

WK2124 SPI接口 四通道通用异步收发器 无铅封装



1. 产品概述

WK2124是SPI™接口的4通道UART器件，WK2124实现SPI桥接/扩展4个增强功能串口（UART）的功能。

扩展的子通道的UART具备如下功能特点：

每个子通道UART的波特率、字长、校验格式可以独立设置，最高可以提供2Mbps的通信速率。

每个子通道可以独立设置工作在IrDA红外通信。

每个子通道具备收/发独立的256 BYTE FIFO，FIFO的中断可按用户需求进行编程触发点。

WK2124采用SSOP20绿色环保的无铅封装，可以工作在2.5~5.0V的宽工作电压范围，具备可配置自动休眠/唤醒功能。

[注]：SPI™ 为MOTOROLA公司的注册商标。

2. 基本特性

2.1 总体特性

- 低功耗设计，可以配置自动休眠，自动唤醒模式（ μ S级唤醒）
- 超大硬件收发缓存，支持256级FIFO
- 宽工作电压设计，工作电压为2.5V~5.0V
- 精简的配置寄存器和控制字，操作简单可靠
- 提供工业级
- 高速CMOS工艺，子串口速率最高可达2Mbps@5V、1.5Mbps@3.3V、1Mbps@2.5V
- 采用符合绿色环保政策的SSOP20无铅封装

2.2 扩展子通道UART特性

- 子通道串口独立配置，高速、灵活：
 - 每个子串口为全双工，每个子串口可以通过软件开启/关闭
 - 波特率可以独立设置，子串口最高可以达到2M bit/s
 - 每个子串口字符格式包括数据长度、停止位数、奇偶校验模式可以独立设置
 - 完善的子串口状态查询功能
 - 可以实现对单个子串口软件复位
- FIFO功能：
 - 每个子串口具备独立的256级发送FIFO，发送FIFO触发点可按用户需求进行编程
 - 每个子串口具备独立的256接收FIFO，接收FIFO触发点可按用户需求进行编程
 - 软件FIFO使能和清空
 - FIFO状态和计数器输出
- 错误检测：
 - 支持奇偶校验错，数据帧错误、溢出错误及Line-Break错误检测
 - 支持起始位错误检测
- 中断特点：
 - 具备子串口接收FIFO超时中断
 - 支持Line-Break错误中断

- 内置符合SIR标准的IrDA红外收发编解码器，传输速度可达115.2K bit/s

2.3 SPI主接口特性

- 最高速度10M bit/s
- 仅支持SPI从模式
- SPI模式0
- 支持最长256字节连续收发

3. 应用领域

- 多串口服务器/多串口卡
- 工业/自动化现场RS-485控制
- 通过2G/3G/4G的无线数据传输
- 车载信息平台/车载GPS定位系统
- 远传自动抄表（AMR）系统
- POS/税控POS/金融机具
- DSP/嵌入式系统

