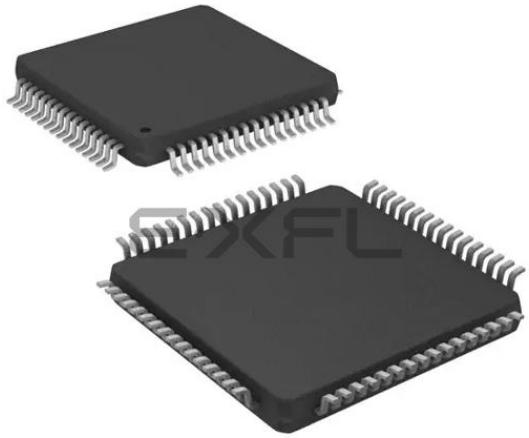

Welcome to E-XFL.COM

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	Active
Core Processor	ARM® Cortex®-M0
Core Size	32-Bit Single-Core
Speed	48MHz
Connectivity	Ethernet/TCP/IP, I²C, SSP, UART/USART
Peripherals	DMA, POR, PWM, WDT
Number of I/O	34
Program Memory Size	128KB (32K x 32)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	16K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 3.6V
Data Converters	A/D 9x12b
Oscillator Type	External, Internal
Operating Temperature	0°C ~ 70°C (TA)
Mounting Type	Surface Mount
Package / Case	64-LQFP
Supplier Device Package	64-TQFP (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/wiznet/w7500p

List of table

IRQ[13]	Dualtimer1	Dualtimer1 global interrupt	0x0000_0074
IRQ[14]	PWM0	PWM0 global interrupt	0x0000_0078
IRQ[15]	PWM1	PWM1 global interrupt	0x0000_007C
IRQ[16]	PWM2	PWM2 global interrupt	0x0000_0080
IRQ[17]	PWM3	PWM3 global interrupt	0x0000_0084
IRQ[18]	PWM4	PWM4 global interrupt	0x0000_0088
IRQ[19]	PWM5	PWM5 global interrupt	0x0000_008C
IRQ[20]	PWM6	PWM6 global interrupt	0x0000_0090
IRQ[21]	PWM7	PWM7 global interrupt	0x0000_0094
IRQ[22]	reserved		0x0000_0098
IRQ[23]	ADC	ADC acquisition end interrupt	0x0000_009C
IRQ[24]	TCPIP	TCPIP global interrupt	0x0000_00A0
IRQ[25]	EXT_INT	External pin interrupt	0x0000_00A4
IRQ[26]	reserved		0x0000_00A8
IRQ[27]	reserved		0x0000_00AC
IRQ[28]	reserved		0x0000_00B0
IRQ[29]	reserved		0x0000_00B4
IRQ[30]	reserved		0x0000_00B8
IRQ[31]	reserved		0x0000_00BC

4.3 Event

W7500P is able to handle internal events in order to wake up the core(WFE). The wakeup event can be generated by

- When after DMA process finished

5 Power supply

5.1 Introduction

W7500P embeds a voltage regulator in order to supply the internal 1.5V digital power domain.

- Require a 2.7V ~ 5.5V operating supply voltage (VDD)
- ADC ref voltage is same as VDD

5.2 Voltage regulator

The voltage regulator is always enabled after Reset and works in only one mode.

- In Run mode, the regulator supplies full power to the 1.5V domain.
- There is no power down or sleep mode

Two of them are almost the same except the clock gated peripherals kinds. Table 2 shows the Sleep mode summary.

Table 2 W7500P sleep mode summary

Mode	Entry	Wakeup	Effect on clocks
Sleep mode	DEEPSLEEP = 0 Enable WFI	Any interrupt	CPU clock OFF APB Bus Clock ON AHB Bus clock ON Memory clocks ON
	DEEPSLEEP = 0 Enable WFE	Wakeup event	
Deep Sleep mode	DEEPSLEEP = 1 Enable WFI	Any interrupt	CPU clock OFF APB Bus Clock OFF AHB Bus clock OFF Memory clocks OFF
	DEEPSLEEP = 1 Enable WFE	Wakeup event	

5.4.2 Peripheral clock gating

In Run mode, individual clocks can be stopped at any time to reduce power.

Peripheral clock gating is controlled by the CRG block.

Below is the list of clocks which can be gating in CRG block.

- ADC clock (ADCCLK)
- SSP0, SSP1 clock (SSPCLK)
- UART0, UART1 clock (UARTCLK)
- Two Timer clocks (TIMCLK0, TIMCLK1)
- 8ea PWM clocks (PWMCLK0 ~ PWMCLK7)
- WDOG clock (WDOGCLK)
- Random number generator clock (RNGCLK)

6 System tick timer

6.1 Introduction

System tick timer(SysTick) is part of the ARM Cortex-M0 core

6.2 Features

Simple 24bit timer.

Clocked internally by the system clock or the system clock/2.

9.2.4 Flash program operation

The main Flash memory can be programmed word, half word, or 1 byte at a time by SZ bit of FACCR. The program operation is started when the CPU writes a data into a main Flash memory address with the WRI or WR bit of FCTRLR register set.

