## UART协议

### 1 介绍

#### 1.1 概述

UART协议基于行业标准TL16C550异步通信协议，TL16C450的功能升级版本。UART可以放置在备用FIFO (TL16C550)模式。通过缓冲接收和传输的字符来减轻CPU过多的开销。接收器和发送器FIFO存储多达16个字节，包括每个字节的错误状态位。

UART将从外设接收到的数据由串行转换到并行，而将从CPU接收到的数据从并行转换到串行。CPU可以随时读取UART状态。UART具有自控系统和中断系统，以最大限度地减少通信链路的软件管理。

UART包括一个可编程时钟发生器，除频系数在 1到65535之间，并为内部发送器和接收器产生一个参考时钟。

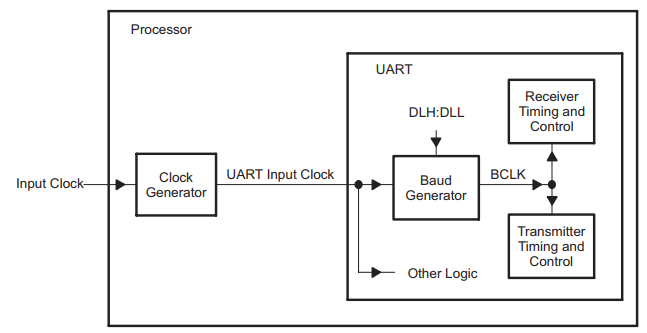
#### 1.2 UART协议特性

根据不同设备的数据手册查看UART支持和不支持的特性列表。

### 2 协议结构

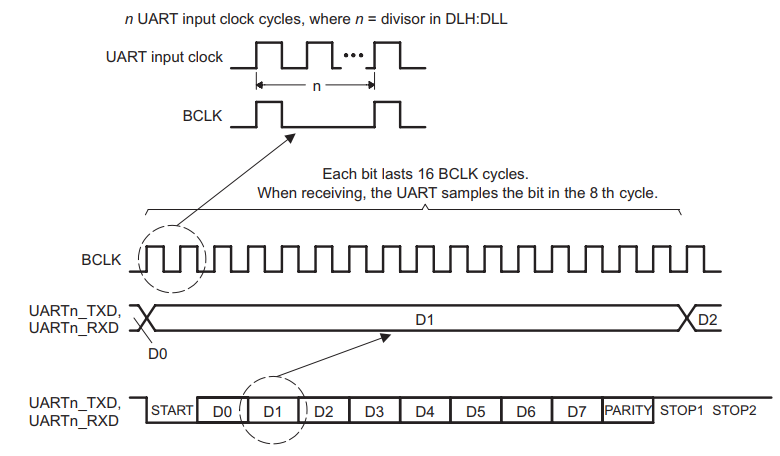
#### 2.1 时钟生成

UART传输数据时钟根据UART的输入时钟生成。



如图所示，处理器的时钟发生器接收来自外部时钟源的信号并产生具有相应频率的UART输入时钟。UART包含一个可编程波特发生器，UART输入时钟并将其除以1到65535范围内的除数从而产生波特时钟(BCLK)。BCLK频率为16倍(16×)波特率(每接收或发送1位持续16个BCLK周期)或13倍(13×)波特率(每接收或发送1位持续13个BCLK周期)。当UART接收时，比特在第8个或第6个BCLK周期采样。

16×或13×参考时钟是通过配置MDR中OSM\_SEL位来配置的。除频系数由两个8位寄存器字段(DLH和DLL)组成。DLH保存高8位，而DLL保存低8位。这些除数锁存器必须在初始化UART期间加载，以确保波特发生器按照预期进行操作。写入该寄存器会导致在访问期间插入两个等待状态，同时波特率生成器加载新值。



