HS6209 2.4G Wireless SOC

V0.3

Liuc

2019/4/17

Revision History

Revision	Date	Description
0.1	2018.08.18	Initial version
0.2	2018.10.28	Update GPIO
0.3	2019.04.17	Update package information

CONTENTS

1 Introduction

HS6209 is a member of the low-cost, high-performance family of intelligent 2.4 GHz RF transceivers with embedded microcontrollers. The HS6209 is optimized to provide a single chip solution for Ultra Low Power (ULP) wireless applications. The combination of processing power, memory, low power oscillators, real-time counter,and arange of power saving modes provides an ideal platform for implementation of RF protocols. Benefits of using HS6209 include tighter protocol timing, lower power consumption and improved co-existence performance. For the application layer HS6209 offers USB and GPIO ports.

1.1 Package

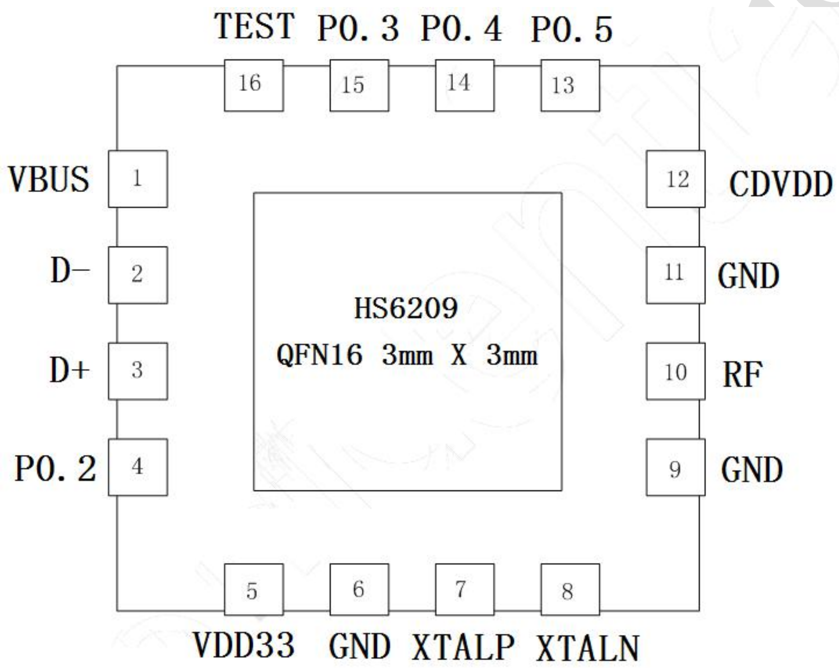


Figure1.1 HS6209 QFN16

Table1.1 HS6209 QFN16 pin function

Pin#	Name	Description
1	VBUS	5V power, need 3.3 ohm resistor in series and 4.7uF capacitor to ground, 4.7uF capacitor nearby VBUS pin
2	D-	USB data-
3	D+	USB data+
4	P0.2	General purpose IO
5	VDD33	3.3V LDO output， connected to 2.2uF capacitor
6	GND	Ground
7	XTALP	Crystal+
8	XTALN	Crystal-

9	GND	Ground
10	RF	RF
11	GND	Ground
12	CDVDD	Digital 1.2v LDO output, connected to 100nF , CDVDD can be NC for low cost
13	P0.5	General purpose IO
14	P0.4	General purpose IO
15	P0.3	General purpose IO
16	TEST	Test pin, TEST=0 for normal work

2 Product overview

2.1 Features

Features of HS6209 include:

- Fast 8-bit microcontroller:
  - Intel MCS 51 compliant instruction set
  - Reduced instruction cycle time, up to 12 times compared to legacy 8051
- Memory:
  - Program memory: 16 kB of OTP memory with security features
  - Internal RAM: 256Byte
- A number of on-chip hardware resources are available through programmable multi-purpose input/output pins:
  - 3GPIO
  - External interrupts
  - USB
- High performance 2.4 GHz RF transceiver
  - True single chip GFSK transceiver
  - Enhanced ShockBurst™ link layer support in HW:
    - Packet assembly/disassembly
    - Address and CRC computation
    - Auto ACK and retransmit
  - On the air data rate 1 Mbps or 2 Mbps
  - 125 RF channels operation, with 79 (2.402 GHz –2.480 GHz) channels within 2.400–2.4835 GHz
  - Short switching time enable frequency hopping
- On-chip timers:
  - Two 16-bit timers/counters operating at the system clock (sources from the 16 MHz on-chip oscillators)
    - Timer 0, Timer 1 are compatible to standard 8051
- On-chip oscillators:
  - 16 MHz crystal oscillator XOSC16M
- Power management function:
  - Low power design supporting fully static stop/ standby
  - Programmable MCU clock frequency from 125 kHz to 16 MHz

3 RF transceiver

HS6209uses the same 2.4 GHz GFSK RF transceiver with embedded protocol engine. The RF transceiver is designed for operation in the world wide ISM frequency band at 2.400–2.4835 GHz and is very well suited for ultra-low power wireless applications.

The RF transceiver module is configured and operated through the RF transceiver map. This register map is accessed by the MCU through a dedicated on-chip Serial Peripheral interface (SPI) and is available in all power modes of the RF transceiver module. The embedded protocol engine enables data packet communication and supports various modes from manual operation to advanced autonomous protocol operation. Data FIFOs in the RF transceiver module ensure a smooth data flow between the RF transceiver module and HS6209 MCU.

The rest of this chapter is written in the context of the RF transceiver module as the core and the rest of HS6209 as external circuitry to this module.

3.1 Features

Features of the RF transceiver include:

- General
  - Worldwide 2.4 GHz ISM band operation
  - Common antenna interface in transmit and receive
  - GFSK modulation
  - 1 Mbpsand 2 Mbps on air data rate
- Transmitter
  - Programmable output power: 8, 0, -6, -12 or -32dBm
  - 18.5mA at 0dBm output power
- Receiver
  - Integrated channel filters
  - 19.5mA at 2 Mbps
  - -83dBm sensitivity at 2 Mbps
  - -88dBm sensitivity at 1 Mbps
- RF Synthesizer
  - Fully integrated synthesizer
  - 1 MHz frequency programming resolution
  - Accepts low cost ±60ppm 16 MHz crystal
  - 1 MHz non-overlapping channel spacing at 1 Mbps
  - 2 MHz non-overlapping channel spacing at 2 Mbps
- Protocol Engine
  - 1 to 32 bytes dynamic payload length
  - Automatic packet handling (assembly/disassembly)
  - Automatic packet transaction handling (auto ACK, auto retransmit)
- 6 data pipe MultiSlave for 6:1 star networks

3.2 Block

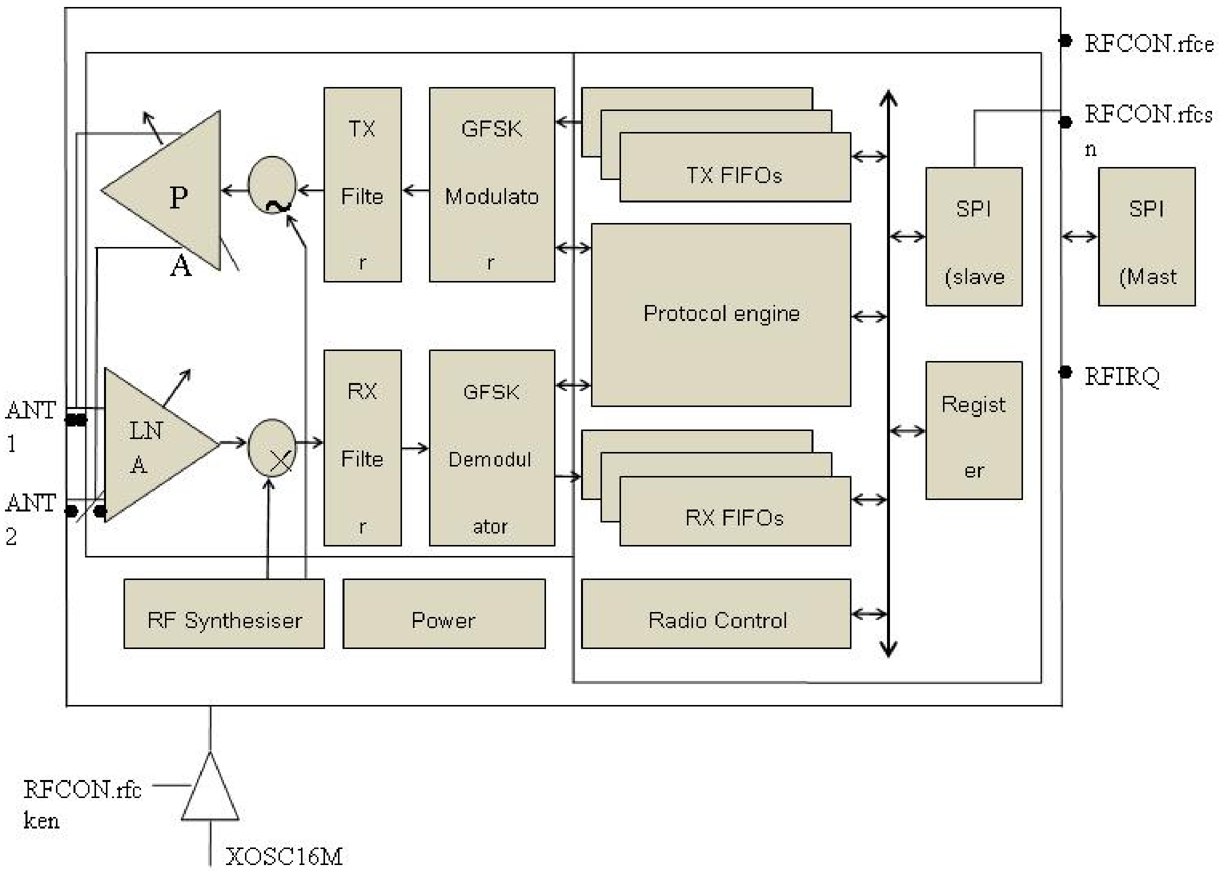


Fig.3.1 RF transceiver block diagram

3.3 Functional description

This section describes the different operating modes of the RF transceiver and the parameters used to control it.

The RF transceiver module has a built-in state machine that controls the transitions between the different operating modes. The state machine is controlled by SFR register RFCON and RF transceiver register CONFIG.

3.4 Operational Modes

HS6209 has a built-in state machine that controls the transitions between the chip's operating modes. The state machine takes input from user defined register values and internal signals.

The state diagrams in Figure 3.2 shows the operating modes and how theyfunction. There are three types of distinct states highlighted in the state diagram:

- Recommended operating mode: is a recommended state used during normal operation.
- Possible operating mode: is a possible operating state, but is not used during normal operation.
- Transition state: is a time limited state used during start upof the oscillator and settling of the PLL.

When the VDD reaches 1.8V or higher HS6209 enters the Power on reset state where it remains in reset until entering the PowerDown mode.

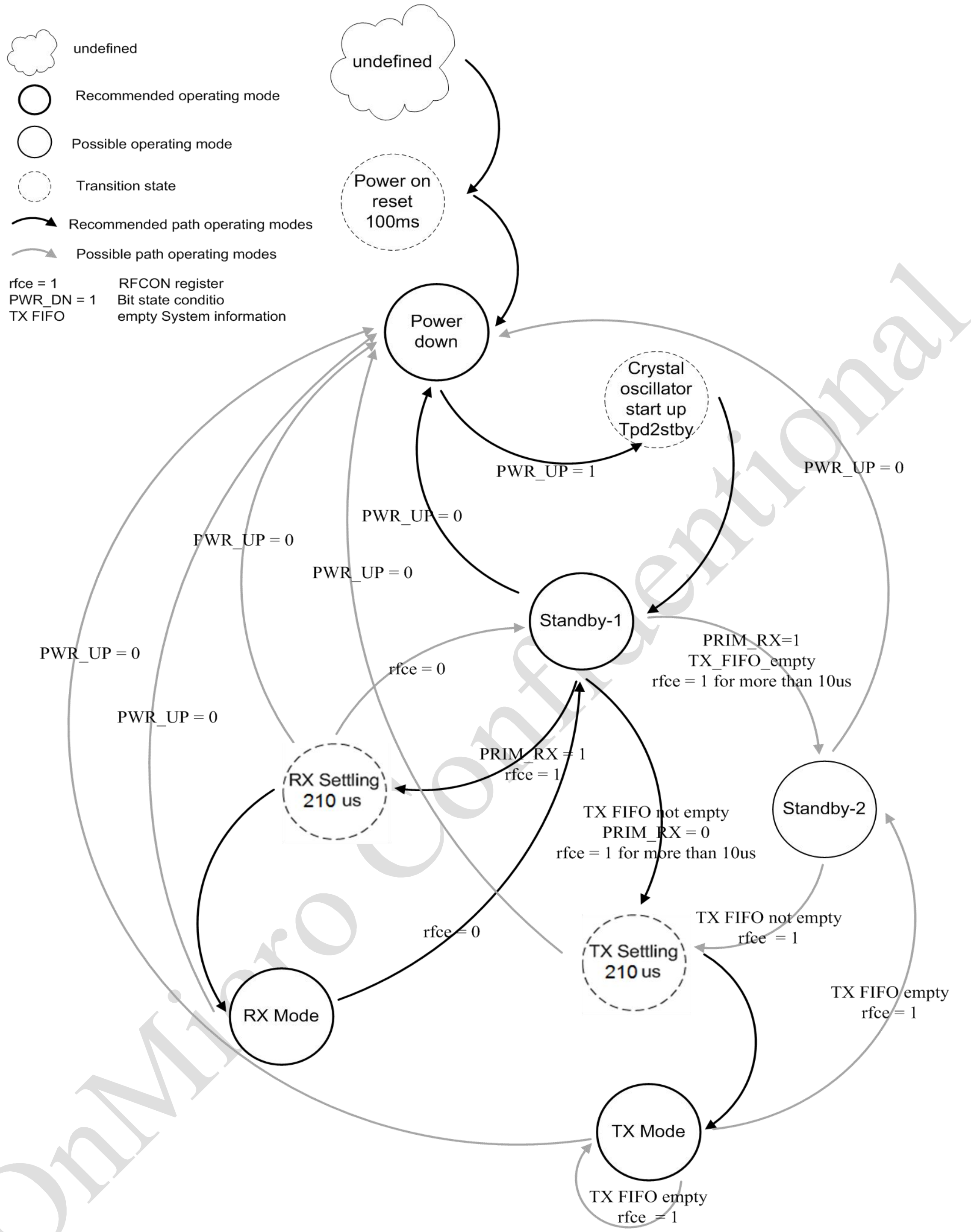


Fig. 3.2 Radio control state diagram

3.4.1 PowerDown Mode

In power down mode, the HS6209is disabled using minimal current consumption. All register values available are maintained and the SPI is kept active, enabling change of configuration and the uploading/down-loading of data registers.

3.4.2 Standby Modes

3.4.2.1 Standby-I mode

When the chip is power up or wake up from deep sleep or register retention mode, the device enters standby-I mode. Standby-I mode is used to minimizeaverage current consumption while maintaining short start up times. In this mode only part of the crystal oscillator is active. Change to active modes only happens if CE is set high and when CE is set low, the HS6209 returns to standby-I mode from both the TX and RX modes.

3.4.2.2 Standby-II mode

In standby-II mode extra clock buffers are active and more current is used compared to standby-I mode. The HS6209enters standby-II mode if CE is held high on a PTX device with an empty TX FIFO. If a new packet is uploaded to the TX FIFO, the PLL immediately starts and the packet is transmitted after the normal PLL settling delay (210µs). Register values are maintained and the SPI can be activated during both standby modes.

Notice: From Standby-I mode to standby-II mode CE more than 20us

3.4.3 RX mode

The RX mode is an active mode where the HS6209 radio is used as a receiver. To enter this mode, the chip must have PRIM_RX bit and the CE pin set high.

In RX mode the receiver demodulates the signals from the RF channel, constantly presenting the demodulated data to the baseband protocol engine. The baseband protocol engine constantly searches for a valid packet. If a valid packet is found (by a matching address and a valid CRC) the payload of the packet is presented in a vacant slot in the RX FIFOs. If the RX FIFOs are full, the received packet is discarded.

The chip remains in RX mode until the MCU configures it to standby-I mode or power down mode. However, if the automatic protocol features in the baseband protocol engine are enabled, the chip can enter other modes in order to execute the protocol.

In RX mode a Received Power Detector (RPD) signal is available. The RPD register is a signed number which corresponds to different level of received power in dBm. The highest bit of RPD is sign bit. The range of the received power is -100 ~ +10dBm. The RPD has about +/-5dBm deviation from the real level.

3.4.4 TX mode

The TX mode is an active mode for transmitting packets. To enter this mode, the chip must have the PWR_UP bit set high, PRIM_RX bit set low, a payload in the TX FIFO and a high pulse on the CE for more than 20µs.

The HS6209 stays in TX mode until it finishes transmitting a packet. If CE = 0, the chip returns to standby-I mode. If CE = 1, the status of the TX FIFO determines the next action. If the TX FIFO is not empty the HS6209 remains in TX mode and transmits the next packet. If the TX FIFO is empty the chip goes into standby-II mode. The HS6209 transmitter PLL operates in open loop when in TX mode. It is important never to keep the HS6209 in TX mode for more than 4ms at a time. If the protocol features are enabled, the chip is never in TX mode longer than 4ms.

3.4.5 Operational modes configuration

The following (table 1) describes how to configure the operational modes:

Mode	PWR_UP register	PRIM_RX register	CE input pin	FIFO state
RX mode	1	1	1	-
TX mode	1	0	1	Data TX FIFO. Will empty all level in TX FIFOs[a]
TX mode	1	0	Minimum 20us high pulse	Data TX FIFO. Will empty one level in TX FIFOs[b]
Standby-2	1	0	1	TX FIFO empty
Standby-1	1	-	0	No ongoing packet transmission
Power Down	0	-	-	-

Table 3.1 the HS6209 main modes

a. If CE is held high all TX FIFOs are emptied and all necessary ACK and possible retransmits are carried out. The transmission continues as long as the TX FIFO is refilled. If the TX FIFO is empty when the CE is still high, the chip enters standby-II mode. In this mode the transmission of a packet is started as soon as the CSN is set high after an upload (UL) of a packet to TX FIFO.

b. This operating mode pulses the CE high for at least 20µs. This allows one packet to be transmitted. This is the normal operating mode. After the packet is transmitted, the chip enters standby-1mode.

3.4.6 Timing Information

The timing information in this section relates to the transitions between modes and the timing for the CE pin. The transition from TX mode to RX mode or vice versa is the same as the transition from the standby modes to TX mode or RX mode (max. 210µs), as described in Table 3.2.

Name	The chip	Notes	Max.	Min.	Comments
$T_{pd2stby}$	Power down→Standby mode		150us		With external clock
		a	1.5ms		External crystal, Ls< 30mH
			3ms		External crystal, Ls< 60mH
			4.5ms		External crystal, Ls< 90mH
T_{stby2a}	Standby mode→TX/RX mode		210us		
T_{hce}	Minimum CE high			20us	
$T_{pece2csn}$	Delay from CE positive edge to CSN low			4us	

a. See crystal specifications.

Tab. 3.2 operational timing of the HS6209 chip

HS6209 to go from power down mode to TX or RX mode it must first pass through stand-by mode. There must be a delay of Tpd2stby (see Table 3.2.) after the chip leaves power down mode before theRFCON.rfce is set high.

Note: If VDD is turned off the register value is lost and you must configure chip before entering the TX or RX modes.

3.4.7 Air data rate

The air data rate is the modulated signaling rate the chip uses when transmitting and receiving data. It can be 500kbps, 1Mbps or2Mbps. Using lower air data rate gives better receiver sensitivity than higher air data rate. But, high air data rate gives lower average current consumption and reduced probability of on-air collisions.The air data rate is set by the RF_DR bit in the RF_SETUP register. A transmitter and a receiver must be programmed with the same air data rate to communicate with each other.

3.4.8 RF channel frequency

The RF channel frequency determines the center of the channel used by the chip. The channel occupies a bandwidth of less than 1MHz at 250kbps and 1Mbps and a bandwidth of less than 2MHz at 2Mbps. The chip can operate on frequencies from 2.400GHz to2.525GHz. The programming resolution of the RF channel frequency setting is 1MHz.

At 2Mbps the channel occupies a bandwidth wider than the resolution of the RF channel frequency setting. To ensure non-overlapping channels in 2Mbps mode, the channel spacing must be 2MHz or more. At 1Mbps the channel bandwidth isthe same as or lower than the resolution of the RF frequency.

The RF channel frequency is set by the RF_CH register according to the following formula:

$$F0 = 2400 + RF\_CH \text{ [MHz]}$$

You must program a transmitter and a receiver with the same RF channel frequency to communicate with each other.

3.4.9 Received Power Detector measurements

Received Power Detector (RPD), located in register 09, is a signed number which corresponds to different level of received power in dBm.The highest bit of RPD is sign bit. The range of the received power is -100 ~ +10dBm.

The RPD can be read out at any time while the chip is in received mode. This offers a snapshot of thecurrent received power level in the channel. The status of RPD is correct when RX mode is enabled and after a wait time of T_{stby2a} +T_{delay_AGC}= 210us + 20us. The RX gain varies over temperature which means that the RPD value also varies over temperature.

3.4.10 PA control

The PA (Power Amplifier) control is used to set the output power from the chip power amplifier. In TX mode PA control has four programmable steps, seeTable 3.3.

The PA control is set by the RF_PWR bits in theRF_SETUP register.

SPI RF-SETUP (RF_PWR)	RF output power	DC current consumption
1000	0dBm	18.5mA
0100	-6dBm	16mA
0010	-12dBm	14mA
0001	-18dBm	12mA

Conditions: VDD = 3.0V, VSS = 0V, TA = 27ºC

Note: set PA_PWR[3:0] to 1111 can obtain maximum +4dBm output power

Tab.3.3 RF output power setting for the HS6209

3.4.11 RX/TX control

The RX/TX control is set by PRIM_RX bit in the CONFIG register and sets the HS6209 chip in transmit/receive mode.

3.5 Protocol Engine

Protocol engine is a packet based data link layer that features automatic packet assembly and timing, automatic acknowledgement and retransmissions of packets. Protocol engine enables the implementation of ultralow power and high performance communication. The Protocol engine features enable significant improvements of power efficiency for bi-directional and uni-directional systems, without adding complexity on the host controller side.

3.5.1 Features

The main features of Protocol engine are:

- 1 to 32 bytes dynamic payload length
- Automatic packet handling
- Automatic packet transaction handling
  - Auto Acknowledgement with payload
  - Auto retransmit
- 6 data pipe for 1:6 star networks

3.6 Protocol engine overview

Protocol engine uses self-defined protocol for automatic packet handling and timing. During transmit, Protocol engine assembles the packet and clocks the bits in the data packet for transmission. During receive, Protocol engine constantly searches for a valid address in the demodulated signal. When Protocol engine finds a validaddress, it processes the rest of the packet and validates it by CRC. If the packet is valid the payload is moved into a vacant slot in the RX FIFOs. All high speed bit handling and timing is controlled by protocol engine.

Protocol engine features automatic packet transaction handling for the easy implementation of a reliable bi-directional data link. A protocol engine packet transaction is a packet exchange between two transceivers, with one transceiver acting as the Primary Receiver

(PRX) and the other transceiver acting as the Primary Transmitter (PTX). A protocol engine packet transaction is always initiated by a packet transmission from the PTX,the transaction is complete when the PTX has received an acknowledgment packet (ACK packet) from the PRX. The PRX can attach user data to the ACK packet enabling a bi-directional data link.

The automatic packet transaction handling works as follows:

1. You begin the transaction by transmitting a data packet from the PTX to the PRX. Protocol engine automatically sets the PTX in receive mode to wait for the ACK packet.
2. If the packet is received by the PRX, Protocol engine automatically assembles and transmits an acknowledgment packet (ACK packet) to the PTX before returning to receive mode.
3. If the PTX does not receive the ACK packet immediately, Protocol engine automatically retransmits the original data packet after a programmable delay and sets the PTX in receive mode to wait for the ACK packet.

In Protocol engine it is possible to configure parameters such as the maximum number of retransmits and the delay from one transmission to the next retransmission. All automatic handling is done without the involvement of the MCU.

3.7 Protocol engine packet format

The format of the Protocol engine packet is described in this section. The Protocol engine packet contains a preamble field, address field, packet control field, payload field and a CRC field. Figure3.3 shows the packet format with MSB to the left.

Preamble 1 byte	Address 4-5 byte	2byte guard	Packet control field 9 bit	Payload 0-32 bytes	CRC 1-2 bytes

Figure3.3 AProtocol engine packet with payload (0-32 bytes)

3.7.1 Preamble

The preamble is a bit sequence used to synchronize the receivers demodulator to the incoming bit stream. The preamble is one byte long and is either 01010101 or 10101010. If the first bit in the address is 1 the preamble is automatically set to 10101010 and if the first bit is 0 the preamble is automatically set to 01010101. This is done to ensure there are enough transitions in the preamble to stabilize the receiver.

3.7.2 Address

This is the address for the receiver. An address ensures that the packet is detected and received by the correct receiver, preventing accidental cross talk between multiple HS6209 systems. You can configure the address field width in the AW register to be 5 bytes or 4 bytes address.

3.7.3 Guard

Figure 3.3 shows the format of the 2bytesguard packet has better synchronous characteristics.

3.7.4 Packet Control Field (PCF)

Figure 3.4 shows the format of the 9 bit packet control field, MSB to the left.

Payload length 6bit	PID 2bit	NO_ACK 1bit

Figure 3.4 Packet control field (PCF)

The packet control field contains a 6 bit payload length field, a 2 bit PID (Packet Identity) field and a 1 bit NO_ACK flag.

Payload length

This 6 bit field specifies the length of the payload in bytes. The length of the payload can be from 0 to 32 bytes.
Coding: 000000 = 0 byte (only used in empty ACK packets.The 0 length packet also need to be read out use R_RX_PAYLOAD with no data following) 100000 = 32 byte, 100001 = Don't care

This field is only used if the Dynamic Payload Length function is enabled.

PID (Packet identification)

The 2 bit PID field is used to detect if the received packet is new or retransmitted. PID prevents the PRX operation from presenting the same payload more than once to the MCU. The PID field is incremented at the TX side for each new packet received through the SPI. The PID and CRC field are used by the PRX operation to determine if a packet is retransmitted or new. When several data packets are lost on the link, the PID fields may become equal to the last received PID. If a packet has the same PID as the previous packet, the RF transceiver compares the CRC sums from both packets. If the CRC sums are also equal, the last received packet is considered a copy of the previously received packet and discarded.

No Acknowledgment flag (NO_ACK)

The Selective Auto Acknowledgement feature controls the NO_ACK flag.

This flag is only used when the auto acknowledgement feature is used. Setting the flag high, tells the receiver that the packet is not to be auto acknowledged.

On the PTX you can set the NO_ACK flag bit in the Packet Control Field with this command:
W_TX_PAYLOAD_NOACK

However, the function must first be enabled in the FEATURE register by setting the EN_DYN_ACK bit. When you use this option, the PTX goes directly to standby-I mode after transmitting the packet. The PRX does not transmit an ACK packet when it receives the packet.

3.7.5 Payload

The payload is the user defined content of the packet. It can be 0 to 32 bytes wide and is transmitted on-air when it is uploaded to the device.

Protocol engine provides two alternatives for handling payload lengths; static and dynamic.

The default is static payload length. With static payload length all packets between a transmitter and a receiver have the same length. Static payload length is set by the RX_PW_Px registers on the receiver side. The payload length on the transmitter side is set by the number of bytes clocked into the TX_FIFO and must equal the value in the RX_PW_Px register on the receiver side.

Dynamic Payload Length (DPL) is an alternative to static payload length. DPL enables the transmitter to send packets with variable payload length to the receiver. This means that for a system with different payload lengths it is not necessary to scale the packet length to the longest payload.

With the DPL feature the HS6209 can decode the payload length of the received packet automatically instead of using theRX_PW_Pxregisters. The MCU can read the length of the received payload by using the R_RX_PL_WID command.

> ***Note:*** Always check if the packet width reported is 32 bytes or shorter when using the R_RX_PL_WID command. If its width is longer than 32 bytes then the packet contains errors and must be discarded. Discard the packet by using the Flush_RXcommand.

In order to enable DPL the EN_DPL bit in the FEATURE register must be enabled. In RX mode the DYNPD register must be set. A PTX that transmits to a PRX with DPL enabled must have the DPL_P0 bit in DYNPD set.

3.7.6 CRC (Cyclic Redundancy Check)

The CRC is the error detection mechanism in the packet. It may either be 1 or 2 bytes and is calculated over the address, Packet Control Field and Payload.

The polynomial for 1 byte CRC is $X^8 + X^2 + X + 1$. Initial value 0Xff.

The polynomial for 2 byte CRC is $X^{16} + X^{12} + X^5 + 1$. Initial value 0Xffff.

The number of bytes in the CRC is set by the CRCO bit in the CONFIG register. No packet is accepted by protocol engine if the CRC fails.

3.7.7 Automatic packet assembly

The automatic packet assembly assembles the preamble, address, packet control field, payload and CRC to make a complete packet before it is transmitted.

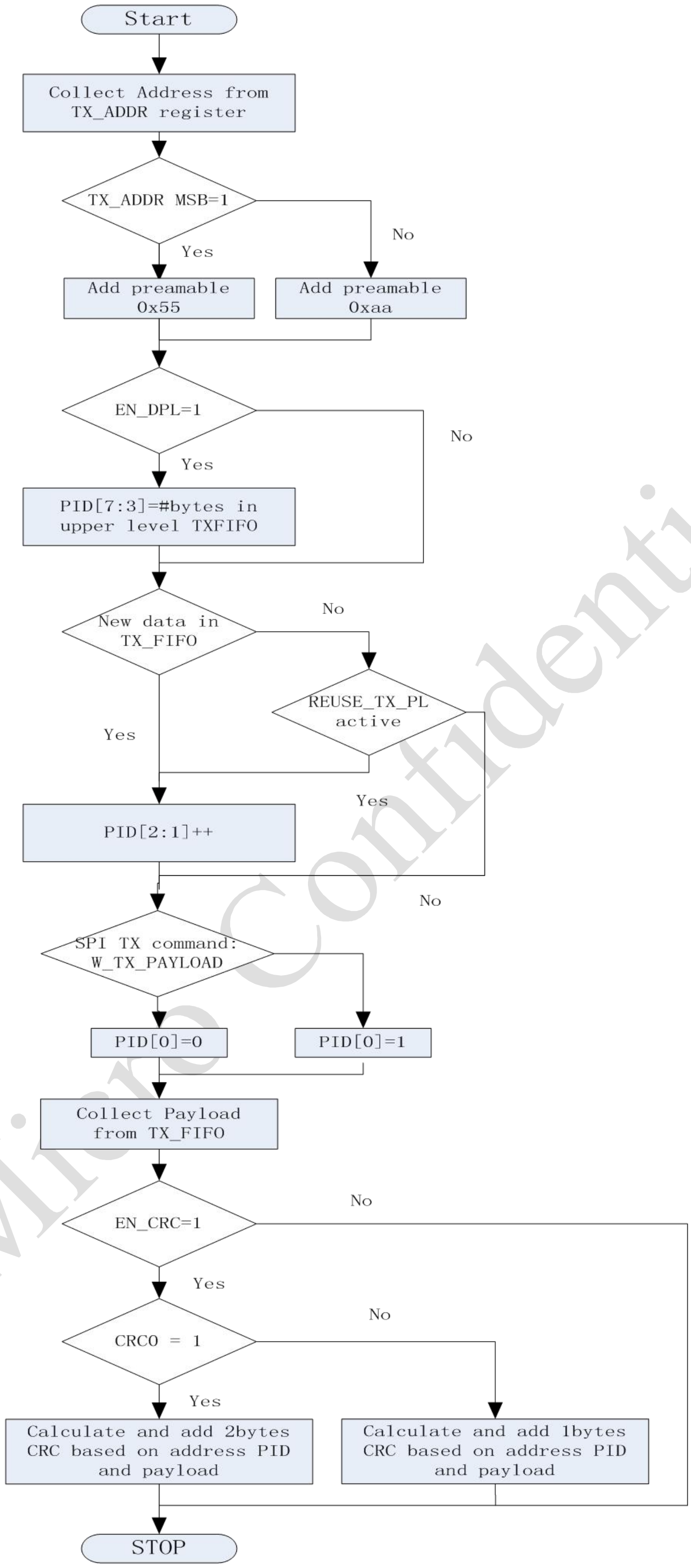


Figure 3.5 Automatic packetassembly

3.7.8 Automatic packet disassembly

After the packet is validated, Protocol engine disassembles the packet and loads the payload into the RX FIFO, and asserts the RX_DR IRQ.

