frequency of approximately 32764.505119 Hz. This yields a negative 106.6 PPM (1 / 9375) error with respect to 32768Hz. The RTC circuit provides hardware to compensate for this error by providing an offset circuit that will adjust the RTC counter.

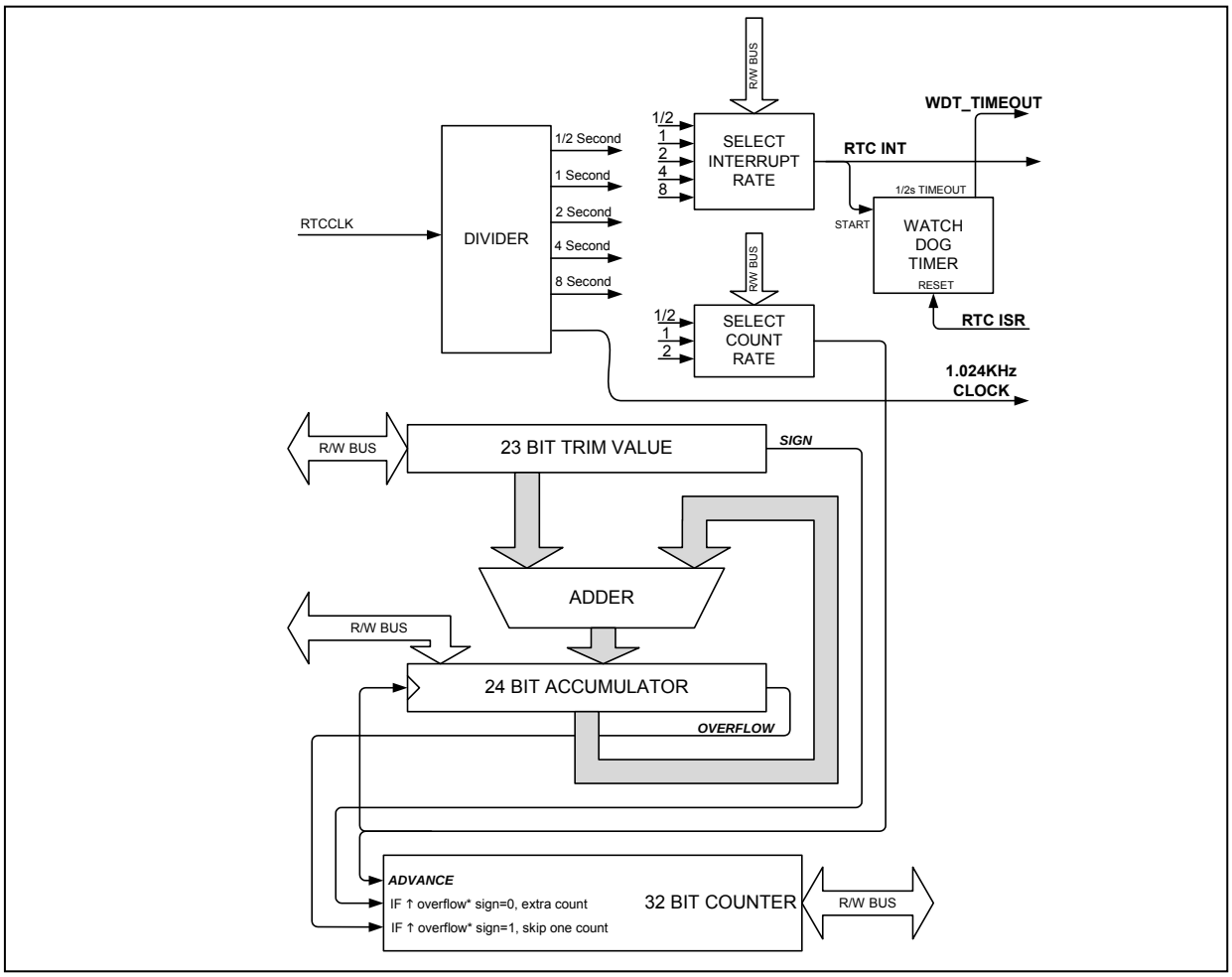


Figure 9: Real Time Clock Block Diagram

1.7.11 I²C Master Interface

The 73S1215F includes a dedicated fast mode, 400kHz I²C Master interface. The I²C interface can read or write 1 or 2 bytes of data per data transfer frame. The MPU communicates with the interface through six dedicated SFR registers:

