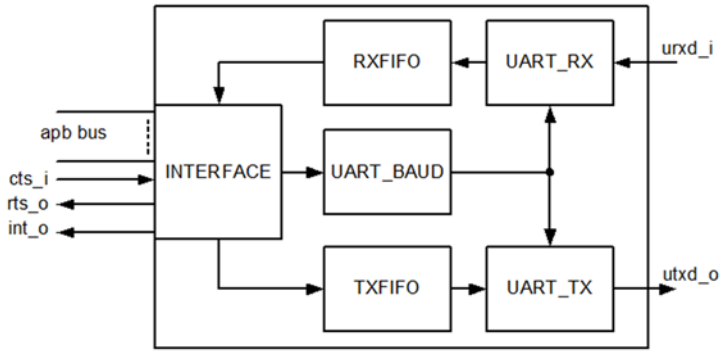
Specification for Design Under Test (DUT) for Project

（仅作为本课程实验所用， 严禁外传， 否则后果自负）

本次 System Verilog Project 我们提供 APB 总线连接的 UART 模块作为 DUT 单元用于后续的验证。系统框图如下。



一、基本概念

1.1 APB 总线

总线：计算机内部和计算机之间传输数据的共用通道。

总线位宽：总线能够一次性传送的二进制数据位数，例如 8/16/32 bit 等。

总线工作频率：即时钟频率。

总线带宽：总线数据的传输速率（单位时间内，总线上传送的数据量。即每秒钟传送 MB 的最大稳态数据传输率），主要用来衡量同步通信工作效率。

APB 总线接口信号定义：APB 总线是 ARM 公司提出的 AMBA 总线结构之一，几乎已成为一种标准的片上总线结构。APB 总线主要用于 SOC 系统中低带宽的周边外设之间的连接，如 UART、IIC 等。外设模块的 APB 信号(slave 端)接口定义如下图所示。

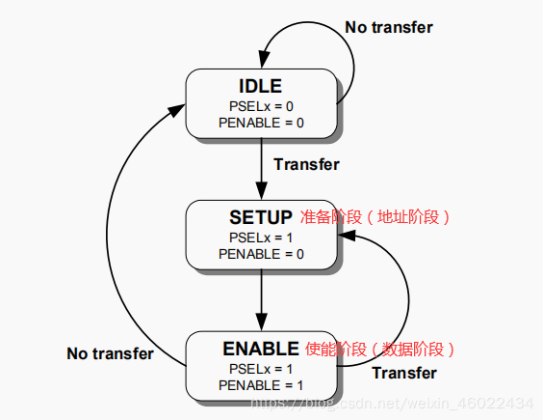
信号名	含义	IO	源	描述
PCLK	总线时钟		CGU	总线时钟，同步所有传输，正时钟沿触发所有信号,所有信号时序与 HCLK 的上升沿相关
PRESETn	复位		RCU	总线复位信号低有效，复位系统与总线。
PADDR[31:0]	地址总线	I:M2S	主	32 位系统地址总线(一般的系统仲裁可以不用到地址)
PSELx	从选择	I	译码器	S 选择信号： 表示当前哪个 S 被选择在传输. 地址选择就是地址译码出来的 S 选择信号 HSELx
PENABLE	APB 选通	I	主	指示 APB 操作的第 2 个周期.
PWRITE	传输方向	I	主	读写操作: 1-写; 0-读
PRDATA[31:0]	读数据总线	O: S2M	从	S2M: 读操作从返回给主的数据总线. 推荐最少 32 位数据总线带宽。可以很方便地扩展到高带宽操作.
PWDATA[31:0]	写数据总线	I	主	M2S: 写操作主传给从的数据总线. 推荐最少 32 位数据总线带宽。可以很方便地扩展到高带宽操作.