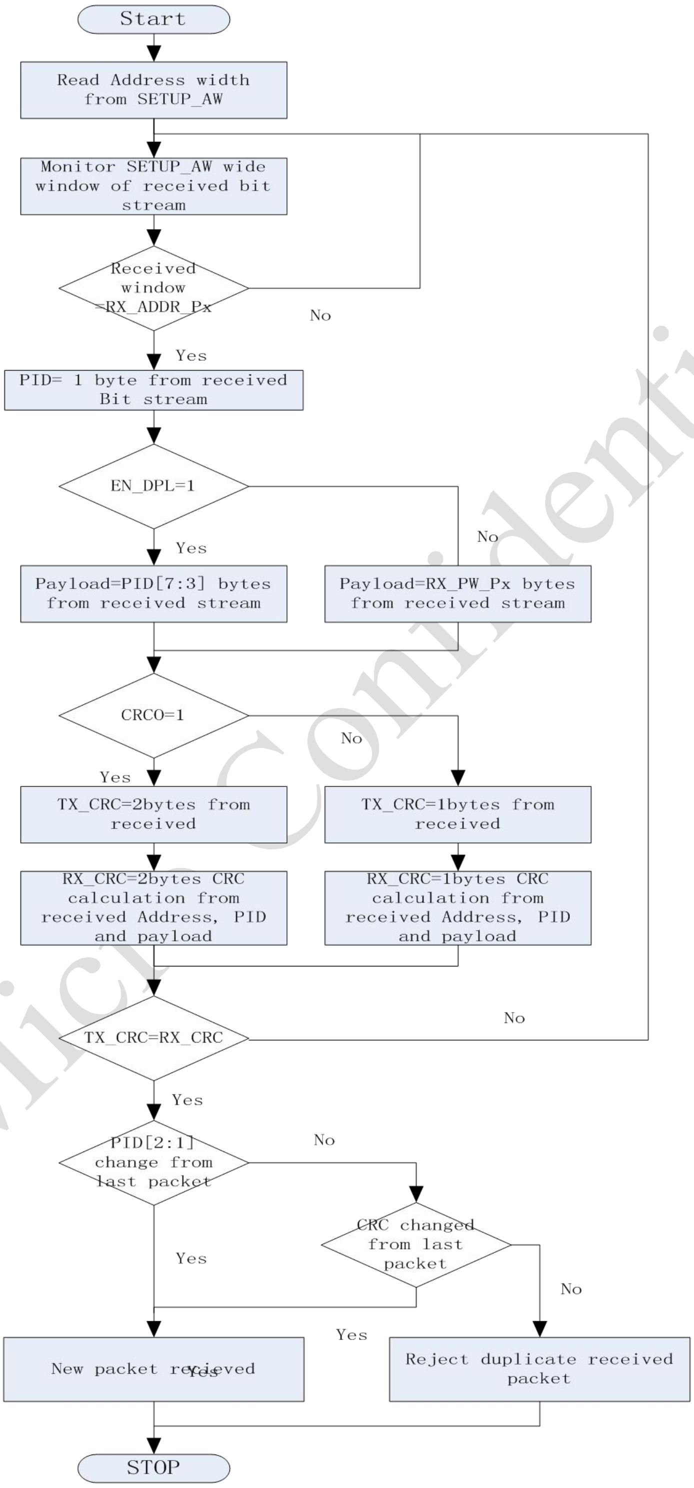


Figure 3.6Automatic packet disassembly

3.8 Automatic packet transaction handling

Protocol engine features two functions for automatic packet transaction handling; auto acknowledgement and auto re-transmit.

3.8.1 Auto Acknowledgement

Auto Acknowledgment is a function that automatically transmits an ACK packet to the PTX after it has received and validated a packet. The Auto Acknowledgement function reduces the load of the system MCU and reduces average current consumption. The Auto Acknowledgement feature is enabled by setting the EN_AA register.

Note: If the received packet has the NO_ACK flag set, auto acknowledgement is not executed.

An ACK packet can contain an optional payload from PRX to PTX. In order to use this feature, the Dynamic Payload Length (DPL) feature must be enabled. The MCU on the PRX side has to upload the payload by clocking it into the TX FIFO by using the W_ACK_PAYLOAD command. The payload is pending in the TX FIFO (PRX) until a new packet is received from the PTX. The RF transceiver can have three ACK packet payloads pending in the TX FIFO (PRX) at the same time.

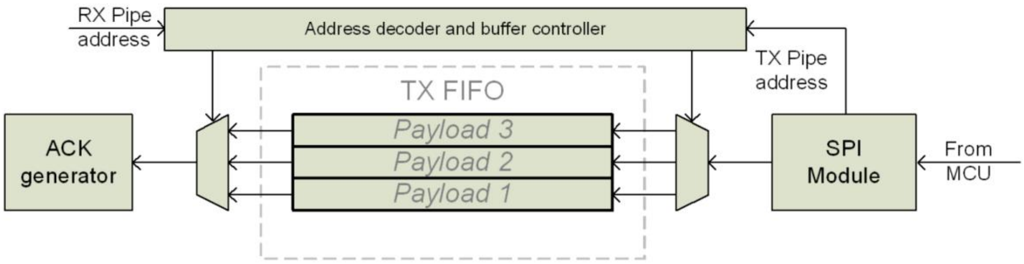


Fig. 3.7TX FIFO (PRX) with pending payloads

Figure 3.7shows how the TX FIFO (PRX) is operated when handling pending ACK packet payloads. From the MCU the payload is clocked in with the W_ACK_PAYLOAD command. The address decoder and buffer controller ensure that the payload is stored in a vacant slot in the TX FIFO (PRX). When a packet is received, the address decoder and buffer controller are notified with the PTX address. This ensures that the right payload is presented to the ACK generator.

If the TX FIFO (PRX) contains more than one payload to a PTX, payloads are handled using the first in–first out principle. The TX FIFO (PRX) is blocked if all pending payloads are addressed to a PTX where the link is lost. In this case, the MCU can flush the TX FIFO (PRX) by using the FLUSH_TX command.

In order to enable Auto Acknowledgement with payload the EN_ACK_PAY bit in the FEATURE register must be set.

3.8.2 Auto Retransmission (ART)

The auto retransmission is a function that retransmits a packet if an ACK packet is not received. It is used in an Auto Acknowledgement system on the PTX. When a packet is not acknowledged, you can set the number of times it is allowed to retransmit by setting the ARC bits in the SETUP_RETR register. PTX enters RX mode and waits a time period for an ACK packet each time a packet is transmitted. The amount of time the PTX is in RX mode is based on the following conditions:

- Auto Retransmit Delay (ARD) has elapsed.

- No address match within 256µs.
- After received packet (CRC correct or not) if address match within 256µs.

The RF transceiver asserts the TX_DS IRQ when the ACK packet is received.

The RF transceiver enters standby-I mode if there is no more un-transmitted data in the TX FIFO and the CEpin is low. If the ACK packet is not received, the RF transceiver goes back to TX mode after a delay defined by ARD and retransmits the data. This continues until acknowledgment is received, or the maximum number of retransmits is reached.

Two packet loss counters are incremented each time a packet is lost, ARC_CNT and PLOS_CNT in the OBSERVE_TX register. The ARC_CNT counts the number of retransmissions for the current transaction. You reset ARC_CNT by initiating a new transaction. The PLOS_CNT counts the total number of retransmissions since the last channel change. You reset PLOS_CNT by writing to the RF_CH register. It is possible to use the information in the OBSERVE_TX register to make an overall assessment of the channel quality.

The ARD defines the time from the end of a transmitted packet to when a retransmit starts on the PTX. ARD is set in SETUP_RETR register in steps of 256µs. A retransmit is made if no ACK packet is received by the PTX.

There is a restriction on the length of ARD when using ACK packets with payload. The ARD time must never be shorter than the sum of the startup time and the time on-air for the ACK packet.

- For 2 Mbps data rate and 5-byte address; 15 byte is maximum ACK packet payload length for ARD=256µs (reset value).
- For 1 Mbps data rate and 5-byte address; 5 byte is maximum ACK packet payload length for ARD=256µs (reset value).

ARD=512µs is long enough for any ACK payload length in 1 or 2 Mbps mode.

- For 500kbps data rate and 5-byte address the following values apply:

ARD	ACK packet size (in byte)
1536us	All ACK payload sizes
1280us	<=24
1024us	<=16
768us	<=8
512us	Empty ACK with no payload

Table 3.4 Maximum ACK payload length for different retransmit delays

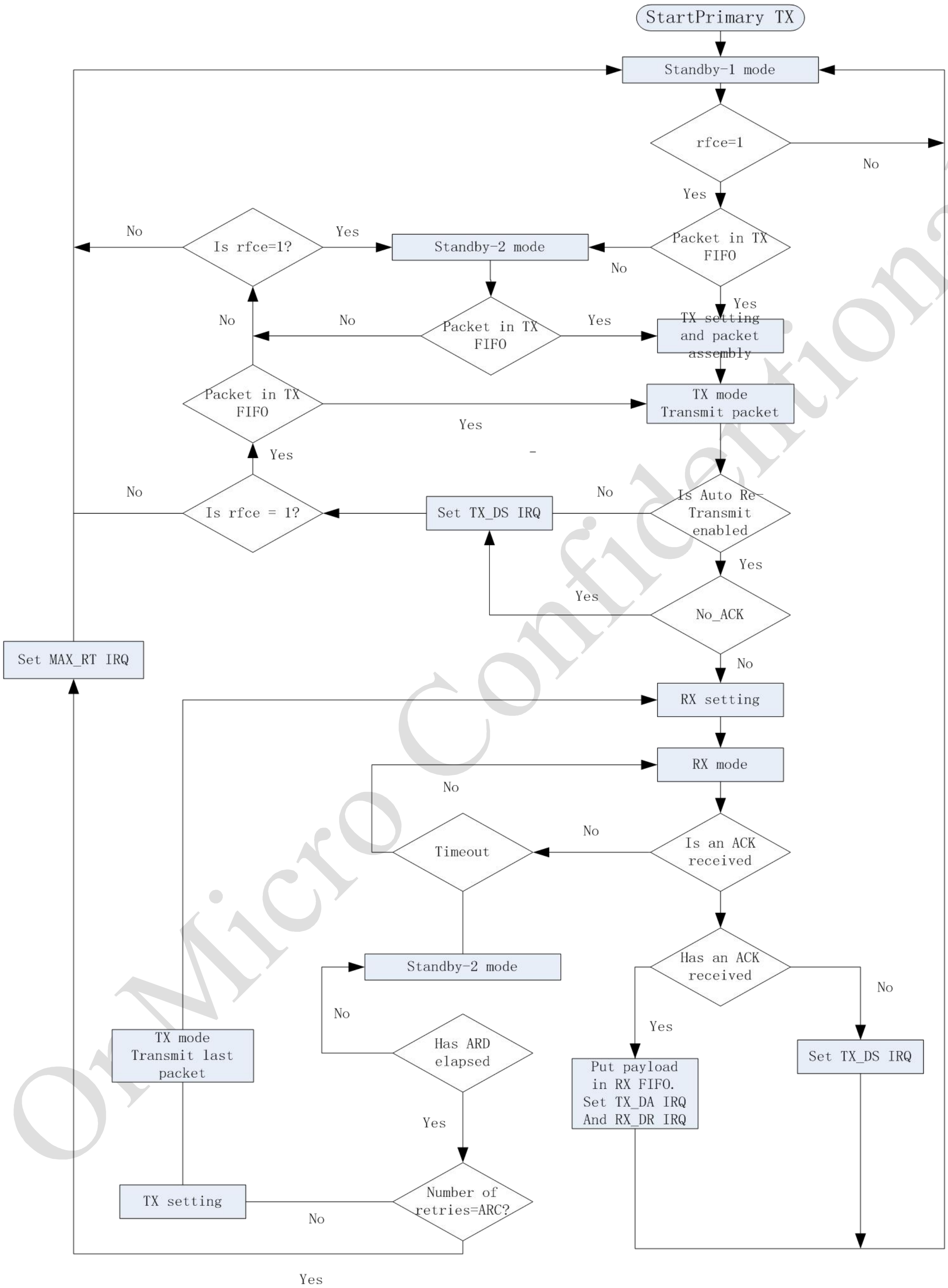
As an alternative to Auto Retransmit it is possible to manually set the RF transceiver to retransmit a packet a number of times. This is done by the REUSE_TX_PL command. The MCU must initiate each transmission of the packet with a pulse on the CE pin when this command is used.

3.9 Protocol engine flowcharts

This section contains flowcharts outlining PTX and PRX operation in Protocol engine.

3.9.1 PTXoperation

The flowchart in Figure 3.8 outlines how a RF transceiver configured as a PTX behaves after entering standby-I mode.



Note:Protocol engine operation is outlined with a dashed square.

Figure 3.8 PTX operations in Protocol engine

Activate PTX mode by setting the rfcein the RFCON register high. If there is a packet present in the TX FIFO the RF transceiver enters TX mode and transmits the packet. If Auto Retransmit is enabled, the state machine checks if the NO_ACK flag is set. If it is not set, the RF transceiver enters RX mode to receive an ACK packet. If the received ACK packet is empty, only the TX_DS IRQ is asserted. If the ACK packet contains a payload, both TX_DS IRQ and RX_DR IRQ are asserted simultaneously before the RF transceiver returns to standby-I mode.

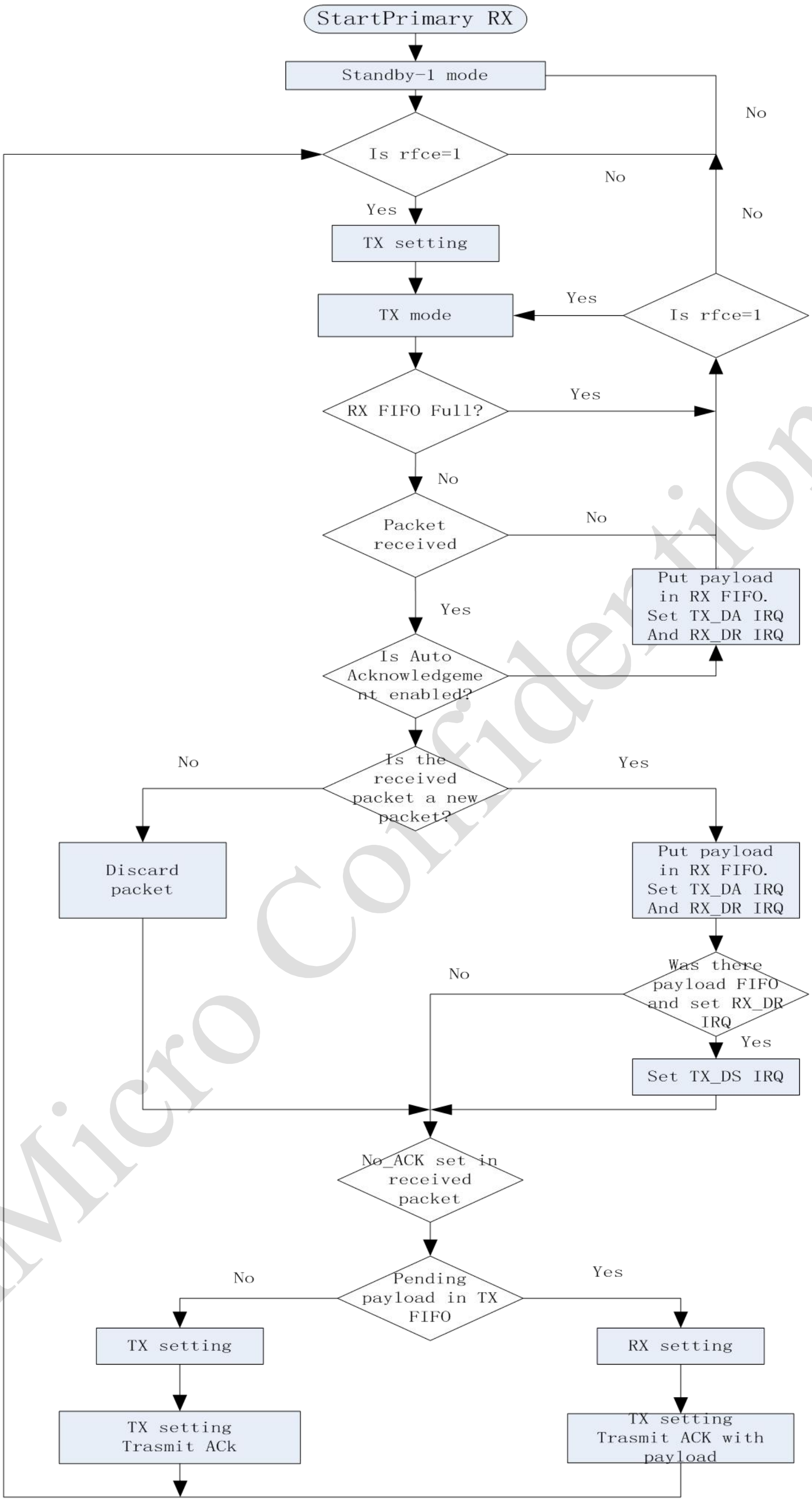
If the ACK packet is not received before timeout occurs, the RF transceiver returns to standby-II mode. It stays in standby-II mode until the ARD has elapsed. If the number of retransmits has not reached the ARC, the RF transceiver enters TX mode and transmits the last packet once more.

While executing the Auto Retransmit feature, the number of retransmits can reach the maximum number defined in ARC. If this happens, the RF transceiver asserts the MAX_RT IRQ and returns to standby-I mode.

If the rfce bit in the RFCON register is high and the TX FIFO is empty, the RF transceiver enters Standby-II mode.

3.9.2 PRX operation

The flowchart in Figure 3.9 outlines how a RF transceiver configured as a PRX behaves after entering standby-I mode.



Note:Protocol engine operation is outlined with a dashed square.

Figure3.9 PRX operations in Protocol engine

Activate PRX mode by setting the rfce bit in the RFCON register high. The RF transceiver enters RX mode and starts searching for packets. If a packet is received and Auto Acknowledgement is enabled, the RF transceiver decides if the packet is new or a copy of a previously received packet. If the packet is new payload is made available in the RX FIFO and the RX_DR IRQ is asserted. If the last received packet from the transmitter is acknowledged with an ACK packet with payload, the TX_DS IRQ indicates that the PTX received the ACK packet with payload. If the No_ACK flag is not set in the received packet, the PRX enters TX mode. If there is a pending payload in the TX FIFO it is attached to the ACK packet. After the ACK packet is transmitted, the RF transceiver returns to RX mode.

A copy of a previously received packet might be received if the ACK packet is lost. In this case, the PRX discards the received packet and transmits an ACK packet before it returns to RX mode.

3.10 MultiSlave

MultiSlave is a feature used in RX mode that contains a set of six parallel data pipes with unique addresses. A data pipe is a logical channel in the physical RF channel. Each data pipe has its own physical address (data pipe address) decoding in the RF transceiver.

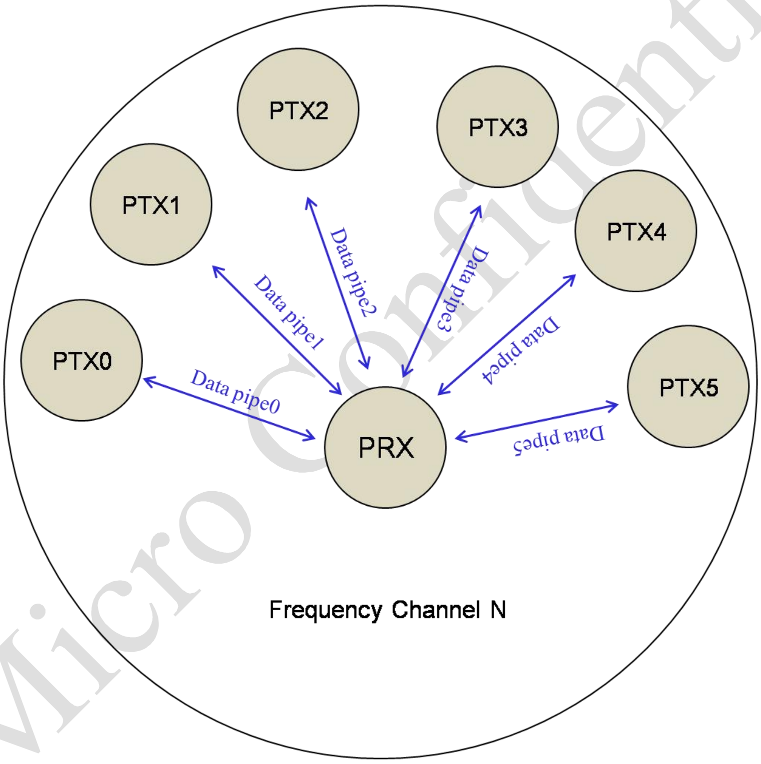


Fig.3.10 PRX using MultiSlave

The RF transceiver configured as PRX (primary receiver) can receive data addressed to six different data pipes in one frequency channel as shown in Figure 3.10. Each data pipe has its own unique address and can be configured for individual behavior.

Up to six RF transceivers configured as PTX can communicate with one RF transceiver configured as PRX. All data pipe addresses are searched for simultaneously. Only one data pipe can receive a packet at a time. All data pipes can perform Protocol engine functionality.

The following settings are common to all data pipes:

- CRC enabled/disabled (CRC always enabled when Protocol engine is enabled)
- CRC encoding scheme
- RX address width

- Frequency channel
- Air data rate
- LNA gain

The data pipes are enabled with the bits in the EN_RXADDR register. By default only data pipe 0 and 1 are enabled. Each data pipe address is configured in the RX_ADDR_PX registers.

Note: Always ensure that none of the data pipes have the same address.

Each pipe can have up to a 5 byte configurable address. Data pipes 0-5 share the four most significant address bytes. The LSByte must be unique for all six pipes. Figure 3.11 is an example of how data pipes 0-5 are addressed.Only pipe0 can have up to a 5 byte configurable address, other's pipes have 1bytes configurable address.

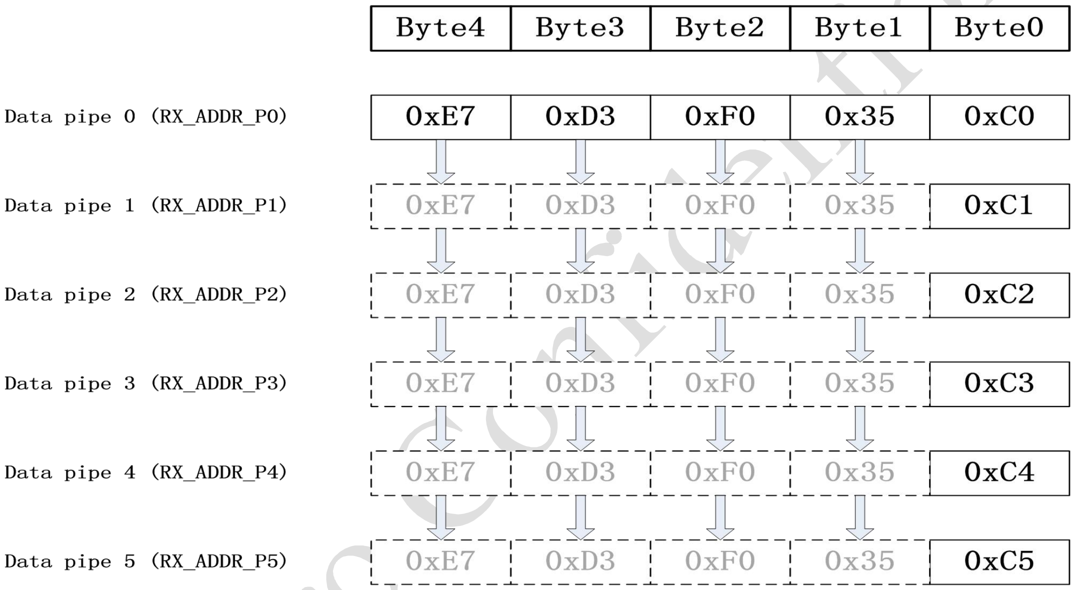


Figure 3.11 Addressing data pipes 0-5

The PRX, using MultiSlave and Protocol engine, receives packets from more than one PTX. To ensure that the ACK packet from the PRX is transmitted to the correct PTX, the PRX takes the data pipe address where it received the packet and uses it as the TX address when transmitting the ACK packet. Figure 3.12 is an example of an address configuration for the PRX and PTX. On the PRX the RX_ADDR_Px, defined as the pipe address, must be unique. On the PTX the TX_ADDR must be the same as the RX_ADDR_P0 and as the pipe address for the designated pipe.

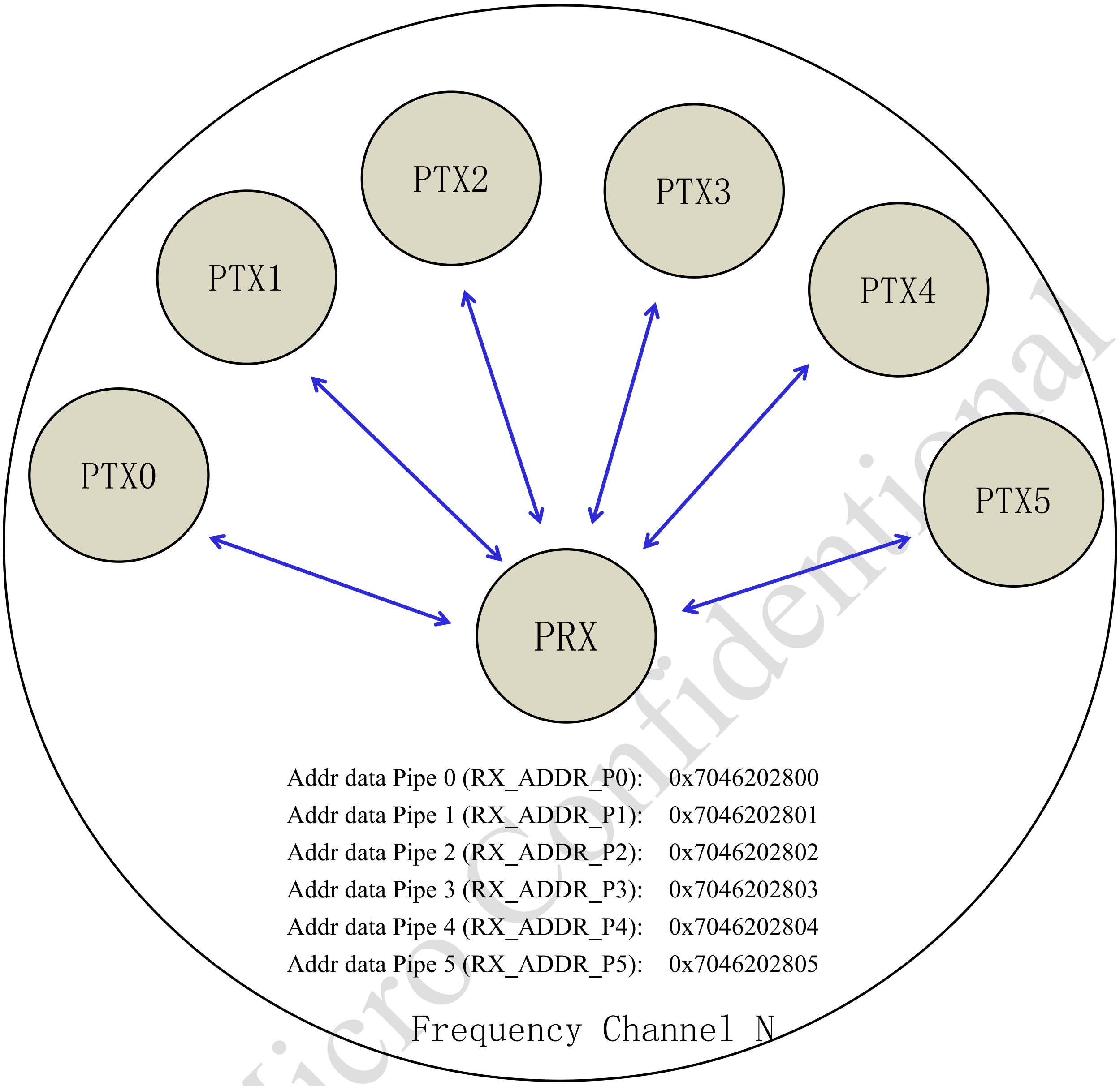
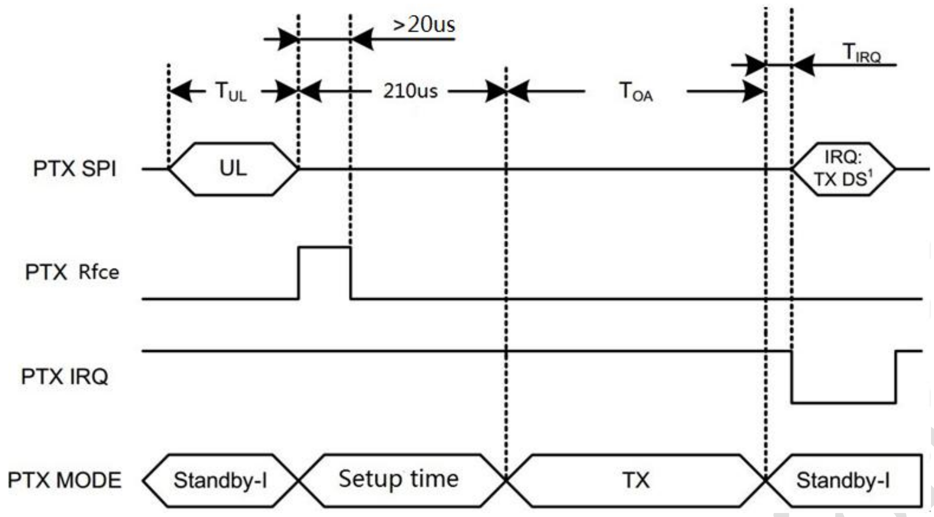


Figure 3.12 Example of data pipe addressing in MultiSlave

Only when a data pipe receives a complete packet can other data pipes begin to receive data. When multiple PTXs are transmitting to a PRX, the ARD can be used to skew the auto retransmission so that they only block each other once.

3.11 Protocol engine timing

This section describes the timing sequence of Protocol engine and how all modes are initiated and operated. The Protocol engine timing is controlled through the Data and Control interface. The RF transceiver can be set to static modes or autonomous modes where the internal state machine controls the events. Each autonomous mode/sequence ends with a RFIRQ interrupt. All the interrupts are indicated as IRQ events in the timing diagrams.



1 IRQ if No Ack is on.

T_{IRQ}=3us @ 1Mbps, @2Mbps

Figure 3.13Transmitting one packet with NO_ACK on

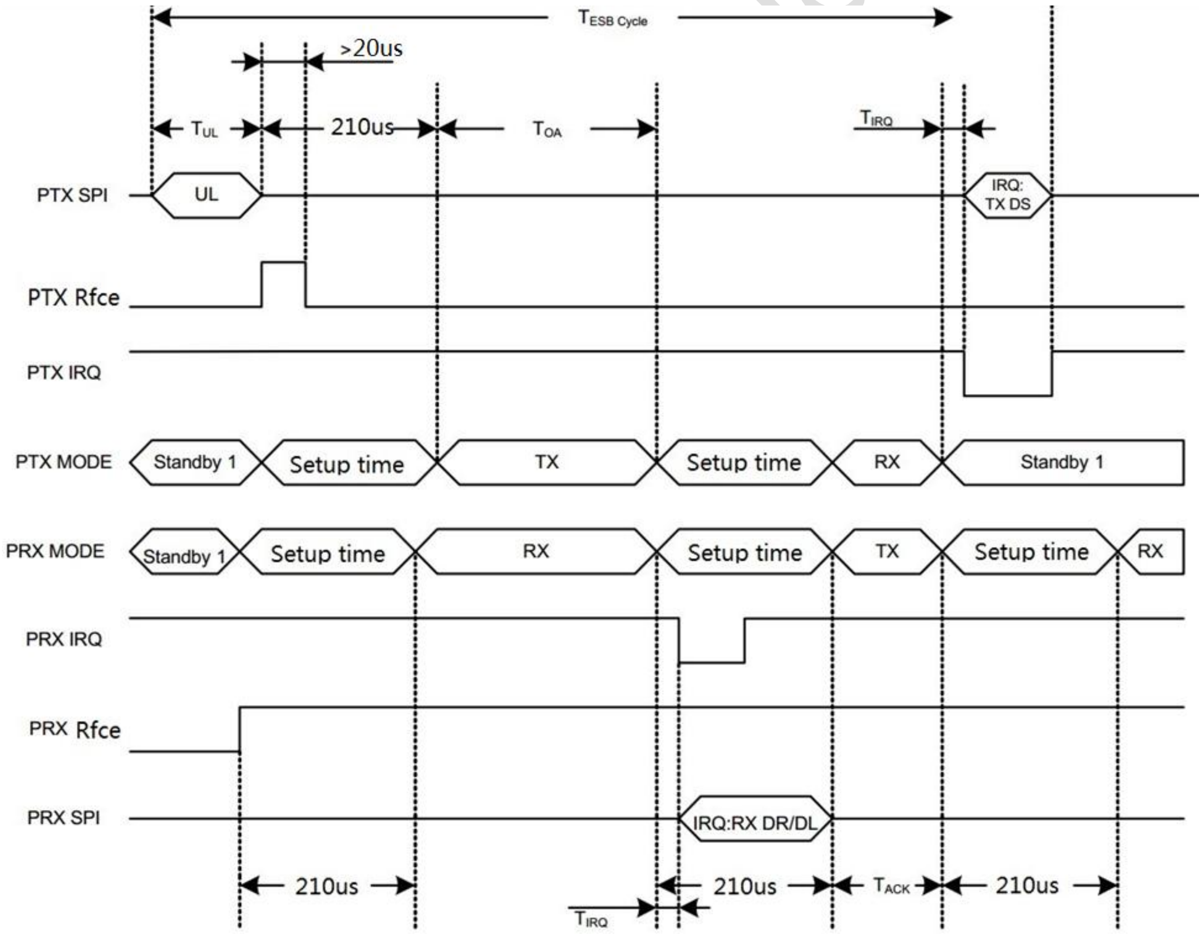


Figure 3.14 Timing of Protocol engine for one packet upload(2Mbps)

In Figure 3.14, the transmission and acknowledgement of a packet is shown. The PRX operation activates RX mode (rfce=1), and the PTX operation is activated in TX mode (rfce=1 for minimum 20µs). After 210µs the transmission starts and finishes after the elapse of

T_{OA}.