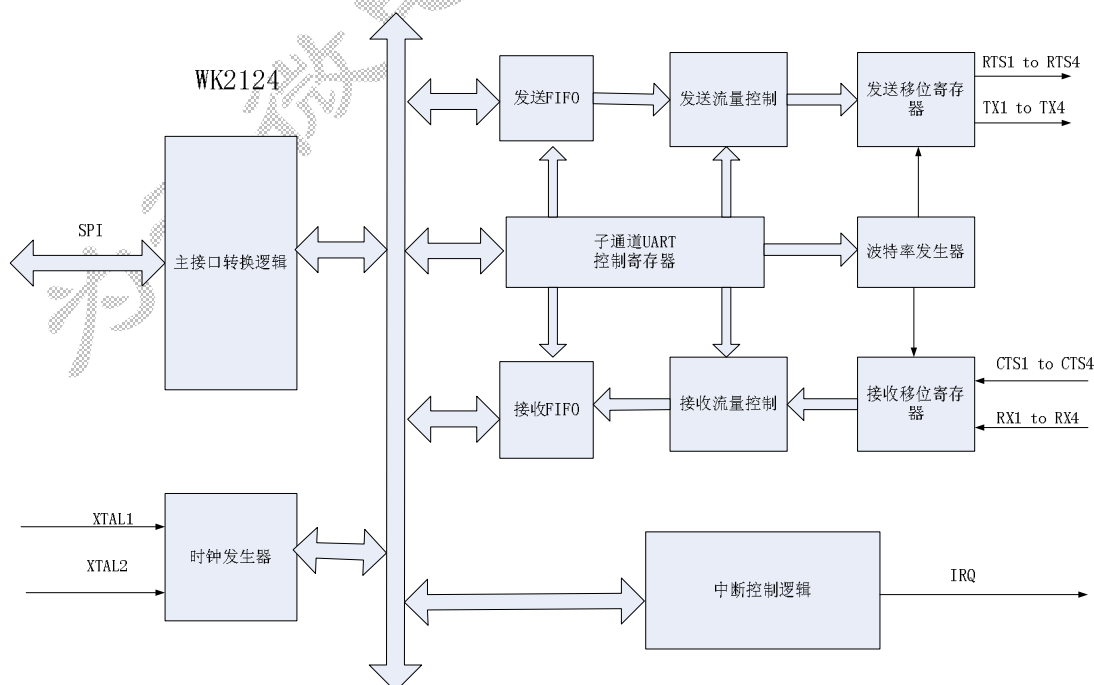
4. 订购信息

表4.1 WK2124 订购信息

产品型号	封装	说明
WK2124-ISSG	SSOP20 无铅封装	普通工业级；工作温度 -45℃~+85℃

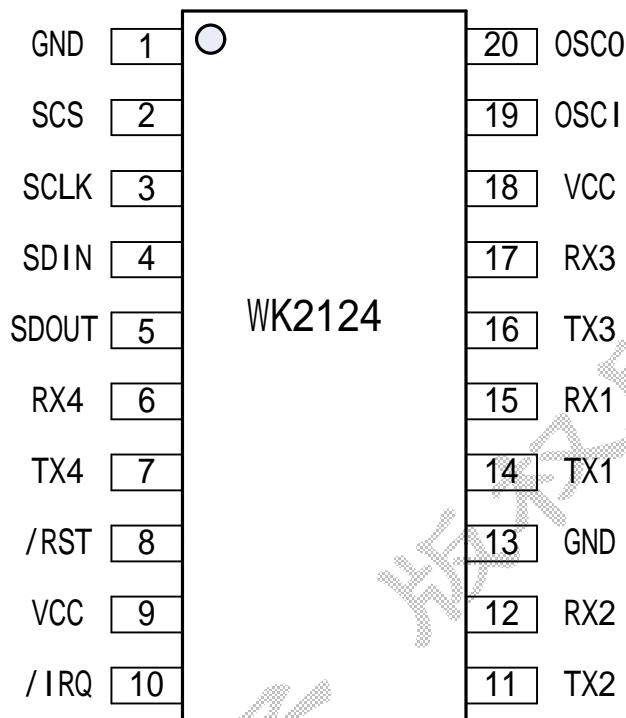
5. 原理框图

图5.1 WK2124 原理框图



6. 封装引脚

6.1 封装图



6.2 引脚描述

表6.2 WK2124 引脚描述

名称	管脚	类型	描述
GND	1, 13	-	地
SCS	2	I/O	当主接口为SPI时, 为SCS(SPI片选SSEL)功能引脚; 低电平有效
SCLK	3	I/O	当主接口为SPI时, 为SCLK(SPI 时钟输入)功能引脚;
SDIN	4	I/O	当主接口为SPI时, 为MOSI功能引脚;
SDOUT	5	I/O	当主接口为SPI时, 为MISO功能引脚;
RX1	15	I	子通道串口串行数据输入。 RX 将所连数据UART的串行数据输入WK2124的相应管脚。
RX2	12		
RX3	17		
RX4	6		
TX1	14	O	子通道串口串行数据输出。 TX 将串行数据输出到与其连接的器件引脚。
TX2	11		
TX3	16		
TX4	7		
RST	8	I	硬件复位引脚, 低电平复位有效
VCC	9, 18	-	电源 2.5V~5.0V工作电压
IRQ	10	O	中断输出信号, 低电平有效。建议外接上拉电阻, 典型取值5.1K
OSCI	19	I	晶振输入。注意: 需要和晶振并联1M的电阻。

OSCO	20	0	晶振输出
------	----	---	------

7. 寄存器描述

7.1 寄存器列表

WK2124的寄存器按地址编号为6位地址编号，地址000000~111111，分为全局寄存器和子串口寄存器。

全局寄存器4个，全局寄存器的地址具体排列见表7.1:

表7.1 全局寄存器列表

寄存器地址[5:0]	寄存器名称	类型	寄存器功能描述
000000	GENA	R/W	全局控制寄存器
000001	GRST	R/W	全局子串口复位寄存器
010000	GIER	R/W	全局中断寄存器
010001	GIFR	R	全局中断标志寄存器

子串口寄存器18个，其排列为C1C0 REG[3:0]，高两位为子串口通道号，低4位为寄存器地址，按低4位的寄存器地址具体排列见表7.2:

表7.2 子串口控制寄存器

寄存器地址[3:0]	寄存器名称	类型	寄存器功能描述	
(C1,C0) 0011	SPAGE	R/W	子串口页控制寄存器	
(C1,C0) 0100	SCR	R/W	子串口使能寄存器	SPAGE0
(C1,C0) 0101	LCR	R/W	子串口配置寄存器	SPAGE0
(C1,C0) 0110	FCR	R/W	子串口 FIFO 控制寄存器	SPAGE0
(C1,C0) 0111	SIER	R/W	子串口中断使能寄存器	SPAGE0
(C1,C0) 1000	SIFR	R/W	子串口中断标志寄存器	SPAGE0
(C1,C0) 1001	TFCNT	R	子串口发送 FIFO 计数寄存器	SPAGE0
(C1,C0) 1010	RFCNT	R	子串口接收 FIFO 计数寄存器	SPAGE0
(C1,C0) 1011	FSR	R	子串口 FIFO 状态寄存器	SPAGE0
(C1,C0) 1100	LSR	R	子串口接收状态寄存器	SPAGE0
(C1,C0) 1101	FDAT	R/W	子串口 FIFO 数据寄存器	SPAGE0
(C1,C0) 0100	BAUD1	R/W	子串口波特率配置寄存器高字节	SPAGE1
(C1,C0) 0101	BAUD0	R/W	子串口波特率配置寄存器低字节	SPAGE1
(C1,C0) 0110	PRES	R/W	子串口波特率配置寄存器小数部分	SPAGE1
(C1,C0) 0111	RFTL	R/W	子串口接收 FIFO 中断触发点配置寄存器	SPAGE1
(C1,C0) 1000	TFTL	R/W	子串口发送 FIFO 中断触发点配置寄存器	SPAGE1

C1,C0: 子通道号, 00~11 分别对应子串口 1 到子串口 4

7.2 寄存器描述

7.2.1 GENA 全局控制寄存器: (000000)

位	复位值	功能描述	类型
Bit7	0	RSV (保留位)	R

Bit6	0	RSV (保留位)	R
Bit5	1	RSV (保留位)	R
Bit4	1	RSV (保留位)	R
Bit3	0	UT4EN 子串口4时钟使能位(关断子串口时钟,可实现更低功耗) 0: 未使能 1: 使能	W/R
Bit2	0	UT3EN 子串口3时钟使能位(关断子串口时钟,可实现更低功耗) 0: 未使能 1: 使能	W/R
Bit1	0	UT2EN 子串口2时钟使能位(关断子串口时钟,可实现更低功耗) 0: 未使能 1: 使能	W/R
Bit0	0	UT1EN 子串口1时钟使能位(关断子串口时钟,可实现更低功耗) 0: 未使能 1: 使能	W/R

7.2.2 GRST 全局子串口复位寄存器: (000001)

位	复位值	功能描述	类型
Bit7	0	UT4SLEEP 子串口4休眠状态位(降低功耗,可自动唤醒) 0: 未休眠 1: 休眠	R
Bit6	0	UT3SLEEP 子串口3休眠状态位(降低功耗,可自动唤醒) 0: 未休眠 1: 休眠	R
Bit5	0	UT2SLEEP 子串口2休眠状态位(降低功耗,可自动唤醒) 0: 未休眠 1: 休眠	R
Bit4	0	UT1SLEEP 子串口1休眠状态位(降低功耗,可自动唤醒) 0: 未休眠 1: 休眠	R
Bit3	0	UT4RST 子串口4软复位控制位 0: 未复位子串口4 1: 复位子串口4	W1/R0
Bit2	0	UT3RST 子串口3软复位控制位 0: 未复位子串口3 1: 复位子串口3	W1/R0
Bit1	0	UT2RST 子串口2软复位控制位 0: 未复位子串口2 1: 复位子串口2	W1/R0
Bit0	0	UT1RST 子串口1软复位控制位 0: 未复位子串口1 1: 复位子串口1	W1/R0

7.2.3 GIER 全局中断寄存器: (010000)

位	复位值	功能描述	类型
Bit7 --- 5	000	RSV (保留位)	R
Bit4	0	RSV (保留位)	W/R
Bit3	0	UT4IE 子串口4 中断使能控制位 0: 未使能 1: 使能	W/R
Bit2	0	UT3IE 子串口3 中断使能控制位 0: 未使能 1: 使能	W/R
Bit1	0	UT2IE 子串口2 中断使能控制位 0: 未使能 1: 使能	W/R
Bit0	0	UT1IE 子串口1 中断使能控制位 0: 未使能 1: 使能	W/R

7.2.4 GIFR 全局中断标志寄存器: (010001)

位	复位值	功能描述	类型
Bit7 --- 4	000	RSV (保留位)	R
Bit3	0	UT4INT 子串口4 中断标志位 0: 无中断 1: 有中断	R
Bit2	0	UT3INT 子串口3 中断标志位 0: 无中断 1: 有中断	R
Bit1	0	UT2INT 子串口2 中断标志位 0: 无中断 1: 有中断	R
Bit0	0	UT1INT 子串口1 中断标志位 0: 无中断 1: 有中断	R