控制数据传输的两个阶段

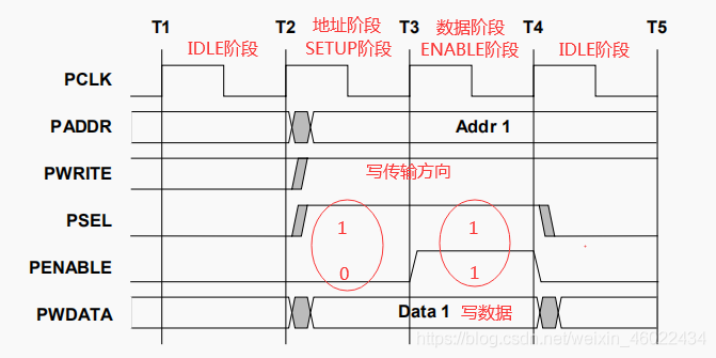
由上表可以看出，APB 信号主要有系统信号（PCLK、PRESETn）、地址（PADDR[31:0]）和控制信号（PENABLE、PWRITE、PSELx）、数据信号（PWDATA[31:0]、PRDATA[31:0]）、状态信号（PREADY）四部分组成，本次实验中使用的 APB 总线为 AMBA 2.0 协议，因此没有 PREADY 信号。

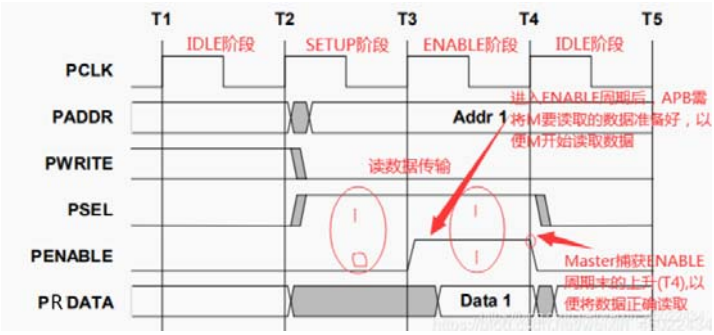
APB 总线时序：APB 总线协议规定了 2 个周期发送一组数据，状态机如下图所示，当一笔数据需要传送时，APB master 会在 SETUP 状态拉高 Psel 信号(选中对应的 slave 设备)，

下一个周期进入 ENABLE 状态拉高 PENABLE 信号进行数据传输，之后如果有数据需要继续传送则进入 SETUP 状态并拉低 PENABLE，否则进入 IDLE。



下面分别是 APB 写传输和读传输时的时序图。





1.2 UART

通用异步收发传输器(Universal Asynchronous Receiver/Transmitter)，通常称作 UART，该接口可以实现双向通信，即全双工传输和接收。UART 进行异步传输时只需要数据收发两根信号线(txd 和 rxd)，无时钟线。

UART 协议数据传输格式以字符为单位，按 bit 传输。一个字符(也称一帧)数据传输格式如下图所示。

空闲位：处于逻辑“1”状态，表示当前线路上没有资料传送。

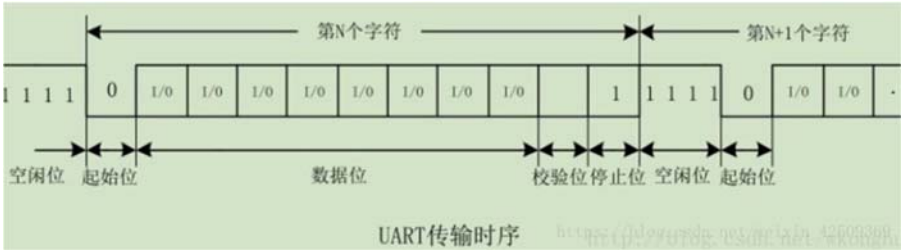
起始位：信号线拉低，表示传输字符的开始。

数据位：依次发送 8bit 数据，从最低位开始发送。

校验位：数据位加上这一位后，使得“1”的位数为偶数（偶校验）或奇数（奇校验）以此来校验数据传送的正确性。

停止位：一个字符数据的结束标志，可以是 1 位、1.5 位、2 位的高电平。

帧间隔：即传送数据的帧与帧之间的间隔大小。



UART传输时序

波特率：每秒钟传输的数据位数，计算公式如下，f表示系统时钟频率，$Uartdiv$则是分频系数，典型的波特率有 2400，9600，115200 等。

$$baud = \frac{f}{16 * Uartdiv}$$

二、APB UART 模块介绍

本次 project 提供的 DUT 实现了一个在 ARM 中通过 APB 总线连接的 UART 模块。有以下特点：

1.系统最大工作频率满足 100MHz，功能时钟 26MHz(参考时钟)；

2.寄存器配置接口满足 AMBA2.0-APB 总线时序接口，总线位宽 32bit；

3.数据传输符合串口时序，奇偶校验功能可配置；

4.波特率可任意配置，配置范围为 2400-115200bps；

5.具有数据收发中断功能，可配置中断触发深度；

6.UART 数据发送帧间隔可配置；

7.具有状态指示功能。

根据系统框图，INTERFACE 模块为 APB 总线寄存器配置模块，实现 APB 读写、中断、功能选择等操作，UART_BAUD 为波特率产生模块，根据分频系数和系统时钟产生 uart 工作波特率，UART TX/RX 模块实现 uart 发送时序。