When the transmission ends the PTX operation automatically switches to RX mode to wait for the ACK packet from the PRX operation. When the PRX operation receives the packet it sets the interrupt for the host MCU and switches to TX mode to send an ACK. After the PTX operation receives the ACK packet it sets the interrupt to the MCU and clears the packet from the TX FIFO.

In Figure 3.15, the PTX timing of a packet transmission is shown when the first ACK packet is lost.

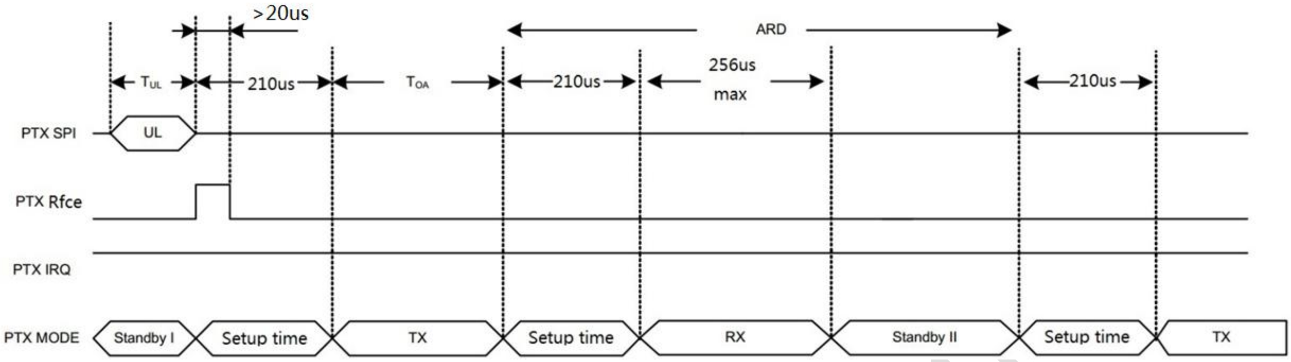


Figure 3.15 Timing of Protocol engine when the first ACK packet is lost (2 Mbps)

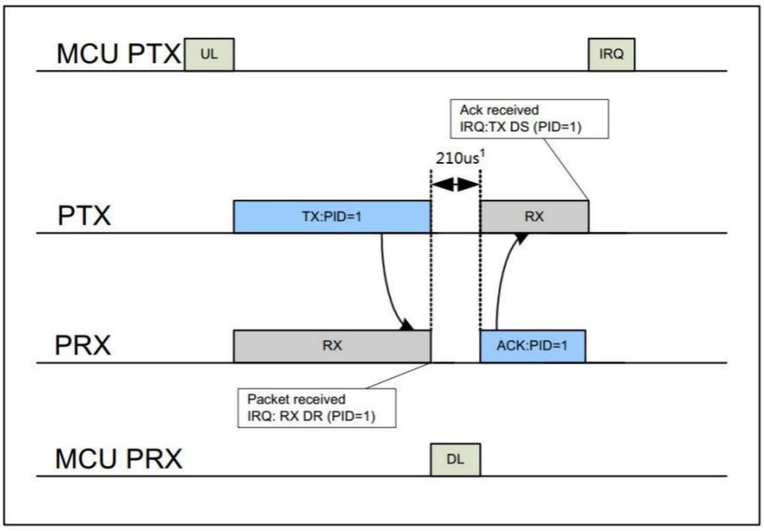
3.12 Protocol engine transaction diagram

This section describes several scenarios for the Protocol engine automatic transaction handling. The call outs in this section's figures indicate the IRQs and other events. For MCU activity the event may be placed at a different timeframe.

Note: The figures in this section indicate the earliest possible download (DL) of the packet to the MCU and the latest possible upload (UL) of payload to the transmitter.

3.12.1 Single transaction with ACK packet and interrupts

In Fig.3.16, the basic auto acknowledgement is shown. After the packet is transmitted by the PTX and received by the PRX the ACK packet is transmitted from the PRX to the PTX. The RX_DR IRQ is asserted after the packet is received by the PRX, whereas the TX_DS IRQ is asserted when the packet is acknowledged and the ACK packet is received by the PTX.

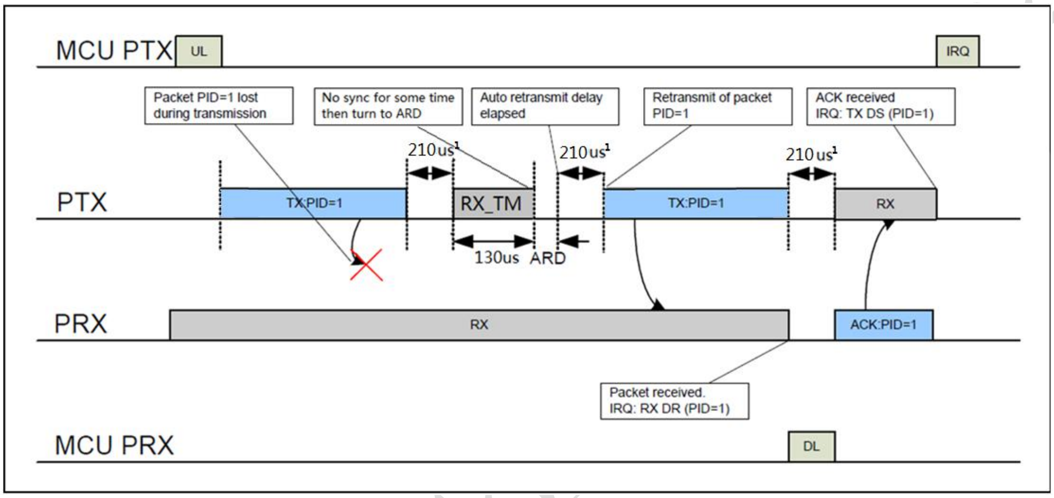


1 Radio Turn Around delay

Fig.3.16TX/RX cycles with ACK and the according interrupts

3.12.2 Single transaction with a lost packet

Figure 3.17 is a scenario where a retransmission is needed due to loss of the first packet transmits. After the packet is transmitted, the PTX enters RX mode to receive the ACK packet. After the first transmission, the PTX waits a specified time (including setup time, RX_TM and ARD) for the ACK packet, if it is not in the specific time slot the PTX retransmits the packet as shown in Figure 3.17. PTX will turn to RX mode after 210us setup time when packet is transmitted, after 130us RX timeout (RX_TM is RX timeout for PTX, it can be set shorter), then PTX turn to ARD (can be set to 0us, 256us, 512us to 3840us).



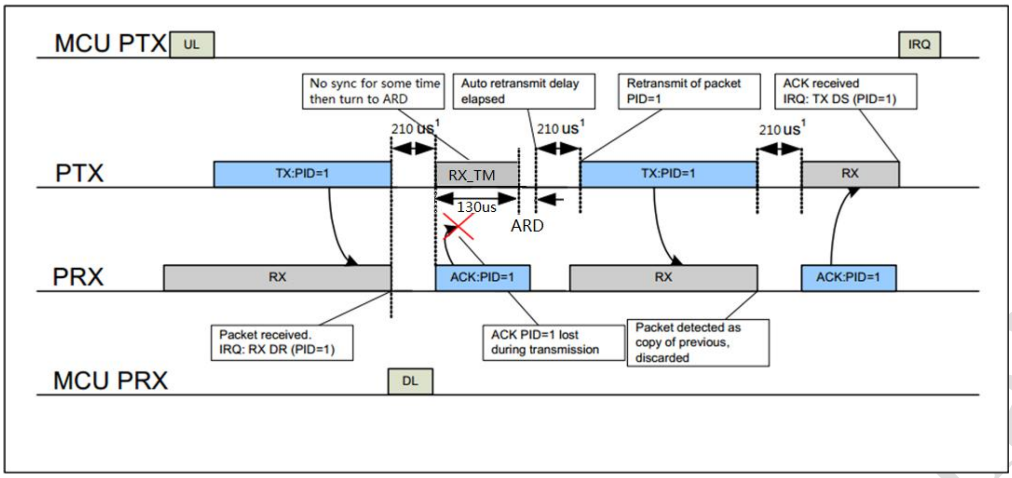
1 Radio Turn Around delay

Figure 3.17TX/RX cycles with ACK and the according interrupts when the first packet transmit fails

When an address is detected the PTX stays in RX mode until the packet is received. When the retransmitted packet is received by the PRX (see Figure 3.17). The RX_DR IRQ is asserted and an ACK is transmitted back to the PTX. When the ACK is received by the PTX, the TX_DS IRQ is asserted.

3.12.3 Single transaction with a lost ACK packet

Figure 3.18 is a scenario where a retransmission is needed after a loss of the ACK packet. The corresponding interrupts are also indicated.

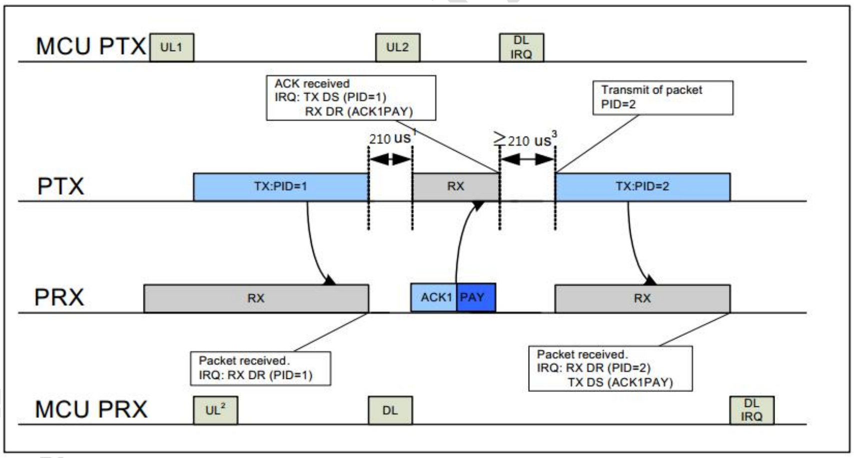


1 Radio Turn Around delay

Fig.3.18TX/RX cycles with ACK and the according interrupts when the ACK packet fails

3.12.4 Single transaction with ACK payload packet

Figure 3.19 is a scenario of the basic auto acknowledgement with payload. After the packet is transmitted by the PTX and received by the PRX the ACK packet with payload is transmitted from the PRX to the PTX. The RX_DR IRQ is asserted after the packet is received by the PRX, whereas on the PTX side the TX_DS IRQ is asserted when the ACK packet is received by the PTX. On the PRX side, the TX_DS IRQ for the ACK packet payload is asserted after a new packet from PTX is received.



1 Radio Turn Around delay

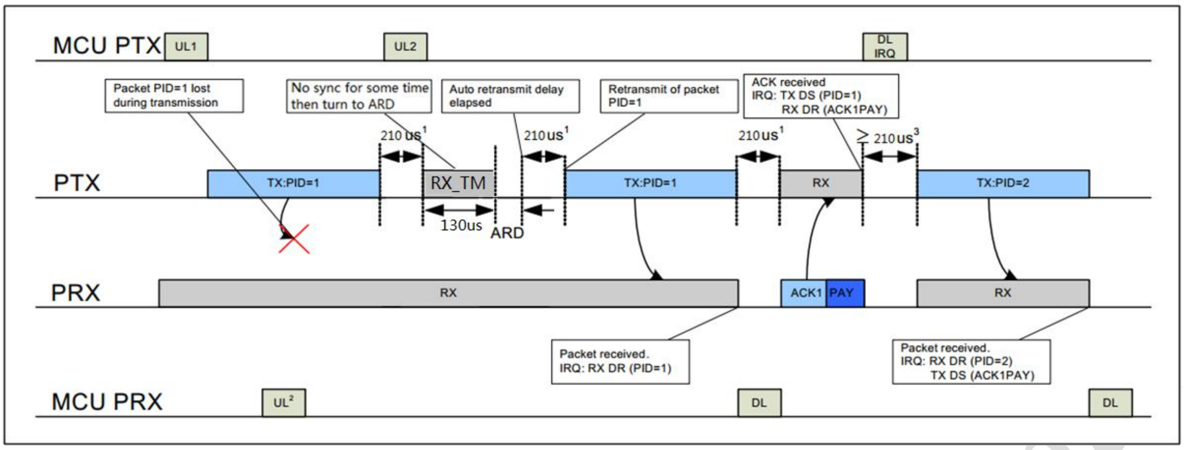
2 Uploading payload for Ack Packet

3 Delay defined by MCU on PTX side, >=210us

Fig.3.19TX/RX cycles with ACK Payload and the according interrupts

3.12.5 Single transaction with ACK payload packet and lost packet

Figure 3.20 is a scenario where the first packet is lost and a retransmission is needed before the RX_DR IRQ on the PRX side is asserted. For the PTX both the TX_DS and RX_DR IRQ are asserted after the ACK packet is received. After the second packet (PID=2) is received on the PRX side both the RX_DR (PID=2) and TX_DS (ACK packet payload) IRQ are asserted.



1 Radio Turn Around delay

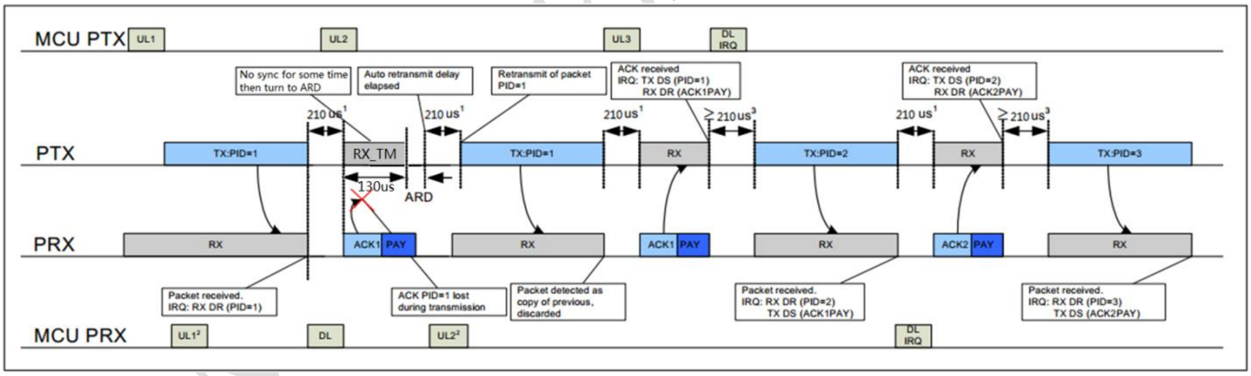
2 Uploading payload for Ack Packet

3 Delay defined by MCU on PTX side, >=210us

Fig.3.20TX/RX cycles and the according interrupt when the packet transmission fails

3.12.6 Two transactions with ACK payload packet and the first ACK packet lost

In Fig 3.21 the ACK packet is lost and a retransmission is needed before the TX_DS IRQ is asserted, but the RX_DR IRQ is asserted immediately. The retransmission of the packet (PID=1) results in a discarded packet. For the PTX both the TX_DS and RX_DR IRQ are asserted after the second transmission of ACK, which is received. After the second packet (PID=2) is received on the PRX both the RX_DR (PID=2) and TX_DS (ACK1PAY) IRQ is asserted. The callouts explains the different events and interrupts.



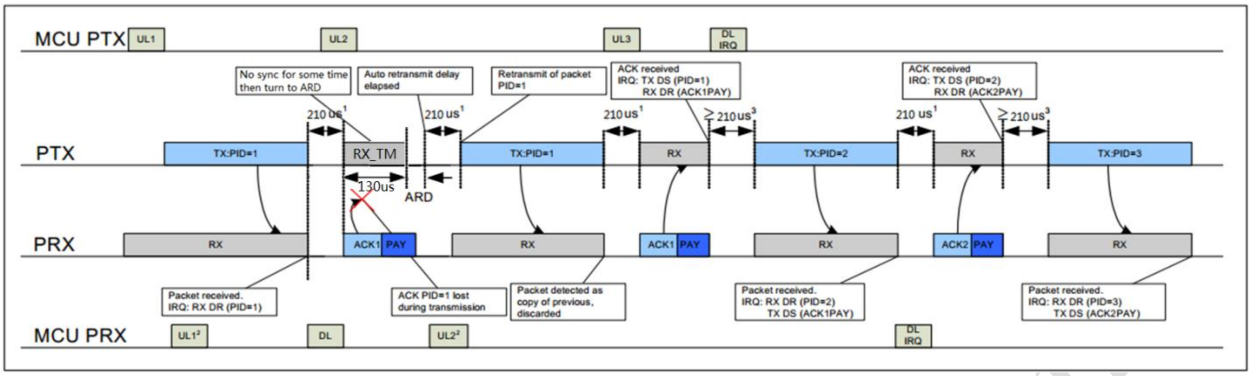
1 Radio Turn Around delay

2 Uploading payload for Ack Packet

3 Delay defined by MCU on PTX side, >=210us

Fig.3.21TX/RX cycles with ACK Payload and the according interrupts when the ACK packet fails

3.12.7 Two transactions where max retransmissions is reached



1 Radio Turn Around delay

2 Uploading payload for Ack Packet

3 Delay defined by MCU on PTX side, >=210us

Fig 3.22TX/RX cycles with ACK Payload and the according interrupts when the transmission fails. ARCis set to 2.

MAX_RT IRQ is asserted if the auto-retransmit counter (ARC_CNT) exceeds the programmed maximum limit (ARC). In Fig 3.22, the packet transmission ends with a MAX_RT IRQ. The payload in TX FIFO is NOT removed and the MCU decides the next step in the protocol. A toggle of the rfce bit in the RFCON register starts a new transmitting sequence of the same packet. The payload can be removed from the TX FIFO using the FLUSH_TX command.

3.13 Data and control interface

The data and control interface gives you access to all the features in the RF transceiver. Compared to the standalone component SFR registers are used instead of port pins. Otherwise the interface is identical to the standalone HS6209 chip.

3.13.1 SFR registers

Address (Hex)	Name/mnemonic	Bit	Reset Value	Type	Description
0xE5	SPIRCON	3:0	0x0F	R/W	SPI MASTER configuration register1
	maskIrqRxFifoFull	3	1	R/W	1: Disable interrupt when RX FIFO is full. 0: Enable interrupt when RX FIFO is full.
	maskIrqRxDataReady	2	1	R/W	1: Disable interrupt when data is available in RX FIFO. 0: Enable interrupt when data is available in RX FIFO.
	maskIrqTxFifoEmpty	1	1	R/W	1: Disable interrupt when TX FIFO is empty. 0: Enable interrupt when TX FIFO is empty.
	maskIrqTxFifoReady	0	1	R/W	1: Disable interrupt when a location is available in TX FIFO. 0: Enable interrupt when a location is available in TX FIFO.
0xE6	SPIRSTAT	3:0	0x03	R	SPI Mater status register
	Rxfifo_full	3	0	R	1: RX FIFO full.

					0: RX FIFO can accept more data from SPI.
	Rxdata_ready	2	0	R	1: Data available in RX FIFO. 0: No data in RX FIFO.
	Txfifo_empty	1	1	R	1: TX FIFO empty. 0: Data in TX FIFO.
	Txfifo_ready	0	1	R	1: Location available in TX FIFO. 0: TX FIFO full.
0xE7	SPIRDAT	7:0	0x00	R/W	SPI Master data register. Accesses TX (write) and RX (read) FIFO buffers, both two bytes deep.
0xE8	RFCON	1:0	0x2	R/W	
	Rfcsn	1	1	R/W	Enable RF command. 0: enabled
	rfce	0	0	R/W	Enable RF transceiver. 1: enabled

The RF transceiver SPI Master is configured through SPIRCON1. After transferred one byte can generate interrupt (MSDONE), unless they are masked by their respective bits in SPIRCON1. SPIRSTAT reveals which sources these are active.

SPIRDAT accesses both the TX (write) and the RX (read) FIFOs, which are two bytes deep. The FIFOs are dynamic and can be refilled according to the state of the status flags: "FIFO ready" means that the FIFO can accept data. Data readymeans that the FIFOcan provide data, minimum one byte.**RFCON** controls the RF transceiver SPI Slave chip select signal (rfcsn), the RF transceiver chip enable signal (rfce).

3.13.2 SPI Operation

This section describes the SPI commands and timing.

3.13.2.1 SPI commands

The SPI commands are shown in Table 3.5. Every new command must be started by writing 0 to rfcsn in the RFCON register.

The SPI command is transferred to RF transceiver by writing the command to the SPIRDAT register. After the first transfer the RF transceiver's STATUS register can be read from SPIRDAT when the transfer is completed.

The serial shifting SPI commands is in the following format:

<**Command word:**MSBit to LSBit (one byte)>

<**Data bytes:**MSBit in each byte first>

Command	Command word (binary)	#Data bytes	Operation
R_RESISTER	000A AAAA	1 to 5 LSByte first	Read command and status registers. AAAAA=5 bit register map address
W_RESISTER	001A AAAA	1 to 5 LSByte first	Read command and status registers. AAAAA=5 bit register map address Executable in power down or standby modes only.
R_TX_PAYLOAD	0110 0001	1 to 32 LSByte first	Read RX-payload: 1-32 bytes. A read operation always starts at byte 0. Payload is deleted from FIFO after it is read. Used in RX mode.

W_TX_PAYLOAD	1010 0000	1 to 32 LSByte first	Write TX-payload: 1 – 32 bytes. A write operation always starts at byte 0 used in TX payload.
FLUSH_TX	1110 0001	0	Flush TX FIFO, used in TX mode
FLUSH_RX	1110 0010	0	Flush TX FIFO, used in TX mode Should not be executed during transmission of acknowledge, that is, acknowledge package will not be completed
REUSE_TX_PL	1110 0011	0	Used for a PTX operation Reuse last transmitted payload. TX payload reuse is active until W_TX_PAYLOAD or FLUSH TX is executed. TX payload reuse must not be activated or deactivated during package transmission.
R_RX_PL_WID	0110 0000	1	Read RX payload width for the top R_RX_PAYLOAD in the RX FIFO. Note: Flush RX FIFO if the read value is larger than 32 bytes.
W_ACK_PAYLOAD	1010 1PPP	1 to 32 LSByte first	Used in RX mode. Write Payload to be transmitted together with ACK packet on PIPE PPP. (PPP valid in the range from 000 to 101). Maximum three ACK packet payloads can be pending. Payloads with same PPP are handled using first in - first out principle. Write payload: 1– 32 bytes. A write operation always starts at byte 0.
W_TX_PAYLOAD_NO_ACK	1011 0000	1 to 32 LSByte first	Used in TX mode. Disables AUTOACK on this specific packet specific packet.
NOP	1111 1111	0	No Operation. Might be used to read the STATUS register

Tab.3.5Command set for the RF transceiver SPI

The W_REGISTER and R_REGISTER commands operate on single or multi-byte registers. When accessing multi-byte registers readorwrites to the MSBit of LSByte first. You can terminate the writing before all bytes in a multi-byte register are written, leaving the unwritten MSByte(s) unchanged. For example, the LSByte of RX_ADDR_P0 can be modified by writing only one byte to the RX_ADDR_P0 register. The content of the status register is always read to MISO after a high to low transition on CSN.

Note: The 3 bit pipe information in the STATUS register is updated during the RFIRQ high to low transition. The pipe information is unreliable if the STATUS register is read during an RFIRQ high to low transition.

3.13.3 Data FIFO

The data FIFOs store transmitted payloads (TX FIFO) or received payloads that are ready to be clocked out (RX FIFO). The FIFOs are accessible in both PTX mode and PRX mode. The following FIFOs are present in the RF transceiver:

- TX three level, 32 byte FIFO
- RX three level, 32 byte FIFO

Both FIFOs have a controller and are accessible through the SPI by using dedicated SPI commands. A TX FIFO in PRX can store payloads for ACK packets to three different PTX devices. If the TX FIFO contains more than one payload to a pipe, payloads are handled using the first in - first out principle. The TX FIFO in a PRX is blocked if all pending payloads are addressed to pipes where the link to the PTX is lost. In this case, the MCU can flush the TX FIFO using the FLUSH_TXcommand.

The RX FIFO in PRX can contain payloads from up to three different PTX devices and a TX FIFO in PTX can have up to three payloads stored.

You can write to the TX FIFO using these three commands; W_TX_PAYLOAD and W_TX_PAYLOAD_NO_ACK in PTX mode and W_ACK_PAYLOAD in PRX mode. All three commands provide access to the TX_PLD register.

The RX FIFO can be read by the command R_RX_PAYLOAD in PTX and PRX mode. This command provides access to the RX_PLD register.

The payload in TX FIFO in a PTX is not removed if the MAX_RT IRQ is asserted.

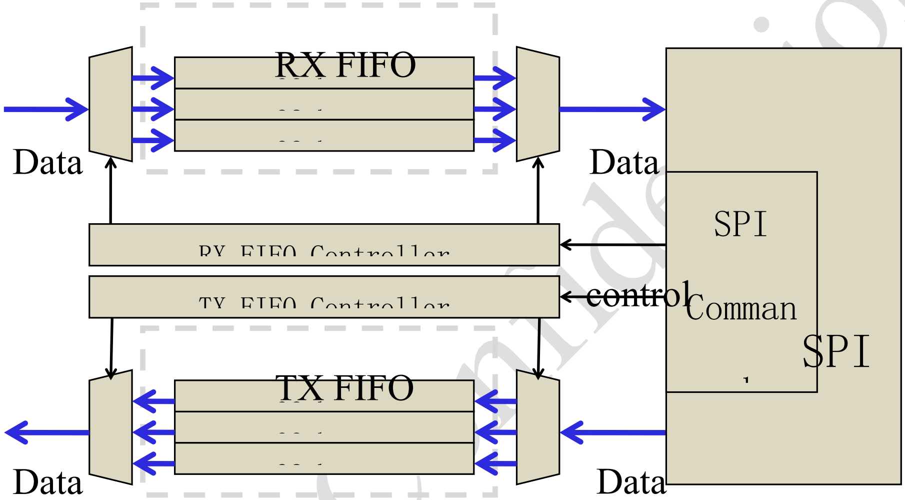


Figure 3.23 FIFO (RX and TX) block diagram

You can read if the TX and RX FIFO are full or empty in the FIFO_STATUS register.

3.13.4 Interrupt

The RF transceiver can send interrupts to the MCU. The interrupt (RFIRQ) is activated when TX_DS, RX_DR or MAX_RT are set high by the state machine in the STATUS register. RFIRQ is deactivated when the MCU writes '1' to the interrupt source bit in the STATUS register. The interrupt mask in the CONFIG register is used to select the IRQ sources that are allowed to activate RFIRQ. By setting one of the mask bits high, the corresponding interrupt source is disabled. By default all interrupt sources are enabled.

Note: The 3 bit pipe information in the STATUS register is updated during the RFIRQ high to low transition. The pipe information is unreliable if the STATUS register is read during a RFIRQ high to low transition.

3.14 Register map

You can configure and control the radio by accessing the register map through the SPI.

3.14.1 Register map table

All undefined bits in the table below are redundant. They are read out as '0'.

Note: Addresses 18 to 1B are reserved for test purpose, alteringthem makes the chip malfunction.