7.2.5 SPAGE 子串口页控制寄存器: (0011)

位	复位值	功能描述	类型
Bit7 --- 1	000000	RSV (保留位)	R
Bit0	0	PAGE 子串口页控制位 (子串口寄存器分布在 PAGE0 和 PAGE1 上, 不同页之间的切换, 通过该寄存器来控制) 0: PAGE0 1: PAGE1	W/R

7.2.6 SCR 子串口控制寄存器: (PAGE0:0100)

位	复位值	功能描述	类型
---	-----	------	----

Bit7 --- 3	000	RSV (保留位)	W/R
Bit2	0	SLEEPEN 子串口休眠使能位 0: 不使能 1: 使能	W/R
Bit1	0	TXEN 子串口发送使能位 0: 不使能 1: 使能	W/R
Bit0	0	RXEN 子串口接收使能位 0: 不使能 1: 使能	W/R

7.2.7 LCR 子串口配置寄存器: (PAGE0:0101)

位	复位值	功能描述	类型
Bit7 --- 6	00	RSV (保留位)	W/R
Bit5	0	BREAK 子串口 Line-Break 输出控制位 0: 正常输出 1: Line-Break 输出 (TX 强制输出 0)	W/R
Bit4	0	IREN 子串口红外使能位 0: 普通模式 1: 红外模式	W/R
Bit3	0	PAEN 子串口校验使能位 0: 无校验位 (8 位数据) 1: 有校验位 (9 位数据)	W/R
Bit2 --- 1	0	PAM1—0 子串口校验模式选择位 当 PAEN=1 子串口校验使能时: 00: 0 校验 ; 01: 奇校验 ; 10: 偶校验 ; 11: 1 校验	W/R
Bit0	0	STPL 子串口停止位长度控制位 0: 1bit 1: 2bits	W/R

7.2.8 FCR 子串口 FIFO 控制寄存器: (PAGE0:0110)

位	复位值	功能描述	类型
Bit7 --- 6	00	TFTRIG[1:0] 子串口发送 FIFO 触点设置位 当 TFTL[7:0]等于 0 时: 00: 8Byte 01:16 Byte 10: 24 Byte 11:30 Byte	W/R
Bit5 --- 4	00	RFTRIG[1:0] 子串口接收 FIFO 触点设置位 当 RFTL[7:0]等于 0 时: 00: 8Byte 01:16 Byte 10: 24 Byte 11:28 Byte	W/R
Bit3	0	TFEN 子串口发送 FIFO 使能位 0: 不使能	W/R

		1: 使能	
Bit2	0	RFEN 子串口接收 FIFO 使能位 0: 不使能 1: 使能	W/R
Bit1	0	TFRST 子串口发送 FIFO 复位位 (该位写 1 复位, 完成后自动置 0) 0 未使能复位 1: 复位 FIFO	W1/R0
Bit0	0	RFRST 子串口接收 FIFO 复位位 (该位写 1 复位, 完成后自动置 0) 0: 未使能复位 1: 复位 FIFO	W1/R0

7.2.9 SIER 子串口中断使能寄存器: (PAGE0:0111)

位	复位值	功能描述	类型
Bit7	0	FERR_IEN 接收 FIFO 数据错误中断使能位 0: 禁止接收 FIFO 数据错误中断 1: 使能接收 FIFO 数据错误中断	W/R
Bit6	0	RSV (保留位)	W/R
Bit5	0	RSV (保留位)	W/R
Bit4	0	RSV (保留位)	W/R
Bit3	0	TFEMPTY_IEN 发送 FIFO 空中断使能位 0: 禁止发送 FIFO 空中断 1: 使能发送 FIFO 空中断	W/R
Bit2	0	TFTRIG_IEN 发送 FIFO 触点中断使能位 0: 禁止发送 FIFO 触点中断 1: 使能发送 FIFO 触点中断	W/R
Bit1	0	RXOVT_IEN 接收 FIFO 超时中断使能位 0: 禁止接收 FIFO 超时中断 1: 使能接收 FIFO 超时中断	W/R
Bit0	0	RFTRIG_IEN 接收 FIFO 触点中断使能位 0: 禁止接收 FIFO 触点中断 1: 使能接收 FIFO 触点中断	W/R

7.2.10 SIFR 子串口中断标志寄存器: (PAGE0:1000)

位	复位值	功能描述	类型
Bit7	0	FERR_INT 接收 FIFO 数据错误中断标志位 0: 无接收 FIFO 数据错误中断 1: 有接收 FIFO 数据错误中断	W/R
Bit6	0	RSV (保留位)	W/R
Bit5	0	RSV (保留位)	W/R
Bit4	0	RSV (保留位)	W/R
Bit3	0	TFEMPTY_INT 发送 FIFO 空中断标志位	W/R

		0: 无发送 FIFO 空中断 1: 有发送 FIFO 空中断	
Bit2		TFTRIG_INT 发送 FIFO 触点中断标志位 0: 无发送 FIFO 触点中断 1: 有发送 FIFO 触点中断	W/R
Bit1		RXOVT_INT 接收 FIFO 超时中断标志位 0: 无接收 FIFO 超时中断 1: 有接收 FIFO 超时中断	W/R
Bit0	0	RFTRIG_INT 接收 FIFO 触点中断标志位 0: 无接收 FIFO 触点中断 1: 有接收 FIFO 触点中断	W/R

7.2.11 TFCNT 子串口发送 FIFO 计数寄存器: (PAGE0:1001)

位	复位值	功能描述	类型
Bit7 -- 0	00000000	子串口发送 FIFO 中的数据个数	R

7.2.12 RFCNT 子串口接收 FIFO 计数寄存器: (PAGE0:1010)

位	复位值	功能描述	类型
Bit7 -- 0	00000000	子串口接收 FIFO 中的数据个数	R

7.2.13 FSR 子串口 FIFO 状态寄存器: (PAGE0:1011)

位	复位值	功能描述	类型
Bit7	0	RFOE 子串口接收 FIFO 中数据溢出出错标志位 0: 无 OE 错误 1: 有 OE 错误	R
Bit6	0	RFBI 子串口接收 FIFO 中数据有 Line-Break 错误 0: 无 Line-Break 错误 1: 有 Line-Break 错误 (Rx 信号一直为 0 的状态, 包括校验位和停止位在内)	W/R
Bit5	0	RFFE 子串口接收 FIFO 中数据帧错误标志位 0: 无 FE 错误 1: 有 FE 错误	W/R
Bit4	0	RFPE 子串口接收 FIFO 中数据校验错误标志位 0: 无 PE 错误 1: 有 PE 错误	W/R
Bit3	0	RDAT 子串口接收 FIFO 空标志位 0: 子串口接收 FIFO 空 1: 子串口接收 FIFO 未空	W/R
Bit2	0	TDAT 子串口发送 FIFO 空标志位 0: 子串口发送 FIFO 空 1: 子串口发送 FIFO 未空	W/R
Bit1	0	TFULL 子串口发送 FIFO 满标志位 0: 子串口发送 FIFO 未满	W/R

