BK2535

Datasheet

FLIP51 MCU+RF

CONFIDENTIAL

Beken Corporation
Building 41, Capital of Tech Leaders, 1387 Zhangdong Road, Zhangjiang High-Tech Park, Pudong New District, Shanghai, China
Tel： (86)21 51086811
Fax： (86)21 60871089

This document contains information that may be proprietary to, and/or secrets of, Beken Corporation. The contents of this document should not be disclosed outside the companies without specific written permission.

Disclaimer: Descriptions of specific implementations are for illustrative purpose only, actual hardware implementation may differ.

Downloaded From Oneyac.com

Revision History

Version	Date	Author(s)	Description
0.1	Jan. 20,2015	Lizhen	Initial flash version
0.2	Feb. 15,2014	Lizhen	Update USB RF etc.
0.3	Apr. 15,2014	Lizhen	Update package and mode select, add FLASH control
0.4	Jun. 15,2014	Lizhen	Update LBD part
0.5	Aug. 18,2015	Lizhen	Update the format
0.6	Nov. 19,2015	Ronghui	Updated the feature part
0.7	4.12.2016	Guodong	Updatad PWM address

CONFIDENTIAL

Downloaded From Oneyac.com

Table of Contents

CONFIDENTIAL

Downloaded From Oneyac.com

CONFIDENTIAL

Downloaded From Oneyac.com

CONFIDENTIAL

Downloaded From Oneyac.com

List of Figures

CONFIDENTIAL

Downloaded From Oneyac.com

List of tables

CONFIDENTIAL

Downloaded From Oneyac.com

DRAFT IDENTIAL

Downloaded From Oneyac.com

1. Introduction

The BK2535 is a RF SOC chip, which combine a 2.4 GHz radio, a fast 8051 compatible CPU named FILE51, essential peripherals, and on-air compatibility with the other product series from BEKEN. The BK2535 is a perfect fit for HID applications, RF tags and remote controlled products.

2. Feature

- 1.8 V to 3.6 V power supply
- FLIP51 MCU compatible with 8051
- A 4-stage pipeline architecture that enables to execute most of the instructions in a single clock cycle.
- 32k bytes FLASH for program
- 256 Bytes IRAM and 2k Bytes SRAM
- Embedded three Timer/Counter
- Support UART I2C SPI interface
- Support AES encryption
- A pseudo random number generator embedded
- Total 40 GPIO available
- The dedicated 2 PWM available and 6 PCA can be used as PWM
- The embedded BIRD (Built-In Real-time Debugger) system for online debug
- 8 channel ADC embedded
- Booster embedded
- Integrated 2.4G RF transceiver
- low power consumption, embedded with 32k RC oscillator

CONFIDENTIAL

Downloaded From Oneyac.com

3. Block Diagram

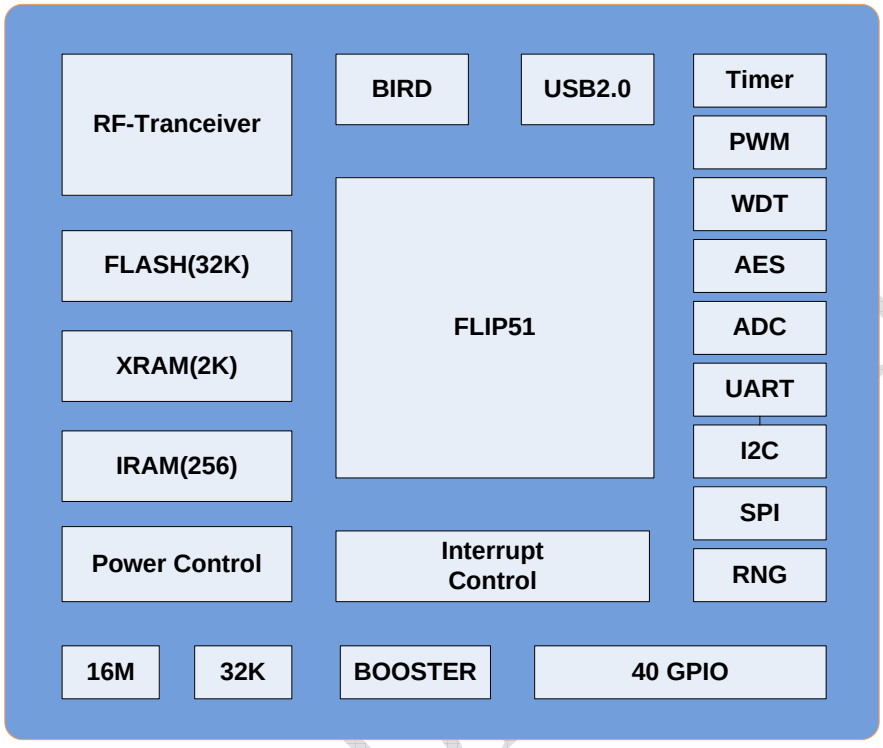


Figure 1 BK2535 Block Diagram

CONFIDENTIAL

Downloaded From Oneyac.com

4. PIN information

4.1. BK2535_QFN32

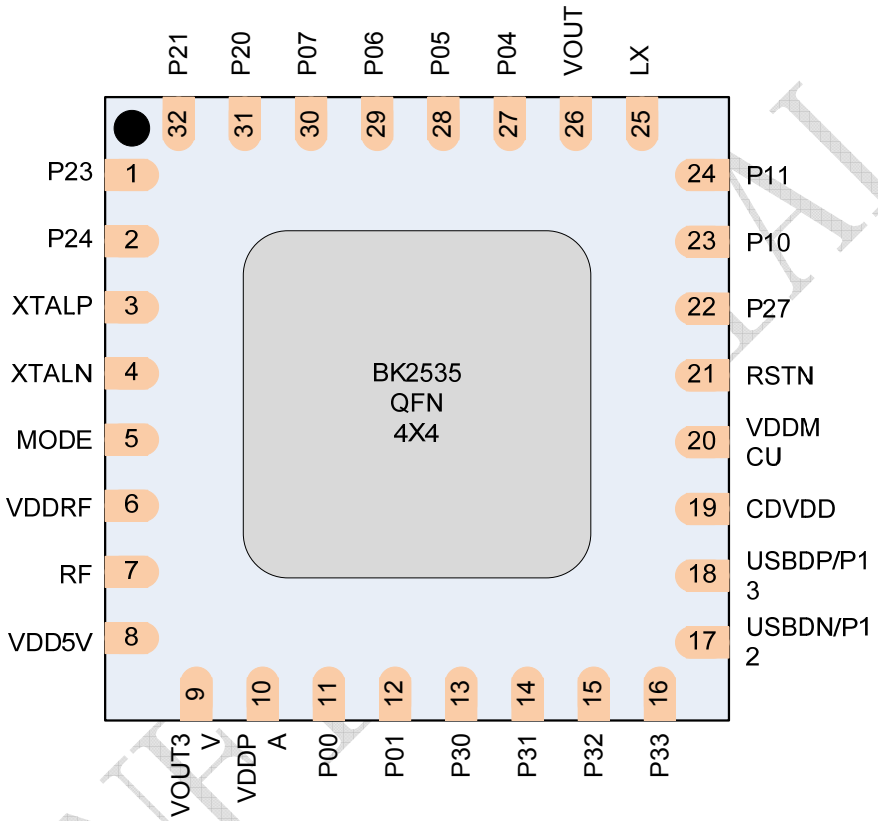


Figure 2 BK2535_M

NO.	Name	Pin Function	Description
1	P2.3	Digital I/O	General I/O, or clock for SMBUS (I2C)
2	P2.4	Digital I/O	General I/O, or data I/O for SMBUS (I2C)
3	XTALP	Analog output	Oscillator output
4	XTALN	Analog input	Oscillator input
5	MODE	Digital input	Mode selection
6	VDDRF	Power supply	Power supply for RF part module
7	RF	RF port	ANTENNA port
8	VDD5V	5V Power	5V supply from USB, used for inner 5V LDO
9	VOUT3V	Analog output	3v power output, connected with decoupling CAP
10	VDDPA	Power supply	Power supply for PA part

CONFIDENTIAL

Downloaded From Oneyac.com

11	P0.0	Digital I/O	General I/O, or input for timer0
12	P0.1	Digital I/O	General I/O, or input for timer1
13	P3.0	I/O or analog input	General I/O, or input of ADC0
14	P3.1	I/O or analog input	General I/O, or input of ADC1
15	P3.2	I/O or analog input	General I/O, or input of ADC2
16	P3.3	I/O or analog input	General I/O, or input of ADC3
17	USB_DN/P12	Digital I/O	USB input N, or General I/O
18	USB_DP/P13	Digital I/O	USB input P, or General I/O
19	CDVDD	Analog output	1.5V power output, connected with decoupling CAP
20	VDDMCU	Power supply	Power supply for MCU part
21	RSTN	Digital input	Reset for chip, active low
22	P2.7	Digital I/O	General I/O, or PWM1
23	P1.0	Digital I/O	General I/O, or input for external interrupt 0, active low
24	P1.1	Digital I/O	General I/O, or input for external interrupt 1, active low
25	LX	Analog input	Switch, connect to BAT through a inductance
26	VOUT	Analog output	The output of booster
27	P0.4	Digital I/O	General I/O, or MOSI for SPI
28	P0.5	Digital I/O	General I/O, or MISO for SPI
29	P0.6	Digital I/O	General I/O, or SCK for SPI
30	P0.7	Digital I/O	General I/O, or chip select for SPI
31	P2.0	Digital I/O	General I/O, or input for UART0
32	P2.1	Digital I/O	General I/O, or output for UART0

Table 1 QFN32 PIN Definition

CONFIDENTIAL

Downloaded From Oneyac.com

4.2. BK2535_QFN56

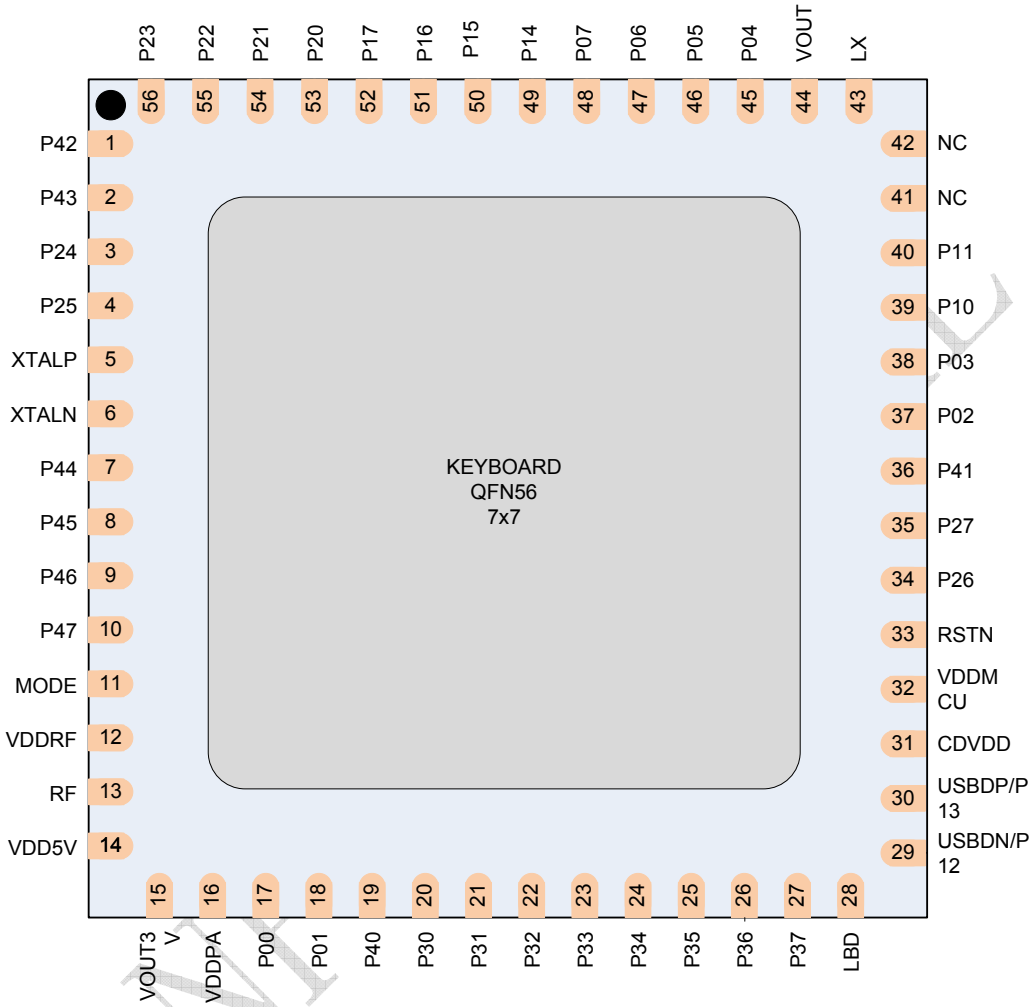


Figure 3 BK2535_K

PIN	Name	Pin Function	Description
1	P4.2	Digital I/O	General I/O, or PCA port
2	P4.3	Digital I/O	General I/O, or PCA port
3	P2.4	Digital I/O	General I/O, or data I/O for SMBUS (I2C)
4	P2.5	Digital I/O	General I/O
5	XTALP	Analog output	Oscillator output
6	XTALN	Analog input	Oscillator input
7	P44	Digital I/O	General I/O, or PCA port
8	P45	Digital I/O	General I/O, or PCA port
9	P46	Digital I/O	General I/O,
10	P47	Digital I/O	General I/O,

Downloaded From Oneyac.com

11	MODE	Digital input	Mode selection
12	VDDRF	Power supply	Power supply for RF part module
13	RF	RF port	ANTENNA port
14	VDD5V	5V Power	5V supply from USB, used for inner 5V LDO
15	VOUT3V	Analog output	3v power output, connected with decoupling CAP
16	VDDPA	Power supply	Power supply for PA part
17	P0.0	Digital I/O	General I/O, or input for timer0
18	P0.1	Digital I/O	General I/O, or input for timer1
19	P4.0	Digital I/O	General I/O, or PCA port
20	P3.0	I/O or analog input	General I/O, or input of ADC0
21	P3.1	I/O or analog input	General I/O, or input of ADC1
22	P3.2	I/O or analog input	General I/O, or input of ADC2
23	P3.3	I/O or analog input	General I/O, or input of ADC3
24	P3.4	I/O or analog input	General I/O, or input of ADC4
25	P3.5	I/O or analog input	General I/O, or input of ADC5
26	P3.6	I/O or analog input	General I/O, or input of ADC6
27	P3.7	Digital I/O or analog input	General I/O, or input of ADC7
28	LBD	Analog input	Low power detect pin
29	USB_DN/P12	Digital I/O	USB input N, or General I/O
30	USB_DP/P13	Digital I/O	USB input P, or General I/O
31	CDVDD	Analog output	1.5V power output, connected with decoupling CAP
32	VDDMCU	Power supply	Power supply for MCU part
33	RSTN	Digital input	Reset for chip, active low
34	P2.6	Digital I/O	General I/O, or PWM0
35	P2.7	Digital I/O	General I/O, or PWM1
36	P4.1	Digital I/O	General I/O, or PCA port
37	P0.2	Digital I/O	General I/O, or input for timer2
38	P0.3	Digital I/O	General I/O, or acquire/reload trigger signal for timer2
39	P1.0	Digital I/O	General I/O, or input for external interrupt 0, active low
40	P1.1	Digital I/O	General I/O, or input for external interrupt 1, active low
41	NC		
42	NC		
43	LX	Analog input	Switch, connect to BAT through a inductance
44	VOUT	Analog output	The output of booster
45	P0.4	Digital I/O	General I/O, or MOSI for SPI
46	P0.5	Digital I/O	General I/O, or MISO for SPI
47	P0.6	Digital I/O	General I/O, or SCK for SPI
48	P0.7	Digital I/O	General I/O, or chip select for SPI
49	P1.4	Digital I/O	General I/O
50	P1.5	Digital I/O	General I/O
51	P1.6	Digital I/O	General I/O
52	P1.7	Digital I/O	General I/O
53	P2.0	Digital I/O	General I/O, or input for UART0

CONFIDENTIAL

Downloaded From Oneyac.com

54	P2.1	Digital I/O	General I/O, or output for UART0
55	P2.2	Digital I/O	General I/O,
56	P2.3	Digital I/O	General I/O, or clock for SMBUS (I2C)

Table 2 QFN56 PIN definition

CONFIDENTIAL

Downloaded From Oneyac.com

5. MCU Description

5.1. FLIP51 Micro-Controller

5.1.1. Introduction

The FLIP8051 is an improved option of the 80c51 microcontroller. It is 100% binary code upward compatible with the legacy 80c51.

Its pipeline architecture provides an increase of processing speed an average nine times, when running at the same clock frequency as a standard 80c51 real component.

- Full binary code compatibility with the legacy 80C51/52
- 2 Data Pointers for faster memory copies and indexing.
- Hardware-controlled Wait State solution for asynchronous peripherals
- Static synchronous design with no internal tri-state busses
- Power-saving modes provide solutions for low-power applications

CONFIDENTIAL

Downloaded From Oneyac.com

5.1.2. MCU diagram

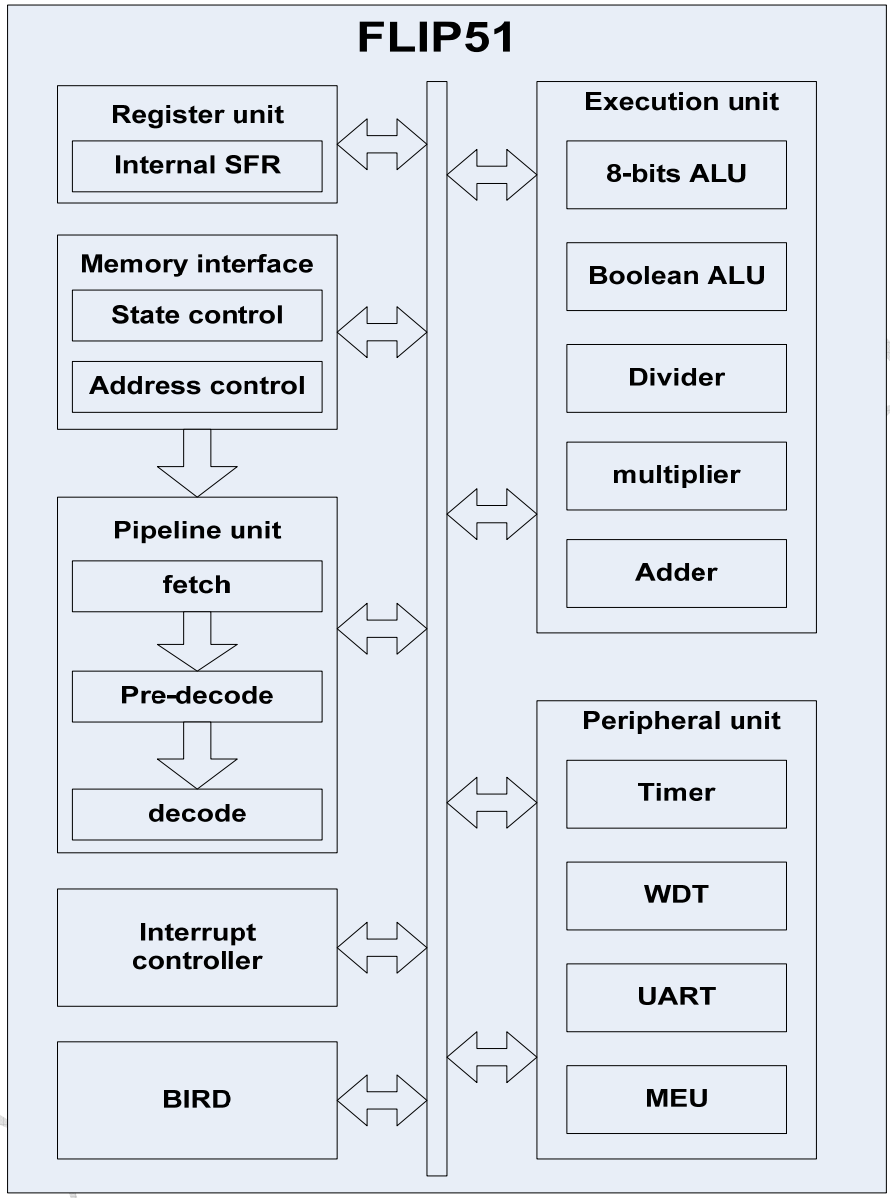


Figure 4 FlIP51 architecture

Downloaded From Oneyac.com

5.2. FLIP8051 address space

5.2.1. Overview

The memory organization of the Flip8051 is similar to that of standard 80C51. There are three separate memory spaces: CODE space (program memory), the XDATA space (external data memory) and the IDATA space (internal data memory).

These memory spaces shared the same address space but are accessed with different instruction types.

There are organized as follow for BK2535:

CODE space: up to 32K Bytes of addressing range

XDATA space: up to 2K Bytes of addressing range

IDATA space: up to 256 Bytes.

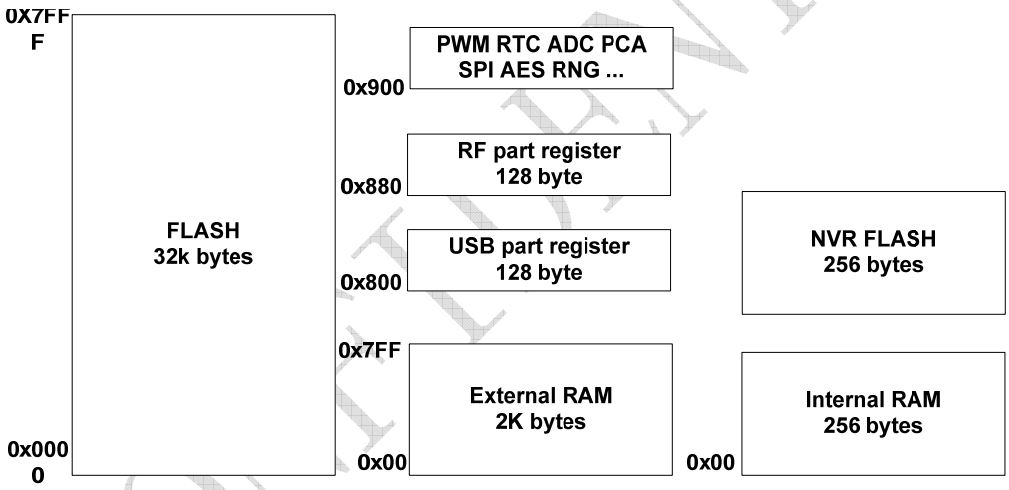


Figure 5 FLIP51 Space

5.2.2. Program Memory (CODE space)

The Flip8051 has a 64K Bytes code space (32K for BK2535). Program memory is normally assumed to be read only and can be accessed only by MOVC instruction (or of course by the instruction fetch).

Two addressing modes are available for MOVC instructions:

16-bit data pointer (@A+DPTR).

The MOVC instructions use these indirect modes to access the current 64 K page of the code memory.

CONFIDENTIAL

Downloaded From Oneyac.com

16-bit program counter (@A+PC).

The MOVC instruction uses this indirect mode to access the 64 K page of the code memory.

5.2.3. External Data Memory (XDATA space)

The **External Data memory** shares address bus with program memory. This data space can be up to 64K Bytes (2K for BK2535).

The external data memory can be accessed only by the standard MOVX instructions (plus some new instructions of the WHIRL instruction set)

Two addressing modes are available for MOVX instructions:

Byte register (@Ri, i = 0,1).

Registers R0 and R1 indirectly address external data memory locations 00h-FFh. When MOVX instructions use this indirect mode, the MSB of the 16-bit address is filled with the content of MPAGE SFR (0A1h). Then, it allows MOVX @Ri instruction to access to 64K Bytes of external data memory. Usually, in 80C51 application, the Port 2 is used to this address extension. In order to keep software compatibility with existing 80C51 program, the register MPAGE is also updated by any value written at P2 register.

16-bit data pointer (@DPTR).

The MOVX instructions use these indirect modes to access the page of the external data RAM pointed by the extended data pointer (DPX).

5.2.4. Internal Data Memory (IDATA space)

The **Internal data memory** is composed by 256 bytes of internal RAM and by a number of SFRs.

The main difference between these IDATA and XDATA spaces is the kind of instructions that enable to access to these memories. Most of the “data transfer” instructions are dedicated to access internal data memory (IDATA) since there are only four instructions (MOVX) dedicated to access external data memory.

Moreover, only indirect addressing mode is available for XDATA whilst IDATA can be addressed by register, direct, register-indirect or immediate addressing mode. This provides a higher flexibility to access data. In addition, the Flip8051 memory interface with IDATA space is optimized and then access time to this space is faster than the access time of XDATA for both read/write operations.

CONFIDENTIAL

Downloaded From Oneyac.com

5.2.4.1. Internal Data memory organization

The internal data memory is divided into 3 spaces, which are referred to as the **Lower 128**, **Upper 128** and **SFR space**. Either direct or indirect addressing may be used to access the **lower 128** bytes of internal data memory. The **upper 128** bytes of internal data memory are accessible by indirect addressing only while direct addressing to region above 0x7F will access **SFR space**.

In the Flip8051, the SFRs are implemented internally to the model using Flip-Flops.

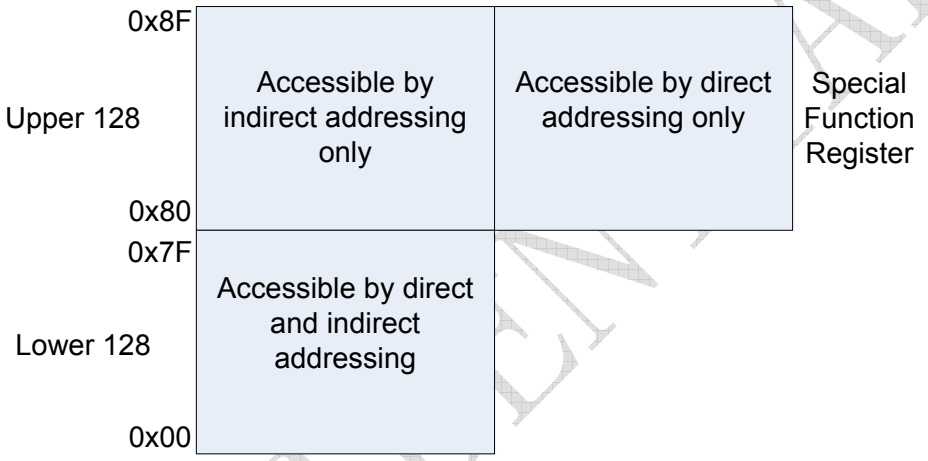


Figure 6 Internal Data Memory

5.2.4.2. Internal ram: lower 128 byes

0X78	General Purpose Memory								0X7F	
0X70									0X77	
0X68									0X6F	
0X60									0X67	
0X58									0X5F	
0X50									0X57	
0X48									0X4F	
0X40									0X47	
0X38									0X3F	
0X30									0X37	
0X28	Bit Addressable Memory								0X2F	
0X20									0X27	
0X18	R0	R1	R2	R3	R4	R5	R6	R7	0X1F	BANK3
0X10	R0	R1	R2	R3	R4	R5	R6	R7	0X17	BANK2
0X08	R0	R1	R2	R3	R4	R5	R6	R7	0X0F	BANK1

CONFIDENTIAL

Downloaded From Oneyac.com

0X00	R0	R1	R2	R3	R4	R5	R6	R7	0X07	BANK0

Figure 7 Internal Ram Lower 128 Bytes

The lower 128 bytes of Internal Data Memory is organized in three distinct areas:

0x00-0x1F: The Register Banks are at the lowest 32 bytes of the internal data memory. Only one Register Bank is used at a time when an instruction uses R0 to R7. 2 bits in Processor Status Word (PSW), called RS1 and RS0, control the selection of the Register Bank. Bank 0 is selected upon reset. Indirect addressing mode used R0 and R1 as index registers

0x20-0x2F: This memory space contains a general-purpose memory, which is bit addressable as well as byte addressable. The bit address ranged from 0 to 0x7F. When bit addressing is used in an instruction, the bit access in this region will occur. In this memory range, when bit addressing is used, bit address 0x00 is the bit 0 of address 0x20 while bit 7 of the byte 0x20 has bit address 0x07. Bit address 0x7F is the bit 7 of address 0x2F. A bit access is different than a byte access by the type of instruction used.

0x30-0x7F: A general-purpose byte-addressable memory is located above address 0x30. It can be accessed both by direct or indirect addressing mode.