Register Definition

3.14.2 BANK0

3.14.2.1 CONFIG (RW)Address: 00h

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reserved	MASK_RX_DR	MASK_TX_DS	MASK_MAX_RT	EN_CRC	CRCO	PWR_UP	PRIM_RX
0	0	0	0	1	0	0	0
RW	RW	RW	W	RW	RW	RW	RW

Description of Word

Bit	Value	Symbol	Description	
7	0	Reserved	Only 0 allowed	
			0	Keep the current value
			1	Reset to default values
6	0	MASK_RX_DR	Mask interrupt caused by RX_DR	
			0	Reflect RX_DR as active low interrupt on the IRQ pin
			1	Interrupt not reflected on the IRQ pin
5	0	MASK_TX_DS	Mask interrupt caused by TX_DS	
			0	Reflect TX_DS as active low interrupt on the IRQ pin
			1	Interrupt not reflected on the IRQpin
4	0	MASK_MAX_RT	Mask interrupt caused by MAX_RT	
			0	Reflect MAX_RT as active low interrupt on the IRQ pin
			1	Interrupt not reflected on the IRQ pin
3	1	EN_CRC	Enable CRC. Forced high if one of the bits in the EN_AA is high	
			0	Disable CRC
			1	Enable CRC
2	0	CRCO	CRC encoding scheme	
			0	1 byte
			1	2 byte
1	0	PWR_UP	Power up control	
			0	POWER DOWN
			1	POWER UP
0	0	PRIM_RX	RX/TX control	
			0	PTX

			1	PRX

3.14.2.2 EN_AA (RW) Address: 01h

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reversed		ENAA_P5	ENAA_P4	ENAA_P3	ENAA_P2	ENAA_P1	ENAA_P0
0		1	1	1	1	1	1
RW		RW	W	RW	RW	RW	RW

Description of Word

Bit	Value	Symbol	Description	
7	0	Reserved	Only 0 allowed	
			0	Keep the current value
			1	Reset to default values
6	0	Reserved	Only 0 allowed	
			0	Keep the current value
			1	Reset to default values
5	1	ENAA_P5	Enable auto acknowledgement data pipe 5	
			0	Disable
			1	Enable
4	1	ENAA_P4	Enable auto acknowledgement data pipe 4	
			0	Disable
			1	Enable
3	1	ENAA_P3	Enable auto acknowledgement data pipe 3	
			0	Disable
			1	Enable
2	1	ENAA_P2	Enable auto acknowledgement data pipe 2	
			0	Disable
			1	Enable
1	1	ENAA_P1	Enable auto acknowledgement data pipe 1	
			0	Disable
			1	Enable
0	1	ENAA_P0	Enable auto acknowledgement data pipe 0	
			0	Disable
			1	Enable

3.14.2.3 EN_RXADDR (RW)Address: 02h

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Reversed		ERX_P5	ERX_P4	ERX_P3	ERX_P2	ERX_P1	ERX_P0
0		0	0	0	0	1	1
RW		RW	W	RW	RW	RW	RW

Description of Word

Bit	Value	Symbol	Description	
7:6	0	Reserved	Only 0 allowed	
			0	Keep the current value
			1	Reset to default values
5	0	ERX_P5	Enable data pipe 5	
			0	Disable
			1	Enable
4	0	ERX_P4	Enable data pipe 4	
			0	Disable
			1	Enable
3	0	ERX_P3	Enable data pipe 3	
			0	Disable
			1	Enable
2	0	ERX_P2	Enable data pipe 2	
			0	Disable
			1	Enable
1	1	ERX_P1	Enable data pipe 1	
			0	Disable
			1	Enable
0	1	ERX_P0	Enable data pipe 0	
			0	Disable
			1	Enable

3.14.2.4 SETUP_AW (RW) Address: 03h

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reserved						SETUP_AW	
0						2’b11	
RW						RW	

Description of Word

Bit	Value	Symbol	Description	
7:2	0	Reserved	Only 0 allowed	
			0	Keep the current value

			1	Reset to default values
1:0	2'b11	SETUP_AW	Setup of Address Widths (common for all data pipes)	
			2'b11	5 bytes
			2'b10	4 bytes
			2'b01	Illegal
			2'b00	Illegal

3.14.2.5 SETUP_RETR (RW) Address: 04h

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ARD				ARC			
4'b0				4'b11			
RW				RW			

Description of Word

Bit	Value	Symbol	Description	
7:2	4'b0	ARD	Auto Retransmit Delay	
			4'hf	Wait 3840µS
			…	…
			4'h1	Wait 512µS
			4'h0	Wait 256µS
3:0	4'b11	ARC	Auto Retransmit Count	
			4'hf	Up to 15 Re-Transmit on fail of AA
			…	…
			4'h1	Up to 1 Re-Transmit on fail of AA
			4'h0	Re-Transmit disabled

3.14.2.6 RF_CH (RW) Address: 05h

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reg_Rf_ch							
8'b2							
RW							

Description of Word

Bit	Value	Symbol	Description
8:0	2	Reg_Rf_ch	Sets the frequency channel HS6200 operates on

3.14.2.7 RF_SETUP (RW) Address: 06h

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CONT_WAVE	PA_PWR[3]	RF_DR_LOW	Reserved	RF_DR_HIGH	Pa_power		
0	1	0	0	1	3’b010		
RW	RW	RW	RW	RW	RW		

Description of Word

<table>
<tr><th>Bit</th><th>Value</th><th>Symbol</th><th colspan="3">Description</th></tr>
<tr><td rowspan="3">7</td><td rowspan="3">0</td><td rowspan="3">CONT_WAVE</td><td colspan="3">Enables continuous carrier transmit when high</td></tr>
<tr><td>0</td><td colspan="2">Disable</td></tr>
<tr><td>1</td><td colspan="2">Enable</td></tr>
<tr><td>6</td><td>1</td><td>PA_PWR[3]</td><td colspan="3">PA power select bit 3</td></tr>
<tr><td>5</td><td>0</td><td>RF_DR_LOW</td><td colspan="3">See RF_DR_HIGH for encoding.</td></tr>
<tr><td>4</td><td>0</td><td>reserved</td><td colspan="3">Reserved</td></tr>
<tr><td rowspan="5">3</td><td rowspan="5">1</td><td rowspan="5">RF_DR_HIGH</td><td colspan="3">Select between the high speed data rates. This bit is donot care if RF_DR_LOW is set.Encoding:
[RF_DR_LOW, RF_DR_HIGH]:</td></tr>
<tr><td>11</td><td colspan="2">reserved</td></tr>
<tr><td>10</td><td colspan="2">reserved</td></tr>
<tr><td>01</td><td colspan="2">2Mbps</td></tr>
<tr><td>00</td><td colspan="2">1Mbps</td></tr>
<tr><td rowspan="10">2:0</td><td rowspan="10">3’b010</td><td rowspan="10">PA_PWR[2:0]</td><td colspan="3">PA power control, PA_PWR[3:0] with pa_voltage of RF_IVGEN in bank1</td></tr>
<tr><td>PA_PWR[3:0]</td><td>Pa_voltage(bank1 of RF_IVGEN)</td><td></td></tr>
<tr><td>1111</td><td>0</td><td>Output 6 dbm, 40mA</td></tr>
<tr><td>1000</td><td>0</td><td>Output 5 dbm</td></tr>
<tr><td>0111</td><td>1</td><td>Output 4 dbm, 25mA</td></tr>
<tr><td>0011</td><td>0</td><td>Output 0 dbm, 18.5mA</td></tr>
<tr><td>0001</td><td>0</td><td>Output -6 dbm</td></tr>
<tr><td>0001</td><td>1</td><td>Output -12 dbm</td></tr>
<tr><td>0000</td><td>0</td><td>Output -16 dbm</td></tr>
<tr><td>0000</td><td>1</td><td>Output -43 dbm</td></tr>
</table>

3.14.2.8 STATUS (RW) Address: 07h

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
BANK	RX_DR	TX_DS	MAX_RT	RX_P_NO			TX_FULL
0	0	0	0	3’b111			0

R	RW	RW	RW	R	R

Description of Word

Bit	Value	Symbol	Description	
7	0	BANK	Register BANK status	
			1	Register R/W is to register BANK1
			0	Register R/W is to register BANK0
6	0	RX_DR	Data Ready RX FIFO interrupt. Asserted when new data arrives RX FIFO Write 1 to clear bit.	
5	0	TX_DS	Data Sent TX FIFO interrupt. Asserted when packet transmitted on TX. If AUTO_ACK is activated, this bit is set high only when ACK is received. Write 1 to clear bit.	
4	0	MAX_RT	Maximum number of TX retransmits interrupt, Write 1 to clear bit. If MAX_RT is asserted it must be cleared to enable further communication.	
3:1	3'b111	RX_P_NO	Data pipe number for the payload available for reading from RX_FIFO	
			111	RX FIFO Empty
			110	Not Used
			000-101	Data Pipe Number
0	0	TX_FULL	TX FIFO full flag	
			0	Available locations in TX FIFO
			1	TX FIFO full

3.14.2.9 OBSERVE_TX (RW) Address: 08h

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PLOS_CNT				ARC_CNT			
4'h0				4'h0			
R				R			

Description of Word

Bit	Value	Symbol	Description
7:4	4'h0	PLOS_CNT	Count lost packets. The counter is overflow protected to 15, and discontinues at max until reset. The counter is reset by writing to RF_CH.
3:0	4'h0	ARC_CNT	Count retransmitted packets. The counter is reset when transmission of a new packet starts.

3.14.2.10 RX_ADDR_P0 (RW) Address: 0Ah

Bit 39	Bit 38	Bit 37	Bit 36	Bit 35	Bit 34	Bit 33	Bit 32
RX_ADDR_P0							

8'h70							
RW							
Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
RX_ADDR_P0							
8'h41							
RW							
Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16
RX_ADDR_P0							
8'h88							
RW							
Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
RX_ADDR_P0							
8'h20							
RW							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RX_ADDR_P0							
8'h46							
RW							

Description of Word

Bit	Value	Symbol	Description
39:0	40'7041 882046	RX_ADDR_P0	Receive address data pipe 0. 5 Bytes maximum length. (LSByte is written first. Write the number of bytes defined by SETUP_AW)

3.14.2.11 RX_ADDR_P1 (RW) Address: 0Bh

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RX_ADDR_P1							
8'hC2							
RW							

Description of Word

Bit	Value	Symbol	Description
7:0	8'hc2	RX_ADDR_P1	Receive address data pipe 2. Only LSB. MSBytes are equal to RX_ADDR_P0[39:8]

3.14.2.12 RX_ADDR_P2 (RW) Address: 0Ch

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RX_ADDR_P2							

8’hC3
RW

Description of Word

Bit	Value	Symbol	Description
7:0	8’hc3	RX_ADDR_P2	Receive address data pipe 2. Only LSB. MSBytes are equal to RX_ADDR_P0[39:8]

3.14.2.13 RX_ADDR_P3 (RW) Address: 0Dh

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RX_ADDR_P3							
8’hC4							
RW							

Description of Word

Bit	Value	Symbol	Description
7:0	8’hc4	RX_ADDR_P3	Receive address data pipe 3. Only LSB. MSBytes are equal to RX_ADDR_P0[39:8]

3.14.2.14 RX_ADDR_P4 (RW) Address: 0Eh

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RX_ADDR_P4							
8’hC5							
RW							

Description of Word

Bit	Value	Symbol	Description
7:0	8’hc5	RX_ADDR_P4	Receive address data pipe 4. Only LSB. MSBytes are equal to RX_ADDR_P0[39:8]

3.14.2.15 RX_ADDR_P5 (RW) Address: 0Fh

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RX_ADDR_P5							
8’hC6							
RW							

Description of Word

Bit	Value	Symbol	Description
7:0	8’hc6	RX_ADDR_P5	Receive address data pipe 5. Only LSB. MSBytes are equal to RX_ADDR_P0[39:8]

3.14.2.16 TX_ADDR(RW) Address: 10h

Bit 39	Bit 38	Bit 37	Bit 36	Bit 35	Bit 34	Bit 33	Bit 32
TX_ADDR							
8’h70							
RW							
Bit 31	**Bit 30**	**Bit 29**	**Bit 28**	**Bit 27**	**Bit 26**	**Bit 25**	**Bit 24**
TX_ADDR							
8’h41							
RW							
Bit 23	**Bit 22**	**Bit 21**	**Bit 20**	**Bit 19**	**Bit 18**	**Bit 17**	**Bit 16**
TX_ADDR							
8’h88							
RW							
Bit 15	**Bit 14**	**Bit 13**	**Bit 12**	**Bit 11**	**Bit 10**	**Bit 9**	**Bit 8**
TX_ADDR							
8’h20							
RW							
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
TX_ADDR							
8’h46							
RW							

Description of Word

Bit	Value	Symbol	Description
39:0	40’h704 1882046	TX_ADDR	Transmit address. Used for a PTX device only. (LSByte is written first)Set RX_ADDR_P0 equal to this address to handle automatic acknowledge if this is a PTX device with Protocol engine enabled.

3.14.2.17 RX_PW_P0 (RW) Address: 11h

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reserved		RX_PW_P0					
0		0					
RW		RW					

Description of Word

Bit	Value	Symbol	Description	
7:6	2'b00	Reserved	Only 0 allowed	
			0	Keep the current value
			1	Reset to default values
5:0	0	RX_PW_P0	Number of bytes in RX payload in data pipe 0 (1 to 32 bytes)	
			32	32 bytes
			…	…
			1	1 byte
			0	Pipe not used

3.14.2.18 RX_PW_P1 (RW) Address: 12h

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reserved		RX_PW_P1					
0		0					
RW		RW					

Description of Word

Bit	Value	Symbol	Description	
7:6	2'b00	Reserved	Only 0 allowed	
			0	Keep the current value
			1	Reset to default values
5:0	0	RX_PW_P1	Number of bytes in RX payload in data pipe 1 (1 to 32 bytes)	
			32	32 bytes
			…	…
			1	1 byte
			0	Pipe not used

3.14.2.19 RX_PW_P2 (RW) Address: 13h

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reserved		RX_PW_P2					
0		0					
RW		RW					

Description of Word

Bit	Value	Symbol	Description	
7:6	2'b00	Reserved	Only 0 allowed	
			0	Keep the current value

			1	Reset to default values
5:0	0	RX_PW_P2	Number of bytes in RX payload in data pipe 2 (1 to 32 bytes)	
			32	32 bytes
			…	…
			1	1 byte
			0	Pipe not used

3.14.2.20 RX_PW_P3 (RW) Address: 14h

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reserved		RX_PW_P3					
0		0					
RW		RW					

Description of Word

Bit	Value	Symbol	Description	
7:6	2’b00	Reserved	Only 0 allowed	
			0	Keep the current value
			1	Reset to default values
5:0	0	RX_PW_P3	Number of bytes in RX payload in data pipe 3 (1 to 32 bytes)	
			32	32 bytes
			…	…
			1	1 byte
			0	Pipe not used

3.14.2.21 RX_PW_P4 (RW) Address: 15h

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reserved		RX_PW_P4					
0		0					
RW		RW					

Description of Word

Bit	Value	Symbol	Description	
7:6	2’b00	Reserved	Only 0 allowed	
			0	Keep the current value
			1	Reset to default values
5:0	0	RX_PW_P4	Number of bytes in RX payload in data pipe 4 (1 to 32 bytes)	
			32	32 bytes

			…	…
			1	1 byte
			0	Pipe not used

3.14.2.22 RX_PW_P5 (RW) Address: 16h

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reserved		RX_PW_P4					
0		0					
RW		RW					

Description of Word

Bit	Value	Symbol	Description	
7:6	2'b00	Reserved	Only 0 allowed	
			0	Keep the current value
			1	Reset to default values
5:0	0	RX_PW_P5	Number of bytes in RX payload in data pipe 5 (1 to 32 bytes)	
			32	32 bytes
			…	…
			1	1 byte
			0	Pipe not used

3.14.2.23 FIFO_STATUS (RW) Address: 17h

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reserved	TX_REUSE_PL	TX_FULL	TX_EMPTY	Reserved		RX_FULL	RX_EMPTY
0	0	0	1	0		0	1
RW	R	R	R	RW		R	R

Description of Word

Bit	Value	Symbol	Description	
7	0	Reserved	Only '0' allowed	
			0	Keep the current value
			1	Reset to default values
6	0	TX_REUSE_PL	TX REUSE flag.	
			1	Tx data reused
			0	Tx data not reused
5	0	TX_FULL	TX FIFO full flag.	

			1	TX FIFO full
			0	Available locations in TX FIFO
4	1	TX_EMPTY	TX FIFO empty flag.	
			1	TX FIFO empty
			0	Data in TX FIFO
3:2	2'b00	Reserved	Only '00' allowed	
			0	Keep the current value
			1	Reset to default values
1	0	RX_FULL	RX FIFO full flag.	
			1	RX FIFO full
			0	Available locations in RX FIFO
0	1	RX_EMPTY	RX FIFO empty flag.	
			1	RX FIFO empty
			0	Data in RX FIFO

3.14.2.24 DYNPD (RW) Address: 1Ch

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reserved		DPL_P5	DPL_P4	DPL_P3	DPL_P2	DPL_P1	DPL_P0
0		0	0	0	0	0	0
RW		RW	RW	RW	RW	RW	RW

Description of Word

Bit	Value	Symbol	Description	
7:6	2'b00	Reserved	Only 0 allowed	
			0	Keep the current value
			1	Reset to default values
5	0	DPL_P5	Enable dynamic payload length data pipe 5. (Requires EN_DPL and ENAA_P5)	
4	0	DPL_P4	Enable dynamic payload length data pipe 4. (Requires EN_DPL and ENAA_P4)	
3	0	DPL_P3	Enable dynamic payload length data pipe 3. (Requires EN_DPL and ENAA_P3)	
2	0	DPL_P2	Enable dynamic payload length data pipe 2. (Requires EN_DPL and ENAA_P2)	
1	0	DPL_P1	Enable dynamic payload length data pipe 1. (Requires EN_DPL and ENAA_P1)	
0	0	DPL_P0	Enable dynamic payload length data pipe 0. (Requires EN_DPL and ENAA_P0)	

3.14.2.25 FEATURE (RW) Address: 1Dh

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reserved			bp_pd_pll48	bp_pd_rc2m	EN_DPL	EN_ACK_PAY	EN_DYN_ACK
0			0	0	0	0	0

RW					RW	RW	RW

Description of Word

Bit	Value	Symbol	Description
7:5	3'b00	Reserved	Reserved
4	0	bp_pd_pll48	1 indicates bypass power down pll48 clock
3	0	bp_pd_rc2m	1 indicates bypass power down rc2m clock
2	0	EN_DPL	Enables Dynamic Payload Length
1	0	EN_ACK_PAY	Enables Payload with ACK
0	0	EN_DYN_ACK	Enables the W_TX_PAYLOAD_NOACK command

3.14.2.26 SETUP_VALUE (RW) Address: 1Eh

Bit 39	Bit 38	Bit 37	Bit 36	Bit 35	Bit 34	Bit 33	Bit32
REG_LNA_WAIT							
8'h00							
RW							
Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit24
REG_MBG_WAIT							
8'h06							
RW							
Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16
RX_TM_CNT							
8'h80							
RW							
Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
TX_SETUP_VALUE							
8'h32							
RW							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RX_SETUP_VALUE							
8'h28							
RW							

Description of Word

Bit	Value	Symbol	Description
39:32	0	REG_LNA_WA	Lna wait counter

		IT	8'hff	255 cycle
			…	…
			1	1 cycle
			0	0 cycle
31:24	10	REG_MBG_WAIT	Main bandgap wait counter	
			8'hff	255us
			…	…
			1	1 us
			0	0 us
23:16	80	RX_TM_CNT	Rx timeout counter.	
			8'hff	255us
			…	…
			1	1 us
			0	0 us
15:8	8'h32	TX_SETUP_VALUE	TX_SETUP time, the time between Standby to TX mode	
			8'hff	255us
			…	…
			1	1 us
			0	0 us
7:0	8'h28	RX_SETUP_VALUE	RX_SETUP time, the time between Standby to RX mode	
			8'hff	255us
			…	…
			1	1 us
			0	0 us

3.14.2.27 PRE_GURD (RW) Address: 1Fh

Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
SPARE_REG[23:16]							
0							
RW							
Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
SPARE_REG[15:8]							
0							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TAIL_CTL			GRD_EN	GRD_CNT			
3'h3			1	4'h7			
RW			RW	RW			

Description of Word

Bit	Value	Symbol	Description	
23:8	0	SPARE_REG	Output to analogue	
7:5	3'h3	TAIL_CTL	Number of repeat bit after the CRC	
			7	7 repeat tail
			…	…
			1	1 repeat tail
			0	0 No repeat tail
4	1	GRD_EN	Pre-Guard enable	
3:0	4'h7	GRD_CNT	Number of Pre-Guard bit before preamble	
			4'hf	16 bit pre_guard
			…	…
			1	2 bit pre_guard
			0	1 bit pre_guard

3.14.3 Bank1

3.14.3.1 LINE (R) Address: 00h

Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
CID							
8'h05							
R							
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
CID							
8'h6c							
R							

Description of Word

Bit	Value	Symbol	Description

15:0	16’h056c	cid	

3.14.3.2 PLL_CTL0 (RW) Address: 01h

Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
SDM_EN	Pll_dac_en	PLL_RSTN_PFD	CRY_PD_REG	CRY_PD_MN	CAL_EN	PLL_FOFFSET_SEL	
1	1	1	0	0	0	0	
RW	RW	RW	RW	RW	RW	RW	
Bit 23	**Bit 22**	**Bit 21**	**Bit 20**	**Bit 19**	**Bit 18**	**Bit 17**	**Bit 16**
DAC_CAL-EN_REG	DAC_CAL_EN_MN	DAC_CALMODE_REG	BP_RC_BP	DOC_CAL_EN_REG	DOC_CAL_EN_MN_	DOC_DAC_MN	I_Q_RVS
0	0	1	0	0	0	0	0
RW	RW	RW	RW	RW	RW	RW	RW
Bit 15	**Bit 14**	**Bit 13**	**Bit 12**	**Bit 11**	**Bit 10**	**Bit 9**	**Bit 8**
PLL_TEST_EN	Reserved		PD_PLL_REG	PD_PLL_MN	PLL_TX_EN_REG	PLL_TX_EN_MN	AFC_COR_MN
0	0		0	0	0	0	0
RW	RW		RW	RW	RW	RW	RW
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
SDM_DITH_IN	SDM_DITH_EN	DAC_RANGE_MN	DAC_IN_MN	AFC_EN_REG	AFC_EN_MN	CTUNING_MN	FTUNING_MN
0	0	0	0	0	0	0	0
RW	RW	RW	RW	RW	RW	RW	RW

Description of Word

Bit	Value	Symbol	Description	
31	1	SDM_EN	SDM enable	
			1	Enable the SDM
			0	Disable the SDM
30	1	Pll_dac_en	Output DAC_EN to analog	
			0	Output 0
			1	Output 1

29	1	PLL_RSTN_PFD		PLL_RSTN_PFD control
			1	Set 1
			0	Set 0
28	0	CRY_PD_REG		Control the Power down of Crystal
			1	Power down the crystal
			0	Power up the crystal
27	0	CRY_PD_MN		Select the source of crystal power down
			1	CRY_PD = CRY_PD_REG
			0	CRY_PD = CRY_PD_FSM
26	0	CAL_EN		Calibration enable signal
			1	Enable the calibration when CE high 10us trigger the calibration
			0	Disable the calibration
25:24	0	PLL_FOFFSET_SEL		PLL_FOFFSET_SEL control
23	0	DAC_CAL_EN_REG		DAC calibration enable
			1	Enable the DAC calibration
			0	Disable the DAC calibration
22	0	DAC_CAL_EN_MN		Select the source of DAC_CAL_EN
			1	DAC_CAL_EN = DAC_CAL_REG
			0	DAC_CAL_EN = DAC_CAL_FSM
21	1	DAC_CALMODE_REG		Control the VCO gain
			1	Normal gain
			0	5 times of the normal gain
20	0	BP_RC_BP		Bypass RC_BP phase in the FSM
			1	Bypass RC_BP phase
			0	No bypass RC_BP phase
19	0	DOC_CAL_EN_REG		Enable the DOC calibration
			1	Enable the DOC calibration
			0	Disable the DOC calibration
18	0	DOC_CAL_EN_MN		Select the source of DOC_CAL_EN source
			1	DOC_CAL_EN = DOC_CAL_EN_REG
			0	DOC_CAL_EN = DOC_CAL_EN_FSM
17	0	DOC_DAC_MN		Select the source of DOC source
			1	DOC_DACI = DOC_DACI_REG DOC_DACQ=DOC_DACQ_REG
			0	DOC_DACI = DOC_DACI_FSM DOC_DACQ = DOC_DACQ_FSM
16	0	I_q_rvs		Only 0 allowed
			0	Keep the current value
			1	Reset to default values
15	0	PLL_TEST_EN		Enable the PLL in test mode
			1	PLL A_CNT and B_CNT come from register

			0	PLL A_CNT and B_CNT come from SDM
14:13	0	Reserved		
12	0	PD_PLL_REG	PLL power down control	
			1	Power down PLL
			0	Power up PLL
11	0	PD_PLL_MN	Select the source of PLL Power down	
			1	PD_PLL = PD_PLL_REG
			0	PD_PLL = PD_PLL_FSM
10	0	PLL_TX_EN_REG	PLL TX mode	
			1	PLL in TX mode
			0	PLL in RX mode
9	0	PLL_TX_EN_MN	Select the source of PLL_TX_EN	
			1	PLL_TX_EN = PLL_TX_EN_REG
			0	PLL_TX_EN = PLL_TX_EN_FSM
8	0	AFC_COR_MN	Select the source of AFC_COR	
			1	AFC_COR = AFC_COR_REG
			0	AFC_COR = AFC_COR_FSM
7	0	SDM_DITH_IN	SDM dither in value	
			1	Value 1
			0	Value 0
6	0	SDM_DITH_EN	SDM dither enable	
			1	Enable the SDM dither
			0	Disable the SDM dither
5	0	DAC_RANG_MN	Select the source of DAC_RANG	
			1	DAC_RANG = DAC_RANG_REG
			0	DAC_RANG = DAC_RANG_FSM
4	0	DAC_IN_MN	Select the source of DAC_IN	
			1	DAC_IN = DAC_IN_REG
			0	DAC_IN = DAC_IN_FSM
3	0	AFC_EN_REG	AFC enable	
			1	Enable the AFC
			0	Disable the AFC
2	0	AFC_EN_MN	Select the source of AFC_EN	
			1	AFC_EN = AFC_EN_REG
			0	AFC_EN = AFC_EN_FSM
1	0	CTUNING_MN	Select the source of CTUNING	
			1	CTUNING = CTUNING_REG
			0	CTUNING = CTUNING_FSM
0	0	FTUNING_MN	Select the source of FTUNING	
			1	FTUNING = FTUNING_REG
			0	FTUNING = FTUNING_FSM

3.14.3.3 PLL_CTL1 (RW) Address: 02h

Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
Soft_rst	SYNC_DET_DIS	PHY_FPGA	BP_HF	DC_BW		DC_RM_EN	BP_GAU
0	0	0	0	01		1	1
RW	RW	RW	RW	RW		RW	RW
Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16
REG_TX_PA_WAIT							
8'h10							
RW							
Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
Tx_PLL_WAIT							
8'h42							
RW							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
REG_AFC_WAIT							
8'h0							
RW							

Description of Word

Bit	Value	Symbol	Description	
31	0	Soft_rst	software reset (please note when write to the bit, the bank is switch to bank0)	
			0	Keep the current value
			1	write 1 reset the 6200
30	0	SYNC_DET_DIS	Disable AGC when sync detected	
			1	Disable
			0	Enable
29	0	PHY_FPGA	TX data and strobe come from FPGA	
			1	TX data and strobe come from FPGA
			0	TX data and strobe com
28	0	BP_HF	Bypass half band filter	
			1	Bypass
			0	Not bypass
27:26	0	DC_BW	DC remove factor	
			3	2^-7
			2	2^-6
			1	2^-5
			0	2^-4
25	0	DC_RM_EN	RX DC remove enable	

			1	Enable DC remove
			0	Disable DC remove
24	0	BP_GAU	Bypass Gauss filter	
			1	Bypass Gauss filter
			0	Not Bypass Gauss filter
23:16	8’h10	REG_TX_PA_WAIT	The time between power up PA to transmit data	
			8’hff	255 cycle
			…	…
			1	1 cycle
			0	0 cycle
15:8	8’h5a	Tx_PLL_WAIT	Pll lock wait time in Tx mode	
			8’hff	255 us
			…	…
			1	1 us
			0	0 us
7:0	8’h0	REG_AFC_WAIT	The time between RC done and AFC start	
			8’hff	255 us
			…	…
			1	1 us
			0	0 us

3.14.3.4 CAL_CTL (RW) Address: 03h

Bit 39	Bit 38	Bit 37	Bit 36	Bit 35	Bit 34	Bit 33	Bit 32
Tx_wait_cnt							
8’h50							
RW							
Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
Rx_pll_wait							
8’h28							
RW							
Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16
Bp_ch_chg	afc_w_sel		Scramble_en	Bp_dc	Bp_dac	Bp_afc	Bp_rc
0	2’b10		1	0	0	0	0
RW	RW		RW	RW	RW	RW	RW
Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
bp_vco_ldo	Psudo_rnd	Bp_rx_addr	Reg_vc_det_en	Bp_cp_diox	Vco_ldo_cal_reg		
0	0	0	0	1	0		
WR	RW	RW	RW	RW			
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Vco_ldo_cal_mn	Rc_cal_ctr_mn	Pll_rst_cnt	Rc_cal_ctl_reg
0	0	1	0
RW	RW	RW	RW

Description of Word

Bit	Value	Symbol	Description	
39:32	8’h50	Tx_wait_cnt	Wait cycles between retransmits	
31:24	8’h40	Rx_pll_wait	Pll lock wait time in rx mode Force_cal : before every transmit and receive do calibration	
23	0	Bp_ch_chg	When rf_ch not change do not do calibration when the force_cal is set. 0 : calibration even the channel is not change 1: do not calibration if the channel is no change	
22:21	2’b10	afc_w_sel	Select the AFC wait reference counter	
			0	512
			1	255
			2	127
			3	63
20	1	Scramble_en	Scramble enable in the transmit bit	
			0	Disable
			1	Enable
19	0	Bp_dc	Bypass DC calibration phase	
			0	Disable
			1	Enable
18	0	Bp_dac	Bypass DAC calibration phase	
			0	Disable
			1	Enable
17	0	Bp_afc	Bypass AFC calibration phase	
			0	Disable
			1	Enable
16	0	Bp_rc	Bypass RC calibration phase	
			0	Disable
			1	Enable
15	0	bp_vco_ldo	Bypass VCO_LDO calibration phase	
			0	Disable
			1	Enable
14	0	Psudo_rnd	Transmit Random data	
			0	Disable
			1	Enable
13	0	Bp_rx_addr	Bypass the RX_ADDR phase in the main FSM	

12	0	Reg_vc_det_en	Enable the detect pll calibration	
			0	Disable
			1	Enable
11	1	Bp_cp_diox	Bpassdio	
			0	Disable
			1	Enable
10:8	0	Vco_ldo_cal_reg	Vco_ldomamual set value	
7	0	Vco_ldo_cal_mn	Vcoldo calibration select	
			0	Vco_ldo_cal = vco_ldo_cal_fsm
			1	Vco_ldo_cal = vco_ldo_cal_reg
6	0	Rc_cal_ctr_mn	Rc calibration select	
			0	Rc_cal_ctr = rc_cal_ctr_fsm
			1	Rc_cal_ctr = rc_cal_ctr_reg
5	1	Pll_rst_cnt	Pll_rst_cnt	
			0	Output 0
			1	Output 1
4:0	0	Rc_cal_ctr_reg	Rc calibration register	

3.14.3.5 B_CNT_REG (RW) Address: 05h

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reserved			Pd_pkdet_reg	Pd_pkdet_mn	Reserved		
0			0	0	0		
R			RW	RW	RW		

<u>Description of Word</u>

Bit	Value	Symbol	Description
4	0	Pd_pkdet_reg	Pd_pkdet manual value
3	0	Pd_pkdet_mn	Pd_pkdet manual select
2:0	3’h0	B_CNT_REG	Control the B_CNT to the PLL

3.14.3.6 STATUS (RW) Address: 07h

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
BANK	RX_DR	TX_DS	MAX_RT	RX_P_NO			TX_FULL
0	0	0	0	3’b111			0
R	RW	RW	RW	RW			R

Description of Word

Bit	Value	Symbol	Description	
7	0	BANK	Register BANK status	
			1	Register R/W is to register BANK1
			0	Register R/W is to register BANK0
6	0	RX_DR	Data Ready RX FIFO interrupt. Asserted when new data arrives RX FIFO Write 1 to clear bit.	
5	0	TX_DS	Data Sent TX FIFO interrupt. Asserted when packet transmitted on TX. If AUTO_ACK isactivated, this bit is set high only when ACK is received. Write 1 to clear bit.	
4	0	MAX_RT	Maximum number of TX retransmits interruptWrite 1 to clear bit. If MAX_RT is asserted it must be cleared to enable further communication.	
3:1	3'b111	RX_P_NO	Data pipe number for the payload available for reading from RX_FIFO	
			111	RX FIFO Empty
			110	Not Used
			000-101	Data Pipe Number
0	0	TX_FULL	TX FIFO full flag	
			0	Available locations in TX FIFO
			1	TX FIFO full

3.14.3.7 STATE (RW) Address: 08h

Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
Reserved				CAL_ST_CS			
0				0			
RW				R			
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Reserved		STATE_CS					
0		0					
RW		R					

Description of Word

Bit	Value	Symbol	Description	
15:12	0	Reserved	Only 0 allowed	
			0	Keep the current value
			1	Reset to default values
11:8	4'b0	Cal_st_cs	Describe the state of calibration state machine	
7:6	0	Reserved	Only 0 allowed	
			0	Keep the current value

			1	Reset to default values
5:0	6'b0	State_cs	Describe the state of main state machine	

3.14.3.8 CHAN (RW) Address: 09h

Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
Chan_mn	Reserved	CHAN_FRAC_REG					
0	0	0					
RW	RW	RW					
Bit 23	**Bit 22**	**Bit 21**	**Bit 20**	**Bit 19**	**Bit 18**	**Bit 17**	**Bit 16**
CHAN_FRAC_REG							
0							
RW							
Bit 15	**Bit 14**	**Bit 13**	**Bit 12**	**Bit 11**	**Bit 10**	**Bit 9**	**Bit 8**
CHAN_FRAC_REG							CHAN_INT_REG
0							0
RW							RW
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
CHAN_INT_REG							
0							
RW							

Description of Word

Bit	Value	Symbol	Description	
31	1'b0	Chan_mn	Channel int and frac part select	
			0	Come from the calculation from rf_ch
			1	Come from register
30	2'b0	Reserved	Only 0 allowed	
			0	Keep the current value
			1	Reset to default values
29:9	0	CHAN_FRAC_REG	PLL frequency fragment pat	
8:0	0	CHAN_INT_REG	PLL frequency integer pat	

3.14.3.9 FDEV (RW) Address: 0ch

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reserved	FDEV						

0	0
R	RW

Description of Word

Bit	Value	Symbol	Description	
7	0	Reserved	Only 0 allowed	
			0	Keep the current value
			1	Reset to default values
6:0	6'b0	FDEV	The max offset of the frequency	

3.14.3.10 DAC_RANGE (RW) Address: 0dh

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reserved		DAC_RANGE_REG					
0		0					
R		RW					

Description of Word

Bit	Value	Symbol	Description	
7:6	0	Reserved	Only 0 allowed	
			0	Keep the current value
			1	Reset to default values
5:0	6'b0	DAC_RANGE_REG	DAC calibration Range control	

3.14.3.11 CTUNING (RW) Address: 0fh

Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
Reserved			CTUNING_REG_RX				
0			0				
R			RW				
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reserved			CTUNING_REG_TX				
0			0				
R			RW				

Description of Word

Bit	Value	Symbol	Description

15:13	0	Reserved	Only 0 allowed	
			0	Keep the current value
			1	Reset to default values
12:8	6'b0	CTUNING_REG_RX	AFC coarse tuning register control	
7:5	0	Reserved	Only 0 allowed	
			0	Keep the current value
			1	Reset to default values
4:0	6'b0	CTUNING_REG_TX	AFC coarse tuning register control	

3.14.3.12 FTUNING (RW) Address: 10h

Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
Reserved					FTUNING_REG_RX		
0							
R					RW		
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Reserved					FTUNING_REG_TX		
0							
R					RW		

Description of Word

Bit	Value	Symbol	Description	
15:11	0	Reserved	Only 0 allowed	
			0	Keep the current value
			1	Reset to default values
10:8	3'b0	CTUNING_REG_RX	AFC fine tuning register control	
7:3	0	Reserved	Only 0 allowed	
			0	Keep the current value
			1	Reset to default values
2:0	3'b0	CTUNING_REG_TX	AFC fine tuning register control	