- Device Address (DAR)
- Write Data (WDR)
- Secondary Write Data (SWDR)
- Read Data (RDR)
- Secondary Read Data (SRDR)
- Control and Status (CSR)

The DAR register is used to set up the slave address and specify if the transaction is a read or write operation. The CSR register sets up, starts the transaction and reports any errors that may occur. When the I²C transaction is complete, the I²C interrupt is reported via external interrupt 6. The I²C interrupt is automatically de-asserted when a subsequent I²C transaction is started. The I²C interface uses a 400kHz clock from the time-base circuits.

1.7.11.1 I²C Write Sequence

To write data on the I²C Master Bus, the 80515 has to program the following registers according to the following sequence:

1. Write slave device address to Device Address register (DAR). The data contains 7 bits for the slave device address and 1 bit of op-code. The op-code bit should be written with a 0 to indicate a write operation.
2. Write data to Write Data register (WDR). This data will be transferred to the slave device.
3. If writing 2 bytes, set bit 0 of the Control and Status register (CSR) and load the second data byte to Secondary Write Data register (SWDR).
4. Set bit 1 of the CSR register to start I²C Master Bus.
5. Wait for I²C interrupt to be asserted. It indicates that the write on I²C Master Bus is done. Refer to information about the INT6Ctl, IEN1 and IRCON register for masking and flag operation.

Device Address Register (DAR): 0xFF80 ← 0x00

Table 63: The DAR Register

MSB							LSB
DVADR.6	DVADR.5	DVADR.4	DVADR.3	DVADR.2	DVADR.1	DVADR.0	I2CRW

Bit	Symbol	Function
DAR.7	DVADR [0:6]	Slave device address.
DAR.6		
DAR.5		
DAR.4		
DAR.3		
DAR.2		
DAR.1		
DAR.0	I2CRW	If set = 0, the transaction is a write operation. If set = 1, read.

I2C Write Data Register (WDR): 0XFF81 ← 0x00

Table 64: The WDR Register

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

Bit	Function
WDR.7	Data to be written to the I^2C slave device.
WDR.6	
WDR.5	
WDR.4	
WDR.3	
WDR.2	
WDR.1	
WDR.0	

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

Table 69: 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.

Keypad Scan Time Register (KSIZE): 0xD5 ← 0x00

This register is not applicable when HWSCEN is not set. Unused row inputs should be connected to VDD.

Table 74: The KSIZE Register

MSB							LSB
–	–	ROWSIZ.2	ROWSIZ.1	ROWSIZ.0	COLSIZ.2	COLSIZ.1	COLSIZ.0

Bit	Symbol	Function
KSIZE.7	–	
KSIZE.6	–	
KSIZE.5	ROWSIZ.2	Defines the number of rows in the keypad. Maximum number is 6 given the number of row pins on the package. Allows for a reduced keypad size for scanning.
KSIZE.4	ROWSIZ.1	
KSIZE.3	ROWSIZ.0	
KSIZE.2	COLSIZ.2	Defines the number of columns in the keypad. Maximum number is 5 given the number of column pins on the package. Allows for a reduced keypad size for scanning.
KSIZE.1	COLSIZ.1	
KSIZE.0	COLSIZ.0	

Keypad Column LS Scan Order Register (KORDERL): 0xD6 ← 0x00

In registers KORDERL and KORDERH, Column Scan Order(14:0) is grouped into 5 sets of 3 bits each. Each set determines which column (COL(4:0) pin) to activate by loading the column number into the 3 bits. When in HW_Scan_Enable mode, the hardware will step through the sets from 1Col to 5Col (up to the number of columns in Colsize) and scan the column defined in the 3 bits. To scan in sequential order, set a counting pattern with 0 in set 0, and 1 in set 1,and 2 in set 2, and 3 in set 3, and 4 in set 4. The firmware should update this as part of the interrupt service routine so that the new scan order is loaded prior to the next key being pressed. For example, to scan COL(0) first, 1Col(2:0) should be loaded with 000'b. To scan COL(4) fifth, 5Col(2:0) should be loaded with 100'b.