5.2.4.3. Internal Ram: Upper 128 Bytes

The usage of the addresses between 0x80 and 0xFF is up to the user. This memory can be used for any purpose providing that indirect addressing mode is used when accessing this memory space, otherwise the Special Function Register memory will be accessed.

5.2.4.4. The Stack and the stack pointer

The stack refers to an area of internal RAM that is used in conjunction with certain instructions (PUSH, POP) to store and retrieve data quickly. The Stack pointer register (SP, 0x81) is used to hold an internal RAM address that is called the “top of the stack”. The data held in the SP register is the address in internal RAM where the last byte of data was stored by a stack operation. The reset value of Stack pointer register is 0X07 and which can be changed to any internal RAM address by the programmer. Usually, the stack is located high in the RAM to avoid conflict with the work register, bit and byte area in internal RAM.

5.2.4.5. Special Function Registers

The direct-access data memory locations from 0x80 to 0xFF constitute the special function registers (SFRs). All the special function registers of the original 80C51 are present in the Flip8051. SFRs with addresses ending in 0x0 or 0x8 (e.g. P0, TCON, P1, SCON, IE, etc.) are bit-addressable as well as byte-addressable. All other SFRs are byte-addressable only.

Downloaded From Oneyac.com

The special function registers (SFRs) reside in their associated peripherals or in the core. The following tables shows the SFR address space with the SFR mnemonics and reset values. Unoccupied locations in the SFR space are unimplemented, i.e. no register exists. If an instruction attempts to write to an unimplemented SFR location, the instruction executes, but nothing is actually written. If an unimplemented SFR location is read, it returns an unspecified value.

SFR Name	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	addr
									0x80
SP	-	-	-	-	-	-	-	-	0x81
DPL	-	-	-	-	-	-	-	-	0x82
DPH	-	-	-	-	-	-	-	-	0x83
CKCON	CKDIV1	CKDIV0	smod0	x	x	x	x	x	0x84
CLK_EN CFG	adc_en	timer_en	uart_en	pwm_en	spi_en	i2c_en	aes_mud_ en	wdt_en	0x85
PCON2	SMOD	EUSB	CMD_RST	Latch_en	deep_sleep	OSC32k	RC32k	IDLE	0x86
									0x87
TCON	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	0x88
TMOD	GATE	C/T	M1	M0	GATE	C/T	M1	M0	0x89
TL0	-	-	-	-	-	-	-	-	0x8A
TL1	-	-	-	-	-	-	-	-	0x8B
TH0	-	-	-	-	-	-	-	-	0x8C
TH1	-	-	-	-	-	-	-	-	0x8D
CCMCON	-	-	-	-	-	-	-	-	0x8E
CCMVAL	-	-	-	-	-	-	-	-	0x8F
P1	-	-	-	-	-	-	-	-	0x90
									0x91
DPSEL	x	x	x	x	x	x	x	DPSEL0	0x92
P1IN_EN	-	-	-	-	-	-	-	-	0x93
P2IN_EN	-	-	-	-	-	-	-	-	0x94
P3IN_EN	-	-	-	-	-	-	-	-	0x95
									0x96
MMS	-	-	-	-	-	-	-	-	0x97
SCON0	SM0	SM1	SM2	REN	TB8	RB8	TI	RI	0x98
SBUF0	-	-	-	-	-	-	-	-	0x99
									0x9A
PAGE_A	x	x	x	x	x	x	x	x	0x9B

Downloaded From Oneyac.com

（NA）									
PAGE_B（NA）	x	x	x	x	x	x	x	x	0x9C
PAGE_C（NA）	x	x	x	x	x	x	x	x	0x9D
P1OUT_N	-	-	-	-	-	-	-	-	0x9E
P2OUT_N	-	-	-	-	-	-	-	-	0x9F
P2	-	-	-	-	-	-	-	-	0xA0
MPAGE	-	-	-	-	-	-	-	-	0xA1
									0xA2
									0xA3
									0xA4
P3OUT_N	-	-	-	-	-	-	-	-	0xA5
WDT	x	x	x	state	x	ps2	ps1	ps0	0xA6
									0xA7
IE	EA	x	ET2	ES	ET1	EX1	ET0	EX0	0xA8
									0xA9
P1_PU	-	-	-	-	-	-	-	-	0xAA
P2_PU	-	-	-	-	-	-	-	-	0xAB
P3_PU	-	-	-	-	-	-	-	-	0xAC
									0xAD
									0xAE
P1_PD	-	-	-	-	-	-	-	-	0xAF
P3	-	-	-	-	-	-	-	-	0xB0
									0xB1
									0xB2
									0xB3
P2_PD	-	-	-	-	-	-	-	-	0xB4
P3_PD	-	-	-	-	-	-	-	-	0xB5
									0xB6
									0xB7
IP	-	-	-	-	-	-	-	-	0xB8
									0XB9
									0xBA
									0xBB
									0xBC
									0xBD
									0xBE
									0xBF
AIF	-	-	-	-	-	-	-	-	0xC0

Downloaded From Oneyac.com

									0xC1
									0xC2
									0xC3
									0xC4
									0xC5
									0xC6
									0xC7
T2CON	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	CT2	CPRL2	0xC8
NMI	x	x	x	x	x	x	nmi_en	nmi_flag	0xC9
RCAP2L	-	-	-	-	-	-	-	-	0xCA
RCAP2H	-	-	-	-	-	-	-	-	0xCB
TL2	-	-	-	-	-	-	-	-	0xCC
TH2	-	-	-	-	-	-	-	-	0xCD
									0xCE
									0xCF
PSW	-	-	-	-	-	-	-	-	0xD0
									0xD1
I2CM DATA_IE	x	x	ETBE	x	ETBF	ERBE	x	ERBF	0xD2
									0xD3
I2CM CALLADDR0	RWN	CADDR[6]	CADDR[5]	CADDR[4]	CADDR[3]	CADDR[2]	CADDR[1]	CADDR[0]	0xD4
									0xD5
									0xD6
									0xD7
									0xD8
P1_OPDR	-	-	-	-	-	-	-	-	0xD9
P2_OPDR	-	-	-	-	-	-	-	-	0xDa
P3_OPDR	-	-	-	-	-	-	-	-	0xDb
									0xDc
									0xDd
									0xDe
									0xDf
ACC	-	-	-	-	-	-	-	-	0xE0
I2CM CTRL	x	x	WAIT	x	STOP	SRST	STA	BUSY	0xE1
I2CM RXDATA	-	-	-	-	-	-	-	-	0xE2
I2CM TXDATA	-	-	-	-	-	-	-	-	0xE3
I2CM PRESC	-	-	-	-	-	-	-	-	0xE4

Downloaded From Oneyac.com

I2CM TXRX_STS		x	x	DNA	SANA	UNF	OVF	NEND	0XE5
I2CM DATA_STS	x	x	TBE	x	TBF	RBE	x	RBF	0XE6
I2CM TXRX_IE	x	x	x	EDNA	ESANA	EUNF	EOVF	ENEND	0xE7
AIE	-	-	-	-	-	-	-	-	0xE8
PALT0	x	x	T2_EX_EN	T2_IN_EN	T1_IN_EN	T0_IN_EN	EX1_IN_EN	EX0_IN_EN	0xE9
EXSLEEP	x	x	x	x	x	x	further_sleep	supper_sleep	0xEA
									0xEB
P1_WUEN	-	-	-	-	-	-	-	-	0xEC
P2_WUEN	-	-	-	-	-	-	-	-	0xED
P3_WUEN	-	-	-	-	-	-	-	-	0xEE
									0xEF
B									0xF0
									0xF1
									0xF2
									0xF3
									0xF4
									0xF5
									0xF6
PALT1			PCA_IO	PWM_IO		UART0_IO	SPI_IO	I2C_IO	0xF7
AIP	-	-	-	-	-	-	-	-	0xF8
									0xF9
P1_WUMOD	-	-	-	-	-	-	-	-	0xFA
P2_WUMOD	-	-	-	-	-	-	-	-	0xFB
P3_WUMOD	-	-	-	-	-	-	-	-	0xFC
									0xFD
									0xFE
									0xFF

Table 3 *Special Function Registers Memory Map*

5.2.4.6. SFR table for MCU part

Register	Address	Description	Reset value
ACC	0xE0	Accumulator	00h
B	0xF0	B Register	00h
DPH	0x83	Data Pointer high byte	00h
DPL	0x82	Data Pointer low byte	00h

Downloaded From Oneyac.com

DPSEL	0x92	Data Pointer selection	00h
IE	0xA8	Interrupt Enable Control	00h
IP	0xB8	Interrupt Priority Control	00h
MPAGE	0xA1	Memory page register	00h
PCON2	0x86	Power Control	00h
PSW	0xD0	Program Status Word	00h
SP	0x81	Stack Pointer	07h

Table 4 *Core SFRs*

Register	Address	Description	Reset value
AIE	0xE8	Additional interrupt enable	00h
AIF	0xC0	Additional interrupt flag	00h
AIP	0xF8	Additional interrupt priority	00h

Table 5 *Additional interrupt SFRs*

Register	Address	Description	RST value
P0	0x80	Port0 value	0xFF
P0IN_EN_	0x91	Port input enable, active high	0xFF
P0OUT_N_	0x9A	Port output enable, active low	0xFF
P0_PU	0xA9	Port pull-up selection	0xFF
P0_PD	0xAE	Port pull-down selection	0x00
P0_OPDR	0xD8	Open drain selection	0x00
P0_WUEN	0xEB	Port wake up enable	0x00
P0_WKMOD	0xF9	Port wake up mode selection	0x00
P1	0x80	Port0 value	0xFF
P1IN_EN_	0x93	Port input enable, active high	0xFF
P1OUT_N_	0x9E	Port output enable, active low	0xFF
P1_PU	0xAA	Port pull-up selection	0xFF
P1_PD	0xAF	Port pull-down selection	0x00
P1_OPDR	0xD9	Open drain selection	0x00
P1_WUEN	0xEC	Port wake up enable	0x00
P1_WKMOD	0xFA	Port wake up mode selection	0x00
P2	0xA0	Port0 value	0xFF
P2IN_EN_	0x94	Port input enable, active high	0xFF
P2OUT_N_	0x9F	Port output enable, active low	0xFF

CONFIDENTIAL

Downloaded From Oneyac.com

P2_PU	0xAB	Port pull-up selection	0xFF
P2_PD	0XB4	Port pull-down selection	0x00
P2_OPDR	0xDA	Open drain selection	0x00
P2_WUEN	0xED	Port wake up enable	0x00
P2_WKMOD	0xFB	Port wake up mode selection	0x00
P3	0xB0	Port0 value	0xFF
P3IN_EN_	0x95	Port input enable, active high	0xFF
P3OUT_N_	0Xa5	Port output enable, active low	0xFF
P3_PU	0xAC	Port pull-up selection	0xFF
P3_PD	0xB5	Port pull-down selection	0x00
P3_OPDR	0xDB	Open drain selection	0x00
P3_WUEN	0xEE	Port wake up enable	0x00
P3_WKMOD	0xFC	Port wake up mode selection	0x00
P4	0xB7	Port0 value	0xFF
P4IN_EN_	0x96	Port input enable, active high	0xFF
P4OUT_N_	0xa7	Port output enable, active low	0xFF
P4_PU	0xAD	Port pull-up selection	0xFF
P4_PD	0xB6	Port pull-down selection	0x00
P4_OPDR	0xDC	Open drain selection	0x00
P4_WUEN	0xEF	Port wake up enable	0x00
P4_WKMOD	0xFD	Port wake up mode selection	0x00

Table 6 *I/O ports SFRs*

NOTE: some ports are not available in BK2535; please refer to the package information.

Register	Address	Description	Reset value
SBUF	0x99	Serial Buffer	00h
SCON	0x98	Serial Control	00h

Table 7 *Serial Port SFRs*

Register	Address	Description	RST value
T2CON	0xC8	Timer/Counter 2 control	00h
TCON	0x88	Timer/Counter 0 and 1 control	00h
TH0	0x8C	Timer/Counter 0 high byte	00h
TH1	0x8D	Timer/Counter 1 high byte	00h
TH2	0xCD	Timer/Counter 2 high byte	00h

CONFIDENTIAL

Downloaded From Oneyac.com

TL0	0x8A	Timer/Counter 0 low byte	00h
TL1	0x8B	Timer/Counter 1 low byte	00h
TL2	0xCC	Timer/Counter 2 low byte	00h
TMOD	0x89	Timer/Counter 0 and 1 mode control	00h
RCAP2H	0xCB	Timer 2 Reload/Capture high byte	00h
RCAP2L	0xCA	Timer 2 Reload/Capture low byte	00h
WDTRST	0xA6	WDT enable register	00h

Table 8 *Timers SFRs*

Register	Address	Description	RST value
CCMCON	0x8E	BIRD Communication Control	00h
CCMVAL	0x8F	BIRD Communication Value	00h
MMS	0x97	Reserved for emulation purpose	07h

Table 9 *BIRD SFRs*

5.3. Power management

For applications where power consumption is critical, the BK2535 provides all kinds of power saving modes.

5.3.1. Power Control Register

PCON2	7	6	5	4	3	2	1	0
0x87	SMOD	EUSB	CMD_RST	Latch_en	Deep_sleep	OSC32K_sel	RC32k_sel	IDLE

Table 10 power management register

SMOD: – Serial Port 0 baud rate doublers enable. When SMOD0=1, the baud rate for Serial Port 0 is doubled.

EUSB: R/W by software only. USB enable, the 48MHz clock will exist when EUSB=1.

CMD_RST: Write 1 to reset MCU (not include RF part).

Latch_en: this register used for deep sleep mode. In deep sleep mode, the power supply to digital part will be shut down, but the GPIO setting must be hold use this register.

Deep_sleep: System will enter deep sleep mode when setting this register. The lowest current consumption can be got by setting this register.

RC32k_sel: System will select RC32k clock when write 1 to this position. Interrupts and software can clear it.. In this state, the system clock changed to 32K RC clock, so the power consumption is very low.

CONFIDENTIAL

Downloaded From Oneyac.com

OSC32k_sel: System will select OSC32k (divided by OSC16M) clock when write 1 to this position. Interrupts and software can clear it. In this state, the system clock changed to 32K OSC clock, so the power consumption will decrease evidently.

Please note that OSC32k clock is more accurate than RC 32k clock, but need more power consumption.

IDLE: When set by software, system enter stop mode, and it can only be wake up by enabled interrupt.(Clear it to 0 by hardware). In this state, the most of clocks are shut down for power saving.

Note: set RC32k_sel and IDLE bit simultaneously can get the lowest power consumption, and in this state, all the register settings are retained.

5.3.2. Work State

5.3.2.1. Active Mode

Normally, the FLIP51 fetch the instruction from the program space and execute it step by step at the selected clock source, we call the state as active mode.

5.3.2.2. Idle Mode

An instruction that sets the IDLE bit (PCON2.0) causes the FLIP51 to enter idle mode when that instruction completes. In idle mode, CPU processing is suspended; internal registers maintain their current data. However, unlike the standard 8051, the clock is not disabled internally.

Activation of any enabled interrupt causes the hardware to clear the IDLE bit and terminate idle mode.

In idle mode, the power consumption is decreased evidently.

5.3.2.3. Sleep Mode

An instruction that sets the IDLE bit (PCON2.0) causes the FLIP51 to enter idle mode when that instruction completes. Also, you can decrease the power consumption to a lower level thanks for the register PCON2.1. When setting the register, the system clock changed from 16MHz to RC32KHz. We define this state as sleep mode.

After reset, the system will enter normal mode running at 16MHz immediately.

Note: You should always clear PCON2.6 to 0 to save current when USB module doesn't need work at any time.

As showed above, the IDLE bit decide CPU run or not, the PCON2.1 bit decide the system clock source. (16MHz or RC32KHz)

CONFIDENTIAL

Downloaded From Oneyac.com

5.3.2.4. Further Sleep Mode

To get the lower power consumption, another SFR register EXSLEEP can be use to set for this aim. You can set EXSLEEP[1] firstly, then enter sleep mode. We define this mode as further sleep mode. At this mode, you can get the lower current than sleep mode.

5.3.2.5. Supper Sleep Mode

Also, you can set EXSLEEP[0] firstly, then enter sleep mode. We define this mode as supper sleep mode. At this mode, you can get the lower current than further sleep mode.

Note: only the GPIO can be use to wake up the MCU in this sleep mode.

5.3.2.6. Deep Sleep Mode

If you want to get the lowest power consumption, you can let BK2535 enter deep sleep mode. Firstly, you should set all the GPIO to certain setting, then, set latch_en (PCON2 [4]) to latch all the register setting, lastly, set deep sleep bit (PCON2 [3]) to enter deep sleep status. In this state, the power supplied to digital will be shut down.

Note: after wake up from this state, the instruction will run from the address zero.

	current	Wakeup delay	Wake up source	condition	wake up mode	Note
Active			--	Run	-	
Standby		Fast	the enabled INT,GPIO	OSC16M+IDLE	Level or edge	
Sleep	4 uA	long	the enabled INT,GPIO	RC32K+IDLE	Level or edge	RC32K running
Further sleep	2.5 uA	longer	RTC timer, GPIO	RC32K+IDLE +further sleep	Level or edge	RC32K running
Super sleep	2 uA	Longest	GPIO	关闭 RC32K+IDLE +supper sleep	edge	RC32K closed
Deep sleep	0.6uA	Restart	GPIO	Off digital power	edge	

Table 11 power management register

Note: MCU would enter RTC interrupt process after waked up from the further or supper sleep modes, so you must open the RTC interrupt before enter these sleep modes.

CONFIDENTIAL

Downloaded From Oneyac.com

5.3.2.7. Wake Up

5.3.2.8. Wake Up from sleep mode

When the MCU entered IDLE/SLEEP mode, all the enabled GPIO ports and interrupt sources can be used to wake up the MCU separately. Configure the corresponding SFR bit can enable or disenable the wake up function.

PX_WKEN: port x wake up enable or disable 0: disable; 1: enable

PX_WKMOD: wake up mode setting. 0: low level trigger; 1: edge trigger

You can get the detail GPIO register address from SFR table part.

The process wake up from sleep mode is showed in next figure.

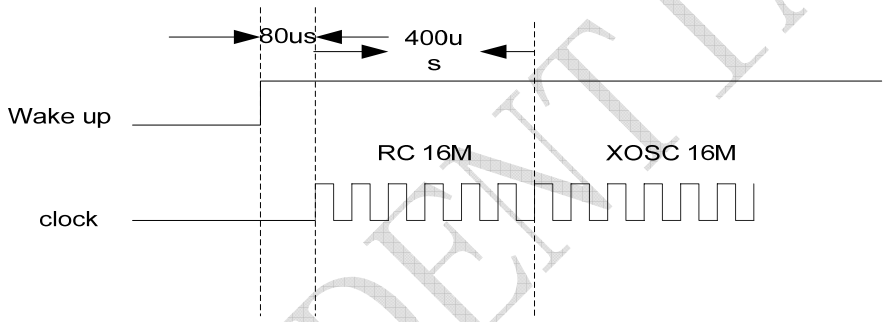


Figure 8 wake up process

After wake up, RC 16M clock will spend 80us to wake up, and after 400us, the clock source will switch to XOSC 16M automatically. RC 16M clock is not very accurate, so, during this period, you can only run ordinary MCU instruction, but cannot send or receive RF package.

Please note that: The RF part will also resume 120us for PLL locking after power up RF part. So, if you want to send/receive package through RF, you should wait 600us after wake up from sleep mode.

5.3.2.9. Wake up from deep sleep mode

All the ports can be set to wake up MCU from deep sleep status; also, you can enable or disable them separately. After wake up from deep sleep, a POR will be generated to reset the whole digital system.

5.4. Clock system

5.4.1. System clock topology

The BK2535 clock topology is showed as below. There are two clock sources, one is 16M, and the other is RC32k. You can select them by setting related register.

CONFIDENTIAL

Downloaded From Oneyac.com

PCON[2] used to select 32K clock for the system. CKCON used to select the divider number for 16M clock. 00:16M; 01:8M; 10:4M; 11:2M.

If the wake up time is a critical parameter for some application, a RC16M clock can be used before the OSC16M oscillating.

The clock source of ext timer is always fixed to 32k. In working mode, the OSC32K is used for counter, and in idle mode, the RC32k can be automatically selected. The ext timer is very suitable used for the application which has periodic behavior, such as mouse.

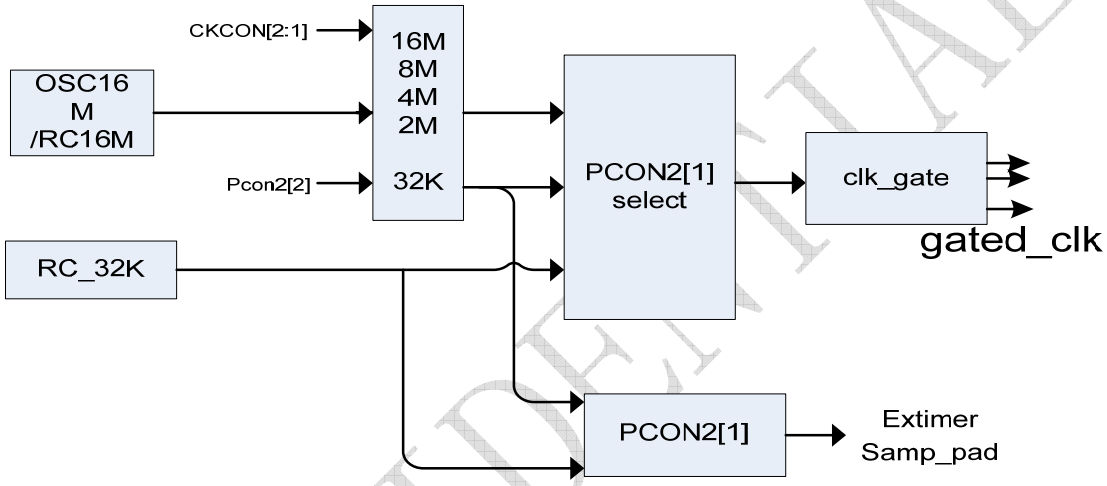


Figure 9 clock topology

5.4.2. Peripherals clock management

The peripherals clock source can be enabled or disenabled, to do this, you can refer to the next register. Evidently, the clock must be enabled when you want to use some peripheral equipment.

CLK_EN_CFG	7	6	5	4	3	2	1	0
0x85	**adc_en**	**timer_en**	**uart_en**	**pwm_en**	**Spi_en**	**i2c_en**	**Aes_mdu_en**	**WDT_en**

Table 12 clock enable register

CLK_EN_CFG: this register can use to power on or off all the peripheral equipment clocks for saving power.

Name	Description	value	defualt
adc_en	ADC clock enable or not	1: enable 0: disable	0
timer_en	TIMER 0 /1 /2 clock enable or not	1: enable 0: disable	0
uart_en	UART clock enable or not	1: enable 0: disable	0

CONFIDENTIAL

Downloaded From Oneyac.com

pwm_en	PWM clock enable or not	1: enable 0: disable	0
spi_en	SPI clock enable or not	1: enable 0: disable	0
i2c_en	I2C clock enable or not	1: enable 0: disable	0
Aes_mdu_en	AES and MUD clock enable or not	1: enable 0: disable	0
WDT_en	MDU clock enable or not	1: enable 0: disable	0

Table 13 CLK_EN_CFG register

CONFIDENTIAL

Downloaded From Oneyac.com

5.5. Reset system

There are three active low reset source in BK2535, they are power on reset, reset pin, watch dog reset. After reset, the MCU will re-start from address 0.

Also, a brown out circuit is integrated in BK2535 for detecting the supply voltage. Once the supply voltage decreased under 1.7V, a reset signal would be generated to reset the MCU to the initial state.

5.6. Interrupt system

The Flip8051 has the same interrupt sources as the original 80C51. These are handled the same as on the original 80C51, however the Flip8051 has a shorter interrupt latency period, and can distinguish shorter external interrupt pulses. The interrupt sources are sampled every clock cycle (clock rising edge), and the decision of whether an interrupt will be accepted takes place at the last clock cycle of each instruction execution, or every clock cycle during idle mode.

5.6.1. Introduction

When an enabled interrupt occurs, this operation branches to a subroutine and performs some service in response to the interrupt. When the subroutine completes, execution resumes at the point where the interrupt occurred. Interrupts may occur as a result of internal activity (e.g. timer0 overflow) or at the initiation of an external device (external interrupt pin). In any case, interrupt operation is programmed by the system designer, who determines the priority of interrupt service, compare to relative normal code execution or other interrupt service routines. All the interrupts may be enabled / disabled dynamically by the system designer except the TRAP (software) is non-maskable.

A typical interrupt process occurs as follow:

An interrupt event on the signal, connected to an input pin and sampled by the Flip8051, is registered into a flag buffer.

The priority of the flag is compared to the priority of the other interrupt by the interrupt controller. A higher priority causes the controller to set an interrupt flag.

The setting of the interrupt flag indicates to the control unit to execute a context switch. This context switch breaks the current instruction execution flow[1]. The control unit completes the current instruction execution prior to saving the two bytes of the program counter (PC) and reloads the PC with the interrupt vector address, which is the start address of a software service routine.

The software service routine performs the assigned tasks and executes a RETI instruction as a final instruction. This instruction signals the completion of the

Downloaded From Oneyac.com

interrupt, resets the interrupt-in-progress priority. The RETI instruction reloads the two bytes of the program counter and uses them as the 16-bit return address. Program execution then continues from the original point of interruption.

5.6.1.1. Interrupt source

The Flip8051 has one software interrupt, the TRAP instruction (always enabled) and up to fifteen interrupt sources controlled by hardware. Fifteen of these hardware interrupt are maskable interrupt sources. The maskable sources include two external interrupts (*int0_n* and *int1_n*), three timer interrupts (timers 0, 1, and 2), and one serial port (UART) interrupt. Depending on configuration, eight additional external interrupt (*intextra_n[7:0]*) are available and maskable.

Each interrupt (except TRAP and *intnmi*) has an interrupt request flag, which can be set by software as well as by hardware. For some interrupts, hardware clears the request flag when it grants an interrupt. Software can clear any request flag to cancel an impending interrupt.

For BK2535, the available interrupts are showed in the next table:

Interrupt Number	Interrupt Flag	Interrupt Source	Interrupt Routine Code Address
0	EX0	GPIO P1.0 Input	0x0003
1	ET0	Flip8051 Timer0 Interrupt	0x0013
2	EX1	GPIO P1.1 Input	0x000B
3	ET1	Flip8051 Timer1 Interrupt	0x001B
4	UART	RI+TI	0x0023
5	TF2+EXF2	Flip8051 Timer2 Interrupt	0x002B
6	TRAP	DEBUG system	0x0033
7	Intnmi	LBD interrupt	0x003B
8	EX2	SPI interrupt	0x0043
9	EX3	I2CM I2CS Interrupt	0x004B
10	EX4	USB Interrupt	0x0053
11	EX5	BK2401 Transceiver Interrupt	0x005B
12	EX6	External Timer Interrupt/GPIO wake up	0x0063
13	EX7	PCA Interrupt	0x006B
14	EX8	ADC Interrupt	0x0073
15	EX9	AES Interrupt	0x007B

Table 14 interrupt sources

Downloaded From Oneyac.com

5.6.1.2. Int0_n and int1_n

External interrupt *int0_n* and *int1_n* may be each programmed to be level-activated or transition-activated, depending on bits IT0 and IT1 in TCON register. External interrupts are enabled with bits EX0 and EX1 in IE register. Events on *int0_n* or *int1_n* set respectively the interrupt request flag IE0 or IE1 in TCON register. If the interrupt is transition-activated, the hardware jump to the service routine clears the request flag. Otherwise, if the interrupt is level activated, then the interrupt must be de-asserted before the end of the ISR.

External interrupt pins must be de-asserted for at least two clock cycles prior to a request. External interrupt inputs are sampled at each clock cycle. A level-triggered interrupt pin held low or high for any two clock cycles time period guarantees detection. Edge-triggered external interrupts must hold the request pin low for at least two clock cycles. This ensures edge recognition and sets interrupt request bit IEx. The CPU clears IEx automatically during service routine fetch cycles for edge-triggered interrupts.