3.14.3.13 RX_CTRL (RW) Address: 11h

Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit24
Rss_est_en	Rss_clr_en	Rssi_est_mode	xcorr_th_high[6:2]				
1	0	1	0	1	1	0	0

RW	RW	RW	RW	RW	RW	RW	RW
Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16
xcorr_th_high[1:0]		rssi_cond_th					rssi_cond_en
0	0	0	0	1	0	0	1
RW	RW	RW	RW	RW	RW	RW	RW
Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
buf_setl_len					buf_setl_confg	Spwr_th_mode	
1	1	0	0	0	0	1	0
RW	RW	RW	RW	RW	RW	RW	RW
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
rssi_k		bw_mode		sync_mode		h_idx	
0	1	0	1	0	0	0	1
RW	RW	RW	RW	RW	RW	RW	RW

Description of Word

Bit	Value	Symbol	Description
31	1	Rss_est_en	Rssiestimate enable
30	0	Rss_clr_en	Rssi clear enable
29	1	Rssi_est_mode	Rssiestimate mode
28:22	7'b0110000	xcorr_th_high	xcorr trigger threshold, high threshold for high snr packet
21:17	4'b0100	rssi_cond_th	don't active trigger when rssi>rsssi_cond_th
16	1	rssi_cond_en	rssi_cond_th enable
15:11	5'b11000	buf_setl_len	buffer settle configure symbols length
10	0	buf_setl_confg	'1' - buffer settle using corrlen, '0' - buffer settle using BUF_SETL_LEN configured;
9:8	2'b10	Spwr_th_mode	
7:6	1	rssi_k	Default: 2^-5(about 80KHz bandwidth); when 0, 2^-4 when 1, 2^-5 when 2, 2^-6 when 3, 2^-7
5:4	1	bw_mode	bandwidth of the one-order loop: when 0, 2^-4 when 1, 2^-5 when 2, 2^-6 when 3, 2^-7
3:2	0	sync_mode	When 0, 4 address bytes are used for sync When 1, 1preamble byte and 3 address bytes are used for sync When 2, only 3 address bytes are used for sync

1:0	1	h_idx	modualtion index: when 0: 0.25, when 1: 0.32, when 2: 0.50;

3.14.3.14 FAGC_CTRL (RW) Address: 12h

Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit24
Reserved		Reserved					
0		0	0	0	0	0	0
RW	RW	RW	RW	RW	RW	RW	RW
Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16
Sig_dbm_est							
0	0	0	0	0	0	0	0
RW	RW	RW	RW	RW	RW	RW	RW
Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
Reserved	rx_fagc_ref		lwin	gfagc_wen	gfagc_reg[10:8]		
0	1	0	0	0	0	0	0
RW	RW	RW	RW	RW	RW	RW	RW
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
gfagc_reg[7:0]							
0	0	0	0	0	0	0	0
RW	RW	RW	RW	RW	RW	RW	RW

Description of Word

Bit	Value	Symbol	Description
31:30	0	Reserved	
29:24	0	Snr_est	Snr estimate
23:16	0	Sig_dbm_est	Signal estimate dbm
15	0	Reserved	
14:13	2	rx_fagc_ref	[-6, -3, 0, 3]dB from 0.5;
12	0	lwin	0-64; 1-128 length;
11	0	gfagc_wen	Gfagc_o write enable signal
10:0	0	gfagc_reg	Gain of the fine AGC calculated

3.14.3.15 FAGC_CTRL_1 (RW) Address: 13h

Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit24
cry_500k_pd_reg	cry_500k_pd_mn	Force_cal	pkdet_vrefc			pkdet_vref2c	

0	0	1	3’h2			1	
RW	RW	RW	RW			RW	
Bit 23	**Bit 22**	**Bit 21**	**Bit 20**	**Bit 19**	**Bit 18**	**Bit 17**	**Bit 16**
agc_sel	agc_dpd_lo_thr[10:4]						
1	7’h8						
RW	RW						
Bit 15	**Bit 14**	**Bit 13**	**Bit 12**	**Bit 11**	**Bit 10**	**Bit 9**	**Bit 8**
agc_dpd_lo_thr[3:0]				Agc_reset_cnt [11:8]			
1				4’h4			
RW				RW			
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Agc_reset_cnt[7:0]							
8’h80							
RW							

Description of Word

Bit	Value	Symbol	Description	
31	0	cry_500k_pd_reg	500k crystal power down control	
30	0	cry_500k_pd_mn	500k crystal power down control select. 1: register 0: state machine	
29	1	Force_cal	Before every transmit and receive do calibration	
28:26	3’b010	pkdet_vrefc	Pkdetvrefc value	
25:24	1	pkdet_vref2c	Pkdet vref2c value	
23	0	agc_sel	Agc select	
			0	Select agc 0
			1	Select agc 1
22:12	11’h081	agc_dpd_lo_thr	Agc_dpd_lo threshold	
11:0	12’h400	Agc_reset_cnt	Agc reset counter	

3.14.3.16 DOC_DACI (RW) Address: 1ah

Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Reserved	doc_daci						
0	0						
RW	RW						

Description of Word

Bit	Value	Symbol	Description

7	0	Reserved	Only 0 allowed	
			0	Keep the current value
			1	Reset to default values
6:0	0	doc_daci	Doc calibrationdaci value	

3.14.3.17 DOC_DACQ (RW) Address: 1bh

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reserved	doc_dacq						
0	0						
RW	RW						

Description of Word

Bit	Value	Symbol	Description	
7	0	Reserved	Only 0 allowed	
			0	Keep the current value
			1	Reset to default values
6:0	0	doc_dacq	Doc calibration dacq value	

3.14.3.18 AGC_CTRL (RW) Address: 1ch

Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit24
test_pat							
0	0	0	0	0	0	0	0
RW	RW	RW	RW	RW	RW	RW	RW
Bit 23	**Bit 22**	**Bit 21**	**Bit 20**	**Bit 19**	**Bit 18**	**Bit 17**	**Bit 16**
test_pat_en	adp_samp_mode			agc_samp_mode		agc_dpd_mode	agc_dpd_thr_db[5]
0	0	0	0	1	0	0	1
RW	RW	RW	RW	RW	RW	RW	RW
Bit 15	**Bit 14**	**Bit 13**	**Bit 12**	**Bit 11**	**Bit 10**	**Bit 9**	**Bit 8**
agc_dpd_thr_db[4:0]					agc_dpd_thr[10:8]		
1	0	0	1	1	0	1	0
RW	RW	RW	RW	RW	RW	RW	RW
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
agc_dpd_thr[7:0]							
0	0	0	0	0	0	0	0
RW	RW	RW	RW	RW	RW	RW	RW

Description of Word

Bit	Value	Symbol	Description
31:24	0	test_pat[7:0]	Test pattern of Analog
23	0	test_pat_en	Test pattern enable
22:20	0	adp_samp_mode	Number of clock cycles delay between every two apd detection: When 0, 0 cycles When 1, 2 cycles When 2, 4 cycles When 3, 8 cycles When 4, 16 cycles When 5, 32 cycles
19:18	2	agc_samp_mode	The number of shift registers which are required for Dpdcalculation. When 0, 2 registers When 1, 4 registers When 2, 8 registers When 3, 16 registers
17	0	agc_dpd_mode	'0': mean of sample amplitudes, '1': max of sample amplitudes;
16:11	33	agc_dpd_thr_db	dB value of dpd_hi_th minus backoff dB value of the Dpd which is 3 in default.
10:0	200	agc_dpd_thr	Default thresholdwhendpd is detected.

3.14.3.19 AGC_GAIN (RW) Address: 1dh

Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit24
agc_gain_mn	apd_clr_cnt_th					apd_det_cnt_th[4:3]	
0	0	0	1	1	1	0	0
RW	RW	RW	RW	RW	RW	RW	RW
Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16
apd_det_cnt_th[2:0]			dpd_clr_cnt_th[5:1]				
1	1	1	0	1	0	1	1
RW	RW	RW	RW	RW	RW	RW	RW
Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
dpd_clr_cnt_th [0]	reserved		Agc_apd_state_reg[1:0]		Agc_dpd_state_reg[2:0]		
1	1	0	0	0	0	0	1
RW	RW	RW	RW	RW	RW	RW	RW
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Agc_dpd_state_reg[10:3]							
0	0	0	0	0	0	0	1
RW	RW	RW	RW	RW	RW	RW	RW

Description of Word

Bit	Value	Symbol	Description

31	1	agc_gain_mn	Agc output gain is get from software or hardware caculation	
			1	From software
			0	From hardware
30:26	7	apd_clr_cnt_th [4:0]	Analog Gsw+Gsetl time interval	
25:21	6	apd_det_cnt_th [4:0]	Apd detection time interval	
20:15	6’h17	dpd_clr_cnt_th [5:0]	DigitalGsw+Gsetl time interval, and add Dpd detection time interval	
14:13	2	reserved		
12:11	0	Agc_apd_state_reg	When agc_gain_mn is 1, the value read from this register is the Lna gain given by the software. When agc_gain_mn is 0, the value read from this register is the Lna gain calculated from the hardware module	
10:0	101	Agc_dpd_state_reg	When agc_gain_mn is 1, the value read from this register is the filter gain given by the software. When agc_gain_mn is 0, the value read from this register is the filter gain calculated from the hardware module	

3.14.3.20 RF_IVGEN (RW) Address: 1eh

Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit24
Calib_rx_reg	Tx_0_1_rvs	Pd_lna_reg	Pd_lna_mn	Pd_pa_reg	Pd_pa_mn	Tia_lobias	
0	0	0	0	0	0	0	1
RW	RW	RW	RW	RW	RW	RW	RW
Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16
Pd_adc_ldo_reg	Pd_adc_ldo_mn	Pd_mix_reg	Pd_mix_mn	Pd_bg_reg	Pd_bg_mn	Bm_lna	
0	0	0	0	0	0	0	1
RW	RW	RW	RW	RW	RW	RW	RW
Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
Bm_filter		filter_mcap1			filter_mcap2		
0	1	1	0	0	1	0	0
RW	RW	RW	RW	RW	RW	RW	RW
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Pa_voltage	Pd_xtal_reg	Pd_xtal_mn	Xtal_cc				
0	0	0	1	1	0	0	1
RW	RW	RW	RW	RW	RW	RW	RW

Description of Word

Bit	Value	Symbol	Description	
31	0	Calib_rx_reg	Manual calibration TX/RX select	
			1	Manual calibration for RX

			0	Manual calibration for TX
30	0	Tx_0_1_rvs	TX 0/1 reverse	
			1	Reverse the 0/1 when transmit
			0	Not reverse
29	0	Pd_lna_reg	Pd_lna manual value	
28	0	Pd_lna_mn	Pd_lna manual select	
			1	from register
			0	From FSM
27	0	Pd_pa_reg	Pd_pa manual value	
26	0	Pd_pa_mn	Pd_pa manual select	
			1	from register
			0	From FSM
25:24	1	Tia_lobias	Output to analogue	
23	0	Pd_adc_ldo_reg	Pd_adc_ldo manual value	
22	0	Pd_adc_ldo_mn	Pd_adc_ldo manual select	
			1	from register
			0	From FSM
21	0	Pd_mix_reg	Pd_mix manual value	
20	0	Pd_mix_mn	Pd_mix manual select	
			1	from register
			0	From FSM
19	0	Pd_bg_reg	Pd_bg manual value	
18	0	Pd_bg_mn	Pd_bg manual select	
			1	from register
			0	From FSM
17:16	0	Bm_lna	Output to analogue	
15:14	1	Bm_filter	Output to analogue	
13:11	4	filter_mcap1	Output to analogue	
10:8	4	filter_mcap2	Output to analogue	
7	0	Pa_voltage	Pa voltage output to analogue	
6	0	Pd_xtal_reg	Pd_xtal manual value	
5	0	Pd_xtal_mn	Pd_xtal manual select	
4:0	5'b11111	Xtal_cc	Output to analogue	

3.14.3.21 TEST_PKDET (RW) Address: 1fh

Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit24
test_en	Xcorr_filt_en	sync_burst_cnt				pll_icp_sel	
0	1	1	1	1	1	0	1
RW	RW	RW	RW	RW	RW	RW	RW
Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16

test_mode			test_point_sel1			test_point_sel0	
0	0	0	0	0	0	0	0
RW	RW	RW	RW	RW	RW	RW	RW
Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
pkdet_vref							
0	0	0	1	0	0	0	0
RW	RW	RW	RW	RW	RW	RW	RW
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
xtal_resc		bm_xtal	Dc_est_bypass	pll_vdiv2_sel		pll_det_set	
0	0	1	0	1	0	0	1
RW	RW	RW	RW	RW	RW	RW	RW

Description of Word

Bit	Value	Symbol	Description	
31	0	test_en	Test enable output to analogue	
30	1	Xcorr_filt_en		
29:26	4'hf	sync_burst_cnt	The number of symbols delay are required before generating the sync_det pulse	
25:24	1	pll_icp_sel [1:0]	Output to analogue	
23:21	0	test_mode[2:0]	Output to analogue	
20:18	0	test_point_sel1 [2:0]	Output to analogue	
17:16	00	test_point_sel0 [1:0]	Output to analogue	
15:8	8'h10	pkdet_vref[7:0]	Output to analogue	
7:6	0	xtal_resc[1:0]	Output to analogue	
5	1	bm_xtal	Output to analogue	

4	0	Dc_est_bypass		
3:2	2	pll_vdiv2_sel [1:0]	Output to analogue	
1:0	1	pll_det_set[1:0]	Output to analogue	

4 MCU

4.1 Features

- Control Unit
  - 8-bit instruction decoder
  - Reduced instruction cycle time (up to 12 times in respect to standard 80C51)
- Arithmetic-Logic Unit
  - 8-bit arithmetic and logical operations
  - Boolean manipulations
- Three 16-bit Timers/Counters
  - 80C51-like Timer 0 & 1
- Interrupt Controller
  - Four Priority Levels with 8 interrupt sources
- Memory interface
  - 16-bit address bus
  - Dual Data Pointer for fast data block transfer

4.2 Instructions in Functional Order

4.2.1 Arithmetic Operations

Mnemonic	Description	Code	Bytes	Cycles
ADD A,Rn	Add register to accumulator	0x28-0x2F	1	1
ADD A,direct	Add directly addressed data to accumulator	0x25	2	2
ADD A,@Ri	Add indirectly addressed data to accumulator	0x26-0x27	1	2
ADD A,#data	Add immediate data to accumulator	0x24	2	2
ADDC A,Rn	Add register to accumulator with carry	0x38-0x3F	1	1
ADDC A,direct	Add directly addressed data to accumulator with carry	0x35	2	2
ADDC A,@Ri	Add indirectly addressed data to accumulator with carry	0x36-0x37	1	2
ADDC A,#data	Add immediate data toaccumulator with carry	0x34	2	2
SUBB A,Rn	Subtract register from accumulator with borrow	0x98-0x9F	1	1
SUBB A,direct	Subtract directly addressed data from accumulator with borrow	0x95	2	2
SUBB A,@Ri	Subtract indirectly addressed data from accumulator with borrow	0x96-0x97	1	2
SUBB A,#data	Subtract immediate data from accumulator with borrow	Ox94	2	2
INC A	Increment accumulator	0x04	1	1
INC Rn	Increment register	0x08-0x0F	1	1
INC direct	Increment directly addressed location	0x05	2	2

INC @Ri	Increment indirectly addressed location	0x06-0x07	1	2
INC DPTR	Increment data pointer	0xA3	1	1
DEC A	Decrement accumulator	0x14	1	1
DEC Rn	Decrement register	0x18-0x1F	1	1
DEC direct	Decrement directly addressed location	0x15	2	2
DEC @Ri	Decrement indirectly addressed location	0x16-0x17	1	2
MUL AB	Multiply A and B	0xA4	1	4
DIV	Divide A by B	0x84	1	4
DA A	Decimally adjust accumulator	0xD4	1	1

4.2.2 Logic Operations

Mnemonic	Description	Code	Bytes	Cycles
ANL A,Rn	AND register to accumulator	0x58-0x5F	1	1
ANL A,direct	AND directly addressed data to accumulator	0x55	2	2
ANL A,@Ri	AND indirectly addressed data to accumulator	0x56-0x57	1	2
ANL A,#data	AND immediate data to accumulator	0x54	2	2
ANL direct,A	AND accumulator to directly addressed location	0x52	2	2
ANL direct,#data	AND immediate data to directly addressed location	0x53	3	3
ORL A,Rn	OR register to accumulator	0x48-0x4F	1	1
ORL A,direct	OR directly addressed data to accumulator	0x45	2	2
ORL A,@Ri	OR indirectly addressed data to accumulator	0x46-0x47	1	2
ORL A,#data	OR immediate data to accumulator	0x44	2	2
ORL direct,A	OR accumulator to directly addressed location	0x42	2	2
ORL direct,#data	OR immediate data todirectly addressed location	0x43	3	3
XRL A,Rn	Exclusive OR registerto accumulator	0x68-0x6F	1	1
XRL A,direct	Exclusive OR directly addressed data to accumulator	0x65	2	2
XRL A,@Ri	Exclusive OR indirectly addressed data to accumulator	0x66-0x67	1	2
XRL A,#data	Exclusive OR immediate data to accumulator	0x64	2	2
XRL direct,A	Exclusive OR accumulator to directly addressed location	0x62	2	2
XRL direct,#data	Exclusive ORimmediate data to directly addressed location	0x63	3	3
CLR A	Clear accumulator	0xE4	1	1
CPL A	Complement accumulator	0xF4	1	1
RL A	Rotate accumulator left	0x23	1	1
RLC A	Rotate accumulator left through carry	0x33	1	1
RR A	Rotate accumulator right	0x03	1	1
RRC A	Rotate accumulator right through carry	0x13	1	1
SWAP A	Swap nibbles within the accumulator	0xC4	1	1

4.2.3 Data Transfer Operations

Mnemonic	Description	Code	Bytes	Cycles
MOV A,Rn	Move register to accumulator	0xE8-0xEF	1	1
MOV A,direct	Move directly addressed data to accumulator	0xE5	2	2
MOV A,@Ri	Move indirectly addressed data to accumulator	0xE6-0xE7	1	2
MOV A,#data	Move immediate data to accumulator	0x74	2	2
MOV Rn,A	Move accumulator to register	0xF8-0xFF	1	1
MOV Rn,direct	Move directly addressed data to register	0xA8-0xAF	2	2
MOV Rn,#data	Move immediate data to register	0x78-0x7F	2	2
MOV direct,A	Move accumulator to direct	0xF5	2	2
MOV direct,Rn	Move register to direct	0x88-0x8F	2	2
MOV direct1,direct2	Move directly addressed data to directly addressed location	0x85	3	3
MOV direct,@Ri	Move indirectly addressed data to directly addressed location	0x86-0x87	2	2
MOV direct,#data	Move immediate data to directly addressed location	0x75	3	3
MOV @Ri,A	Move accumulator to indirectly addressed location	0xF6-0xF7	1	1
MOV @Ri,direct	Move directly addressed data to indirectly addressed location	0xA6-0xA7	2	2
MOV @Ri,#data	Move immediate data to in directly addressed location	0x76-0x77	2	2
MOV DPTR,#data16	Load data pointer with a 16-bit immediate	0x90	3	3
MOVCA,@A+DPTR	Load accumulator with a code byte relative to DPTR	0x93	1	3
MOVC A,@A+PC	Load accumulator with a code byte relative to PC	0x83	1	3
MOVX A,@Ri	Move external RAM (8-bit addr.) to accumulatora	0xE2-0xE3	1	3-10
MOVX A,@DPTR	Move external RAM (16-bit addr.) to accumulatora	0xE0	1	3-10
MOVX @Ri,A	Move accumulator to external RAM (8-bit addr.)a	0xF2-0xF3	1	3-12
MOVX @DPTR,A	Move accumulator to external RAM (16-bit addr.)a	0xF0	1	3-12
PUSH direct	Push directly addressed data onto stack	0xC0	2	2
POP direct	Pop directly addressed location from stack	0xD0	2	2
XCH A,Rn	Exchange register with accumulator	0xC8-0xCF	1	1
XCH A,direct	Exchange directly addressed location with accumulator	0xC5	2	2
XCH A,@Ri	Exchange indirect RAM with accumulator	0xC6-0xC7	1	2
XCHD A,@Ri	Exchangelow-ordernibbles of indirectand accumulator	0xD6-0xD7	1	2

Note:
a、The MOVX instructions perform one of two actions depending on the state of “pmw” bit (pcon.4).

4.2.4 Program Branches

Mnemonic	Description	Code	Bytes	Cycles
ACALL addr11	Absolute subroutine call	xxx10001b	2	2
LCALL addr16	Long subroutine call	0x12	3	3
RET	Return from subroutine	0x22	1	4

RETI	Return from interrupt	0x32	1	4
AJMP addr11	Absolute jump	xxx00001b	2	2
LJMP addr16	Long jump	0x02	3	3
SJMP rel	Short jump (relative address)	0x80	2	2
JMP @A+DPTR	Jump indirect relative to the DPTR	0x73	1	3
JZ rel	Jump if accumulator is zero	0x60	2	3
JNZ rel	Jump if accumulator is not zero	0x70	2	3
JC rel	Jump if carry flag is set	0x40	2	3
JNC	Jump if carry flag is not set	0x50	2	3
JB bit,rel	Jump if directly addressed bit is set	0x20	3	4
JNB bit,rel	Jump if directly addressed bit is not set	0x30	3	4
JBC bit,rel	Jump if directly addressed bit is set and clear bit	0x10	3	4
CJNE A,direct,rel	Compare directly addressed data to accumulator and jump if not equal	0xB5	3	4
CJNE A,#data,rel	Compare immediate data to accumulator and jump if not equal	0xB4	3	4
CJNERn,#data,rel	Compare immediate data to register and jump if not equal	0xB8-0xBF	3	4
CJNE @Ri,#data,rel	Compare immediate to indirect and jump if not equal	B6-B7	3	5
DJNZ Rn,rel	Decrement register and jump if not zero	D8-DF	2	3
DJNZ direct,rel	Decrement directly addressed location and jumpif not zero	D5	3	4
NOP	No operation	00	1	1

4.2.5 Boolean Manipulation

Mnemonic	Description	Code	Bytes	Cycles
CLR C	Clear carry flag	0xC3	1	1
CLR bit	Clear directly addressed bit	0xC2	2	2
SETB C	Set carry flag	0xD3	1	1
SETB bit	Set directly addressed bit	0xD2	2	2
CPL C	Complement carry flag	0xB3	1	2
CPL bit	Complement directly addressed bit	0xB2	2	2
ANL C,bit	AND directly addressed bit to carry flag	0x82	2	2
ANL C,/bit	AND complement of directly addressed bit to carry	0xB0	2	2
ORL C,bit	OR directly addressed bit to carry flag	0x72	2	2
ORL C,/bit	OR complement of directly addressed bit to carry	0xA0	2	2
MOV C,bit	Move directly addressed bit to carry flag	0xA2	2	2
MOV bit,C	Move carry flag to directly addressed bit	0x92	2	2

The duration of each instruction can becalculated using the formula below.

If (BYTES > 1 or CYCLES = 1) then

DURATION = CYCLES + (BYTES+R)*P + X*D

else

DURATION = CYCLES + (2+R)*P + X*D

Where:

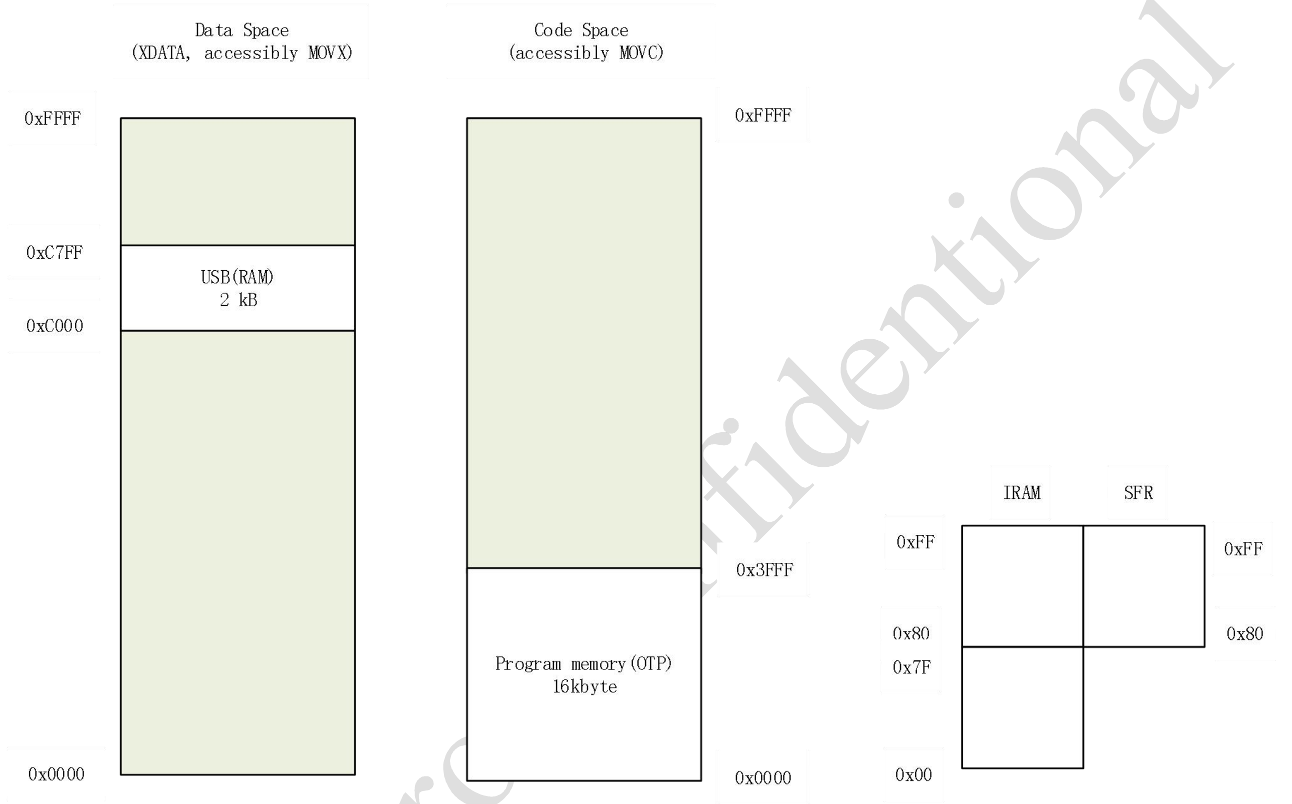
- BYTES is the number of bytes for the instruction (see tables above)
- CYCLES is the number of cycles for no wait states (see tables above)
- R = 1 for the MOVC instruction, otherwise R = 0
- X = 1 for MOVX instructions, otherwise X = 0
- P = number of program memory wait states (= “ckcon[6:4]”)
- D = number of data memory wait states (= “ckcon[2:0]”).

In the Program Memory Write mode (PMW) the formula for MOVX is as follows:

DURATION = CYCLES + (2+X)*P

5 Memory and I/O organization

The MCU has 64 kB of separate address space for code and data, an area of 256 byte for internal data (IRAM) and an area of 128 byte for Special Function Registers (SFR).
The memory block has a default setting of 16 kB program memory (OTP).



The lower 128 bytes of the IRAM contains work registers (0x00–0x1F) and bit addressable memory (0x20–0x2F). The upper half can only be accessed by indirect addressing.

The lowest 32 bytes of the IRAM form four banks, each consisting of eight registers (R0–R7). Two bits of the program memory status word (PSW) select which bank is used. The next 16 bytes of memory form a block of bit-addressable memory, accessible through bit addresses 0x00–0x7.

5.1 CPU Special Function Registers

5.1.1 Accumulator – ACC

Accumulator is used by most of the MCU instructions to hold the operand and to store the result of an operation. The mnemonics for accumulator specific instructions refer to accumulator as A, not ACC.

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0xE0	acc.7	acc.6	acc.5	acc.4	acc.3	acc.2	acc.1	acc.0

5.1.2 B register – B

The B register is used during multiplying and division instructions. It can also be used as a scratch-pad register to hold temporary data.

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0xF0	b.7	b.6	b.5	b.4	b.3	b.2	b.1	b.0

5.1.3 Program Status Word Register – PSW

The PSW register contains status bits that reflect the current state of the MCU.

Note: The Parity bit can only be modified by hardware upon the state of ACC register.

Address	Bit	Name	Description
0xD0	7	cy	Carry flag: Carry bit in arithmetic operations and accumulator for Boolean operations.
	6	ac	Auxiliary Carry flag: Set if there is a carry-out from 3rd bit of Accumulator in BCD operations
	5	f0	General purpose flag 0
	4:3	rs	Register bank select, bank 0..3 (0x00–0x07, 0x08–0x0f, 0x10–0x17, 0x18–0x1f)
	2	ov	Overflow flag: Set if overflow in Accumulator during arithmetic operations
	1	f1	General purpose flag 1
	0	p	Parity flag: Set if odd number of '1' in ACC

5.1.4 Stack Pointer – SP

This register points to the top of stack in internal data memory space. It is used to store the return address of a program before executing interrupt routine or subprograms. The SP is incremented before executing PUSH or CALL instruction and it is decremented after executing POP or RET(I) instruction (it always points to the top of stack).

Address	Register name
0x81	SP

5.1.5 Data Pointer – DPH, DPL

Address	Register name

0x82	DPL
0x83	DPH

The Data Pointer Registers can be accessed through DPL and DPH. The actual data pointer is selected by DPS register.These registers are intended to hold 16-bit address in the indirect addressing mode used by MOVX (move external memory), MOVC (move program memory) or JMP (computed branch) instructions. They may be manipulated as 16-bit register or as two separate 8-bit registers. DPH holds higher byte and DPL holds lower byte of indirect address.It is generally used to access external code or data space (for example, MOVC A, @A+DPTR or MOV A, @DPTR respectively).

5.1.6 Data Pointer 1 – DPH1, DPL1

Address	Register name
0x84	DPL1
0x85	DPH1

The Data Pointer Register 1 can be accessed through DPL1 and DPH1. The actual data pointer is selected by DPS register.
These registers are intended to hold 16-bit address in the indirect addressing mode used by MOVX (move external memory), MOVC (move program memory) or JMP (computed branch) instructions. They may be manipulated as 16-bit register or as two separate 8-bit registers. DPH1 holds higher byte and DPL1 holds lower byte of indirect address.
It is generally used to access external code or data space (for example, MOVC A,@A+DPTR or MOV A,@DPTR respectively).The Data Pointer 1 is an extension to the standard 8051 architecture to speed up block data transfers.

5.1.7 Data Pointer Select Register – DPS

The MCU contains two Data Pointer registers. Both of them can be used as 16-bits address source for indirect addressing. The DPS register serves for selecting active data pointer register.

Address	Bit	Name	Description
0x92	7:1	-	Not used
	0	dps	Data Pointer Select. 0: select DPH:DPL, 1: select DPH1:DPL1

5.1.8 PCON register

The PCON register is used to control the Program Memory Write Mode and Serial Port 0 baud rate select.
Default value: 0x08

Address	Bit	Name	Description
0x87	7		
	6	-	Not used

	5	isr_tm	Interrupt Service Routine Test Mode flag: When set to 1, the interrupt vectors assigned to Timer 0 & 1, Serial Port 0 & 1 interfaces can be triggered only with the use of external inputs of the core
	4	pmw	Program memory write mode: 1: MOVX instructions will access memory code space 0: MOVX instructions will access memory data space
	3	-	Not used
	2	gf0	General purpose flag 0
	1	-	Not used. This bit must always be cleared. Always read as 0.
	0	-	Not used. This bit must always be cleared. Always read as 0.

5.1.9 Clock Control Register – CKCON

The content of this register defines the number of internally generated wait states that occur during read/write accesses to external data and program memory. It also controls the type of write access to either of the memory spaces.
Default value: 0x71

Address	Bit	Name	Description
0x8E	7	-	Keep setting 1
	6:4	pw	Program memory wait state control
	3	-	Keep setting 1
	2:0	ds	External data memory stretch cycle control.

5.1.10 Software Reset Register SRST

The software reset will be accomplished through the SRST SFR register. The contents of this register are presented below.

Address	Bit	Name	Description
0xF7	7:1	-	Not used
	0	srstreq	Software reset request. Writing ‘0’ value to this bit will have no effect. Single writing ‘1’ value to this bit will have no effect. Double writing ‘1’ value (in two consecutive instructions) will generate an internal software reset. Reading this bit will inform about the reset source: if ‘0’ – source of last reset sequence was not a software reset (hardware, watchdog or pin reset); If ‘1’ – source of last reset sequence was a software reset (caused by double writing ‘1’ value to the “srstreq” bit).

