

1 产品预览

介 绍

三星的 S3C44B0X 16/32 位 RISC 处理器被设计来为手持设备等提供一个低成本高性能的方案。

S3C44B0X提供以下配置：2.5V ARM7TDMI 内核带有8Kcache ;可选的internal SRAM;LCD Controller(最大支持256色STN , 使用LCD专用DMA) ; 2-ch UART with handshake(IrDA1.0, 16-byte FIFO) / 1-ch SIO ; 2-ch general DMAs / 2-ch peripheral DMAs with external request pins ; External memory controller (chip select logic, FP/ EDO/SDRAM controller) ; 5-ch PWM timers & 1-ch internal timer ; Watch Dog Timer ; 71 general purpose I/O ports / 8-ch external interrupt source ; RTC with calendar function ; 8-ch 10-bit ADC ; 1-ch multi-master IIC-BUS controller ; 1-ch IIS-BUS controller ; Sync. SIO interface and On-chip clock generator with PLL。

S3C44B0X采用一种新的三星ARM CPU嵌入总线结构-SAMBA2 , 最大达66MHZ。

电源管理支持 : Normal, Slow, Idle, and Stop mode。

系统管理功能 :

- 1 Little/Big endian support.
- 2 Address space: 32Mbytes per each bank. (Total 256Mbyte)
- 3 Supports programmable 8/16/32-bit data bus width for each bank.
- 4 Fixed bank start address and programmable bank size for 7 banks.
- 5 . 8 memory banks.
 - 6 memory banks for ROM, SRAM etc.
 - 2 memory banks for ROM/SRAM/DRAM(Fast Page, EDO, and Synchronous DRAM)
6. Fully Programmable access cycles for all memory banks.
- 7 Supports external wait signal to expend the bus cycle.
8. Supports self-refresh mode in DRAM/SDRAM for power-down.
9. Supports asymmetric/symmetric address of DRAM.

Cache 和内部存储器功能:

- 4-way set associative ID(Unified)-cache with 8Kbyte.
- The 0/4/8 Kbytes internal SRAM using unused cache memory.
- Pseudo LRU(Least Recently Used) Replace Algorithm.
- Write through policy to maintain the coherence between main memory and cache content.
- Write buffer with four depth.
- Request data first fill technique when cache miss occurs.

时钟和电源管理

- Low power
- The on-chip PLL makes the clock for operating MCU at maximum 66MHz.
- Clock can be fed selectively to each function block by software.
- Power mode: Normal, Slow, Idle and Stop mode.

Normal mode: Normal operating mode.

Slow mode: Low frequency clock without PLL

Idle mode: Stop the clock for only CPU

Stop mode: All clocks are stopped

- Wake up by EINT[7:0] or RTC alarm interrupt from idle mode.

中断控制器

- 30 Interrupt sources(Watch-dog timer, 6 Timer, 6 UART, 8 External interrupts, 4 DMA , 2 RTC, 1 ADC, 1 IIC, 1 SIO)
- Vectored IRQ interrupt mode to reduce interrupt latency.
- Level/edge mode on the external interrupt sources
- Programmable polarity of edge and level
- Supports FIQ (Fast Interrupt request) for very urgent interrupt request

定时器功能

- 5-ch 16-bit Timer with PWM / 1-ch 16-bit internal timer with DMA-based or interrupt-based operation
- Programmable duty cycle, frequency, and polarity
- Dead-zone generation.
- Supports external clock source.

RTC 功能:

- Full clock feature: msec, sec, min, hour, day, week, month, year.
- 32.768 KHz operation.
- Alarm interrupt for CPU wake-up.
- Time tick interrupt

通用输入输出功能:

- 8 external interrupt ports
- 71 multiplexed input/output ports

UART 功能:

- 2-channel UART with DMA-based or interrupt-based operation
- Supports 5-bit, 6-bit, 7-bit, or 8-bit serial data transmit/receive
- Supports H/W handshaking during transmit/receive
- Programmable baud rate
- Supports IrDA 1.0 (115.2kbps)
- Loop back mode for testing
- Each channel have two internal 32-byte FIFO for Rx and Tx.

DMA控制器功能:

- 2 channel general purpose Direct Memory Access controller without CPU intervention.
- 2 channel Bridge DMA (peripheral DMA) controller.
- Support IO to memory, memory to IO, IO to IO with the Bridge DMA which has 6 type's DMA requestor: Software, 4 internal function blocks (UART, SIO, Timer, IIS), and External pins.
- Programmable priority order between DMAs (fixed or round-robin mode)
- Burst transfer mode to enhance the transfer rate on the FPD RAM, EDODRAM and SDRAM.
- Supports fly-by mode on the memory to external device and external device to memory transfer mode

A/D 转换器:

- 8-ch multiplexed ADC.

- Max. 100KSPS/10-bit.

LCD控制器:

- Supports color/monochrome/gray LCD panel
- Supports single scan and dual scan displays
- Supports virtual screen function
- System memory is used as display memory
- Dedicated DMA for fetching image data from system memory
- Programmable screen size
- Gray level: 16 gray levels
- 256 Color levels

看门狗定时器:

- 16-bit Watchdog Timer
- Interrupt request or system reset at time-out

IIC-BUS 接口

- 1-ch Multi-Master IIC-Bus with interrupt-based operation.
- Serial, 8-bit oriented, bi-directional data transfers can be made at up to 100 Kbit/s in the standard mode or up to 400 Kbit/s in the fast mode.

IIS-BUS接口

- 1-ch IIS-bus for audio interface with DMA-based operation.
- Serial, 8/16bit per channel data transfers
- Supports MSB-justified data format

SIO (同步串口):

- 1-ch SIO with DMA-based or interrupt –based operation.
- Programmable baud rates.
- Supports serial data transmit/receive operations 8-bit in SIO.

操作电压范围:

- 核电压 : 2.5V I/O电压 : 3.0 V to 3.6 V

工作频率:

- Up to 66 MHz

封装:

- 160 LQFP / 160 FBGA

2 管脚描述

om[1:0]: 输入 om[1:0]设置S3C44B0X在测试模式和确定nGCS0的总线宽度,逻辑电平在复位期间由这些管脚的上拉下拉电阻确定.

00:8-bit 01:16-bit 10:32-bit 11:Test mode

ADDR[24:0] 输出: 地址总线 输出相应段的存储器地址.

DATA[31:0] 输入输出:数据总线,总线宽度可编程为8/16/32 位

nGCS[7:0] 输出:芯片选择,当存储器地址在相应段的地址区域时被激活.存取周期和段尺寸可编程.

nWE 输出 :写允许信号,指示当前的总线周期为写周期.

nWBE[3:0] 输出: 写字节允许信号

nBE[3:0] 输出:在使用SRAM情况下字节允许信号.

nOE输出 :读允许信号,指示当前的总线周期为读周期.

nXBREQ 输入: nXBREQ 总线控制请求信号, 允许另一个总线控制器请求控制本地总线, nXBACK信号激活指示已经得到总线控制权。

nXBACK 输出: 总线应答信号。

nWAIT 输入: nWAIT请求延长当前的总线周期, 只要nWAIT为低, 当前的总线周期不能完成。

ENDIAN 输入: 它确定数据类型是little endian还是big endian, 逻辑电平在复位期间由该管脚的上拉下拉电阻确定。

0:little endian 1:big endian

nRAS[1:0] 输出: 行地址选通信号。

nCAS[3:0] 输出: 列地址选通信号。

nSRAS输出: SDRAM行地址选通信号。

nSCAS输出: SDRAM列地址选通信号。

nSCS[1:0] 输出: SDRAM芯片选择信号。

DQM[3:0] 输出: SDRAM数据屏蔽信号。

SCLK输出: SDRAM时钟信号。

SCKE输出: SDRAM时钟允许信号。

VD[7:0]输出: LCD数据线, 在驱动4位双扫描的LCD时, VD[3:0]为上部显示区数据, VD[7:4]为下部显示区数据。

VFRAME输出: LCD场信号, 指示一帧的开始, 在开始的第一行有效。

VM输出: VM极性变换信号, 变化LCD行场扫描电压的极性, 可以每帧或可编程多少个VLINE信号打开。

VLINE输出: LCD行信号, 在一行数据左移进LCD驱动器后有效。

VCLK输出: LCD点时钟信号, 数据在VCLK的上升沿发送, 在下降沿被LCD驱动器采样。

TOUT[4:0]输出: 定时器输出信号。

TCLK输入: 外部时钟信号输入。

EINT[7:0]输入: 外部中断请求信号。

nXDREQ[1:0]输入: 外部DMA请求信号。

nXDACK[1:0]输出: 外部DMA应答信号。

RxD[1:0]输入: UART接收数据输入线。

TxD[1:0]输出: UART发送数据线。

nCTS[1:0]输入: UART清除发送输入信号。

nRTS[1:0]输出: UART请求发送输出信号。

IICSDA输入输出: IIC总线数据线。

IICSCL输入输出: IIC总线时钟线。

IISLRCK输入输出: IIS总线通道时钟选择信号线。

IISDO输出: IIS总线串行数据输出信号。

IISDI输入: IIS总线串行数据输入信号。

IISCLK输入输出: IIS总线串行时钟。

CODECLK输出: CODEC系统时钟。

SIORXD输入: SIO接收数据输入线。

SIOTXD输出: SIO发送数据线。

SIOCK输入输出: SIO时钟信号。

SIORDY输入输出: 当SIO的DMA完成SIO操作时的握手信号。

AIN[7:0] : ADC模拟信号输入

AREFT输入: ADC顶参考电压输入。

AREFB输入：ADC底参考电压输入。

AVCOM输入：ADC公共参考电压输入。

P[70:0]输入输出：通用I/O口（一些口只有输出模式）。

nRESET：复位信号，nRESET挂起程序，放S3C44B0X进复位状态。在电源打开已经稳定时，nRESET必须保持低电平至少4个MCLK周期。

OM[3:2]输入：OM[3:2]确定时钟模式。
 00 = Crystal(XTAL0,EXTAL0), PLL on 01 = EXTCLK, PLL on
 10, 11 = Chip test mode.

EXTCLK输入：当OM[3:2]选择外部时钟时的外部时钟输入信号线，不用时必须接高（3.3V）。

XTAL0模拟输入：系统时钟内部振荡线路的晶体输入脚。不用时必须接高（3.3V）。

EXTAL0模拟输出：系统时钟内部振荡线路的晶体输出脚，它是XTAL0的反转输出信号。不用时必须悬空。

PLLCAP模拟输入：接系统时钟的环路滤波电容（700PF）。

XTAL1模拟输入：RTC时钟的晶体输入脚。

EXTAL1模拟输出：RTC时钟的晶体输出脚。它是XTAL1的反转输出信号。

CLKout输出：时钟输出信号

nTRST输入：TAP控制器复位信号，nTRST在TAP启动时复位TAP控制器。若使用debugger，必须连接一个10K上拉电阻，否则nTRST必须为低电平。

TMS输入：TAP控制器模式选择信号，控制TAP控制器的状态次序，必须连接一个10K上拉电阻。

TCK输入：TAP控制器时钟信号，提供JTAG逻辑的时钟信号源，必须连接一个10K上拉电阻。

TDI输入：TAP控制器数据输入信号，是测试指令和数据的串行输入脚，必须连接一个10K上拉电阻。

TDO输出：TAP控制器数据输出信号，是测试指令和数据的串行输出脚。

VDD：S3C44B0X内核逻辑电压（2.5V）

VSS: S3C44B0X内核逻辑地。

VDDIO: S3C44B0X I/O口电源(3.3V)。

VSSIO: S3C44B0X I/O地。

RTCVDD:RTC电压(2.5V或3V,不支持3.3V)。

VDDADC:ADC电压(2.5V)。

VSSADC:ADC地。

3 指令集

4 存储管理

1 BWSCON 0x01C80000 R/W 总线宽度与等待状态控制寄存器 初始值为0

位名称	BIT	功能
ST7	31	该位确定BANK7上的SRAM 是否使用UB/LB 0- 不使用（PIN[14:11] 作为 nWBE[3:0]） 1- 使用（PIN[14:11] 作为 nBE[3:0]）
WS7	30	该位确定BANK7上的SRAM存储器的等待状态 0- WAIT disable 1 = WAIT enable
DW7	[29:28]	该两位确定BANK7的数据总线宽度 00 = 8-bit 01 = 16-bit, 10 = 32-bit
ST6	27	该位确定BANK6上的SRAM 是否使用UB/LB

		2- 不使用 (PIN[14:11] 作为 nWBE[3:0])
		3- 使用 (PIN[14:11] 作为 nBE[3:0])
WS6	26	该位确定BANK6上的SRAM存储器的等待状态 0- WAIT disable 1 = WAIT enable
DW6	[25:24]	该两位确定BANK6的数据总线宽度 00 = 8-bit 01 = 16-bit, 10 = 32-bit
ST5	23	该位确定BANK5上的SRAM 是否使用UB/LB 4- 不使用 (PIN[14:11] 作为 nWBE[3:0]) 5- 使用 (PIN[14:11] 作为 nBE[3:0])
WS5	22	该位确定BANK5上的存储器的等待状态 0- WAIT disable 1 = WAIT enable
DW5	[21:20]	该两位确定BANK5的数据总线宽度 00 = 8-bit 01 = 16-bit, 10 = 32-bit
ST4	19	该位确定BANK4上的SRAM 是否使用UB/LB 6- 不使用 (PIN[14:11] 作为 nWBE[3:0]) 7- 使用 (PIN[14:11] 作为 nBE[3:0])
WS4	18	该位确定BANK4上的存储器的等待状态 0- WAIT disable 1 = WAIT enable
DW4	[17:16]	该两位确定BANK4的数据总线宽度 00 = 8-bit 01 = 16-bit, 10 = 32-bit
ST3	15	该位确定BANK3上的SRAM 是否使用UB/LB 8- 不使用 (PIN[14:11] 作为 nWBE[3:0]) 9- 使用 (PIN[14:11] 作为 nBE[3:0])
WS3	14	该位确定BANK3上的存储器的等待状态 0- WAIT disable 1 = WAIT enable
DW3	[13:12]	该两位确定BANK3的数据总线宽度 00 = 8-bit 01 = 16-bit, 10 = 32-bit
ST2	11	该位确定BANK2上的SRAM 是否使用UB/LB 10- 不使用 (PIN[14:11] 作为 nWBE[3:0]) 11- 使用 (PIN[14:11] 作为 nBE[3:0])
WS2	10	该位确定BANK2上的存储器的等待状态 0- WAIT disable 1 = WAIT enable
DW2	[9:8]	该两位确定BANK2的数据总线宽度 00 = 8-bit 01 = 16-bit, 10 = 32-bit
ST1	7	该位确定BANK1上的SRAM 是否使用UB/LB 12- 不使用 (PIN[14:11] 作为 nWBE[3:0]) 13- 使用 (PIN[14:11] 作为 nBE[3:0])
WS1	6	该位确定BANK1上的存储器的等待状态 0- WAIT disable 1 = WAIT enable
DW1	[5:4]	该两位确定BANK1的数据总线宽度 00 = 8-bit 01 = 16-bit, 10 = 32-bit
DW0	[2:1]	该两位指示BANK0的数据总线宽度(ONLY READ,由OM[1:0] 脚确定) 00 = 8-bit 01 = 16-bit, 10 = 32-bit
ENDIAN	0	该位指示endian mode (read only 状态由ENDIAN pins确定)

0 = Little endian 1 = Big endian

2

BANKCON0	0x01C80004	R/W	Bank 0 control register	0x0700
BANKCON1	0x01C80008	R/W	Bank 1 control register	0x0700
BANKCON2	0x01C8000C	R/W	Bank 2 control register	0x0700
BANKCON3	0x01C80010	R/W	Bank 3 control register	0x0700
BANKCON4	0x01C80014	R/W	Bank 4 control register	0x0700
BANKCON5	0x01C80018	R/W	Bank 5 control register	0x0700

位名称	BIT	功能
Tacs	[14:13]	在nGCSn 有效之前地址建立时间
00 = 0 clock	01 = 1 clock	
10 = 2 clocks	11 = 4 clocks	
Tcos	[12:11]	在nOE上芯片选择建立时间
00 = 0 clock	01 = 1 clock	
10 = 2 clocks	11 = 4 clocks	
Tacc	[10:8]	存取周期
000 = 1 clock	001 = 2 clocks	
010 = 3 clocks	011 = 4 clocks	
100 = 6 clocks	101 = 8 clocks	
110 = 10 clocks	111 = 14 clocks	
Toch	[7:6]	在nOE上芯片选择保持时间
00 = 0 clock	01 = 1 clock	
10 = 2 clocks	11 = 4 clocks	
Tcah	[5:4]	在nGCSn有效地址保持时间
00 = 0 clock	01 = 1 clock	
10 = 2 clocks	11 = 4 clocks	
Tpac	[3:2]	页模式存取周期
00 = 2 clocks	01 = 3 clocks	
10 = 4 clocks	11 = 6 clocks	
PMC	[1:0]	页模式配置
00 = normal (1 data)	01 = 4 data	
10 = 8 data	11 = 16 data	

寄存器名称	地址	描述	初始值
BANKCON6	0x01C8001C	R/W Bank 6 control register	0x18008
BANKCON7	0x01C80020	R/W Bank 7 control register	0x18008

位名称	BIT	功能
MT	[16:15]	这两位确定bank6 和 bank7存储器类型
00 = ROM or SRAM	01 = FP DRAM	
10 = EDO DRAM	11 = Sync. DRAM	

对于ROM和SRAM类型

Tacs	[14:13]	在nGCSn 有效之前地址建立时间
00 = 0 clock	01 = 1 clock	
10 = 2 clocks	11 = 4 clocks	
Tcos	[12:11]	在nOE上芯片选择建立时间

00 = 0 clock 01 = 1 clock
 10 = 2 clocks 11 = 4 clocks
 Tacc [10:8] 存取周期
 000 = 1 clock 001 = 2 clocks
 010 = 3 clocks 011 = 4 clocks
 100 = 6 clocks 101 = 8 clocks
 110 = 10 clocks 111 = 14 clocks
 Toch [7:6] 在nOE上芯片选择保持时间
 00 = 0 clock 01 = 1 clock
 10 = 2 clocks 11 = 4 clocks
 Tcah [5:4] 在nGCSn有效时地址保持时间
 00 = 0 clock 01 = 1 clock
 10 = 2 clocks 11 = 4 clocks
 Tpac [3:2] 页模式存取周期
 00 = 2 clocks 01 = 3 clocks
 10 = 4 clocks 11 = 6 clocks
 PMC [1:0] 页模式配置
 00 = normal (1 data) 01 = 4 consecutive accesses
 10 = 8 consecutive accesses 11 = 16 consecutive accesses