External interrupt inputs *int0_n* and *int1_n* provide both the capability to exit from idle mode on low-level signal. GPIO description

TCON--Address: 0X88

BIT	7	6	5	4	3	2	1	0
FIELD	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
RESET	0x00							

Bit Number	Bit Mnemonic	Function
7	TF1	Timer 1 overflow flag Set by hardware when the timer 1 overflows. Cleared by hardware when the processor vectors to the interrupt routine
6	TR1	Timer 1 run control bit Set/cleared by software to turn timer 1 on/off
5	TF0	Timer 0 overflow flag Set by hardware when the timer 0 overflows. Cleared by hardware when the processor vectors to the interrupt routine
4	TR0	Timer 0 run control bit Set/cleared by software to turn timer 0 on/off
3	IE1	See interrupt system chapter, *int1_n* interrupt flag
2	IT1	See interrupt system chapter
1	IE0	See interrupt system chapter, *int0_n* interrupt flag
0	IT0	See interrupt system chapter

Table 15 Timer/counter control register (TCON low)

Downloaded From Oneyac.com

5.6.1.3. Intnmi interrupt

Intnmi is used for LBD interrupt.

ADDR	7	6	5	4	3	2	1	0
0XC9							Nmi_en	Nmi_flag
							Nmi interrupt enable	Nmi interrupt flag, need clear by software

Table 16 NMI Registers

5.6.1.4. Additional interrupts

This configuration requires the use of three new SFRs: Additional interrupt Flag register (AIF), Additional Interrupt Enable Register (AIE) and Additional Interrupt Priority Register (AIP).

Register	Address	Description	Reset value
AIE	0xE8	Additional interrupt enable	00h
AIF	0xC0	Additional interrupt flag	00h
AIP	0xF8	Additional interrupt priority	00h

Table 17 Additional interrupt registers

The additional external sources are level-activated for *intextra_n[5:0]* and transition-activated for *intextra_n [7:6].*

The flags that actually generate these interrupts are bits AIFj in Special Function Register AIF. When an external interrupt is generated, the flag that generated it is NOT cleared by hardware when the service routine is vectored to. This has to be done in the user's software.

All of the bits that generate interrupt (AIFj) can be set by software, with the same result as though it had been set by hardware. That is, interrupts can be generated in software.

Each of the additional external interrupt sources can be individually enabled or disabled by setting or clearing bit AIEj in Special Function Register AIE.

The interrupt global disable bit EA in IE register also disables the additional interrupts.

Like *int0_n* and *int0_n* inputs, *intextra_n* inputs are synchronized once on clock rising edge before internal use.

AIF--Address: 0XC0

BIT	7	6	5	4	3	2	1	0

Downloaded From Oneyac.com

FIELD	AIF7	AIF6	AIF5	AIF4	AIF3	AIF2	AIF1	AIF0
RESET	0x00							

Bit Number	Bit Mnemonic	Function
7:0	AIF	Additional Interrupt Flags: Set when respective Additional Interrupt detected. Must be cleared by software

Table 18 Additional interrupt flag register (AIF)

5.6.1.5. Timer Interrupts

Two timer-interrupt request bits (TF0 and TF1 in TCON register) are set by timer overflow (except Timer 0 in Mode 3). When a timer interrupt is generated, the bit is cleared by a hardware jump to an interrupt service routine. Timer interrupts are enabled by bits ET0, ET1, and ET2 in the IE register.

Timer 2 interrupts are generated by a logical OR of bits TF2 and EXF2 in register T2CON. Neither flag is cleared by a hardware jump to a service routine. In fact, the interrupt service routine must determine if TF2 or EXF2 generated the interrupt, and then clear the bit. Timer 2 interrupt is enabled by ET2 in register IE0.

NOTE: EXF2 is not available for P03(T2EX) is not exist.

5.6.1.6. Serial Port Interrupt

Serial port interrupts are generated by the logical OR of bits RI and TI in the SCON register. Neither flag is cleared by a hardware jump to the interrupt service routine. The service routine resolves RI or TI interrupt generation and clears the serial port request flag. The serial port interrupt is enabled by bit ES in the IE register.

5.6.1.7. TRAP interrupt

The function of TRAP instruction is like a software breakpoint, which is useful in software debug. The coding of this instruction is [0xA5]. By execution of the TRAP instruction, the Flip8051 generates an interrupt and executes the interrupt service routine at address 0x0033. It acts like the highest priority non-interruptible interrupt.

5.6.2. Interrupt enable

Each interrupt source (with the exception of TRAP) may be individually enabled or disabled by the appropriate interrupt enable bit in the IE register (or in the AIE register for additional interrupt sources). Note IE also contains a global disable bit (EA) that applies to all interrupts (except TRAP that is not maskable). If EA is set,

CONFIDENTIAL

Downloaded From Oneyac.com

interrupts are individually enabled or disabled by bits in IE. If EA is clear, all interrupts are disabled.

IE--Address: 0XA8

BIT	7	6	5	4	3	2	1	0
FIELD	EA	--	ET2	ES	ET1	EX1	ET0	EX0
RESET	0x00							

Bit Number	Bit Mnemonic	Function
7	EA	**Global Interrupt Enable** Clear to globally disable all Interrupt sources. Set to 1 to allow Individual interrupts to be enabled by their enable bits
6	--	Not used
5	ET2	**Timer2 Interrupt Enable** Set to enable Timer2 Interrupt. Cleared to disable Timer2 Interrupt
4	ES	**Serial Port Interrupt Enable** Set to enable Serial Port Interrupt. Cleared to disable Serial Port Interrupt
3	ET1	**Timer1 Interrupt Enable** Set to enable Timer1 Interrupt. Cleared to disable Timer1 Interrupt
2	EX1	**External Interrupt 1 enable** Set to enable External Interrupt 1. Cleared to disable External Interrupt 1.
1	ET0	**Timer0 Interrupt Enable** Set to enable Timer0 Interrupt. Cleared to disable Timer0 Interrupt
0	EX0	**External Interrupt 0 enable** Set to enable External Interrupt 0. Cleared to disable External Interrupt 0.

Table 19 Interrupt Enable register (IE)

AIE--Address: 0XE8

BIT	7	6	5	4	3	2	1	0
FIELD	AIE7	AIE6	AIE5	AIE4	AIE3	AIE2	AIE1	AIE0
RESET	0x00							

Bit Number	Bit Mnemonic	Function
7:0	AIE	Additional interrupt Enable Set to enable respective Additional Interrupt. Cleared to disable respective Additional interrupt.

Table 20 Additional Interrupt Enable register (AIE)

Downloaded From Oneyac.com

5.6.3. Interrupt priority

Each of the hardware interrupt sources may be individually programmed to high or low priority levels (except the NMI input and the TRAP, which have a higher priority level). This is accomplished by clearing/setting the corresponding bit in the Interrupt Priority registers (IP or AIP)

The TRAP instruction is the highest priority level interrupt. A TRAP cannot be interrupted by any other interrupt source including the TRAP. A low-priority interrupt can be itself interrupted by a higher priority level interrupt, but not by another lower or equal priority interrupts. Higher priority level interrupts are serviced before lower priority interrupts.

Interrupt source	Interrupt flag	Priority level	Vector Addresses	Cleared by hardware (H) or by software (S)
TRAP	-	3 (highest - not interruptible)	0x0033	-
Intnmi	-	2	0x003B	-
int0_n	IE0	0 or 1	0x0003	H if edge
Timer 0	TF0	0 or 1	0x000B	H
int1_n	IE1	0 or 1	0x0013	H if edge
Timer 1	TF1	0 or 1	0x001B	H
UART	RI+TI	0 or 1	0x0023	S
Timer2	TF2+EXF2	0 or 1	0x002B	S
Intextra_n[0]	AIF0	0 or 1	0x0043	S
Intextra_n[1]	AIF1	0 or 1	0x004B	S
Intextra_n[2]	AIF2	0 or 1	0x0053	S
Intextra_n[3]	AIF3	0 or 1	0x005B	S
Intextra_n[4]	AIF4	0 or 1	0x0063	S
Intextra_n[5]	AIF5	0 or 1	0x006B	S
Intextra_n[6]	AIF6	0 or 1	0x0073	S
Intextra_n[7]	AIF7	0 or 1	0x007B	S

Table 21 Interrupt priority levels and vector addresses

If two interrupt requests with the same priority level (0 or 1) are received simultaneously, an internal polling sequence determines which request is serviced, according to the table below:

Downloaded From Oneyac.com

Interrupt source	Interrupt flag	Servicing priority order	Inter source
int0_n	IE0	1 (highest)	GPIO P1.0 Input
Timer 0	TF0	2	Flip8051 Timer0 Interrupt
int1_n	IE1	3	GPIO P1.1 Input
Timer 1	TF1	4	Flip8051 Timer1 Interrupt
UART	RI+TI	5	RI+TI
Timer2	TF2+EXF2	6	Flip8051 Timer2 Interrupt
Intextra_n[0]	AIF0	7	SPI interrupt
Intextra_n[1]	AIF1	8	I2CM I2CS Interrupt
Intextra_n[2]	AIF2	9	USB Interrupt
Intextra_n[3]	AIF3	10	BK2401 Transceiver Interrupt
Intextra_n[4]	AIF4	11	External Timer Interrupt/GPIO wake up
Intextra_n[5]	AIF5	12	PCA Interrupt
Intextra_n[6]	AIF6	13	ADC Interrupt
Intextra_n[7]	AIF7	14 (lowest)	AES Interrupt

Table 22 Interrupt priority within a same priority level (0 or 1)

NOTE: some interrupts are not available for BK2535, please refer to the interrupt source for detail.

AIE--Address: 0XB8

BIT	7	6	5	4	3	2	1	0
FIELD	--	--	PT2	PS	PT1	PX1	PT0	PX0
RESET	0x00							

Bit Number	Bit Mnemonic	Function
7:6	--	Not used
5	PT2	**Priority of timer 2 Interrupt:** Timer 2 Interrupt priority is determined by default priority order when cleared to 0. Timer 2 Interrupts set to high priority level when set to 1.
4	PS	**Serial Port Interrupt priority level.** UART interrupt priority determined by default priority order when cleared to 0. UART interrupts set to high priority level when set to 1

Downloaded From Oneyac.com

3	PT1	**Timer1 overflow Interrupt priority level** Timer 1 Interrupt priority determined by default priority order when cleared to 0 Timer 1 Interrupts set to high priority level when set to 1.
2	PX1	**External Interrupt 1 priority level** External Interrupt 1 priority determined by default priority order when cleared to 0. External Interrupt 1 set to high priority level when set to 1.
1	PT0	**Timer0 overflow Interrupt priority level** Timer 0 Interrupt priority determined by default priority order when cleared to 0. Timer 0 interrupt set to high priority level when set to 1.
0	PX0	**External Interrupt priority level** External Interrupt 0 priority determined by default priority order when cleared to 0. External Interrupt 0 set to high priority level when set to 1.

Table 23 Interrupt Priority Register (IP)

AIP--Address: 0XF8

BIT	7	6	5	4	3	2	1	0
FIELD	AIP7	AIP6	AIP5	AIP4	AIP3	AIP2	AIP1	AIP0
RESET	0x00							

Bit Number	Bit Mnemonic	Function
7:0	AIP	Additional Interrupt priority level Respective Additional interrupt set to high priority level when set to 1

Table 24 Additional Interrupt Priority Register (AIP)

5.6.4. Interrupt blocking conditions

If all enable and priority requirements have been met, a single prioritized interrupt request at a time branches to an interrupt service routine. There are 3 causes of blocking conditions with hardware-generated interrupt request:

1. An interrupt of equal or higher priority level is already in progress (defined as any point after the flag has been set and the RETI of the ISR has not executed).

2. The current polling cycle is not the final cycle of the instruction in progress.

3. The instruction in progress is RETI or any write to the IE, IP, AIE or AIP registers.

Any of these conditions blocks calls to interrupt service routines. Condition 2 ensures the instruction in progress completes before the system vectors to the ISR. Condition 3 ensures at least one more instruction executes before the system vectors to interrupts if the instruction in progress is a RETI or any write to

Downloaded From Oneyac.com

an interrupt control registers.

: *If the interrupt flag for a level-triggered external interrupt is set but denied for one of the above conditions and is clear when the blocking condition is removed, then the denied interrupt is ignored. In other words, blocked interrupt requests are not buffered for retention.*

6. Peripheral module

6.1. OVERVIEW

BK2535 have various peripheral devices which can be used for different applications.

6.2. UART

6.2.1. Serial port overview

The Flip8051 provides a standard serial communication interface (UART). The Serial Port uses the signals Serial In and Serial Out to receive and transmit serial data. The modes of operation and baud rate generation are the same as the original 80C51. The serial interface in the Flip8051 supports all operation modes, as in standard 80C51.

6.2.2. Operation mode

6.2.2.1. Mode 0 (synchronous mode, half duplex)

Not supported for BK2535.

6.2.2.2. Mode 1 (asynchronous mode, full duplex)

In Mode 1, data is transmitted through *serial out* signal and received through *serial in* signal. The data is composed of 10 bits: starting with a start bit “0”, then followed by 8 data bits (LSB first, MSB last), and then the stop bit “1”. The Baud Rate in Mode 1 is controlled by Timer1 or Timer2 and is programmable. Please refer to Programming the Baud Rate, in later part of this chapter for details. To select the mode 1, clear SCON.SM0 and set SCON.SM1.

- **Transmission**

To send out data, clear the SCON.REN bit and write the data into the SBUF special function register. The data will then be shifted out (LSB first, MSB last), at the *serial out* pin.

CONFIDENTIAL

Downloaded From Oneyac.com

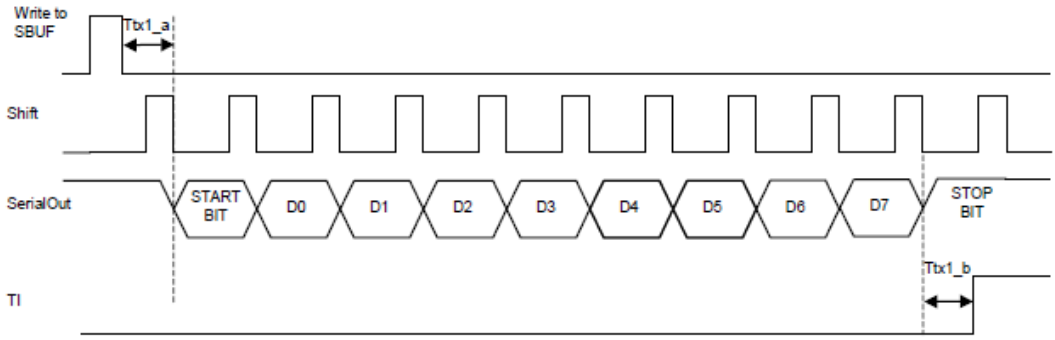


Figure 10 *Serial Transmit Mode 1*

- **Reception**

To receive data, set the SCON.REN bit and clear the SCON.RI, this will enable the receive function. When received the data value can be read from the SBUF special function register.

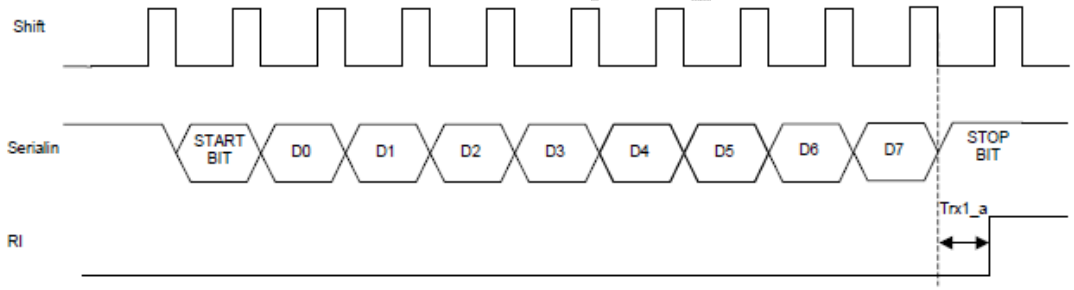


Figure 11 *Serial receive Mode 1*

6.2.2.3. Mode 2 (asynchronous mode, full duplex)

In Mode 1, data is transmitted through *serial out* signal and received through *serial in* signal. The data is composed of 11 bits: 1 start bit, 8 data bits, 1 TB8 bit (in SCON) and the stop bit. The extra TB8 bit is for use in a multiprocessor communication environment. When multiprocessor communication support is not needed, this bit can also be used as a parity bit. The data transfer rate in Mode 2 is fixed as clock/32 or clock/64. Timer 1 and Timer 2 are independent of the Baud Rate generation and can be used for other purposes. To select the mode 2, set SCON.SM0 and clear SCON.SM1.

Transmission

To send out data, clear the SCON.REN bit and write the data into the SBUF special function register. The data will then be shifted out (LSB first, MSB last), at the *serial out* pin.

Downloaded From Oneyac.com

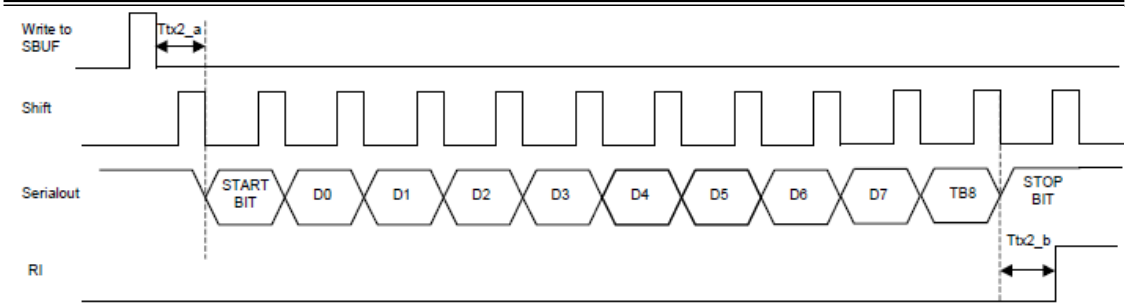


Figure 12 *Serial Transmit Mode 2*

Reception

To receive data, set the SCON.REN bit and clear the SCON.RI, this will enable the receive function. When received the data value can be read from the SBUF special function register.

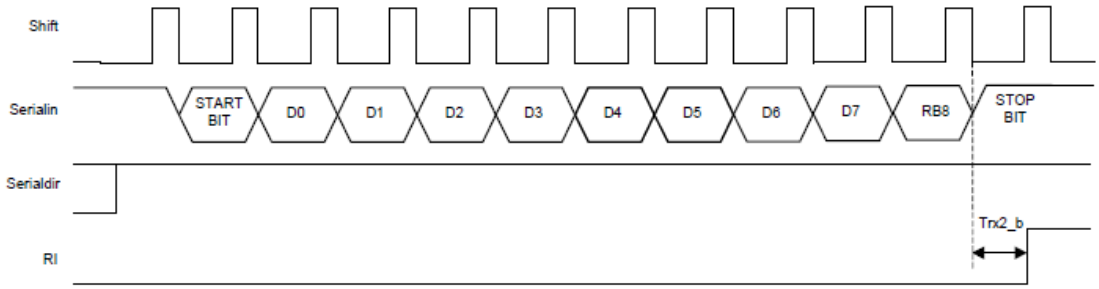


Figure 13 *Serial receive Mode 2*

6.2.2.4. Mode 3 (asynchronous mode, full duplex)

The operation of Mode 3 is same as Mode 2. The only difference is that Timer1 (or Timer 2) controls the Baud Rate. Serial Mode 3 has the same timing diagram as Mode 2 (above), but the source of the shift pulse is different. To select the mode 1, set SCON.SM0 and SCON.SM1.

6.2.3. Programming the Baud Rate

6.2.3.1. Mode 0

Not available for BK2535.

6.2.3.2. Modes 1 & 3 - Timer1 generating Baud Rate

Timer 1 generates the Receive Clock when T2CON.RCLK=0 and the Transmit Clock when T2CON.TCLK=0, (or always in the Flip8051 without the Timer2). Timer1 should be set up in timer auto-reload mode.

Baud Rate = ((PCON2.SMOD+1)*clock)/(32*12*(256-TH1))

Given a baud rate, the reload value for TH1 is

Downloaded From Oneyac.com

TH1 = (256 - (PCON2.SMOD+1)*clock)/(384*Baud Rate)

If TH1 is not an integer value then either the Baud Rate or clock frequency must be changed.

6.2.3.3. Modes 1 & 3 - Timer2 generating Baud Rate

Timer 2 can generate the Receive Clock in the Flip8051, when T2CON.RCLK=1 and the Transmit Clock when T2CON.TCLK=1. If Timer2 is being clocked internally,

Baud Rate = clock /(32*(65536-(RCAP2H,RCAP2L)))

The reload value for RCAP2H, RCAP2L is given by

RCAP2H, RCAP2L = 65536 - clock /(32*Baud Rate)

Otherwise if Timer2 is being clocked by the Timer2 signal, Baud Rate = Timer2 Overflow rate/16.

6.2.3.4. Mode 2

In serial mode 2 the Baud Rate is fixed to (PCON2.SMOD +1)/64.

6.2.4. Serial port registers

The serial port uses two SFR registers.

Register	Address	Description	Reset value
SCON	0x98	Serial Control	00h
SBUF	0x99	Serial Buffer	00h

Table 25 Serial Port registers

SCON—address: 0x98

BIT	7	6	5	4	3	2	1	0
FIELD	SM7	SM 6	SM 5	SM 4	SM 3	SM 2	SM 1	SM 0
Reset value	0x00							

Bit Number	Bit Mnemonic	Function
7:6	SM0, SM1	**Serial port mode bit** **SM0** / **SM1** / **Mode** / **Description** / **Baud rate** 0 / 0 / 0 / Shift register / Clk/12 0 / 1 / 1 / 8 bit UART / Variable 1 / 0 / 2 / 9 bit UART / Clk/32 or Clk/64 1 / 1 / 3 / 9 bit UART / Variable
5	SM2	**Serial port mode bit 2** If set in serial modes 2 or 3, RI is only activated if the 9th received

CONFIDENTIAL

Downloaded From Oneyac.com

		data bit (RB8) is 1. If set in serial mode 1, then RI is only activated if a valid stop bit is received.
4	REN	**Receiver Enable Bit** Set for reception, clear for transmission
3	TB8	**Transmit bit 8** In serial modes 2 and 3, the 9th data bit transmitted
2	RB8	**Receiver bit 8** In serial modes 2 and 3, the 9th data bit received.
1	TI	**Transmit interrupt flag** Set at the beginning of the stop bit, or at the end of the 8th bit time in mode 0. Cleared by software.
0	RI	**Receive interrupt flag** Set halfway through the stop bit, or at the end of the 8th bit time in mode 0. Cleared by software.

Table 26 Serial Port control register (SCON)

CONFIDENTIAL

Downloaded From Oneyac.com

6.3. ADC

6.3.1. introduction

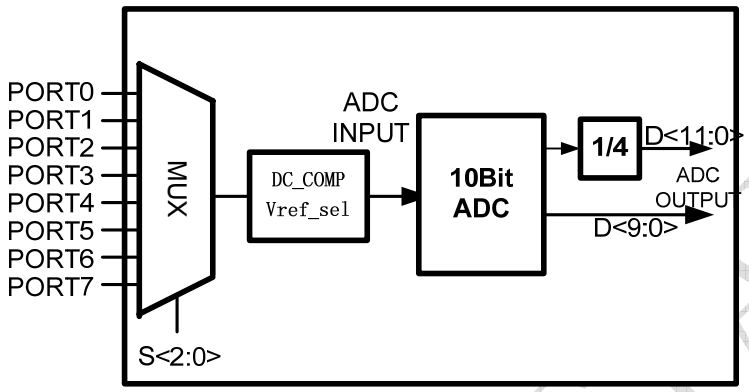


Figure 14 ADC block

A 10bits/12bits SAR ADC is integrated in BK2535. Total 8 channels can be selected used for ADC transfer. The ADC supports continue mode and single transfer mode, and the sample rate can be 1kHz to 32kHz. In single transfer mode, it will generate interrupt every time after transform. The input of ADC is share with P3 general I/O port.

In single transfer mode, the time used to convert is very little.

Convert time < 30us(single mode) → Convert Done),

The ADC register located at the XRAM space, the basic address is 0X920.

6.3.2. Register explain

ADDR	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0x0922	adc_mode		adc_chnn			intr_en	adc_ch_en	ready
adc_mode:=00, power down mode adc_mode:=01, single mode adc_mode:=10, soft mode adc_mode:=10, continue mode; 12bitsmode(filt_mode=1) is only effective for this mode. adc_chnn：eight channel corresponding to GPIO3.0- GPIO3.7 intr_en: generate interrupt or not to MCU adc_ch_en: ADC channel enable. If no GPIO port used to ADC transfer, this bit should be set to 0. ready： When the transfer is done, the bit will be set to zero. After read, it								

Downloaded From Oneyac.com

will be set to 1 automatically.

Table 27 ADC register0

ADDR	adc_dataL[7:0]
0x0920	adc_data low 8 bits

Table 28 ADC register1

ADDR	[7]	[6:4]	[3:0]
0x0921	adc_setting	pre_divid[2:0]	adc_dataH
adc_setting: the setting time for ADC after power on, 1：40us。 0：20us pre_divid[2:0]: clock divider (the number should be fixed as 0x01) adc_dataH：the higher 4 bits for adc_dat. high resolution mode:{adc_dataH[3:0],adc_dataL[7:0]}, normal mode: {adc_dataH[1:0],adc_dataL[7:0]}.			

Table 29 ADC register2

ADDR	*adc_rate*
0x0923	*adc_rate low 8 bits*

Table 30 ADC register3

ADDR	[7]	[6:5]	[4:0]
0x0924	High_res_mode	adc_dly[1：0]	adc_rate[12:8]
High_res_mode: 12 bits mode or 10 bits mode adc_dly: the valid sample for the first conversion. please set as 2 or 3 for this register. adc_rate： the high 5 bits for adc_rate			

Table 31 ADC register4

6.3.3. Sample rate:

Given an ADC sample rate, you can calculate the adc_rate value as below:

ADC sample = system_clk/((pre_divid+1)*(adc_rate+1))

ADC sample = system_clk/((pre_divid+1)*(adc_rate+1))/4 High_res_mode

CONFIDENTIAL

Downloaded From Oneyac.com

adc_rate = system_clk/((pre_divid+1)*(ADC sample))-1

= system_clk/(2*(ADC sample)) -1

Note1: the sample rate should be not greater than 85k for 10bits mode.

Note2: the sample rate will decrease 4 times for high resolution mode.

Note3: the pre_divid should be set as 1.

Note4: in continue mode, the sample rate is fixed in spite of read or not by MCU. In software mode, ADC will enter waiting state until the result is read by MCU.

6.3.4. ADC usage

The reference voltage can be set as 1.2V or Vdd/2 for different application. Also, you can decide whether add DC compensation for improving the negative input voltage.

BANK1		
Reg07[24]	Select the reference voltage. 0: 1.2V; 1: VDD/2	
Reg07[25]	Compensate the DC or not. 0: not compensate; 1: yes	

Table 32 ADC analog register

CONFIDENTIAL

Downloaded From Oneyac.com

6.4. PWM

6.4.1. OVERVIEW

The PWM peripheral is an additional peripheral. The PWM is connected to the Flip8051 through the external RAM interface. The Pulse Width Modulation can be used in several kinds of applications. Typically, the PWM can be used to drive DC motors in automotive applications, to generate DTMF in telecom applications or to generate AM radio quality equivalent audio signals.

6.4.2. FUNCTIONAL DESCRIPTION

The PWM generates pulses of programmable length and period. To set up the duty cycle, four registers are needed (depending on the resolution mode): one control register PWMC, one register for the resolution, one register for the duty cycle PWMDCLSB and in the case of high resolution, one other register for the duty cycle PWMDCMSB. The PWM can operate in two modes:

- High-resolution mode (10 bits): registers PWMDCLSB and PWMDCMSB are used.
- Standard-resolution mode (8 bits): the register PWMDCMSB is not used.

When operating in the standard-resolution mode, only the PWMDCLSB is taken into account.

When the resolution for the application is decided, it's advised not to change it again.

The PWM address is show as below; the basic address is 0XA00 in external RAM space.

ADDRESS	PWM_CTRL	PWM_DCLSB	PWM_DCMSB	PWM_RESOLUTION
PWM0	0X A2	0XA3	0XA4	0XCE
PWM1	0XDD	0XDE	0XDF	0XCF

Table 33 PWM register address

All the five PWM have the same operation. Next, we will describe the detail usage of PWM0.

register[bit]	Control name	ADDR	Operation	Function
[7]	pwm0_dcresol	0XA00	R/W	PWM0 Duty Cycle Resolution 1'b0 => standard resolution 1'b1 => high resolution
[6]	en_pwm0	0XA00	R/W	Enable PWM0

Downloaded From Oneyac.com

[5:0]	pwm0_prescaler	0XA00	R/W	PWM0 prescaler
[7:0]	pwm0_dclsb	0XA01	R/W	PWM0 Duty Cycle LSB
[7:0]	pwm0_dcmsb	0XA02	R/W	PWM0 Duty Cycle MSB
[7:0]	pwm0_resol	0XA03	R/W	PWM0 resolution

Table 34 PWM register summarize

pwm0_dcresol: Duty Cycle Resolution.