5.1.11 Special Function Register Map

Address	X000	X001	X010	X011	X100	X101	X110	X111
0xF8–0xFF	FSR							
0xF0–0xF7	B				CPLDOCON	CIDH	CIDL	SRST
0xE8–0xEF	RFCON							
0xE0–0xE7	ACC	CFG_DFEN_P0						
0xD8–0xDF				USBPCON1				
0xD0–0xD7	PSW	-						
0xC8–0xCF		-					USBCON	USBPCON
0xC0–0xC7								
0xB8–0xBF	IEN1	IP1						
0xB0–0xB7		RSTREAS		RTC_CON	RTC_CMPL	RTC_CMPM	RTC_CMPH	
0xA8–0xAF	IEN0	IP0			CLKGATE	CLKLFCTRL		
0xA0–0xA7				CLKCTRL	PWRDWN	WUCON	INTEXP	MEMCON
0x98–0x9F			-				P0CON	
0x90–0x97			DPS	P0DIR				
0x88–0x8F	TCON	TMOD	TL0	TL1	TH0	TH1	CKCON	
0x80–0x87	P0	SP	DPL	DPH	DPL1	DPH1		PCON

6 OTP memory

This section describes the operation of the embedded OTP (One Time Programmable) memory. The primary use for this memory is for read only program and data storage, but the MCU may also perform write operations, for instance to store pairing information persistently.

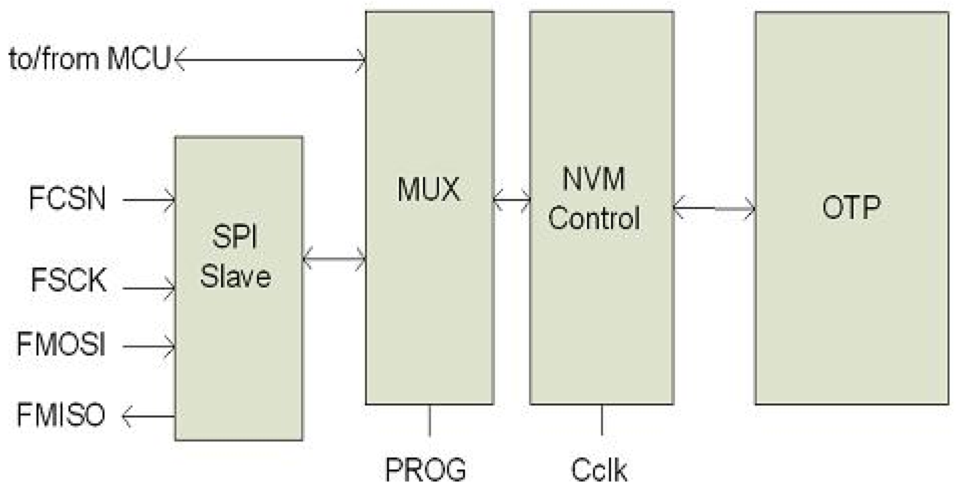
The OTP memory is configured and programmed through an external SPI slave interface. After programming, read and write operations from the external interfaces can be disabled for code protection.

6.1 Features

- 16 kB code memory
- Up to 2 kB can be mapped to data memory
- Direct SPI programmable
- Read and write protection

6.2 Block diagram

OTP is accessible by SPI slave during programming and MCU as. A memory controller generates the necessary signals for accessing the OTP memory. FMOSI and FMISO share one GPIO port which is P0.2.



6.3 Functional description

The OTP block gives the MCU its code space for program storage and NV (not violation) memory space for storing of application data. The NV memory can be accessed by the MCU through both normal code and data space operations.

6.3.1 Using the NV data memory

By default, the 2 kB NV memory is addressed by MCU as code memory in the range of 0x3800~0x3FFF. By setting the value of MEMCON, this NV memory can be addressed as XDATA memory in the range of 0x3800~0x3FFF. This memory is physically mapped (for SPI access) to OTP memory area 0x3800–0x3FFF.

6.3.2 OTP memory configuration

6.3.2.1 OTP read and write protection

The last byte of the OTP memory which is at address 0x3FFF is used to disable read and write to the OTP from the SPI interface. This byte can be accessed by MCU and SPI interface. When this byte is:

- 0x00: OTP block is accessible from external interfaces
- Other value: No read/ write of OTP block from external interfaces

6.3.2.2 Memory status register FSR

Address	Bit	Name	Reset value	SPI	SFR	Description
0xF8	7		1	R	R	
	6	VRF_MODE	0	RW	RW	Only used in program and verify operation 1: one cycle to read(50ns OTP) 0: two cycle to read(50ns and 70ns OTP)
	5	WEN	0	RW	RW	OTP write enable latch. Enables OTP write operations from SPI or MCU.
	4	RDYN	1	R	R	OTP ready flag, active low. Before write date to OTP, you must check if OTP has been ready.
	3	prog_fail	0	R	R	1 indicate OTP program and verify process is failed
	2	RDISMB	0	R	R	OTP main block read and write protection enabled 0:ExternalSPI have full access (read and write) to the OTP. 1: External SPI do not access to the OTP.
	1	TSTSTU	0	R	R	When 1, indicate the previous TEST (Blank Check or TESTDEC) is failed.

	0	TSTEND	0	R	R	When 1, indicate the previous TEST (Blank Check or TESTDEC) is finished.

6.3.2.3 Memory Control Register – MEMCON

The MEMCON register is used to control the number of OTP memory which can be mapped into the data space of MCU.

Address	Bit	Name	Reset value	RW	Description
0xA7	7:2	-	--	--	Not used
	1:0	mc	2'b00	RW	0: no OTP is mapped in data space. 1: the upper 512 bytes ofOTP are mapped to data space. 2: the upper 1K bytes ofOTP are mapped to data space. 3: the upper 2K bytes ofOTP are mapped to data space.

6.3.3 OTP programming from the MCU

This section describes how you can write the OTP memory by using the MCU.

The clock frequency of the microcontroller must be 16 MHz during OTP write operations, that bit[7] in FSR register must be 1.

To allow write OTP operations the MCU must run the following sequence:

1. Set WEN (bit 5) in the FSR high to enable OTP writing access. The OTP is now open for writing from the MCU until WEN in FSR is set low again.
2. If write to code space is intended, set PMW (bit 4) in the PCON register high to enable program memory write mode.
3. Programming the OTP is done through normal memory write operations from the MCU. Bytes are written individually (there is no auto increment) to the OTP using the specific memory address.

When the programming code executes from the OTP, write operation is self-timed and the CPU stops until the operation is finished. If the programming code executes from the XDATA RAM the code must wait until the operation has finished. Do not set WEN low before the write operation is finished. Memory address is identical to the OTP address, see chapter 5 for memory mapping.

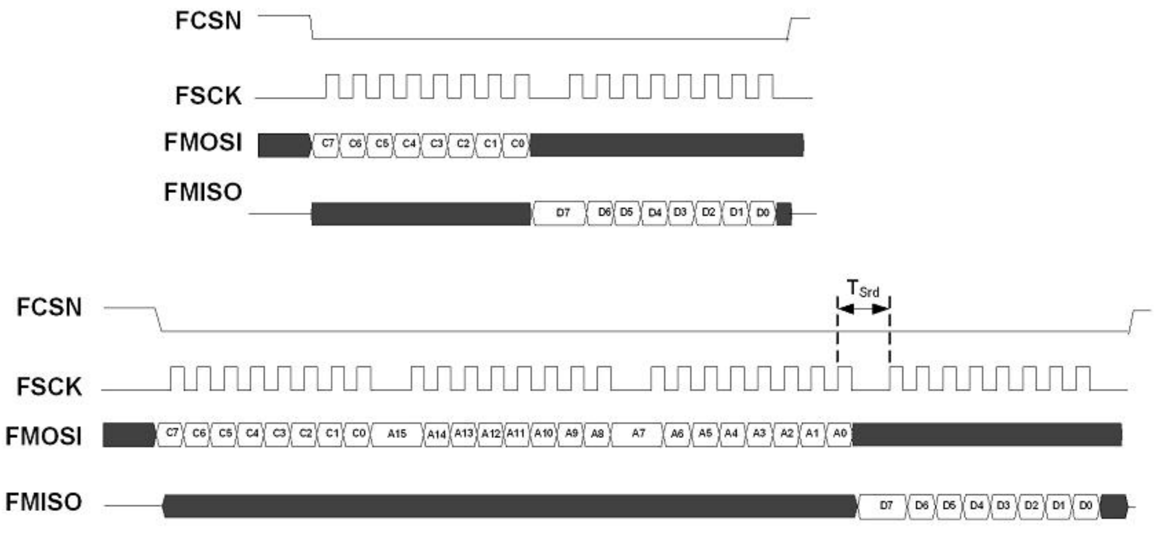
6.3.4 OTP programming through SPI

The on-chip OTP is designed to interface a standard SPI device for programming. The interface uses an 8-bit command register and a set of commands to program and configure the OTP memory. Before write and read OTP, you should check if the RDYN bit of the register FSR is low. If the RDYN bit of the register FSR isn't low, wait until it becomes low.

To program the memory the SPI slave interface is used. SPI slave connection to the OTP memory is activated by sending pattern 0xACA1 trough P0.0 and P0.1 in the first 20ms after the chip is power on.

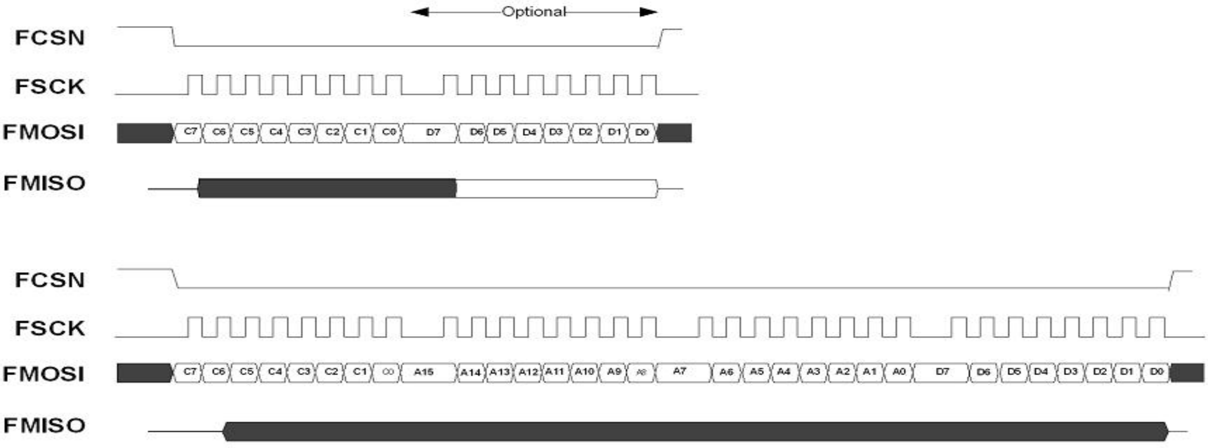
Command	Format	Address	Data	Command operation
WREN	0x06	NA	0	Set OTP write enable latch. Bit WEN in register FSR
WRDIS	0x04	NA	0	Reset OTP write enable latch. Bit WEN in register FSR
RDSR	0x05	NA	1	Read OTP Status Register (FSR)
WRSR	0x01	NA	1	Write OTP Status Register (FSR)
READ	0x03	2 bytes, first OTP address to be read	1–16384	Read data from OTP
PROGRAM	0x02	2 bytes, first OTP address to be written	1-256	Write data to OTP Note: WEN must be set.
RESET	0x80	NA	0	Reset the OTP control state
BLKCHK	0x8a	NA	0	Enable Blank Check test
TESTDEC	0x8b	NA	0	Enable TESTDEC test
SPRON	0x8c	NA	0	Enable spare bit write/read in OTP
SPROFF	0x8d	NA	0	Disable spare bit write/read in OTP

The signaling of the SPI interface is shown:



SPI read operation for direct and addressed command

Note: For the READ command there must be a delay, Tsrd, from the last address bit to the first data bit. Minimum value of Tsrd is 5 clock cycles + 10ns, which is 322.5 ns when system clock is 16MHz.



SPI write operations for direct and addressed commands

WREN / WRDIS OTP write enable/disable

SPI commands WRENandWRDISsets and resets the OTP write enable latch WEN in register FSR. This latch enables all write operations in the OTP blocks. The device will power-up in write disable state, and automatically go back to write disable state after each write SPI command (FCSN set high). Each write command over the SPI interface must therefore be preceded by a WRENcommand.Both WRENand WRDISare 1-byte SPI commands with no data.

RDSR / WRSR read/write OTP status register

SPI commands RDSRand WRSRread and write to the OTP status register FSR. Both commands are 1 and are followed by a data byte for the FSR content.

READ

SPI command READreads out the content of an addressed position in the OTP main block. It must be followed by 2 bytes denoting the start address of the read operation.If the FCSN line is kept active after the first data byte is read out the read command can be extended, the address is auto incremented and data continues to shift out.

A read back of the OTP main block content is only possible if the read disable bit RDISMBin the FSRregister is not set.

PROGRAM

SPI command PROGRAM, programs the content of the addressed position in the OTP main block. It must be followed by 2 bytes denoting the start address of the write operation. Before each write operation the write enable latch WEN must be enabled through the WREN SPI command. It is possible to write up to 256Byte in one PROGRAMcommand. The first byte can be at any address.

6.4 Production Test

In the following sections, the test patterns for the OTP Memory are described. The tests can be performed during wafer sort or final test or in-system/in-field depending on the test flow being used by the end user. Below each test is described.

6.4.1 Blank Check (Defective Bit Screen Test)

Blank Check enables an end user to verify that OTP memory is preloaded with “0” before actual programming. Blank Check must be performed before any other tests or any programming is performed on the OTP memory. Blank check must only be run once on OTP memory.

OTP memory technology depends on hard breakdown of gate oxide in standard logic CMOS process. It is important that integrity of gate oxide be checked before any tests or any programming is performed on OTP memory. This is done by reading every bit location at nominal VDD, nominal VDDIO and elevated VPP which ensures that defective bits caused due to defective gate oxides are screened out. If any of the bits do not read a “0”, the chip is screened out.

To do the blank check test:

- Write BLKCHK command through SPI. Then the OTP controller block will do the blank check automatically.
- Send RDSR command to read FSR register, if Bit 0 of it is 1 which indicates the test is finished. Then check the value of Bit 1. If it is 0, the blank test is passed, otherwise it is failed.
- Write RESET command to reset SPI controller.

6.4.2 TESTDEC (Word-line and Bit-line Integrity Test)

TESTDEC enables an end user to verify the integrity of word-lines and bit-lines as well as screen out the gross defects in the peripheral logic. It is performed on an un-programmed unit and is valid only for un-programmed units.

The expected test pattern is a variation of a checkerboard pattern. If the expected test pattern is not observed, the chip is screened out. The expected test patterns are shown in Table below.

Density	Program width	Read width	TESTDEC pattern		
16Kb	1	8		A<3>=0(even)	A<3>=1(odd)
			A<9>=0(even)	8’h00	8’hFF
			A<9>=1(odd)	8’hFF	8’h00
16Kb	1	1		A<3>=0(even)	A<3>=1(odd)
			A<9>=0(even)	1’b0	1’b1
			A<9>=1(odd)	1’b1	1’b0

To do the TESTDEC test:

- Write TESTDEC command through SPI. Then the OTP controller block will do the TESTDEC test automatically.
- Send RDSR command to read FSR register, if Bit 0 of it is 1 which indicates the test is finished. Then check the value of Bit 1. If it is 0, the TESTDEC test is passed, otherwise it is failed.
- Write RESET command to reset SPI controller.

6.4.3 WRTEST (Pre-program Test)

WRTEST enables an end user to screen out gross defects in programming circuitry before programming of the actual OTP memory array is done. This is enabled by the availability of two spare rows for programming.

This mode is especially useful if the end user is planning to program the OTP memory in the field or in-system since it gives an indication of program circuitry functionality before field-programming or in-system programming is done. If the user is planning to program the OTP memory array in the factory during wafer sort, then this test mode can be skipped.

Table 5 specifies the valid address ranges for the WRTEST mode during programming and during read for the various XPM memories in the 0.11um process node.

Program width	Read width	Address range during programming and during read for WRTEST mode
1	8	If A[13]=0, spare row 0 is selected; if A[13]=1, spare row 1 is selected; A[8:3] selects one of 64 8-bit words; A[12:9] and A[2:0] are don't cares.

To do the WRTEST test:

- Write SPRON command through SPI to enable the spare row for writing.
- Write RESET command to reset SPI controller.
- Write WREN command to enable OTP writing.
- Write 64 byte to OTP start of address 0x0000 in spare row 0, and then wait writing process is finished by reading the OTP RDYN bit of FSR.
- Write another 64 byte to OTP start of address 0x2000 in spare row 1, and then wait writing process is finished by reading the OTP RDYN bit of FSR.
- Read the previous 128 byte data by sending READ command, if the readout values is the same as the data written into, then the WRTEST is passed.
- Write SPROFF command through SPI to disable the spare row.
- Write RESET command to reset SPI controller.

7 Power management

7.1 Mode of operation

The chip can enter standby mode to save power. In this mode， MCU will stop execution and MCU clock will be stopped. This mode can be wakeup by external interrupt, USB and RTC interrupt.

7.2 Functional description

7.2.1 Power down control – PWRDWN

The PWRDWN register is used by the MCU to set the system to a power saving mode:

Address	Bit	Reset value	R/W	Description
0xA4	7: 4			Reserved
	3	1'b0	RW	1:enable USB 0:disable USB(default)
	2:0	3'b0	RW	111: set system to standby (stop MCU clock)

7.2.2 Reset – RSTREAS

Address	Bit	Reset value	R/W	Description
0xB1	7:6		-	Not used
	5		W	Writing 1 to reset RF SPI master module
	4:0	2'b0	R	Not used

7.2.3 Wakeup configuration register – WUCON

The following wakeup sources are available in STANDBY power down mode.

Address	Bit	Reset value	R/W	Description
0xA5	7:6	2'b0	RW	00: Enable wakeup on RFIRQ if interrupt is enabled (IEN1.1=1) 01: Reserved, not used 10: Enable wakeup on RFIRQ 11: Ignore RFIRQ

	5:4	2'b0	RW	00: Enable wakeup on WU, if IEN1[5]=1 01: Reserved, not used 10: Enable wakeup on WU, regardless of IEN1[5] 11: Ignore WU
	3:2	2'b0	RW	00: Enable wakeup on USBIRQ, if IEN1[4]=1 01: Reserved, not used 10: Enable wakeup on USBIRQ, regardless of EN1 [4] 11: Ignore USBIRQ
	1:0	2'b0	RW	00: Enable wakeup on USBWU, if IEN1[3]=1 01: Reserved, not used 10: Enable wakeup on USBWU, regardless of IEN1[3] 11: Ignore USBWU

8 Clock management

8.1 Clock control

The source and frequency of the clock to the microcontroller system is controlled by the CLKCTRL register. Because the read data delay of OTP memory is maximum 70ns, when the microcontroller system clock is 16MHz, the program memory wait cycle defined in register CKCON bit [6:4] should be at least 1, otherwise the system will be failed.

Address	Bit	Reset value	R/W	Description
0xA3	7:5	3'b0	RW	Clock frequency to debounce block: 000: 16 MHz 001: 8 MHz 010: 4 MHz 011: 2 MHz 100: 1 MHz 101: 500 kHz 110: 250 kHz 111: 125 kHz
	4		R	When 1 indicates the XOSC 16M clock is enable
	3			Not used
	2:0	3'b0	RW	Clock frequency to microcontroller system: 000: 16 MHz 001: 8 MHz 010: 4 MHz 011: 2 MHz 100: 1 MHz 101: 500 kHz 110: 250 kHz 111: 125 kHz

Each module can be clock gated by CLKGATE register

Address	Bit	Reset value	R/W	Description
0xAC	7:6	1'b0	RW	Not used
	5	1'b0	RW	1 enable debounce block
	4:3	2'b11	RW	Not used
	2	1'b1	RW	Clock enable for RF SPI

	0	1’b1	RW	Clock enable for OTP controller, program still can be read by MCU even when OTP clock is gated

8.2 32K clock

The frequency of the 32 kHz clock is controlled by the CLKLFCTRL register.

Address	Bit	Reset value	R/W	Description
0xAD	7		R	Read CLKLF (phase).
	6:5	2’b00	RW	Source for CLKLF: 00: 16MHz 01:32K generated from XOSC16M when active, off otherwise[a] 10:1M generated from XOSC16M when active, off otherwise[a] 11:4M generated from XOSC16M when active, off otherwise[a]
	4:3	0		Null
	1	1’b1	RW	Internal 32 kHz clock gating is enable when 1.
	0	1’b0		Null

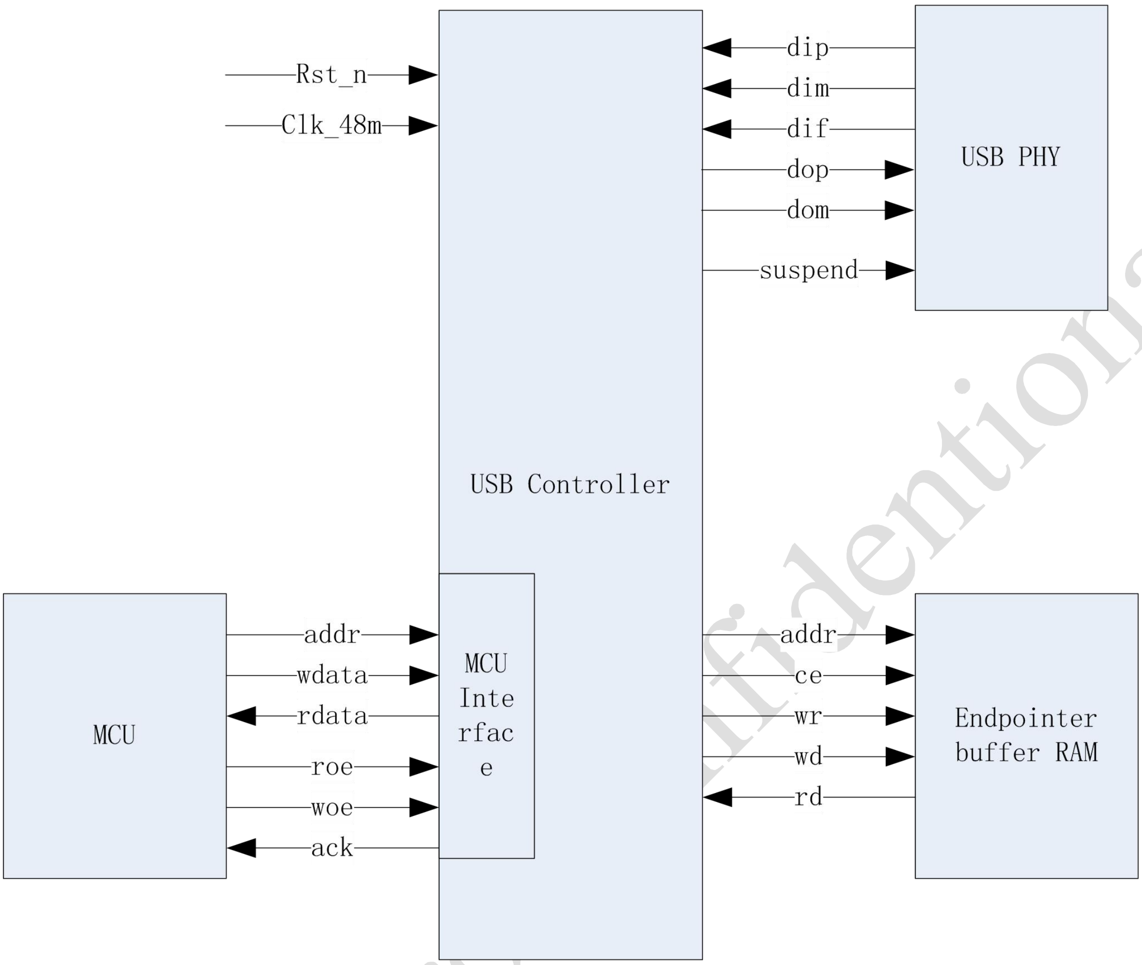
9 USB Configuration

The USB device controller is compatible with USB 2.0 Full-speed(12 Mbps) function.

9.1 Feature

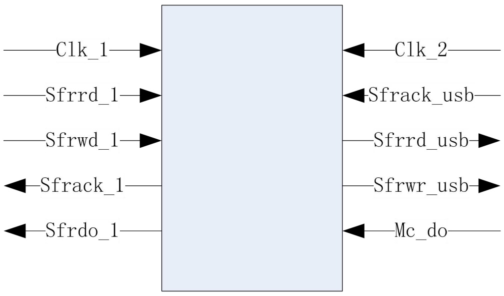
- Conforms to 1.1 and 2.0 revision of the USB specification.
- Support 8 endpoints (include endpoint 0).
- Each endpoint have 8 bytes IN buffer and 8bytes OUT buffer
- Serial Interface Engine
  - Supports full speed devices
  - NRZI decoding/encoding
  - Bit stuffing/stripping
  - CRC checking/generation
  - On-chip pull-up resistorwith software controlled disconnect
- Automatic data retry mechanism
- Data toggle synchronization mechanism
- Suspend and resume power management functions
- Remote Wakeup function

9.2 Block diagram



9.3 MCU interface

The USB controller host interface is work at 12M. The MCU clock is asynchronous to the USB host interface so there is a MCU interface to transfer the MCU cycle to the USB.



9.4 Wakeup/suspend control

If the USB has not find activity on the USB for 3ms, it will generate a Suspend interrupt (if enabled).It is up to the software to decide what, if anything, to disable when the USB is in Suspend mode. The USB 48M PLL will stop. When USB detect Resume signaling (by monitoring the DIM andDIP signals) it will enable the 48M PLL and generate Resume interrupt. The en_48M will manual turn the 48M PLL on.

If the USB is in Suspend mode and the software wants to initiate a remote wakeup, it should write to the sw_wu to enable the 48M PLL. It will take 100us to let the PLL stable then software need to set the Resume bit (D2) in Power register to 1. The software should leave this bit set for approximately 10ms (minimum of 2ms, a maximum of 15ms) then reset it to 0. By thistime the hub should have taken over driving Resume signaling on the USB. The sw_wu must be clear before USB goes into the suspend mode otherwise the 48M PLL will not stop.

Note: No Resume interrupt will be generated when the software initiates a remote wakeup.

9.5 SFR register

9.5.1 USB Control register-USBCON

USBCON(RW) 0xCE

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit0
Sw_rst	Sw_wu	Pll_48m_rstn	Sw_mux	En_48m	Pll_48m_icp		
0	0	1	0	0	3'b011		

Description of Word

Bit	Value	Symbol	Description
7	0	Sw_rst	Software reset 1: assert reset 0: de-assert reset
6	0	Sw_wu	1: wakeup USB must be cleared before setting USB suspend in Enable Suspend in POWER (0xc001).
5	1	Pll_48m_rstn	Output to analogue
4	0	Sw_mux	USB PIN debug select 1: DP, DM uses P1.5 and P1.4 0: DP, DM use normal PIN
3	0	En_48m	Manual enable the 48m clock 1: enable the 48m clock 0: FSM control the 48m clock
2:0	3'h3	Pll_48m_icp	Output to analogue

9.5.2 USBPCON registers —USBPCON

USBPCON (RW) 0xCF

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit0
Not used						Usb_suspend	usb_ndie
----						1’b0	1’b1
----						R	RW

Description of Word

Bit	Value	Symbol	Description
7:2	--	---	Not used
1	1’b0	usb_suspend	usb suspend 1: usb in suspend mode 0: usb not in suspend mode
0	1’b1	usb_ndie	1: usb input disable 0: usb input enable

9.5.3 USBPCON registers —USBPCON1

USBPCON1 (RW) 0xDB

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit0
usb_drv		usb_p_pull_down	usb_p_pull_up	usb_p_pull_up_lo	usb_m_pull_down	usb_m_pull_up	usb__pull_up_lo
2’b10		1’b0	1’b0	1’b0	1’b0	1’b0	1’b0
RW		RW	RW	RW	RW	RW	RW

Description of Word

Bit	Value	Symbol	Description
7:6	2’b10	usb_drv	usb output driver strength control
5	1’b0	usb_p_pull_down	1: usb D+ pull down enable 0: usb D+ pull down disable
4	1’b0	usb_p_pull_up	1: usb D+ pull up enable 0: usb D+ pull up disable
3	1’b0	usb_p_pull_up_lo	1: usb D+ low resistor pull up enable 0: usb D+ low resistor pull up disable
2	1’b0	usb_m_pull_down	1: usb D- pull down enable 0: usb D- pull downdisable

1	1'b0	usb_m_pull_up	1: usb D- pull up enable 0: usb D- pull up disable
0	1'b0	usb_pull_up_lo	1: usb D- low resistor pull up enable 0: usb D- low resistor pull up disable

9.6 USB Registers

Note: In the following bit descriptions:

- 'R' means that the bit is read only
- 'RW' means that the bit can be both read and written
- 'Set' means that the bit can only be written to set it
- 'R/Set' means that the bit can be read or set but it can't be cleared
- 'Clear' means that the bit can only be written to clear it
- 'R/Clear' means that the bit can be read or cleared but it can't be set
- 'Self-clearing' means the bit will be cleared automatically when the associated action has been executed.

9.6.1 Function address register– FADDR

FAddr is an 8-bit register that should be written with the function's 7-bit address. It is then used for decoding the function addressin subsequent token packets.

FADDR (RW) 0xC000

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit0
Update	Function Address						
1'b0	7'b0000000						
R	RW						

Description of Word

Bit	Value	Symbol	Description
7	1'b0	Update	Set when FAddr is written. Cleared when the new address takes effect (at the end of the currenttransfer).
6:0	7'h0	FAddr	The function address.

9.6.2 Power management register-POWER

POWER (RW) 0xC001

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit0
ISO Update	---			Reset	Resume	SuspendMode	EnableSuspend

1’b0	3’b000	1’b0	1’b0	1’b0	1’b0
RW	R	R	RW	R	RW

Description of Word

Bit	Value	Symbol	Description
7	1’b0	ISO Update	When set by the CPU the MUSBFSFC will wait for an SOF token from the time InPktRdy is setbefore sending the packet. If an IN token is received before an SOF token, then a zero length datapacket will be sent. This bit is only used by endpoints performing Isochronous transfers.
6:4	3’b000	---	Unused, always return zero.
3	1’b0	Reset	This read only bit is set while Reset signaling is present on the bus.
2	1’b0	Resume	Set by the CPU to generate Resume signaling when the function is in Suspend mode. The CPUshould clear this bit after 10ms (a maximum of 15ms) to end Resume signaling.
1	1’b0	SuspendMode	Set by the USB when Suspend mode is entered. Cleared when the CPU reads the interrupt register,or sets the Resume bit of this register.
0	1’b0	EnableSuspend	Set by the CPU to enable entry into Suspend mode when Suspend signaling is received on the bus.

9.6.3 Interrupt register for Endpoint 0 plus IN Endpoints 1 to 7-INTRIN1

IntrIn1 is an 8-bit read-only register that indicates which of the interrupts for IN Endpoints 1–7 are currently active. It alsoindicates whether the Endpoint 0 interrupt is currently active. Note: All active interrupts will be cleared when this register is read.

INTRIN1 (R) 0xC002

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit0
EP7	EP6	EP5	EP4	EP3	EP2	EP1	EP0
1’b0	1’b0	1’b0	1’b0	1’b0	1’b0	1’b0	1’b0
R	R	R	R	R	R	R	R

Description of Word

Bit	Value	Symbol	Description
7	1’b0	EP7	IN Endpoint 7 interrupt.
6	1’b0	EP6	IN Endpoint 6 interrupt.
5	1’b0	EP5	IN Endpoint 5 interrupt.
4	1’b0	EP4	IN Endpoint 4 interrupt.
3	1’b0	EP3	IN Endpoint 3 interrupt.
2	1’b0	EP2	IN Endpoint 2 interrupt.
1	1’b0	EP1	IN Endpoint 1 interrupt.
0	1’b0	EP0	IN Endpoint 0 interrupt.

9.6.4 Interrupt register for IN Endpoints 8 to 15-INTRIN2

IntrIn2 is an 8-bit read-only register that indicates which of the interrupts for IN Endpoints 8–15 are currently active. Thisregister will only be present if the MUSBFSFC is configured for more than 7 IN endpoints.Note: All active interrupts will be cleared when this register is read.

INTRIN2 (R) 0xC003

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit0
EP15	EP14	EP13	EP12	EP11	EP10	EP9	EP8
1'b0	1'b0	1'b0	1'b0	1'b0	1'b0	1'b0	1'b0
R	R	R	R	R	R	R	R

Description of Word

Bit	Value	Symbol	Description
7	1'b0	EP15	IN Endpoint 14 interrupt.
6	1'b0	EP14	IN Endpoint 13 interrupt.
5	1'b0	EP13	IN Endpoint 13 interrupt.
4	1'b0	EP12	IN Endpoint 12 interrupt.
3	1'b0	EP11	IN Endpoint 11 interrupt.
2	1'b0	EP10	IN Endpoint 10 interrupt.
1	1'b0	EP9	IN Endpoint 9 interrupt.
0	1'b0	EP8	IN Endpoint 8 interrupt.