The main Flash memory programming sequence in standard mode is as below:

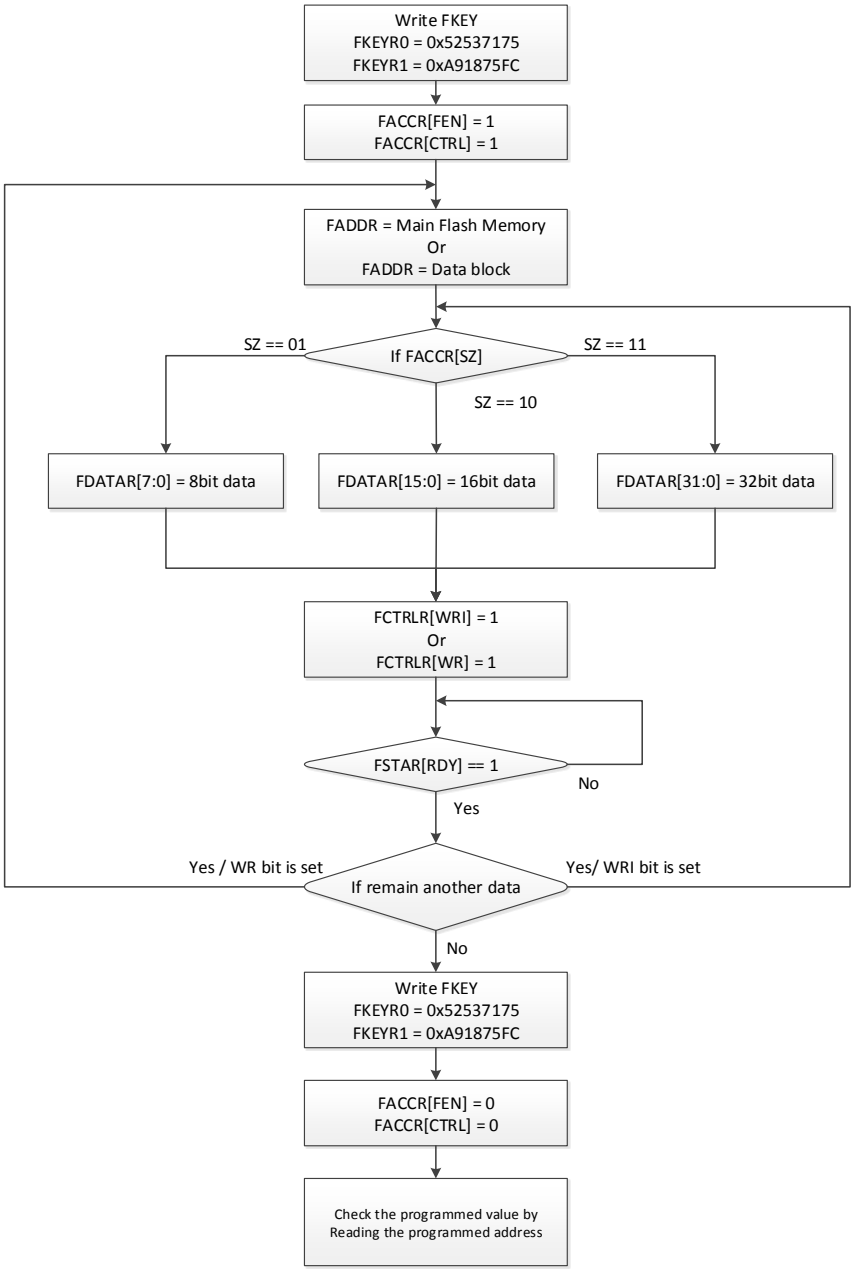

Figure 9. main Flash memory programming sequence

1. Check that no main Flash memory operation is ongoing by checking the RDY bit in the FSTATR register.

10 Clock Reset generator (CRG)

10.1 Introduction

CRG is clock reset generator block for W7500P System. It provides every clock/reset for all other block include CPU and peripherals. CRG includes PLL and POR.

10.2 Features

10.2.1 Reset

- Three types of reset – external reset, Power reset, system reset
- External reset is generated by low level on the RSTn pin (external reset)
- Power reset is generated by Power-on reset (POR)
- Power on reset is generated by POR
- System reset is generated when one of the following events occurs
  - Watchdog event
  - After remapping
  - Software reset (SYSRESETREQ bit in Cortex-M0. Refer to the Cortex-M0 technical reference manual for more detail)
- Power reset sets all registers to their reset values.
- System reset sets all registers to their reset values except the CRG block registers and remap register to protect remap value

10.2.2 Clock

Two clock sources can be used to drive the system clock.

- External oscillator clock (8MHz ~ 24MHz) (OCLK)
- Internal 8MHz RC oscillator clock (RCLK)

One additional clock source

- 32.768KHz low speed external crystal which derives the real time clock.

There is a PLL

One PLL is integrated

- Input clock range is from 8MHz to 24MHz
- Frequency can be generated by M/N/OD registers. (refer register description)
- Bypass option enabled

There are many generated clocks for independent operating with system clock

- System clock (FCLK)
- ADC clock (ADCCLK)
- SSP0, SSP1 clock (SSPCLK)
- UART0, UART1 clock (UARTCLK)

- Two Timer clocks (TIMCLK0, TIMCLK1)
- 8ea PWM clocks (PWMCLK0 - PWMCLK7)
- Real time clock (RTCCLK)
- WDOG clock (WDOGCLK)
- Random number generator clock (RNGCLK)

RNGCLK have only one source (pll output) and no prescaler

Some of the generated clocks turn off automatically when CPU enters sleep mode.

- ADCCLK, RNGCLK

Generate two Hardware TCPIP Clocks (MII_RXC, MII_TXC) are from external PADs.

Hardware TCPIP Clocks can be gated by register control.

All clocks generated from CRG can be monitored.

10.3 Functional description

Figure 10 shows the CRG block diagram.