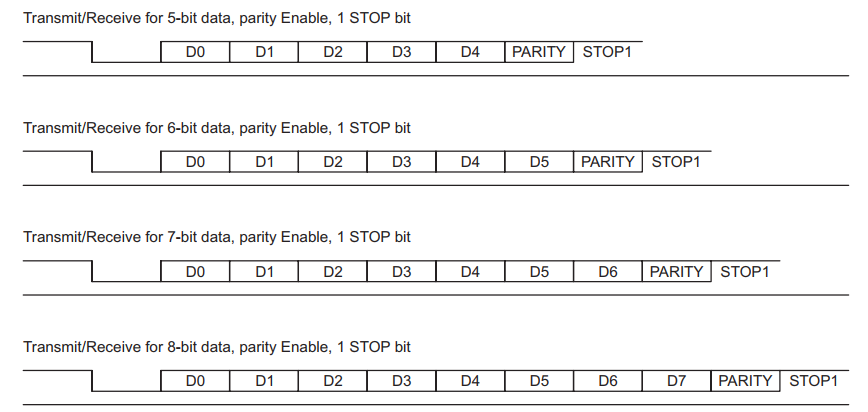
#### 2.2 信号描述

UART使用最少数量的信号连接来连接外部设备。

* TXD：串行数据输出；
* RXD：串行数据输入；
* CTS：可选，发送完毕握手信号；
* RTS：可选，发送请求握手信号；

#### 2.3 协议描述

* 发送：UART发送器包括THR寄存器和移位寄存器(TSR)。当UART为FIFO模式时，THR为16字节的FIFO。传输控制是LCR寄存器所提供的功能。基于LCR中的设置，UART传输1bit起始位、5到8位数据、1bit奇偶校验位(可选)和1到2bit停止位；
* 接收：UART接收器包括移位寄存器(RSR)和缓冲寄存器(RBR)。当UART为FIFO模式时，RBR为16字节的FIFO。由(LCR)寄存进行控制。根据LCR中的设置，UART接收1bit起始位、5到8位数据、1bit奇偶校验位(可选)和1bit停止位；



#### 2.4 操作

##### 2.4.1 发送数据

UART发射器包括THR寄存器和移位寄存器(TSR)。当UART为FIFO模式时，THR为16字节的FIFO。

THR从内部数据总线接收数据；当TSR准备好时，UART将数据从THR移动到TSR。UART将TSR中的数据进行移位并在TXD引脚上传输数据。在非FIFO模式下，如果THR为空并且在中断使能寄存器(IER)中启用了THR空中断，则会产生中断。当数据加载到THR中时，该中断将被清除。在FIFO模式下，当发送FIFO为空时产生中断，否则中断被清除。

##### 2.4.2 接收数据

UART接收器包括移位寄存器(RSR)和缓冲寄存器(RBR)。当UART为FIFO模式时，RBR为16字节的FIFO。

RSR从RXD引脚接收数据。然后将RSR中的数据移动到RBR(接收FIFO)。UART为每个接收到的数据存储三位错误状态位，以记录奇偶校验错误、帧错误或停止。

在非FIFO模式下，当数据被放置在RBR中并且在中断使能寄存器(IER)中启用了接收数据完成中断时，将生成一个中断。当从RBR中读取数据时，该中断将被清除。在FIFO模式下，如果接收数据数目大于FCR中选择的阈值时产生中断，否则清除中断。

##### 2.4.3 FIFO模式

FIFO可以通过中断或轮询方式获取数据。当发生特定事件时FIFO中断被发送到CPU进行相关处理。当中断被禁用时，FIFO使用轮询模式。CPU轮询状态以检测特定事件。由于接收FIFO和发送FIFO是分开控制的，因此可以将其中一个或两者置于中断模式或轮询模式。

1. **中断模式**

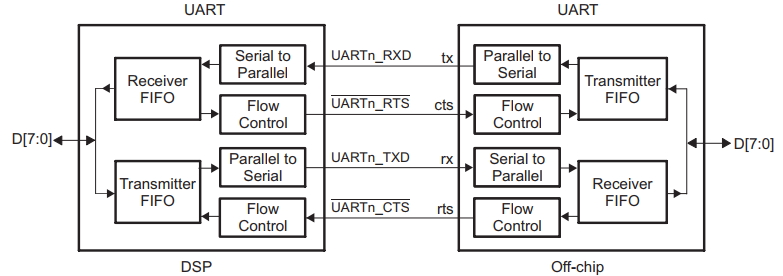
* 当FIFO数据达到FCR设置的阈值时，接收数据完成中断被发送到CPU。当CPU或DMA控制器从FIFO读取足够多的数据时，使得FIFO下降到触发阈值以下时将被清除；
* 接收状态中断是在响应溢出错误、奇偶校验错误、帧错误或停止时产生。此中断比接收数据完成中断具有更高的优先级；
* 状态寄存器(LSR)中的数据完成(DR)位表示接收FIFO中数据是否存在。DR位是当一个字符从接收移位寄存器(RSR)传输到接收FIFO时设置的。直到FIFO再次为空DR位才会被清除；
* 字符时间通过波特率计算；
* 当接收超时中断发生时，当CPU或DMA控制器从接收FIFO读取数据时将被清除，超时定时器被清除。如果在FIFO中接收到新数据或者PWREMU\_MGMT中的URRST位被清除，中断也会被清除；
* 如果接收超时中断没有发生，则在接收到新数据或CPU或DMA读取接收FIFO后超时计时器将被清除；

1. **轮询模式**

* RXFIFOE位表示接收FIFO中是否有错误；
* TEMT位表示THR和TSR都为空；
* THRE位表示THR为空；
* BI(停止)、FE(帧错误)、PE(奇偶校验错误)和OE(溢出错误)位指定发生了哪些错误；
* DR(数据完成)位设表示在接收FIFO中具有数据；