对于**FP DRAM**和**EDO DRAM**类型

Trcd [5:4] RAS 到 CAS 延时
 00 = 1 clock 01 = 2 clocks
 10 = 3 clocks 11 = 4 clocks
 Tcas [3] CAS 脉冲宽度
 0 = 1 clock 1 = 2 clocks
 Tcp [2] CAS 预充电周期
 0 = 1 clock 1 = 2 clocks
 CAN [1:0] 列地址数目
 00 = 8-bit 01 = 9-bit
 10 = 10-bit 11 = 11-bit

对于**SDRAM**类型

Trcd [3:2] RAS 到 CAS 延时
 00 = 2 clocks 01 = 3 clocks 10 = 4 clocks
 SCAN [1:0] 列地址数目
 00 = 8-bit 01 = 9-bit 10 = 10-bit

BANK7 BANK6 支持的存储器类型组合

SROM DRAM
 SDRAM SROM
 SROM SDRAM
 DRAM SROM
 不支持的组合
 SDRAM DRAM
 DRAM SDRAM

REFRESH 0x01C80024 R/W DRAM/SDRAM刷新控制寄存器 初始值0xac0000

位名称	BIT	功能
-----	-----	----

REFEN	[23]	DRAM/SDRAM刷新允许
-------	------	----------------

0 = Disable 1 = Enable (self or CBR/auto refresh)

TREFMD	[22]	DRAM/SDRAM刷新模式
--------	------	----------------

0 = CBR/Auto Refresh 1 = Self Refresh

在self-refresh 时, DRAM/SDRAM 控制信号被适当电平驱动

Trp	[21:20]	DRAM/SDRAM RAS 预充电时间
-----	---------	----------------------

DRAM :

00 = 1.5 clocks 01 = 2.5 clocks 10 = 3.5 clocks 11 = 4.5 clocks

SDRAM :

00 = 2 clocks 01 = 3 clocks 10 = 4 clocks 11 = Not support

Trc	[19:18]	SDRAM RAS 和CAS 最小时间
-----	---------	---------------------

00 = 4 clocks 01 = 5 clocks 10 = 6 clocks 11 = 7 clocks

Tchr	[17:16]	DRAM的CAS保持时间
------	---------	--------------

00 = 1 clock 01 = 2 clocks 10 = 3 clocks 11 = 4 clocks

Reserved	[15:11]	Not use
----------	---------	---------

Refresh Counter	[10:0]	DRAM/SDRAM刷新计数值
-----------------	--------	-----------------

刷新周期计算公式:

Refresh period = $(2^{11} - \text{refresh_count} + 1) / \text{MCLK}$

如果刷新周期是15.6 us和 MCLK 是 60 MHz,

refresh count如下计算

refresh count = $2^{11} + 1 - 60 \times 15.6 = 1113$

4

BANKSIZE 0x01C80028 R/W 段尺寸寄存器 初始值为 0x0

位名称	BIT	功能
-----	-----	----

SCLKEN	[4]	设置为1, 则SCLK仅在SDRAM存取周期产生, 这个特征将使功耗减少, 推荐设置为 1。
--------	-----	--

0 = normal SCLK =1

Reserved	[3]	保留为0
----------	-----	------

BK76MAP	[2:0]	BANK6/7存储器映射
---------	-------	--------------

000 = 32M/32M 100 = 2M/2M 101 = 4M/4M

110 = 8M/8M 111 = 16M/16M

4 SDRAM模式设置寄存器

MRSRB6 0x01C8002C R/W bank6模式设置寄存器 初始值 xxx

MRSRB7 0x01C80030 R/W bank7模式设置寄存器 初始值 xxx

位名称	BIT	功能
-----	-----	----

WBL	[9]	写突发脉冲长度
-----	-----	---------

0是推荐值

TM	[8:7]	测试模式
----	-------	------

00: 测试模式

01, 10, 11: 保留

CL	[6:4]	CAS 突发响应时间
----	-------	------------

000 = 1 clock, 010 = 2 clocks, 011=3 clocks

其它 = 保留

BT [3] 突发类型

0: 连续 (推荐)

1: N/A

BL [2:0] 突发长度

000: 1

其它: N/A

注：1 当程序在SDRAM运行时该寄存器不必重新配置。

2 所有的存储控制寄存器必须使用STMIA指令设置

3 在停止和SL_IDLE DRAM/SDRAM必须进入自刷新模式

5 时钟和电源管理模式

S3C44B0X的电源管理有5种模式：

1 Normal mode,

在正常模式CPU和所有的外设都正常工作，这时功耗最大，但用户能使用S/W指令停止每个外设的时钟供应。

2 Slow mode

Slow mode为非PLL模式，PLL不工作，使用外部时钟作为主时钟。在该模式，电源消耗依赖于外部时钟的频率。

3 Idle mode

Idle mode停止CPU CORE的时钟供应，仅对所有外设提供时钟，因此可以减少电源消耗。一个对CPU的中断请求能使S3C44B0X从该模式唤醒。

4 Stop mode

Stop mode 冻结所有的时钟供应，PLL也停止。这时的电源消耗最少，电流消耗仅是S3C44B0X的漏电流，少于10UA。外部中断能使CPU从该模式唤醒。

5 SL Idle mode

SL Idle mode 除了LCD控制器冻结所有的时钟。

S3C44B0X的时钟源可以用外部晶体来产生，也可以直接输入外部时钟，这有OM[3:2]的状态决定。M[3:2]的状态在nRESET的上升沿由OM3 和 OM2脚的电平决定。

M[3:2]=00 Crystal clock M[3:2]=01 Ext. Clock 其它 测试模式

注:在复位后PLL启动,但在用S/W指令设置PLLCON为有效的值之前,PLL OUTPUT (FOUT)不能使用,这时FOUT直接输出Crystal clock或外部时钟。

如果S3C44B0X的PLL的时钟源使用晶体,这时EXTCLK能作为Timer 5的时钟源TCLK。

1 PLL控制寄存器

PLLCON 0x01D80000 R/W PLL控制寄存器 复位值 0x38080

该寄存器设置PLL参数。

PLL输出频率计算公式如下:

$$F_{p1lo} = (m * Fin) / (p * 2^s)$$

$$m = (MDIV + 8), p = (PDIV + 2), s = SDIV$$

Fp1lo必须大于20MHZ 和少于66MHZ.

Fp1lo * 2^s 必须少于170MHZ

Fin / pT推荐为1MHZ 或大于 但小于2MHZ.

位名称	BIT	描述	默认值
-----	-----	----	-----

MDIV	[19:12]	<i>MDIV值</i>	0x38
PDIV	[9:4]	<i>PDIV值</i>	0x08
SDIV	[1:0]	<i>SDIV值</i>	0x0

2 时钟控制寄存器

CLKCON 0x01D80004 R/W 时钟控制寄存器 初始值 0x7ff8

位名称 BIT 描述

IIS [14] 控制 IIS block的钟控

0 = Disable, 1 = Enable

IIC [13] 控制 IIC block的钟控

0 = Disable, 1 = Enable

ADC [12] 控制 ADC block的钟控

0 = Disable, 1 = Enable

RTC [11] 控制 RTC block的钟控,即使该位为0,. RTC定时器仍工作

0 = Disable, 1 = Enable

GPIO [10] 控制 GPIO block的钟控,设置为1,允许使用EINT[4:7]的中断.

0 = Disable, 1 = Enable

UART1 [9] 控制 UART1 block的钟控

0 = Disable, 1 = Enable

UART0 [8] 控制 UART0 block的钟控

0 = Disable, 1 = Enable

BDMA0,1 [7] 控制 BDMA block的钟控,如果BDMA关断,在外设总线上的外设不能存取

0 = Disable, 1 = Enable

LCDC [6] 控制 LCDC block的钟控

0 = Disable, 1 = Enable

SIO [5] 控制 SIO block的钟控

0 = Disable, 1 = Enable

ZDMA0,1 [4] 控制 ZDMA block的钟控

0 = Disable, 1 = Enable

PWMTIMER [3] 控制 PWMTIMER block的钟控

0 = Disable, 1 = Enable

IDLE [2] 进入 IDLE mode.该位不能自动清除

0 = Disable, 1 =进入 IDLE mode

SL_IDLE [1] 进入SL_IDLE mode option. 该位不能自动清除.

为了进入SL_IDLE mode, CLKCON 寄存器必须等于 0x46.

0 = Disable, 1 = SL_IDLE mode.

STOP [0] 进入 STOP mode. 该位不能自动清除.

0 = Disable 1 =进入STOP mode

3 慢时钟控制寄存器

CLKSLOW 0x01D80008 R/W 慢时钟控制寄存器 初始值 0x9

位名称 BIT 描述

PLL_OFF [5]

0 : PLL 打开,. PLL 仅能在SLOW_BIT=1时打开,在PLL稳定后(150US), SLOW_BIT位可以清除

1 : PLL 关掉, PLL 仅能在SLOW_BIT=1时关掉

SLOW_BIT [4]

0 : $F_{out} = F_{pllo}$ (PLL output)

1: $F_{out} = F_{in} / (2 \times SLOW_VAL)$, ($SLOW_VAL > 0$) $F_{out} = F_{in}$, ($SLOW_VAL = 0$)

SLOW_VAL [3:0] 这四位是在SLOW_BIT 位打开时slow clock的分频值

4 锁定时间计数值寄存器

LOCKTIME 0x01D8000C R/W 锁定时间计数值寄存器 初始值 0xffff

6 CPU WRAPPER & BUS PRIORITIES

CPU WRAPPER 包括一个8KBYTEmemory, write buffer,和CPU 核.

8KBYTEmemory可以以三种方式使用 :

1. 全部8K作为指令/数据cache ,
2. 4K做为内部SRAM,另外4K做为cache
- 3 全部8K作为内部存储器使用.

Cache使用 最近最少使用算法来提高命中率,使用write-through策略保持数据一致性.

内部SRAM主要用来减少中断线程执行时间.

cache set 和LRU的存储映射地址

cache 和LRU	存储映射地址	尺寸
cache set 0	0x10000000 - 0x100007ff	2KB
cache set 1	0x10000800 - 0x10000fff	2KB
cache set 2	0x10001000 - 0x100017ff	2KB
cache set 3	0x10001800 - 0x10001fff	2KB
cache tag 0	0x10002000 - 0x100027f0	512bytes
cache tag 1	0x10002800 - 0x10002ff0	512bytes
cache tag 2	0x10003000 - 0x100037f0	512bytes
cache tag 3	0x10003800 - 0x10003ff0	512bytes
LRU	0x10004000 - 0x100047f0	512bytes

在cache set的地址连续增加,在cache tag和LRU中的地址以16BYTE来增加,每次读写一个字, 地址的 bit[3:0]必须是0.

S3C44B0X有四个写缓冲区寄存器,每个写缓冲区寄存器包括一个32BIT数据域,一个28BIT的地址域,指示写数据的地址,2BIT 的状态域MAS.

状态域MAS确定数据模式

00 = 8-bit data mode

01 = 16-bit data mode

10 = 32-bit data mode

11 = Not used

在S3C44B0X,有七个总线控制块: LCD_DMA, BDMA0, BDMA1, ZDMA0, ZDMA1, nBREQ (外部总线控制块),和 CPU wrapper.

在复位后优先级从高到低如下:

1. DRAM refresh controller
2. LCD_DMA
3. ZDMA0,1
4. BDMA0,1
5. External bus master

6. Write buffer

7. Cache & CPU

LCD_DMA, ZDMA, BDMA, and an external bus master的优先级可以通过SBUSCON寄存器编程改变,但CPU wrapper总是最低的优先级,不能改变

可以选择循环队列优先级模式和固定优先级模式,在循环队列优先级模式,所有的总线控制块都有相同的优先级,

寄存器定义.

1 系统配置寄存器

SYSCFG 0x01C00000 R/W 系统配置寄存器 默认值 0x01

位名称	位	描述
-----	---	----

Reserved	[7]	保留
----------	-----	----

Reserved	[6]	保留
----------	-----	----

DA(reserved)	[5]	数据异常控制,推荐值为0.
--------------	-----	---------------

0: 允许 数据异常 1: 不允许数据异常

RSE(reserved)	[4]	允许读停止选项. 推荐值为0.
---------------	-----	-----------------

读停止选项在cache & CPU core读数据时插入一个间歇等待周期

0: 不允许读停止 1: 允许读停止

WE	[3]	该位确定写缓冲的允许/不允许. 一些外部装置需要最小的写周期时间,如果允许写缓冲,在由于连续的写之间的周期将使之工作不正常.
----	-----	--

0 =不允许写缓冲操作 1 = 允许写缓冲操作

CM	[2:1]	这两位确定cache 模式
----	-------	---------------

00 = Disable cache (8KB internal SRAM)

01 = Half cache enable (4KB cache, 4KB internal SRAM)

10 = Reserved

11 = Full Cache enable (8KB cache)

SE	[0]	允许停止选项. 推荐值为0.
----	-----	----------------

停止选项在使用Cache,产生非连续地址时,插入一个间歇等待周期

0: 不允许停止选项1: 允许停止选项

2 NON-CACHEABLE区域控制寄存器

NCACHBE0 0x01C00004 R/W non-cacheable0区域的开始结束地址 0x00000000

NCACHBE1 0x01C00008 R/W non-cacheable1区域的开始结束地址1 0x00000000

位名称	位	描述
-----	---	----

SE0	[31:16]	non-cacheable0的结束地址, 因为on-cacheable的最小尺寸为4Kbyte。因此SE0计算如下.
-----	---------	--

$SE0 = (End\ address + 1)/4K$

SA0	[15:0]	non-cacheable0的开始地址, 因为on-cacheable的最小尺寸为4Kbyte。因此SA0计算如下.
-----	--------	--

$SA0 = Start\ address/4K$

SE1	[31:16]	non-cacheable1的结束地址, 因为on-cacheable的最小尺寸为4Kbyte。因此SE1计算如下.
-----	---------	--

$SE1 = (End\ address + 1)/4K$

SA1	[15:0]	non-cacheable1的开始地址, 因为on-cacheable的最小尺寸为
-----	--------	---

4Kbyte。因此SA0计算如下.

$$SA1 = \text{Start address}/4K$$

3 系统总线优先级控制寄存器

SBUSCON 0x01C40000 R/W 总线优先级控制寄存器 初始值 0x80001B1B

位名称	位	描述
FIX	[31]	优先级模式 0: round-robin priorities 1: fixed priorities
S_LCD_DMA	[15:14]	指示 LCD_DMA 的总线优先级（只读） 00: 1st 01: 2nd 10: 3rd 11: 4th
S_ZDMA	[13:12]	指示ZDMA的总线优先级（只读） 00: 1st 01: 2nd 10: 3rd 11: 4th
S_BDMA	[11:10]	指示BDMA的优先级（只读） 00: 1st 01: 2nd 10: 3rd 11: 4th
S_nBREQ	[9:8]	指示 nBREQ的优先级（只读） 00: 1st 01: 2nd 10: 3rd 11: 4th
LCD_DMA	[7:6]	确定LCD_DMA总线优先级 00: 1st 01: 2nd 10: 3rd 11: 4th
ZDMA	[5:4]	确定ZDMA 总线优先级 00: 1st 01: 2nd 10: 3rd 11: 4th
BDMA	[3:2]	确定BDMA 总线优先级 00: 1st 01: 2nd 10: 3rd 11: 4th
nBREQ	[1:0]	确定nBREQ总线优先级 00: 1st 01: 2nd 10: 3rd 11: 4 th

7 DMA

S3C44B0X有4路DMA控制器，其中两路称为ZDMA（General DMA）被连到SSB(Samsung System Bus)总线上，另外两路称为BDMA（Bridge DMA）是SSB和SPB(Samsung Peripheral Bus)之间的接口层，相当于一个桥，因此称为桥DMA。ZDMA和BDMA都可以由指令启动，也可以有内部外设和外部请求引脚来请求启动。ZDMA用来在存储器到存储器、存储器到IO存储器、IO装置到存储器之间传输数据。BDMA控制器只能在连到SPB上的IO装置（外设）与存储器之间传输数据。ZDMA最大的特性是on-the-fly模式，on-the-fly模式有不可分割的读写周期，在这点上ZDMA与普通的DMA不同，从而可以减少在外部存储器和外部可寻址的外设之间DMA操作的周期数。

对于ZDMA，S3C44B0X有一个4字的FIFO缓冲来支持4字突发DMA传输，而BDMA不支持突发DMA传输，因此存储器之间的传输数据最好用ZDMA传输，来提供高的传输速度。

2 外部DMA REQ/ACK 协议

对于ZDMA，有四种类型的外部DMA请求/应答协议。

握手模式

一个单独的应答对应一个单独的DMA请求，在该模式，DMA操作期间的读写周期不可分割，因此在一个DMA操作完成前，不能把总线让给其它总线控制器使用。一次nXDREQ请求引起一次DMA传输（可以是一个字节，一个半字或一个字）。

单步模式

单步模式意味着一次DMA传输有两个DMA应答周期（产生两个应答信号nXDACK）指示DMA读和写周期，主要用与测试和调试模式，在读写周期之间，总线控制权可以让给其它总线控制器。

Whole Service Mode

在该模式，一次DMA请求将产生连续的DMA传输，直到规定的DMA传输数传输完，在DMA传输期间，nXDACK一直有效，DMA请求信号被释放。并且在每次传输一个数据单元后，释放一次总线控制权，以便其它总线控制器有机会可以占用总线。

Demand Mode

在该模式，只要DMA请求信号一直有效，DMA传输就持续进行，并且一直占用总线控制权，因此应该预防传输周期超过规定的最大时间。

有三种DMA传输模式（单位传输模式，块传输模式，On the fly块传输模式）。

Unit 传输模式	1 个单位读, 然后 1个单位写
Block传输模式	4 个字突发读, 然后4 个字突发写，因此传输的数据个数应当是16字节的倍数。
On-the-fly 传输模式	1个单位读或 1个单位写，读写同时进行。

DMA寄存器

1 ZDMA控制寄存器

ZDCON0	0x01E80000	R/W	ZDMA 0 控制寄存器 初始值 0x00
ZDCON1	0x01E80020	R/W	ZDMA 1 控制寄存器 初始值 0x00

位名称	BIT	描述
INT	[7:6]	保留 00
STE	[5:4]	DMA通道的状态（只读） 在DMA的传输计数开始之前，STE呆在准备好状态
		00 = Ready 01 = Not TC yet
		10 = Terminal Count 11 = N/A
QDS	[3:2]	忽落/允许外部 DMA 请求 (nXDREQ) 00 = Enable other = Disable
CMD	[1:0]	软件命令

- 00: 没有命令.在写 01,10,11后, CMD 位被自动清除， nXDREQ允许
01: 由S/W启动DMA操作 S/W启动功能能用在whole mode.
10: 停止DMA操作. 但 nXDREQ仍允许
11: 取消DMA 操作

2 ZDMA0 初始源/目标地址和计数寄存器

ZDISRC0	0x01E80004	R/W	ZDMA 0初始源地址寄存器 初始值 0x00000000
ZDIDES0	0x01E80008	R/W	ZDMA 0初始目标地址寄存器 初始值 0x00000000
ZDICNT0	0x01E8000C	R/W	ZDMA 0初始计数寄存器 初始值 0x00000000

3 ZDMA0 当前源/目标地址和计数寄存器

ZDCSRC0	0x01E80010	R	ZDMA 0当前源地址寄存器 初始值 0x00000000
ZDCDES0	0x01E80014	R	ZDMA 0当前目标地址寄存器 初始值 0x00000000
ZDCCNT0	0x01E80018	R	ZDMA 0 当前计数寄存器 初始值 0x00000000

4 ZDMA1 初始源/目标地址和计数寄存器

ZDISRC1	0x01E80024	R/W	ZDMA 1初始源地址寄存器 初始值 0x00000000
ZDIDES1	0x01E80028	R/W	ZDMA 1初始目标地址寄存器 初始值 0x00000000

ZDICNT1 0x01E8002C R/W ZDMA 1初始计数寄存器 初始值 0x00000000

5 ZDMA1 当前/目标源地址和计数寄存器

ZDCSRC1 0x01E80030 R ZDMA 1当前源地址寄存器 初始值 0x00000000

ZDCDES1 0x01E80034 R ZDMA 1当前目标地址寄存器 初始值 0x00000000

ZDCCNT1 0x01E80038 R ZDMA 1 当前计数寄存器 初始值 0x00000000

ZDMA_n的初始当前源地址寄存器位定义

位名称	BIT	描述
-----	-----	----

DST	[31:30]	传输的数据类型，在块传输模式，DST必须是10
-----	---------	-------------------------

00 = Byte, 01 = Half word

10 = Word, 11 = Not used

DAL	[29:28]	加载地址变动方向
-----	---------	----------

00 = N/A, 01 = Increment

10 = Decrement, 11 = Fixed

ISADDR/CSADDR	[27:0]	ZDMA _n 的初始/当前源地址
---------------	--------	-----------------------------

ZDMA_n的初始当前目标地址寄存器位定义

位名称	BIT	描述
-----	-----	----

OPT	[31:30]	DMA内部选项推荐值 OPT = 10
-----	---------	---------------------

bit 31: 指示在单步模式 nXDREQ 如何采样

bit 30: 如果DST是半字或字和 DMA 模式不是块传输模式，该位起作用

1: DMA做字或半字交换

传输前: B0,B1,B2,B3,B4,B5,B6,B7...