		1: 子串口发送 FIFO 满	
Bit0	0	TBUSY 子串口发送 TX 忙标志位 0: 子串口发送 TX 空 1: 子串口发送 TX 忙	W/R

7.2.14 LSR 子串口接收状态寄存器: (PAGE0:1100)

位	复位值	功能描述	类型
Bit7 - 4	0	RSV (保留位)	
Bit3	0	OE 子串口接收 FIFO 当前被读取的字节溢出出错标志位 0: 无 OE 错误 1: 有 OE 错误	R
Bit2	0	BI 子串口接收 FIFO 中当前被读取字节 Line-Break 错误标志位 0: 无 Line-Break 错误 1: 有 Line-Break 错误 (Rx 信号一直为 0 的状态, 包括校验位和停止位在内)	R
Bit1	0	FE 子串口接收 FIFO 中当前被读取字节帧错误标志位 0: 无 FE 错误 1: 有 FE 错误	R
Bit0	0	PE 子串口接收 FIFO 中当前被读取字节校验错误标志位 0: 无 PE 错误 1: 有 PE 错误	R

7.2.15 FDAT 子串口 FIFO 数据寄存器: (PAGE0:1101)

位	复位值	功能描述	类型
Bit7 -- 0	00000000	写操作时: 写入子串口发送 FIFO 的数据 读操作时: 读出子串口接收 FIFO 的数据	W/R

7.2.16 BAUD1 子串口波特率配置寄存器高字节: (PAGE1:0100)

位	复位值	功能描述	类型
Bit7 --- 0	00000000	BAUD[15:8] 子串口波特率配置寄存器高字节	W/R

7.2.17 BAUD0 子串口波特率配置寄存器低字节: (PAGE1:0101)

位	复位值	功能描述	类型
Bit7 --- 0	00000000	BAUD[7:0] 子串口波特率配置寄存器低字节	W/R

7.2.18 PRES 子串口波特率配置寄存器小数部分: (PAGE1:0110)

位	复位值	功能描述	类型
Bit7 --- 4	0000	RSV (保留位)	R
Bit3 --- 0	0000	PRES[3:0]	W/R

7.2.19 RFTL 子串口接收 FIFO 触发中断寄存器: (PAGE1:0111)

位	复位值	功能描述	类型

Bit7 --- 0	00000000	接收 FIFO 触点控制	W/R
------------	----------	--------------	-----

7.2.20 TFTL 子串口发送 FIFO 触发中断寄存器：(PAGE1:1000)

位	复位值	功能描述	类型
Bit7 --- 0	00000000	发送 FIFO 触点控制	W/R

8. 全局功能描述

8.1 复位

WK2124为低电平复位。

各寄存器的复位值见7.2寄存器表中所列。

复位期间及复位后，各子串口处于禁止收发状态。当子串口处于联网模式下时，该特性使得该子串口所在的子节点在上电、复位期间不会对联网的其它节点产生干扰。

每个子串口可独立实现软件复位。

8.2 时钟选择

WK2124可以选择使用晶振时钟作为芯片的时钟源。注意：需要和晶振并联1M的电阻。见图8.2

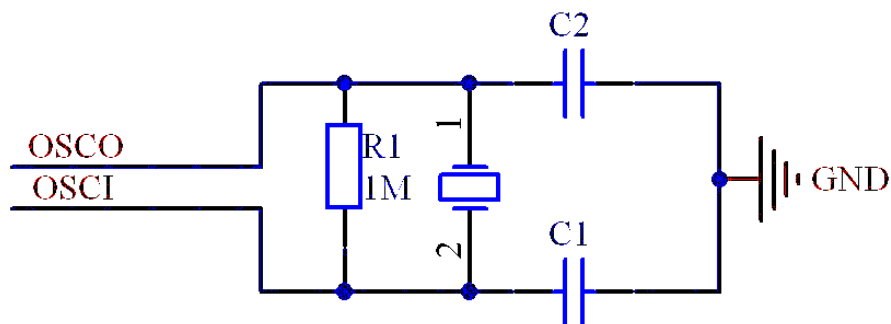


图 8.2 WK2124 时钟电路

8.3 中断控制

WK2124有两级中断：子串口中断及全局中断。当IRQ引脚指示有中断时，可以通过读取全局中断寄存器GIFR以判断当前中断的类型，然后去读取相应的中断状态寄存器，以确定当前的中断源。

WK2124的中断结构如下图所示：

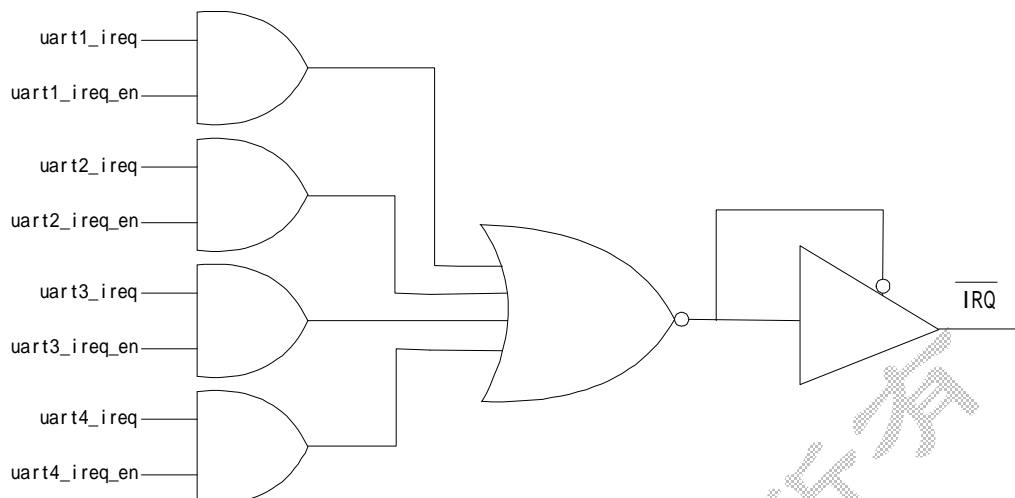


图 8.3 WK2124 中断结构图

WK2124的每个子串口都有独立的中断系统，包括：FIFO数据错误中断，发送FIFO空中断，发送FIFO触发点中断，接收FIFO超时中断，接收FIFO触发点中断。

当任意一个中断使能后，满足中断条件就会产生相应的中断。

8.3.1 FIFO数据错误中断

FIFO数据错误中断表明当前接收FIFO中有一个或以上的数据错误，产生错误的条件包括0（数据溢出错误），FE（数据帧错误），和PE（奇偶校验错），BE（Line-Break错误）。

一旦有接收FIFO中有出错数据，将产生该中断，当读取FSR寄存器后，清除相应的中断。

8.3.2 发送FIFO空中断

当发送FIFO中没有数据，产生该中断。当发送FIFO中的数据个数大于设定的发送FIFO触发点时，该中断被清除。

8.3.3 发送FIFO触发点中断

当发送FIFO中的数据个数小于设定的发送FIFO触发点时，产生该中断。当发送FIFO中的数据个数大于设定的发送FIFO触发点时，该中断被清除。

8.3.4 接收FIFO超时中断

当接收FIFO中数据个数小于设定的接收FIFO触发点并且RX引脚4个字节之内没有数据，产生该中断。当接收FIFO中的数据被读走或者RX继续接收数据时，该中断消失。