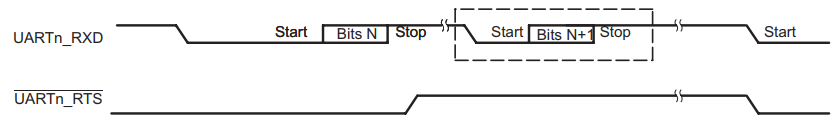
##### 2.4.4. 流量控制

UART可以通过连接CTS和RTS信号来实现流量控制。在发送FIFO可以传输数据之前，CTS必须输入有效电平。当接收需要更多数据并通知发送设备时，RTS变为有效状态。当RTS连接到CTS时，除非接收方FIFO有数据空间，否则数据传输不会发生。如图所示，当启用流量控制时，两个UART连接在一起可以消除溢出错误。



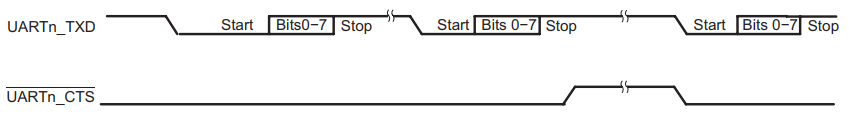
1. **RST功能**

RTS数据流控制起源于接收器。当接收FIFO级别达到触发阈值1、4、8或14时， RTS会被清除。在达到触发阈值之后，发送UART可能会发送一个额外的数据，因为可能直到开始发送额外的数据之后才会检测到RTS被清除。对于触发阈值1、4和8，在接收FIFO为空后自动重新设置RTS信号。对于触发阈值14，当接收方FIFO下降到触发阈值以下时， RTS将自动重新设置为有效信号。



1. **CTS功能**

发送器在发送下一个数据之前检查CTS信号。如果CTS是有效的，发送方发送下一个数据。为了停止发送方发送下一个字节， CTS必须在当前正在发送的最后一个STOP位中间之前释放。当流量控制被启用时， CTS的变化不会触发中断，因为设备自动控制自己的发送器。如果没有使能流量控制，发送器发送FIFO中存在的任何数据，可能会导致接收方发生数据溢出错误。



1. **环回控制**

可以使用MCR中的LOOP位将UART置于诊断模式，MCR在内部将UART输出连接回UART输入。该模式下，可以在不连接到另一个UART的情况下验证发送和接收数据路径、发送器和接收器中断以及控制中断。

#### 2.5 复位设置

* 软复位： PWREMU\_MGMT中UTRST和URRST控制着接收器和发送器的复位；在每种情况下，将接收器或发射器置于复位状态将重置受影响部分的状态机，但不影响UART寄存器；
* 当处理器RESET引脚被设置时，整个处理器将被重置并保持在重置状态，直到RESET信号被释放；作为设备重置的一部分，UART状态机将被重置，UART寄存器将被强制恢复到默认状态；

#### 2.6 初始化

1. 设置设备引脚多路复用；
2. 通过将适当的时钟除数值写入DLL和DLH来设置所需的波特率；
3. 如果将使用FIFO，请选择所需的触发阈值并通过将适当的值写入FIFO控制寄存器(FCR)来启用FIFO。在配置FCR中的其他位之前，必须先设置FCR中的FIFOEN位；
4. 通过将适当的值写入LCR来选择所需的协议设置；
5. 如果需要流量控制，将适当的值写入MCR；
6. 通过配置FREE位选择挂起事件的响应并通过设置PWREMU\_MGMT中的UTRST和URRST位启用UART；

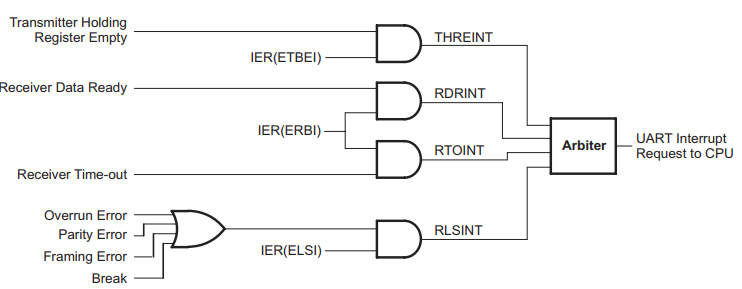
#### 2.7 中断支持

##### 2.7.1 中断请求

UART生成的所有中断请求都通过仲裁器多路复用到CPU的单个UART中断请求。每个中断请求在中断使能寄存器(IER)中都有一个使能位，并记录在中断识别寄存器(IIR)中。如果发生中断，并且使能位设置为1，则中断请求记录在IIR中并转发给CPU。如果中断发生，对应的使能位被清除为0，中断请求将被屏蔽。中断请求既不记录在IIR中，也不转发给CPU。