字交换后: B3,B2,B1,B0,B7,B6,B5,B4,...

半字交换后: B1,B0,B3,B2,B5,B4,B7,B6,...

0: 正常

DAS	[29:28]	地址方向
-----	---------	------

00 = N/A 01 = Increment

10 = Decrement 11 = Fixed

IDADDR/CDADDR	[27:0]	ZDMA _n 的初始/当前目标地址
---------------	--------	------------------------------

ZAMA计数寄存器位定义

位名称	BIT	描述
-----	-----	----

QSC	[31:30]	选择DMA 请求源
-----	---------	-----------

00 = nXDREQ[0] 01 = nXDREQ[1]

10 = N/A 11 = N/A

QTY	[29:28]	DREQ 协议类型
-----	---------	-----------

00 = Handshake 01 = Single step

10 = Whole Service 11 = Demand

TMD	[27:26]	传输模式
-----	---------	------

00 = Not used 01 = Unit transfer mode

10 = Block(4-word) transfer mode 11 = On the fly

OTF	[25:24]	On the fly mode 有效
-----	---------	--------------------

00 = N/A 01 = N/A

10 = Read time on the fly 11 = Write time on the fly

INTS	[23:22]	中断模式设置
------	---------	--------

00 = 查询模式 01 = N/A

10 =无论什么时候传输都产生中断

11 =当中断计数时产生中断

AR [21] 在DMA计数到0时自动加载和自动开始

0 = Disable

1 = Enable.

EN [20] DMA H/W允许/不允许

0 = Disable DMA

1 = Enable DMA.

如果 S/W 命令取消, DMA 操作也将被取消和EN位被清除

在中断计数时, EN位也被清除。

注:不要同时设置ZDICNTD的EN位和其它位 用户必须在设置其它位后设置EN位, 步骤如下:

1. Set ZDICNT register with disabled En bit.

2. Set EN bit enable.

ICNT/CCNT [19:0] ZDMAn的初始/当前传输计数值, 必须正确设置

如果传输单位为字节, ICNT每次减小1

如果传输单位为半字, ICNT每次减小2

如果传输单位为字, ICNT每次减小4

6 BDMA控制寄存器

BDCON0 0x01F80000 R/W BDMA 0 控制寄存器 初始值 0x00

BDCON1 0x01F80020 R/W BDMA 1 控制寄存器 初始值 0x00

位名称 BIT 描述

INT [7:6] 保留 00

STE [5:4] DMA通道的状态 (只读)

在DMA的传输计数开始之前, STE呆在准备好状态

00 = Ready 01 = Not TC yet

10 = Terminal Count 11 = N/A

QDS [3:2] 忽略/允许外部 DMA 请求 (nXDREQ)

00 = Enable other = Disable

CMD [1:0] 软件命令

00: 没有命令.在写 01,10,11后, CMD 位被自动清除。 ,

01: 保留

10: 保留

11: 取消DMA 操作

7 BDMA0 初始源/目标地址和计数寄存器

BDISRC0 0x01F80004 R/W BDMA 0初始源地址寄存器 初始值 0x00000000

BDIDES0 0x01F80008 R/W BDMA 0初始目标地址寄存器 初始值 0x00000000

BDICNT0 0x01F8000C R/W BDMA 0初始计数寄存器 初始值 0x00000000

8 BDMA0 当前源/目标地址和计数寄存器

BDCSRC0 0x01F80010 R BDMA 0当前源地址寄存器 初始值 0x00000000

BDCDES0 0x01F80014 R BDMA 0当前目标地址寄存器 初始值 0x00000000

BDCCNT0 0x01F80018 R BDMA 0 当前计数寄存器 初始值 0x00000000

9 BDMA1 初始源/目标地址和计数寄存器

BDISRC1 0x01F80024 R/W BDMA 1初始源地址寄存器 初始值 0x00000000

BDIDES1 0x01F80028 R/W BDMA 1初始目标地址寄存器 初始值 0x00000000
 BDICNT1 0x01F8002C R/W BDMA 1初始计数寄存器 初始值 0x00000000

10 BDMA1 当前/目标源地址和计数寄存器

BDCSRC1 0x01F80030 R BDMA 1当前源地址寄存器 初始值 0x00000000
 BDCDES1 0x01F80034 R BDMA 1当前目标地址寄存器 初始值 0x00000000
 BDCCNT1 0x01F80038 R BDMA 1 当前计数寄存器 初始值 0x00000000

BDMA的初始当前源地址寄存器位定义

位名称	BIT	描述
DST	[31:30]	传输的数据类型
00 = Byte, 01 = Half word		
10 = Word, 11 = Not used		
DAL	[29:28]	加载地址变动方向
00 = N/A, 01 = Increment		
10 = Decrement, 11 = Fixed		

ISADDR/CSADDR [27:0] BDMA的初始/当前源地址

BDMA的初始当前目标地址寄存器位定义

位名称	BIT	描述
TDM	[31:30]	传输方向模式
00 = 保留		
01 = M2IO (从外部存储器到内部外设)		
10 = IO2M (从内部外设到外部存储器)		
11 = IO2IO (从内部外设到内部外设)		

注：即使不使用BAMA通道，你也必须改变该值。

DAS	[29:28]	地址方向
00 = N/A 01 = Increment		
10 = Decrement 11 = Fixed		

IDADDR/CDADDR [27:0] BDMA的初始/当前目标地址

BAMA计数寄存器位定义

位名称	BIT	描述
QSC	[31:30]	选择DMA 请求源
00 = N/A 01 = IIS		
10 = UART0 11 = SIO		

Reserved	[29:28]	
00 = Handshake		

Reserved	[27:26]	传输模式
01 = Unit transfer mode		

Reserved	[25:24]	00 = N/A
----------	---------	----------

INTS	[23:22]	中断模式设置
00 = 查询模式 01 = N/A		
10 = 无论什么时候传输都产生中断		
11 = 当中断计数时产生中断		

AR	[21]	在DMA计数到0时自动加载和自动开始
0 = Disable		
1 = Enable.		

EN [20] DMA H/W允许/不允许
0 = Disable DMA
1 = Enable DMA.

如果 S/W 命令取消, DMA 操作也将被取消和EN位被清除
在中断计数时, EN位也被清除。

注:不要同时设置BDICNTD的EN位和其它位 用户必须在设置其它位后设置EN位, 步骤如下:

1. Set BDICNT register with disabled En bit.
2. Set EN bit enable.

ICNT/CCNT [19:0] BDMA的初始/当前传输计数值.
如果传输单位为字节, ICNT每次减小1
如果传输单位为半字, ICNT每次减小2
如果传输单位为字, ICNT每次减小4

8 I/O PORTS

S3C44B0X有71个复合功能的I/O口引脚, 分成7个端口

端口E和F (9位I/O口)

端口D和G (8位I/O口)

端口C (16位I/O口)

端口A (10位I/O口)

端口B (11位I/O口)

在主程序开始前, 必须定义每个I/O管脚的功能。在特殊功能不用时, 作为I/O脚使用

Port A

	Function 1	Function 2	Function 3	Function 4
PA9	output	ADDR24		
PA8	output	ADDR23		
PA7	output	ADDR22		
PA6	output	ADDR21		
PA5	output	ADDR20		
PA4	output	ADDR19		
PA3	output	ADDR18		
PA2	output	ADDR17		
PA1	output	ADDR16		
PA0	output	ADDR0		
PB10	output	nGCS5		
PB9	output	nGCS4		
PB8	output	nGCS3		
PB7	output	nGCS2		
PB6	output	nGCS1		
PB5	output	nWBE3:nBE3:DQM3		
PB4	output	nWBE2:nBE2:DQM2		
PB3	output	nSRAS:nCAS3		
PB2	output	nSCAS:nCAS2		
PB1	output	SCLK		
PB0	output	SCKE		

PC15	Input/output	DATA31	nCTS0		
PC14	Input/output	DATA30	nRTS0		
PC13	Input/output	DATA29	RxD1		
PC12	Input/output	DATA28	TxD1		
PC11	Input/output	DATA27	nCTS1		
PC10	Input/output	DATA26	nRTS1		
PC9	Input/output	DATA25	nXDREQ1		
PC8	Input/output	DATA24	nXDACK1		
PC7	Input/output	DATA23	VD4		
PC6	Input/output	DATA22	VD5		
PC5	Input/output	DATA21	VD6		
PC4	Input/output	DATA20	VD7		
PC3	Input/output	DATA19	IISCLK		
PC2	Input/output	DATA18	IISDI		
PC1	Input/output	DATA17	IISDO		
PC0	Input/output	DATA16	IISLRCK		
PD7	Input/output	VFRAME			
PD6	Input/output	VM			
PD5	Input/output	VLINE			
PD4	Input/output	VCLK			
PD3	Input/output	VD3			
PD2	Input/output	VD2			
PD1	Input/output	VD1			
PD0	Input/output	VD0			
PE8	ENDIAN	CODECLK	input/output		
PE7	Input/output	TOUT4	VD7		
PE6	Input/output	TOUT3	VD6		
PE5	Input/output	TOUT2	TCLK		
PE4	Input/output	TOUT1	TCLK		
PE3	Input/output	TOUT0			
PE2	Input/output	RxD0			
PE1	Input/output	TxD0			
PE0	Input/output	Fpllo	Fout		
PF8	input/output	nCTS1	SIOCK	IISCLK	
PF7	input/output	RxD1	SIORxD	IISDI	
PF6	input/output	TxD1	SIORDY	IISDO	
PF5	input/output	nRTS1	SIOTxD	IISLRCK	
PF4	input/output	nXBREQ	nXDREQ0 –		
PF3	input/output	nXBACK	nXDACK0 –		
PF2	input/output	nWAIT – –			
PF1	input/output	IICSDA – –			
PF0	input/output	IIC SCL			
PG7	input/output	IISLRCK	EINT7		
PG6	input/output	IISDO	EINT6		

PG5	input/output	IISDI	EINT5
PG4	input/output	IISCLK	EINT4
PG3	input/output	nRTS0	EINT3
PG2	input/output	nCTS0	EINT2
PG1	input/output	VD5	EINT1
PG0	input/output	VD4	EINT0

I/O端口控制寄存器

1 PORT A 控制寄存器

PCONA	0x01D20000	R/W	port A的配置	初始值	0x3ff
PDATA	0x01D20004	R/W	port A的数据寄存器	初始值	Undef.

位名称	BIT	描述
PA9	[9]	0 = Output 1 = ADDR24
PA8	[8]	0 = Output 1 = ADDR23
PA7	[7]	0 = Output 1 = ADDR22
PA6	[6]	0 = Output 1 = ADDR21
PA5	[5]	0 = Output 1 = ADDR20
PA4	[4]	0 = Output 1 = ADDR19
PA3	[3]	0 = Output 1 = ADDR18
PA2	[2]	0 = Output 1 = ADDR17
PA1	[1]	0 = Output 1 = ADDR16
PA0	[0]	0 = Output 1 = ADDR0

数据寄存器位描述

PA[9:0]	[9:0]	当端口配置为输出口时，对应脚的状态和该位的值相同
		当端口配置作为功能脚时，如果读该位的值，将是一个不确定的值

2 PORT B 控制寄存器

PCONB	0x01D20008	R/W	port B的配置	初始值	0x7ff
PDATB	0x01D2000C	R/W	port B的数据寄存器	初始值	Undef.

位名称	BIT	描述
PB10	[10]	0 = Output 1 = nGCS5
PB9	[9]	0 = Output 1 = nGCS4
PB8	[8]	0 = Output 1 = nGCS3
PB7	[7]	0 = Output 1 = nGCS2
PB6	[6]	0 = Output 1 = nGCS1
PB5	[5]	0 = Output 1 = nWBE3/nBE3/DQM3
PB4	[4]	0 = Output 1 = nWBE2/nBE2/DQM2
PB3	[3]	0 = Output 1 = nSRAS/nCAS3
PB2	[2]	0 = Output 1 = nSCAS/nCAS2
PB1	[1]	0 = Output 1 = SCLK
PB0	[0]	0 = Output 1 = SCKE

数据寄存器位描述

PB[10:0]	[10:0]	当端口配置为输出口时，对应脚的状态和该位的值相同
		当端口配置作为功能脚时，如果读该位的值，将是一个不确定的值

3 PORT C 控制寄存器

PCONC	0x01D20010	R/W	port C的配置	初始值	0xaaaaaaaa
-------	------------	-----	-----------	-----	------------

PDATC	0x01D20014	R/W	port C的数据寄存器	初始值	Undef.
PUPC	0x01D20018	R/W	port C的上拉电阻配置寄存器	初始值	0x0
位名称	BIT		描述		
PC15	[31:30]		00 = Input 01 = Output 10 = DATA31 11 = nCTS0		
PC14	[29:28]		00 = Input 01 = Output 10 = DATA30 11 = nRTS0		
PC13	[27:26]		00 = Input 01 = Output 10 = DATA29 11 = RxD1		
PC12	[25:24]		00 = Input 01 = Output 10 = DATA28 11 = TxD1		
PC11	[23:22]		00 = Input 01 = Output 10 = DATA27 11 = nCTS1		
PC10	[21:20]		00 = Input 01 = Output 10 = DATA26 11 = nRTS1		
PC9	[19:18]		00 = Input 01 = Output 10 = DATA25 11 = nXDREQ1		
PC8	[17:16]		00 = Input 01 = Output 10 = DATA24 11 = nXDACK1		
PC7	[15:14]		00 = Input 01 = Output 10 = DATA23 11 = VD4		
PC6	[13:12]		00 = Input 01 = Output 10 = DATA22 11 = VD5		
PC5	[11:10]		00 = Input 01 = Output 10 = DATA21 11 = VD6		
PC4	[9:8]		00 = Input 01 = Output 10 = DATA20 11 = VD7		
PC3	[7:6]		00 = Input 01 = Output 10 = DATA19 11 = IISCLK		
PC2	[5:4]		00 = Input 01 = Output 10 = DATA18 11 = IISDI		
PC1	[3:2]		00 = Input 01 = Output 10 = DATA17 11 = IISDO		
PC0 [1:0]			00 = Input 01 = Output 10 = DATA16 11 = IISLRCK		

数据寄存器位描述

PC[15:0]	[15:0]	当端口配置为输入口时，该位的值是对应脚的状态 当端口配置为输出口时，对应脚的状态和该位的值相同 当端口配置作为功能脚时，如果读该位的值，将是一个不确定的值
----------	--------	---

上拉电阻配置寄存器位描述

PC[15:0]	[15:0]	0: 允许上拉电阻连接到对应脚 1: 不允许.
----------	--------	----------------------------

4 PORT D 控制寄存器

PCOND	0x01D2001C	R/W	port D的配置	初始值	0x0000
-------	------------	-----	-----------	-----	--------

PDATD	0x01D20020	R/W	port D的数据寄存器	初始值	Undef.
PUPD	0x01D20024	R/W	port D的上拉电阻配置寄存器	初始值	0x0
位名称	BIT		描述		
PD7	[15:14]		00 = Input 01 = Output 10 = VFRAME 11 = Reserved		
PD6	[13:12]		00 = Input 01 = Output 10 = VM 11 = Reserved		
PD5	[11:10]		00 = Input 01 = Output 10 = VLINE 11 = Reserved		
PD4	[9:8]		00 = Input 01 = Output 10 = VCLK 11 = Reserved		
PD3	[7:6]		00 = Input 01 = Output 10 = VD3 11 = Reserved		
PD2	[5:4]		00 = Input 01 = Output 10 = VD2 11 = Reserved		
PD1	[3:2]		00 = Input 01 = Output 10 = VD1 11 = Reserved		
PD0 [1:0]			00 = Input 01 = Output 10 = VD0 11 = Reserved		

数据寄存器位描述

PD[15:0]	[15:0]	当端口配置为输入时，该位的值是对应脚的状态 当端口配置为输出时，对应脚的状态和该位的值相同 当端口配置作为功能脚时，如果读该位的值，将是一个不确定的值
----------	--------	---

上拉电阻配置寄存器位描述

PD[15:0]	[15:0]	0: 允许上拉电阻连接到对应脚 1: 不允许.
----------	--------	----------------------------

5 PORT E 控制寄存器

PCONE	0x01D20028	R/W	port E的配置	初始值	0x00
PDATE	0x01D2002C	R/W	port E的数据寄存器	初始值	Undef.
PUPE	0x01D20030	R/W	port E的上拉电阻配置寄存器	初始值	0x00
位名称	BIT		描述		
PE8	[17:16]		00 = Reserved(ENDIAN) 01 = Output 10 = CODECLK 11 = Reserved		
PE7	[15:14]		00 = Input 01 = Output 10 = TOUT4 11 = VD7		
PE6	[13:12]		00 = Input 01 = Output 10 = TOUT3 11 = VD6		
PE5	[11:10]		00 = Input 01 = Output 10 = TOUT2 11 = TCLK in		
PE4	[9:8]		00 = Input 01 = Output 10 = TOUT1 11 = TCLK in		
PE3	[7:6]		00 = Input 01 = Output 10 = TOUT0 11 = Reserved		
PE2	[5:4]		00 = Input 01 = Output		

		10 = RxD0 11 = Reserved
PE1	[3:2]	00 = Input 01 = Output
		10 = TxD0 11 = Reserved
PE0	[1:0]	00 = Input 01 = Output
		10 = FpI/O out 11 = Fout out

数据寄存器位描述

PE[8:0]	[8:0]	当端口配置为输出口时，对应脚的状态和该位的值相同
		当端口配置作为功能脚时，如果读该位的值，将是一个不确定的值

上拉电阻配置寄存器位描述

PE[7:0]	[7:0]	0: 允许上拉电阻连接到对应脚
		1: 不允许.