9.6.5 Interrupt register for OUT Endpoints 1 to 7-INTROUT1

IntrOut1 is an 8-bit read-only register that indicates which of the interrupts for OUT Endpoints 1–7 are currently active.Note:All active interrupts will be cleared when this register is read.

INTROUT1 (R) 0xC004

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit0
EP7	EP6	EP5	EP4	EP3	EP2	EP1	EP0
1'b0	1'b0	1'b0	1'b0	1'b0	1'b0	1'b0	1'b0
R	R	R	R	R	R	R	R

Description of Word

Bit	Value	Symbol	Description
7	1'b0	EP7	OUT Endpoint 7 interrupt.

6	1’b0	EP6	OUT Endpoint 6 interrupt.
5	1’b0	EP5	OUT Endpoint 5 interrupt.
4	1’b0	EP4	OUT Endpoint 4 interrupt.
3	1’b0	EP3	OUT Endpoint 3 interrupt.
2	1’b0	EP2	OUT Endpoint 2 interrupt.
1	1’b0	EP1	OUT Endpoint 1 interrupt.
0	1’b0	--	Unused, always returns 0

9.6.6 Interrupt register for OUT Endpoints 8 to 15-INTROUT2

IntrOut2 is an 8-bit read-only register that indicates which of the interrupts for OUT Endpoints 8–15 are currently active.Thisregister will only be present if the MUSBFSFC is configured for more than 7 OUT endpoints. Note:All active interrupts will be cleared when this register is read.

INTROUT2 (R) 0xC005

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit0
EP15	EP14	EP13	EP12	EP11	EP10	EP9	EP8
1’b0	1’b0	1’b0	1’b0	1’b0	1’b0	1’b0	1’b0
R	R	R	R	R	R	R	R

Description of Word

Bit	Value	Symbol	Description
7	1’b0	EP15	OUT Endpoint 15 interrupt.
6	1’b0	EP14	OUT Endpoint 14 interrupt.
5	1’b0	EP13	OUT Endpoint 13 interrupt.
4	1’b0	EP12	OUT Endpoint 12 interrupt.
3	1’b0	EP11	OUT Endpoint 11 interrupt.
2	1’b0	EP10	OUT Endpoint 10 interrupt.
1	1’b0	EP9	OUT Endpoint 9 interrupt.
0	1’b0	EP8	OUT Endpoint 8 interrupt.

9.6.7 Interrupt register for common USB interrupts-INTRUSB

INTRUSB is an 8-bit read-only register that indicates which USB interrupts are currently active. All active interrupts will be clearedwhen this register is read.

INTRUSB (R) 0xC006

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit0

---	SOF	Reset	Resume	Suspend
4’b0000	1’b0	1’b0	1’b0	1’b0
R	R	R	R	R

Description of Word

Bit	Value	Symbol	Description
7:4	4’b0000	---	Unused, always return 0.
3	1’b0	SOF	Set at the start of each frame.
2	1’b0	Reset	Set when Reset signaling is detected on the bus.
1	1’b0	Resume	Set when Resume signaling is detected on the bus while the MUSBFSFC is in Suspend mode.
0	1’b0	Suspend	Set when Suspend signaling is detected on the bus.

9.6.8 Interrupt enable register for INTRIN1-INTRIN1E

IntrIn1E is 8-bit registers that provide interrupt enable bits for the interrupts in IntrIn1.On reset, the bits corresponding to Endpoint 0 and the IN endpoints included in the design are set to 1, while the remaining bitsare set to 0.

INTRIN1E (RW) 0xC007

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit0
EP7	EP6	EP5	EP4	EP3	EP2	EP1	EP0
1’b0	1’b0	1’b0	1’b0	1’b1	1’b1	1’b1	1’b1
RW	RW	RW	RW	RW	RW	RW	RW

Description of Word

Bit	Value	Symbol	Description
7	1’b0	EP7	enable bit for the IN Endpoint 7 interrupt 1:enable 0:disable
6	1’b0	EP6	enable bit for the IN Endpoint 6 interrupt 1:enable 0:disable
5	1’b0	EP5	enable bit for the IN Endpoint 5 interrupt 1:enable 0:disable
4	1’b0	EP4	enable bit for the IN Endpoint 4 interrupt 1:enable 0:disable

3	1'b1	EP3	enable bit for the IN Endpoint 3 interrupt 1:enable 0:disable
2	1'b1	EP2	enable bit for the IN Endpoint 2 interrupt 1:enable 0:disable
1	1'b1	EP1	enable bit for the IN Endpoint 1 interrupt 1:enable 0:disable
0	1'b1	EP0	enable bit for the IN Endpoint 0 interrupt 1:enable 0:disable

9.6.9 Interrupt enable register for INTRIN1-INTRIN2E

IntrIn2E is 8-bit registers that provide interrupt enable bits for the interrupts in IntrIn2.On reset, the bits corresponding to Endpoint 0 and the IN endpoints included in the design are set to 1, while the remaining bitsare set to 0.

INTRIN2E (RW) 0xC008

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit0
EP15	EP14	EP13	EP12	EP11	EP10	EP8	EP8
1'b0	1'b0	1'b0	1'b0	1'b0	1'b0	1'b0	1'b0
RW	RW	RW	RW	RW	RW	RW	RW

Description of Word

Bit	Value	Symbol	Description
7	1'b0	EP15	enable bit for the IN Endpoint 15 interrupt 1:enable 0:disable
6	1'b0	EP14	enable bit for the IN Endpoint 14 interrupt 1:enable 0:disable
5	1'b0	EP13	enable bit for the IN Endpoint 13 interrupt 1:enable 0:disable
4	1'b0	EP12	enable bit for the IN Endpoint 12 interrupt 1:enable 0:disable

3	1'b0	EP11	enable bit for the IN Endpoint 11 interrupt 1:enable 0:disable
2	1'b0	EP10	enable bit for the IN Endpoint 10 interrupt 1:enable 0:disable
1	1'b0	EP9	enable bit for the IN Endpoint 9 interrupt 1:enable 0:disable
0	1'b0	EP8	enable bit for the IN Endpoint 8 interrupt 1:enable 0:disable

9.6.10 Interrupt enable register for INTROUT1-INTROUT1E

IntrOut1Eis 8-bit register that provides interrupt enable bits for the interrupts in IntrOut1. On reset, the bits corresponding to the OUT endpoints included in the design are set to 1, while the remaining bitsare set to 0.

INTROUT1E (R) 0xC009

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit0
EP7	EP6	EP5	EP4	EP3	EP2	EP1	---
1'b0	1'b0	1'b0	1'b0	1'b1	1'b1	1'b1	1'b0
RW	RW	RW	RW	RW	RW	RW	R

Description of Word

Bit	Value	Symbol	Description
7	1'b0	EP7	enable bit for the OUT Endpoint 7 interrupt 1:enable 0:disable
6	1'b0	EP6	enable bit for the OUT Endpoint 6 interrupt 1:enable 0:disable
5	1'b0	EP5	enable bit for the OUT Endpoint 5 interrupt 1:enable 0:disable
4	1'b0	EP4	enable bit for the OUT Endpoint 4 interrupt 1:enable 0:disable

3	1'b1	EP3	enable bit for the OUT Endpoint 3 interrupt 1:enable 0:disable
2	1'b1	EP2	enable bit for the OUT Endpoint 2 interrupt 1:enable 0:disable
1	1'b1	EP1	enable bit for the OUT Endpoint 1 interrupt 1:enable 0:disable
0	1'b0	--	Unused, always returns 0

9.6.11 Interrupt enable register for INTROUT2-INTROUT2E

IntrOut2Eis 8-bit register that provides interrupt enable bits for the interrupts in IntrOut2. On reset, the bits corresponding to the OUT endpoints included in the design are set to 1, while the remaining bitsare set to 0.

INTROUT2E (RW) 0xC00A

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit0
EP15	EP14	EP13	EP12	EP11	EP10	EP9	EP8
1'b0	1'b0	1'b0	1'b0	1'b0	1'b0	1'b0	1'b0
RW	RW	RW	RW	RW	RW	RW	RW

Description of Word

Bit	Value	Symbol	Description
7	1'b0	EP15	enable bit for the OUT Endpoint 15 interrupt 1:enable 0:disable
6	1'b0	EP14	enable bit for the OUT Endpoint 14 interrupt 1:enable 0:disable
5	1'b0	EP13	enable bit for the OUT Endpoint 13 interrupt 1:enable 0:disable
4	1'b0	EP12	enable bit for the OUT Endpoint 12 interrupt 1:enable 0:disable
3	1'b0	EP11	enable bit for the OUT Endpoint 11 interrupt 1:enable 0:disable

2	1’b0	EP10	enable bit for the OUT Endpoint 10 interrupt 1:enable 0:disable
1	1’b0	EP9	enable bit for the OUT Endpoint 9 interrupt 1:enable 0:disable
0	1’b0	EP8	enable bit for the OUT Endpoint 8 interrupt 1:enable 0:disable

9.6.12 Interrupt enable register for INTRUSB-INTRUSBE

INTRUSBE is an 8-bit register that provides interrupt enable bits for each of the interrupts in INTRUSB.

INTRUSBE (R) 0xC00B

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit0
---				SOF	Reset	Resume	Suspend
4’b0000				1’b0	1’b1	1’b1	1’b0
R				RW	RW	RW	RW

Description of Word

Bit	Value	Symbol	Description
7:4	4’b0000	---	Unused, always return 0.
3	1’b0	SOF	enable bit for the interrupt 1:enable 0:disable
2	1’b1	Reset	enable bit for the interrupt 1:enable 0:disable
1	1’b1	Resume	enable bit for the interrupt 1:enable 0:disable
0	1’b0	Suspend	enable bit for the interrupt 1:enable 0:disable

9.6.13 Frame number bits 0 to 7-FRAME1

Frame1 is an 8-bit read-only register that holds the lower 8 bits of the last received frame number.

FRAME1 (R) 0xC00C

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit0
Frame1							
8’h00							
R							

Description of Word

Bit	Value	Symbol	Description
7:0	8’h00	Frame1	Lower 8 bits of Frame Number.

9.6.14 Frame number bits 8 to 10-FRAME2

Frame2 is a 3-bit read-only register that holds the upper 3 bits of the last received frame number.

FRAME2 (R) 0xC00D

Bit 2	Bit 1	Bit 0
Frame2		
3’b000		
R		

Description of Word

Bit	Value	Symbol	Description
2:0	3’b000	Frame2	Upper 3 bits of Frame Number.

9.6.15 Index register for selecting the endpoint status and control registers-INDEX

Index is a 4-bit register that determines which endpoint control/status registers are accessed at addresses 0xC010h to 0xC017h.Each IN endpoint and each OUT endpoint have their own set of control/status registers. Only one set of IN control/status andone set of OUT control/status registers appear in the memory map at any one time. Before accessing an endpoint’scontrol/status registers, the endpoint number should be written to the Index register to ensure that the correct control/statusregisters appear in the memory map.

INDEX (RW) 0xC00E

Bit 3	Bit 2	Bit 1	Bit 0
Index			
4’b0000			
RW			

Description of Word

Bit	Value	Symbol	Description
3:0	4’b0000	Index	Selected Endpoint.

9.6.16 Control Status register for Endpoint 0-CSR0

CSR0 is an 8-bit register that provides control and status bits for Endpoint 0.CSR0 is used for all control/status of Endpoint0.

CSR0 0xC011

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit0
Serviced SetupEnd (self-clearing)	Serviced OutPktRdy (self-clearing)	SendStall (self-clearing)	SendStall (self-clearing)	DataEnd (self-clearing)	SentStall	InPktRdy (self-clearing)	OutPktRdy
1’b0	1’b0	1’b0	1’b0	1’b0	1’b0	1’b0	1’b0
Set	Set	Set	R	Set	R/Clear	R/Set	R

Description of Word

Bit	Value	Symbol	Description
7	1’b0	ServicedSetupEnd	The CPU writes a 1 to this bit to clear the SetupEnd bit. It is cleared automatically.
6	1’b0	ServicedOutPktRdy	The CPU writes a 1 to this bit to clear the OutPktRdy bit. It is cleared automatically.
5	1’b0	SendStall	The CPU writes a 1 to this bit to terminate the current transaction. The STALL handshakewill be transmitted and then this bit will be cleared automatically.
4	1’b0	SendStall	This bit will be set when a control transaction ends before the DataEnd bit has been set. An interrupt will be generated and the FIFO flushed at this time. The bit is cleared by the CPU writing a 1 to the ServicedSetupEnd bit.
3	1’b0	DataEnd	The CPU sets this bit: 1. When setting InPktRdy for the last data packet. 2. When clearing OutPktRdy after unloading the last data packet. 3. When setting InPktRdy for a zero length data packet. It is cleared automatically.
2	1’b0	SentStall	This bit is set when a STALL handshake is transmitted. The CPU should clear this bit.
1	1’b0	InPktRdy	The CPU sets this bit after loading a data packet into the FIFO. It is cleared automaticallywhen the data packet has been transmitted. An interrupt is generated when the bit iscleared.
0	1’b0	OutPktRdy	This bit is set when a data packet has been received. An interrupt is generated when this bitis set. The CPU clears this bit by setting the ServicedOutPktRdy bit.

9.6.17 Maximum packet size for IN endpoint-INMAXP

InMaxP is an 8-bit register that holds the maximum packet size for transactions through the currently-selected IN endpoint – inunits of 8 bytes, except that a value of 2 sets the maximum packet size to 16.In setting this value, you should note the constraints placed by the USBSpecification on packet sizes for Bulk, Interrupt and Isochronous transactions in Full-speed operations.
There is an InMaxP register for each IN endpoint (except Endpoint 0).

INMAXP (RW) 0xC010

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Inmaxp							
8'h00							
RW							

Description of Word

Bit	Value	Symbol	Description
7:0	8'h00	Inmaxp	Maximum Packet Size/transaction.

9.6.18 Control Status register 1 for IN endpoint-INCSR1

InCSR1 is an 8-bit register that provides control and status bits for IN transactions through the currently-selected endpoint.There is an InCSR1 register for each IN endpoint (not including Endpoint 0).

INCSR1 0xC011

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit0
---	ClrDataTog	SentStall	SendStall	FlushFIFO (self-clearing)	UnderRun	FIFO NotEmpty	InPktRdy
1'b0	1'b0	1'b0	1'b0	1'b0	1'b0	1'b0	1'b0
R	Set	R/Clear	RW	Set	R/Clear	R/Clear	R/Set

Description of Word

Bit	Value	Symbol	Description
7	1'b0	---	Unused. Returns zero when read.
6	1'b0	ClrDataTog	The CPU writes a 1 to this bit to reset the endpoint IN data toggle to 0.
5	1'b0	SentStall	This bit is set when a STALL handshake is transmitted. The FIFO is flushed and the InPktRdybitis cleared (see below). The CPU should clear this bit.

4	1’b0	SendStall	The CPU writes a 1 to this bit to issue a STALL handshake to an IN token. The CPU clears this bitto terminate the stall condition. This bit has no effect if the IN endpoint is in ISO mode.
3	1’b0	FlushFIFO	The CPU writes a 1 to this bit to flush the next packet to be transmitted from the endpoint INFIFO. The FIFO pointer is reset and the InPktRdy bit (below) is cleared. Note:If the FIFOcontains two packets, FlushFIFO will need to be set twice to completely clear the FIFO.
2	1’b0	UnderRun	In ISO mode, this bit is set when a zero length data packet is sent after receiving an IN token withthe InPktRdy bit not set. In Bulk/Interrupt mode, this bit is set when a NAK is returned inresponse to an IN token. The CPU should clear this bit.
1	1’b0	FIFONotEmpty	This bit is set when there is at least 1 packet in the IN FIFO.
0	1’b0	InPktRdy	The CPU sets this bit after loading a data packet into the FIFO. It is cleared automatically when adata packet has been transmitted. An interrupt is generated (if enabled) when the bit is cleared.

9.6.19 Control Status register 2 for IN endpoint-INCSR2

INCSR2 is an 8-bit register that provides further control bits for IN transactions through the currently-selected endpoint. There isan INCSR2 register for each IN endpoint (not including Endpoint 0).

INCSR2 0xC012

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit0
AutoSet	ISO	Mode	DMAEnab	FrcDataTog	---	---	---
1’b0	1’b0	1’b0	1’b0	1’b0	1’b0	1’b0	1’b0
RW	RW	RW	RW	RW	R	R	R

Description of Word

Bit	Value	Symbol	Description
7	1’b0	AutoSet	If the CPU sets this bit, InPktRdy will be automatically set when data of the maximum packet size(value in InMaxP) is loaded into the IN FIFO. If a packet of less than the maximum packet size isloaded, then InPktRdy will have to be set manually.
6	1’b0	ISO	The CPU sets this bit to enable the IN endpoint for isochronous transfers (ISO mode), and clears itto enable the IN endpoint for bulk/interrupt transfers.
5	1’b0	Mode	The CPU sets this bit to enable the endpoint direction as IN, and clears it to enable the endpointdirection as OUT. Valid only where the same endpoint FIFO is used for both IN and OUT transactions.
4	1’b0	DMAEnab	The CPU sets this bit to enable the DMA request for the IN endpoint.
3	1’b0	FrcDataTog	The CPU sets this bit to force theendpoint’s IN data toggle to switch after each data packet is sentregardless of whether an ACK was received. This can be used by interrupt IN endpoints which areused to communicate rate feedback for Isochronous endpoints.

2:0	3'b000	---	Unused, always return 0.

9.6.20 Maximum packet size for OUT endpoint-OUTMAXP

OutMaxP is an 8-bit register that holds the maximum packet size for transactions through the currently-selected OUT endpoint–in units of 8 bytes, except that a value of 2 sets the maximum packet size to 16.In setting this value, you should note the constraints placed by the USB Specification on packet sizes forBulk, Interrupt and Isochronous transactions in Full-speed operations.
There is an OutMaxP register for each OUT endpoint (except Endpoint 0).

OUTMAXP (RW) 0xC013

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Outmaxp							
8'h00							
RW							

Description of Word

Bit	Value	Symbol	Description
7:0	8'h00	Outmaxp	Maximum Packet Size/transaction.

9.6.21 Control Status register 1 for OUT endpoint-OUTCSR1

OutCSR1 is an 8-bit register that provides control and status bits for OUT transactions through thecurrently-selected endpoint.It is reset to 0.

OUTCSR1 0xC014

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit0
ClrDataTog	SentStall	SendStall	FlushFIFO (self-clearing)	DataError	UnderRun	FIFO Full	OutPktRdy
1'b0	1'b0	1'b0	1'b0	1'b0	1'b0	1'b0	1'b0
Set	R/Clear	RW	Set	R	R/Clear	R	R/Clear

Description of Word

Bit	Value	Symbol	Description
7	1'b0	ClrDataTog	The CPU writes a 1 to this bit to reset the endpoint data toggle to 0.
6	1'b0	SentStall	This bit is set when a STALL handshake is transmitted. The CPU should clear this bit.
5	1'b0	SendStall	The CPU writes a 1 to this bit to issue a STALL handshake. The CPU clears this bit to terminatethe stall condition. This bit has no effect if the OUT endpoint is in ISO mode.

4	1'b0	FlushFIFO	The CPU writes a 1 to this bit to flush the next packet to be read from the endpoint OUT FIFO. Note:If the FIFO contains two packets, FlushFIFO will need to be set twice to completely clear theFIFO.
3	1'b0	DataError	This bit is set when OutPktRdy is set if the data packet has a CRC or bit-stuff error. it is clearedwhen OutPktRdy is cleared. The bit is only valid in ISO mode.
2	1'b0	UnderRun	This bit is set if an OUT packet cannot be loaded into the OUT FIFO. The CPU should clear thisbit. The bit is only valid in ISO mode.
1	1'b0	FIFOfull	This bit is set when no more packets can be loaded into the OUT FIFO.
0	1'b0	OutPktRdy	This bit is set when a data packet has been received. The CPU should clear this bit when the packetas been unloaded from the OUT FIFO. An interrupt is generated when the bit is set.

9.6.22 Control Status register 2 for OUT endpoint-OUTCSR2

OutCSR2 is an 8-bit register that provides further control bits for OUT transactions through the currently-selected endpoint. It isreset to 0.

OUTCSR2 0xC015

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit0
AutoClear	ISO	DMAEnab	DMAMode	---	---	---	---
1'b0	1'b0	1'b0	1'b0	1'b0	1'b0	1'b0	1'b0
RW	RW	RW	RW	RW	R	R	R

Description of Word

Bit	Value	Symbol	Description
7	1'b0	AutoClear	If the CPU sets this bit then the OutPktRdy bit will be automatically cleared when a packet ofOutMaxP bytes has been unloaded from the OUT FIFO. When packets of less than the maximumpacket size are unloaded, OutPktRdy will have to be cleared manually
6	1'b0	ISO	The CPU sets this bit to enable the OUT endpoint for Isochronous transfers, and clears it to enablethe OUT endpoint for Bulk/Interrupt transfers
5	1'b0	DMAEnab	The CPU sets this bit to enable the DMA request for the OUT endpoint.
4	1'b0	DMAMode	Two modes of DMA operation are supported: DMA Mode 0 in which a DMA request is generatedfor all received packets, together with an interrupt (if enabled); and DMA Mode 1 in which a DMArequest (but no interrupt) is generated for OUT packets of sizeOutMaxP bytes and an interrupt(but no DMA request) is generated for OUT packets of any other size. The CPU sets this bit toselect DMA Mode 1 and clears this bit to select DMA Mode 0.
3:0	4'b0000	---	Unused, always return 0.

9.6.23 Number of received bytes in Endpoint 0 FIFO-COUNT0

Count0 is a 7-bit read-only register that indicates the number of received data bytes in the Endpoint 0 FIFO. The value returnedis valid while OutPktRdy (CSR0.bit0) is set.

COUNT0 (R) 0xC016

Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Count0						
7’h00						
R						

Description of Word

Bit	Value	Symbol	Description
6:0	7’h00	Count0	Endpoint 0 OUT Count.

9.6.24 Number of bytes in OUT endpoint FIFO-OUTCOUNT1

OutCount1 is an 8-bit read-only register that holds the lower 8 bits of the number of received data bytes in the packet in the FIFOassociated with the currently-selected OUT endpoint. The value returned is valid while OutPktRdy (OutCSR1.bit0) is set.

OUTCOUNTT1 (R) 0xC016

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Outcounter1							
8’h00							
R							

Description of Word

Bit	Value	Symbol	Description
7:0	8’h00	Outcounter1	Endpoint OUT Count – lower 8 bits

9.6.25 Number of bytes in OUT endpoint FIFO-OUTCOUNT2

OutCount2 is a 3-bit read-only register that holds the upper 3 bits of the number of received data bytes in the packet in the FIFO associated with the currently-selected OUT endpoint. The value returned is valid while OutPktRdy (OutCSR1.bit0) is set.

OUTCOUNTER2 (R) 0xC017

Bit 2	Bit 1	Bit 0
Outcounter2		
3’b000		
R		

Description of Word

Bit	Value	Symbol	Description
2:0	3’b000	Outcounter2	Endpoint OUT Count – upper 3 bits.

9.6.26 FIFOs for Endpoints 0 to 3-FIFOx

(Addresses0xC020–0xC023)

This address range provides 4 addresses for CPU access to the FIFOs for each endpoint. Writing to these addresses loads datainto the IN FIFO for the corresponding endpoint. Reading from these addresses unloads data from the OUT FIFO for thecorresponding endpoint.

The FIFOs are located on byte boundaries (Endpoint 0 at 0xC020, Endpoint 1 at 0xC021 ... Endpoint 3 at 0xC023).

e = ‘1’. One byte transmission

10 Timer 0 and Timer 1

The HS6209 contain two 16-bit timer/counters, Timer 0 and Timer 1 which compatible to standard 8051. Inthe “timer mode”, timer registersare incremented everysystem 1/12 clock period, whenappropriate timer is enabled. Inthe “countermode” the timerregisters are incremented every falling transition on theirscorresponding input pins: T0 or T1. Theinput pins are sampled every system clockperiod.

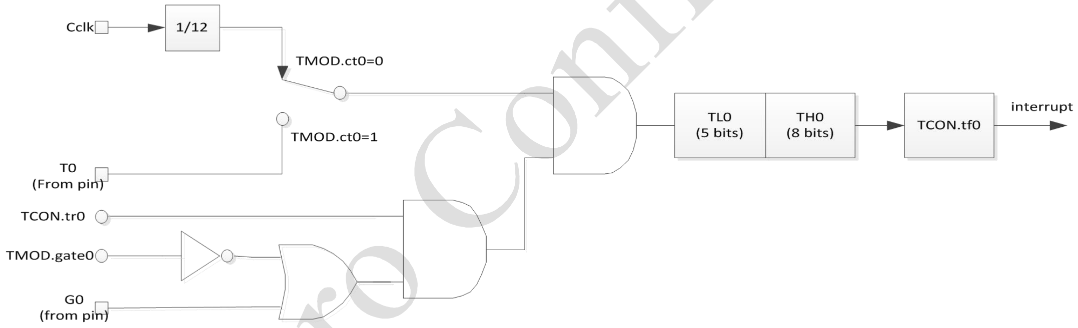
10.1 Timer 0

10.1.1 Mode 0

In this mode, the timer register is configured as a 13-bit register. As the count rolls over from all 1’sto all 0’s, it sets the timer overflow flag TCON.tf0. The overflow flag TCON.tf0 then can be used to request aninterrupt.

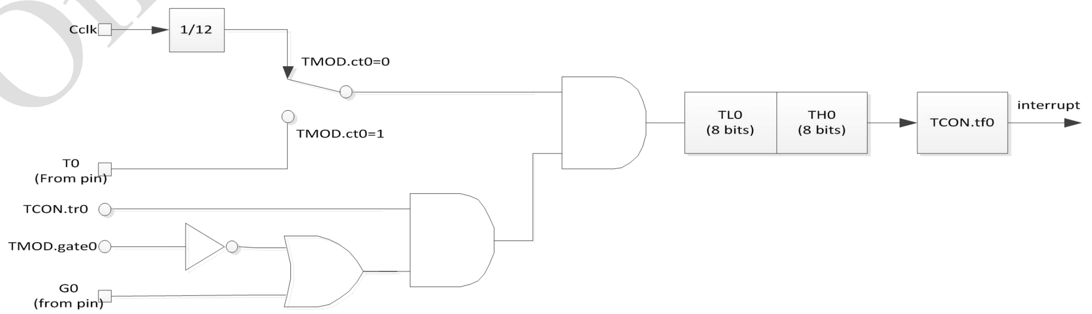
The counted input is enabled to the timer when TCON.tr0=1 and either TMOD.gate0=0 or G0=1 (setting TMOD.gate0=1 allows the timer 0 to be controlled by external input G0, to facilitate pulse widthmeasurements).

The 13-bit register consists of all 8 bits of TH0 and the lower 5 bits of TL0. The upper 3 bits of TL0are indeterminate and should be ignored. Setting the run flag (TCON.tr0) does not clear the registers.



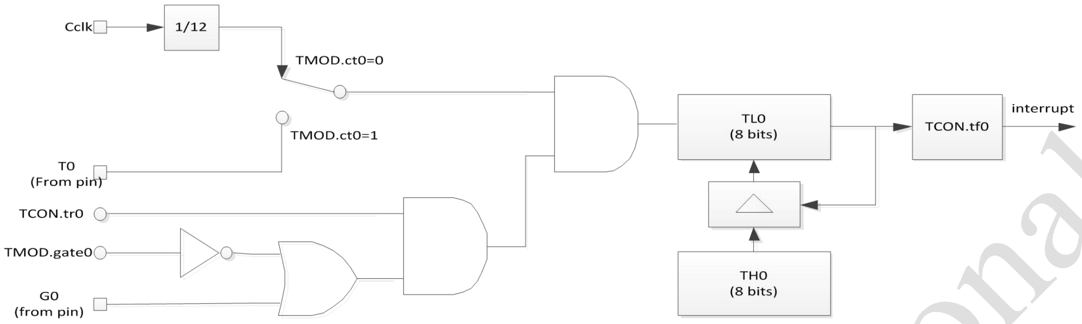
10.1.2 Mode 1

Mode 1 is the same as mode 0, except that the timer register is run with all 16 bits. Mode 1 is shownin figure below.



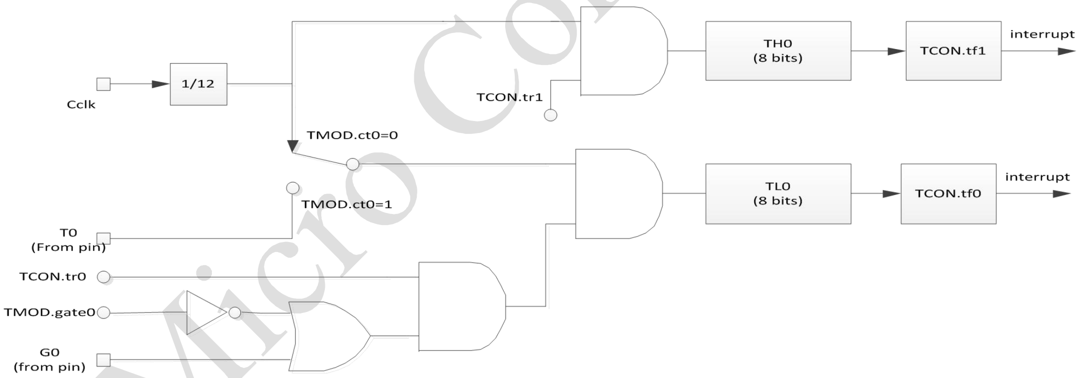
10.1.3 Mode 2

Mode 2 configures the timerregister as an 8-bit counter (TL0) with automatic reloads as shown in figure below. Overflow from TL0 not only sets TCON.tf0, but also reloads TL0 with the contents of TH0,which is preset bysoftware. The reload leaves TH0 unchanged.



10.1.4 Mode 3

Timer 0 inMode 3 establishes TL0 and TH0 as two separatecounters. The logic forMode 3 on Timer 0 isshownin figure below. TL0uses the Timer 0control bits: TCON.tr0, TMODE.gate0, TCON.ct0, TCON.tf1 and G0. TH0 is locked into a timer function(counting machine cycles) and uses the TCON.tr1 and TCON.tf1 flagsfrom Timer 1 and controls Timer1 interrupt. Mode3is provided for applications requiring an extra 8-bit timer/counter. When Timer 0 is in Mode 3, Timer 1 can be turned off by switching it into its ownMode 3or can still be used by the serialchannel asa baud rate generator or in any application where interrupt from Timer1 is not required.



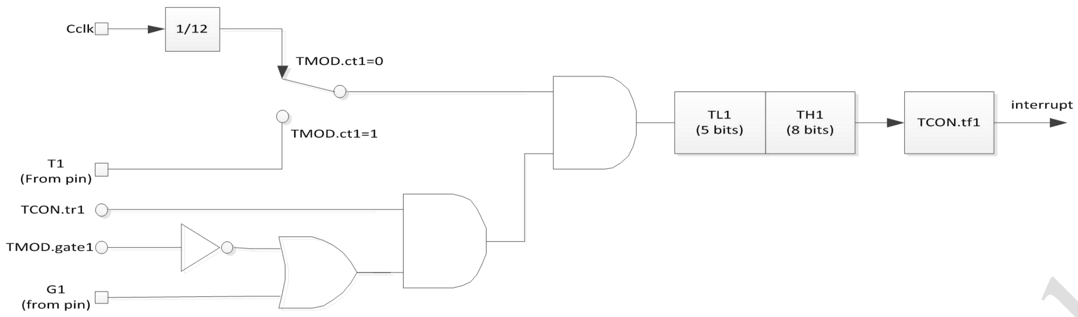
10.2 Timer 1

10.2.1 Mode0

In this mode, the timer register is configured as a 13-bit register. As the count rolls over from all 1'sto all 0's, it sets the timer overflow flag TCON.tf1. The overflow flag TCON.tf1 then can be used to request aninterrupt.

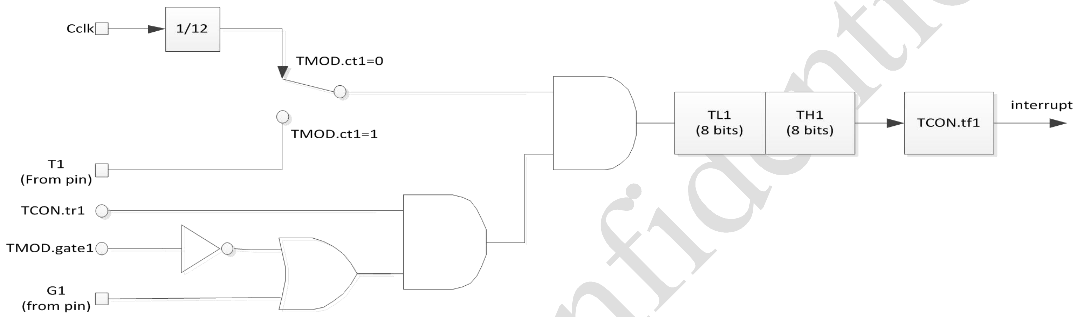
The counted input is enabled to the timer when TCON.tr1=1 and either TMOD.gate1=0 or G1=1 (setting TMOD.gate1=1 allows the timer 0 to be controlled by external input G1, to facilitate pulse widthmeasurements).