**UART中断源**：

* THREIN：THR或发送FIFO为空；所有数据已从THR复制到移位寄存器；
* DAINT：在非FIFO模式下接收可用数据或在FIFO模式下达到触发阈值；
* RTOINT：接收器超时(仅在FIFO模式下)；在最后四个字符时间内，没有数据输入到接收FIFO并且在此期间接收FIFO中至少有一个字节数据；
* RLSINT：接收状态异常：溢出错误、奇偶校验错误、帧错误或停止发生；



##### 2.7.2 中断多路复用

UART有专用的中断信号给CPU并且中断不与任何其他中断源多路复用。

#### 2.8 DMA事件支持

* 接收事件(URXEVT)：接收FIFO的触发阈值(1,4,8或14个字节)由FIFO控制寄存器(FCR)中的RXFIFTL位设置。每次达到触发阈值或发生接收超时，UART都会向DMA控制器发送一个接收事件。作为响应，EDMA控制器通过接收缓冲区寄存器(RBR)从接收FIFO读取数据；
* 传输事件(UTXEVT):当发送FIFO为空时(当发送FIFO中的最后一个数据已复制到发送移位寄存器)，UART发送UTXEVT信号到EDMA控制器。作为响应，EDMA控制器通过THR重新填充发送FIFO。当PWREMU\_MGMT中的UTRST位设置为1时，UTXEVT信号也被发送到DMA控制器；

DMA通道中的活动可以与这些事件同步。在非FIFO模式下，UART不产生DMA事件。在生成UART事件时，必须启用与这些事件同步的DMA通道，否则DMA通道将错过该事件，除非UART生成一个新事件，否则不会发生数据传输。

#### 2.9 电源管理

UART外设可以设置为低功耗模式，以在低活动期间节省功耗。UART外设的电源管理由PSC控制。PSC作为设备上所有外设的电源管理的主控制器。

#### 2.10 仿真

PWREMU\_MGMT中的FREE位决定UART如何响应挂起事件。如果FREE为0并且正在进行传输，则UART在完成一个字节传输后停止；如果FREE为0并且传输没有进行，则UART立即停止。如果FREE为1，则UART不停止，继续正常运行。

还要注意，仿真器访问对UART操作本质上是透明的。仿真器读取操作不会影响任何寄存器内容、状态位或操作状态。然而，仿真器写入操作可能会影响寄存器内容和UART操作，这取决于访问的寄存器和写入的值。

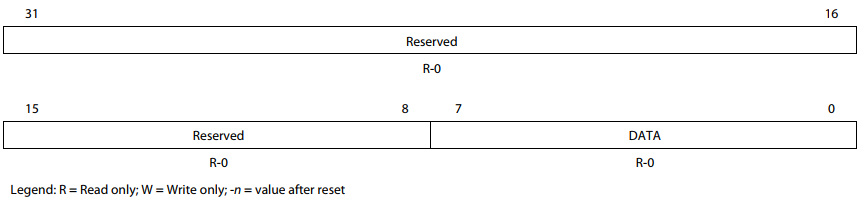
在仿真挂起事件期间，即使UART活动已经停止，也可以读取或写入UART寄存器。

#### 2.11 异常处理

* 除数未编程：由于处理器复位信号对除数寄存器没有影响，上电后除数寄存器将有一个未知值。如果上电后没有对除数锁存器进行编程，则波特时钟(BCLK)将不工作，而是将其设置为高电平。除数寄存器应该总是在处理器重置之后重新初始化；
* 在串行通信繁忙时改变工作模式：由于串行链路的特性是基于控制寄存器的编程方式，所以当UART忙于串行通信时，期望控制寄存器是不变的。因此，当模块仍忙于与另一个串行设备通信时，更改控制寄存器很可能会导致错误，应该避免；

### 3 寄存器

#### 3.1 接收缓冲寄存器(RBR)

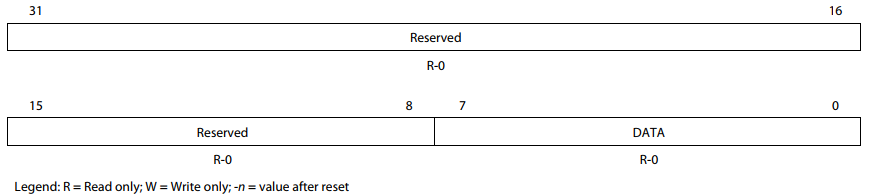


UART接收器部分由接收移位寄存器(RSR)和接收缓冲寄存器(RBR)组成。当UART为FIFO模式时，RBR为16字节的FIFO。采样时钟由16×或13×接收时钟通过设置MDR寄存器的OSM\_SEL位提供。

RSR从RXD引脚接收串行数据。然后RSR将数据移动到RBR(或接收FIFO)。在非FIFO模式下，当一个字节被放置在RBR中并且启用了接收数据完成中断(IER中DR设置为1)，则会生成一个中断。当从RBR中读取该字符时，该中断将被清除。在FIFO模式下，当FIFO填充到FCR中选择的触发阈值时产生中断，当FIFO内容低于触发阈值时中断被清除。