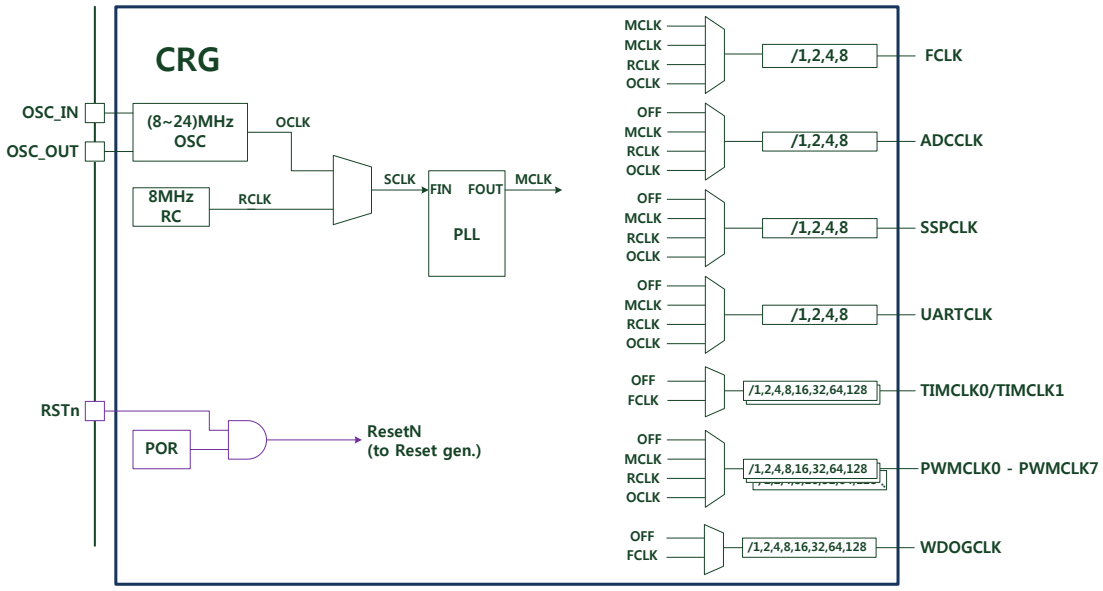

Figure 10 CRG block diagram

10.3.1 External Oscillator Clock

External oscillator clock (OCLK) can be generated from two possible clock source

- External crystal/ceramic resonator (8 to 24MHz external oscillator)
- User external clock

Table 7 shows the two clock sources of external oscillator clock

Table 7 External oscillator clock sources

	External clock	Crystal/

11.3.1 Operation RNG

Figure 12 shows the flowchart of RNG operation.

A random number is automatically generated after powering on reset,

Follow the procedure below to manually generate a random number.

1. Change MODE to start/stop by register.
2. Change clock source / seed value / polynomial value if need.
3. Run and Stop the RNG.
4. Read Random value.

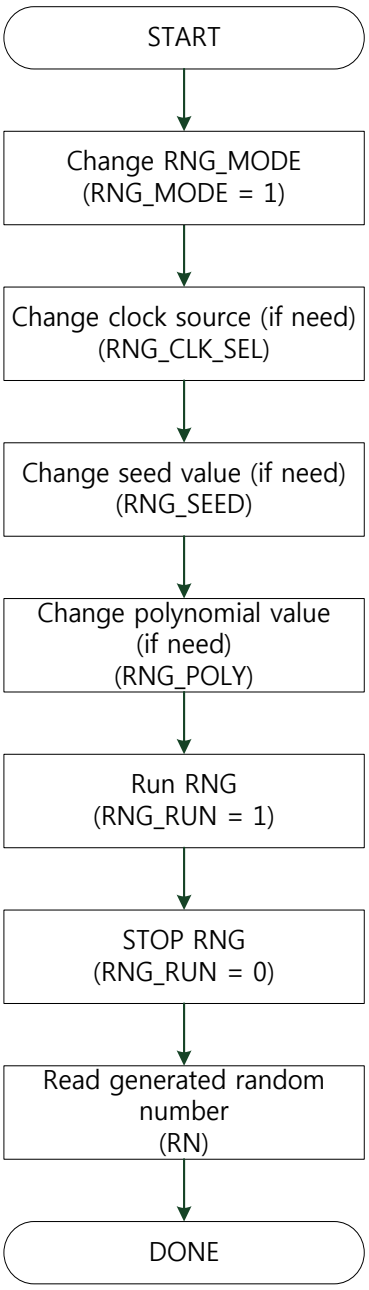

Figure 12. Flow chart of RNG operation

PB_06	15	DUP	GPIOB_6		
PC_00	53	U_CTS1	GPIOC_0	PWM0/CAP0	
PC_01	54	U_RTS1	GPIOC_1		
PC_02	55	U_TXD1	GPIOC_2	PWM2/CAP2	
PC_03	56	U_RXD1	GPIOC_3		
PC_04	57	SCL1	GPIOC_4	PWM4/CAP4	
PC_05	58	SDA1	GPIOC_5	PWM5/CAP5	
PC_06	11	GPIOC_6	GPIOC_6	U_TXD2	
PC_08	1	PWM0/CAP0	GPIOC_8	SCL0	AIN7
PC_09	2	PWM1/CAP1	GPIOC_9	SDA0	AIN6
PC_10	3	U_TXD2	GPIOC_10	PWM2/CAP2	AIN5
PC_11	4	U_RXD2	GPIOC_11	PWM3/CAP3	AIN4
PC_12	5	AIN3	GPIOC_12	SSEL0	AIN3
PC_13	6	AIN2	GPIOC_13	SCLK0	AIN2
PC_14	7	AIN1	GPIOC_14	MISO0	AIN1
PC_15	8	AIN0	GPIOC_15	MOSI0	AIN0

13 External Interrupt (EXTI)

13.1 Introduction

Each functional pads are connected to the external interrupt(EXTINT) source.