8.3.5 接收FIFO触发点中断

当接收FIFO中的数据个数大于设定的发送FIFO触发点时，产生该中断。当接收FIFO中的数据个数小于设定的发送FIFO触发点时，该中断被清除。

8.4 红外模式操作

WK2124的主串口和子串口都可以设置成为红外通信模式。当WK2124的UART设置为IrDA模式时，可以与符合SIR红外通信协议标准的设备通信，或者直接应用于光隔离通信中。

在IrDA模式下，一位数据的周期缩短到普通UART一位数据的3/16，小于1/16波特周期的脉冲将被作为干扰而忽略。

8.4.1 红外接收操作

在红外数据接收的时序和普通UART数据接收的对应图如图8.4.1所示：IRX为接收到的红外数据信号，RX为通过红外数据解码后的数据。解码后的数据与IRX上的数据有1个BIT（ $16 \times \text{CLOCK}$ ）的延迟。接收模式下，与普通UART不同的是，RX在脉冲的中间进行一次采样（区别与普通UART的3次采样），IrDA解码器将IRX上的3/16波特周期的脉冲解码为数据0，持续低电平解码为数据1。

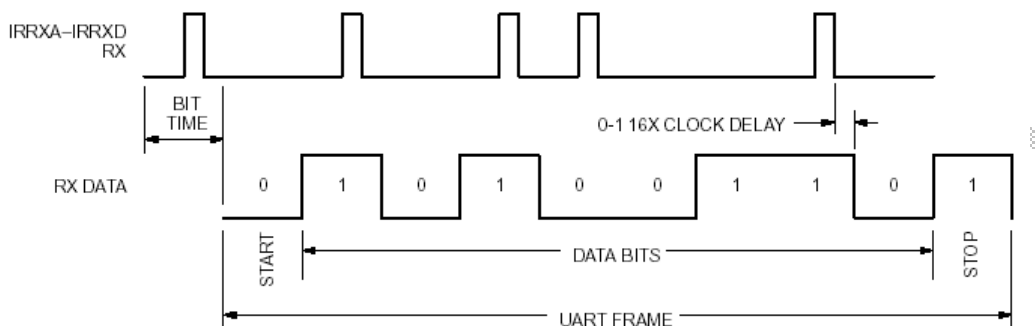


图 8.4.1 红外接收时序

8.4.2 红外发送操作

红外数据发送和普通UART数据发送的对应图如图8.4.2所示，TX为普通UART数据发送时序，IRTX为红外发送时序。当发送数据0时，红外编码器将产生一个3/16位宽的脉冲通过TX发送。当发送数据1时，保持低电平不变。

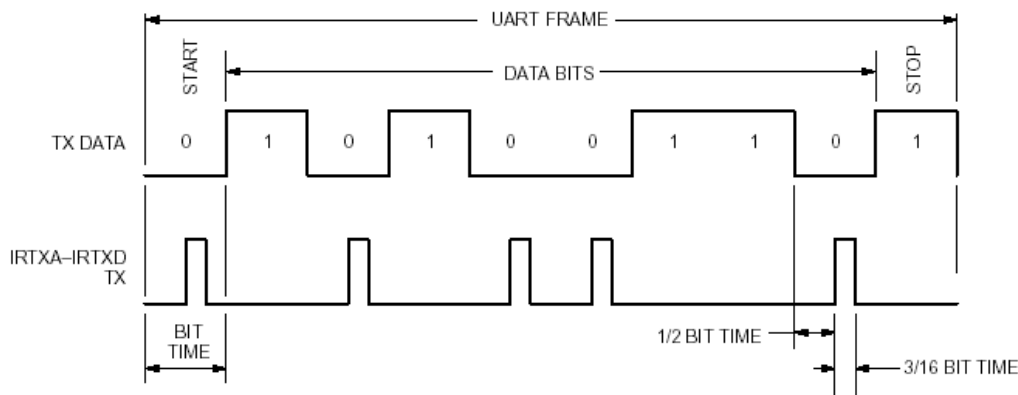


图 8.4.2 红外发送时序

8.5 可编程波特率发生器

WK2124的主串口和子串口采用相同的独立可编程波特率发生器。该波特率发生器产生 $16 \times$ 系统时钟的分频系数，分频率可以通过软件设置。

8.5.1 常见波特率和晶振对照表

下表给出了在不同系统时钟频率下的串口波特率设置表：

表 8.6.1

BAUD BAUD[15-0]	PRES	波特率 Fosc=	波特率 Fosc=	波特率 Fosc=	波特率 Fosc=	波特率 Fosc=

		1.8432MHz	3.6864MHz	7.3728MHz	11.0592MHz	14.7456MHz
0X0002	0X00	38400	76800	153600	230400	307200
0X0005	0X00	19200	38400	76800	115200	153600
0X000b	0X00	9600	19200	38400	57600	76800
0X0017	0X00	4800	9600	19200	28800	38400
0X002f	0X00	2400	4800	9600	14400	19200
0X005f	0X00	1200	2400	4800	7200	9600
0X00bf	0X00	600	1200	2400	3600	4800
0X017f	0X00	300	600	1200	1800	2400
0X0000	0X00	115200	230400	460800	691200	921600
0X0001	0X00	57600	115200	230400	345600	460800
0X0003	0X00	28800	57600	115200	172800	230400
0X0007	0X00	14400	28800	57600	86400	115200
0X000f	0X00	7200	14400	28800	43200	57600
0X001f	0X00	3600	7200	14400	21600	28800
0X003f	0X00	1800	3600	7200	10800	14400
0X007f	0X00	900	1800	3600	5400	7200

[注] 上表中蓝底部分的设置为 WK2124 复位后的初始值。

8.5.2 任意晶振下波特率计算

$$\text{计算公式: } f_s / (\text{baud} * 16) = \text{Reg}$$

注: f_s 为系统时钟, baud 为需要设置的波特率, Reg 为计算结果 (通常需要精确到小数点后两位)

Reg 整数部分减一并换算成16进制写入 {BAUD1,BAUA0}; 如果还有小数部分, 则取小数部分第一位写入 PRES。如果没有小数部分, 只需把整数部分写入 {BAUD1,BAUA0}, PRES 写入 0 即可。

例1: $f_s=11.0592\text{MHz}$, $\text{baud}=115200$. 根据公式可得 $\text{Reg}=6$ 。那么填入寄存器的数据为:

BAUD1=0X00;BAUD0=0X05;PRES=0X00.

例2: $f_s=12\text{MHz}$, $\text{baud}=115200$. 根据公式可得 $\text{Reg}=6.51$ (精确到小数点后两位)。那么填入寄存器的数据为 BAUD1=0X00;BAUD0=0X05; PRES=0X05.