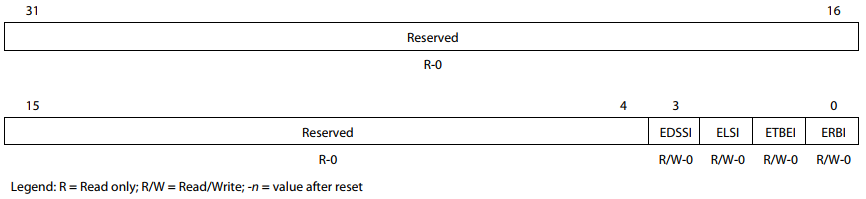
RBR、THR和DLL共用一个地址。如果要读取RBR，将LCR中DLAB位设置为0，然后从共享地址读取。当DLAB设置为0时，向共享地址写入将修改THR。当DLAB设置为1时，共享地址上的所有访问都读取或修改DLL。DLL也有一个专用地址。如果使用专用地址，DLAB可以设置为0，以便在共享地址处选择RBR和THR。

#### 3.2 发送保持寄存器(THR)



THR从内部总线接收数据。当TSR空闲时，UART将数据从THR移动到TSR。UART将TSR中的数据转为串行并在TX引脚上传输。在非FIFO模式下，如果THR为空、TSR为空且则使能THRE中断(IER中ETBEI设置为1)则产生中断。当字节加载到THR中时，该中断将被清除。当UART为FIFO模式时，THR为16字节的FIFO。在FIFO模式下，当发送FIFO为空时产生中断，当有字节加载到FIFO时中断被清除。

#### 3.3 中断使能寄存器(IER)

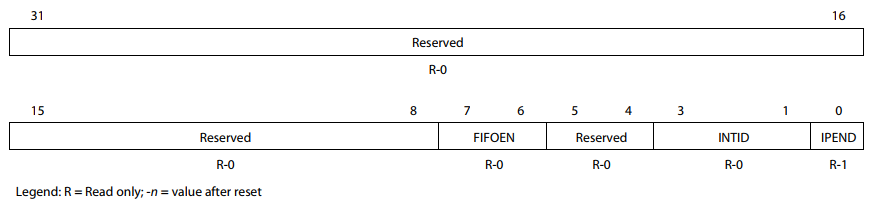


* EDSSI：使能调制解调器状态中断；
* ELSI：接收状态中断使能；
* ETBEI：发射器THR为空中断使能；
* ERBI：接收器数据可用中断和超时中断使能；

IER用于单独启用或禁用UART可以生成的中断请求。在IER中启用的每个中断请求都被转发到CPU。

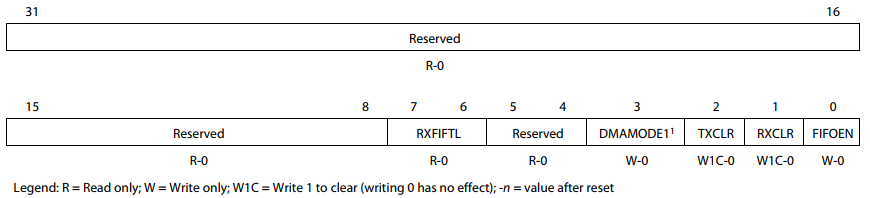
IER和DLH共享一个地址。要读取或修改IER，需要将0写入LCR中DLAB位。当DLAB为1时，共享地址上的所有访问都读取或修改DLH。DLH还有一个专用地址。如果使用专用地址，DLAB可以设置为0，以便始终在共享地址处选择IER。

#### 3.4 中断标识寄存器



* FIFOEN：启用FIFO；0表示为非FIFO模式；当FCR中FIFOEN设置为1时，3表示启用FIFO；
* INTID：中断类型；1表示发射器THR为空(优先级3)。2表示接收器数据可用(优先级2)；3表示接收机状态(优先级1，最高)；6表示字符超时(优先级2)；
* IPEND：中断挂起；当产生任何UART中断并在IER中启用时，IPEND被设置为0。在所有挂起的中断被清除或硬件复位之前，IPEND保持为0。如果没有启用中断，则IPEND被设置为1；