PE8没有可编程的上拉电阻

6 PORT F 控制寄存器

PCONF	0x01D20034	R/W	port F的配置	初始值	0x0000
PDATF	0x01D20038	R/W	port F的数据寄存器	初始值	Undef.
PUPF	0x01D2003C	R/W	port F的上拉电阻配置寄存器	初始值	0x000
位名称	BIT		描述		
PF8	[21:19]		000 = Input 001 = Output 010 = nCTS1 011 = SIOCLK 100 = IISCLK Others = Reserved		
PF7	[18:16]		000 = Input 001 = Output 010 = RxD1 011 = SIORxD 100 = IISDI Others = Reserved		
PF6	[15:13]		000 = Input 001 = Output 010 = TxD1 011 = SIORDY 100 = IISDO Others = Reserved		
PF5	[12:10]		000 = Input 001 = Output 010 = nRTS1 011 = SIOTxD 100 = IISLRCK Others = Reserved		
PF4	[9:8]		00 = Input 01 = Output 10 = nXBREQ 11 = nXDREQ0		
PF3	[7:6]		00 = Input 01 = Output 10 = nXBACK 11 = nXDACK0		
PF2	[5:4]		00 = Input 01 = Output 10 = nWAIT 11 = Reserved		
PF1	[3:2]		00 = Input 01 = Output 10 = IICSDA 11 = Reserved		
PF0	[1:0]		00 = Input 01 = Output 10 = IICSCL 11 = Reserved		

数据寄存器位描述

PF[8:0]	[8:0]	当端口配置为输入口时，该位的值是对应脚的状态
		当端口配置为输出口时，对应脚的状态和该位的值相同
		当端口配置作为功能脚时，如果读该位的值，将是一个不确定的值

上拉电阻配置寄存器位描述

PF[8:0]	[8:0]	0: 允许上拉电阻连接到对应脚
		1: 不允许.

7 PORT G 控制寄存器

PCONG	0x01D20040	R/W	port G的配置	初始值	0x0
-------	------------	-----	-----------	-----	-----

PDATG	0x01D20044	R/W	port G的数据寄存器	初始值	Undef.
PUPG	0x01D20048	R/W	port G的上拉电阻配置寄存器	初始值	0x0
位名称	BIT		描述		
PG7	[15:14]		00 = Input 01 = Output 10 = IISLRCK 11 = EINT7		
PG6	[13:12]		00 = Input 01 = Output 10 = IISDO 11 = EINT6		
PG5	[11:10]		00 = Input 01 = Output 10 = IISDI 11 = EINT5		
PG4	[9:8]		00 = Input 01 = Output 10 = IISCLK 11 = EINT4		
PG3	[7:6]		00 = Input 01 = Output 10 = nRTS0 11 = EINT3		
PG2	[5:4]		00 = Input 01 = Output 10 = nCTS0 11 = EINT2		
PG1	[3:2]		00 = Input 01 = Output 10 = VD5 11 = EINT1		
PG0	[1:0]		00 = Input 01 = Output 10 = VD4 11 = EINT0		

数据寄存器位描述

PG[7:0]	[7:0]	当端口配置为输入口时，该位的值是对应脚的状态 当端口配置为输出口时，对应脚的状态和该位的值相同 当端口配置作为功能脚时，如果读该位的值，将是一个不确定的值
---------	-------	---

上拉电阻配置寄存器位描述

PG[7:0]	[7:0]	0: 允许上拉电阻连接到对应脚 1: 不允许.
---------	-------	----------------------------

8 上拉电阻控制寄存器 控制D[15 : 0]脚的上拉电阻。

SPUCR	0x01D2004C	R/W	上拉电阻控制寄存器	初始值	0x4
位名称	BIT		描述		
HZ@STOP	[2]		0 =在停止模式存储器的控制信号保持先前的状态 1 = 控制信号保持高阻状态		
SPUCR1	[1]		0 = DATA[15:8] 上拉电阻允许 1 = DATA[15:8] 上拉电阻不允许		
SPUCR0	[0]		0 = DATA[7:0] 上拉电阻允许 1 = DATA[7:0] 上拉电阻不允许		

8 外部中断控制寄存器

EXTINT	0x01D20050	R/W	外部中断控制寄存器	初始值	0x000000
位名称	BIT		描述		
EINT7	[30:28]		000 = 低电平触发 001 =高电平中断 01x = 下降沿触发 10x = 上升沿触发 11x =双边沿触发		
EINT6	[26:24].		000 = 低电平触发 001 =高电平中断 01x = 下降沿触发 10x = 上升沿触发 11x =双边沿触发		

EINT5	[22:20]	000 = 低电平触发 001 =高电平中断 01x = 下降沿触发 10x = 上升沿触发 11x =双边沿触发
EINT4	[18:16]	000 = 低电平触发 001 =高电平中断 01x = 下降沿触发 10x = 上升沿触发 11x =双边沿触发
EINT3	[14:12]	000 = 低电平触发 001 =高电平中断 01x = 下降沿触发 10x = 上升沿触发 11x =双边沿触发
EINT2	[10:8]	000 = 低电平触发 001 =高电平中断 01x = 下降沿触发 10x = 上升沿触发 11x =双边沿触发
EINT1	[6:4]	000 = 低电平触发 001 =高电平中断 01x = 下降沿触发 10x = 上升沿触发 11x =双边沿触发
EINT0	[2:0]	000 = 低电平触发 001 =高电平中断 01x = 下降沿触发 10x = 上升沿触发 11x =双边沿触发

9 外部中断挂起寄存器

外部中断请求4、5、6、7通过或逻辑公用一个中断请求线。

EXTINTPND 0x01D20054 R/W 外部中断挂起寄存器 初始值 0x00

位名称	BIT	描述
EXTINTPND3	[3]	如果 EINT7激活, EXINTPND3设置为1, INTPND[21] 也设置为1
EXTINTPND2	[2]	如果 EINT6激活, EXINTPND2设置为1, INTPND[21]也设置1
EXTINTPND1	[1]	如果 EINT5激活, EXINTPND1设置为1, INTPND[21]也设置1
EXTINTPND0	[0]	如果 EINT4激活, EXINTPND0设置为1, INTPND[21]也设置为1

中断服务线程必须在清除外部挂起条件EXTINTPND后清除中断挂起条件（INTPND），EXTINTPND通过写入1来清除。

9 PWM TIMER

S3C44B0X有六个16位定时器，每个可以以中断模式或DMA模式来工作。定时器0、1、2、3、4有PWM功能，定时器5是一个内部定时器没有输出脚。定时器0有一个DEAD ZONE 产生器。定时器0、1共享一个8位预定标器，定时器2、3共享另一个8位预定标器，定时器4、5也共享一个8位预定标器。定时器0、1、2、3有一个时钟除法器(1/2, 1/4, 1/8, 1/16, 1/32)。定时器4、5有一个时钟除法器(1/2, 1/4, 1/8, 1/16)和一个输入时钟TCLK/EXTCLK。

每个定时器从时钟除法器接收时钟输入，而时钟除法器的输入时钟为相应的预定标器

每个定时器有一个计数缓冲寄存器，保持定时器的计数初始值，当定时器允许和它的向下计数器计到0时，该初始值重新加载到定时器的向下计数器，并产生中断请求。每个定时器还有一个比较缓冲寄存器，初始值加载到定时器的比较寄存器，当比较寄存器的值与定时器的向下计数器值相同时，定时器控制逻辑改变TOUT的输出电平。

由于双缓冲特征。TCMPn和TCNTBn可以随时加载，在下一个定时周期起作用。

8位预定标器和4位时钟除法器可以形成下列除出频率

4-bit divider settings	(prescaler = 1)	(prescaler = 255)	(TCNTBn = 65535)
1/2 (MCLK = 66 MHz)	0.030 us (33.0 MHz)	7.75 us (58.6 KHz)	0.50 sec

1/4 (MCLK = 66 MHz)	0.060 us (16.5 MHz)	15.5 us (58.6 KHz)	1.02 sec
1/8 (MCLK = 66 MHz)	0.121 us (8.25 MHz)	31.0 us (29.3 KHz)	2.03 sec
1/16 (MCLK = 66 MHz)	0.242 us (4.13 MHz)	62.1 us (14.6 KHz)	4.07 sec
1/32 (MCLK = 66 MHz)	0.485 us (2.06 MHz)	125 us (7.32 KHz)	8.13 sec

无论何时和定时器是否运行，TOUT 反转器的 on/off 位变化，TOUT逻辑都变化。

DEAD ZONE 产生器

DEAD ZONE 产生器使TOUT0 和nTOUT0 (TOUT0的反转信号) 输出不能同时打开，而是之间有一个小的间隙。

DMA 请求模式

PWM定时器能在规定时间内产生一个DMA请求，定时器在收到ACK信号之前一直保持DMA请求信号为低。通过设置TCFG1寄存器，6个定时器之一能产生一个DMA请求，不在产生中断请求。

DMA mode DMA request 配置

DMA mode DMA request

0000	No select
0001	Timer0
0010	Timer1
0011	Timer2
0100	Timer3
0101	Timer4
0110	Timer5
0111	No select

定时器控制寄存器

1 定时器配置寄存器0

TCFG0 0x01D50000 R/W 三个 8位预定标器的配置 初始值 0x00000000

位名称	BIT	描述
Dead zone length	[31:24]	这 8 位确定dead zone的长度. dead zone的长度的一个时间单位与定时器0的一个时间单位相等.
Prescaler 2	[23:16]	这8位确定定时器4、5的预定标器的值
Prescaler 1	[15:8]	这8位确定定时器2、3的预定标器的值
Prescaler 0	[7:0]	这8位确定定时器0、1的预定标器的值

2 定时器配置寄存器1

TCFG1 0x01D50004 R/W MUX和DMA模式选择寄存器 初始值 0x00000000

位名称	BIT	描述
DMA mode	[27:24]	选择DMA请求通道 0000 = No select (all interrupt) 0001 = Timer0 0010 = Timer1 0011 = Timer2 0100 = Timer3 0101 = Timer4 0110 = Timer5 0111 = Reserved
MUX 5	[23:20]	选择定时器5的MUX输入 . 0000 = 1/2 0001 = 1/4 0010 = 1/8 0011 = 1/16 01xx = EXTCLK
MUX 4	[19:16]	选择定时器4的MUX输入. 0000 = 1/2 0001 = 1/4 0010 = 1/8

		0011 = 1/16 01xx = TCLK
MUX 3	[15:12]	选择定时器3的MUX输入. 0000 = 1/2 0001 = 1/4 0010 = 1/8 0011 = 1/16 01xx = 1/32
MUX 2	[11:8]	选择定时器2的MUX输入. 0000 = 1/2 0001 = 1/4 0010 = 1/8 0011 = 1/16 01xx = 1/32
MUX 1	[7:4]	选择定时器1的MUX输入. 0000 = 1/2 0001 = 1/4 0010 = 1/8 0011 = 1/16 01xx = 1/32
MUX 0	[3:0]	选择定时器0的MUX输入. 0000 = 1/2 0001 = 1/4 0010 = 1/8 0011 = 1/16 01xx = 1/32

3 定时器控制寄存器

TCON 0x01D50008 R/W 定时器控制寄存器 初始值 0x00000000

位名称	BIT	描述
Timer 5 auto reload on/off	[26]	这位确定定时器5的自动加载的开/关 0 = One-shot 1 = Interval mode (auto reload)
Timer 5 manual update	[25]	这位确定定时器5的手动更新 0 = No operation 1 = Update TCNTB5
Timer 5 start/stop	[24]	这位确定定时器5的启动/停止. 0 = Stop 1 = Start for Timer 5
Timer 4 auto reload on/off	[23]	这位确定定时器4的自动加载的开/关 0 = One-shot 1 = Interval mode (auto reload)
Timer 4 outputinverter on/off	[22]	这位确定定时器4的输出反转器的开/关 0 = Inverter off 1 = Inverter on for TOUT4
Timer 4 manual update	[21]	这位确定定时器4的手动更新 0 = No operation 1 = Update TCNTB4 , TCMPB4
Timer 4 start/stop	[20]	这位确定定时器5的启动/停止. 0 = Stop 1 = Start for Timer 4
Timer 3 auto reload on/off	[19]	这位确定定时器3的自动加载的开/关 0 = One-shot 1 = Interval mode (auto reload)
Timer 3 outputinverter on/off	[18]	这位确定定时器3的输出反转器的开/关 0 = Inverter off 1 = Inverter on for TOUT3
Timer 3 manual update	[17]	这位确定定时器3的手动更新 0 = No operation 1 = Update TCNTB3 , TCMPB3
Timer 3 start/stop	[16]	这位确定定时器3的启动/停止. 0 = Stop 1 = Start for Timer 3
Timer 2 auto reload on/off	[15]	这位确定定时器2的自动加载的开/关 0 = One-shot 1 = Interval mode (auto reload)
Timer 2 outputinverter on/off	[14]	这位确定定时器2的输出反转器的开/关 0 = Inverter off 1 = Inverter on for TOUT2
Timer 2 manual update	[13]	这位确定定时器2的手动更新 0 = No operation 1 = Update TCNTB2 , TCMPB2

Timer 2 start/stop [12] 这位确定定时器2的启动/停止.
0 = Stop 1 = Start for Timer 2

Timer 1 auto reload on/off [11] 这位确定定时器1的自动加载的开/关
0 = One-shot 1 = Interval mode (auto reload)

Timer 1 outputinverter on/off [10] 这位确定定时器1的输出反转器的开/关
0 = Inverter off 1 = Inverter on for TOUT1

Timer 1 manual update [9] 这位确定定时器1的手动更新
0 = No operation 1 = Update TCNTB1 , TCMPB1

Timer 1 start/stop [8] 这位确定定时器1的启动/停止.
0 = Stop 1 = Start for Timer 1

Dead zone enable [4] 这位确定dead zone的操作.
0 = Disable 1 = Enable

Timer 0 auto reload on/off [4] 这位确定定时器0的自动加载的开/关
0 = One-shot 1 = Interval mode (auto reload)

Timer 0 outputinverter on/off [3] 这位确定定时器0的输出反转器的开/关
0 = Inverter off 1 = Inverter on for TOUT0

Timer 0 manual update [2] 这位确定定时器0的手动更新
0 = No operation 1 = Update TCNTB0 , TCMPB0

Timer 0 start/stop [1] 这位确定定时器0的启动/停止.
0 = Stop 1 = Start for Timer 0

4 定时器0计数/比较缓冲寄存器

TCNTB0	0x01D5000C	R/W	定时器0计数缓冲寄存器	初始值	0x00000000
TCMPB0	0x01D50010	R/W	定时器0比较缓冲寄存器	初始值	0x00000000

5 定时器0计数观察寄存器

TCNTO0	0x01D50014	R	定时器0计数观察寄存器	初始值	0x00000000
--------	------------	---	-------------	-----	------------

6 定时器1计数/比较缓冲寄存器

TCNTB1	0x01D50018	R/W	定时器1计数缓冲寄存器	初始值	0x00000000
TCMPB1	0x01D5001C	R/W	定时器1比较缓冲寄存器	初始值	0x00000000

7 定时器1计数观察寄存器

TCNTO1	0x01D50020	R	定时器1计数观察寄存器	初始值	0x00000000
--------	------------	---	-------------	-----	------------

8 定时器2计数/比较缓冲寄存器

TCNTB2	0x01D50024	R/W	定时器2计数缓冲寄存器	初始值	0x00000000
TCMPB2	0x01D50028	R/W	定时器2比较缓冲寄存器	初始值	0x00000000

9 定时器2计数观察寄存器

TCNTO2	0x01D5002C	R	定时器2计数观察寄存器	初始值	0x00000000
--------	------------	---	-------------	-----	------------

10 定时器3计数/比较缓冲寄存器

TCNTB3	0x01D50030	R/W	定时器3计数缓冲寄存器	初始值	0x00000000
TCMPB3	0x01D50034	R/W	定时器3比较缓冲寄存器	初始值	0x00000000

11 定时器3计数观察寄存器

TCNTO3	0x01D50038	R	定时器3计数观察寄存器	初始值	0x00000000
--------	------------	---	-------------	-----	------------

12 定时器4计数/比较缓冲寄存器

TCNTB4	0x01D5003C	R/W	定时器4计数缓冲寄存器	初始值	0x00000000
TCMPB4	0x01D50040	R/W	定时器4比较缓冲寄存器	初始值	0x00000000

13 定时器4计数观察寄存器

TCNTO4	0x01D50044	R	定时器4计数观察寄存器	初始值	0x00000000
14 定时器5计数/比较缓冲寄存器					
TCNTB5	0x01D50048	R/W	定时器5计数缓冲寄存器	初始值	0x00000000
15 定时器5计数观察寄存器					
TCNTO5	0x01D5004C	R	定时器5计数观察寄存器	初始值	0x00000000

10 UART

S3C44B0X UART单元提供了两个异步串口（SIO），每个SIO可以操作在中断模式或DMA模式，支持波特率最大115.2kbps，每个SIO通道包含都有一个16字节的接收与发送FIFO缓冲区。

UART 发送接收单元提供可编程的波特率,红外发送/接收（IrDA 1.0），一个开始位，一个或两个停止位,5、6、7、8位数据位和奇偶校验功能，波特率产生器可以有MCLK钟控。发送器也可以产生暂停条件，暂停条件强制串口输出保持在逻辑0状态比一场长的时间。接收器可以检测溢出错误，奇偶错误，frame 错误和暂停条件。其中每一个设置一个错误标志。溢出错误在老的数据还没有读取新的数据到达，frame 错误是接收数据没有有效的停止位，暂停条件指示Rx/Dn输入保持逻辑0状态比一场长的时间。

当在FIFO模式接收缓冲区FIFO不为空时，在可以接收3个字的时间期间没有接收到数据时接收超时。

自动流控制（AFC）

S3C44B0X的 UART在UART连接UART时支持用nRTS 和nCTS信号进行自动流控制，在UART连接MODEM时不支持自动流控制，这时由软件控制nRTS和nCTS信号。