例3: 高波特率计算

BAUD BAUD[15-0]	PRES	波特率 Fosc= 8MHz	波特率 Fosc= 16MHz	波特率 Fosc= 24MHz	波特率 Fosc= 32MHz
0X0000	0X00	500K	1M	1.5M	2M
0X0001	0X00	250K	500K	750K	1M
0X0003	0X00	125K	250K	375K	500K

8.6 数据格式设置

8.6.1 校验模式

WK2124的UART能提供强制校验，计算校验和无校验的数据格式，通过LCR（子串口配置寄存器）进行设置：

强制校验模式

WK2124支持强1校验，强0校验和用户指定校验模式。在这种模式下，校验设置仅影响数据发送，数据接收将忽略奇偶校验。

WK2124支持1校验、0校验，奇校验、偶校验模式。在该模式下，接收和发送的数据都进行奇偶校验计算。

8.6.2 数据长度

WK2124支持1或2位停止位模式。主串口通过GMUT.GSTPL设置，子串口通过LCR.STPL设置。

8.7 休眠和自动唤醒

WK2124支持休眠和自动唤醒模式，并且每个子串口可以单独进行休眠设置。

休眠条件：1、SCR.SLEEPEN=1 2、接收FIFO和发送FIFO要为空

3、RX上没有数据接收和TX上没有数据发送

4、子串口没有任何中断

当同时满足上述条件，并保持以上状态4个Bytes的时间后，子串口自动进入休眠状态，子串口的时钟自动关闭，以降低功耗。此时通过读取GRST来判断子串口是否进入sleep状态。

当子串口进入休眠状态后，满足下面条件之一，休眠中的子串口可自动唤醒，此时通过读取GRST来判断子串口是否被唤醒。

唤醒条件：1、RX上开始接收数据

2、向子串口发送FIFO写数据

3、CTS引脚电平变化

8.8 FIFO触点设置

WK2124支持每个子串口设置不同的触发点，接收FIFO和发送FIFO可以独立设置不同触发点。设置触点的方法有两种：1、配置固定触点：通过FCR寄存器中的TFTRIG[1:0]和RFTRIG[1:0]位固定编程来配置触发点位置。2、配置任意触点：通过设置TFTL和RFTL两个寄存器来设置任意触发点位置。具体配置见表8.8.1：

表 8.8.1

TFTL [7:0]	TFTRIG [1:0]		TX Trigger Level	RFTL [7:0]	RFTRIG [1:0]		RX Trigger Level
= =0	0	0	8	= =0	0	0	8
= =0	0	1	16	= =0	0	1	16
= =0	1	0	24	= =0	1	0	24
= =0	1	1	30	= =0	1	1	28
! =0	X	X	TFTL	! =0	X	X	RFTL

9. SPI接口模式操作

9.1 SPI 与主机的连接:

如图 9.1 所示 SPI 接口包括如下四个信号:

SDIN: SPI 从器件数据输入 (MOSI)。

SDOUT: SPI 从器件数据输出 (MISO)。

SCLK: SPI 串行时钟 (SCLK)。

SCS: SPI 片选 (从属选择信号 SSEL)。

WK2124 与主机的连接如图 9.1 所示。

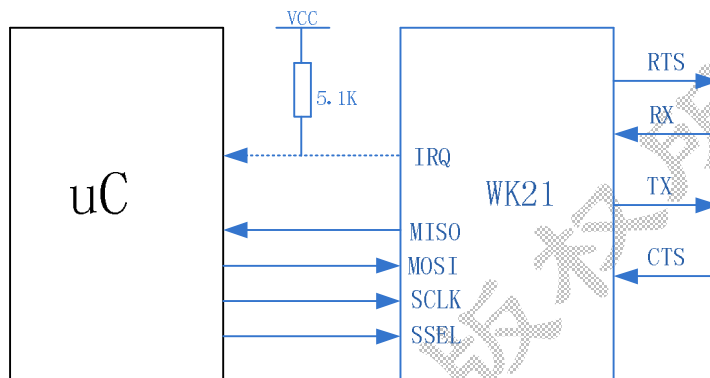


图 9.1 SPI 与主机连接图

9.2 SPI 接口的操作时序

WK2124 工作在 SPI 同步串行通信的从机模式下，支持 SPI 模式 0 标准。为实现主机和 WK2124 的通信，在主机端需要设置 CPOL=0(SPI 时钟极性选择位),CPHA=0(SPI 时钟相位选择位)。

WK2124 SPI 接口的操作时序如下所示:

写寄存器操作时序如图 9.2 所示: 先写入一个命令字节 (Command Byte)，随后写入相应的数据字节,数据字节的寄存器地址自动增加。

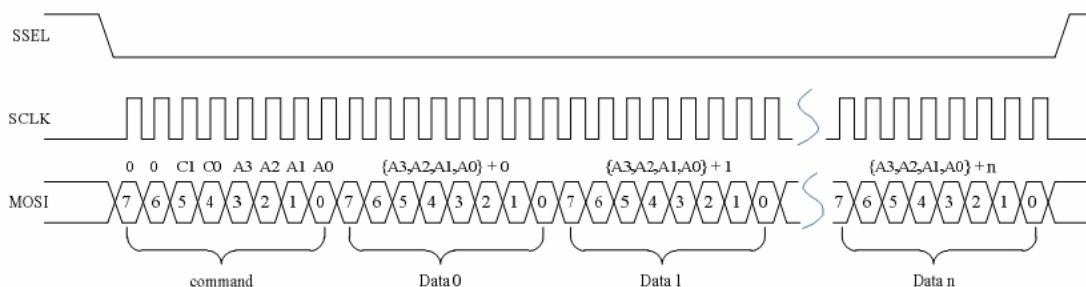


图 9.2 SPI 写寄存器时序图

读寄存器操作时序如图 9.3 所示: 先写入一个命令字节 (Command Byte)，随后芯片 MISO 线上会返回相应的数据字节。返回数据字节的寄存器地址自动增加。