#### 3.5 FIFO控制寄存器(FCR)

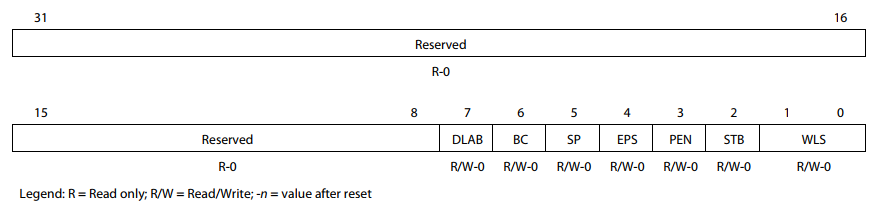


* RXFIFTL：接收器FIFO触发阈值；RXFIFTL设置接收FIFO的触发阈值；当触发阈值达到时，将生成接收数据完成中断(启用)；当FIFO降到触发阈值以下时，中断被清除。0表示1字节；1 表示 4字节；2 表示 8字节；3 表示14字节；
* DMAMODE1：如果使能FIFO, DMA MODE1使能。始终将1写入DMAMODE1。硬件复位后，将DMAMODE1从0修改为1。DMAMOD1设置为1是UART和EDMA控制器之间正确通信的必要条件；
* TXCLR：清除发射器FIFO。TXCLR写1清除。0表示无效；1表示清除发射器FIFO并重置FIFO计数。移位寄存器不会被清除；
* RXCLR：清除接收器FIFO。RXCLR写1清除；
* FIFOEN：开启FIFO模式；FIFOEN必须在写入其他FCR位或未编程FCR位之前设置。清除此位将清除FIFO计数；

FCR是一个只写的寄存器，与IIR的地址相同，而IIR是一个只读寄存器。使用FCR启用和清除FIFO，并设置接收FIFO触发阈值。FIFOEN位必须在写入其他FCR位或不编程FCR位之前设置为1。

IIR和FCR共享一个地址。不管DLAB位的值是多少，从地址读取将得到IIR的内容，而向地址写入将修改FCR。

#### 3.6 线路控制寄存器(LCR)

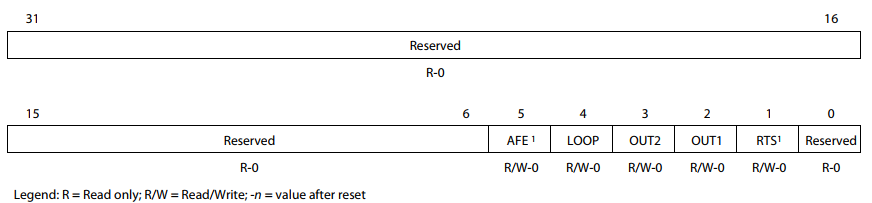


* DLAB：除数寄存器访问位；除数寄存器(DLL和DLH)可以通过专用地址或由RBR、THR和IER共享的地址进行访问；使用共享地址需要切换DLAB来更改所选择的寄存器。如果使用专用地址，DLAB可以设置为0。在RBR、THR和DLL共享的地址上，CPU可以对DLL进行读写。在IER和DLH共享的地址上，CPU可以对DLH进行读写；
* BC：停止控制；0 表示禁用停止条件；1表示停止条件被发送到接收UART；停止是TXD信号被强制设置为清除状态；
* SP：固定校验使能位。SP与EPS和PEN一起工作；
* EPS：偶校验选择；启用奇偶校验时选择奇偶校验(PEN设置为1)；
* PEN：奇偶校验使能。PEN位与SP位和EPS位一起工作；
* STB：停止位个数；STB在每个传输字节中指定1、1.5或2个STOP位。当STB设置为1时，WLS位决定停止位的个数；
* WLS：数据长度选择。每次传输或接收的串行字符中的比特数。当STB 设置为1时，WLS位决定停止位的个数；0表示5位；1表示6位；2表示7位；3表示8位；

使用线路控制寄存器(LCR)来控制异步数据通信的格式。此外，可以检索、检查和修改LCR的内容。

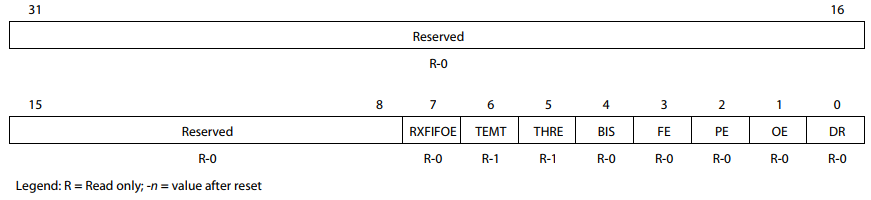
固定校验指校验位始终为1或0，检查通信是否受到干扰。

#### 3.7 调制解调控制寄存器(MCR)



* AFE：流量控制启用。流量控制允许RTS和CTS信号在数据传输过程中提供UART之间的握手信号。AFE设置为1时，RTS位决定启用自动流控制。如果该特性不可用，则该位被保留并且应该被设置为0；
* LOOP：开启环回模式。LOOP用于使用环回特性进行诊断测试。当设置LOOP时，TXD信号设置高。发射器移位寄存器(TSR)的输出回环到接收器移位寄存器(RSR)输入；
* OUT2：OUT2控制位；
* OUT1：OUT1控制位；
* RTS：RTS控制。AFE设置为1时，0表示禁用RTS，只启用CTS；1表示启用RTS和CTS；