13.2 Features

- All functional pads can be used as an external interrupt source regardless of any set of pad function.
- External Interrupt controller has the following functions and can be controlled by registers.
  - Interrupt mask (enable or disable, default : disable)
  - Interrupt polarity (rising or falling, default : rising)

13.3 Functional description

All pads are connected to the control register individually. (External interrupt mask register and External Interrupt polarity register)

External interrupt working as following expression:

- Each pad interrupt = Interrupt mask & (Interrupt polarity ^ Pad input)
- EXTINT = any Each pad interrupt

Figure 13 shows the External Interrupt diagram.

In this mode, the controller can be configured to use either the primary or the alternate channel control data structure. After the channel is enabled and the controller receives a request for this channel, the flow for the auto-request cycle is as below:

1. The controller performs 2^R transfers. If the number of transfers remaining is zero the flow continues at step 3.
2. The controller arbitrates if there are any transfers remaining after 2^R transfers. If the current channel c has the highest priority, the cycle continues at step 1.

The controller sets dma_done[c] signal for this channel HIGH for one system clock cycle. This indicates to the host processor that the DMA cycle is complete.

16.3.3.4 Ping-pong cycle

In this mode, the controller performs a DMA cycle using one of the data structures and then performs a DMA cycle using the other data structure. The controller continues to switch between primary and alternate structures until it reads a data structure that is invalid, until the user reprograms the cycle_type to basic, or until the host processor disables the channel.

In ping-pong mode, the user can program or reprogram one of the two channel data structures (primary or alternate) while using the other channel data structure for the active transfer. When a transfer is done, the next transfer can be started immediately using the prepared channel data structure – provided that a higher priority channel does not require servicing. If the user does not reprogram the channel control data structure not in use for a transfer, the cycle type remains invalid (which is the value at the end of the last transfer using that structure), and the ping-pong cycle completes.

The ping-pong cycle can be used for transfers to or from peripherals or for memory- to-memory transfers.

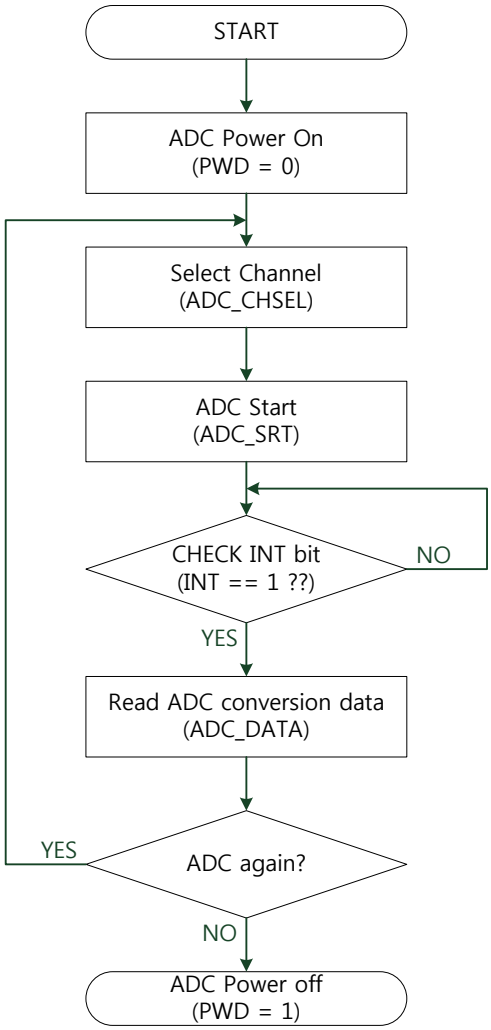

Figure 23. The ADC operation flowchart with non-interrupt

19.3 Functional description

19.3.1 Clock and clock enable

The dual timers contain PCLK and TIMERCLK clock inputs. PCLK is the main APB system clock and is used by the register interface. TIMERCLK is the input to the prescale units and the decrementing counters. PCLK and TIMERCLK are synchronous.

The dual timers consist two programmable 32-bit Free-Running Counters(FRC) which operate independently. The two timers operate from one TIMERCLK but Each FRC is controlled independently by individual clock enable.

19.3.2 Timer size

Users can select FRC as 16-bit or 32-bit using the control register.

19.3.3 Prescaler

The timer has a prescaler that can divide down the enabled clock rate by 1, 16 or 256.

19.3.4 Repetition mode

There are two repetition mode: one-shot and wrapping mode. Wrapping mode has two modes: free-running and periodic mode.

One-shot mode

The counter generates an interrupt once. When the counter reaches 0, it halts until users reprogram it. Users can do this as below:

- Clear the one-shot count bit in the control register, in which case the count proceeds according to the selection of wrapping mode(free-running or periodic mode).
- Write a new value to the Load Value register.

Wrapping mode

Free-running mode

The counter wraps after reaching its zero value, and continues to count down from the maximum value. This is the default mode.

Periodic mode

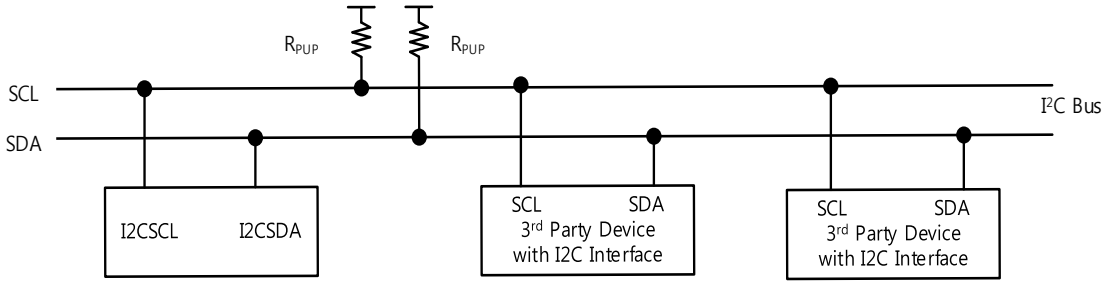

Figure 45. I2C Bus Configuration

Figure 46 shows the I2C block diagram.