This bit is used to select the duty cycle resolution. It is advised to set the resolution only once, at the beginning of the application and not to change it after.

0 = the 8-bit resolution mode is selected (Standard resolution).

1 = the 10-bit resolution mode is selected (High resolution).

ENPWM: Enable Pulse Width Modulation.

This bit controls the pulse width modulation output. While this bit is low, the output is disabled.

0 = The PWM output is disabled.

1 = The PWM output is enabled.

When the bit ENPWM is cleared, the user can change the prescaler, and then the period of the pulse width modulation output is modified. The period can be changed at each cycle of clock.

The way to configure the output period is:

Disable the bit ENPWM

Write the value of the prescaler (bits 5 down to 0 of the register PWMCTRL)

Enable the bit ENPWM (bit 6 of the register PWMCTRL).

PWMPESCALER: Pulse Width Modulation Prescaler.

This field is used to set the repetition rate of the square wave available at output PWM. The frequency of this square wave is given by the following formula:

PWM_resoluation: the 8 bit register decide the stop counter of the PWM. It can be used to adjust the resolution of PWM.

PWMDCLSB and **PWMDCMSB** registers are used to set the duty cycle of the square wave generated. These registers are constantly compared to an internal counter. The size of this counter is function of the resolution (8 or 10 bits). This gives a pulse width modulation in the range of 0/(1~255) to 255/(1~255) for the

CONFIDENTIAL

Downloaded From Oneyac.com

standard-resolution and 0/(769~1023) to 1023/(769~1023)for the high-resolution.

The PWMDC value indicates the duration of the high level:

If PWMDC=all zeros, PWMOUT stays low.

If PWMDC=all ones, PWMOUT stays high.

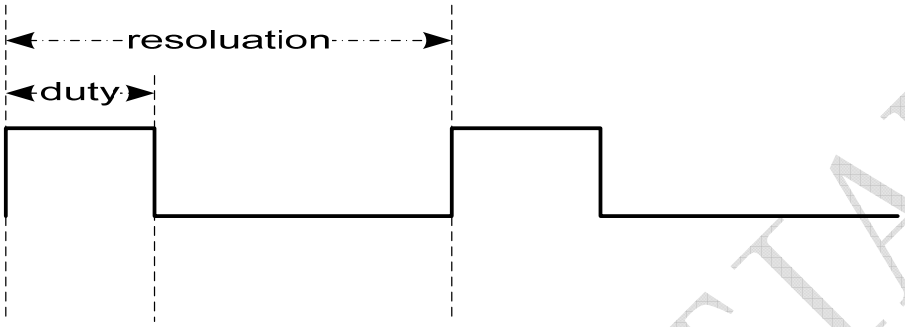


Figure 15 *PWM Parameter*

6.4.3. Frequency of PWM

The frequency of PWM can be calculated by the next formula.

$$F_{pwm} = \frac{F_{osc}}{2 * (PWMP + 1) * pwm\_resol} \quad (Standard\ resolution\ mode)$$

$$F_{pwm} = \frac{F_{osc}}{2 * (PWMP + 1) * (pwm\_resol + 768)} \quad (High\ resolution\ mode)$$

NOTE1: the default value of pwm_resol is 255, it cannot be set as zero. The value is preferred be set more than127.

NOTE2: PWMPESCALER cannot be set as zero, or overflow will be happened.

NOTE3: The duty cycle is set dynamically. During a period, it is possible to change the duty cycle.

6.5. I2C

6.5.1. I2C master

6.5.1.1. Overview

The Inter-Integrated Circuit (I2C) master controller is a simple bi-directional 2-wire bus, which provides an interface between BK2535 and an I2C bus.

The I2C mater handles all functions necessary to establish and maintain data link:

- Fast and standard transfer rates.

CONFIDENTIAL

Downloaded From Oneyac.com

- 7-bit addressing on I2C.
- Simple master operations.
- Clock Stretching and Wait State generation.
- Operates from a wide range of input frequencies.
- Interrupt generation.
- Fully synthesizable, static synchronous design.

Received and Transmit Data are stored respectively in Receive Buffer and Transmit Buffer. Only one byte at a time can be stored in each buffer. During an I2C transaction, the CPU needs to read regularly the Receive Buffer and to write regularly the Transmit Buffer.

NOTE: Arbitration for multi-master use is not supported by BK2535.

6.5.1.2. List of I2CM register

Mnemonic	Name	Address
MCON	I2CM Control register	S:0E1h
MRXBUF	I2CM Reception buffer	S:0E2h
MTXBUF	I2CM Transmission Buffer	S:0E3h
MPRESC	I2CM Pre-scalar clock register	S:0E4h
MSTAT0	I2CM Status register 0	S:0E5h
MSTAT1	I2CM Status register 1	S:0E6h
MIEN0	I2CM Interrupt Enable register 0	S:0E7h
MIEN1	I2CM Interrupt Enable register 1	S:0D2h
MCADDR	I2CM Call Address register	S:0D4h

Table 35 I2CM register

MCON (S:E1h) I2CM Control Register

BIT	7	6	5	4	3	2	1	0
FIELD	--	--	WAIT	--	STOP	SRST	STA	BUSY
RESET	0000							

Bit Number	Bit Mnemonic	Function
7	--	**Reserved** The value read from this bit is indeterminate.
6		**Reserved** The value read from this bit is indeterminate.
5	WAIT	**Wait state mode** '1': Generate wait state on SCL line when RX overflows. '0': Send "Not Acknowledge" to stop the transmission when RX
4	--	**Reserved** The value read from this bit is 0.

CONFIDENTIAL

Downloaded From Oneyac.com

3	STOP	**Generate Stop condition** When this bit is set, the current byte ends normally and a STOP condition is generated just after the acknowledge cycle. This bit is automatically cleared by the controller when the STOP
2	SRST	**Software reset** This bit is automatically cleared once IDLE state is reached.
1	STA	**Generate Start condition** This bit is automatically cleared by the controller when the transmission has begun or if an error is detected.
0	BUSY	**BUSY flag** This bit is set to '1' when an I2C frame transfer is in progress on

Table 36 I2CM Control Register (MCON)

MRXBUF (S:E2h) Read only I2CM Receive Buffer

Bit Number	Bit Mnemonic	Function
7:0	RXBUF	**Data received by I2CM**

Table 37 I2CM Receive Register (MRXBUF)

MTXBUF (S:E3h) Write only I2CM Transmit Buffer

Bit Number	Bit Mnemonic	Function
7:0	TXBUF	**Data transmitted by I2CM**

Table 38 I2CM Transmit Buffer (MTXBUF)

MPRESC (S:E4h) I2CM Clock Prescaler Register

MPRESC register enables to generate *OSCL* output from a large range of CLK frequency.

Bit Number	Bit Mnemonic	Function	Default value
7:0	PRESC	**Clock pre scalar register** Fscl = *Fclk*/10*(1+*PRESC)*	**'h00**

Table 39 I2CM MPRESC Register

Note: This register should not be written during a transmission.

MSTAT0 (S:E5h) I2CM Status Register 0

Bit Number	Bit Mnemonic	Function
7	--	**Reserved** The value read from this bit is indeterminate.
6	--	**Reserved** The value read from this bit is indeterminate.

CONFIDENTIAL

Downloaded From Oneyac.com

5	--	**Reserved** The value read from this bit is indeterminate.
4	DNA	**Data byte not acknowledged** Data byte not acknowledged during transmission. Stop condition
3	SANA	**Slave Address Not Acknowledged** Slave Address not acknowledged. Stop condition sent.
2	UNF	**Under Flow** Transmit Data Byte not ready (Transmit Buffer is empty) while a new data byte needs to be sent. A STOP condition is sent.
1	OVF	**Receive Overflow** Received Data Byte could not be written (Receive Buffer is full) while a new byte was received.
0	NEND	**Normal End (End of access with no error)** Set when a stop is sent at the end of a successful access. Clear automatically when a new I2C

Table 40 I2CM Status Register 0 (MSTAT0)

These interrupt sources are automatically cleared after a read access to this register.

When DNA, SANA, UNF or OVF flags have been set, reception and transmission processes are disabled until the CPU has read MSTAT0 register. This read operation automatically resets MSTAT0 register and MCON.STA bit, if one of these error bits is set. If this read operation is performed while no error bit is set, MCON.STA bit is not cleared.

These interrupt sources can all be individually enabled/disabled by MIEN0 register.

MSTAT1 (S:E6h) Read only I2CM Status Register 1

Bit Number	Bit Mnemonic	Function
7	--	**Reserved** The value read from this bit is indeterminate.
6	--	**Reserved** The value read from this bit is indeterminate.
5	TBE	**Transmission buffer is empty** **'1'**: Transmit Buffer empty. This flag is cleared when the CPU performs a write access to MTXBUF register.
4	--	**Reserved** The value read from this bit is 0.
3	TBF	**Transmission buffer is full** **'1'**: Transmit Buffer full. No more write operation into transmit buffer or memory is performed (CPU write request to TXDATA not taken in account). This flag is cleared when a new Data Byte is requested by the TXRX controller. **'0'**: Transmit Buffer is empty

CONFIDENTIAL

Downloaded From Oneyac.com

2	RBE	**Reception buffer is empty** **'1'**: Receive Buffer empty. No more write operation into receive buffer or memory is performed (CPU read request to RXDATA not taken in account). This flag is cleared when a new Data Byte is received by the TXRX controller.
1	--	**Reserved** The value read from this bit is indeterminate.
0	RBF	**Reception buffer is full** **'1'**: Receive Buffer full. This flag is cleared when the CPU performs a read operation to MRXBUF register.

Table 41 I2CM Status Register 1 (MSTAT1)

These interrupt sources can all be individually enabled/disabled by MIEN1 register.

MIEN0 (S:E7h) I2CM Interrupt Enable Register 0

Bit Number	Bit Mnemonic	Function
7	--	**Reserved** The value read from this bit is indeterminate.
6	--	**Reserved** The value read from this bit is indeterminate.
5	--	**Reserved** The value read from this bit is indeterminate.
4	EDNA	**Data byte Not Acknowledged Interrupt enable bit** Clear to disable MSTAT0.DNA bit to generate an interrupt request Set to enable MSTAT0.DNA bit to generate an interrupt request
3	ESANA	**Slave Address Not Acknowledged Interrupt enable bit** Clear to disable MSTAT0.SANA bit to generate an interrupt request Set to enable MSTAT0.SANA bit to generate an interrupt request
2	EUNF	**Underflow Interrupt enable bit** Clear to disable MSTAT0.UNF bit to generate an interrupt request Set to enable MSTAT0.UNF bit to generate an interrupt request
1	EOVF	**Overflow Interrupt enable bit** Clear to disable MSTAT0.OVF bit to generate an interrupt request Set to enable MSTAT0.OVF bit to generate an interrupt request
0	ENEND	**Normal End Interrupt enable bit** Clear to disable MSTAT0.NEND bit to generate an interrupt request Set to enable MSTAT0.NEND bit to generate an interrupt request

Table 42 I2CM Interrupt Enable register 0 (MIEN0)

MIEN1 (S:D2h) I2CM Interrupt Enable Register 1

Bit Number	Bit Mnemonic	Function
7	--	**Reserved** The value read from this bit is indeterminate.
6	--	**Reserved** The value read from this bit is indeterminate.

CONFIDENTIAL

Downloaded From Oneyac.com

5	ETBE	**Transmission Buffer Empty Interrupt enable bit** Clear to disable MSTAT1.TBE bit to generate an interrupt request Set to enable MSTAT1.TBE bit to generate an interrupt request
4	--	**Reserved** The value read from this bit is indeterminate.
3	ETBF	**Transmission Buffer Full Interrupt enable bit** Clear to disable MSTAT1.TBF bit to generate an interrupt request Set to enable MSTAT1.TBF bit to generate an interrupt request
2	ERBE	**Reception Buffer Empty Interrupt enable bit** Clear to disable MSTAT1.RBE bit to generate an interrupt request Set to enable MSTAT1.RBE bit to generate an interrupt request
1	--	**Reserved** The value read from this bit is indeterminate.
0	ERBF	**Reception Buffer Full Interrupt enable bit** Clear to disable MSTAT1.RBF bit to generate an interrupt request Set to enable MSTAT1.RBF bit to generate an interrupt request

Table 43 I2CM Interrupt Enable register 1 (MIEN1)

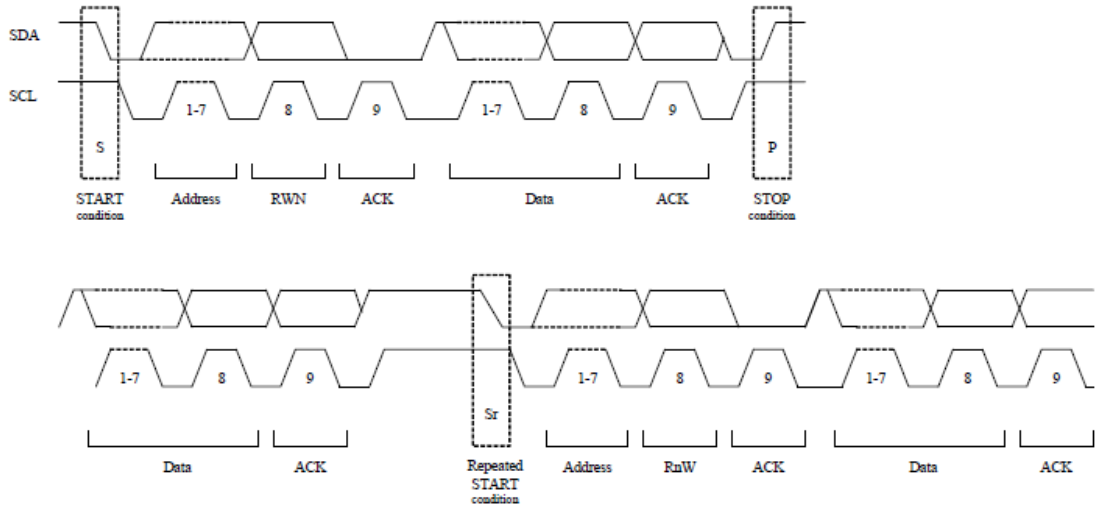
MCADDR (S:D4h)I2CM Call Address Register

Bit Number	Bit Mnemonic	Function
7	RWN	**Read/write control bit for I2C transaction** Set to read data from addressed slave device Clear to write data to the addressed slave
6:0	CADDR	**7-bit Call Address** This register must be written before the beginning of an I2C

Table 44 I2CM Call address register (MCADDR)

6.5.1.3. I2C frame data format

This I2C master controller only support 7-bit format as shown on the figure below:



START Condition : A HIGH to LOW transition on the SDA line while SCL is HIGH.
STOP Condition : A LOW to HIGH transition on the SDA line while SCL is HIGH.
DATA validity : The data on the SDA line must be stable during the HIGH period of SCL. The state of the data line can only change when SCL is LOW.

Figure 16 *Complete data transfer*

Downloaded From Oneyac.com

All words put on the SDA line are 8-bits long

Each byte is followed by an acknowledge bit set by receiver. Data is transferred with the most significant bit (MSB) first.

After the Start condition (S), a slave address is sent. This address is 7 bits long. The eighth bit determines the direction of the message (R/WN): a '0' means that the master will write data to a selected slave, a '1' means that the master will read data from a selected slave.

A data transfer is always terminated by Stop condition (P). However, if the master still wishes to communicate on the bus, it can generate a Repeated Start (Sr) that this to say to generate another START without first generating a STOP. Various combinations of read/write formats are then possible within such a transfer.

Note that two groups of eight addresses (0000XXX and 1111XXX) are reserved for purposed shown in the following table.

CALL ADDRESS	RWN Bit	Description
0000 000	0	General call address. It is used to address every device connected to I2C-bus.
0000 000	1	START byte(1)
0000 001	X	CBUS address (2).
0000 010	X	Reserved for different bus format.
0000 011	X	Reserved for future purposes.
0000 1XX	X	High Speed master code.
1111 1XX	X	Reserved for future purposes.
1111 0XX	X	10-bit slave addressing. Not yet supported.

Table 45 Reserved addresses for I2Cansactions

Note1: No device enables to acknowledge at the reception of the START byte.
Note2: The CBUS address has been reserved to enable the intermixing of CBUS compatible and the I2C-bus compatible devices in the same system. I2C-bus compatible devices are not allowed to respond on reception of this address.

Downloaded From Oneyac.com

6.5.1.4. Acknowledge

Data transfer with acknowledge is mandatory. The clock pulse related to acknowledge is generated by the master. The transmitter releases the SDA line (High) during the acknowledge clock pulse. The receiver must pull down the SDA line during the acknowledge pulse so that it remains stable Low during the High period of this clock pulse.

When a slave does not acknowledge the slave address or the data, the data line SDA must be left high by the slave. Then the master can generate a Stop condition to abort the transfer.

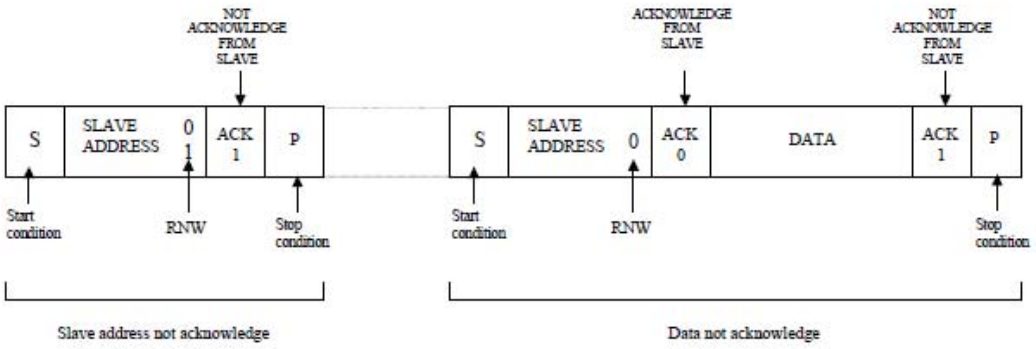


Figure 17 *"not acknowledge" by slave*

If a master receiver is involved in a transfer it must signal the end of data to the slave transmitter by not generating an acknowledgement after sending a byte. The slave will release SDA line to allow the master to generate Stop or repeated condition.

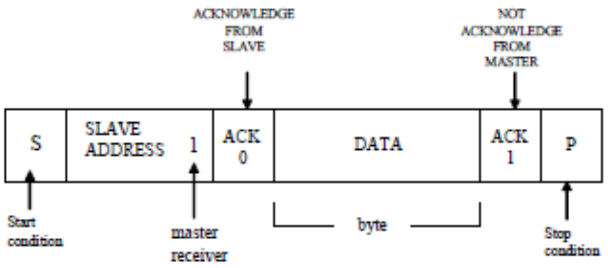


Figure 18 *"not acknowledge" by master (end of transmission)*

6.5.1.5. Clock synchronization and wait state

For this device the clock synchronization will be used to enable slaves to hold the

Downloaded From Oneyac.com

SCL line Low after reception and acknowledgement of a byte to force the master into a wait state. This enables to slave devices to get more time to store a received byte or prepare another byte to be transmitted.

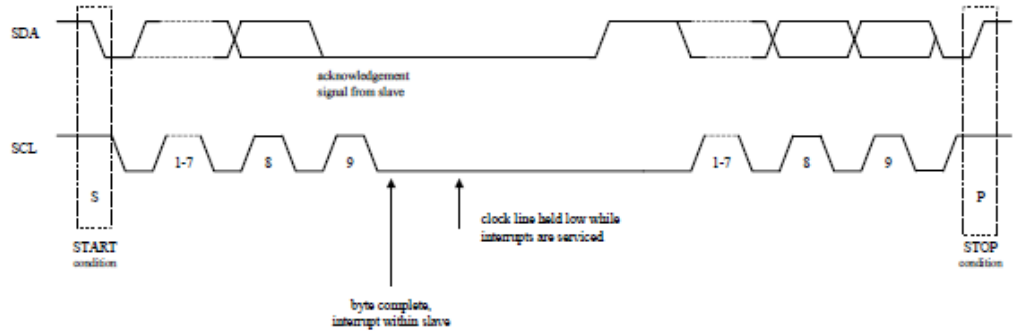


Figure 19 ***Clock synchronization1***

Clock synchronization is performed using the “wired AND” connection of I2C interface to the SCL line. This means that a High to Low transition on the SCL line will cause devices concerned to start counting off their Low period. At the end of their own Low period, devices will set their clocks High. However, SCL line will stay Low as long as one clock is still within its Low period. The SCL line will therefore be held Low by the device with the longest Low period. Devices with shorter Low periods enter a High wait state during this time. When all devices concerned have counted off their Low period, SCL line will be released and go High. There will then be no difference between device clocks and SCL line, and all devices will start counting their High periods. The first device to complete its High period will again pull the SCL line Low. In this way, a synchronized clock is generated.

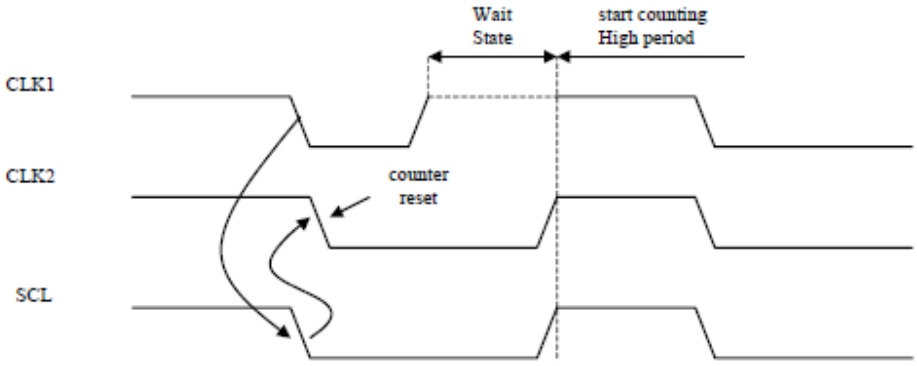


Figure 20 *Clock synchronization2*

Downloaded From Oneyac.com

6.5.1.6. Master mode: Transmission

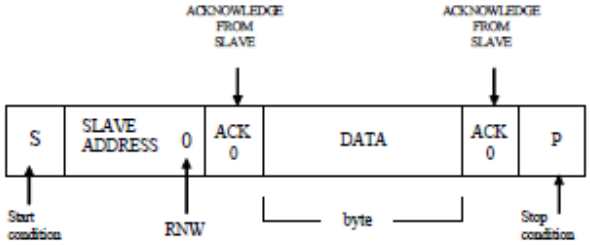


Figure 21 ***Typical transmission***

Initialization

Before starting the transmission, the CPU has to write slave address into MCADDR register. Note that for a write request the LSB of the slave address must be set to '0'.

The CPU will have to write a data byte regularly into the Transmit Buffer (MTXBUF register) during the transaction (care must be taken to avoid underflow). TBF (Transmit Buffer Full) or TBE (Transmit Buffer Empty) flags can be used to check the status of the Transmit Buffer.

Start

After initialization, CPU can start the transmission by setting the STA bit of the MCON register. Then, the master controller generates the Start condition on the I2C-bus. The STA bit is automatically cleared when the transmission has begun slave address transmission.

After that start has been sent, the slave address is loaded in the shift register to be transmitted on the I2C-bus and the master controller requests the first data byte to the Transmit. Once the slave address had been transmitted, the master controller waits for the slave address Acknowledge from the slave controller.

Data transmission

If the slave controller returns a slave address acknowledge, the master controller loads the data byte in the shift register to be transmitted on the I2C-bus. If this data byte was not the last one, the master sends a request to read the next data byte. Once data byte had been transmitted, the master controller waits for the data acknowledge from slave controller. In case of acknowledge and if data byte sent was not the last one, the controller sends another data byte.

Detection of last data byte:

CONFIDENTIAL

Downloaded From Oneyac.com

The data byte is the last data byte if STOP bit is set.

Note: Due to the size of the transmit buffer (1 byte), the first transmitted data byte is also the last data byte.

Stop and Repeated Start

Once last data byte had been transmitted, the master controller waits for the data acknowledge from slave controller. In case of acknowledge, if STA is set to '1' by the CPU, the controller will generate a Repeated Start in order to access to another slave device or change the direction of the transfer (Master Mode Reception) else a Stop condition is sent to finish the communication. The STOP bit is automatically cleared once the Stop condition or repeated Start has been sent.

In case of Repeated Start, the CPU must initialize the next transmission (write of Slave address, length and data bytes) before the end of the current transmission.

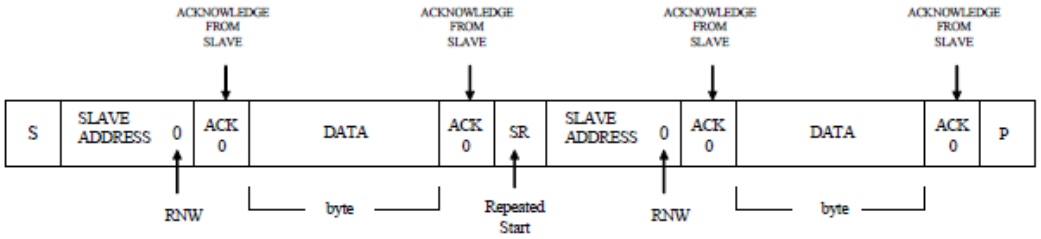


Figure 22 ***Repeated Start or Stop condition after last byte***

Transmission error

Not acknowledge from Slave Controller :

If the slave address is not acknowledged by the slave controller, the master interrupts the transmission by sending a Stop condition, and sets SANA flag.

If data is not acknowledged by the slave controller, the master interrupts the transmission by sending a Stop condition, and sets DNA flag.

Transmit Underflow :

If no data byte is valid from the Transmit Buffer when the controller needs to transmit a data byte, the master interrupts the transmission by sending a Stop condition, and sets UNF flag.

Such underflow occurs when the Transmit Buffer is empty (the CPU did not fill in time the Transmit Buffer).

Downloaded From Oneyac.com

End of error

When an error is detected, *OTXRXINT* output is set if the corresponding interrupt source is enabled. The Controller is blocked until MSTAT0 register is read by the CPU. This read operation resets MSTAT0 register and STA bit to disable potential Repeated Start. To pursue the transmission, the CPU must set STA only. To restart the same transmission from the beginning, the CPU must set software reset, refill MTXBUF and then set STA.