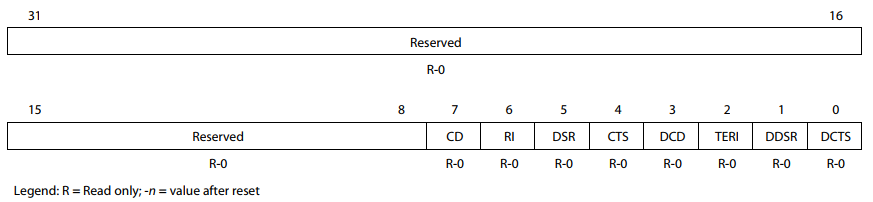
#### 3.8 线路状态寄存器(LSR)



* RXFIFOE：接收器FIFO错误；0表示没有错误或者RXFIFOE被清除；1 表示接收缓冲区寄存器(RBR)中有奇偶校验错误，帧错误或停止；
* TEMT：发送区为空；0表示发送器THR或发送器TSR中包含一个字节数据；1表示THR和TSR都为空；
* THRE：发送保持寄存器(THRE)为空。如果设置了THRE位，并设置了相应的中断使能位(IER中的ETBEI = 1)，则产生中断请求。0表示THR不为空。THR已被CPU加载；1表示THR为空(准备接受一个新字节)，THR的内容已传输到TSR；
* BI：停止标志位。在发送接收数据输入(RXD)保持低位的时间超过全字节传输时间时BI位将会被设置。全字传输时间定义为传输开始位、数据位、奇偶校验位和停止位的总时间。如果设置了BI位，并设置了相应的中断使能位，则产生中断请求；
* FE：帧错误。当接收到的字符没有有效的停止位时，会发生帧错误。为了响应帧错误，UART设置FE位并等待直到RX引脚上的信号变高。当RX信号变高时，接收器准备检测新的START位并接收新数据。如果设置了FE位，并设置了相应的中断使能位，则产生中断请求；
* PE：奇偶检验错误。当接收字符的奇偶校验与LCR中EPS位选择的奇偶校验不匹配时，发生奇偶校验错误。如果设置了PE位，并设置了相应的中断使能位，则会产生中断请求。在非FIFO模式下，0 表示没有检测到校验错误或者PE位被清除；1表示在RBR中检测到字符的奇偶校验错误。在FIFO模式下，1表示在接收FIFO顶部的字符检测到奇偶校验错误；
* OE：溢出错误。非FIFO模式和FIFO模式的溢出错误是不同的。如果设置了OE位，并设置了相应的中断使能位，则产生中断请求。在非FIFO模式下。检测到溢出错误。在RBR中的数据被读取之前，会被RBR中的下一个字符覆盖。在FIFO模式下，如果数据继续填充FIFO超过触发阈值，则只有在FIFO已满并且移位寄存器中已完全接收到下一个字符后才会发生溢出错误。一旦发生溢出错误，新字符覆盖移位寄存器中的数据，但不转移到FIFO；
* DR：接收器数据完成。如果设置了DR位并设置了相应的中断使能位，则产生中断请求；1表示接收数据完成；

线路状态寄存器(LSR)向CPU提供有关数据传输状态的信息。仅用于读操作，

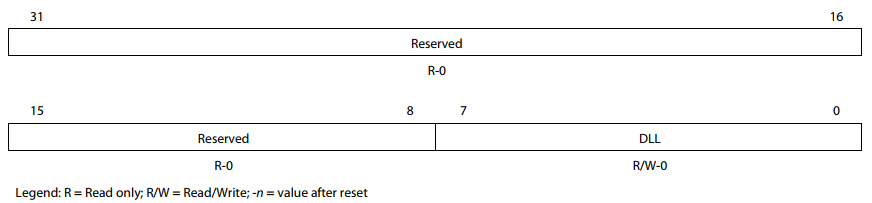
#### 3.9 调制解调状态寄存器(MSR)