In addition to receiving and transmitting data, this interface converts it from serial to parallel format and vice versa. The interrupt is enabled or disabled by software. The interface is connected to the I^2C bus by a data pin (SDA) and by a Clock pin (SCL). It can be connected with a standard (up to 100 KHz) I^2C bus.

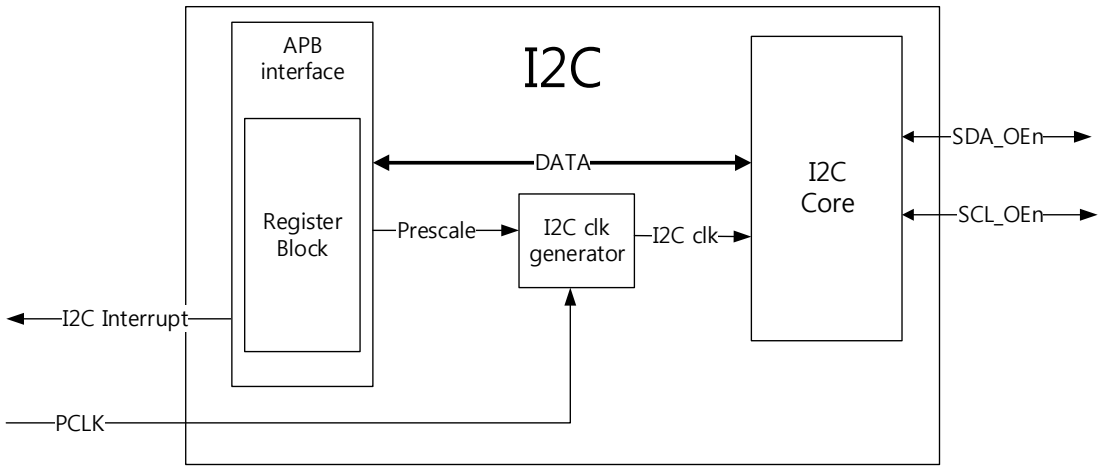

Figure 46. I2C block diagram

SDA is the bi- directions serial data line and SCL is the bi-directions serial clock line. The bus is considered idle when both lines are high. Every transaction on the I^2C bus is nine bits long, consisting of eight data bits and a single acknowledge bit and data must be transferred MSB first.

21.3.1 Data validity

The data on the SDA line must be stable during the HIGH period of the SCL. The HIGH or LOW state of the data line can only change when the clock signal on the SCL line is LOW (Figure 47). One clock pulse is generated for each data bit transferred.

Figure 55 shows the operation of repeated START.

The repeated START operates for data read operation execution.

The operation sequences are Slave address, send data, repeated START, and send data.