CONFIDENTIAL

Downloaded From Oneyac.com

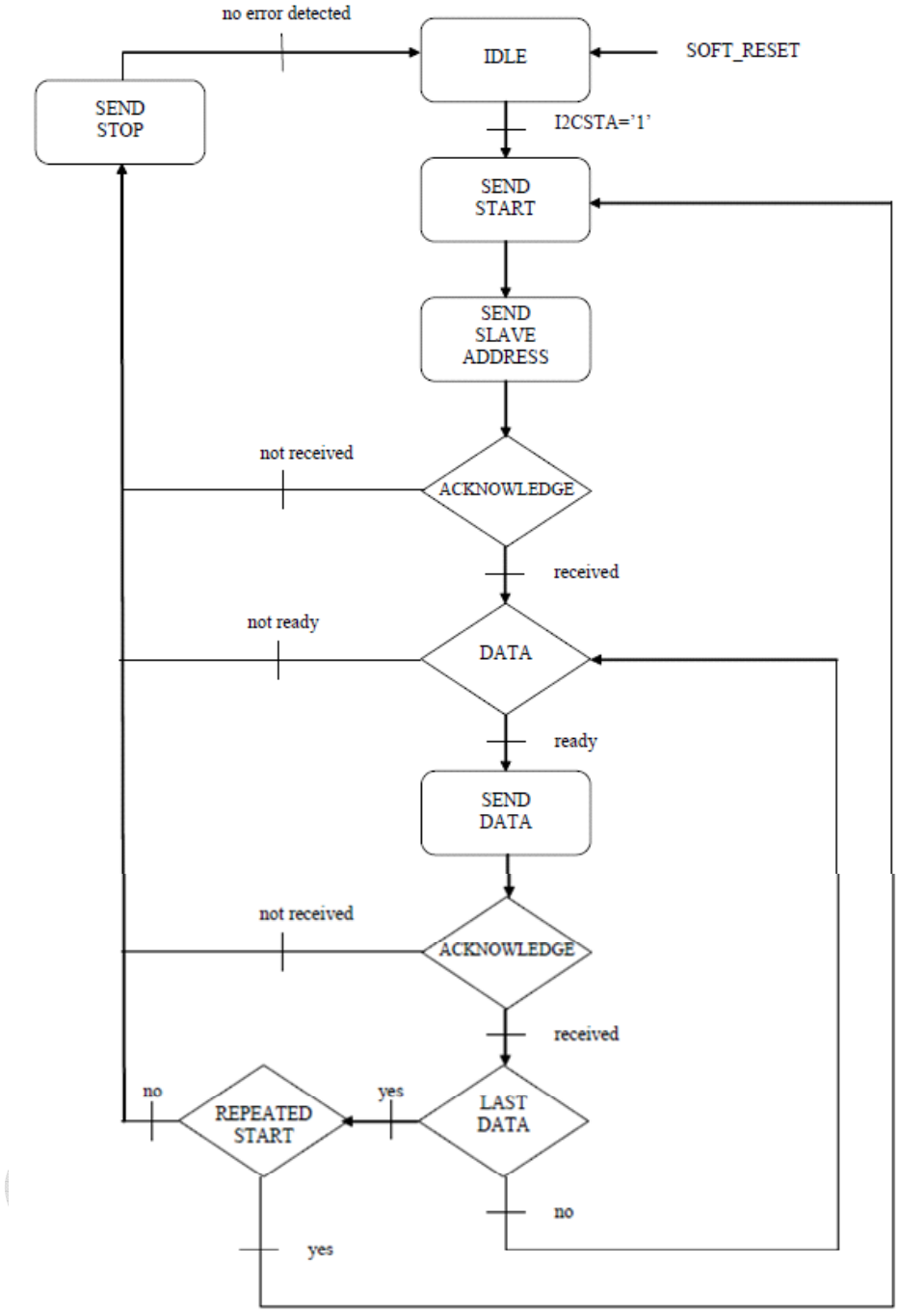


Figure 23 *Transmission FSM*

Note: If an error occurs during the transmission, FSM will stay into “SEND STOP” state until MSTAT0 register had been read by the CPU. This read operation will clear the STA bit of MCON register and MSTAT0 register. Since MSTAT0 register and MCON register had been reinitialized, the FSM is released into

Downloaded From Oneyac.com

"IDLE" state.

6.5.1.7. Master mode: Reception

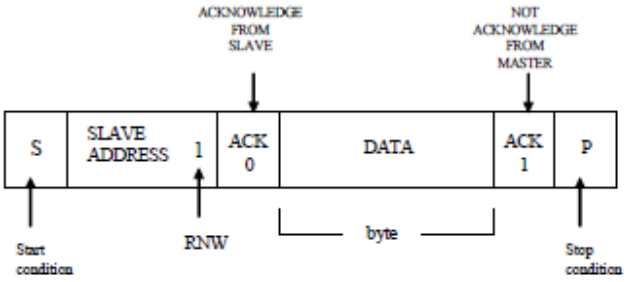


Figure 24 Typical reception

Initialization

Before starting the reception, the CPU has to write Slave address into the MCADDR register. Note that for a read request the LSB of the first byte (RNW bit) must be set to '1'.

The CPU will have to read the received data bytes in the Receive Buffer (MRXBUF register) regularly during the transaction (care must be taken to avoid overflow).

Start

After initialization, the CPU can start the reception by setting the STA bit of the MCON register. Then, the master controller generates the Start condition on the I2C-bus. The STA bit is automatically cleared when the transmission has begun.

Now the slave address is loaded in the shift register to be transmitted on the I2C-bus. Once the slave address had been transmitted, the master controller waits for the slave address acknowledge from the slave controller. If the slave controller returns a slave address acknowledgement, the master controller is waiting for first received data byte.

Reception

Once a data byte had been received, it is stored by the master controller in the Receive. Moreover, if received data byte is not the last one, the master controller sends an acknowledgement on the I2C-bus. Otherwise a "Not Acknowledge" is sent to indicate that it was the last read request and that slave controller must release the I2C bus to allow generating stop condition.

After the data acknowledge transmission, a new data reception can be done and the CPU can read the stored data using MRXBUF register.

CONFIDENTIAL

Downloaded From Oneyac.com

Detection of last data byte:

The data byte is the last data byte if STOP bit is set.

Note: Due to the size of the receive buffer (1 byte), the first received data byte is also the last data byte.

6.5.1.8. Stop and Repeated Start

After the “data not acknowledge” transmission, if STA is set to ‘1’ by the CPU, the controller will generate a Repeated Start in order to access to another slave device or change the direction of the transfer (Master mode Transmission) else a Stop condition is sent to finish the communication. The STOP bit is automatically cleared once the Stop condition or repeated Start has been sent.

In case of Repeated Start, the CPU must initialize the next transmission (write of Slave address, length and data bytes) before the end of the current reception.

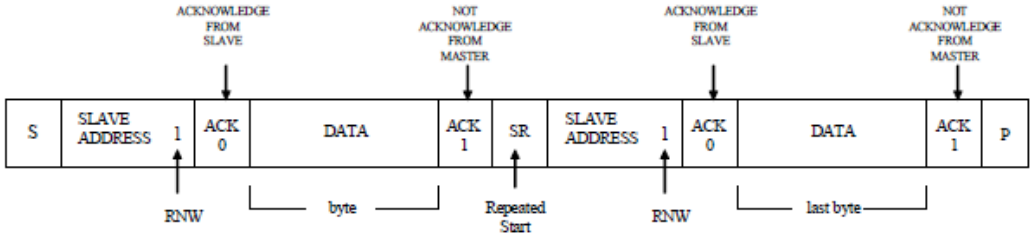


Figure 25 *Repeated Start or Stop condition after last byte*

Reception error

Not acknowledge from Slave Controller

If the slave address is not acknowledged by the slave controller, the master interrupts the transmission by sending a Stop condition, and sets SANA flag.

Receive Overflow

When a data byte is received, the TXRX controller checks that the previous data byte has been handled. If it is not the case (RXBUF overflow), the master interrupts the reception by sending “not acknowledge “ and a Stop condition, and sets OVF flag.

End of error

When an error is detected, OTXRXINT output is set if the corresponding interrupt

Downloaded From Oneyac.com

source is enabled. The controller is blocked until MSTAT0 register is read by the CPU. This read operation resets MSTAT0 register and STA bit to disable a potential Repeated Start. If the CPU wants to discard previous received data byte, it must set software reset. To restart the same transmission, the CPU just has to set STA.

6.5.1.9. Reception FSM

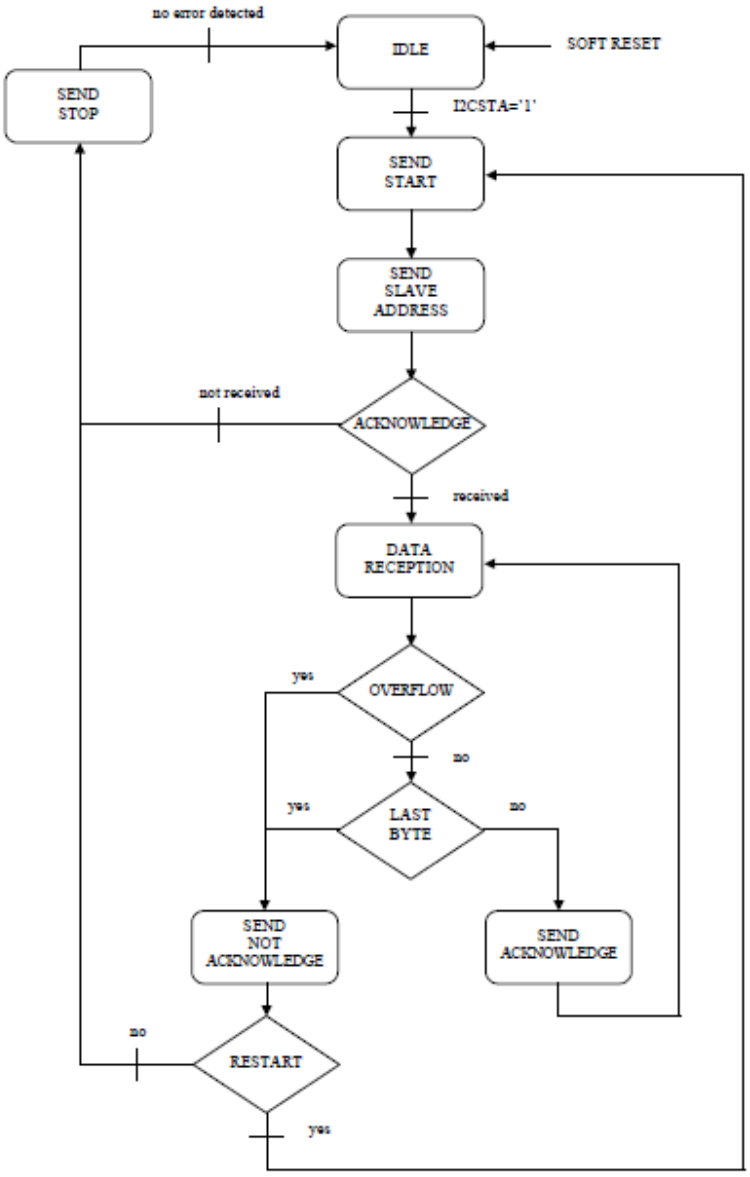


Figure 26 *Reception FSM*

Downloaded From Oneyac.com

Note: If an error occurs during the reception, FSM will stay into “SEND STOP” state until MSTAT0 register had been read by the CPU. This read operation will clear the STA bit of MCON register and MSTAT0 register. Since MSTAT0 register and MCON register had been reinitialized, the FSM is released into “IDLE” state.

Note： Due to the size of the receive buffer (1 byte), the first received data byte is also the last data byte. So, only one byte can be read from slave in one I2C process.

6.5.1.10. Data Handling

The Data Bytes exchanged on the I2C line are available in MRXBUF (received data) and MTXBUF (transmitted data) registers.

It is up to the user of the FlipI2CM to read/write data byte exchanged on the I2C line when they are available. This can be handled by software routine thanks to the status flags.

6.5.1.11. Software Reset

The software reset is activated by CPU setting bit SRST of control register (MCON). The software reset is used to stop current access on I2C bus.

Software reset initializes MCON, MSTAT0 registers and also TXRX controller.

If a software reset occurs during an I2C access, the master controller finishes the transmit or reception of current data byte, it send a Stop condition (in case of reception, send a ”not acknowledge" first) and next, MCON and MSTAT0 registers and TXRX controller are cleared.

At the end of software reset process, the master controller is ready to restart a new or the same access. For same access, the CPU must refill TXBUF (for transmission only) and set STA (MCADDR register is not affected by software reset).

6.5.2. I2C slave

6.5.2.1. Overview

This I2C Slave controller handles all functions necessary to respond to a request from an I2C master controller.

- Main features
- Support fast and standard transfer rates.
- 7-bit addressing on I2C.

CONFIDENTIAL

Downloaded From Oneyac.com

- Slave operations.
- Clock Stretching and Wait State generation.
- Operates from a wide range of input frequencies.
- Interrupt generation.

Received and Transmit Data are stored respectively in Receive Buffer and Transmit Buffer. Only one byte at a time can be stored in each buffer. During an I2C transaction, the CPU needs to read regularly the Receive Buffer and to write regularly the Transmit Buffer.

6.5.2.2. Register description

Mnemonic	Address	Description	Reset
STCON	S:0F1h	I2CS Transfer Control register	00h
SRXBUF	S:0F2h	I2CS Reception Buffer	00h
STXBUF	S:0F3h	I2CS Transmission Buffer	00h
SSTAT0	S:0F5h	I2CS Status register 0	00h
SSTAT1	S:0F6h	I2CS Status register 1	00h
SIEN0	S:0D5h	I2CS Interrupt Enable register 0	00h
SIEN1	S:0D6h	I2CS Interrupt Enable register 1	00h
SSADDR	S:0D7h	I2CS Self Address register	00h

Table 46 I2C slave register

6.5.2.3. STCON register

Bit Number	Bit Mnemonic	Function
7	--	**Reserved.** The value read from this bit is indeterminate.
6	I2CEN	**I2CS enable.** Set to activate FlipI2CS (I2CS responds to calls to its slave address and to the general call.) Clear to deactivate FlipI2CS (does not respond to any call through
5	SWS	**I2CS Wait State; default 0** Set to generate wait state on SCL line when RX overflows. When clear, FlipI2CS sends a "not acknowledge" to stop the transmission when RX overflows.
4:1	--	**Reserved.** The value read from these bits is indeterminate.
0	TIG	**Transfer In Progress.** Set to 1 by hardware when an I2C transfer is in progress on the I2C bus. Clear otherwise

Table 47 I2CS Transfer Control Register

CONFIDENTIAL

Downloaded From Oneyac.com

6.5.2.4. SRXBUF/STXBUF register

	Bit Number	Bit Mnemonic	Function
SRXBUF	7:0		Data received by I2CS
STXBUF	7:0		Data transmitted by I2CS

Table 48 DATA Register

6.5.2.5. SSTAT0 register

Bit Number	Bit Mnemonic	Function
7	--	**Reserved** The value read from this bit is indeterminate.
6	GC	**General call** Set to indicates that a general call has been detected.
5:3	--	**Reserved** The value read from these bits is indeterminate.
2	SUNF	**Transmission underflow** Transmitted Data Byte not ready (Transmission Buffer is empty) while a new data byte needs to be sent. A wait state is
1	SOVF	**Reception overflow** Received data byte could not be written (Reception Buffer is full) while a new bit was received. Set to 1 when Rx overflows and STCON.SWS =0. It indicates that receive Buffer is full while receiving a new byte. A Not Acknowledge is sent If STCON.SWS =1 when a new byte is received, a wait state is
0	SNE	**Normal End (End of access with no error)** Set when a stop is sent at the end of a successful access. Clear automatically when a new I2C access starts. It can also be cleared by software

Table 49 SSTAT0 Register

When GC, SUNF or SOVF flags have been set, reception and transmission process are disabled until the CPU reads SSTAT0 register. This read operation automatically clears these flags. These interrupt sources can all be individually enabled/disabled by SIEN0 register.

When a disabled interrupt occurs, *the interrupt* won't trigger the FLIP51, but the corresponding interrupt bit is set. When a general call is detected, the Slave controller sets SSTAT0.GC to '1'. The CPU has to handle received data as General Call information.

6.5.2.6. SSTAT1 register

Bit Number	Bit Mnemonic	Function
7	--	**Reserved** The value read from this bit is indeterminate.

CONFIDENTIAL

Downloaded From Oneyac.com

6	--	**Reserved** The value read from this bit is indeterminate.
5	STBE	**I2CS Transmission buffer is empty** Set to 1 when Transmit Buffer is empty. Clear to 0 when at least one Byte is ready for data transmission This flag is cleared when the CPU performs a write access to STXBUF
4	--	**Reserved** The value read from this bit is 0.
3	STBF	**I2CS Transmission buffer is full** Set to 1 when Transmission Buffer is full. When set, no more write operation into transmission buffer or memory is performed (CPU write request to STXBUF is not taken in account). Clear to 0 when the transmission buffer is empty This flag is cleared when a new Data Byte is requested by the I2C
2	SRBE	**I2CS Reception buffer is empty** Set to 1 when reception Buffer is empty. No more write operation into reception buffer or memory is performed (CPU read request to SRXBUF not taken in account). This flag is cleared when a new Data Byte is received by the TXRX controller.
1	--	**Reserved** The value read from this bit is indeterminate.
0	SRBF	**I2CS Reception buffer is full** Set to 1 when reception Buffer is full. This flag is cleared when the CPU performs a read operation to SRXBUF register. Clear to 0 when reception Buffer is empty

Table 50 SSTAT1 Register

These interrupt sources can all be individually enabled/disabled by SIEN1 register. When a disabled interrupt occurs, *the interrupt* won't trigger the FLIP51, but the corresponding interrupt bit is set. These interrupt sources are cleared when the condition which has set them disappears.

6.5.2.7. I2CS Interrupt Enable Register 0

Bit Number	Bit Mnemonic	Function
7	--	**Reserved** The value read from this bit is indeterminate.
6	EGC	**I2CS Enable General Call interrupt (SSTAT0.SGC)** Clear to disable SSTAT0.SGC bit to generate an interrupt request Set to enable SSTAT0.SGC bit to generate an interrupt request
5	--	**Reserved** The value read from this bit is indeterminate.
4	--	**Reserved** The value read from this bit is indeterminate.
3	--	**Reserved** The value read from this bit is indeterminate.
2	ESUNF	**I2CS Underflow Interrupt enable bit** Clear to disable SSTAT0.SUNF bit to generate an interrupt request Set to enable SSTAT0.SUNF bit to generate an interrupt request

CONFIDENTIAL

Downloaded From Oneyac.com

1	ESOVF	**I2CS Overflow Interrupt enable bit** Clear to disable SSTAT0.SOVF bit to generate an interrupt request Set to enable SSTAT0.SOVF bit to generate an interrupt request
0	ESNE	**I2CS Normal End Interrupt enable bit** Clear to disable SSTAT0.SNE bit to generate an interrupt request Set to enable SSTAT0.SNE bit to generate an interrupt request

Table 51 SIEN0 *Register*

6.5.2.8. I2CS Interrupt Enable Register 1

Bit Number	Bit Mnemonic	Function
7	--	**Reserved** The value read from this bit is indeterminate.
6	--	**Reserved** The value read from this bit is indeterminate.
5	ESTBE	**I2CS Transmission Buffer Empty Interrupt enable bit** Clear to disable SSTAT1.STBE bit to generate an interrupt request Set to enable SSTAT1.STBE bit to generate an interrupt request
4	--	**Reserved** The value read from this bit is indeterminate.
3	ESTBF	**I2CS Transmission Buffer Full Interrupt enable bit** Clear to disable SSTAT1.STBF bit to generate an interrupt request Set to enable SSTAT1.STBF bit to generate an interrupt request
2	ESRBE	**I2CS Reception Buffer Empty Interrupt enable bit** Clear to disable SSTAT1.SRBE bit to generate an interrupt request Set to enable SSTAT1.SRBE bit to generate an interrupt request
1	--	**Reserved** The value read from this bit is indeterminate.
0	ESRBF	**I2CS Reception Buffer Full Interrupt enable bit** Clear to disable SSTAT1.SRBF bit to generate an interrupt request Set to enable SSTAT1.SRBF bit to generate an interrupt request

Table 52 SIEN1 Register

6.5.2.9. I2CS Self Address Register

Bit Number	Bit Mnemonic	Function
7	--	**Reserved** The value read from this bit is indeterminate
6:0	SADDR	**7-bit Self Address** This register must be written before the beginning of an I2C

Table 53 SSADDR Register

6.5.2.10. I2C Frame data format

The I2C-bus supports two formats: 7-bit address format and 10-bit address format.

This I2C Slave controller only support 7-bit format as shown on the figure below:

Downloaded From Oneyac.com

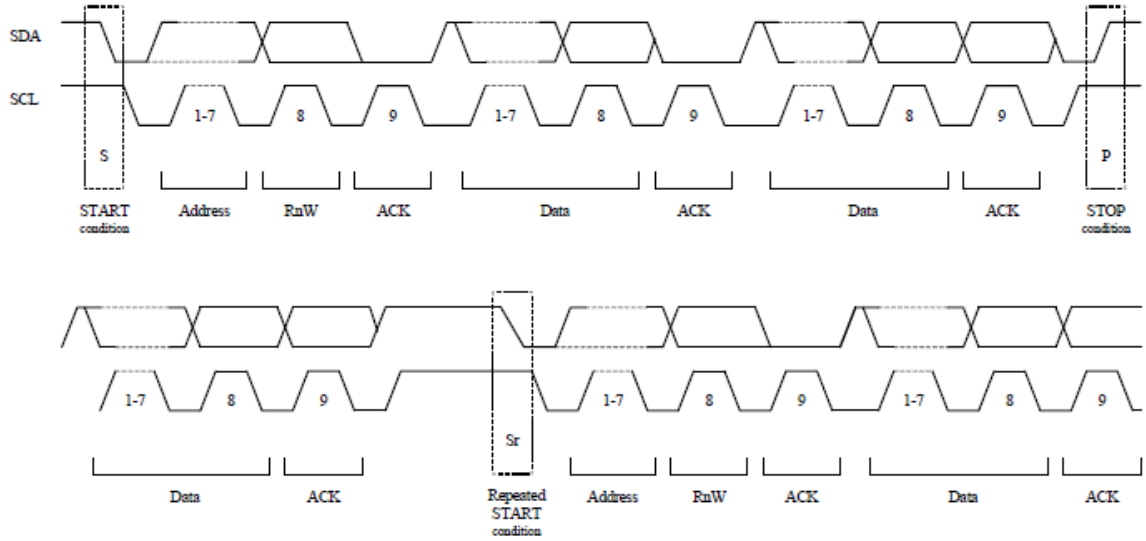


Figure 27 *Complete data transfer 7-bit addressing*

All words put on the SDA line are 8-bits long. The number of bytes that can be transmitted on an I2C line is unrestricted. Each byte is followed by an acknowledge bit set by the receiver. Data is transferred with the most significant bit (MSB) first.

After the Start condition (S), a slave address is sent. This address is 7 bits long. The eighth bit determines the direction of the message (R/WN): a '0' means that the Master will write data to a selected slave, a '1' means that the Master will read data from a selected slave.

A data transfer is always terminated by a STOP condition (P). However, if the Master still wishes to communicate on the bus, it can generate a Repeated Start (Sr) that this to say to generate another START without first generating a STOP. Various combinations of read/write formats are then possible within such a transfer.

Note that two groups of eight addresses (0000XXX and 1111XXX) are reserved for purposed shown in the following table.

SLAVE ADDRESS	RnW Bit	Description
0000 000	0	General call address. It is used to address every device connected to I2C-
0000 000	1	START byte(1)
0000 001	X	CBUS address(2).

Downloaded From Oneyac.com

0000 010	X	Reserved for different bus format.
0000 011	X	Reserved for future purposes.
0000 1XX	X	High Speed master code.
1111 1 XX	X	Reserved for future purposes.
1111 0XX	X	10 bit slave addressing.
(1): No device allowed to send acknowledge at the reception of the START byte.		
(2): The CBUS address has been reserved to enable the intermixing of CBUS compatible and the I2C-bus compatible devices in the same system. I2C-bus compatible devices are not allowed to respond on reception of this address.		

Table 54 Reserved addresses for I2C transactions

6.5.2.11. Acknowledge

Data transfer with acknowledge is mandatory. The clock pulse related to acknowledge is generated by the master. The transmitter releases the SDA line (High) during the acknowledge clock pulse. The receiver must pull down the SDA line during the acknowledge pulse so that it remains stable Low during the High period of this clock pulse.

When a slave device does not acknowledge the slave address or the data, the data line SDA must be left high by the slave. Then the master can generate a Stop condition to abort the transfer.

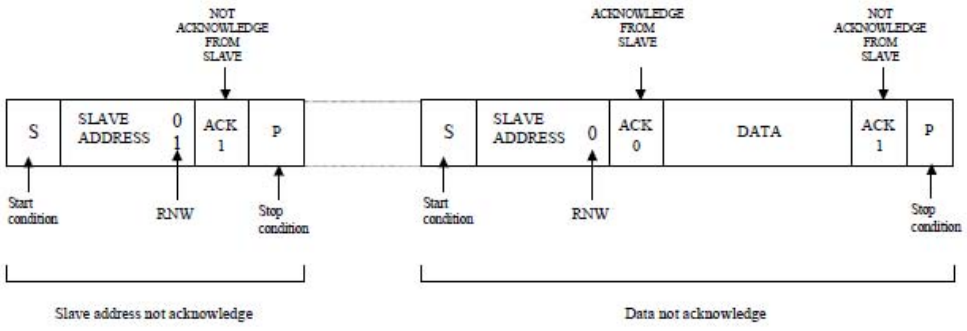


Figure 28 *"not acknowledge" by slave device*

If a master receiver is involved in a transfer it must signal the end of data to the slave transmitter by not generating an acknowledgement on the last byte. The slave will release SDA line to allow the master to generate Stop or repeated condition.

Downloaded From Oneyac.com

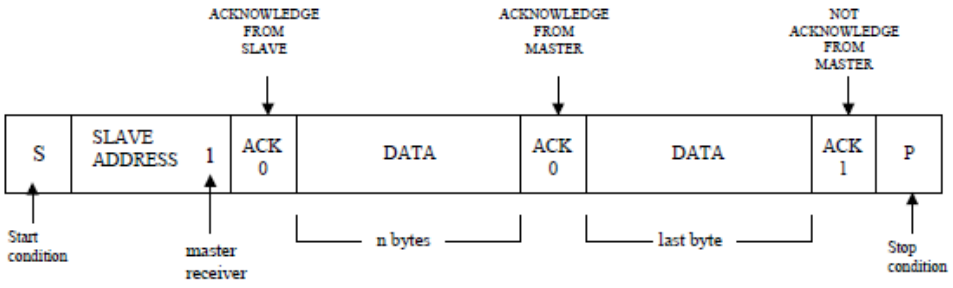


Figure 29 *"not acknowledge" by master device*

6.5.2.12. Clock synchronization

For this device the clock synchronization will be used to enable slave devices to hold the SCL line Low after reception and acknowledgement of a byte to force the master into a wait state. This enables to slave devices to get more time to store a received byte or prepare another byte to be transmitted.

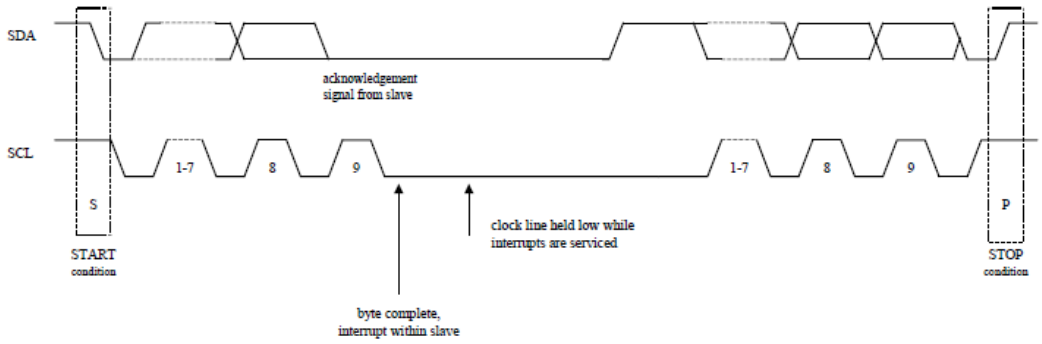


Figure 30 *Clock synchronization*

Clock synchronization is performed using the wired_AND connection of I2C interface to the SCL line. This means that a High to Low transition on the SCL line will cause devices concerned to start counting off their Low period. At the end of their own Low period, devices will set their clocks High. However, SCL line will stay Low as long as one clock is still within its Low period. The SCL line will therefore be held Low by the device with the longest Low period. Devices with shorter Low periods enter a High wait state during this time. When all devices concerned have counted off their Low period, SCL line will be released and go High. There will then be no difference between device clocks and SCL line, and all devices will start counting their High periods. The first device to complete its High period will again pull the SCL line Low. In this way, a synchronized clock is generated.

Downloaded From Oneyac.com

6.6. RNG

There is a pseudo random number generator in BK2535.

The RNG is operated through two registers; RNG_CTL and RNG_DAT. RNG_CTL contains only one control bit. RNG_DAT contains the random data and the seed.

RNG_CTL	7	6	5	4-0
0x931	En_RNG			/

RNG_DAT	7-0
0x930	Data/seeds

Table 55 RNG Register

En_RNG: write 1 to enable the pseudo RNG module.

When use pseudo random generator, RNG_DAT is always ready in the register, you can read the result at any time.

Also, you can change the seed by writing the seed to 0X930.

The recommended steps for using RNG descript as below:

Enable the RNG module.

Write an initial seed to the RNG_DAT.

Read out the pseudo random number.

CONFIDENTIAL

Downloaded From Oneyac.com

6.7. LBD

The LBD circuit is used to monitor power supply.

The recommend detect process is described as follow:

1. Configure the LBD_THD register correctly according to Table 56.
2. Set the hysteresis as 0x3 through writing the register 0x8E2[2:0] ,
3. Power up LBD circuit (write 1 into bit 7 of address 0x8E3.)
4. Enable the NMI interrupt by setting the SFR 0xc9[1]
5. The interrupt will be generated when LBD happen.