在自动流控制时，nRTS由接收条件控制，发送运行由nCTS信号控制，发送时当nCTS信号有效时（意味着对方的UART准备好接收数据），发送器传输数据到FIFO中。在接收数据时，当FIFO有多于两个字节的空闲时，nRTS激活（指示接收FIFO准备好接收数据），FIFO剩余空间在一个字节以下时nRTS无效。

在非自动流控制操作时：

接收操作：1 选择接收模式（中断或BDMA）

- 2 检查在UFSTATn寄存器中的Rx FIFO计数值，如果值小于15，用户必须设置UMCONn[0]为1（激活nRTS），否则设置UMCONn[0]为0（不激活nRTS）。
- 3 重复步骤2

发送操作：1 选择发送模式（中断或BDMA）

- 2 检查UMSTATn[0]的值，如果值为1（nCTS激活），用户可以发送数据到发送缓冲区或发送FIFO寄存器。

每个UART有七个状态：溢出错误，奇偶错误，frame 错误和暂停条件，接收FIFO/缓冲准备好，发送FIFO/缓冲为空，发送移位寄存器为空。这些状态由UART状态寄存器指示。

溢出错误，奇偶错误，frame 错误和暂停条件作为接收错误状态可以引起接收错误中断请求。

如果在中断模式，当接收器传送接收移位寄存器的数据到接收FIFO时，激活FIFO满状态引起接收中断。当发送器从发送FIFO传送数据到发送移位寄存器后，发送FIFO空状态激活引起发送中断。

接收FIFO满状态和发送FIFO空状态信号在DMA模式用来产生DMA请求。

中断产生条件总结如下

类型	FIFO模式	非FIFO模式
Rx中断	1 FIFO满状态产生上时	接收移位寄存器的数据满时
	2 接收超时	

Tx 中断	发送FIFO空状态激活	每次发送数据完成时
Error中断	1 检测到奇偶错误，frame 错误和暂停条件	所有的错误立即产生一个中断，如果
	2当接收FIFO到达顶端时（溢出错误）	错误同时出现，只产生一个中断

UART还有一个错误状态FIFO，错误状态FIFO指示在FIFO寄存器中的数据接收到一个错误。错误中断仅在具有错误的数据准备读出时产生。

波特率产生

时钟源通过一个16分频器和一个16位分频器产生波特率时钟。16位分频器的值由UART 波特率分频寄存器 (UBRDIVn)规定，值从1到($2^{16}-1$)。公式如下：

$$UBRDIVn = (\text{round_off})(MCLK / (\text{bps} \times 16)) - 1.$$

Loop-back 模式

在该模式，发送的数据立即被接收，主要用来测试。

IR (Infrared) 模式

S3C44B0X UART支持红外发送和接收。

在红外发送时，数据0的发送脉冲是位宽度的3/16。接收时，采样脉冲是位时间的1/16，因此RX脉冲宽度必须大于采样脉冲宽度。

UART 寄存器

1 UART线控制 寄存器

ULCON0	0x01D00000	R/W	UART0的线控制 寄存器	初始值	0x00
ULCON1	0x01D04000	R/W	UART1的线控制 寄存器	初始值	0x00
位名称	BIT	描述			
Reserved	[7]				
Infra-Red Mode	[6]	该位确定是否使用红外模式 0 = Normal mode operation 1 = Infra-Red Tx/Rx mode			
Parity Mode	[5:3]	该位确定奇偶如何产生和校验 0xx = No parity 100 = Odd parity 101 = Even parity 110 = Parity forced/checked as 1 111 = Parity forced/checked as 0			
stop bit	[2]	该位确定停止位的个数 0 = One stop bit per frame 1 = Two stop bit per frame			
Word length	[1:0]	该位确定数据位的个数 00 = 5-bits 01 = 6-bits 11 = 7-bits 11 = 8-bits			

2 UART控制寄存器

UCON0	0x01D00004	R/W	UART0控制寄存器	初始值	0x00
UCON1	0x01D04004	R/W	UART1控制寄存器	初始值	0x00
位名称	BIT	描述			
Tx interrupt type	[9]	发送中断请求类型 0 = Pulse 1 = Level			
Rx interrupt type	[8]	接收中断请求类型 0 = Pulse			

		1 = Level
Rx time out enable	[7]	允许/不允许Rx超时中断 0 = Disable 1 = Enable
Rx error status interrupt enable	[6]	允许/不允许UART错误中断 0 = Do not generate receive error status interrupt 1 = Generate receive error status interrupt
Loop-back Mode	[5]	该位为1使UART进入loop back 模式 0 = Normal operation 1 = Loop-back mode
Send Break Signal	[4]	该位为1使UART发送一个暂停条件，该位在发送一个暂停信号后自动清除 0 = Normal transmit 1 = Send break signal
Transmit Mode	[3:2]	这两位确定哪个模式可以写TX数据到UART发送保持寄存器 00 = Disable 01 = Interrupt request or polling mode 10 = BDMA0 request (Only for UART0) 11 = BDMA1 request (Only for UART1)
Receive Mode	[1:0]	这两位确定哪个模式可以从UART接收缓冲寄存器读数据 00 = Disable, 01 = Interrupt request or polling mode 10 = BDMA0 request (Only for UART0) 11 = BDMA1 request (Only for UART1)

3 UART FIFO 控制寄存器

UFCON0	0x01D00008	R/W	UART0 FIFO控制寄存器 初始值 0x00
UFCON1	0x01D04008	R/W	UART1 FIFO控制寄存器 初始值 0x00
位名称	BIT	描述	
Tx FIFO Trigger Level	[7:6]	这两位确定发送FIFO的触发条件 00 = Empty 01 = 4-byte 10 = 8-byte 11 = 12-byte	
Rx FIFO TriggerLevel	[5:4]	这两位确定接收FIFO的触发条件 00 = 4-byte 01 = 8-byte 10 = 12-byte 11 = 16-byte	
Reserved	[3]		
Tx FIFO Reset	[2]	TX FIFO复位位，该位在FIFO复位后自动清除 0 = Normal 1= Tx FIFO reset	
Rx FIFO Reset	[1]	Rx FIFO复位位，该位在FIFO复位后自动清除 0 = Normal 1= Rx FIFO reset	
FIFO Enable	[0]	0 = FIFO disable 1 = FIFO mode	

4 UART MODEM 控制寄存器

UMCON0	0x01D0000C	R/W	UART0 MODEM控制寄存器 初始值 0x00
UMCON1	0x01D0400C	R/W	UART1 MODEM控制寄存器 初始值 0x00

位名称	BIT	描述
Reserved	[7:5]	这两位必须为 0
AFC(Auto Flow Control)	[4]	AFC是否允许 0 = Disable 1 = Enable 0
Reserved	[3:1]	这两位必须为 0
Request to Send	[0]	如果AFC允许,该位忽略 0 = 'H' level(Inactivate nRTS) 1 = 'L' level(Activate nRTS)
5 UART TX/RX状态寄存器		
UTRSTAT0	0x01D00010	R UART0 TX/RX状态寄存器 初始值 0x6
UTRSTAT1	0x01D04010	R UART1 TX/RX状态寄存器 初始值 0x6
位名称	BIT	描述
Transmit shifter Empty	[2]	该位在发送移位寄存器没有有效的数据或发送移位寄存器为空时为1 0 = Not empty 1 = Transmit holding & shifter register empty
Transmit buffer Empty	[1]	该位在发送缓冲寄存器没有包含有效的数据为1 如果 UART使用FIFO, 用户应当检查UFSTAT寄存器的Tx FIFO计数位和Tx FIFO满标志位代替检查该位。 0 =The buffer register is not empty 1 = Empty
Receive buffer data ready	[0]	无论何时接收缓冲寄存器包含有效数据, 该位为1 如果 UART使用FIFO, 用户应当检查UFSTAT寄存器的Rx FIFO计数位代替检查该位。 0 = Completely empty 1 = The buffer register has a received data
6 UART 错误状态寄存器		
UERSTAT0	0x01D00014	R UART0 错误状态寄存器 初始值 0x6
UERSTAT1	0x01D04014	R UART1 错误状态寄存器 初始值 0x6
位名称	BIT	描述
Break Detect	[3]	该位为1指示一个暂停信号已经接收到 0 = No break receive 1 = Break receive
Frame Error	[2]	该位为1指示一个frame 错误发生 0 = No frame error during receive 1 = Frame error
Parity Error	[1]	该位为1指示在接收时一个奇偶错误发生 0 = No parity error during receive

1 = Parity error

Overrun Error [0] 该位为1指示一个溢出错误发生
 0 = No overrun error during receive
 1 = Overrun error

注：当UART错误状态寄存器时，UERSATn[3:0]自动清除。

7 UART FIFO状态寄存器

UFSTAT0	0x01D00018	R	UART0 FIFO状态寄存器	初始值	0x6
UFSTAT1	0x01D04018	R	UART1 FIFO状态寄存器	初始值	0x6

位名称	BIT	描述
Reserved	[15:10]	
Tx FIFO Full	[9]	当发送 FIFO满时该位为1 0 = 0-byte ≤ Tx FIFO data ≤ 15-byte 1 = Full
Rx FIFO Full	[8]	当接收 FIFO满时该位为1 0 = 0-byte ≤ Rx FIFO data ≤ 15-byte 1 = Full
Tx FIFO Count	[7:4]	Tx FIFO里的数据数量
Rx FIFO Count	[3:0]	Rx FIFO里的数据数量

8 UART **MODEM**状态寄存器

UMSTAT0	0x01D0001C	R	UART0 MODEM 状态寄存器	初始值	0x6
UMSTAT1	0x01D0401C	R	UART1 MODEM 状态寄存器	初始值	0x6

位名称	BIT	描述
Delta CTS	[4]	该位指示输入到S3C44B0X的nCTS信号自从上次读后已经改变状态 0 = Has not changed 1 = Has changed
Reserved	[3:1]	Reserved
Clear to Send	[0]	0 = CTS signal is not activated(nCTS pin is high) 1 = CTS signal is activated(nCTS pin is low)

9 UART 发送缓冲寄存器和FIFO寄存器

UTXH0	0x01D00020(Little endian) 0x01D00023(Big endian)	W(byte)	UART0发送缓冲寄存器	初始值	-
UTXH1	0x01D04020(Little endian) 0x01D04023(Big endian)	W(byte)	UART1发送缓冲寄存器	初始值	-

10 UART 接收缓冲寄存器和FIFO寄存器

URXH0	0x01D00024(Little endian) 0x01D00027(Big endian)	R(byte)	UART0接收缓冲寄存器	初始值	-
URXH1	0x01D04024(Little endian) 0x01D04027(Big endian)	R(byte)	UART1接收缓冲寄存器	初始值	-

注:当溢出错误出现时, URXHn,否则,即使USTATn的溢出位已经清除,下一个接收的数据也将使溢出错误出现.

11 UART 波特率分频寄存器

UBRDIV0	0x01D00028	R/W	UART0波特率分频寄存器	初始值	-
UBRDIV1	0x01D04028	R/W	UART1波特率分频寄存器	初始值	-

11 INTERRUPT CONTROLLER

S3C44B0X的中断控制器有30个中断源。

S3C44B0X支持新的中断处理模式称为（ vectored interrupt mode ），在多个中断请求发生时，由硬件优先级逻辑确定应该有哪个中断得到服务 ,同时硬件逻辑使中断相量表的跳转指令加载到（ 0X18或0X1C ）位置，在该位置执行跳转指令使程序跳到相应的中断服务线程，因此相对与传统的ARM的软件方法能够大大减少中断进入延时。

分支指令机器代码= 0xea000000 +((<destination address> - <vector address> - 0x8)>>2)

destination address为中断服务线程ISR 的开始地址

vector address 为中断源在中断相量表中的地址，即分支指令所在地址。

分支指令机器代码有硬件自动产生。

中断源在中断相量表中的位置如下

中断源	向量地址
EINT0	0x00000020
EINT1	0x00000024
EINT2	0x00000028
EINT3	0x0000002c
EINT4/5/6/7	0x00000030
INT_TICK	0x00000034
INT_ZDMA0	0x00000040
INT_ZDMA1	0x00000044
INT_BDMA0	0x00000048
INT_BDMA1	0x0000004c
INT_WDT	0x00000050
INT_UERR0/1	0x00000054
INT_TIMER0	0x00000060
INT_TIMER1	0x00000064
INT_TIMER2	0x00000068
INT_TIMER3	0x0000006c
INT_TIMER4	0x00000070
INT_TIMER5	0x00000074
INT_URXD0	0x00000080
INT_URXD1	0x00000084
INT_IIC	0x00000088
INT_SIO	0x0000008c
INT_UTXD0	0x00000090
INT_UTXD1	0x00000094
INT_RTC	0x000000a0
INT_ADC	0x000000c0

有两种类型的中断模式，FIQ（快速中断）和 IRQ.所有的中断源在中断请求时应该确定使用的中断模式。

中断源	描述	Master Group	Slave ID
EINT0	外部中断 0	mGA	sGA
EINT1	外部中断 1	mGA	sGB

EINT2	外部中断 2	mGA	sGC
EINT3	外部中断3	mGA	sGD
EINT4/5/6/7	外部中断4/5/6/7	mGA	sGKA
TICK_RTC	定时器滴答中断	mGA	sGKB
INT_ZDMA0	DMA0 中断	mGB	sGA
INT_ZDMA1	DMA1 中断	mGB	sGB
INT_BDMA0	桥DMA0 中断	mGB	sGC
INT_BDMA1	桥DMA1 中断	mGB	sGD
INT_WDT	看门狗定时器中断	mGB	sGKA
INT_UERR0/1	UART0/1错误中断	mGB	sGKB
INT_TIMER0	定时器0中断	mGC	sGA
INT_TIMER1	定时器1中断	mGC	sGB
INT_TIMER2	定时器2中断	mGC	sGC
INT_TIMER3	定时器3中断	mGC	sGD
INT_TIMER4	定时器4中断	mGC	sGKA
INT_TIMER5	定时器5中断	mGC	sGKB
INT_URXD0	UART0接收中断	mGD	sGA
INT_URXD1	UART1接收中断	mGD	sGB
INT_IIC	IIC 中断	mGD	sGC
INT_SIO	SIO 中断	mGD	sGD
INT_UTXD0	UART0发送中断	mGD	sGKA
INT_UTXD1	UART1发送中断	mGD	sGKB
INT_RTC	RTC 警告中断	mGKA –	
INT_ADC	ADC 结束中断	mGKB –	

中断优先级产生模块

对于IRQ中断请求有一个中断优先级产生模块，如果中断向量模式使用和一个中断源被配置为ISQ中断，中断将被中断优先级产生模块处理。中断优先级产生模块处理包括五个单元：1个master单元，4个slave单元，每个slave单元管理6个中断源，包括4个可编程的优先级源(sGn)和2个固定优先级源(sGKn，其优先级在6个优先级源中最低，其中sGKA的优先级高于sGKB的优先级)。一个master单元管理4个slave单元mGn和2个中断源mGKn，用来确定4个slave单元，和2个中断源的优先级mGKn，其中4个slave单元的优先级次序可编程，其中的2个中断源INT_RTC 和 INT_ADC中断源在26个中断源中优先级最低，并且INT_RTC的优先级高于INT_ADC的优先级。

中断寄存器

1 中断控制寄存器

INTCON	0x01E00000	R/W	中断控制寄存器	0x7
位名称	BIT	名称		
Reserved	[3]			
V	[2]	该位允许IRQ使用向量模式		
		0 = Vectored interrupt mode		
		1 = Non-vectored interrupt mode		
I	[1]	该位允许IRQ中断		
		0 = IRQ interrupt enable		
		1 = Reserved		

注 :在使用IRQ中断之前该位必须清除

F [0] 该位允许FIQ中断
0 = FIQ interrupt enable (FIQ中断不支持向量中断模式)
1 = Reserved

注: 在使用FIQ中断之前该位必须清除

1 中断挂起寄存器

INTPND	0x01E00004	R/W	指示中断请求状态	0x0000000
位名称	BIT		名称	
EINT0	[25]		0 = Not requested, 1 = Requested	
EINT1	[24]		0 = Not requested, 1 = Requested	
EINT2	[23]		0 = Not requested, 1 = Requested	
EINT3	[22]		0 = Not requested, 1 = Requested	
EINT4/5/6/7	[21]		0 = Not requested, 1 = Requested	
INT_TICK	[20]		0 = Not requested, 1 = Requested	
INT_ZDMA0	[19]		0 = Not requested, 1 = Requested	
INT_ZDMA1	[18]		0 = Not requested, 1 = Requested	
INT_BDMA0	[17]		0 = Not requested, 1 = Requested	
INT_BDMA1	[16]		0 = Not requested, 1 = Requested	
INT_WDT	[15]		0 = Not requested, 1 = Requested	
INT_UERR0/1	[14]		0 = Not requested, 1 = Requested	
INT_TIMER0	[13]		0 = Not requested, 1 = Requested	
INT_TIMER1	[12]		0 = Not requested, 1 = Requested	
INT_TIMER2	[11]		0 = Not requested, 1 = Requested	
INT_TIMER3	[10]		0 = Not requested, 1 = Requested	
INT_TIMER4	[9]		0 = Not requested, 1 = Requested	
INT_TIMER5	[8]		0 = Not requested, 1 = Requested	
INT_URXD0	[7]		0 = Not requested, 1 = Requested	
INT_URXD1	[6]		0 = Not requested, 1 = Requested	
INT_IIC	[5]		0 = Not requested, 1 = Requested	
INT_SIO	[4]		0 = Not requested, 1 = Requested	
INT_UTXD0	[3]		0 = Not requested, 1 = Requested	
INT_UTXD1	[2]		0 = Not requested, 1 = Requested	
INT_RTC	[1]		0 = Not requested, 1 = Requested	
INT_ADC	[0]		0 = Not requested, 1 = Requested	

3 中断模式寄存器

INTMOD	0x01E00008	R/W	中断模式寄存器	0x0000000
位名称	BIT		名称	
EINT0	[25]		0 = IRQ mode 1 = FIQ mode	
EINT1	[24]		0 = IRQ mode 1 = FIQ mode	
EINT2	[23]		0 = IRQ mode 1 = FIQ mode	
EINT3	[22]		0 = IRQ mode 1 = FIQ mode	
EINT4/5/6/7	[21]		0 = IRQ mode 1 = FIQ mode	
INT_TICK	[20]		0 = IRQ mode 1 = FIQ mode	
INT_ZDMA0	[19]		0 = IRQ mode 1 = FIQ mode	