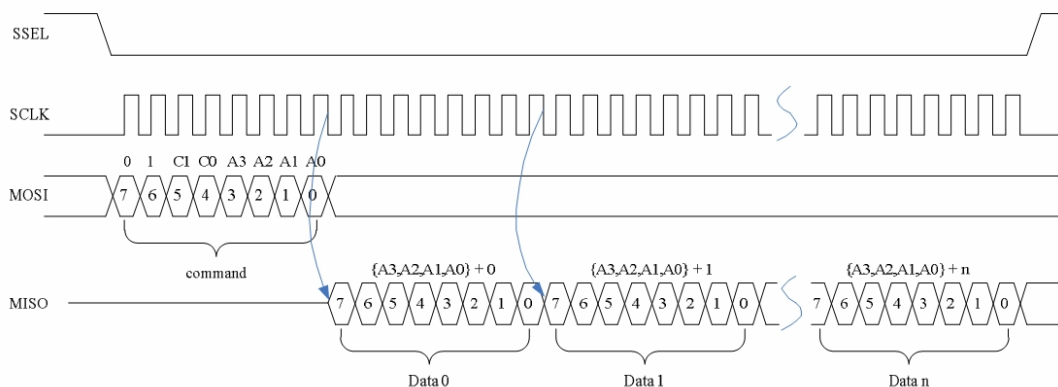


图 9.3 SPI 读寄存器时序图

写 FIFO 操作时序如图 9.4 所示：先写入一个命令字节（Command Byte），随后再写入相应的数据字节。FIFO 地址自动增加。

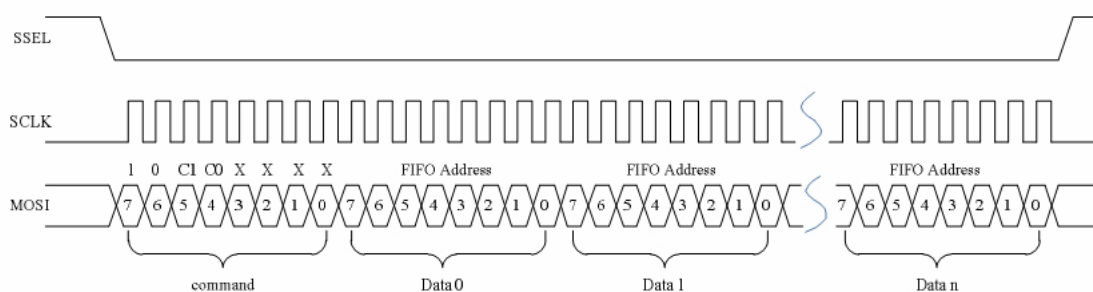


图 9.4 SPI 写 FIFO 时序图

读 FIFO 操作时序如图 9.5 所示：先写入一个命令字节（Command Byte），随后芯片 MISO 线上会返回相应的数据字节。FIFO 地址自动增加。

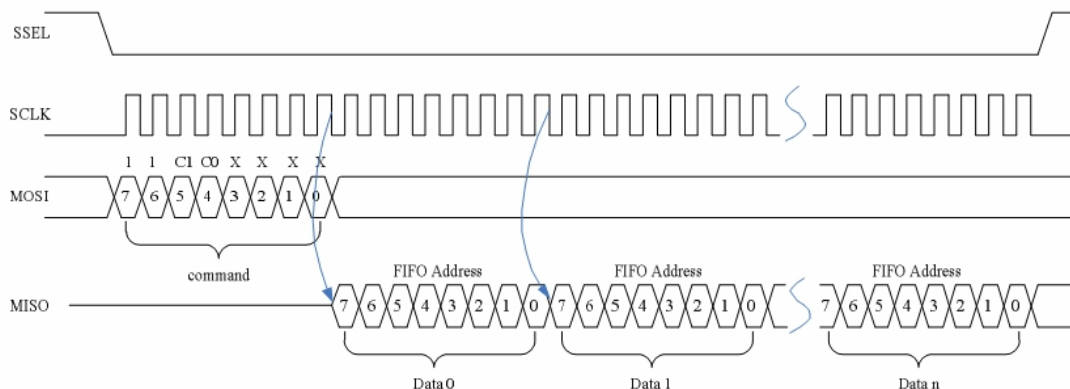


图 9.5 SPI 读 FIFO 时序图

9.3 SPI 总线通信协议描述：

9.3.1.SPI 写寄存器

SPI	控制字节 CMD								数据字节 DB(写 N 个数据字节, 寄存器地址自动增加)							
	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
BIT	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
MOSI	0	0	C1	C0	A3	A2	A1	A0	D7t	D6t	D5t	D4t	D3t	D2t	D1t	D0t
MISO	HZ	HZ	HZ	HZ	HZ	HZ	HZ	HZ	HZ	HZ	HZ	HZ	HZ	HZ	HZ	HZ

9.3.2.SPI 读寄存器

SPI	控制字节 CMD								数据字节 DB(读 N 个数据字节, 寄存器地址自动增加)							
	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
BIT	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
MOSI	0	1	C1	C0	A3	A2	A1	A0	X	X	X	X	X	X	X	X
MISO	HZ	HZ	HZ	HZ	HZ	HZ	HZ	HZ	D7t	D6t	D5t	D4t	D3t	D2t	D1t	D0t

9.3.3.SPI 写 FIFO

SPI	控制字节 CMD								数据字节 DB(写 N 个数据字节到 {C1C0} 的 FIFO, FIFO 地址自动增加)							
	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
BIT	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
MOSI	1	0	C1	C0	X	X	X	X	D7t	D6t	D5t	D4t	D3t	D2t	D1t	D0t
MISO	HZ	HZ	HZ	HZ	HZ	HZ	HZ	HZ	HZ	HZ	HZ	HZ	HZ	HZ	HZ	HZ

9.3.4.SPI 读 FIFO

SPI	控制字节 CMD								数据字节 DB(从 {C1C0} 的 FIFO 读 N 个数据字节, FIFO 地址自动增加)							
	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
BIT	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
MOSI	1	1	C1	C0	X	X	X	X	X	X	X	X	X	X	X	X
MISO	HZ	HZ	HZ	HZ	HZ	HZ	HZ	HZ	D7t	D6t	D5t	D4t	D3t	D2t	D1t	D0t

说明:

C1 C0: 子串口通道号 00~11 分别对应子串口 1 到子串口 4

A3-A0: 子串口寄存器地址

D7t...D0t: 8 位数据字节

10. 子串口操作描述

10.1 子串口使能/禁止

WK2124 允许独立使能或禁止每个子串口通道。

在使用中可以禁止不使用的子串口通道。

子串口通道只有在处在使能状态才能接收和发送数据。

10.2 收发 FIFO 控制

WK2124 提供了独立的 256 级 FIFO 接收和发送 FIFO。(子串口 FIFO 控制寄存器) 进行设置。

10.2.1 发送 FIFO 触发点操作

WK2124为每个通道提供独立的可编程发送FIFO触发点设置(见图13.2.1),以产生相应的发送FIFO触发点中断。

当发送FIFO触发点中断使能时,发送FIFO中的数据数目小于设定的触发点时产生相应中断。

10.2.2 接收FIFO触发点操作

WK2124为每个通道提供独立的可编程接收FIFO触发点设置,以产生相应的接收FIFO触发点中断。当接收FIFO触发点中断使能时,接收FIFO中的数据数目大于设定的触发点时产生相应中断。

10.2.3 发送FIFO的使能/禁止

复位后,发送FIFO处于禁止状态。如果希望将数据写入发送FIFO,需要首先使能发送FIFO。

发送FIFO中的数据是否发送,取决于相应的子通道UART是否使能。一旦相应子通道UART处于使能状态,则发送FIFO中的数据将会立即发送,否则,发送FIFO中的数据将不会被发送直到相应的子通道被使能。

10.2.4 接收FIFO的使能/禁止

复位后,接收FIFO处于禁止状态。如果希望接收子串口数据,需要首先使能相应的子串口通道及其接收FIFO。只有相应的UART和接收FIFO使能后,接收到的数据才能写入接收FIFO存储。