6.7.1. LBD threshold

lbd_thre <4:0>	Vth(V)	lbd_thre <4:0>	Vth(V)	lbd_thre <4:0>	Vth(V)	lbd_thre <4:0>	Vth(V)
0	0.51	8	1.02	16	1.51	24	2
1	0.58	9	1.08	17	1.57	25	2.07
2	0.64	10	1.14	18	1.64	26	2.13
3	0.7	11	1.2	19	1.7	27	2.19
4	0.76	12	1.26	20	1.76	28	2.25
5	0.83	13	1.32	21	1.82	29	2.32
6	0.9	14	1.38	22	1.88	30	2.38
7	0.95	15	1.45	23	1.94	31	2.44

Table 56 LBD threshold Register

CONFIDENTIAL

Downloaded From Oneyac.com

6.8. FLASH control

Except the main 32k FLASH program space, there are Two 256 bytes (NVR space) information space had been integrated in BK2535. This main and the NVR space could be operated by MCU directly. The FLIP51 would enter IDLE mode during operating the main/NVR FLASH space.

FLASH control base address: 0X900

ADDR	register	7	6	5	4	3	2	1	0
0x00	FLASH_KEY	A5/49							
0x01	FLASH _CTL	W	R	E				N/M	Clk_en
0x02	FLASH _ADR	Addr[7:0]							
0x03	FLASH _ADR	Addr[15:8]							
0x04	FLASH _DAT	Data[7:0]							

Table 57 FLASH control Register1

addr	register	7	6	5	4	3	2	1	0
0X05	WP0	This register must equal to A5 when E/W operation							
0X06	WP1	This register must equal to C3 when E/W operation							
0X07	reserved	Don't write any value in it							
0X08	NVR_WF	Write any value in it will forbid E/W operation on NVR space							
0X09	MAIN_WF	Write any value in it will forbid E/W operation on MAIN space							

Table 58 FLASH control Register2

//FLASH_CTL register

Control bit [7]: write, write 1 to operate, the bit will be cleared automatically after operate.

Control bit [6]: read, write 1 to operate, the bit will be cleared automatically after operate.

Control bit [5]: erase, write 1 to operate, the bit will be cleared automatically after operate.

Control bit [1]: NVRMAIN space control. 0: NVR space; 1: main space

Control bit [0]: clock enable bit; the clock of flash operate module will be closed when this bit=0. Once a operation is finished, you should clear the clk_en bits for forbidding any operation to the FLASH space.

Note1: only one operation can run at the same time.

CONFIDENTIAL

Downloaded From Oneyac.com

For example: read data from NVR address 0x20, the following steps are recommended.

1. Write 0XA5 to SFR address 0X05;
2. Write 0XC3 to SFR address 0X06;
3. write 0X01 to SFR address 0x01 //open clock and select the NVR space
4. write 0Xa5 to SFR address 0x00 //key
5. write 0X49 to SFR address 0x00;
6. write the address into 0X02 and 0X03;
7. write 0X41 to SFR address 0x01; //start (NVR space)
8. wait until bit6 changed to zero
9. read out the data from Ox04;
10. Write 0x00 into 0X01 to close the clock
11. Write 0X00 to SFR address 0X05;
12. Write 0X00 to SFR address 0X06;

CONFIDENTIAL

Downloaded From Oneyac.com

6.9. WDT

There is a watch dog timer in BK2535. When overflow happened, the WDT will trigger the CPU into reset status and rerun from the beginning location. The software need feed the dog timely to avoid the overflow happen.

Note: the reset does not affect RF part.

There are two methods to enable the WDT.

One is writing 0Xa5 on SFR address 0XA6(WDCON) and this operation will clear the WDT counter also (feed dog). Once the WDT enabled by this method, you can disable the WDT through writing 0XDE and 0XAD consecutively during eight clock periods. When the WDT enabled by this method, you can also set whether running in IDLE state. To do this, you can enable it by writing 0XD1 on SFR address 0XA6 or disable it by writing 0XDE and 0XDA consecutively during eight clock periods.

The other method is writing 0XFF on SFR address 0XA6. You cannot close it once you enable the WDT with this method except any reset happened. In this status, the WDT will run always even in IDLE state.

WDCON	7	6	5	4	3	2	1	0
0XA6	/	/	/	state	/	ps2	ps1	ps0

Table 59 Watch Dog Register

State: read only 1: the WDT in active status 0: the WDT in inactive status

Ps2, ps1, ps0: the prescaler of watch dog clock.

Note: when write the prescaler value, the bit7 must be set as 0.

PS2	PS1	PS0	PRE_scale
0	0	0	2
0	0	1	4
0	1	0	8
0	1	1	16
1	0	0	32
1	0	1	64
1	1	0	128
1	1	1	256

CONFIDENTIAL

Downloaded From Oneyac.com

Table 60 the Prescaler of Watch Dog clock

The overflow time of watch dog：

错误！不能通过编辑域代码创建对象。

When overflow occur, the whole system will be reset.

6.10. Ext_Timer

A simple timer is integrated in BK2535 for fixed time interrupt for some special application, such as mouse. (8ms wake up)

This timer selects 32k clock always for avoiding the effect brought by clock switch.

The period can be set precisely through the register descript below:

ADDR	[7:3]	[2]	[1:0] **timer_div**	
0X918	reserved	RTC enable	RTC clock divider	
	[7:0] **timer_count**			
0X919	RTC counter high byte			Counter[15:0]= [0X919,0X91A]
0X91A	RTC counter low byte			

Table 61 RTC Register

RTC period = **1/32e3 *（2+ timer_div） * (1+timer_count)**

For example, if you want to get 8ms period wakeup, you can set timer_div=2 and timer_count=63.

Note: to enable the RTC interrupt, you should set EA= 1 and EX6 = 1.

CONFIDENTIAL

Downloaded From Oneyac.com

6.11. Encryption Decryption Unit (AES)

The BK2535 has dedicated HW for data encryption or decryption according to the Advance Encryption Standards (AES).

AES control register: locate XRAM space

AES_CTL	7	6	5	4	3	2	1	0
0X9B0	FINISH	/	/	/	/	/	ENC_TYPE	START_AES

Table 62 AES control register

ENC_TYPE: AES mode select, 1: Encryption; 0: Decryption

START_AES: Posedge will start the operate.

FINISH：1 indicate the current operation finished, you should changed it to 0 before next operation.

KEY/plain text/cipher test register: locate XRAM space

KEY	15	14	….	….	3	2	1	0
REG	0x98F	0x98E	…..	…...	0x983	0x982	0x981	0x980
RW	High bytes ←---------←--------- Low bytes							
TXT IN	15	14	….	….	3	2	1	0
REG	0x99F	0x99E	…..	…...	0x993	0x992	0x991	0x990
RW	High bytes ←---------←--------- Low bytes							
TXT OUT	15	14	….	….	3	2	1	0
REG	0x9AF	0x9AE	…..	…...	0x9A3	0x9A2	0x9A1	0x9A0
R	High bytes ←---------←--------- Low bytes							

Table 63 KEY register

INT register

AES_INT	7	6	5	4	3	2	1	0
0X9B1	/	/	/	/	/	/	/	AES_INT

Table 64 AES INT register

Key[127:0] is the KEY used by Encryption/Decryption, text_in[127:0] is the input of plain text or cipher text, text_out[127:0] is the output of Encryption/Decryption.

CONFIDENTIAL

Downloaded From Oneyac.com

When Encryption/Decryption finished, AES_INT will be generated automatically.

The next steps are recommended when you used AES module.

A: write calculation mode (E/D) into register.

B: write text input and KEY into register.

C: write 0 into START_AES then write 1 into START_AES to start the operation.

D: When the calculation finished, AES_INT will generated and FINISH will change to high, you can wait the interrupt or query the FINISH register to acquire the result.

CONFIDENTIAL

Downloaded From Oneyac.com

MDU

The MDU – Multiplication Division Unit, is an on-chip arithmetic co-processor which enables the MCU to perform additional extended arithmetic operations like 32-bit division, 16-bit multiplication, shift, and normalize operations.

MDU support unsigned integer only. The MDU is handled by seven registers, which are memory mapped as Special Function Registers. The arithmetic unit allows concurrent operations to be performed independent of the MCU's activity.

Operands and results are stored in from MD0 to MD5 registers. The module is controlled by the MDCTL register. Any calculation of the MDU overwrites its operands.

The MDU does not allow reentrant code and cannot be used in multiple threads of the main and interrupt routines at the same time.

Address	SFR
0xC1	MD0
0xC2	MD1
0xC3	MD1
0xC4	MD1
0xC5	MD1
0xC6	MD1
0xC7	MDCTL

Table 65 MDU SFR

MDCTL（R/W by software）MDCTL meaning is different when reading and writing

reading:

MDCTL	7	6	5	4-0
function	mdef	mdov	done	NORM_shift_number

Table 66 MDU Register (Read)

mdef : MDU Error flag MDEF. Indicates an improperly performed operation (when one of the arithmetic operations has been restarted or interrupted by a new operation)

mdov : MDU Overflow flag MDOV. Overflow occurrence in the MDU operation.

Done : Calculate is done or not. 1: operate is done now; 0: busy

CONFIDENTIAL

Downloaded From Oneyac.com

NORM_shift_number: left shift number stored in it when NORM is done.

Writing:

MDCTL	7	6	5	4	3	2	1	0
function	op-code			SC				

Table 67 MDU Register(Write)

SC: Shift counter.

op-code: operate code which is show as next table.

op-code	meaning
1	32 bit / 16 bit divide
2	16bit * 16bit multiply
3	16 bit /16 bit divide
4	Left shift
5	Right shift
6	normalizing
others	idle

Table 68 MDU operation Table

Operate data

OP	32bit / 16bit		16bit / 16bit		16bit x 16bit		shift or normalize	
DATA	MD0(LSB) MD1 MD2 MD3(MSB)	dividend	MD0(LSB) MD1(MSB)	dividend	MD0(LSB) MD1(MSB)	dividend	MD0(LSB) MD1 MD2 MD3(MSB)	Op-code
DATA	MD4(LSB) MD5(MSB)	divisor	MD4(LSB) MD5(MSB)	divisor	MD4(LSB) MD5(MSB)	divisor		

Table 69 Operation Data

Reading result

OPERATION	**32bit / 16bit**		**16bit / 16bit**		**16bit x 16bit**		**shift or normalize**	
DATA	MD0(LSB) MD1 MD2 MD3(MSB)	quotient	MD0(LSB) MD1(MSB)	quotient	MD0(LSB) MD1 MD2 MD3(MSB)	product	MD0(LSB) MD1 MD2 MD3(MSB)	result
DATA	MD4(LSB) MD5(MSB)	remainder	MD4(LSB) MD5(MSB)	remainder				

CONFIDENTIAL

Downloaded From Oneyac.com

Table 70 Operate Result

Normalizing

When set op-code = 6, normalizing start to run. All leading zeroes of 32-bit integer variable stored in the MD0.. MD3 registers are removed by shift left operations. The whole operation is completed when the MSB (Most Significant Bit) of MD3 register contains a '1'. After normalizing, bits NORM_shift_number contain the number of shift left operations that were done.

Example:

Run code：

```
Mov  MD3 , #00001101b;
Mov  MD2 , #00000001b;
Mov  MD1 , #00000011b;
Mov  MD0 , #00000111b;
Mov MDCTL #11000000b;   //start normalizing
after 6 clock period, we can read,
NORM_shift_number =4;
MD3 , #11010000b;   // left shift four bit until the MSB of MD3is 1
MD2 , #00010000b;   // left shift four bit
MD1 , #00110000b;   // left shift four bit
MD0 , #01110000b; //   left shift four bit。
```

Shift operation

N shift operation, 32-bit integer variable stored in the MD0... MD3 registers (the latter contains the most significant byte) is shifted left or right by a specified number of bits. The op-code defines the shift direction and the shift count. During shift operation, zeroes come into the left end of MD3 for shifting right or they come in the right end of the MD0 for shifting left.

Mdef error flag

The mdef error flag indicates an improperly performed operation (when one of the arithmetic operations is restarted or interrupted by a new operation).

The error flag is set when:

* If you write to MD0.. MD5 and/or op-code during phase two of MDU operation (restart or calculations interrupting).

* If any of the MDx registers are read during phase two of MDU operation when the error flag mechanism is enabled. In this case, the error flag is set but the

Downloaded From Oneyac.com

calculation is not interrupted.

Mdef will be set when error happened and be cleared when new operation started.

Mdov MDU－overflow flag

This bit is set by hardware and cleared by software.

The mdov overflow flag is set when one of the following conditions occurs:

* Division by zero

* Multiplication with a result greater than 0XFFFFh

* start of normalizing if the most significant bit of MD3 is set (“md3.7” = ‘1’).

Note: any new operation will clear this bit.

Executing calculation

During executing operation, the MDU works on its own in parallel with the MCU.

operation	Number of clock cycles
32bit / 16bit	9 clock cycles (max and min)
16bit / 16bit	9 clock cycles (max and min)
multiplication	9 clock cycles (max and min)
shift	7 clock cycles (max and min)
normalize	7 clock cycles (max and min)

Table 71 MDU operations execution times

Note： The clock cycle is CPU clock cycle

CONFIDENTIAL

Downloaded From Oneyac.com

7. BOOSTER

A DC-DC booster is embedded in BK2535. The booster is low consumption, high efficiency, low ripple and low startup voltage DC-DC converter. It can deliver 40mA current at 1.8V output. BK2535 could work with wide range of voltage input from 0.7V to 1.5V thanks for the circuit.

The typical application is showed in the next figure.

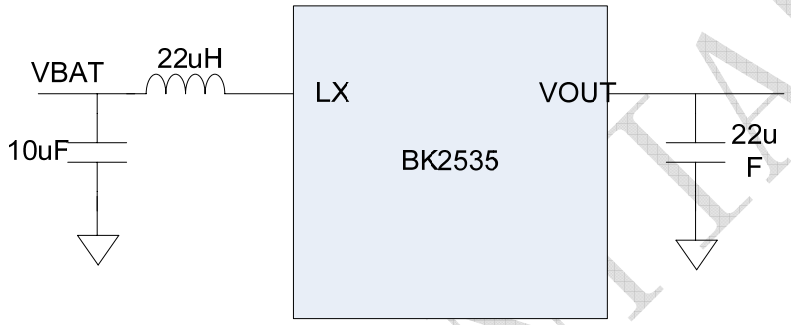


Figure 31 *booster application*

CONFIDENTIAL

Downloaded From Oneyac.com

8. USB

The USB module in BK2535 provides a full speed USB function interface that meets the 1.1 and 2.0 specification. USB module has 8 endpoints and the depth and start address of every endpoint FIFO can be configured. The FIFO can locate any position of the 2K EXRAM. It supports control, interrupt, bulk, synchronous transfer mode; also it supports multiple-buffer operation controlled by software for using the USB bandwidth sufficiently.

Note: It is assumed the reader is familiar with or has access to the supporting documents USB1.1.

8.1. Clock

USB clock is 48MHz which is generated by PLL integrated in the chip. USB module can enter into idle mode for saving power consumption by setting the USB_PWR_CN.1 (0x0841) SFR. In this state, the register of USB can be read or write, but the USB engine is halted and cannot respond any external operation.

8.2. USB Register Access

The register of USB located from 0x0808 to 0x0850 of external RAM. The access method is same to external RAM, use MOVX command.

USB register included interrupt register, configure register, power management register and address register.

8.3. ENDPOINT Configuration

The USB module should be configured before using USB to communicate. The configure item includes endpoint address in EXRAM, the depth of FIFO, how many endpoints are used, and the direction, mode, enable of every endpoints.

NOTE: USB and MCU share the same RAM space, so the overlap should be avoided carefully.

Next is the description of these register. All the register can read or write by software.

EP_ADDR_MSB [0x0840]

CFG_EP0_1 [0x0810], CFG_EP0_0 [0x0811] (endpoint 0 configure register)

CFG_EP1_1 [0x0812], CFG_EP1_0 [0x0813] (endpoint 1 configure register)

CONFIDENTIAL

Downloaded From Oneyac.com

CFG_EP2_1 [0x0814], CFG_EP2_0 [0x0815] (endpoint 2 configure register)

CFG_EP3_1 [0x0816], CFG_EP3_0 [0x0817] (endpoint 3 configure register)

CFG_EP4_1 [0x0818], CFG_EP4_0 [0x0819] (endpoint 4 configure register)

CFG_EP5_1 [0x081a], CFG_EP5_0 [0x081b] (endpoint 5 configure register)

CFG_EP6_1 [0x081c], CFG_EP6_0 [0x081d] (endpoint 6 configure register)

CFG_EP7_1 [0x081e], CFG_EP7_0 [0x081f] (endpoint 7 configure register)

Note: endpoint 0 is the control port. It occupies 64 bytes xram space the size and mode of it cannot be configured.

Next is the detail description of the register：

EP_ADDR_MSB（the MSB address bit of endpoints）：

EP_ADDR_MSB	7	6	5	4	3	2	1	0
0x0840	-	-	-	-	-	-	-	-

Table 72 USB MSB endpoint address

7: the MSB of endpoint 7 address. It decides the port address locates above 1K space or below it.

6: the MSB of endpoint 6 address. It decides the port address locates above 1K space or below it.

5: the MSB of endpoint 5 address. It decides the port address locates above 1K space or below it.

4: the MSB of endpoint 4 address. It decides the port address locates above 1K space or below it.

3: the MSB of endpoint 3 address. It decides the port address locates above 1K space or below it.

2: the MSB of endpoint 2 address. It decides the port address locates above 1K space or below it.

1: the MSB of endpoint 1 address. It decides the port address locates above 1K space or below it.

0: the MSB of endpoint 0 address. It decides the port address locates above 1K space or below it. Default 1, above 1K space.

CONFIDENTIAL

Downloaded From Oneyac.com

CFG_EP0_1 (the configure register 1 of endpoint 0)：

CFG_EP0_1	7	6	5	4	3	2	1	0
0x0810	dir	ep0_en	/				addr[9:8]	

Table 73 Configure Register 1 of Endpoint 0

7. Dir, port direction

1: IN (BK2535 send out data)；

0: OUT (the PC send out data)。

6. ep0_en, endpoint 0 enable

When ep_rdy[0] =0 and ep0_en=0, usb no respond to external now.

addr[9:8] ： The higher 2 bits ([9:8]) address of endpoint0. The low 8 bits address is stored in CFG_EP0_0.

Note: The dir bit of CFG_EP0_1 is set or cleared by software except that it is cleared by hardware when SETUP token coming. The direction is forced to OUT to access 8 bytes setup request in this condition. The setup request has the highest priority.

CFG_EPn_1 (endpoint n configure register)：(n=1 - 7)

CFG_ EPn_1	7	6	5	4	3	2	1	0
	Dir	Mode		Size			Addr[9:8]	

Table 74 Endpoint n Configure Register

7. Dir:

1: IN 0: OUT

6-5. Mode:

0 -- Control Transfer

1 -- Bulk Transfer

2 -- ISO Transfer

3 -- Interrupt Transfer

CONFIDENTIAL

Downloaded From Oneyac.com

4-2. Size：

0— endpoint not available

1— 16 bytes buffer size

2—32 bytes buffer size

3—64 bytes buffer size

4—128 bytes buffer size

5—256 bytes buffer size

6—512 bytes buffer size

7—endpoint not available

1-0. Addr[9:8] ： The higher 2 bits ([9:8]) address of endpoint n.

CFG_EPn_0 (configure register 0 of endpoint n)：(n=0 - 7)

CFG_ EPn_0	7	6	5	4	3	2	1	0
	addr[7:0]							

Table 75 Endpoint nConfigure Register 0

The lower 8 bits address of endpoint n.

8.4. Interrupt

External interrupt 4 is assigned to USB. Int4 will be triggered if any enabled interrupt bit in USBINT0 or USBINT1 is set to 1. Software should query the register to find out the relevant interrupt source. Also, software should clear the interrupt bit by set it to 1 after dealing with the interrupt.

USBINT0 interrupt register

USB INT0	7	6	5	4	3	2	1	0
0x080a	ctl_ rec	ctl_ send	rx_rdy	tx_rdy	usb_rs t	usb_s us	usb_r es	usb_s of

Table 76 USBINT0 Interrupt Register

7. ctl_rec: data received on control port (endpoint 0)
6. ctl_send: data send on control port (endpoint 0)
5. rx_rdy ： data received on endpoint 1-7
4. tx_rdy ： data send on endpoint 1-7

CONFIDENTIAL

Downloaded From Oneyac.com

3. usb_reset ： USB Reset interrupt
2. usb_sus ：USB suspend interrupt
1. usb_res：USB resume interrupt。
0. usb_sof： USB Start Of Frame interrupt

When ctl_rec, rx_rdy or tx_rdy triggered, need to query EP_STATUS register for detail information.

EP_STATUS_IN (set by hardware and cleared by software)

EP_STATUS	7	6	5	4	3	2	1	0
0x080e	EP7	EP6	EP5	EP4	EP3	EP2	EP1	sudat

Table 77 EP_STATUS Register

7. EP7：indicate tx_rdy is triggered by endpoint 7. (IN)
6. EP6：indicate tx_rdy is triggered by endpoint 6. (IN)
5. EP5：indicate tx_rdy is triggered by endpoint 5. (IN)
4. EP4：indicate tx_rdy is triggered by endpoint 4. (IN)
3. EP3：indicate tx_rdy is triggered by endpoint 3. (IN)
2. EP2：indicate tx_rdy is triggered by endpoint 2. (IN)
1. EP1：indicate tx_rdy is triggered by endpoint 1. (IN)
0. Reserved

EP_STATUS_OUT (set by hardware and cleared by software)

EP_STATUS	7	6	5	4	3	2	1	0
0x80F	EP7	EP6	EP5	EP4	EP3	EP2	EP1	sudat

Table 78 EP_STATUS Register

7. EP7：indicate rx_rdy is triggered by endpoint 7. (OUT)
6. EP6：indicate rx_rdy is triggered by endpoint 6. (OUT)
5. EP5：indicate rx_rdy is triggered by endpoint 5. (OUT)
4. EP4：indicate rx_rdy is triggered by endpoint 4. (OUT)
3. EP3：indicate rx_rdy is triggered by endpoint 3. (OUT)
2. EP2：indicate rx_rdy is triggered by endpoint 2. (OUT)
1. EP1：indicate rx_rdy is triggered by endpoint 1. (OUT)
0. Sudat: indicate that 8 bytes set up package arrived

CONFIDENTIAL

Downloaded From Oneyac.com

USBINT1 interrupt register, set by hardware and cleared by software（write 1 to clear it）.

USB INT1	7	6	5	4	3	2	1	0
0x080b	bad_token	crc16_err	overtime	pid_err	/	/	/	/

Table 79 USBINT1 Interrupt Register

7. bad_token: unsupported token received

6. crc16_err : the package received crc16 check error

5. overtime ： timeout interrupt(no data received after OUT token or no ACK received after IN token)

4. pid_err ： endpoint1-7 transfer PID error interrupt

USB_EN0, USB_EN1 interrupt enable register (only can be read or write by software)

USB_EN0	7	6	5	4	3	2	1	0
0x080c	ctl_rec_en	ctl_send_en	rx_rdy_en	tx_rdy_en	usb_rst_en	usb_sus_en	usb_res_en	usb_sof_en

Table 80 USB_EN0 Interrupt Enable Register

7. ctl_rec_en : data received on control endpoint 0 interrupt enable bit

6. ctl_send_en : data sent on control endpoint 0 interrupt enable bit

5. rx_rdy_en ： data received on endpoint 1-7 interrupt enable bit

4. tx_rdy_en ： data sent on endpoint 1-7 interrupt enable bit

3. usb_reset_en: USB Reset interrupt enable bit

2. usb_sus_en ： USB suspend interrupt enable bit

1. usb_res_en ： USB Resume interrupt enable bit

0. usb_sof_en ： USB Start Of Frame interrupt enable bit

CONFIDENTIAL

Downloaded From Oneyac.com

USB_EN1	7	6	5	4	3	2	1	0
0x080d	bad_token_en	crc16_err_en	overtime_en	pid_err_en	/	/	/	/

Table 81 USB_EN1Interrupt Enable Register

7. bad_token_en: unsupported token received interrupt enable bit

6. crc16_err_en : the package received crc16 check error interrupt enable bit

5. overtime_en ： timeout interrupt enable bit

4. pid_err_en ： endpoint1-7 transfer PID error interrupt enable bit

8.5. FIFO

It has been described that how to configure the register above. The next will depict how to use their register and how to operate them.

8.5.1. FIFO SFR register

(1) EP_RDY ： (endpoint ready register)

EP_RDY	7	6	5	4	3	2	1	0
0x0821	ep7_rdy	ep6_rdy	ep5_rdy	ep4_rdy	ep3_rdy	ep2_rdy	ep1_rdy	ep0_rdy

Table 82 FIFO EP_RDY Register

Epn_rdy (n=1-7): endpoint n is ready for transferring USB data now.\

Cleared by hardware and set by software.

Note: Ep0_rdy is not same with Epn_rdy. It will be forced to 1 by hardware when setup token coming to receive 8 bytes setup request. (setup has the highest priority for USB protocol)

When Epn_rdy=0, device will send back NACK pakage for PC's IN/OUT request to indicate not ready now.

(2) FIFO capacity counters

If one endpoint has been configured as IN direction, software need write the length number into the FIFO capacity register to tell USB the package length need send.

CONFIDENTIAL

Downloaded From Oneyac.com

When one port configured as OUT direction, software can read out the package length from the counter register once one package received successfully. (The unit is byte)

Every endpoint use 2 bytes register, so total 16 registers are occupied which are descripted as follow:

CNTn : the lower 8 bits FIFO counter register of endpoint n

CNTn	7	6	5	4	3	2	1	0
	counter [7:0]							

Table 83 FIFO lower 8 bits counter register

CNTn_HBIT: the upper 2 bits FIFO counter register of endpoint n

CNTn_ HBIT	7	6	5	4	3	2	1	0
	/						counter [9:8]	

Table 84 FIFO upper 2 bits counter register

All the 16 registers address：

CNT0 [0x0823]
CNT0_HBIT [0x082b]
CNT1 [0x0824]
CNT1_HBIT [0x082c]
CNT2 [0x0825]
CNT2_HBIT [0x082d]
CNT3 [0x0826]
CNT3_HBIT [0x082e]
CNT4 [0x0827]
CNT4_HBIT [0x082f]
CNT5 [0x0828]
CNT5_HBIT [0x0830]
CNT6 [0x0829]
CNT6_HBIT [0x0831]
CNT7 [0x082a]
CNT7_HBIT [0x0832]

(3) EP_HALT(endpoint suspend register)

EP_ HALT	7	6	5	4	3	2	1	0
0x0820	ep7 _halt	ep6 _halt	ep5 _halt	ep4 _halt	ep3 _halt	ep2 _halt	ep1 _halt	ep0 _halt

Table 85 FIFO EP_HALT Register

Downloaded From Oneyac.com

epn_halt: the suspend flag of endpoint n. 1 indicates the endpoint has been suspended and this endpoint is not available now. This endpoint will send back STALL when IN/OUT token received to indicate it is not available now. (epn_halt can only be read/written by software except ep0_halt)

ep0_halt can be cleared by hardware. According to USB protocol, ep0_halt is cleared by hardware when setup token received to avoid that device can not receive control information.

8.5.2. FIFO Access

The access to FIFO is very simple for BK2535. Software can read or write the 2K EXRAM directly with MOVX instruction and without any register interface or control logic.

When using C language, you only need to initialize a start address for one endpoint FIFO which should be consistent with the address configured in endpoint register.

For example: the address of endpoint 1

addr[10:0]={EP_ADDR_MSB.1, CFG_EP1_1[1:0], CFG_EP1_0 }

=

$$\mathbf{EP\_ADDR\_MSB.1}\times 2^{10}+\mathbf{CFG\_EP1\_1.1}\times 2^{9}+\mathbf{CFG\_EP1\_1.0}\times 2^{8}+\mathbf{CFG\_EP1\_0}$$

This 11 bits address can cover all the 2K EXRAM space from 0 to 0x7FF.

8.5.3. FIFO Operation

The above describe how to access the EXRAM by MCU, and, the USB part need access the EXRAM also. It will be explained next.

Accessing EXRAM by USB is implemented by DMA controller, and it is transparent to software.

According protocol, the host send out SETUP, IN and OUT token to request device transfer. The device would start to transfer data after software inform device the relevant endpoint is “ready”. The DMA controller will write the received data into the FIFO assigned in the EXRAM (out endpoint), or read out the data that needed send to the host from FIFO (IN endpoint).