INT_ZDMA1	[18]	0 = IRQ mode 1 = FIQ mode
INT_BDMA0	[17]	0 = IRQ mode 1 = FIQ mode
INT_BDMA1	[16]	0 = IRQ mode 1 = FIQ mode
INT_WDT	[15]	0 = IRQ mode 1 = FIQ mode
INT_UERR0/1	[14]	0 = IRQ mode 1 = FIQ mode
INT_TIMER0	[13]	0 = IRQ mode 1 = FIQ mode
INT_TIMER1	[12]	0 = IRQ mode 1 = FIQ mode
INT_TIMER2	[11]	0 = IRQ mode 1 = FIQ mode
INT_TIMER3	[10]	0 = IRQ mode 1 = FIQ mode
INT_TIMER4	[9]	0 = IRQ mode 1 = FIQ mode
INT_TIMER5	[8]	0 = IRQ mode 1 = FIQ mode
INT_URXD0	[7]	0 = IRQ mode 1 = FIQ mode
INT_URXD1	[6]	0 = IRQ mode 1 = FIQ mode
INT_IIC	[5]	0 = IRQ mode 1 = FIQ mode
INT_SIO	[4]	0 = IRQ mode 1 = FIQ mode
INT_UTXD0	[3]	0 = IRQ mode 1 = FIQ mode
INT_UTXD1	[2]	0 = IRQ mode 1 = FIQ mode
INT_RTC	[1]	0 = IRQ mode 1 = FIQ mode
INT_ADC	[0]	0 = IRQ mode 1 = FIQ mode

3 中断屏蔽寄存器

INTMSK 0x01E0000C R/W 确定哪个中断源被屏蔽,屏蔽的中断源将不被服务 0x07ffff

位名称	BIT	描述
Reserved	[27]	
Global	[26]	0 = Service available 1 = Masked
EINT0	[25]	0 = Service available 1 = Masked
EINT1	[24]	0 = Service available 1 = Masked
EINT2	[23]	0 = Service available 1 = Masked
EINT3	[22]	0 = Service available 1 = Masked
EINT4/5/6/7	[21]	0 = Service available 1 = Masked
INT_TICK	[20]	0 = Service available 1 = Masked
INT_ZDMA0	[19]	0 = Service available 1 = Masked
INT_ZDMA1	[18]	0 = Service available 1 = Masked
INT_BDMA0	[17]	0 = Service available 1 = Masked
INT_BDMA1	[16]	0 = Service available 1 = Masked
INT_WDT	[15]	0 = Service available 1 = Masked
INT_UERR0/1	[14]	0 = Service available 1 = Masked
INT_TIMER0	[13]	0 = Service available 1 = Masked 1
INT_TIMER1	[12]	0 = Service available 1 = Masked
INT_TIMER2	[11]	0 = Service available 1 = Masked
INT_TIMER3	[10]	0 = Service available 1 = Masked
INT_TIMER4	[9]	0 = Service available 1 = Masked
INT_TIMER5	[8]	0 = Service available 1 = Masked
INT_URXD0	[7]	0 = Service available 1 = Masked
INT_URXD1	[6]	0 = Service available 1 = Masked

INT_IIC	[5]	0 = Service available 1 = Masked
INT_SIO	[4]	0 = Service available 1 = Masked
INT_UTXD0	[3]	0 = Service available 1 = Masked
INT_UTXD1	[2]	0 = Service available 1 = Masked
INT_RTC	[1]	0 = Service available 1 = Masked
INT_ADC	[0]	0 = Service available 1 = Masked

3 IRQ中断向量模式寄存器

寄存器	地址	读/写	描述	初始值
I_PSLV	0x01E00010	R/W	确定slave组的IRQ优先级	0x1b1b1b1b
I_PMST	0x01E00014	R/W	master寄存器的IRQ优先级	0x00001f1b
I_CSLV	0x01E00018	R	当前slave寄存器的IRQ优先级	0x1b1b1b1b
I_CMST	0x01E0001C	R	当前master寄存器的IRQ优先级	0x0000xx1b
I_ISPR	0x01E00020	R	IRQ中断服务挂起寄存器 (同时仅能一个服务位被设置)	0x00000000

I_ISPC	0x01E00024	W	IRQ中断服务清除寄存器	Undef.
F_ISPC	0x01E0003C	W	FIQ 中断服务清除寄存器	Undef.

(对I_ISPC/ F_ISPC写入1,INTPND 将被自动清除)

在ISR中断服务线程的结束，INTPND必须清除。

为了清除I_ISPC/F_ISPC,应该遵守如下两个规则：

- 1 I_ISPC/F_ISPC寄存器仅能在ISR中断服务线程中存取。
- 2 通过写I_ISPC/F_ISPC寄存器来清除I_ISPR/INTPND寄存器

如果不遵守这两个规则，在中断请求发生时，I_ISPR/INTPND寄存器可以为0。

I_PSLV的位描述

位名称	BIT	描述	初始值
PSLAVE@mGA	[31:24]	确定mGA中的sGA, B, C, D的优先级 每个sGn必须有不同的优先级	
PSLAVE@mGB	[23:16]	确定mGB中的sGA, B, C, D的优先级 每个sGn必须有不同的优先级	
PSLAVE@mGC	[15:8]	确定mGC中的sGA, B, C, D的优先级 每个sGn必须有不同的优先级	
PSLAVE@mGD	[7:0]	确定mGD中的sGA, B, C, D的优先级 每个sGn必须有不同的优先级	
PSLAVE@mGA	BIT	描述	初始值
sGA (EINT0)	[31:30]	00: 1 st 01: 2 nd 10: 3 rd 11: 4 th	00
sGB (EINT1)	[29:28]	00: 1 st 01: 2 nd 10: 3 rd 11: 4 th	01
sGC (EINT2)	[27:26]	00: 1 st 01: 2 nd 10: 3 rd 11: 4 th	10
sGD (EINT3)	[25:24]	00: 1 st 01: 2 nd 10: 3 rd 11: 4 th	11
PSLAVE@mGB			
sGA (INT_ZDMA0)	[23:22]	00: 1 st 01: 2 nd 10: 3 rd 11: 4 th	00
sGB (INT_ZDMA1)	[21:20]	00: 1 st 01: 2 nd 10: 3 rd 11: 4 th	01
sGC (INT_BDMA0)	[19:18]	00: 1 st 01: 2 nd 10: 3 rd 11: 4 th	10
sGD (INT_BDMA1)	[17:16]	00: 1 st 01: 2 nd 10: 3 rd 11: 4 th	11
PSLAVE@mGC			

sGA (TIMER0)	[15:14]	00: 1 st 01: 2 nd 10: 3 rd 11: 4 th	00
sGB (TIMER1)	[13:12]	00: 1 st 01: 2 nd 10: 3 rd 11: 4 th	01
sGC (TIMER2)	[11:10]	00: 1 st 01: 2 nd 10: 3 rd 11: 4 th	10
sGD (TIMER3)	[9:8]	00: 1 st 01: 2 nd 10: 3 rd 11: 4 th	11

PSLAVE@mGD

sGA (INT_URXD0)	[7:6]	00: 1 st 01: 2 nd 10: 3 rd 11: 4 th	00
sGB (INT_URXD1)	[5:4]	00: 1 st 01: 2 nd 10: 3 rd 11: 4 th	01
Sgc (INT_IIC)	[3:2]	00: 1 st 01: 2 nd 10: 3 rd 11: 4 th	10
sGD (INT_SIO)	[1:0]	00: 1 st 01: 2 nd 10: 3 rd 11: 4 th	11

注：I_PSLAVE的项目必须配置不同的优先级，即使在相应的中断源不使用的情况下。

I_PMST寄存器的位描述

位名称	BIT	描述
Reserved	[15:13]	
M	[12]	Master 操作模式 0 = round robin 1 = fix mode
FxSLV[A:D]	[11:8]	Slave操作模式 0 = round robin 1 = fix mode
PMaster	[7:0]	确定4个slave单元的优先级

FxSLV

Fx@mGA	[11]	确定slave单元 @mGA的操作模式
Fx@mGB	[10]	确定slave单元 @mGB的操作模式
Fx@mGC	[9]	确定slave单元 @mGC的操作模式
Fx@mGD	[8]	确定slave单元 @mGD的操作模式

PMaster

mGA	[7:6]	00: 1 st 01: 2 nd 10: 3 rd 11: 4 th
mGB	[5:4]	00: 1 st 01: 2 nd 10: 3 rd 11: 4 th
mGC	[3:2]	00: 1 st 01: 2 nd 10: 3 rd 11: 4 th
mGD	[1:0]	00: 1 st 01: 2 nd 10: 3 rd 11: 4 th

注：I_PMST的项目必须配置不同的优先级，即使在相应的中断源不使用的情况下

I_CSLV寄存器的位描述

位名称	BIT	描述	初始值
CSLAVE@mGA	[31:24]	指示mGA中的sGA, B, C, D的当前优先级	0x1b
CSLAVE@mGB	[23:16]	指示mGB中的sGA, B, C, D的当前优先级	0x1b
CSLAVE@mGC	[15:8]	指示mGC中的sGA, B, C, D的当前优先级	0x1b
CSLAVE@mGD	[7:0]	指示mGD中的sGA, B, C, D的当前优先级	0x1b

CSLAVE@mGA

sGA (EINT0)	[31:30]	00: 1 st 01: 2 nd 10: 3 rd 11: 4 th	00
sGB (EINT1)	[29:28]	00: 1 st 01: 2 nd 10: 3 rd 11: 4 th	01
sGC (EINT2)	[27:26]	00: 1 st 01: 2 nd 10: 3 rd 11: 4 th	10
sGD (EINT3)	[25:24]	00: 1 st 01: 2 nd 10: 3 rd 11: 4 th	11

CSLAVE@mGB

sGA (INT_ZDMA0)	[23:22]	00: 1 st 01: 2 nd 10: 3 rd 11: 4 th	00
sGB (INT_ZDMA1)	[21:20]	00: 1 st 01: 2 nd 10: 3 rd 11: 4 th	01
sGC (INT_BDMA0)	[19:18]	00: 1 st 01: 2 nd 10: 3 rd 11: 4 th	10

sGD (INT_BDMA1) [17:16] 00: 1st 01: 2nd 10: 3rd 11: 4th 11

CSLAVE@mGC

sGA (TIMER0) [15:14] 00: 1st 01: 2nd 10: 3rd 11: 4th 00

sGB (TIMER1) [13:12] 00: 1st 01: 2nd 10: 3rd 11: 4th 01

sGC (TIMER2) [11:10] 00: 1st 01: 2nd 10: 3rd 11: 4th 10

sGD (TIMER3) [9:8] 00: 1st 01: 2nd 10: 3rd 11: 4th 11

CSLAVE@mGD

sGA (INT_URXD0) [7:6] 00: 1st 01: 2nd 10: 3rd 11: 4th 00

sGB (INT_URXD1) [5:4] 00: 1st 01: 2nd 10: 3rd 11: 4th 01

sGC (INT_IIC) [3:2] 00: 1st 01: 2nd 10: 3rd 11: 4th 10

sGD (INT_SIO) [1:0] 00: 1st 01: 2nd 10: 3rd 11: 4th 11

I_CMST寄存器的位描述

位名称	BIT	描述	初始值
Reserved	[15:14]		0
VECTOR	[13:8]	对应分支机器代码的低6位	unknown
CMASTER	[7:0]	master的当前优先级	00011011

CMASTER

mGA [7:6] 00: 1st 01: 2nd 10: 3rd 11: 4th 00

mGB [5:4] 00: 1st 01: 2nd 10: 3rd 11: 4th 01

mGC [3:2] 00: 1st 01: 2nd 10: 3rd 11: 4th 10

mGD [1:0] 00: 1st 01: 2nd 10: 3rd 11: 4th 11

I_ISPR寄存器的位描述

位名称	BIT	描述	初始值
EINT0	[25]	0 = not serviced 1 = serviced now	0
EINT1	[24]	0 = not serviced 1 = serviced now	0
EINT2	[23]	0 = not serviced 1 = serviced now	0
EINT3	[22]	0 = not serviced 1 = serviced now	0
EINT4/5/6/7	[21]	0 = not serviced 1 = serviced now	0
INT_TICK	[20]	0 = not serviced 1 = serviced now	0
INT_ZDMA0	[19]	0 = not serviced 1 = serviced now	0
INT_ZDMA1	[18]	0 = not serviced 1 = serviced now	0
INT_BDMA0	[17]	0 = not serviced 1 = serviced now	0
INT_BDMA1	[16]	0 = not serviced 1 = serviced now	0
INT_WDT	[15]	0 = not serviced 1 = serviced now	0
INT_UERR0/1	[14]	0 = not serviced 1 = serviced now	0
INT_TIMER0	[13]	0 = not serviced 1 = serviced now	0
INT_TIMER1	[12]	0 = not serviced 1 = serviced now	0
INT_TIMER2	[11]	0 = not serviced 1 = serviced now	0
INT_TIMER3	[10]	0 = not serviced 1 = serviced now	0
INT_TIMER4	[9]	0 = not serviced 1 = serviced now	0
INT_TIMER5	[8]	0 = not serviced 1 = serviced now	0
INT_URXD0	[7]	0 = not serviced 1 = serviced now	0
INT_URXD1	[6]	0 = not serviced 1 = serviced now	0
INT_IIC	[5]	0 = not serviced 1 = serviced now	0

INT_SIO	[4]	0 = not serviced 1 = serviced now	0
INT_UTXD0	[3]	0 = not serviced 1 = serviced now	0
INT_UTXD1	[2]	0 = not serviced 1 = serviced now	0
INT_RTC	[1]	0 = not serviced 1 = serviced now	0
INT_ADC	[0]	0 = not serviced 1 = serviced now	0

I_ISPC/F_ISPC寄存器的位描述

位名称	BIT	描述	初始值
EINT0	[25]	0 = No change 1 = clear the pending bit	0
EINT1	[24]	0 = No change 1 = clear the pending bit	0
EINT2	[23]	0 = No change 1 = clear the pending bit	0
EINT3	[22]	0 = No change 1 = clear the pending bit	0
EINT4/5/6/7	[21]	0 = No change 1 = clear the pending bit	0
INT_TICK	[20]	0 = No change 1 = clear the pending bit	0
INT_ZDMA0	[19]	0 = No change 1 = clear the pending bit	0
INT_ZDMA1	[18]	0 = No change 1 = clear the pending bit	0
INT_BDMA0	[17]	0 = No change 1 = clear the pending bit	0
INT_BDMA1	[16]	0 = No change 1 = clear the pending bit	0
INT_WDT	[15]	0 = No change 1 = clear the pending bit	0
INT_UERR0/1	[14]	0 = No change 1 = clear the pending bit	0
INT_TIMER0	[13]	0 = No change 1 = clear the pending bit	0
INT_TIMER1	[12]	0 = No change 1 = clear the pending bit	0
INT_TIMER2	[11]	0 = No change 1 = clear the pending bit	0
INT_TIMER3	[10]	0 = No change 1 = clear the pending bit	0
INT_TIMER4	[9]	0 = No change 1 = clear the pending bit	0
INT_TIMER5	[8]	0 = No change 1 = clear the pending bit	0
INT_URXD0	[7]	0 = No change 1 = clear the pending bit	0
INT_URXD1	[6]	0 = No change 1 = clear the pending bit	0
INT_IIC	[5]	0 = No change 1 = clear the pending bit	0
INT_SIO	[4]	0 = No change 1 = clear the pending bit	0
INT_UTXD0	[3]	0 = No change 1 = clear the pending bit	0
INT_UTXD1	[2]	0 = No change 1 = clear the pending bit	0
INT_RTC	[1]	0 = No change 1 = clear the pending bit	0
INT_ADC	[0]	0 = No change 1 = clear the pending bit	0

12 LCD 控制器

LCD 控制器用来把定位在系统存储器中的视频缓冲区的LCD图象数据传输到LCD驱动器，并产生必须的LCD控制信号。

LCD 控制器使用时间抖动算法和帧速率控制方法，支持在灰白LCD上的单色、4级灰度（一个点占两位）、16级灰度（一个点占4位）显示，也能与彩色LCD的接口支持最大256色（每点8位）的显示。

LCD 控制器可以编程支持不同水平和垂直点数（640x480, 320x240, 160x160等等）、不同数据线宽度、不同的接口时序和刷新速率的LCD，支持4位双扫描、4位单扫描、8位单扫描的LCD显示器，并支持水平/垂直卷动，用来支持更大的屏幕显示（如1280*1280）。

LCD显示数据流如下：

当LCDCDMA (LCD专用DMA , 用来在不用CPU参与的情况下自动传输帧存储器的视频数据到 LCD驱动器显示)中的FIFO 存储区为空或部分为空时 ,LCDCDMA请求从帧存储器预取数据(使用突发传输模式 , 一次预取4个字 , 在传输期间 , 不允许总线控制权转让。FIFO 存储区总的尺寸是24个字 (12个FIFOL , 12个FIFOH , 用来支持双扫描 , 在单扫描模式 , 仅一个12个FIFOH 可用) 。

灰度显示：4级灰度显示模式使用查找表，允许在16级可能的灰度中选择四级灰度显示，该查找表和彩色查找表的兰色查找表公用一个寄存器BLUEVAL[15:0]，灰度0由BLUEVAL[3:0]值表示，灰度1由BLUEVAL[7:4]值表示，灰度2由BLUEVAL[11:8]值表示，灰度3由BLUEVAL[15:12]值表示。在4级灰度显示模式不使用查找表。

彩色8位显示模式：3位分配为红，3位绿，可以同时显示8个红色与8个绿色，，2位蓝色位，可以同时显示4个兰色，合起来最大显示256色。红、绿、蓝分别使用不同的查找表，红色、绿色查找表入口都是32位（并分成8组），分别由REDVAL[31:0]、GREENVAL[31:0]寄存器指示，兰色查找表入口为16位，由BLUEVAL[15:0] 寄存器指示，并分成4组。也就是说红、绿色可以在32位中颜色组合中选择8色进行显示，兰色可以在16位中颜色组合中选择4色进行显示。

LCD自刷新模式：S3C44B0X支持LCD自刷新模式，以减少电源消耗，这时VCLK为低，VD保持先前的值，其它信号不变，这时电源管理模式可以进入SL_IDLE模式。

LCD寄存器

1 LCD控制寄存器1

LCDCON1	0x01F00000	R/W	LCD控制寄存器1	初始值 0x00000000
位名称	BIT	描述		
LINECNT(只读)	[31:22]	这些位反映行计数值		
CLKVAL	[21:12]	这些位确定VCLK的频率. 如果该值在ENVID=1改变,将下一帧使用. 公式为 $VCLK = MCLK / (CLKVAL \times 2) (CLKVAL \geq 2)$		
WLH	[11:10]	这些位确定VLINE高电平的宽度 00 = 4 clock, 01 = 8 clock, 10 = 12 clock, 11 = 16 clock		
WDLY	[9:8]	这些位确定VLINE和VCLK之间的延时 00 = 4clock, 01 = 8 clock, 10 = 12 clock, 11 = 16 clock		
MMODE	[7]	这位确定VM的改变速度. 0 = Each Frame, 1 = The rate defined by the MVAL		
DISMODE	[6:5]	这些位选择显示模式 00 = 4-bit dual scan display mode 01 = 4-bit single scan display mode 10 = 8-bit single scan display mode 11 = Not used		
INVCLK	[4]	该位控制VCLK 的极性 0 = The video data is fetched at VCLK falling edge 1 = The video data is fetched at VCLK rising edge		
INVLIN	[3]	该位指示行脉冲的极性 0 = normal 1 = inverted		
INVFRAME	[2]	该位指示帧脉冲的极性 0 = normal 1 = inverted		
INVVD	[1]	该位指示(VD[7:0]) 的极性. 0 = Normal		

1 = VD[7:0] output is inverted.