Table 75: The KORDERL Register

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

Bit	Symbol	Function
KORDERL.7	3COL.1	Column to scan 3rd (lsb's).
KORDERL.6	3COL.0	
KORDERL.5	2COL.2	Column to scan 2nd.
KORDERL.4	2COL.1	
KORDERL.3	2COL.0	
KORDERL.2	1COL.2	Column to scan 1st.
KORDERL.1	1COL.1	
KORDERL.0	1COL.0	

The USB interface includes a Serial Interface Engine (SIE) that handles NRZI encoding/decoding, bit stuffing / unstuffing, and CRC generation/checking. It also generates headers for packets to be transmitted and decodes the headers of received packets. An analog transceiver interfaces with the external USB bus. The USB interface hardware performs error checking and removes the USB protocol fields from the incoming messages before passing the data to the firmware. The hardware also adds the USB protocol fields to the outgoing messages coming from the firmware. The hardware implements NRZI encoding/decoding, CRC checking/generation (both on data and token packets), device address decoding, handshake packet generation, Data0/Data1 toggle synchronization, bit stuffing, bus idle detection and other protocol generation/checking required in Chapter 8 of the *Universal Serial Bus Specification, Revision 2.0*.

The firmware is responsible for servicing and building of the messages required under Chapter 9 of the *Universal Serial Bus Specification, Revision 2.0*. Device configuration is stored in the firmware. Data received from the USB port is stored in the appropriate IN FIFO that is read by the firmware and processed. The messages to be sent back to the USB host are generated by firmware and placed back into the appropriate OUT FIFO. Stall/NAK handshakes are generated as appropriate if the RAM is not available for another message from the USB host. Suspend and resume modes are supported. All register/FIFO spaces are located in Data Memory space. The FIFOs are dedicated for USB storage and are unused in a configuration that is not using USB. All registers in the USB interface are located in external data memory address (XRAM) space starting at address FC00'h.

1.7.15 Smart Card Interface Function

The 73S1215F 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 15 shows the simplified block diagram of the card circuitry (UART + interfaces), with detail of dedicated XRAM registers.