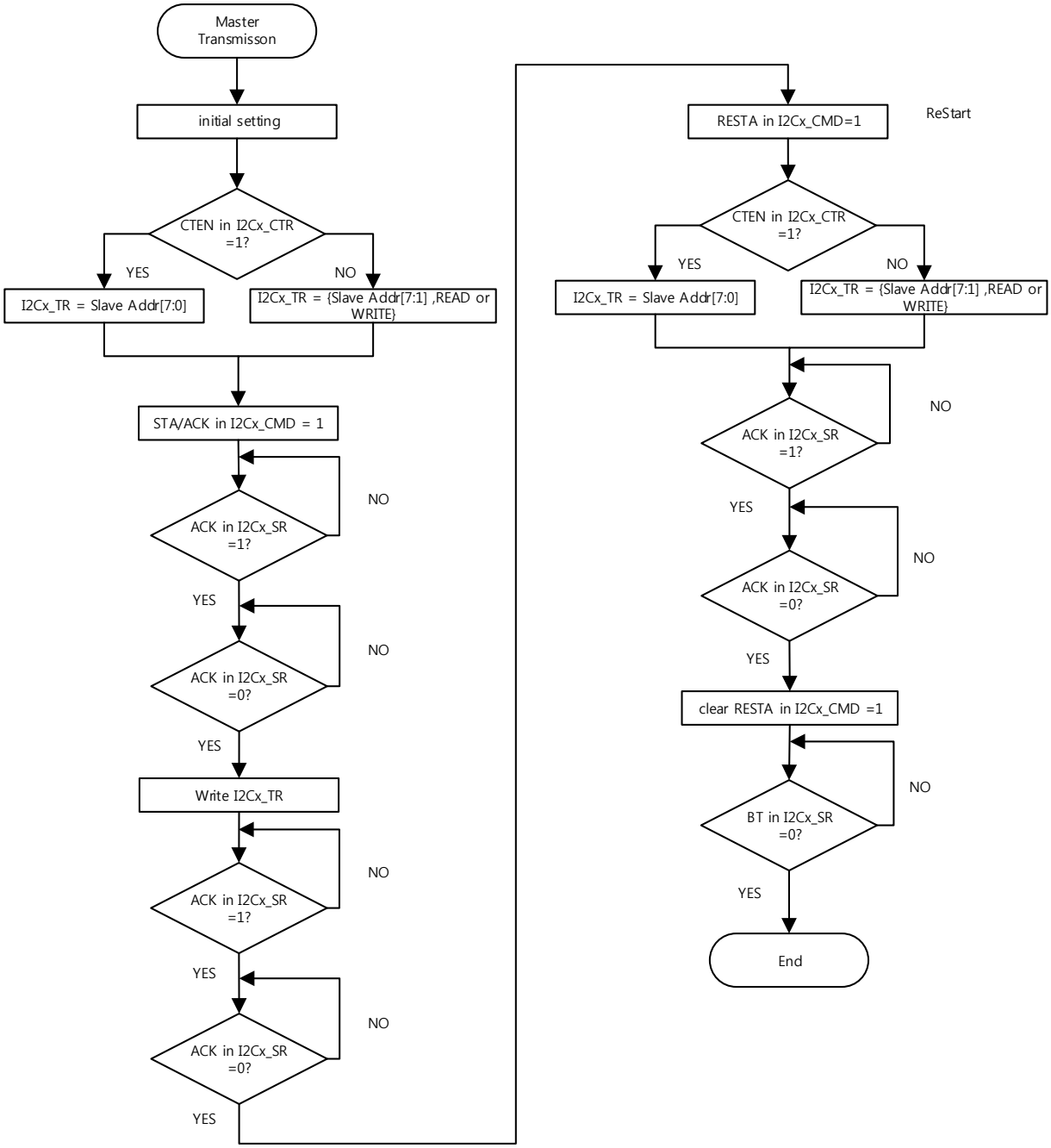

Figure 55. Master Transmit with Repeated START

22.3.2 Data transmission

Data transmitted is stored in a 32-byte FIFOs. Transmit data is written into the transmit FIFO for transmission. If UART is enabled, it causes a data frame to start transmitting with parameters indicated in the UARTxLCR_H.

Data continues to transmit until there is no data left in the transmit FIFO. The BUSY bit of UARTxFR is '1' as soon as data is written to the transmit FIFO, which means the FIFO is not empty, and remains as '1' while data is being transmitted.

22.3.3 Data receive

Received data is stored in the 32-byte FIFOs. When a start bit has been received, it begins running and data is sampled on the eighth cycle of that counter in UART mode. A valid stop bit is confirmed if UARTRXD is '1'. When a full word is received, the data is stored in the receive FIFO. Error bit is stored in bit[10:8] of UARTxCR and overrun is stored in bit[11] of UARTxCR.

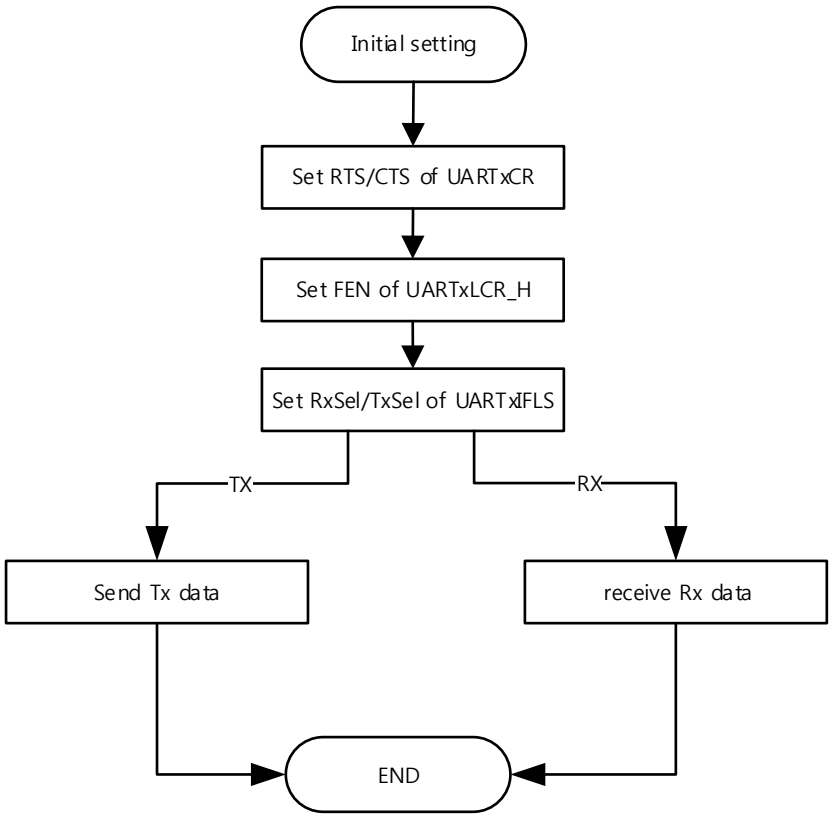

Figure 61. Transmit and Receive data flow chart

22.3.4 Hardware flow control

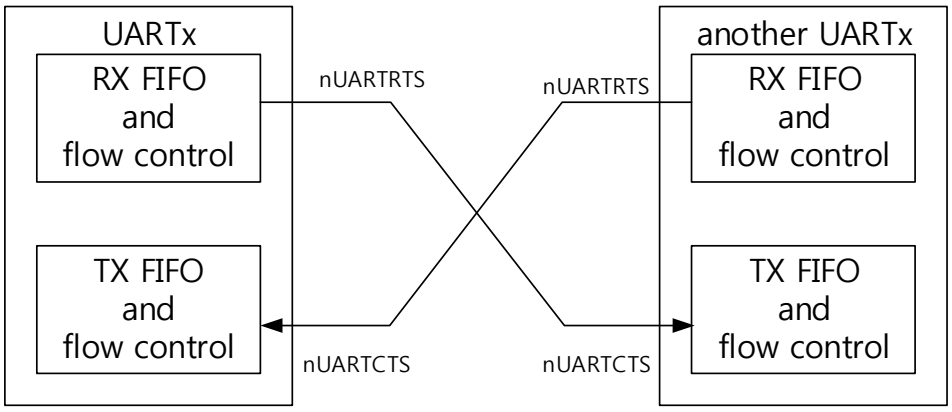

Figure 62. Hardware flow control description

The RTS flow control is enabled by setting the RTSen of UARTxCR. If RTS is enabled, the data flow is controlled as follows.