ENVID [0] LCD视频输出和逻辑的允许与否
0 = 不允许,LCD FIFO 清除
1 = 允许

2 LCD控制寄存器1

LCDCON2 0x01F00004 R/W LCD控制寄存器2 初始值 0x00000000

位名称	BIT	描述
-----	-----	----

LINEBLANK [31:21] 这些位确定行扫描的返回时间. LINEBLANK I的单位是MCLK.

如: LINEBLANK 为10, 返回时间在10个系统时钟期间插入VCLK

HOZVAL [20:10] 这些位确定LCD屏的水平尺寸,HOZVAL值的确定必须满足一行总的字节数是2的倍数 如120点的LCD 的水平尺寸X=120不支持,因为一行包含15个字节,而X=128可以被支持(16个字节),额外的8点将被LCD驱动器放弃.

公式:

$HOZVAL = (Horizontal display size / Number of the valid VD data line) - 1$

彩色模式: $Horizontal display size = 3 * Number of Horizontal Pixel$

LINEVAL [9:0] 这些位确定LCD屏的垂直尺寸

确定公式:

$LINEVAL = (Vertical display size) - 1$:单扫描类型

$LINEVAL = (Vertical display size / 2) - 1$: 双扫描类型

3 LCD控制寄存器3

LCDCON3 0x01F00040 R/W LCD控制寄存器3 初始值 0x00

位名称	BIT	描述
-----	-----	----

Reserved [2:1] 保留

SELFREF [0] LCD刷新模式允许位

0 : LCD self refresh mode disable

1 : LCD self refresh mode enable

4 帧缓冲区开始地址1

LCDSADDR1 0x01F00008 R/W 帧缓冲区开始地址1寄存器 初始值 0x000000

位名称	BIT	描述
-----	-----	----

MODESEL [28:27] 这些位选择显示模式

00 = monochrome mode 01 = 4-level gray mode

10 = 16-level gray mode 11 = color mode

LCDBANK [26:21] 这些位指示视频缓冲区在系统存储器的段地址A[27:22]

LCDBANK在视点移动时不能变化, LCD 帧缓冲区应当与4M区域对齐,因此在分配存储区应当注意

LCDBASEU [20:0] 这些位指示帧缓冲区或在双扫描LCD时的上帧缓冲区的开始地址 A[21:1]

注:1 LCDBANK在ENVID=1时不能变化.

2 如果LCDBASEU,LCDBASEL在ENVID=1时变化,新的量将在下一帧起作用,

5 帧缓冲区开始地址2

LCDSADDR2 0x01F0000C R/W 帧缓冲区开始地址2寄存器 初始值 0x000000

位名称	BIT	描述
-----	-----	----

BSWP	[29]	字节交换控制位 1 : Swap Enable 0 : Swap Disable LCD DMA 以4个字的突发模式预取帧存储区数据,在little endian模式和BSWP = 0, 帧存储区数据以n+3th, 4n+2th, 4n+1th, 4n-th次序显示 否则BSWP =1次序为4n-th, 4n+1th, 4n+2th, 4n+3th. 如果CPU在little endian模式,帧存储区可以以字节存取模式存取,因为BSWP =1,存取的数据也将在little endian模式正常显示,否则BSWP 必须为0
MVAL	[28:21]	如果MMODE=1,这两位定义VM信号以什么速度变化 公式为: $VM\ Rate = VLINE\ Rate / (2 * MVAL)$
LCDBASEL	[20:0]	这两位指示在使用双扫描LCD时的下帧存储区的开始地址 A[21:1] 公式如下: $LCDBASEL = LCDBASEU + (PAGEWIDTH + OFFSIZE) \times (LINEVAL + 1)$ 注:用户通过改变LCDBASEU 和 LCDBASEL的值来滚动屏幕,但在帧结束时,不能改变LCDBASEU 和 LCDBASEL的值,因为预取下一帧的数据优先于改变帧,如果这时改变帧, 预取的数据将无效和将显示不正确.为了检查LINECNT,中断应当被屏蔽,否则如果在读LINECNT后,任意中断刚好执行,因为ISR的执行,LINECNT的值可能无效.,

6 帧缓冲区开始地址3

LCDSADDR3	0x01F00010	R/W	虚拟屏幕地址设置	初始值 0x0000000
位名称	BIT		描述	
OFFSIZE	[19:9]		虚拟屏幕偏移量(半字的数量),该值定义前一显示行的最后的半字和新的显示一行首先的半字之间的距离	
PAGEWIDTH	[8:0]		虚拟屏幕宽度(半字的数量),该值定义帧的观察区域的宽度	

注: PAGEWIDTH 和 OFFSIZE必须在ENVID = 0时变化.

7 红色查找表寄存器

REDLUT	0x01F00014	R/W	红色查找表寄存器	初始值 0x00000000
位名称	BIT		描述	
REDVAL	[31:0]		这些位定义8组的每一组中的16个影射的哪一个将选择 000 = REDVAL[3:0], 001 = REDVAL[7:4] 010 = REDVAL[11:8], 011 = REDVAL[15:12] 100 = REDVAL[19:16], 101 = REDVAL[23:20] 110 = REDVAL[27:24], 111 = REDVAL[31:28]	

8 绿色查找表寄存器

GREENLUT	0x01F00018	R/W	绿色查找表寄存器	初始值 0x00000000
位名称	BIT		描述	
GREENVAL	[31:0]		这些位定义8组的每一组中的16个影射的哪一个将选择 000 = GREENVAL[3:0], 001 = GREENVAL[7:4] 010 = GREENVAL[11:8], 011 = GREENVAL[15:12] 100 = GREENVAL[19:16], 101 = GREENVAL[23:20] 110 = GREENVAL[27:24], 111 = GREENVAL[31:28]	

9 蓝色查找表寄存器

BLUELUT	0x01F0001C	R/W	蓝色查找表寄存器	初始值 0x0000
位名称	BIT		描述	

BLUEVAL [15:0] 这些位定义4组的每一组中的16个影射的哪一个将选择
 00 = BLUEVAL[3:0], 01 = BLUEVAL[7:4]
 10 = BLUEVAL[11:8], 11 = BLUEVAL[15:12]

10 抖动模式寄存器

DP1_2	0x01F00020	R/W	抖动模式占空比为1/2寄存器	初始值 0xa5a5
DP4_7	0x01F00024	R/W	抖动模式占空比为4/7寄存器	初始值 0xba5da65
DP3_5	0x01F00028	R/W	抖动模式占空比为3/5寄存器	初始值 0xa5a5f
DP2_3	0x01F0002C	R/W	抖动模式占空比为2/3寄存器	初始值 0xd6b
DP5_7	0x01F00030	R/W	抖动模式占空比为5/7寄存器	初始值 0xeb7b5ed
DP3_4	0x01F00034	R/W	抖动模式占空比为3/4寄存器	初始值 0x7dbe
DP4_5	0x01F00038	R/W	抖动模式占空比为4/5寄存器	初始值 0x7ebdf
DP6_7	0x01F0003C	R/W	抖动模式占空比为6/7寄存器	初始值 0x7dfbfe
DITHMODE	0x01F00044	R/W	抖动模式寄存器	初始值 0x00000

用户必须改变该值为0x12210.

LCD控制器的VCLK最大频率是16.5MHZ(在系统时钟为66MHZ时),

$$VCLK(Hz) = MCLK / (CLKVAL \times 2)$$

VCLK的速度应大于数据传输速率.

$$\text{数据传输速率} = HS \times VS \times FR \times MV$$

HS: LCD 水平尺寸

VS: LCD 垂直尺寸

FR: 帧速率

MV: 由显示模式确定的值

$$FR(Hz) = 1 / [((1/VCLK) \times (HOZVAL + 1) + (1/MCLK) \times (WLH + WDL + LINEBLANK)) \times (LINEVAL + 1)]$$

$$VCLK(Hz) = (HOZVAL + 1) / [(1 / (FR \times (LINEVAL + 1))) - ((WLH + WDL + LINEBLANK) / MCLK)]$$

显示模式	MV值
单色, 4位单扫描	1/4
单色, 8位单扫描或4位双扫描	1/8
4级灰度, 4位单扫描	1/4
4级灰度, 8位单扫描或4位双扫描	1/8
16级灰度, 4位单扫描	1/4
16级灰度, 8位单扫描或4位双扫描	1/8
彩色, 4位单扫描	3/4
彩色, 8位单扫描或4位双扫描	3/8

LCDBASEU寄存器的量是帧缓冲区的开始地址,在突发4字存取模式,最低4位必须取消.

LCDBASEL寄存器的量由LCD尺寸和LCDBASEU的量确定,公式为;

$$LCDBASEL = LCDBASEU + LCDBASEL \text{ offset}$$

13 A/D 转换器

S3C44B0X的1位A/D 转换器包含一个8路模拟输入混合器,自动归0比较器,时钟发生器,10位连续近似寄存器和一个输出寄存器.

特征如下:

最大转换速率:100KSPS

输入电压范围:0-2.5V

输入带宽: 0-100 Hz(无采样和保持电路)

低的电源消耗

A/D 转换器寄存器

1 A/D 转换器控制寄存器

ADCCON 0x01D40000 R/W A/D转换器控制寄存器 初始值 0x20
(Li/W, Li/HW,Li/B, Bi/W)
0x01D40002(Bi/HW)
0x01D40003(Bi/B)

位名称	BIT	描述
FLAG	[6]	A/D转换状态标志(只读) 0 = A/D conversion in process 1 = End of A/D conversion
SLEEP	[5]	系统省电模式 0 = Normal operation, 1 = Sleep mode
INPUT SELECT	[4:2]	输入源选择 000 = AIN0 001 = AIN1 010 = AIN2 011 = AIN3 100 = AIN4 101 = AIN5 110 = AIN6 111 = AIN7
READ_ START	[1]	A/D转换通过读启动 0 = Disable start by read operation 1 = Enable start by read operation
ENABLE_ START	[0]	A/D转换由允许来启动 如果 READ_ START被允许该位无效. 0 = No operation 1 = A/D conversion starts and this bit is cleared after the start-up.

注:(Li/B/HW/W):表示endian模式是小模式时,通过char/halfword/word单位存取

(Bi/B/HW/W):. 表示endian模式是大模式时,通过char/halfword/word单位存取

2 A/D 转换器预定标器寄存器

ADCPSR 0x01D40004 R/W A/D转换器预定标器寄存器 初始值 0x0
(Li/W, Li/HW,Li/B, Bi/W)
0x01D40006(Bi/HW)
0x01D40007(Bi/B)

位名称	BIT	描述
PRESCALER	[7:0]	预定标器的值 (0-255) Division factor = 2 (prescaler_value+1). Total clocks for ADC conversion = 2*(Prescaler_value+1)*16

3 A/D 转换器数据寄存器

ADCDAT 0x01D40008 R/W A/D转换器数据寄存器 初始值 -
(Li/W, Li/HW, Bi/W)
0x01D4000A(Bi/HW)

14 RTC (REAL TIME CLOCK)

RTC (Real Time Clock)单元能在系统断电时,通过备份电池来供电,RTC能通过ARM的STRB/LDRB指令传输8位BCD数据到CPU,该数据包括时、分、秒、小时、天、月和年,RTC使用一个外部32.768的晶体也能执行报警功能.

特征:

日历功能

支持闰年

解决了2000年问题

独立的供电

支持毫秒滴答时间中断作为RTOS核的时间滴答

报警功能

RTC寄存器

1 RTC控制寄存器

RTCCON	0x01D70040(little endian)				
	0x01D70043(Big endian)	R/W(字节)	RTC control Register	初始值	0x0
位名称	BIT	描述			
CLKRST	[3]	RTC 时钟计数复位 0 = No reset, 1 = Reset			
CNTSEL	[2]	BCD计数复位 0 = Merge BCD counters 1 = 保留 (Separate BCD counters)			
CLKSEL	[1]	BCD 时钟选择 0 = XTAL 1/2 ¹⁵ divided clock 1 = Reserved (XTAL clock only for test)			
RTCEN	[0]	RTC读写允许 0 = Disable, 1 = Enable			

如果RTC 读写允许, STOP电流将大大增大, 为了减少STOP 电流, 当不存取RTC时,设置该位为0,虽然为0,但RTC时钟仍运行.

2 RTC报警控制寄存器

RTCALM	0x01D70050(little endian)				
	0x01D70053(Big endian)	R/W(字节)	RTC 报警控制寄存器	初始值	0x0
位名称	BIT	描述			
Reserved	[7]				
ALMEN	[6]	Alarm全局允许 0 = Disable, 1 = Enable			
YEAREN	[5]	年alarm允许 0 = Disable, 1 = Enable			
MONREN	[4]	月alarm允许 0 = Disable, 1 = Enable			
DAYEN	[3]	天alarm允许 0 = Disable, 1 = Enable			
HOUREN	[2]	小时alarm允许 0 = Disable, 1 = Enable			
MINEN	[1]	分alarm允许 0 = Disable, 1 = Enable			
SECEN	[0]	秒alarm允许 0 = Disable, 1 = Enable			

3 报警秒数据寄存器

ALMSEC	0x01D70054(little endian)				
	0x01D70057(Big endian)	R/W(字节)	秒数据寄存器	初始值	0x00
位名称	BIT	描述			
Reserved	[7]	保留			
SECDATA	[6:4]	秒的BCD值 (0—5)			
	[3:0]	0—9			

4 报警分钟数据寄存器

ALMMIN	0x01D70058(little endian)				
	0x01D7005B(Big endian)	R/W(字节)	分钟数据寄存器	初始值	0x00
位名称	BIT	描述			
Reserved	[7]	保留			
MINDATA	[6:4]	分钟的BCD值 (0—5)			
	[3:0]	0—9			

4 报警小时数据寄存器

ALMHOUR	0x01D7005C(little endian)				
	0x01D7005F(Big endian)	R/W(字节)	小时数据寄存器	初始值	0x00
位名称	BIT	描述			
Reserved	[7:6]	保留			
HOURLDATA	[5:4]	小时的BCD值 (0—2)			
	[3:0]	0—9			

5 报警日数据寄存器

ALMDAY	0x01D70060(little endian)				
	0x01D70063(Big endian)	R/W(字节)	日数据寄存器	初始值	0x01
位名称	BIT	描述			
Reserved	[7:6]	保留			
DAYDATA	[5:4]	日的BCD值 (0—3)			
	[3:0]	0—9			

6 报警月数据寄存器

ALMMON	0x01D70064(little endian)				
	0x01D70067(Big endian)	R/W(字节)	月数据寄存器	初始值	0x01
位名称	BIT	描述			
Reserved	[7:5]	保留			
MONDATA	[4]	月的BCD值 (0—1)			
	[3:0]	0—9			

7 报警年数据寄存器

ALMYEAR	0x01D70068(little endian)				
	0x01D7006B(Big endian)	R/W(字节)	年数据寄存器	初始值	0x00
位名称	BIT	描述			
YEARDATA	[7:0]	年的BCD值 (00—99)			

8 RTC ROUND复位寄存器

RTCRST	0x01D7006C(little endian)				
	0x01D7006F(Big endian)	R/W(字节)	RTC ROUND复位寄存器	初始值	0x00
位名称	BIT	描述			
SRSTEN	[3]	Round 秒复位允许			

		0 = Disable, 1 = Enable
SECCR	[2:0]	产生秒进位的Round边界
		011 = over than 30 sec
		100 = over than 40 sec
		101 = over than 50 sec

9 BCD秒数据寄存器

BCDSEC	0x01D70070(little endian)				
	0x01D70073(Big endian)	R/W(字节)	BCD秒数据寄存器	初始值	Undef.

位名称	BIT	描述
Reserved	[7]	保留
SECDATA	[6:4]	秒的BCD值 (0—5)
	[3:0]	0—9

10 BCD分钟数据寄存器

BCDMIN	0x01D70074(little endian)				
	0x01D70077(Big endian)	R/W(字节)	BCD分钟数据寄存器	初始值	Undef.

位名称	BIT	描述
Reserved	[7]	保留
MINDATA	[6:4]	分钟的BCD值 (0—5)
	[3:0]	0—9

11 BCD小时数据寄存器

BCDHOURL	0x01D70078(little endian)				
	0x01D7007B(Big endian)	R/W(字节)	BCD小时数据寄存器	初始值	Undef.

位名称	BIT	描述
Reserved	[7:6]	保留
HOURLDATA	[5:4]	小时的BCD值 (0—2)
	[3:0]	0—9

12 BCD日数据寄存器

BCDDAY	0x01D7007C(little endian)				
	0x01D7007F(Big endian)	R/W(字节)	BCD日数据寄存器	初始值	Undef.

位名称	BIT	描述
Reserved	[7:6]	保留
DAYDATA	[5:4]	日的BCD值 (0—3)
	[3:0]	0—9

13 BCD星期数据寄存器

BCDDATE	0x01D70080(little endian)				
	0x01D70083(Big endian)	R/W(字节)	BCD星期数据寄存器	初始值	Undef.

位名称	BIT	描述
Reserved	[7:3]	保留
DATEDATA	[2:0]	日的BCD值 (1—7)

14 BCD月数据寄存器

BCDMON	0x01D70084(little endian)				
	0x01D70087(Big endian)	R/W(字节)	BCD月数据寄存器	初始值	Undef.

位名称	BIT	描述
Reserved	[7:5]	保留

MONDATA [4] 月的BCD值 (0—1)
[3:0] 0—9

15 BCD年数据寄存器

BCDYEAR 0x01D70088(little endian)
0x01D7008B(Big endian) R/W(字节) BCD年数据寄存器 初始值 Undef.

位名称	BIT	描述
YEARDATA	[7:0]	年的BCD值 (00—99)

16 时间滴答计数寄存器

TICNT 0x01D7008C(little endian)
0x01D7008F(Big endian) R/W(字节) 时间滴答计数寄存器 初始值 0x00

位名称	BIT	描述
TICK INT ENABLE	[7]	时间滴答中断允许 0 = disable 1 = enable
TICK TIME COUNT	[6:0]	时间滴答计数值. (1-127) 这个计数值内部递减. 用户不能读它的实时值

14WATCHDOG TIMER

S3C44B0X的看门狗定时器用来在由于错误如干扰和系统错误造成的程序运行打乱时恢复正常操作,它也能使用一个正常的16位定时器来请求中断服务. 看门狗定时器产生复位信号128个系统时钟周期.

看门狗定时器时钟频率:

$$t_{watchdog} = 1 / (MCLK / (Prescaler\ value + 1) / Division_factor)$$

当S3C44B0X使用Embedded ICE工作在调试模式时,看门狗定时器必须不工作.看门狗定时器能通过CPU核心信号(DBGACK信号)确定当前的模式是否在调试模式,一旦DBGACK信号有效,看门狗定时器的复位输出当定时器过期时不激活.