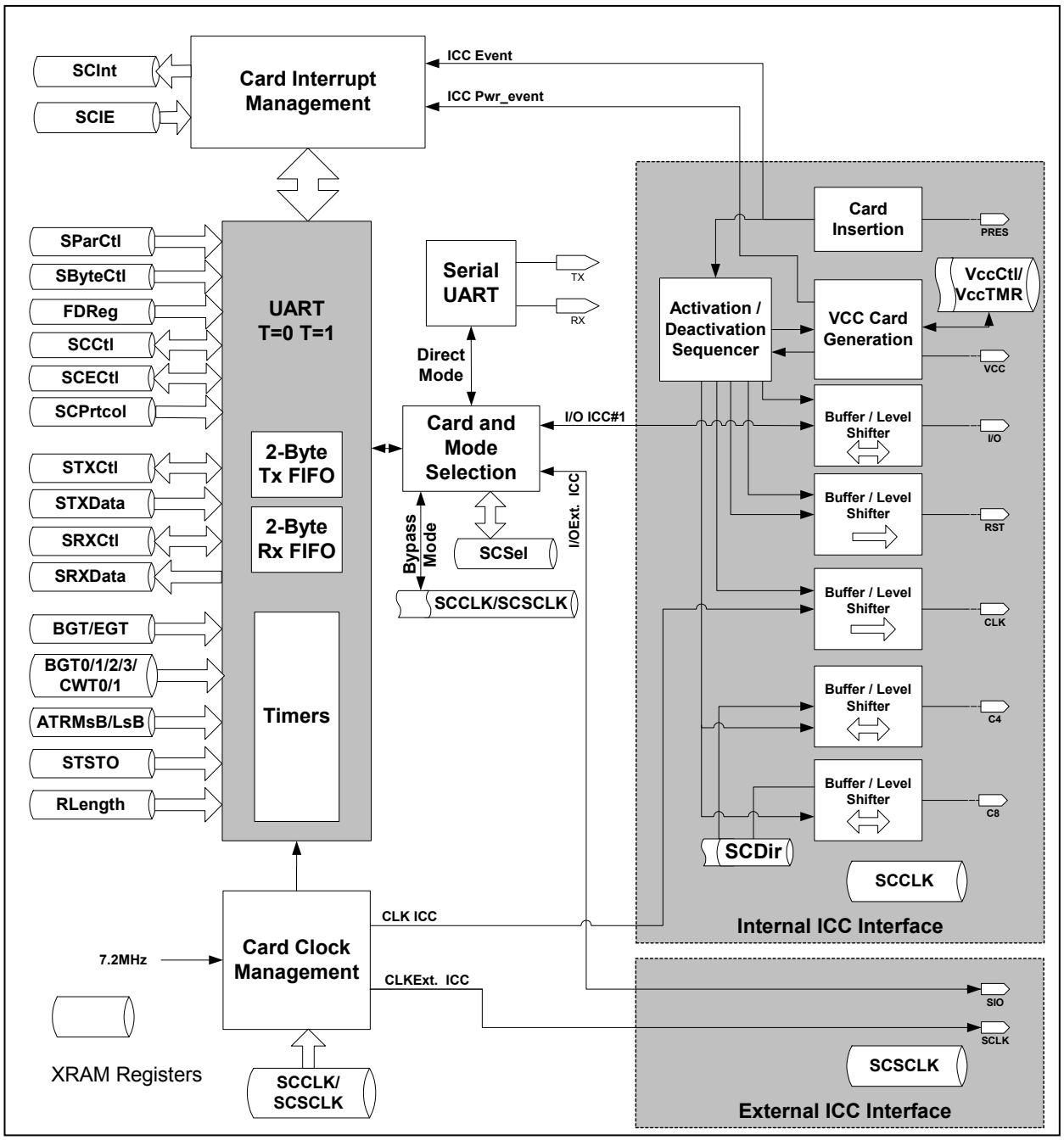
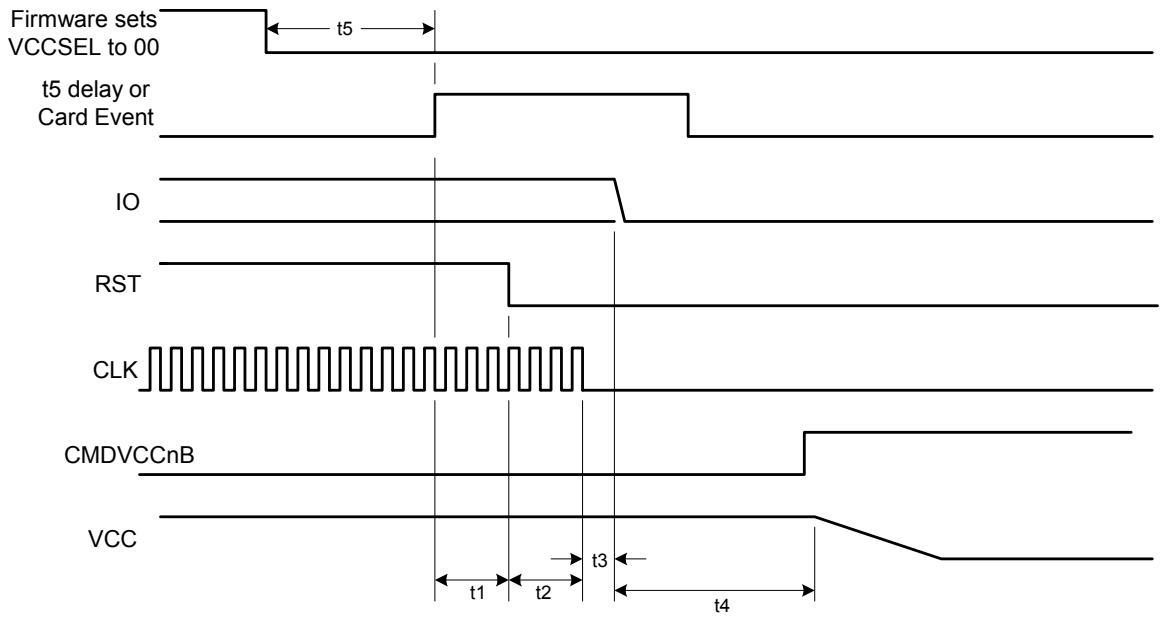


Figure 15: Smart Card Interface Block Diagram

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



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 18: Deactivation Sequence

1.7.15.3 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 19 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.