When the receiver FIFO level reaches the programmed trigger level, nUARTRTS(pin) is asserted(to a low value). nUARTRTS is reasserted(to a low level) once the receiver FIFO has reached the previous trigger level. The reasserted of nUARTRTS signals to the sending UART to continue transmitting data.

The CTS flow control is enabled, the transmitter can only transmit data when nUARTCTS is asserted. When nUARTCTR is re-asserted(to a low) the transmitter sends the next byte. To stop the transmitter from sending the following byte, nUARTCTS must be released before the middle of the last stop bit that is currently being sent.

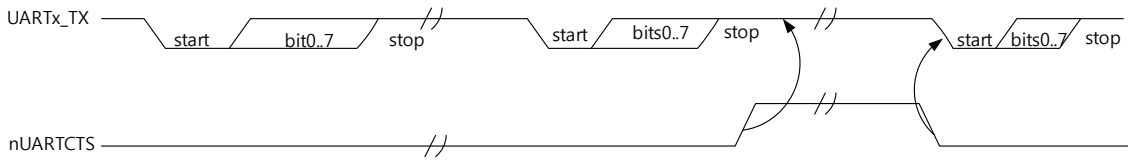

Figure 63. CTS Functional Timing

23.3.8 Enable PrimeCell SSP operation

You can either prime the transmit FIFO, by writing up to eight 16-bit values when the PrimeCell SSP is disabled, or permit the transmit FIFO service request to interrupt the CPU. Once enabled, transmission or reception of data begins on the transmit, SSPTXD, and receive, SSPRXD, pins.

23.3.9 Clock ratios

There is a constraint on the ratio of the frequencies of PCLK to SSPCLK. The frequency of SSPCLK must be less or equal to that of PCLK. This ensures that control signals from the SSPCLK domain to the PCLK domain are guaranteed to get synchronized before one frame duration:
FSSPCLK <= FPCLK.

In the slave mode of operation, the SSPCLKIN signal from the external master is double-synchronized and then delayed to detect an edge. It takes three SSPCLKs to detect an edge on SSPCLKIN. SSPTXD has less setup time to the falling edge of SSPCLKIN on which the master is sampling the line.

The setup and hold times on SSPRXD, with reference to SSPCLKIN, must be more conservative to ensure that it is at the right value when the actual sampling occurs within the SSPMS. To ensure correct device operation, SSPCLK must be at least 12 times faster than the maximum expected frequency of SSPCLKIN.
The frequency selected for SSPCLK must accommodate the desired range of bit clock rates.
The ratio of minimum SSPCLK frequency to SSPCLKOUT maximum frequency in the case of the slave mode is 12, and for the master mode, it is two.

To generate a maximum bit rate of 1.8432Mbps in the master mode, the frequency of SSPCLK must be at least 3.6864MHz. With an SSPCLK frequency of 3.6864MHz, the SSPCPSR register must be programmed with a value of 2, and the SCR[7:0] field in the SSPCR0 register must be programmed with a value of 0.

To work with a maximum bit rate of 1.8432Mbps in the slave mode, the frequency of SSPCLK must be at least 22.12MHz. With an SSPCLK frequency of 22.12MHz, the SSPCPSR register can be programmed with a value of 12 and the SCR[7:0] field in the SSPCR0 register can be programmed with a value of 0. Similarly, the ratio of SSPCLK maximum frequency to SSPCLKOUT minimum frequency is 254 x 256.

The minimum frequency of SSPCLK is calculated by the following equations, both of which must be satisfied:

$F_{SSPCLK}(min) => 2 \times F_{SSPCLKOUT}(max)$, for master mode

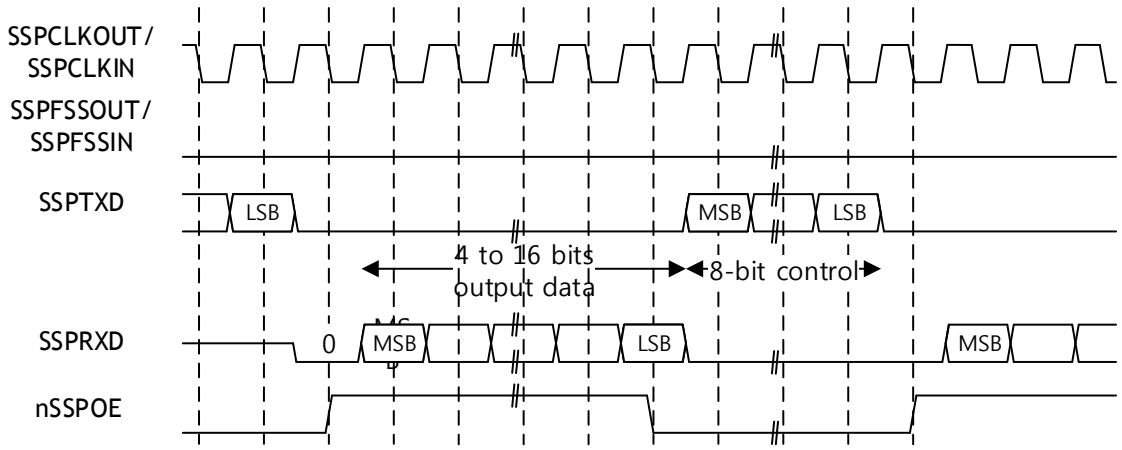

Figure 76. National Semiconductor Microwire frame format, continuous transfers

23.3.16 Master and Slave configurations

Figure 77 shows how a PrimeCell SSP (PL022) configured as master, interfaces to a Motorola SPI slave. The SPI Slave Select (SS) signal is permanently tied LOW and configures it as a slave. Similar to the above operation, the master can broadcast to the slave through the master PrimeCell SSP SSPTXD line. In response, the slave drives its SPI MISO port onto the SSPRXD line of the master.