如果子串口通道使能而接收FIFO禁止,子串口能接收数据,但数据不会写入接收FIFO而被忽略。

10.2.5 发送FIFO 清空

当FCR中发送FIFO清空位(TFRST)被置1时,该子通道发送FIFO中的数据将被清空,发送FIFO计数器和指针都将清零。

TFRST位被置1后,将会在一个时钟后被硬件自动清0。

10.2.6 接收FIFO 清空

当FCR中接收FIFO清空位(RFRST)被置1时,该子通道接收FIFO中的数据将被清空,接收FIFO计数器和指针都将清零。

RFRST位被置1后,将会在一个时钟后被硬件自动清0。

10.2.7 发送FIFO 计数器

WK2124用一个8位寄存器来反应当前发送FIFO中的数据数目:当一个字节的数据写入发送FIFO后,发送FIFO计数器自动加1;当一个发送FIFO中的数据被发送后,发送FIFO计数器自动减1。

注意:当发送FIFO计数器为255(11111111)时,如果再写入一个数据则计数器变为0(00000000)。当发送FIFO计数器为1(00000001)时,发送一个数据之后则计数器也变为0(00000000)。因此,当发送FIFO计数器为0时,表明发送FIFO满或者空,在这种情况下,需要结合子串口状态寄存器(FSR)中的相关状态位进行判断。

10.2.8 接收FIFO计数器

WK2124用一个8位寄存器来反应当前接收FIFO中的数据数目:当一个字节的数据写入接收FIFO后,接收FIFO计数器自动加1;当一个接收FIFO中的数据被读取后,接收FIFO计数器自动减1。

注意:当接收FIFO计数器为255(11111111)时,如果再接收一个数据则计数器变为0(00000000)。当接收FIFO计数器为1(00000001)时,读取一个数据之后则计数器也变为0(00000000)。因此,当接收FIFO计数器为0时,表明接收FIFO满或者空,在这种情况下,需要结合子串口状态寄存器(FSR)中的相关状态位进行判断。

11. 参数指标

11.1 WK2124的静态参数

除非特别说明,满足: VCC=(2.5V±0.2V)或(3.3±0.3V)或(5V); -40℃到+85℃;

符号	说明	条件	VCC=2.5V		VCC=3.0V		VCC=5.0V		单位
			最小	最大	最小	最大	最小	最大	
电源									
VCC	电源电压		2.3	2.7	3.0	3.6	4.5	5.0	V
ICC	工作电流	3.6864MHz 晶振	0.8	2	1	2	2	3	mA
ICCSL	休眠电流	无负载	150	-	200	-	460	-	uA
输入逻辑信号									
V _{IH}	输入高电平		1.8	5.0	2.0	5.0	3.6	5.0	V
V _{IL}	输入低电平		-	0.6	-	0.9	-	1.1	V
I _{IL}	输入漏电流	V _I =5.0 or 0V	-	±10	-	±10	-	±10	uA
C _i	输入电容		-	5	-	5	-	5	pF
输出逻辑信号									
V _{OH}	输出高电平	I _{OH} =3mA	1.9	-	2.4	-	4.5	-	V
V _{OL}	输出低电平	I _{OL} =-3mA	-	0.4	-	0.4	0	0.4	V
I _{OL}	输出漏电流		-	±10	-	±10	-	±10	uA
C _o	输出电容		-	5	-	5	-	5	pF

11.2 WK2124的动态参数

符号	说明	条件	VCC=2.5V		VCC=3.0V		VCC=5.0V		单位
			最小	最大	最小	最大	最小	最大	
FOSI	晶振频率		-	16	-	24	-	32	MHz

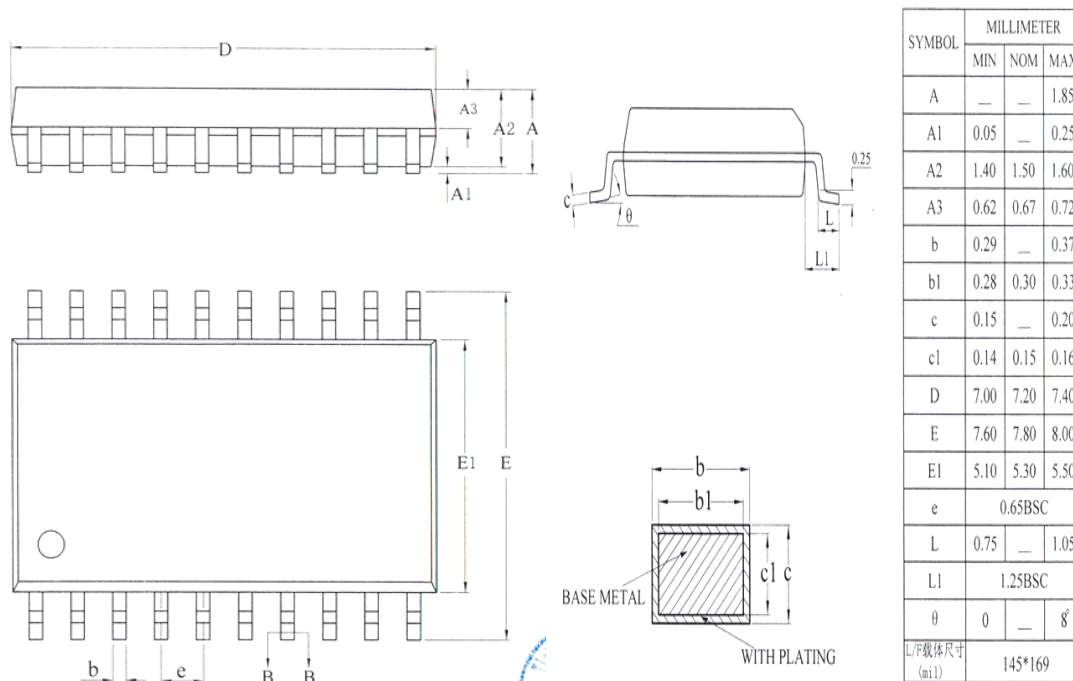
11.3 WK2124的极限参数

符号	说明	条件	最小	最大	单位
VCC	电源电压		-0.5	6	V
V _I	输入电压		-0.5	+5.5	V
V _O	输出电压		-0.5	+5.5	V
P _{TOL}	总功耗		-	300	mW
T _O	工作温度		-40	+85	°C
T _{STG}	存储温度		-65	+150	°C

12. 封装信息

WK2124采用SSOP20无铅绿色封装

图12.1 SSOP20封装信息



13. 焊接工艺

WK2124 采用使用绿色环保材料，引脚采用纯锡电镀。推荐使用峰值温度小于 260℃，符合无铅标准的回流焊工艺进行焊接。

所有 SMD 器件焊接工艺都对湿度敏感(湿度等级及条件见外包装盒)，建议在焊接前进行干燥处理。

采用手工焊接时，应首先焊接两个对角线的引脚进行固定后再焊接其它引脚。焊接温度为 300℃，烙铁与引脚的接触时间控制在 10 秒以内。

14. 特别申明

本产品并非为生命保障系统、航空航天系统设计，将本产品应用于该领域而引发的一切后果，为开微电子将不承担任何责任。为开微电子保留对产品进行性能、功能、参数修改的权利。对于正式量产的产品，为开微电子做出的修改将以公告方式通告用户。

15. 版本历史

V1.0 以前版本均为未正式公开的内部版本。

16. 联系信息

· 请访问为开微电子的网站获取我们的最新联系方式: www.wkmic.com