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

Table 117: 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	I2CMODE		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	SCISYN	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)							

4 Equivalent Circuits

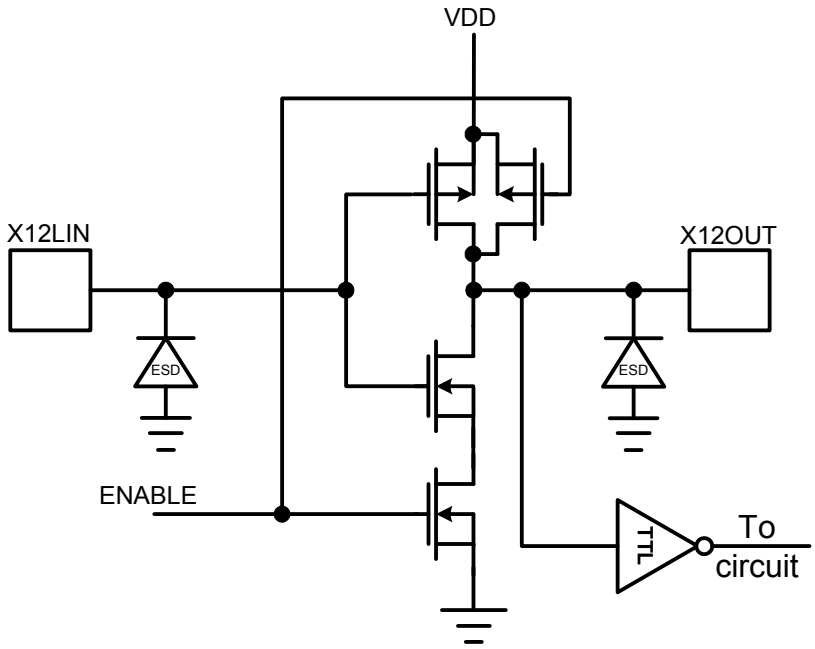