数据通路：发送数据时，CPU 通过 APB 总线将需要写入的数据配置到寄存器中，然后缓存到 FIFO，UART TX 模块从 FIFO 中读取数据并按照配置发送数据；接收数据时，UART RX 模块会将接收到的数据存储到 FIFO 中，并拉高信号，CPU 接收到终端会通过 APB 读取 FIFO 中数据，如果 UART RX 模块接受数据时判断校验位出错则不会将数据放入 FIFO，状态寄存器会有相应指示。

FIFO 触发深度：UART 模块内部发送 FIFO 和接收 FIFO 分别有 16 字节深度，TX 中断触发深度可设置为 0 到 8，RX 中断触发深度可设置为 1 到 8；发送 FIFO 的触发是发送 FIFO

的数据深度小于或等于设定数目的数据时，产生 TX 中断信号，通知 ARM 核向发送 FIFO 写入新数据；接收 FIFO 的触发是接收 FIFO 中数据深度大于或等于设定数目的数据时，产生 RX 中断信号，通知 ARM 核从接收 FIFO 读取数据。

UART 模块中断信号：当状态寄存器出现异常时，会向 CPU 发送中断信号，用户应该先读取状态寄存器判断异常类型，然后再清除状态寄存器相应位，只有清除所有异常位，中断信号才会变为无效的低电平。

三、寄存器说明

地址 0x00 控制 uart 发送数据，可读可写。bit[7:0] 为需要发送的 8bit 数据。

地址 0x04 控制 uart 数据接收，只读。bit[7:0]为接收的 8bit 数据。

地址 0x08 控制 uart 波特率，可读可写。bit[9:0]为波特率分频系数，范围 13-676。

地址 0x0C 控制 uart 功能模式，可读可写。bit[0]决定是否有校验位，bit[1]决定奇偶校验模式(1-奇校验，0-偶校验)，bit[2]决定是否有停止位(只对 TX 有效)，bit[3]决定是否校验停止位(只对 RX 有效)，bit[14]决定 TX FIFO 复位，bit[15]决定 RX FIFO 复位。

地址 0x10 控制 uart rx fifo 触发深度，可读可写。bit[3:0]设置触发深度，可选范围 1-8。

地址 0x14 控制 uart tx fifo 触发深度，可读可写。bit[3:0]设置触发深度，可选范围 0-8。

地址 0x18 控制 uart 发送帧间隔，可读可写。bit[3:0]设置帧间隔 0-8(只对 TX 生效)。

地址 0x1C 控制 uart 状态寄存器，可读可写。读取该寄存器时，bit[0]表示 TX FIFO 触发中断，bit[1]表示 RX FIFO 触发中断，bit[2]表示 RX 接收数据时奇偶校验错误，bit[3]表示 RX 接收时停止位校验错误。向该寄存器写 1 表示清除中断(不要写 0)。

地址 0x20 控制 rx fifo 状态，只读。bit[5:0]表示当前 RX FIFO 数据量。

地址 0x24 控制 tx fifo 状态，只读。bit[5:0]表示当前 TX FIFO 数据量。

四、接口信号定义

模块顶层信号接口定义如下图。

```
input           clk;                // ARM clk
input           clk26m;             // function clk
input           rst_;               // ARM clk's rst_
input           rst26m_;            // function clk's rst_
input  [31:0]   paddr_i;            // APB address bus
input  [31:0]   pwdata_i;           // APB write data bus
input           psel_i;             // APB module select signal,high active
input           penable_i;          // APB module enable signal,high cative
input           pwrite_i;           // APB write or read signal.1:write,0:read
input           urxd_i;             // UART receive data line

output          utxd_o;             // UART send data line
output          uart_int_o;         // UART interrupt signal.active high
output [31:0]   prdata_o;           // APB read data bus
```