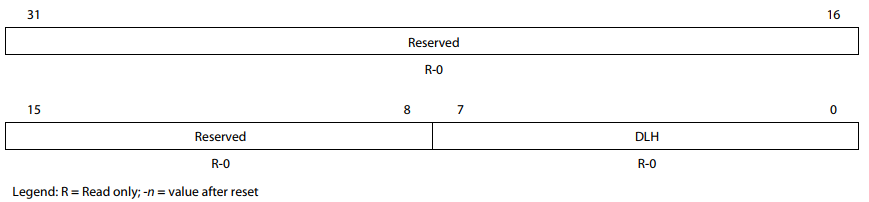


* CD：当UART处于诊断测试模式时，该位等于MCR中OUT2；
* RI：当UART处于诊断测试模式时，该位等于MCR中OUT1；
* DSR：当UART处于诊断测试模式时，该位等于MCR中DTR；
* CTS：当UART处于诊断测试模式时，该位等于MCR中RTS；
* DCD：DCD表示自上次CPU读取DCD输入以来，已经改变了状态。当设置了DCD并启用了调制解调状态中断时，将生成一个调制解调状态中断；
* TERI：TERI表示RI输入由低变为高。当设置了TERI并且启用了调制解调状态中断时，将生成一个调制解调状态中断；
* DDSR：DDSR表示DSR输入自从上次被CPU读取以来已经改变了状态。当设置了DDSR并启用了调制解调状态中断时，将生成一个调制解调状态中断；
* DCTS：DCTS表示CTS输入自从上次被CPU读取后状态发生了变化。当设置DCTS(未启用流量控制且启用调制解调器状态中断)时，将生成一个调制解调器状态中断；当启用流量控制时，不会产生中断。

调制解调状态寄存器(MSR)向CPU提供有关调制解调器控制信号状态的信息，仅用于读操作。

#### 3.10 除频系数寄存器(DLL和DLH)



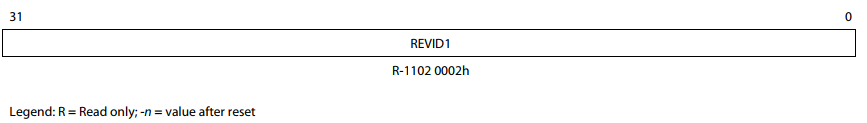


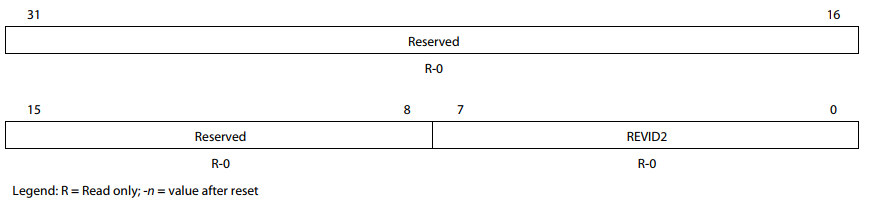
* DLL：波特率发生器中产生波特时钟的16位除数中的低8位；
* DLH：波特率发生器中产生波特时钟的16位除数中的高8位；

DLL和DLH在波特发生器中存储用于波特时钟生成的16位除数。DLH保存除数的最高有效位，DLL保存除数的最低有效位。这些除数必须在初始化UART期间加载，以确保波特发生器的预期操作。写入除数寄存器会导致在写入访问期间插入两个等待状态，同时波特率生成器加载新值。

DLL和DLH也有专用地址。如果使用专用地址，则DLAB位可以保持清除状态，这样共享地址上的RBR、THR和IER总是被选中。

#### 3.11 标识寄存器(REVID1和REVID2)

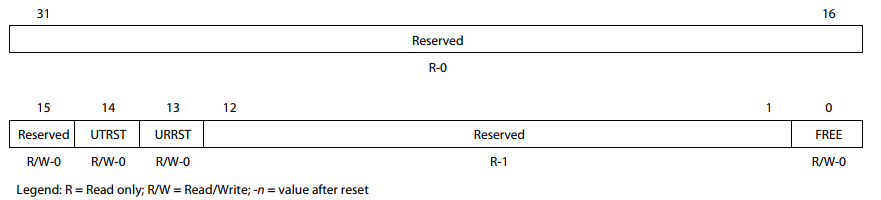




* REVID1：固定值为0x1102\_0002外设识别编号；
* REVID2：外设标识号；

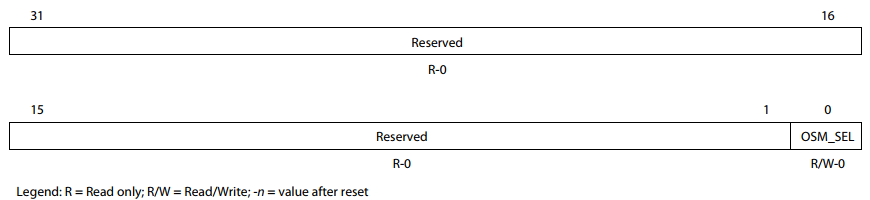
标识寄存器(REVID1和REVID2)包含外设的设备标识数据。

#### 3.12 电源管理寄存器(PWREMU\_MGMT)



* UTRST：UART发送器复位。重置并启用发射机；
* URRST：UART接收器复位。重置并启用接收器；
* FREE：自由运行模式使能。这个位决定了UART的仿真模式功能。当暂停时，UART可以处理寄存器读写请求，但不生成任何传输、接收、中断或事件。0 表示如果传输没有进行，UART立即停止；1表示开启自由运行模式，UART继续正常运行；

#### 3.13 模式选择寄存器(MDR)



* OSM\_SEL：采样模式选择；1表示13×过采样；

模式定义寄存器(MDR)确定UART的采样模式。