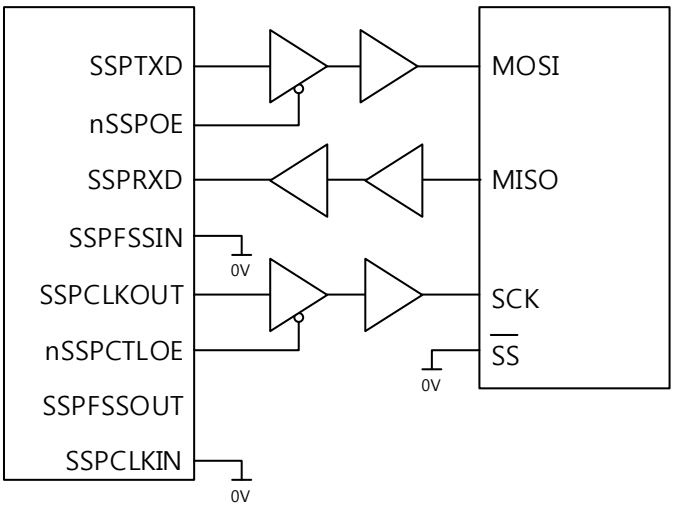

Figure 77. PrimeCell SSP master coupled to an SPI slave

Figure 78 shows a Motorola SPI configured as a master and interfaced to an instance of a PrimeCell SSP (PL022) configured as a slave. In this case, the slave Select Signal (SS) is permanently tied HIGH to configure it as a master. The master can broadcast to the slave through the master SPI MOSI line. In response, the slave drives its nSSPOE signal LOW.
This enables its SSPTXD data onto the MISO line of the master.

t_{VD_DAT}	Data valid time		3.5		1.0	us
t_{SU_DAT}	Data setup time	200		90		ns
t_{VD_ACK}	Data valid acknowledge time		3.5		1.0	us
t_{HD_STA}	Hold time START condition	3.8		0.5		us
t_{SU_STA}	Set-up time for a repeat START condition	4.5		0.5		us
t_{SU_STO}	Set-up time for STOP condition	3.8		0.5		us

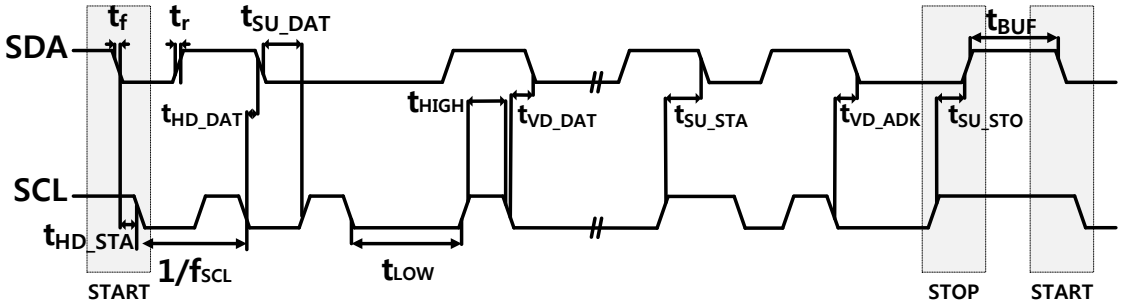

Figure 83. I2C bus AC waveform

25.8 SSP Interface Characteristics

Table 26 shows the SSP characteristics of W7500P.

Table 26 SSP characteristics

Symbol	Parameter	Conditions	Min	Max	Unit
f_{SCL}	SSP clock frequency	Master mode		20	MHz
		Slave mode		20	MHz
t_{r_SCK}	SSP clock rising and fall time	Capacitive load : C = 25pF		8	ns
t_{SU_M}	Data input setup time	Master mode	5		ns
t_{SU_S}		Slave mode	6		ns
t_{H_M}	Data input hold time	Master mode	5		ns
t_{H_S}		Slave mode	6		ns
t_{V_M}	Data output valid time	Master mode		20	ns
t_{V_S}		Slave mode		5	ns
t_{H_M}	Data output hold time	Master mode	13		
t_{H_S}		Slave mode	3		
DuCy	SPI slave input clock duty cycle	Slave mode	45	55	%

Document History Information

Version	Date	Descriptions
Ver. 1.0.0	07SEP2015	Initial Release
Ver. 1.0.1	18NOV2015	Corrected Package diagram error
Ver. 1.0.2	08JAN2016	1. Bit rate in SPI interface changed ‘up to 20MHz and higher’ to ‘up to 20MHz’. 2. Bit rate generation example changed in 23.3.11 Programming the SSPCR1 Control Register.
Ver. 1.0.3	05FEB2016	Modified the problem with none mention of 2nd Function on PB_06 in Table 28 functional description table
Ver. 1.0.4	08MAR2016	1. In 2.2.2Memory map, GPIO0,1,2,3 changed to GPIOA,B,C,D 2. In Table 1 W7500P interrupt assignments, GPIO0,1,2,3 changed to GPIOA,B,C,D 3. In Table 8 functional description table, GPIO1,2,3 changed to GPIOA,B,C 4. In Table 12. W7500P Pin Description, GPIO1,2,3 changed to GPIOA,B,C
Ver. 1.0.5	18JUL2017	1. In Table 7 External oscillator clock sources, switching External clock Schematic with Crystal/Ceramic resonators Schematic.
Ver. 1.0.6	19JUL2017	1. Modified T_A, T_J temperature range in Table 16 General operating conditions.