The 13-bit register consists of all 8 bits of TH1 and the lower 5 bits of TL1. The upper 3 bits of TL1 are indeterminate and should be ignored. Setting the run flag (TCON.tr1) does not clear the registers.



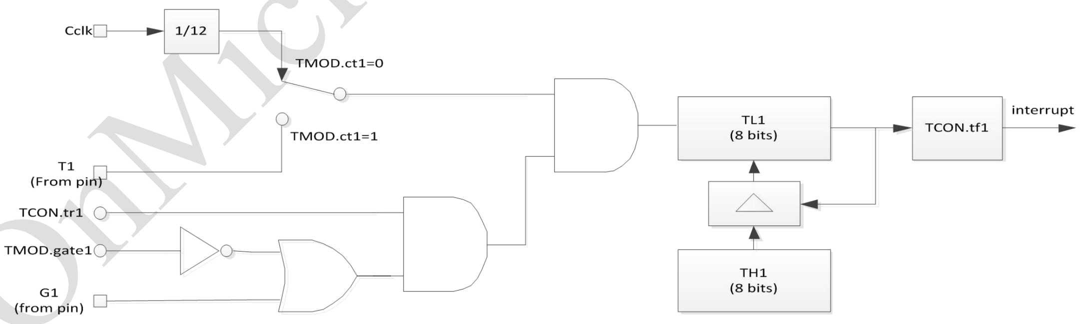
10.2.2 Mode1

Mode 1 is the same as mode 0, except that the timer register is run with all 16 bits. Mode 1 is shownin figure below.



10.2.3 Mode2

Mode 2 configures the timerregister as an 8-bit counter (TL1) with automatic reloads as shown in figure below. Overflow from TL1 not only sets TCON.tf1, but also reloads TL1 with the contents of TH1,which is preset by software. The reload leaves TH1 unchanged.



10.2.4 Mode3

Timer 1 in mode 3 simply holds its count. The effectis the same as setting TCON.tr1=0.

10.3 SFR registers

10.3.1 Timer/Counter control register – TCON

TCONregister reflects the current status of MCU Timer 0 and Timer 1 and it is used to control the operation of these modules.

TCON(RW) 0x88

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit0
tf1	tr1	tf0	tr0	ie1	it1	ie0	it0
1'b0	1'b0	1'b0	1'b0	1'b0	1'b0	1'b0	1'b0
RW	RW	RW	RW	RW	RW	RW	RW

Description of Word

Bit	Value	Symbol	Description
7	1'b0	tf1	Timer 1 overflow flag. Set by hardware when Timer1 overflows.Cleared by hardware when interrupt is processed.
6	1'b0	tr1	Timer 1 Run control. If cleared, Timer 1 stops.
5	1'b0	tf0	Timer 1 overflow flag. Set by hardware when Timer 0 overflows.Cleared by hardware when interrupt is processed.
4	1'b0	tr0	Timer 0Run control. If cleared, Timer 0 stops.
3	1'b0	ie1	External interrupt 1 flag. Set by hardware, when external interrupt int1 (edge/level, depending on settings) is observed. Cleared by hardware when interrupt is processed.
2	1'b0	it1	External interrupt 1 type control. 1: falling edge, 0: low level.
1	1'b0	ie0	External interrupt 0 flag. Set by hardware, when external interrupt int0 (edge/level, depending on settings) is observed. Cleared by hardware when interrupt is processed.
0	1'b0	it0	External interrupt 0 type control. 1:falling edge, 0: low level.

10.3.2 Timer mode register – TMOD

TMODregister is used for configuration of Timer 0 and Timer 1.

TMOD (RW) 0x89

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit0
gate1	ct1	mode1		gate0	ct0	mode0	

1'b0	1'b0	2'b00	1'b0	1'b0	2'b00
RW	RW	RW	RW	RW	RW

Description of Word

Bit	Value	Symbol	Description
7	1'b0	gate1	Timer 1 gate control
6	1'b0	ct1	Timer 1 counter/timer select. 1: Counter, 0: Timer
5:4	2'b00	mode1	Timer 1 mode 00 – Mode 0: 13-bit counter/timer 01 – Mode 1: 16-bit counter/timer 10 – Mode 2: 8-bit auto-reload timer 11 – Mode 3: Timer 1 stopped
3	1'b0	gate0	Timer 0 gate control
2	1'b0	ct0	Timer 0 counter/timer select. 1: Counter, 0: Timer
1:0	2'b00	mode0	Timer 0 mode 00 – Mode 0: 13-bit counter/timer 01 – Mode 1: 16-bit counter/timer 10 – Mode 2: 8-bit auto-reload timer 11 – Mode 3: two 8-bit timers/counters

10.3.3 Timer 0 – TH0, TL0

TH0 (RW) 0x8C

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit0
TH0							
8'h00							
RW							

Description of Word

Bit	Value	Symbol	Description
7:0	8'h00	TH0	higher byte of Timer 0

TL0 (RW) 0x8A

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit0
TL0							
8'h00							
RW							

Description of Word

Bit	Value	Symbol	Description
7:0	8'h00	TL0	lower byte of Timer 0

10.3.4 Timer 1 – TH1, TL1

TH1 (RW) 0x8D

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit0
TH1							
8'h00							
RW							

Description of Word

Bit	Value	Symbol	Description
7:0	8'h00	TH1	higher byte of Timer 1

TL1(RW) 0x8B

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit0
TL1							
8'h00							
RW							

Description of Word

Bit	Value	Symbol	Description
7:0	8'h00	TL1	lower byte of Timer 0

11RTC Wakeup Timer

In order to achieve the lowest possible average current consumption, the processor clock can be stopped under firmware control. Operation can be resumed (wakeup) from the internal RTC wakeup timerorthe RFIRQ.

11.1 Features

- Clock generated from XOSC16Moperation, please refer toCLKLFCTRL register bit6 and bit5
- 24-bit range RTC wakeup timer
- RTC time range from 31.25us to 524s
- RTC wakeup timer only disabled by a power-on reset or by software
- Compare with WU (IRQ), Resolution: 31.25µs.

11.2 Functional description

11.2.1 RTC Wakeup Timer

RTC can give an interrupt at predefined intervals due to value equality between the timer and a compare register. RTC ensures that the functions the interrupt is used for are awoken prior to the interrupt. A wakeup interrupt and RTC timer interrupt are the same interrupt, and known which one is interrupted, please refer to PWRDWN register bit 5.

The RTC timer is a 24-bit timer counting from zero and upwards at the rate of the 32 kHz clock. When the RTC timer is equal to the concatenation of RTCCMP, an RTC IRQ, also referred to as WU, is generated. There is an uncertainty of one CLKLF period, 30.25µs, from when the RTC is started or a new value is given to the RTC compare value registers and until the IRQ is given.If compare mode 11 is used, the RTC IRQ will be given every $\frac{\mathrm{RTCCMP}+1}{32000}(s), RTCCMP = \{RTCCMPH, RTCCMPM, RTCCMPL\}$

The RTC compare value is updated every time RTCCMP is written. This might give unwanted behavior if precaution is not taken when updating any of the variables. When new values are written to RTCCMP, the RTC IRQ should be disabled to prevent unwanted RTC IRQ.

The RTC counter uses the 32 kHz low frequency clock for the RTC timer, and the 32 kHz source must be enabled when using the RTC.

Note

1: The RTC can be disabled by a power-on reset.

2: The RTC can be disabled by software only if the CLKLF is keep on.

3: If the CLKLF has been disabled by software, the RTC timer retains the value before the CLKLF stops and the RTC is enabled, and when the CLKLF is enabled, the RTC timer counts up from current value.

Writing RTCCMPsequences are:

Write:Write RTCCMPL, Write RTCCMPM, Write RTCCMPH

11.3 SFR registers

11.3.1 RTC control register –RTC_CON

WCON (RW) 0xB3

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit0
--	---		---			rtc_cmp_mode	rtc_en
1'b0	1'b0	1'b0	3'b000			1'b0	1'b0
RW	RW	RW	RW			RW	RW

Description of Word

Bit	Value	Symbol	Description
7	1'b0	--	must be 0
6	1'b0	--	Not used
5	1'b0		
4:2	3'b000	--	Not used
1	1'b0	rtc_cmp_mode	Compare mode. 1: The RTCIRQ is assigned when the timer value is equal to RTC Timer. RTC ensures that the function for which the IRQ is intended, are all awoken prior to the RTC IRQ. When the RTC IRQ is assigned, the timer is reset. 0: Same as above, except that the RTC IRQ will not reset the timer. The timer will always wrap around at overflow.
0	1'b0	rtc_en	RTC enable bit 1: RTC is enabled. The clock to the RTC core functionality is running. 0: RTC is disabled. The clock to the RTC core functionality stands still and the timer is reset.

11.3.2 RTC compare register – RTC_CMPL, RTC_CMPM, RTC_CMPH

RTC_CMPL (RW) 0xB4

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit0
rtccmpl							
8'h00							
RW							

Description of Word

Bit	Value	Symbol	Description
7:0	8’h00	rtccmpl	low byte of RTC compare register

RTC_CMPM (RW) 0xB5

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit0
rtccmpm							
8’h00							
RW							

Description of Word

Bit	Value	Symbol	Description
7:0	8’h00	rtccmpm	middle byte of RTC compare register

RTC_CMPH (RW) 0xB6

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit0
rtccmph							
8’h00							
RW							

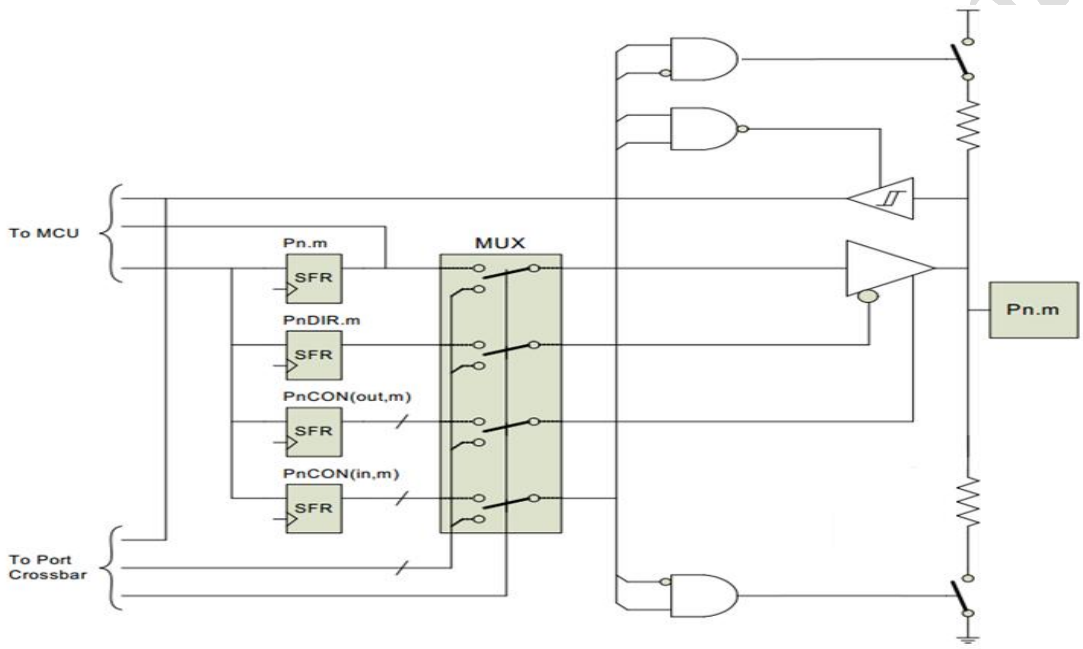
Description of Word

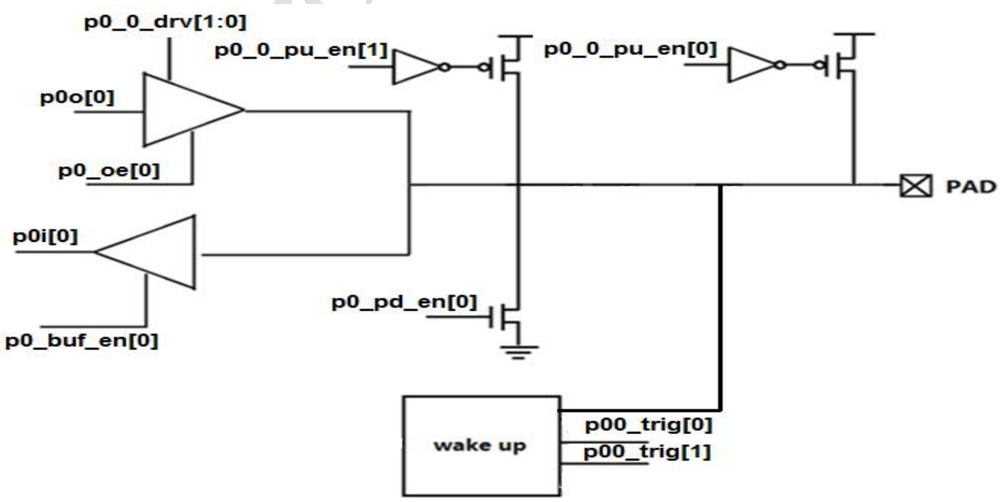
Bit	Value	Symbol	Description
7:0	8’h00	rtccmph	upper byte of RTC compare register

12 GPIO Purpose IO Port and Pin Assignment

The IO pins of the HS6209 are default set to general purpose IO for the MCU. The numbers of available IOs are 18 for the HS6209.The IO pins provide the input, output and inout function to communicate with other components. To filter out high frequency input noise, digital filters are provided.

12.1 Block diagram





12.2 Features

- Digital filter for each pin

- Configurable Direction
- Configurable Drive Strength
  - Output buffer on, normal drive strength
  - Output buffer on, high drive strength
- Configurable Pull Up/Down
  - Input buffer on, no pull up/down resistor
  - Input buffer on, pull up resistor
  - Input buffer on, pull down resistor
  - Input buffer off

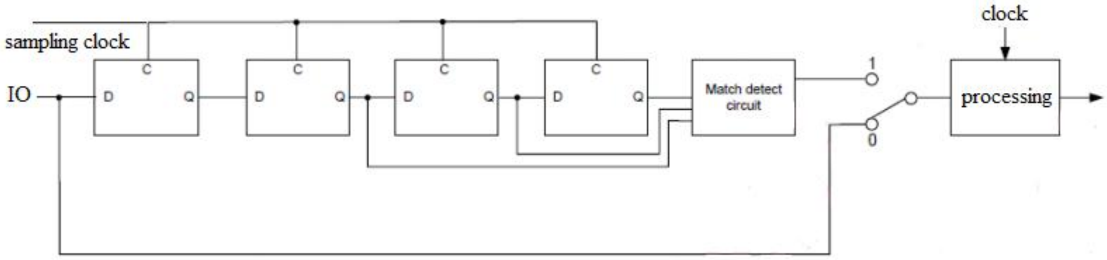
12.3 IO pin maps

pin	Inputs	Outputs	USB		RF SPI		OTP SPI	
			default		rf_spi_mode		otp_mode	
P0.5	p0Di.5	p0Do.5			RF_MOSI	in		
P0.4	p0Di.4	p0Do.4			RF_CSN	in		
P0.3	p0Di.3	p0Do.3			RF_SCK	in		
P0.2	p0Di.2 GPINT0	p0Do.2			RF_IRQ	out	IO	In out
P0.1	p0Di.1	p0Do.1	D+	In	RF_CE	in	FSCK	in
P0.0	p0Di.0	p0Do.0	D-	In	RF_MISO	out	RF_CSN	in

12.4 Functional description

12.4.1 Digital filter

The input signal is sampled, and the level is determined when three matches occur. Block Diagram of Digital Filter is shown in figure below. Registers CFG_DFEN_Px control the enable/disable of digital filter for each I/O pin. To enable the digital filter, set the corresponding bit to 1.



12.4.2 General purpose IO pin functionality

This functionality is multiplexed with the functionality of the PortCrossbar module which takes control and configures the pins depending on the needs of the peripheral block connected.

The pins on the HS6209 are connected by default to a pin Multiplexer (MUX) that is connected to the GPIO registers of the MCU. Register Pn.m(n-port number, m-bit number) contains MCU GPIO data, PDIRn.mregister controls input/output direction and PCONn.mregister controls pin features drive strength and pull up/down resistors for each pin.

When the MCU enables one of the peripheral blocks of the HS6209the pin MUX disconnects the MCU control of the pin and hands control over to the PortCrossbar module to set direction and pin features.

The HS6209 has one Pn.m, PnDIRmand PnCONn for each port. Pn.mand PnDIRmcontrol only one parameter each, this means that a write/read operation to them controls/reads the status of the port directly. However, to control or read the features of a pin you use the PnCONmto write/read to one pin at a time. The PnCONregister contains an address for the pin, information on whether it is an input or an output feature that is to be updated and the feature that is to be enabled.

For example: If four pins in port 1 are set as inputs with the pull up resistor enabled, then this is done with one write to P1DIRand four write operations to P1CONand only updating the pin address in P1CONfor each write.

12.4.3 PortCrossbar functionality

The PortCrossbar sets up connections between the IO pins and the peripheral block of the device.

12.4.4 Dynamic allocation of pins

The PortCrossbar modifies connections dynamically based on run-time variations in system needs of the peripheral blocks of the device. This feature is necessary because the number of available pins is small compared to the combined IO needs of all the peripheral blocks. Consequently, there may be conflicting pin assignments. These are resolved through a set of priorities assigned to each peripheral block.

12.4.5 Dynamic pin allocation for digital blocks

Each digital peripheral block that needs an IO is represented in the pin out tables with the interface names of the block and the direction enforced on each pin. The priority of the blocks relative to potentially conflicting blocks is also shown. If the block is enabled, and no higher priority block is enabled, all the IO needs are granted.

12.5 SFR registers

12.5.1 Control digital filter registers–CFG_DFEN_P0,

Registers CFG_DFEN_Px control the enable/disable of digital filter for each I/O pin. To enable the digital filter, set the corresponding bit to 1.

CFG_DFEN_P0 (RW) 0xE1

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit0
P07_EN	P06_EN	P05_EN	P04_EN	P03_EN	P02_EN	P01_EN	P00_EN
1’b0	1’b0	1’b0	1’b0	1’b0	1’b0	1’b0	1’b0
RW	RW	RW	RW	RW	RW	RW	RW

Description of Word

Bit	Value	Symbol	Description
7	1’b0	P07_EN	Enable bit for digital filter 1:enable 0:disable
6	1’b0	P06_EN	Enable bit for digital filter 1:enable 0:disable
5	1’b0	P05_EN	Enable bit for digital filter 1:enable 0:disable
4	1’b0	P04_EN	Enable bit for digital filter 1:enable 0:disable
3	1’b0	P03_EN	Enable bit for digital filter 1:enable 0:disable
2	1’b0	P02_EN	Enable bit for digital filter 1:enable 0:disable
1	1’b0	P01_EN	Enable bit for digital filter 1:enable 0:disable
0	1’b0	P00_EN	Enable bit for digital filter 1:enable 0:disable

12.5.2 Directioncontrol registers — P0DIR

Desired pin direction and functionality is configured using the configuration registers P0DIR, P1DIR.The PortCrossbar by default (at reset) configures all pins as inputs and connects them to the MCU GPIO.

P0DIR (RW) 0x93

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit0
dir7	dir6	dir5	dir4	dir3	dir2	dir1	dir0
1’b1	1’b1	1’b1	1’b1	1’b1	1’b1	1’b1	1’b1
RW	RW	RW	RW	RW	RW	RW	RW

Description of Word

Bit	Value	Symbol	Description
7	1’b1	dir7	Direction bit for pin P0.7 1:input 0:output
6	1’b1	dir6	Direction bit for pin P0.6 1:input 0:output
5	1’b1	dir5	Direction bit for pin P0.5 1:input 0:output
4	1’b1	dir4	Direction bit for pin P0.4 1:input 0:output
3	1’b1	dir3	Direction bit for pin P0.3 1:input 0:output
2	1’b1	dir2	Direction bit for pin P0.2 1:input 0:output
1	1’b1	dir1	Direction bit for pin P0.1 1:input 0:output
0	1’b1	dir0	Direction bit for pin P0.0 1:input 0:output

12.5.3 The input and output options registers — P0CON,

The input and output options of each pin are configured in the PnCON registers. The PnCON registers have to be written once per pin (one write operation to the PnCON register configures the input/output options of a selected pin in the port).
To read the current input or output options for a pin, you first need to perform a write operation to retrieve the desired bit address and option type (input or output).
For example, to read the output mode of pin P0.5: Write to P0CONwith a bitAddr value of 101, a readAddr value of 1 and ainOut value of 0 (output). Then read from P0CON. The output mode of pin 5 is now found in bits 6:5 of the read data.

P0CON (RW) 0x9E

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit0
pinMode			inOut	readAddr	bitAddr		
3'b000/3'b001			1'b0	1'b0	3'b000		
RW			RW	W	RW		

Description of Word

Bit	Value	Symbol	Description
7:5	3'b000/ 3'b001	pinMode	Functional input or output mode for pins P0.0– P0.7.For a write operation: The functional mode you would like to write to the pin. The inOut field determines if the input or output mode is written, the bitAddr field determines which pin is affected. Output modes using bits 6:5(bit 7 Reserved) 00 (default) 01 10 11 Input modes using bits 7:5 000 Digital input buffer on, no pull up/down resistors 001 Digital input buffer on, pull down resistor connected(default) 010 Digital input buffer on, pull up 30k resistor connected 011 Digital input buffer on, pull up 150k resistor connected other Digital input buffer off For a read operation: The current functional mode of the pin. The inOut field determines if the input or output mode is reported, while the bitAddr field indicates which pin is selected.
4	1'b0	inOut	This bit indicates if the current write operation relates to the input or output configuration of the addressed pin. inOut = 0 – Operate on the output configuration inOut = 1 – Operate on the input configuration

3	1'b0	readAddr	If this bit is set, the purpose of the current write operation is to provide the bit address for later read operations. Consequently, the value of the bitAddr field is saved. The value of the inOut field is also saved, determining if the input or output mode is to be read. The pinMode field is ignored when readAddr is set. If this bit is not set, the pin mode of the addressed pin is updated with the value of the pinMode field. The inOut field determines if the input or output mode is updated.
2:0	3'b000	bitAddr	If the readAddr bit is set, the value of the bitAddr field is stored. For subsequent read operations from P0CON, the pin for which the pinMode will be returned is given by the list below. bitAddr = 000 - P0.0 bitAddr = 001 - P0.1 bitAddr = 010 - P0.2 bitAddr = 011 - P0.3 bitAddr = 100 - P0.4 bitAddr = 101 - P0.5 bitAddr = 110 - P0.6 bitAddr = 111 - P0.7

P1CON (RW) 0x9F

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit0

13 Interrupt

HS6209 has an advanced interrupt controller with 8 sources, as shown in Figure 1. The unit manages dynamic program sequencing based upon important real-time events as signaled from timers, the RF transceiver, pin activity, and so on.

13.1 Feature

- Interrupt controller with 8 sources and 4 priority levels
- Interrupt request flags available
- Interrupt from pin (configurable)

13.2 Block diagram

Block diagram of interrupt structure

13.3 Functional description

When an enabled interrupt occurs, the MCU vectors to the address of the interrupt service routine (ISR) associated with that interrupt, as listed inTable 1.The MCU executes the ISR to completion unless another interrupt of higher priority occurs.

Source	vector	Polarity	Description
IFP	0x0003	low/fall	Interrupt from pin GPINT0
Tf0	0x000B	high	Timer 0 overflow interrupt
tf1	0x001B	high	Timer 1 overflow interrupt
RFRDY	0x0043	rise	RF SPI ready
RFIRQ	0x004B	low	RF interrupt
USBWU	0x005B	rise	USB wakeup interrupt
USBIRQ	0x0063	low	USB interrupt
WU	0x006B	rise	Internal RTC Wakeup interrupt

HS6209 interrupt sources

Note: RFIRQ, X16IRQ and WU are not activated unless wakeup is enabled by WUCON (see section power and clock management).

13.4 SFR registers

Various SFR registers are used to control and prioritize between different interrupts.

The TCON, IRCON,IP0, IP1, IEN0, IEN1 and INTEXP are described in this section. In addition the TCON and T2CON are used, the description for these registers can be found in section Timers/Counters. S0CON is descripted in section Serial port.

13.4.1 Interrupt Enable 0 Register – IEN0

The IEN0 register is responsible for global interrupt system enabling/disabling and also Timer 0, 1 and 2, Port 0 and Serial Port individual interrupts enabling/disabling.

Address	Reset value	Bit	Description
0xA8	0x00	7	1: Enable interrupts. 0: all interrupts are disabled
		6	Not used
		5	
		4	
		3	1: Enable Timer1 overflow (tf1) interrupt
		2	
		1	1: Enable Timer0 overflow (tf0) interrupt
		0	1: Enable Interrupt From Pin (IFP) interrupt

IEN0 register

13.4.2 Interrupt Enable 1 Register – IEN1

Address	Reset value	Bit	Description
0xB8	0x00	7	1:
		6	Not used
		5	1: Internal wakeup (WU) interrupt enable
		4	1: USB interrupt enable
		3	1: USB wakeup interrupt enable
		2	
		1	1: RF (RFIRQ) interrupt enable
		0	1: RF SPI ready (RFRDY) interrupt enable

IEN1 register

Address	Reset value	Bit	Description
0xA6	0x01	7:4	Not used
		3	1: Enable GP INT0 (from pin) 0 to IFP
		2:0	Not used

INTEXP register

13.4.3 Interrupt Priority Registers – IP0, IP1

The 17 interrupt sources are grouped into six priority groups. For each of the groups, one of four priority levels can be selected. They can be selected by setting appropriate values in IP0 and IP1 registers.

The contents of the Interrupt Priority registers define the priority levels for each interrupt source according to the tables below.

Address	Reset value	Bit	Description
0xA9	0x00	7:6	Not used
		5:0	Interrupt priority. Each bit together with corresponding bit from IP1 register specifies the priority level of the respective interrupt priority group.

IP0 register

Address	Reset value	Bit	Description
0xB9	0x00	7:6	Not used
		5:0	Interrupt priority. Each bit together with corresponding bit from IP0 register specifies the priority level of the respective interrupt priority group.

IP1 register

Group	Interrupt bits	Priority group		
0	IP1[0]，IP0[0]	IFP		RFRDY
1	IP1[1]，IP0[1]	Timer 0 interrupt		RFIRQ
2	IP1[2]，IP0[2]			
3	IP1[3]，IP0[3]	Timer 1 interrupt		USB wakeup
4	IP1[4]，IP0[4]			USB interrupt
5	IP1[5]，IP0[5]			WU

Priority groups

IP1.x	IP0.x	Priority level
0	0	Level 0 (lowest)
0	1	Level 1
1	0	Level 2
1	1	Level 3 (highest)

Priority levels (x is the number of priority group)

13.4.4 Interrupt Request Control Registers – IRCON

The IRCON register contains interrupt request flags.

Address	Reset value	Bit	Auto clear[a]	Description
0xC0	0x00	7		
		6		
		5	Yes	Internal wakeup interrupt flag
		4		USB interrupt flag
		3	Yes	USB wakeup interrupt flag
		2		
		1		RF (RFIRQ) interrupt flag

		0		RF SPI ready (RFDY) interrupt flag

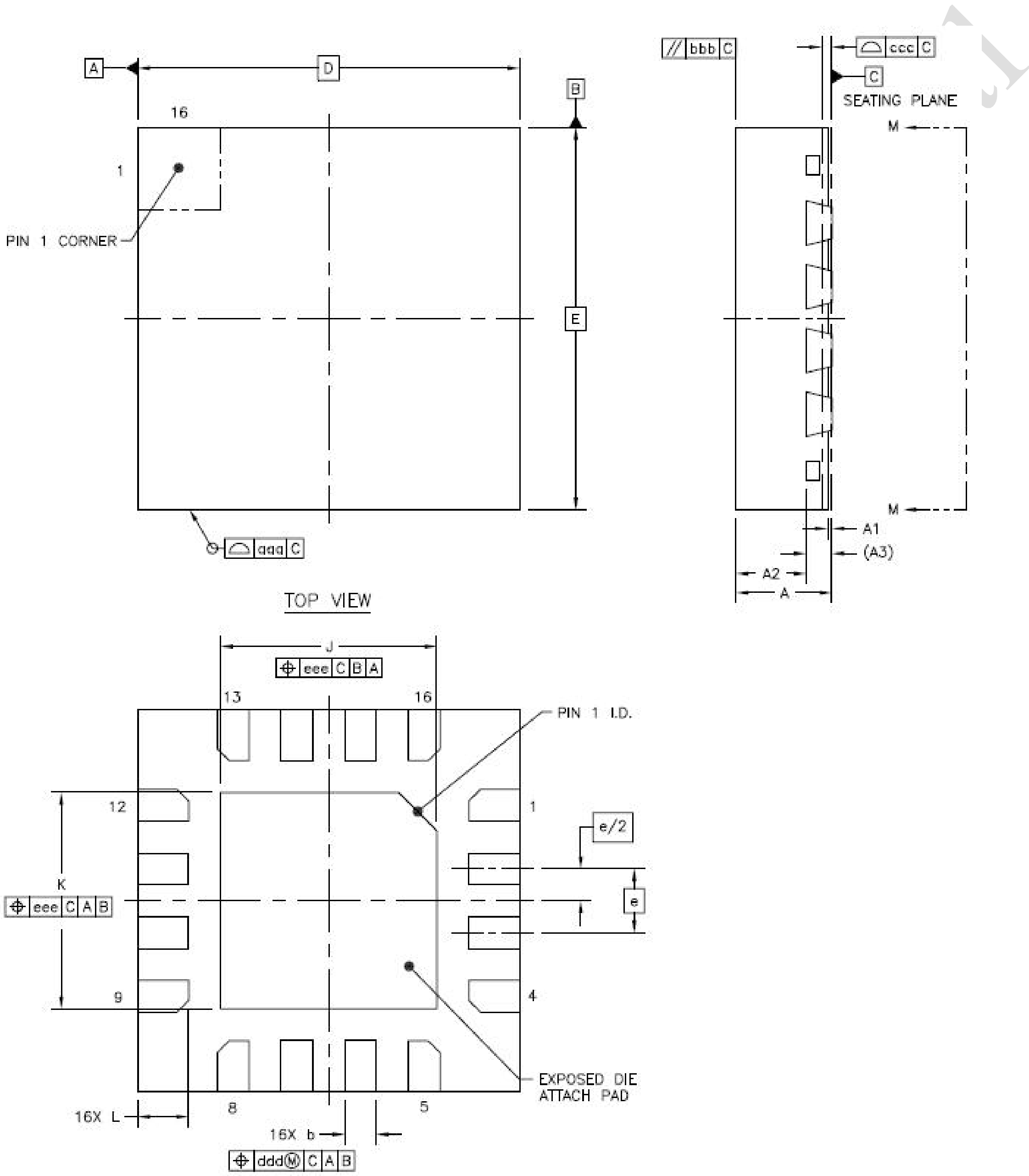
a：Auto clear means that the flag is cleared by hardware automatically when the corresponding service routine is vectored.

b：The flag will be cleared by hardware, but need to be clear the correspondingflag(refer to T2SR register) by software.

IRCON register

14 Package size

14.1 QFN16 3X3 mm



DESCRIPTION		SYMBOL	MILLIMETER		
			MIN	NOM	MAX
TOTAL THICKNESS		A	0.7	0.75	0.8
STAND OFF		A1	0	0.035	0.05
MOLD THICKNESS		A2	---	0.55	0.57
L/F THICKNESS		A3	0.203 REF		
LEAD WIDTH		b	0. 20	0. 25	0. 30
BODY SIZE	X	D	3 BSC		
	Y	E	3 BSC		
LEAD PITCH		e	0.5 BSC		
EP SIZE	X	J	1. 65	1. 70	1. 75
	Y	K	1. 65	1. 70	1. 75
LEAD LENGTH		L	0.35	0.4	0.45
PACKAGE EDGE TOLERANCE		aaa	0.1		
MOLD FLATNESS		bbb	0.1		
COPLANARITY		ccc	0.08		
LEAD OFFSET		ddd	0.1		
EXPOSED PAD OFFSET		eee	0.1		