What time is ready? From software view, there are two cases:

The software had written the data needed send to host into FIFO. It is ready to send now.(IN)

The software had read out the data received from host from FIFO. It is ready to

CONFIDENTIAL

Downloaded From Oneyac.com

receive now.(OUT)

When it is ready, Software can set the corresponding EP_RDY to indicate it is ready now, and then USB will start to work automatically.

8.6. Device Address

The 7 bits function address is stored in FADDR register. The address is set by host through SET_ADDRESS command. The software should write the 7 bits address into FADDR after received this command. The address will act immediately after received SET_ADDRESS command. USB only can accept the data or token send to this address.

Device address(R/W by software only)

FUNCT_ADDR	7	6	5	4	3	2	1	0
0x0822	/	function_addr [6:0]						

Table 86 device address register

8.7. Frame number register

FRAM_NO_0 : Frame number lower 8 bits （write by hardware, read by software only）

FRAM_NO_0	7	6	5	4	3	2	1	0
0x0808	Frame number [7:0]							

Table 87 FRAM_NO_0 lower 8 bits register

FRAM_NO_1: Frame number upper 3 bits, （write by hardware, read by software only）

FRAM_NO_1	7	6	5	4	3	2	1	0
0x0809	/					Frame number [10:8]		

Table 88 FRAM_NO_0 upper 3 bits register

8.8. USB power management

USB_PWR_CN ： USB power control register

USB_PWR_CN	7	6	5	4	3	2	1	0
0x0841	pu_en	DN	DP	/		usb_rst	usb_sus	remote_wakeup

Table 89 USB power control register

CONFIDENTIAL

Downloaded From Oneyac.com

Pu_en: PULL UP enable, D+（dp）pull up enable in chip. When it is disabled, device disconnect with outside circuit.

DN: indicate D+ logic level（can used to debug）.（read only）

DP : indicate D- logic level（can used to debug）.（read only）

USB_sus： USB module will enter low-power mode when write 1 into it. The USB protocol engineer is stopped and no response to outside. It is used as suspend state usually in USB protocol.（can R/W by software）

remote_wakeup： according USB protocol, the device with remote wakeup function can send wake up signal to host.（R/W） When it is set to 1, USB force D+ and D- into K state, and release it when clear it.

8.9. USB debug mode register

FRAM_NO_1	7	6	5	4	3	2	1	0
0x0843	MOD_ctrl7	MOD_ctrl6						

Table 90 USB debug register

MOD_ctrl7: Set this bit will change the register usage from 0X823 to 0X842. All the register would change to RW mode only for IN mode。 It can be used to check the values which write into the register.

MOD_ctrl6: force to clear the FIFO capacity counter register to zero.

8.10. USB RESET

FRAM_NO_1	7	6	5	4	3	2	1	0
0x0843	Reserved							USB_RST

Table 91 USB RESET register

USB_RST: USB module will be reset when write 1 into it.

8.11. Endpoint Buffer

For a transfer without buffer, it is described as follow: (IN direction)

MCU write the first package into the endpoint buffer and set relevant EP_RDY.

Wait transfer command from host, and send out interrupt when transfer is done.

MCU responds to interrupt and enter into relevant interrupt application. Then

Downloaded From Oneyac.com

write the next package into the buffer and set EP_RDY.

Wait transfer command from host, and recurrence as described before.

The USB bandwidth utilize efficiency is the main disadvantage for this transfer mode. The host should wait when MCU wrote data into FIFO, and MCU should wait when USB sent data out.

For this, multi-buffer mechanism is applied in BK2535. MCU can write next package into FIFO when the current package is sending. So, when transfer command coming, the data can be sent immediately.

For example, 2-buffer is implemented as follows:

Configure EP1 as IN endpoint, the capacity of EP1 is 64byte.

Configure the start FIFO address of EP1 as 0x500 and depth is 0x40.

Write the first package into 0x0500-0x0540, and then set EP_RDY register to indicate the data is ready.

At once, write the next package into 0x0540-0x0580 and wait the send out interrupt coming.

When the first package transfer complete, configure the start address of EP1 as 0x540, and then set EP_RDY to indicate the data is ready.

When the send out interrupt come, back to step 1.

Like this, 3 4 5 …-buffer is also can be implemented.

CONFIDENTIAL

Downloaded From Oneyac.com

9. Development and download

The BK2535 have some different development and download methods. The working mode is decided by the MODE pin voltage when power up. The next table describes the different working mode.

MODE pin voltage	mode	description	Note
0.9+-0.3	DEBUG mode	GPIO mapping to BIRD interface used to debug on chip. At this mode, the program can be loaded to the chip from JTAG interface.	P0.4=TDO P0.5=TDI P0.6=TMS P0.7=TCK
3+-0.3	FLASH download mode	At this mode, you can download the program into BK2535 through SPI interface.	P0.4=MOSI P0.5= MISO P0.6= SCK P0.7= CS
0+-0.3	Normal mode (product mode)	At this mode, BK2535 run the program from the FLASH directly.	

Table 92 work mode selection

CONFIDENTIAL

Downloaded From Oneyac.com

10. BK2535 RF transceiver

10.1. General Description

A RF transceiver (BK-RF) is embedded in BK2535, and the BK-RF is a high performance IP of Beken corporation.

BK-RF is a GFSK transceiver operating in the world wide ISM frequency band at 2400-2483.5 MHz. The transceiver has burst mode transmission and up to 2Mbps air data rate make it suitable for applications requiring ultra low power consumption. The embedded packet processing engines enable their full operation with a very simple MCU as a radio system. Auto re-transmission and auto acknowledge give reliable link without any MCU interference.

The BK-RF operates in TDD mode, either as a transmitter or as a receiver.

The RF channel frequency determines the center of the channel used by BK-RF. The frequency is set by the RF_CH register in register bank 0 according to the following formula: F0= 2400 + RF_CH (MHz). The resolution of the RF channel frequency is 1MHz.

A transmitter and a receiver must be programmed with the same RF channel frequency to be able to communicate with each other.

The output power of BK-RF is set by the RF_PWR bits in the RF_SETUP register.

Demodulation is done with embedded data slicer and bit recovery logic. The air data rate can be programmed to 1Mbps or 2Mbps by RF_DR register. A transmitter and a receiver must be programmed with the same setting.

In the following chapters, all registers are in register bank 0 except with explicit claim.

CONFIDENTIAL

Downloaded From Oneyac.com

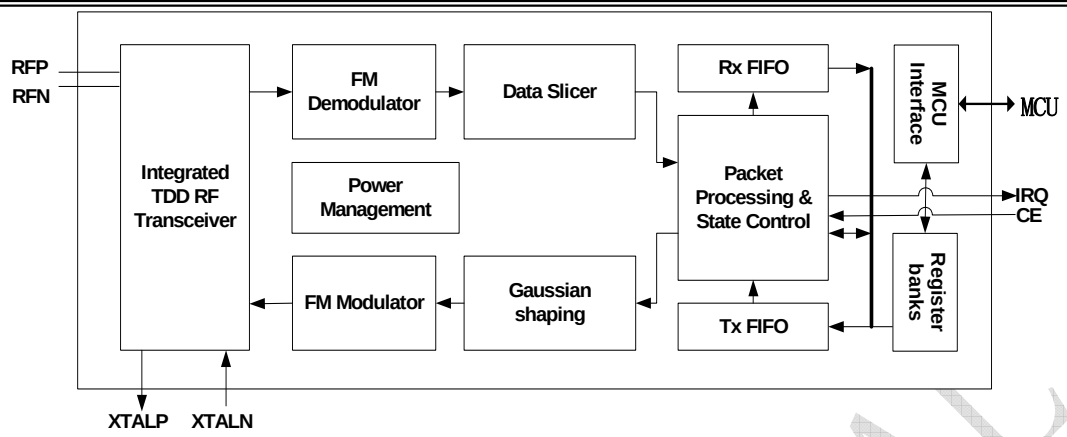


Figure 32 BK2535 RF Block Diagram

CONFIDENTIAL

Downloaded From Oneyac.com

10.2. Abbreviations

ACK	Acknowledgement
ARC	Auto Retransmission Count
ARD	Auto Retransmission Delay
CD	Carrier Detection
CE	Chip Enable
CRC	Cyclic Redundancy Check
CSN	Chip Select Not
DPL	Dynamic Payload Length
FIFO	First-In-First-Out
GFSK	Gaussian Frequency Shift Keying
GHz	Gigahertz
LNA	Low Noise Amplifier
IRQ	Interrupt Request
ISM	Industrial-Scientific-Medical
LSB	Least Significant Bit
MAX_RT	Maximum Retransmit
Mbps	Megabit per second
MCU	Microcontroller Unit
MHz	Megahertz
MISO	Master In Slave Out
MOSI	Master Out Slave In
MSB	Most Significant Bit
PA	Power Amplifier
PID	Packet Identity Bits
PLD	Payload
PRX	Primary RX
PTX	Primary TX
PWD_DWN	Power Down
PWD_UP	Power Up
RF_CH	Radio Frequency Channel
RSSI	Received Signal Strength Indicator
RX	Receive
RX_DR	Receive Data Ready
SCK	SPI Clock
SPI	Serial Peripheral Interface
TDD	Time Division Duplex
TX	Transmit
TX_DS	Transmit Data Sent
XTAL	Crystal

CONFIDENTIAL

Downloaded From Oneyac.com

10.3. State Control

10.3.1. State Control Diagram

BK-RF has built-in state machines that control the state transition between different modes.

When auto acknowledge feature is disabled, state transition will be fully controlled by MCU.

- Internal signal: POR,VDD
- SPI register: CE, PWR_UP, PRIM_RX, EN_AA, NO_ACK, ARC, ARD
- System information: Time out, ACK received, ARD elapsed, ARC_CNT, TX FIFO empty, ACK packet transmitted, Packet received

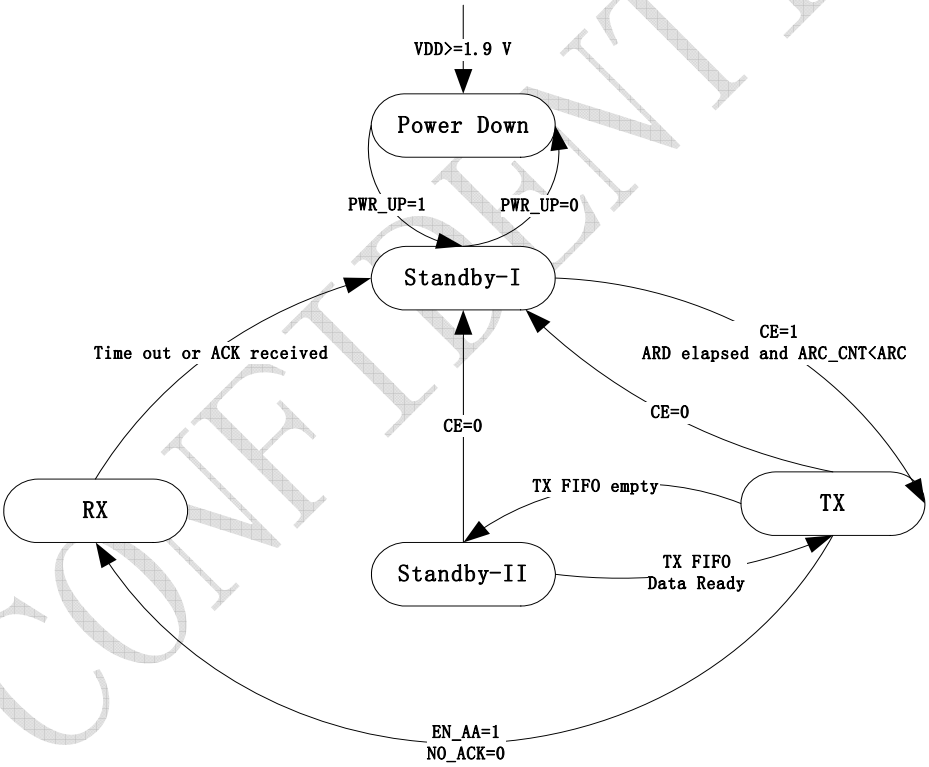


Figure 33 PTX (PRIM_RX=0) state control diagram

CONFIDENTIAL

Downloaded From Oneyac.com

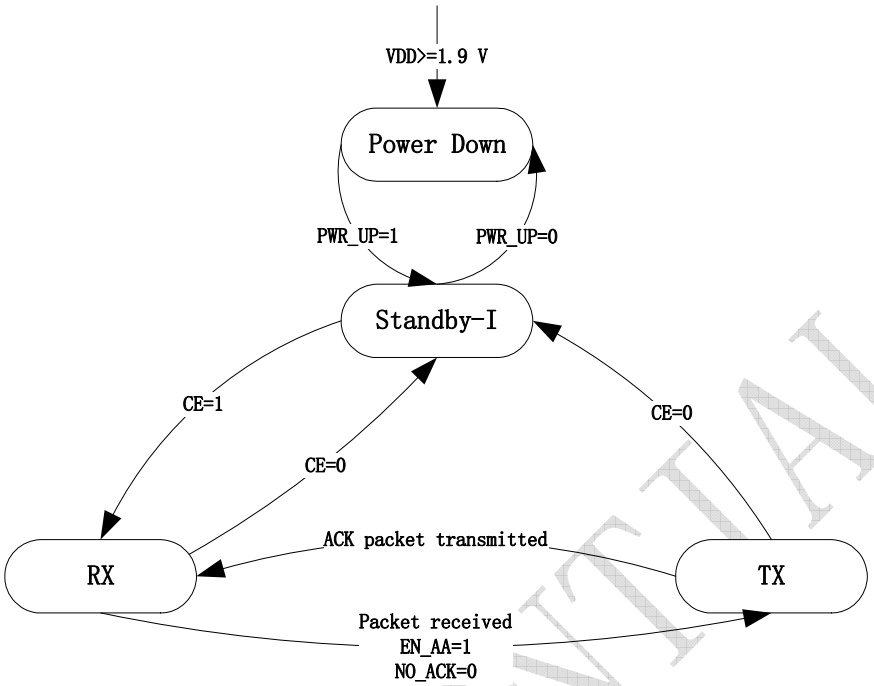


Figure 34 PRX (PRIM_RX=1) state control diagram

CONFIDENTIAL

10.3.2. Power down Mode

In power down mode the BK-RF is in sleep mode with minimal current consumption. SPI interface is still active in this mode, and all register values are available by SPI. Power down mode is entered by setting the PWR_UP bit in the CONFIG register to low.

10.3.3. Standby-I Mode

By setting the PWR_UP bit in the CONFIG register to 1 and de-asserting CE to 0, the device enters standby-I mode. Standby-I mode is used to minimize average current consumption while maintaining short start-up time. In this mode, part of the crystal oscillator is active. This is also the mode which the BK-RF returns to from TX or RX mode when CE is set low.

10.3.4. Standby-II Mode

In standby-II mode more clock buffers are active than in standby-I mode and much more current is used. Standby-II occurs when CE is held high on a PTX device with empty TX FIFO. If a new packet is uploaded to the TX FIFO in this mode, the device will automatically enter TX mode and the packet is transmitted.

Downloaded From Oneyac.com

10.3.5. TX Mode

- PTX device (PRIM_RX=0)

The TX mode is an active mode where the PTX device transmits a packet. To enter this mode from power down mode, the PTX device 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 10μs.

The PTX device stays in TX mode until it finishes transmitting the current packet. If CE = 0 it returns to standby-I mode. If CE = 1, the next action is determined by the status of the TX FIFO. If the TX FIFO is not empty the PTX device remains in TX mode, transmitting the next packet. If the TX FIFO is empty the PTX device goes into standby-II mode.

If the auto retransmit is enabled (EN_AA=1) and auto acknowledge is required (NO_ACK=0), the PTX device will enter TX mode from standby-I mode when ARD elapsed and number of retried is less than ARC.

- PRX device (PRIM_RX=1)

The PRX device will enter TX mode from RX mode only when EN_AA=1 and NO_ACK=0 in received packet to transmit acknowledge packet with pending payload in TX FIFO.

10.3.6. RX Mode

- PRX device (PRIM_RX=1)

The RX mode is an active mode where the BK-RF radio is configured to be a receiver. To enter this mode from standby-I mode, the PRX device must have the PWR_UP bit set high, PRIM_RX bit set high and the CE pin set high. Or PRX device can enter this mode from TX mode after transmitting an acknowledge packet when EN_AA=1 and NO_ACK=0 in received packet.

In this mode the receiver demodulates the signals from the RF channel, constantly presenting the demodulated data to the packet processing engine. The packet processing engine continuously 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 FIFO. If the RX FIFO is full, the received packet is discarded.

The PRX device remains in RX mode until the MCU configures it to standby-I mode or power down mode.

In RX mode a carrier detection (CD) signal is available. The CD is set to high when a RF signal is detected inside the receiving frequency channel. The internal

CONFIDENTIAL

Downloaded From Oneyac.com

CD signal is filtered before presented to CD register. The RF signal must be present for at least 128 μs before the CD is set high.

- PTX device (PRIM_RX=0)

The PTX device will enter RX mode from TX mode only when EN_AA=1 and NO_ACK=0 to receive acknowledge packet.

10.4. Packet Processing

10.4.1. Packet Format

The packet format has a preamble, address, packet control, payload and CRC field.

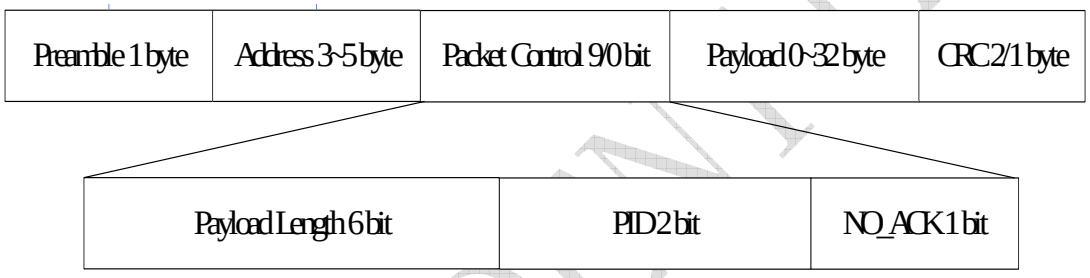


Figure 35 Packet Format

Preamble

The preamble is a bit sequence used to detect 0 and 1 levels in the receiver. 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.

Address

This is the address for the receiver. An address ensures that the packet is detected by the target receiver. The address field can be configured to be 3, 4, or 5 bytes long by the AW register.

The PRX device can open up to six data pipes to support up to six PTX devices with unique addresses. All six PTX device addresses are searched simultaneously. In PRX side, 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.

Downloaded From Oneyac.com

Each pipe can have up to 5 bytes configurable address. Data pipe 0 has a unique 5 byte address. Data pipes 1-5 share the 4 most significant address bytes. The LSB byte must be unique for all 6 pipes.

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.

On the PRX the RX_ADDR_Pn, defined as the pipe address, must be unique. On the PTX the TX_ADDR must be the same as the RX_ADDR_P0 on the PTX, and as the pipe address for the designated pipe on the PRX.

No other data pipe can receive data until a complete packet is received by a data pipe that has detected its address. When multiple PTX devices are transmitting to a PRX, the ARD can be used to skew the auto retransmission so that they only block each other once.

Packet Control

When Dynamic Payload Length function is enabled, 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

The payload length field is only used if the Dynamic Payload Length function is enabled.

PID

The 2 bit PID field is used to detect whether the received packet is new or retransmitted. PID prevents the PRX device 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 fields are used by the PRX device to determine whether a packet is old 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, BK-RF 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_ACK

The NO_ACK 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.

CONFIDENTIAL

Downloaded From Oneyac.com

The PTX can set the NO_ACK flag bit in the Packet Control Field with the 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 and the PRX does not transmit an ACK packet when it receives the packet.

Payload

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

The BK-RF provides two alternatives for handling payload lengths, static and dynamic payload length. The static payload length of each of six data pipes can be individually set.

The default alternative 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. 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. Each pipe has its own payload length.

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 for a system with different payload lengths it is not necessary to scale the packet length to the longest payload.

With DPL feature the BK-RF can decode the payload length of the received packet automatically instead of using the RX_PW_Px registers. The MCU can read the length of the received payload by using the command: R_RX_PL_WID.

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

CRC

The CRC is the error detection mechanism in the packet. The number of bytes in the CRC is set by the CRCO bit in the CONFIG register. It may be either 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 is 0xFF.

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

CONFIDENTIAL

Downloaded From Oneyac.com

No packet is accepted by receiver side if the CRC fails.

10.4.2. Packet Handling

BK-RF uses burst mode for payload transmission and receive.

The transmitter fetches payload from TX FIFO, automatically assembles it into packet and transmits the packet in a very short burst period with 1Mbps or 2Mbps air data rate.

After transmission, if the PTX packet has the NO_ACK flag set, BK-RF sets TX_DS and gives an active low interrupt IRQ to MCU. If the PTX is ACK packet, the PTX needs receive ACK from the PRX and then asserts the TX_DS IRQ.

The receiver automatically validates and disassembles received packet, if there is a valid packet within the new payload, it will write the payload into RX FIFO, set RX_DR and give an active low interrupt IRQ to MCU.

When auto acknowledge is enabled (EN_AA=1), the PTX device will automatically wait for acknowledge packet after transmission, and re-transmit original packet with the delay of ARD until an acknowledge packet is received or the number of re-transmission exceeds a threshold ARC. If the later one happens, BK-RF will set MAX_RT and give an active low interrupt IRQ to MCU. Two packet loss counters (ARC_CNT and PLOS_CNT) are incremented each time a packet is lost. The ARC_CNT counts the number of retransmissions for the current transaction. The PLOS_CNT counts the total number of retransmissions since the last channel change. ARC_CNT is reset by initiating a new transaction. PLOS_CNT is reset 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 PTX device will retransmit if its RX FIFO is full but received ACK frame has payload.

As an alternative for PTX device to auto retransmit it is possible to manually set the BK-RF to retransmit a packet a number of times. This is done by the REUSE_TX_PL command.

When auto acknowledge is enabled, the PRX device will automatically check the NO_ACK field in received packet, and if NO_ACK=0, it will automatically send an acknowledge packet to PTX device. If EN_ACK_PAY is set, and the acknowledge packet can also include pending payload in TX FIFO.

CONFIDENTIAL

Downloaded From Oneyac.com

10.5. Data and Control Interface

10.5.1. TX/RX FIFO

The data FIFOs are used to store payload that is to be transmitted (TX FIFO) or payload that is received and ready to be clocked out (RX FIFO). The FIFO is accessible in both PTX mode and PRX mode.

There are three levels 32 bytes FIFO for both TX and RX, supporting both acknowledge mode or no acknowledge mode with up to six pipes.

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

Both FIFOs have a controller and are accessible by using dedicated SPI commands. A TX FIFO in PRX can store payload 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 by using the FLUSH_TX command.

The RX FIFO in PRX may contain payload from up to three different PTX devices.

A TX FIFO in PTX can have up to three payloads stored.

The TX FIFO can be written to by three commands, W_TX_PAYLOAD and W_TX_PAYLOAD_NO_ACK in PTX mode and W_ACK_PAYLOAD in PRX mode. All three commands give access to the TX_PLD register.

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

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

In the FIFO_STATUS register it is possible to read if the TX and RX FIFO are full or empty. The TX_REUSE bit is also available in the FIFO_STATUS register. TX_REUSE is set by the command REUSE_TX_PL, and is reset by the command: W_TX_PAYLOAD or FLUSH TX.

10.5.2. Interrupt

In BK2535-RF there is an active low interrupt (IRQ), which is activated when TX_DS IRQ, RX_DR IRQ or MAX_RT IRQ are set high by the state machine in the STATUS register. The IRQ resets when MCU writes '1' to the IRQ source bit

CONFIDENTIAL

Downloaded From Oneyac.com

in the STATUS register. The IRQ mask in the CONFIG register is used to select the IRQ sources that are allowed to assert the IRQ. By setting one of the MASK bits high, the corresponding IRQ source is disabled. By default all IRQ sources are enabled.

The 3 bit pipe information in the STATUS register is updated during the IRQ high to low transition. If the STATUS register is read during an IRQ high to low transition, the pipe information is unreliable.

CONFIDENTIAL

Downloaded From Oneyac.com

10.6. RF Command

The RF commands are shown in the table:

Command name	Command word (binary)	# Data bytes	Operation
R_REGISTER	Read directly		
W_REGISTER	Write directly		
W_ANALOG_REG	Write through register0X8B8-0X8BC		
R_RX_PAYLOAD	8'b01000000	1 to 32 LSB byte 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	8'b01100000	1 to 32 LSB byte first	Write TX-payload: 1 – 32 bytes. A write operation always starts at byte 0 used in TX payload. This command used for ENABLE_ACK payload
FLUSH_TX	8'b10100000	0	Flush TX FIFO, used in TX mode
FLUSH_RX	8'b10000000	0	Flush RX FIFO, used in RX mode Should not be executed during transmission of acknowledge, that is, acknowledge package will not be completed.
REUSE_TX_PL	8'b00010000	0	Used for a PTX device Reuse last transmitted payload. Packets are repeatedly retransmitted as long as CE is high. 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	Read register 0x8C4 directly		Read RX-payload width for the top R_RX_PAYLOAD in the RX FIFO.
W_ACK_PAYLOAD	8'b01101ppp	1 to 32 LSB byte first	Used in RX mode. Write Payload to be transmitted together with ACK packet on PIPE PPP.

CONFIDENTIAL

Downloaded From Oneyac.com

			(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	8'b01101000	1 to 32 LSB byte first	Used in TX mode. Disables AUTOACK on this specific packet.
NOP	8'b00000000	0	No Operation.

Table 93 RF command

CONFIDENTIAL

Downloaded From Oneyac.com

10.7. Register Map

There are two register groups in BK2535 that is digital register and analog register.