看门狗定时器寄存器

看门狗定时器控制寄存器

WTCON 0x01D30000 R/W 看门狗定时器控制寄存器 初始值 0x8021

位名称	BIT	描述
Prescaler value	[15:8]	预分频值 (0 to $2^8 - 1$)
Reserved	[7:6]	保留.0
watchdog timer enable/disable	[5]	看门狗定时器的允许位. 0 = Disable watchdog timer 1 = Enable watchdog timer
Clock select	[4:3]	这两位确定时钟除因子 00: 1/16 01: 1/32 10: 1/64 11: 1/128
Interrupt enable/disable	[2]	看门狗中断允许位 0 = Disable interrupt generation 1 = Enable interrupt generation
Reserved	[1]	保留
Reset enable/disable	[0]	看门狗输出复位信号的允许位

1:允许
0: 不允许

看门狗定时器数据寄存器

WTDAT 0x01D30004 R/W 看门狗定时器数据寄存器 初始值 0x8000

WTDAT规定看门狗定时器超时周期。WTDAT的内容在初始操作时，不能自动加载进定时器计数寄存器。可是定时器计数寄存器在使用初始值0X8000第一次超时出现以后，WTDAT的值将自动加载进WTCNT。

看门狗定时器计数寄存器

WTCNT 0x01D30008 R/W 看门狗定时器当前计数寄存器 初始值 0x8000

在第一次使用时，必须设置到初始值。

15 IIC-BUS 接口

S3C44B0X支持一个多主IIC-BUS串行接口，主S3C44B0X能发送或接收串行数据到从设备，并遵守标准的IIC协议。

有如下操作模式：

主发送模式

主接收模式

从发送模式

从接收模式

IIC-BUS接口寄存器

1 IIC-BUS控制寄存器

IICCON 0x01D60000 R/W IIC-BUS控制寄存器 初始值 0000_XXXX

位名称	BIT	描述
Acknowledge		
enable	[7]	IIC-bus应答允许位 1=Enable ACK generation 在发送模式，IICSDA在ACK时间释放。 在接收模式，IICSDA在ACK时间为低
Tx clock source		
Selection	[6]	IIC-bus的源时钟的预分频值 0 = IICCLK = fMCLK /16 1 = IICCLK = fMCLK /512
Tx/Rx Interrupt		
Enable	[5]	IIC-BusTx/Rx中断允许位 0 = Disable interrupt 1 = Enable interrupt
Interrupt pending		
flag	[4]	IIC-busTx/Rx中断挂起标志。 写1是不可能的，当读该位为1时，中断挂起 IICSCS 为低，IIC停止，为了恢复操作，清除该位 0 = 1) 读时，没有中断 2)写入0，清除挂起条件和恢复操作 1 = 1) 读时，中断挂起 2)写时，无操作 N/A

Transmit clock

value [3:0] IIC-Bus发送预定标值

$$\text{Tx clock} = \text{IICCLK}/(\text{IICCON}[3:0]+1)$$

注：1 与EEPROM接口时，在接收模式为了产生停止条件，在读最后的字节之前，ACK产生可以无效。

2 IIC_BUS中断出现情况 1) 当发送或接收一个字节完成时

2) 当从地址匹配时 3)总线仲裁失败时

3 为了满足IISDCL上升沿之前IICSDA的启动时间要求,IICDS必须在清除IIC中断挂起位之前写入

4 Interrupt pending flag 在Tx/Rx Interrupt Enable=0时,不能正常操作,因此即使不使用中断, Tx/Rx Interrupt Enable也推荐为1.

2 IIC-BUS控制/状态寄存器

IICSTAT 0x01D60004 R/W IIC-BUS控制/状态寄存器 初始值 0000_000

位名称 BIT 描述

Mode selection [7:6] IIC-bus master/slave Tx/Rx 模式选择位:

00: Slave receive mode

01: Slave transmit mode

10: Master receive mode

11: Master transmit mode

Busy signal status/

START STOP

Condition [5] IIC-Bus忙信号状态位

0 = 读时) IIC-bus不忙

写时) IIC-bus STOP信号产生

1 =读时) IIC-bus忙

写时) IIC-bus START信号产生.

IICDS 上的数据自动传输在START信号后

Serial output enable [4] IIC-bus 数据输出允许位

0=Disable Rx/Tx, 1=Enable Rx/Tx

Arbitration

status flag [3] IIC-bus仲裁过程状态标志位

0 = Bus arbitration successful

1 = Bus arbitration failed during serial I/O

Address-as-slave

status flag [2] IIC-bus从地址状态标志位

0 =当检测到START/STOP清除

1 = 接收到的slave地址匹配IICADD的值

Address zero

status flag [1] IIC-bus 地址为0状态标志:

0 =当检测到START/STOP清除

1 = 接收到的slave地址是 00000000b

Last-received bit

status flag [0] IIC-bus上一次接收到的状态标志位

0 = Last-received bit is 0 (ACK was received)

1 = Last-receive bit is 1 (ACK was not received)

3 IIC-BUS地址寄存器

IICADD	0x01D60008	R/W	IIC-BUS地址寄存器	初始值 XXXX_XXXX
位名称	BIT		描述	
Slave address	[7:0]		当IICADD为写允许时，从IIC-bus锁存的7位从地址。 IICADD的值可以在任何时候被读。 从地址= [7:1] ， Not mapped = [0]	

4 IIC-BUS发送/接受数据移位寄存器

IICDS	0x01D6000C	R/W	IIC-BUS发送/接受数据移位寄存器	初始值 XXXX_XXXX
-------	------------	-----	---------------------	---------------

当IICSTATserial output enable = 1, IICDS为写允许。IICDS的值可以在任何时候被读。

17 IIS-BUS 接口

S3C44B0X IIS (Inter-IC Sound) 接口能用来连接一个外部8/16位立体声声音CODEC。

IIS总线接口对FIFO存取提供DMA传输模式代替中断模式，它可以同时发送数据和接收数据也可以只发或只收。

特征：支持IIS格式与 MSB-justified格式

每个通道16，32，48fs 的串行位时钟（fs为采样频率）

每个通道可以8位或16位数据格式。

256，384fs主时钟

对主时钟和外部CODEC时钟的可编程的频率分频器

32字节（2*16）的发送和接收FIFO (每个FIFO组织为8*半字)。

正常和DMA传输模式

正常传输模式：IIS控制寄存器有一个FIFO准备好标志位，当发送数据时，如果发送FIFO不空，该标志为1，FIFO准备好发送数据，如果送FIFO为空，该标志为0。当接收数据时，如果接收FIFO不满，该标志设置为1，指示可以接收数据，若FIFO满，则该标志为0。通过该标志位，可以确定CPU读写FIFO的时间，通过该方式实现发送和接收FIFO的存取来发送和接收数据。

DMA传输方式：发送和接收FIFO的存取有DMA控制器来实现，由FIFO准备好标志来自动请求DMA的服务。

发送和接收同时模式：因为只有一个DMA源，因此在该模式，只能是一个通道（如发送通道）用正常传输模式，另一个通道（接收通道）用DMA传输模式，反之则反之，从而实现同时工作目的。

IIS-BUS 格式

IIS有四条线，串行数据输入（IISDI），串行数据输出IISDO),左/右通道选择（IISLRCK），和串行位时钟clock(IISCLK)；产生IISLRCK和IISCLK信号的为主设备。

串行数据以2的补数发送，首先发送高位。高位首先发送是因为发送方和接收方可以有不同的字长度。发送方知道接收方能处理的位数是不必要的，同样接收方也不需要知道发送方正发送多少位的数据。

当系统字长度大于发送放的字长度时，字被切断（最低数据位设置为0）来发送。如果接收方发送比它的字长度更多的位时，多的位被忽略，若接收方发送比它的字长度少的位时，不足的位被内部设置为0。所以高位有固定的位置，而低位的位置依赖于字长度。发送器总是在IISLRCK变化的下一个时钟周期发送下一个字的高位。

发送器的串行数据发送可以在时钟信号的上升沿或下降沿被同步。可是串行数据必须在串行时钟信号的上升沿锁存进接收器，所以当用发送数据用上升沿来同步时有一些限制。

LR通道选择线指示当前正发送的通道。IISLRCK即可以在串行时钟的上升沿变化，也可以在下沿变化，但不需要同步，在从模式这个信号在串行时钟的上升沿锁存。IISLRCK在高位发送之

前变化一个时钟周期，这允许从发送方可以同步发送串行数据，更进一步，它允许接收方存储先前的字和清除输入来接收下一个字。

MSB JUSTIFIED 格式

MSB JUSTIFIED 格式与IIS不同的地方是它总是当IISLRCK变化时发送下一个字的高位。

Serial bit per channel	8-bit	16-bit
Serial clock frequency (IISCLK)		
@CODECLK=256fs	16fs, 32fs	32fs
@CODECLK=384fs	16fs, 32fs, 48fs	32fs, 48fs

IIS-BUS 接口 寄存器

1 IIS 控制寄存器

IISCON	0x01D18000(Li/HW, Li/W, Bi/W)				
	0x01D18002(Bi/HW)	R/W	IIS控制寄存器	初始值	0x100

位名称	BIT	描述
Left/Right channel index (read only)	[8]	0 = 左通道 1 =右通道
Transmit FIFO ready flag (read only)	[7]	0 = 发送FIFO没有准备好(空) 1 =发送FIFO准备好(不空)
Receive FIFO ready flag (read only)	[6]	0 =接收FIFO没有准备好(满) 1 =接收FIFO准备好(不满)
Transmit DMA service request enable	[5]	0 = 发送DMA请求不允许 1 =发送DMA请求允许
Receive DMA service request enable	[4]	0 = 接收DMA请求不允许 1 =接收DMA请求允许
Transmit channel idle Command	[3]	在发送空闲状态，IISLRCK不激活(暂停发送). 该位仅在IIS是一个master时有效 0 = IISLRCK is generated. 1 = IISLRCK is not generated.
Receive channel idle Command	[2]	在接收空闲状态，IISLRCK不激活(暂停接收) I该位仅在IIS是一个master时有效 0 = IISLRCK is generated. 1 = IISLRCK is not generated.

IIS prescaler enable [1] 0 = 预定标器不起作用
1 = 允许预定标器

IIS interface enable
(start) [0] 0 = IIS不允许 (停止)
1 = IIS 允许 (启动)

2 IIS 模式寄存器

IISMOD 0x01D18004(Li/HW, Li/W, Bi/W)
0x01D18006(Bi/HW) R/W IIS 模式寄存器 初始值 0x0

位名称	BIT	描述
Master/slave mode select	[8]	0 = Master模式 (IISLRCK和IISCLK输出) 1 = Slave模式 (IISLRCK和IISCLK输入)

Transmit/receive mode Select	[7:6]	描述
		00 = 不传输 01 =接收模式 10 = 发送模式 11 =发送/接收模式

Active level of left/right Channel	[5]	描述
		0 =低为左通道(高为右通道) 1 =高为左通道(低为右通道)

Serial interface format	[4]	描述
		0 = IIS格式 1 = MSB(Left)-justified 格式

Serial data bit per channel	[3]	描述
		0 = 8-bit 1 = 16-bit

Master clock(CODECLK) frequency select (fs :采样频率)	[2]	描述
		0 = 256fs 1 = 384fs

Serial bit clock frequency Select	[1:0]	描述
		00 = 16fs 01 = 32fs 10 = 48fs 11 = N/A

3 IIS 预定标器寄存器

IISPSR 0x01D18008(Li/B, Li/HW, Li/W Bi/W)
0x01D1800A(Bi/HW) R/W IIS 预定标器寄存器 初始值 x0
0x01D1800B(Bi/B)

位名称	BIT	描述
Prescaler value A	[7:4]	预定标器A的定标因子 $clock_prescaler_A = MCLK / <division factor>$
Prescaler value B	[3:0]	预定标器B的定标因子 $clock_prescaler_B = MCLK / <division factor>$

IISPSR[3:0] / [7:4]	定标因子	IISPSR[3:0] / [7:4]	定标因子
0000b	2	1000b	1
0001b	4	1001b	—
0010b	6	1010b	3*

0011b	8	1011b	—
0100b	10	1100b	5*
0101b	12	1101b	—
0110b	14	1110b	7*
0111b	16	1111b	-

注：1 如果定标因子为3,5,7,Z占空比不是50%,高电平周期为0.5MCLK

4 IIS FIFO控制寄存器

为了启动IIS操作,执行下列过程

- 1) 允许IISFCON寄存器的FIFO
- 2) 允许IISFCON寄存器的DMA请求
- 3) 允许IISFCON寄存器的启动

为了结束IIS操作, 执行如下过程

- 1) 不允许IISFCON寄存器的FIFO,如果你还想发送FIFO的剩余数据,跳过这一步.
- 2) 不允许IISFCON寄存器的DMA请求
- 3) 不允许IISFCON寄存器的启动

IISFCON	0x01D1800C(Li/HW, Li/W Bi/W)				
	0x01D1800E(Bi/HW)	R/W	IISFIFO控制寄存器	初始值	x0
位名称	BIT	描述			
发送FIFO存取模式选择	[11]	0 = 正常存取模式 1 = DMA 存取模式			
接收FIFO存取模式选择	[10]	0 = Normal access mode 1 = DMA access mode			
发送FFO允许位	[9]	0 = FIFO disable 1 = FIFO enable			
接收FIFO允许位	[8]	0 = FIFO disable 1 = FIFO enable			
发送FIFO数据计数值 (只读)	[7:4]	Data count value = 0-8			
接收FIFO数据计数值 (只读)	[3:0]	Data count value = 0-8			

5 IIS FIFO寄存器 发送和接收FIFO的存取入口

IISFIF	0x01D18010(Li/HW)				
	0x01D18012(Bi/HW)	R/W	IIS FIFO寄存器	初始值	0x0

18 SIO (同步 I/O)

S3C44B0X的SIO能与各种类型的串性外设接口.这个SIO模块能以一定的频率(由寄存器设置)发送或接收8位串行数据.时钟源可以选择内部时钟或外部时钟.

SIO模块具有以下功能

- 8位数据缓冲
- 12位的预定标器
- 8位间隔计数器
- 时钟选择逻辑
- 串行数据I/O脚((SIORXD 和SIOTXD)
- 外部时钟输入/输出脚(SIOCK)
- DMA 运行模式

SIO正常操作模式

发送与接收同时进行,一个发送数据脚,一个接收数据脚,当一个字节写入SIODAT数据寄存器,如果SIO运行位设置和发送模式允许,则SIO开始发送数据

为了对SIO模块编程,应该遵守如下步骤:

- 1) 配置I/O脚(SIOTXD, SIOCLK, SIORXD).
- 2) 设置SIOCON为适当的配置
- 3) 设置串行I/O中断允许位
- 4) 如果想发送数据,写数据到SIODAT.
- 5) 设置SIOCON[3]为1,开始数据移位操作
- 6) 当数据移位操作完成时,SIO中断被请求和SIODAT接收到数据
- 7) 返回第4步

SIO DMA 操作

自动运行模式(非握手模式)

在该模式,SIO等待直到发送的数据被外部目标设备读走.在每次8位数据发送后,SIO插入一个可编程的间隔周期.

DMA发送数据步骤如下:

- 1) 清DCNTZ为0,使SIO能请求DMA服务.除了SIOCON[1:0]必须为00外,适当的配置SIO
- 2) 适当的配置DMA
- 3) SIO被配置为DMA发送模式.
- 4) SIO自动请求DMA服务
- 5) SIO发送数据
- 6) 返回步骤4直到DMA计数为0
- 7) 设置DCNTZ为1,停止SIO请求进一步的DMA服务.

DMA接收数据步骤如下:

- 1) 清DCNTZ为0,使SIO能请求DMA服务.除了SIOCON[1:0]必须为00外,适当的配置SIO
- 2) 适当的配置DMA
- 3) SIO被配置为DMA只接模式.
- 4) 设置SIOCON[3] (SIO 开始位)来开始接收操作
- 5) SIO在接收到8位数据后请求DMA服务
- 6) 返回步骤5直到DMA计数为0

7) 设置DCNTZ为1,停止SIO请求进一步的DMA服务.

SIO 接口寄存器

1 SIO 控制寄存器 (SIOCON)

SIOCON 0x01D14000 R/W 控制寄存器 初始值 0x00

位名称	BIT	描述
Clock source select	[7]	SIO2 移位时钟源选择位 0 = Internal clock, 1 = External clock
Data direction	[6]	该位控制MSB还是LSB首先发送 0 = MSB mode, 1 = LSB mode
Tx/Rx selection	[5]	该位决定是否发送操作被允许,如果你只想发送,在SIODAT接收到的数据将被忽略. 如果你想发送和接收,SIO支持数据发送和接收同时进行.用户写发送数据到SIODAT,然后SIO串行发送数据,同时,SIO接收数据,在SIO发送完后, SIODAT的内容就是接收到的数据. 0 = Receive only mode, 1 = Transmit/Receive mode
Clock edge select	[4]	这位确定用做串行发送或接收的时钟T 0 = falling edge clock, 1 = rising edge clock
SIO start	[3]	这位确定SIO是否运行或停止 当 BDMA Tx被用,该位应当为 '0'. 0 = No action 1 = Clear 3-bit counter and start shift. 这位用写入1来清除
Shift operation	[2]	确定SIO 的移位操作模式 0 = 非握手模式 (自动运行模式) 1 = 保留
SIO mode select	[1:0]	确定SIO的操作,即SIODATA 如何读/写 00 = no operations 01 = SIO interrupt mode 10 = BDMA0 mode 11 = BDMA1 mode

2 SIO 数据寄存器 (SIOCON)

SIODAT 0x01D14004 R/W SIO数据寄存器 初始值 0x00

3 SIO 波特率预定标器寄存器 (SBRDR)

Baud rate = $MCLK / 2 / (\text{Prescaler value} + 1)$

SBRDR 0x01D14008 R/W SIO波特率预定标器寄存器 初始值0x00

4 SIO 间隔计数寄存器(IVTCNT)

间隔 Intervals (between 8-bit data) = $MCLK / 4 / (IVTCNT + 1)$

IVTCNT 0x01D1400C R/W SIO间隔计数寄存器 初始值 0x00

5 SIO 间隔计数寄存器(IVTCNT)

当SIO工作在DMA模式时,对应的DCNTZ必须为0,当DMA完成时,对应的DCNTZ设置为1

DCNTZ 0x01D14010 R/W SIO dma计数0寄存器 初始值 0x00

位名称	BIT	描述
-----	-----	----

DCNTZ1	[1]	
--------	-----	--

0: 允许BDMA1服务请求.当该位是0时 ,SIO可以请求DMA服务

DCNTZ0

[0]

1: 不允许BDMA1服务请求

0: 允许BDMA0服务请求.当该位是0时 ,SIO可以请求DMA服务

1: 不允许BDMA0服务请求