Figure 27: 12 MHz Oscillator Circuit

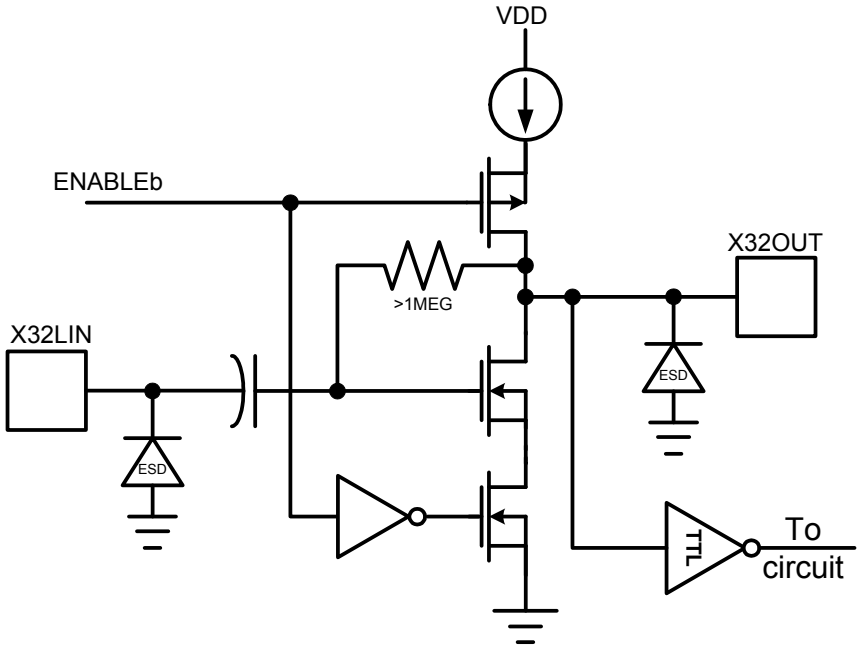


Figure 28: 32kHz Oscillator Circuit

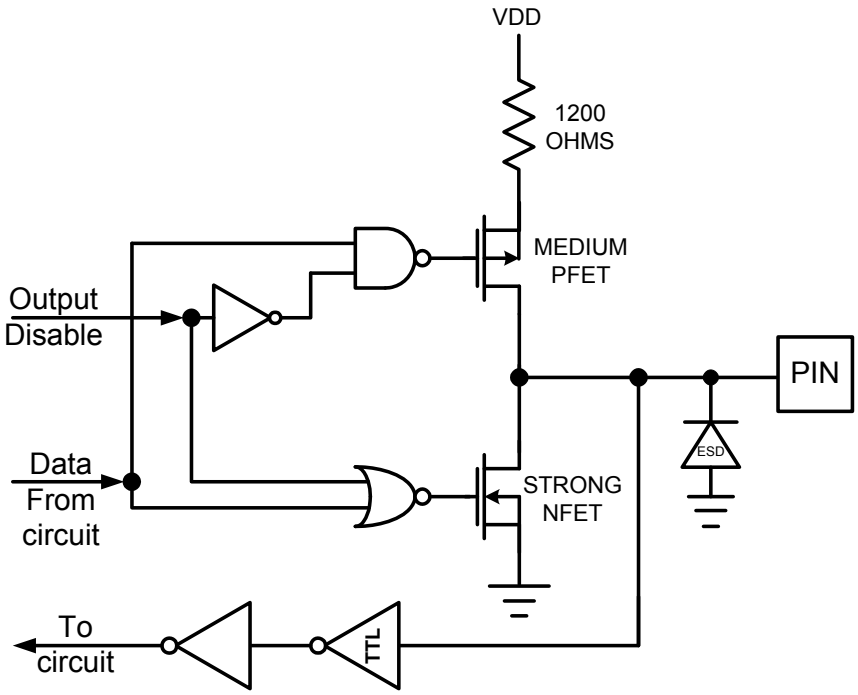


Figure 35: Keypad Column Circuit

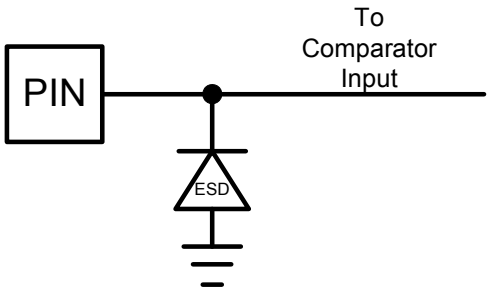


Figure 38: Analog Input Circuit

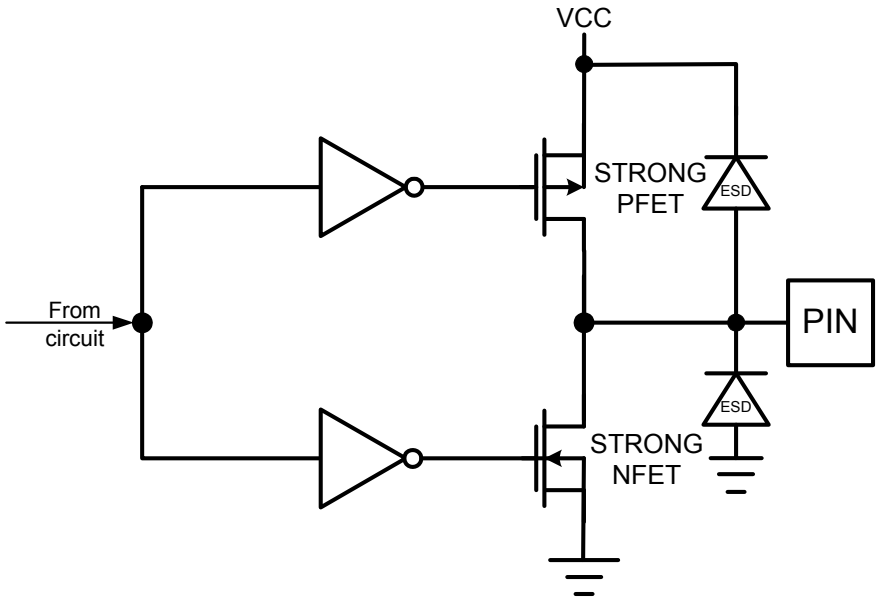


Figure 39: Smart Card Output Circuit