10.7.1. Digital Register

Address (Hex)	Mnemonic	Bit	Reset Value	Type	Description
0x0880	CONFIG				Configuration Register
	Reserved	7	0	R/W	Only '0' allowed
	MASK_RX_DR	6	0	R/W	Mask interrupt caused by RX_DR 1: Interrupt not reflected on the IRQ pin 0: Reflect RX_DR as active low interrupt on the IRQ pin
	MASK_TX_DS	5	0	R/W	Mask interrupt caused by TX_DS 1: Interrupt not reflected on the IRQ pin 0: Reflect TX_DS as active low interrupt on the IRQ pin
	MASK_MAX_RT	4	0	R/W	Mask interrupt caused by MAX_RT 1: Interrupt not reflected on the IRQ pin 0: Reflect MAX_RT as active low interrupt on the IRQ pin
	EN_CRC	3	1	R/W	Enable CRC. Forced high if one of the bits in the EN_AA is high
	CRCO	2	0	R/W	CRC encoding scheme '0' - 1 byte '1' - 2 bytes
	PWR_UP	1	0	R/W	1: POWER UP, 0:POWER DOWN
	PRIM_RX	0	0	R/W	RX/TX control, 1: PRX, 0: PTX
0x0881	EN_AA				Enable 'Auto Acknowledgment' Function (only used by RX part) Need match with TX part
	Reserved	7:6	00	R/W	Only '00' allowed
	ENAA_P5	5	1	R/W	Enable auto acknowledgement data pipe 5
	ENAA_P4	4	1	R/W	Enable auto acknowledgement data pipe 4
	ENAA_P3	3	1	R/W	Enable auto acknowledgement data pipe 3
	ENAA_P2	2	1	R/W	Enable auto acknowledgement data pipe 2
	ENAA_P1	1	1	R/W	Enable auto acknowledgement data pipe 1
	ENAA_P0	0	1	R/W	Enable auto acknowledgement data pipe 0

CONFIDENTIAL

Downloaded From Oneyac.com

0x0882	EN_RXADDR				Enabled RX Addresses
	Reserved	7:6	00	R/W	Only '00' allowed
	ERX_P5	5	0	R/W	Enable data pipe 5.
	ERX_P4	4	0	R/W	Enable data pipe 4.
	ERX_P3	3	0	R/W	Enable data pipe 3.
	ERX_P2	2	0	R/W	Enable data pipe 2.
	ERX_P1	1	1	R/W	Enable data pipe 1.
	ERX_P0	0	1	R/W	Enable data pipe 0.
0x0883	SETUP_AW				Setup of Address Widths (common for all data pipes)
	Reserved	7:2	000000	R/W	Only '000000' allowed
	AW	1:0	11	R/W	RX/TX Address field width '00' - Illegal '01' - 3 bytes '10' - 4 bytes '11' - 5 bytes LSB bytes are used if address width is below 5 bytes
0x0884	SETUP_RETR				Setup of Automatic Retransmission
	ARD	7:4	0000	R/W	Auto Retransmission Delay '0000' – Wait 250 us '0001' – Wait 500 us '0010' – Wait 750 us …….. '1111' – Wait 4000 us (Delay defined from end of transmission to start of next transmission)
	ARC	3:0	0011	R/W	Auto Retransmission Count '0000' –Re-Transmit disabled '0001' – Up to 1 Re-Transmission on fail of AA …… '1111' – Up to 15 Re-Transmission on fail of AA
0x0885	RF_CH				RF Channel
	Reserved	7	0	R/W	Only '0' allowed
	RF_CH	6:0	0000010	R/W	Sets the frequency channel
0 x0886	RF_SETUP				RF Setup Register
	Reserved	7	0	R/W	Reserved
		6	0	R/W	Reserved
	En_250k_rate	5	0	R/W	Set RF data rate to 250k
		4	0		Reserved. please don't change it
	RF_DR	3	1	R/W	Air Data Rate ,decide by 0x886 bit {[5],[3]} '00' – 1Mbps '01' – 2Mbps '10' – 250kbps '11' – reserved
	RF_PWR[1:0]	2:1	11	R/W	Set RF output power in TX

CONFIDENTIAL

Downloaded From Oneyac.com

					mode RF_PWR[1:0] '00' – -10 dBm '01' – -5 dBm '10' – 0 dBm '11' – 5 dBm
	LNA_HCURR	0	1	R/W	Setup LNA gain 0:Low gain(20dB down) 1:High gain
0 x0887 0 x0888 0 x0889 0 x088A 0 x088B	RX_ADDR_P0	39:0	0xE7E7E7E7E7	R/W	Receive address data pipe 0. 5 Bytes maximum length. Write the number of bytes defined by SETUP_AW) {X88B,X88A,X889,X888,X887}
0 x088C 0 x088D 0 x088E 0 x088F 0 x0890	RX_ADDR_P1	39:0	0xC2C2C2C2C2	R/W	Receive address data pipe 1. 5 Bytes maximum length. Write the number of bytes defined by SETUP_AW)
0 x0891	RX_ADDR_P2	7:0	0xC3	R/W	Receive address data pipe 2. Only LSB. MSB bytes is equal to RX_ADDR_P1[39:8]
0 x0892	RX_ADDR_P3	7:0	0xC4	R/W	Receive address data pipe 3. Only LSB. MSB bytes is equal to RX_ADDR_P1[39:8]
0 x0893	RX_ADDR_P4	7:0	0xC5	R/W	Receive address data pipe 4. Only LSB. MSB bytes is equal to RX_ADDR_P1[39:8]
0 x0894	RX_ADDR_P5	7:0	0xC6	R/W	Receive address data pipe 5. Only LSB. MSB bytes is equal to RX_ADDR_P1[39:8]
0 x0895 0 x0896 0 x0897 0 x0898 0 x0899	TX_ADDR	39:0	0xE7E7E7E7E7	R/W	Transmit address. Used for a PTX device only. (LSB byte is written first) Set RX_ADDR_P0 equal to this address to handle automatic acknowledge if this is a PTX device
0 x089A	RX_PW_P0				
	Reserved	7:6	00	R/W	Only '00' allowed
	RX_PW_P0	5:0	000000	R/W	Number of bytes in RX payload in data pipe 0 (1 to 32 bytes). 0: not used 1 = 1 byte … 32 = 32 bytes
0 x089B	RX_PW_P1				
	Reserved	7:6	00	R/W	Only '00' allowed
	RX_PW_P1	5:0	000000	R/W	Number of bytes in RX payload in data pipe 1 (1 to 32 bytes). 0: not used 1 = 1 byte … 32 = 32 bytes
0 x089C	RX_PW_P2				
	Reserved	7:6	00	R/W	Only '00' allowed
	RX_PW_P2	5:0	000000	R/W	Number of bytes in RX payload

CONFIDENTIAL

Downloaded From Oneyac.com

					in data pipe 2 (1 to 32 bytes). 0: not used 1 = 1 byte … 32 = 32 bytes
0 x089D	RX_PW_P3				
	Reserved	7:6	00	R/W	Only '00' allowed
	RX_PW_P3	5:0	000000	R/W	Number of bytes in RX payload in data pipe 3 (1 to 32 bytes). 0: not used 1 = 1 byte … 32 = 32 bytes
0 x089E	RX_PW_P4				
	Reserved	7:6	00	R/W	Only '00' allowed
	RX_PW_P4	5:0	000000	R/W	Number of bytes in RX payload in data pipe 4 (1 to 32 bytes). 0: not used 1 = 1 byte … 32 = 32 bytes
0 x089F	RX_PW_P5				
	Reserved	7:6	00	R/W	Only '00' allowed
	RX_PW_P5	5:0	000000	R/W	Number of bytes in RX payload in data pipe 5 (1 to 32 bytes). 0: not used 1 = 1 byte … 32 = 32 bytes
0 x08A0	DYNPD				Enable dynamic payload length
	Reserved	7:6	0	R/W	Only ‘00’ allowed
	DPL_P5	5	0	R/W	Enable dynamic payload length data pipe 5. (Requires EN_DPL and ENAA_P5)
	DPL_P4	4	0	R/W	Enable dynamic payload length data pipe 4. (Requires EN_DPL and ENAA_P4)
	DPL_P3	3	0	R/W	Enable dynamic payload length data pipe 3. (Requires EN_DPL and ENAA_P3)
	DPL_P2	2	0	R/W	Enable dynamic payload length data pipe 2. (Requires EN_DPL and ENAA_P2)
	DPL_P1	1	0	R/W	Enable dynamic payload length data pipe 1. (Requires EN_DPL and ENAA_P1)
	DPL_P0	0	0	R/W	Enable dynamic payload length data pipe 0. (Requires EN_DPL and ENAA_P0)
0 x08A1	FEATURE			R/W	Feature Register

CONFIDENTIAL

Downloaded From Oneyac.com

	Reserved	7:3	0	R/W	Only '00000' allowed
	EN_DPL	2	0	R/W	Enables Dynamic Payload Length
	EN_ACK_PAY	1	0	R/W	Enables Payload with ACK
	EN_DYN_ACK	0	0	R/W	Enables the W_TX_PAYLOAD_NOACK command
0 x08A5 0 x08A4 0 x08A3 0 x08A2	(cfg0c0--3)	31:0	0		Please initialize with 0x00731200
0 x08A9 0 x08A8 0 x08A7 0 x08A6	NEW_FEATURE (cfg0d0--3)	31:0	0		Please initialize with 0x0080B436
0 x08B4 0 x08B3 0 x08B2 0 x08B1 0 x08B0 0 x08AF 0 x08AE 0 x08AD 0 x08AC 0 x08AB 0 x08AA	RAMP (2402table_0 --2401table_A)	87:0	NA	W	Ramp curve Please write with 0xFFFFFEF7CF208104082041
0 x08B5	BK-RF_ce				
		7:1			reserved
		0			ce
0 x08B6	BK-RF_cmd				8'b10000000 : Flush RX 8'b10100000 : Flush TX 8'b00010000 : Reusle TX PL 8'b01000000 : Read RX Payload 8'b01100000 : Write TX Payload 8'b01101ppp : W_ACK_PAYLOAD 8'b01101000 : W_TX_PAYLAOD_NOACK 8'b00000000 : NOP
0 x08B7	BK-RF_FIFO			R/W	TX MODE: TX data payload register 1 - 32 bytes. RX MODE: RX data payload register 1 - 32 bytes.
0 x08B8	BK-RF_sdata_0				Analog register0[7:0]
0 x08B9	BK-RF_sdata_1				Analog register1[15:8]
0 x08BA	BK-RF_sdata_2				Analog register2[23:16]
0 x08BB	BK-RF_sdata_3				Analog register3[31:24]
0 x08BC	BK-RF_sctrl				Write address of Analog rgister (only can be writen)
0 x08C0	BK-RF_status				Status, read only
	RX_DR	6	0	R/W	Data Ready RX FIFO interrupt Asserted when new data arrives RX FIFO Write 1 to clear bit.
	TX_DS	5	0	R/W	Data Sent TX FIFO interrupt Asserted when packet transmitted on TX. If AUTO_ACK is activated, this bit

CONFIDENTIAL

Downloaded From Oneyac.com

					is set high only when ACK is received. Write 1 to clear bit.
	MAX_RT	4	0	R/W	Maximum number of TX retransmits interrupt Write 1 to clear bit. If MAX_RT is asserted it must be cleared to enable further communication.
	RX_P_NO	3:1	111	R	Data pipe number for the payload available for reading from RX_FIFO 000-101: Data Pipe Number 110: Not used 111: RX FIFO Empty
	TX_FULL	0	0	R	TX FIFO full flag. 1: TX FIFO full 0: Available locations in TX FIFO
0 x08C1	BK-RF_observetx				Status, read only
	PLOS_CNT	7:4	0000	R	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.
	ARC_CNT	3:0	0000	R	Count retransmitted packets. The counter is reset when transmission of a new packet starts.
0 x08C2	BK-RF_cdstatus				Status, read only
	Reserved	7:1	000000	R	
	CD	0	0	R	Carrier Detect
0 x08C3	BK-RF_fifostatus				Status, read only
	Reserved	7	0	R/W	Only '0' allowed
	TX_REUSE	6	0	R	Reuse last transmitted data packet if set high. The packet is repeatedly retransmitted as long as CE is high. TX_REUSE is set by the SPI command REUSE_TX_PL, and is reset by the SPI command W_TX_PAYLOAD or FLUSH TX
	TX_FULL	5	0	R	TX FIFO full flag 1: TX FIFO full; 0: Available locations in TX FIFO
	TX_EMPTY	4	1	R	TX FIFO empty flag. 1: TX FIFO empty 0: Data in TX FIFO
	Reserved	3:2	00	R/W	Only '00' allowed
	RX_FULL	1	0	R	RX FIFO full flag 1: RX FIFO full 0: Available locations in RX FIFO
	RX_EMPTY	0	1	R	RX FIFO empty flag 1: RX FIFO empty 0: Data in RX FIFO
0 x08 C4	BK-RF_rpl_width				Status, read only

CONFIDENTIAL

Downloaded From Oneyac.com

					The width of the payload
0X8C5	BK-RF_mbist_st				Status, read only
	reseved			R	5'b0
	Done	2		R	test_done
	Pass	1		R	test_pass
	Fail	0		R	test_fail
0X8C6~0X8C7	Chip_id			R	
				R	Chip_id [7:0]
				R	Chip_id [15:8]
0X8C8~0X8CB	BK-RF_bit_cnt			R	Status, read only Total number of bits received register
0X8CC~0X8CF	BK-RF_err_cnt			R	Status, read only Error counter register
0X8D0~0X8D3	Tx_freq_offset			RW	RF MOD/DEMOD config
0X8D4~0X8D7	Rx_freq_offset			RW	RF MOD/DEMOD config
0X8D8[7:4]	Mod_sdm_dly			RW	RF MOD/DEMOD config
0X8D8[3:0]	Mod_dac_dly			RW	RF MOD/DEMOD config
0x8d9[7]	clksel_cfg			R/W	pll sdm output latch edge 1: posedge 0:negedge
0x8d9[6]	sdm3bit_cfg			R/W	sdm 2nd/3nd selection 1:3nd 0:2nd
0x8d9[5]	pn25ena_cfg			R/W	PN25 enable
0x8d9[4]	open_loop_en			R/W	1: FracN = 0
0x8d9[3]	rx_if_select			R/W	lo direction select 0:+500k 1: -500k
0x8d9[2]	tbfalcon_reset			R/W	sdm reset 1: reset
0x8d9[1]	bp_kmod			R/W	bypass kmod calibration
0x8d9[0]	vco_cal_en			R/W	vco calibriation enable
0x8da[0] 0x8db[7:0]	fm_gain			R/W	vco after calibriation value is set in this register
0x8da[1]	rx_if_select			R/W	tx mod direction select 0:+500k 1: -500k
0x8da[2]	rf_test_en			R/W	rf test enable if it is 1, then gpio3 and gpio4 output testsignal
0x8dc[0] 0x8dd[7:0]	fm_kmod_set			R/W	if bp_kmod is 1 auto channel compensation is stopped, this register value is the default value
0x8de[7:0] 0x8df[7:0]	mod_coefficient			R/W	tx N value compensation
0x8e0[7]	pwdRSSI			R/W	powerdown RSSI
0x8e0[6:4]	dsplpctrl			R/W	analog control
0x8e0[3]	p11_pusel			R/W	analog control
0x8e0[2]	p10_pusel			R/W	analog control
0x8e0[1]	PAD_DR			R/W	analog control

CONFIDENTIAL

Downloaded From Oneyac.com

0x8e0[0]	boost_mode			R/W	boost mode select 0: auto boost 1: digital control
0x8e1[7]	lnag			R/W	analog control
0x8e1[6:5]]	HQ			R/W	analog control
0x8e1[4:0]]	gPA			R/W	analog control
0x8e2[7:3]	lbd_thre			R/W	analog control
0x8e2[2:0]	lbd_hys			R/W	analog control
0x8e3[7]	pwdlbd			R/W	analog control
0x8e3[6]	lbd_intset			R/W	analog control
0x8e3[5]	TXCWEN			R/W	analog control
0x8e3[0]	samp_pad_c			R/W	analog control
0x8f9[7:3]	fltcal			R	analog indicator
0x8f9[2]	pll48_fast			R	analog indicator
0x8f9[1]	pll48_slow			R	analog indicator
0x8f9[0]	vco_amp_ind			R	analog indicator
0x8fa[7:0] 0x8fb[7:0]	chip_id			R	chip id/2535
0x8fc[7:0] 0x8fd[7:0] 0x8fe[7:0] 0x8ff[7:0]	device_id			R	device id
Note: Don't write reserved registers and registers at other addresses in register bank 0					

Table 94 Digital Register

Note:

1. ARD-auto retransmission delay. If the ACK payload is more than 15 byte in 2Mbps mode the ARD must be 500μS or more, if the ACK payload is more than 5byte in 1Mbps mode the ARD must be 500μS or more. In 250kbps mode (even when the payload is not in ACK) the ARD must be 500μS or more.
2. The RX_DR IRQ is asserted by a new packet arrival event. The procedure for handling this interrupt should be: 1) read payload from FIFO, 2) clear RX_DR IRQ, 3) read FIFO_STATUS to check if thereare more payloads available in RX FIFO, 4) if there are more data in RX FIFO, repeat from step 1).
3. Register 0x881 EN_AA only used for RX part now. For TX part, the command W_TX_PAYLOAD used for auto-ack payload, the command W_TX_PAYLOAD_NOACK used for disable-ack payload.

10.7.2. Analog Register

The analog registers can be written through writing 0X8B8 to 0X8BC.

Analog register data [31:0] = {0X8BB, 0X8Ba, 0X8B9, 0X8B8}

Downloaded From Oneyac.com

Analog register address [7:0] = {0X8BC}

Writing corresponding data and address into these serial registers can update the analog register value.

Please contact BEKEN FAE for the register setting used by the analog register.

10.7.3. TX power control setting

The transmit power can be set from -51dBm to 5dBm, to do this, please refer to the next table.

PALDO<1:0> Ana.Reg3<23:22>	PCsel Ana.Reg4<16>	HQ<1:0> Dig.61h<6:5>	gPA<4:0> dig.61h<4:0> Hex	TX Power (dBm)	RF Current (mA)
0	0	0	0	-54	4.4
0	0	0	1	-36	4.6
0	0	0	3	-33	4.7
0	0	0	4	-30	4.8
0	0	0	6	-27	4.9
0	1	1	1	-23	6
0	1	1	3	-18	6.2
0	1	1	4	-12	6.5
0	1	1	7	-9	6.9
0	1	1	B	-6	7.4
0	1	1	10	-3	8.3
0	1	1	1F	0	9.8
0	1	3	1F	3	12
3	1	3	1F	5	15.5

Table 95 TX power setting

10.7.4. PLL setting time

For some application, the PLL setting time is a important parameter for power saving. You can adjust the stable time by setting the Tx_settling_sel register value. For detail, please refer to the next table.

Tx_settling_sel<2:0> =[digital.add27h<1:0>,add26h<7>]:

.

Tx_settling_sel<2:0>	Pll settling time	Note

Downloaded From Oneyac.com

	(us)	
0	40	
1	50	
2	60	
3	70	
4	80	default
5	100	
6	120	Compatible with BK2533
7	130	Compatible with N

Table 96 PLL setting time

11. Electrical specifications

11.1. RF part

Name	Parameter (Condition)	Min	Typical	Max	Unit	Comment
	Operating Condition					
VDD	Voltage	1.8		3.6	V	
PSR						
TEMP	Temperature	-20		85	℃	
	Digital input Pin					
VIH	High level	VDD-0.3		VDD	V	
VIL	Low level	VSS		0.3VDD	V	
	Digital output Pin					
VOH	High level (IOH=-0.25mA)	VDD- 0.3		VDD	V	
VOL	Low level(IOL=0.25mA)	0		0.3	V	
	Normal condition					
IVDD	Power Down current			0.1	uA	RF part
IVDD	Standby-I current		30		uA	
IVDD	Standby-II current		150		uA	
	Normal RF condition					
FOP	Operating frequency	2400		2527	MHz	
FXTAL	Crystal frequency		16		MHz	
RFSK	Air data rate	0.25	1	2	Mbps	
	Transmitter					
PRF	Output power				dBm	
PBW	Modulation 20 dB bandwidth(2Mbps)		2.3		MHz	
PBW	Modulation 20 dB bandwidth (1Mbps)		1.2		MHz	
PBW	Modulation 20 dB bandwidth (250Kbps)		0.6		MHz	
PRF1	Out of band emission 2 MHz		-25		dBm	1MHz,RBW=100K TXPOWER=5dBm Maxhold
PRF2	Out of band emission 4 MHz		-40		dBm	
IVDD	Current at -36 dBm output power		4.6		mA	RF current
IVDD	Current at -30 dBm output		4.8		mA	

Downloaded From Oneyac.com

	power					
IVDD	Current at -22 dBm output power		6		mA	
IVDD	Current at -18 dBm output power		6.2		mA	
IVDD	Current at -12 dBm output power		6.5		mA	
IVDD	Current at -9 dBm output power		6.9		mA	
IVDD	Current at -6 dBm output power		7.4		mA	
IVDD	Current at -3dBm output power		8.3		mA	
IVDD	Current at 0 dBm output power		9.8		mA	
IVDD	Current at 3 dBm output power		12		mA	
IVDD	Current at 6 dBm output power		15.5		mA	
	Receiver					
IVDD	Current (2Mbps)		14		mA	
IVDD	Current (1Mbps)		13.5		mA	
Max Input	1 E-3 BER			10	dBm	
RXSENS	1 E-3 BER sensitivity (2Mbps)		-87		dBm	
RXSENS	1 E-3 BER sensitivity (1Mbps)		-90		dBm	
RXSENS	1 E-3 BER sensitivity (250Kbps)		-95		dBm	
C/ICO	Co-channel C/I (2Mbps)		10		dB	
C/I1ST	ACS C/I 2MHz (2Mbps)		6		dB	
C/I2ND	ACS C/I 4MHz (2Mbps)		-15		dB	
C/I3RD	ACS C/I 6MHz (2Mbps)		-27		dB	
C/ICO	Co-channel C/I (1Mbps)		7		dB	
C/I1ST	ACS C/I 1MHz (1Mbps)		4		dB	
C/I2ND	ACS C/I 2MHz (1Mbps)		-15		dB	
C/I3RD	ACS C/I 3MHz (1Mbps)		-25		dB	

11.2. MCU part

Name	Parameter (Condition)	Min	Typical（1.8V）	Max	Unit	Comment
	Core functions					
	Sleep mode (RCOSC 32k)		4		uA	
	Further sleep		2.5		uA	
	Supper sleep		2		uA	
	Deep sleep mode		0.6		uA	
	Idle mode at 16M		0.31		mA	
	Idle mode at 8M		0.21		mA	
	Idle mode at 4M		0.16		mA	
	Idle mode at XOSC32k(16M running)				mA	
	Active mode (16M)		3		mA	

CONFIDENTIAL

Downloaded From Oneyac.com

	Active mode (8M)		1.6		mA	
	Active mode (4M)		0.9		mA	
	Active mode (4M)		0.56		mA	
	FLASH					
	NVR read				mA	
	NVR write				mA	
	NVR erase				mA	
	ADC (8k byte rates)				uA	
	ADC SINAD (fin=1khz, fs=8khz)				DB	
	LBD (always on)				uA	
	USB				mA	
	GPIO					
	Drive ability		4	8	mA	

CONFIDENTIAL

Downloaded From Oneyac.com

12. Typical Application Schematic

Please refer to the separate documents for detail.

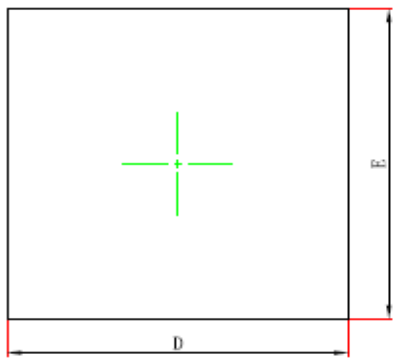
CONFIDENTIAL

Downloaded From Oneyac.com

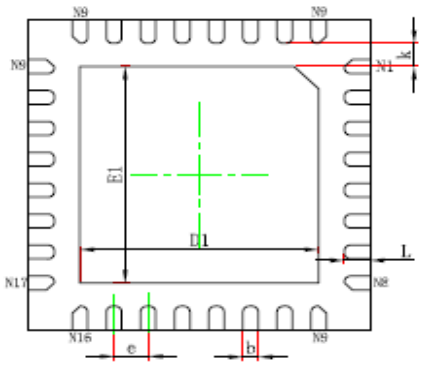
13. Package Information

QFN32-4X4

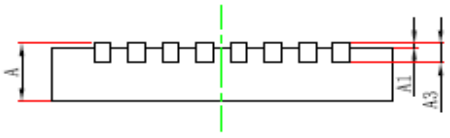
QFNWB4×4-32L-A (P0.40T0.75/0.85) PACKAGE OUTLINE DIMENSIONS



TOP VIEW



BOTTOM VIEW

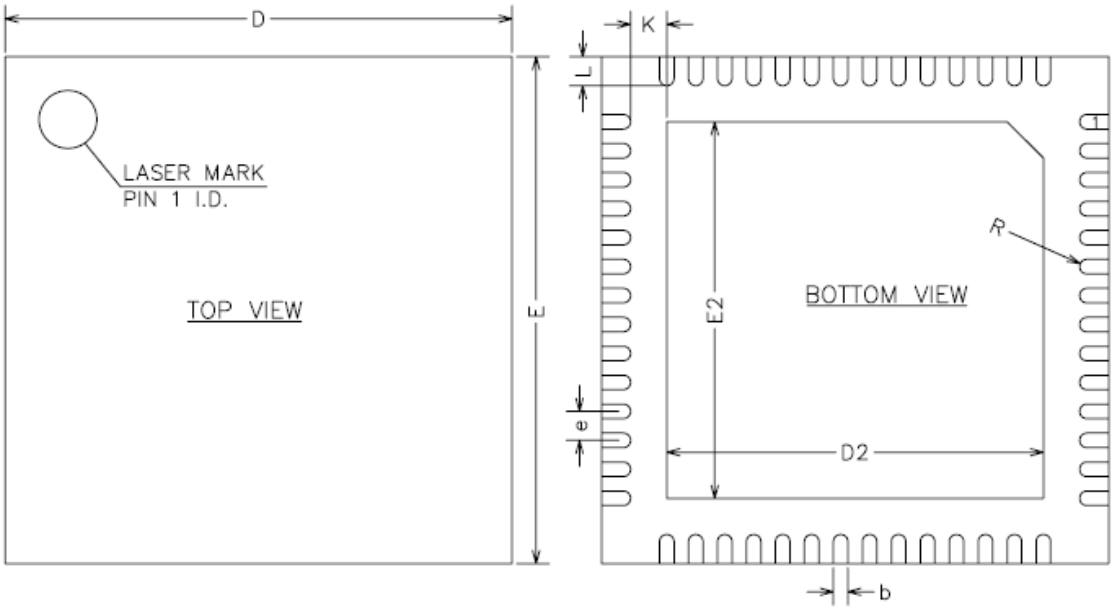


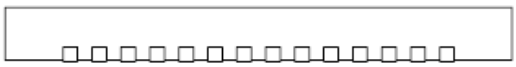
SIDE VIEW

Symbol	Dimensions In Millimeters		Dimensions In Inches	
	Min.	Max.	Min.	Max.
A	0.700/0.800	0.800/0.900	0.028/0.031	0.031/0.035
A1	0.000	0.050	0.000	0.002
A3	0.203REF.		0.008REF.	
D	3.924	4.076	0.154	0.160
E	3.924	4.076	0.154	0.160
D1	2.700	2.900	0.106	0.114
E1	2.700	2.900	0.106	0.114
k	0.200MIN.		0.008MIN.	
b	0.150	0.250	0.006	0.010
e	0.400TYP.		0.016TYP.	
L	0.224	0.376	0.009	0.015

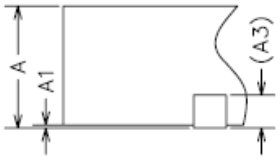
Downloaded From Oneyac.com

QFN56-7X7





SIDE VIEW



COMMON DIMENSIONS
(UNITS OF MEASURE=MILLIMETER)

SYMBOL	MIN	NOM	MAX
A	0.70	0.75	0.80
A1	0	0.02	0.05
A3		0.20REF	
b	0.15	0.20	0.25
D	6.90	7.00	7.10
E	6.90	7.00	7.10
D2	5.05	5.20	5.35
E2	5.05	5.20	5.35
e	0.30	0.40	0.50
K	0.20	–	–
L	0.35	0.40	0.45
R	0.09	–	–

CO

Downloaded From Oneyac.com

14. Solder Reflow Profile

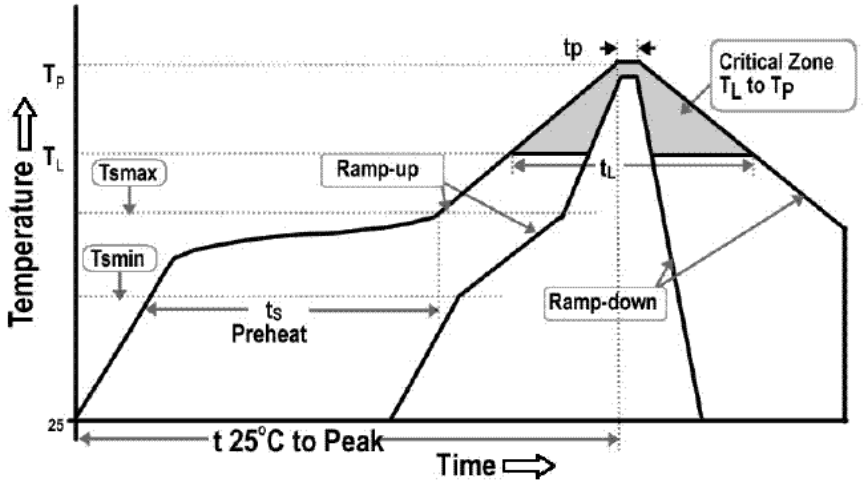


Figure 36 Classification Reflow Profile

Table 97 Solder Reflow Profile

Profile Feature		Specification
Average Ramp-Up Rate (tsmax to tp)		3°C/second max.
Pre_heat	Temperature Min (Tsmin)	150°C
	Temperature Max (Tsmax)	200°C
	Time (ts)	60-180 seconds
Time Maintained above	Temperature (TL)	217°C
	Time (tL)	60-150 seconds
Peak/Classification Temperature (Tp)		260°C
Time within 5°C of Actual Peak Temperature (tp)		20-40 seconds
Ramp-Down Rate 6		6°C/second max.
Time 25°C to Peak Temperature 8		8 minutes max.

CONFIDENTIAL

Downloaded From Oneyac.com

15. Contact Information

Beken Corporation Technical Support Center

Shanghai office
Building 41, 1387 Zhangdong Road, Zhangjiang High-Tech Park, Pudong New District, Shanghai, China
Phone: 86-21-51086811
Fax: 86-21-60871089
Postal Code: 201203
Email: info@bekencorp.com
Website: www.bekencorp.com

CONFIDENTIAL

Downloaded From Oneyac.com

单击下面可查看定价，库存，交付和生命周期等信息

>>Beken(博通集成电路)

Downloaded From Oneyac.com