

# 实验原理

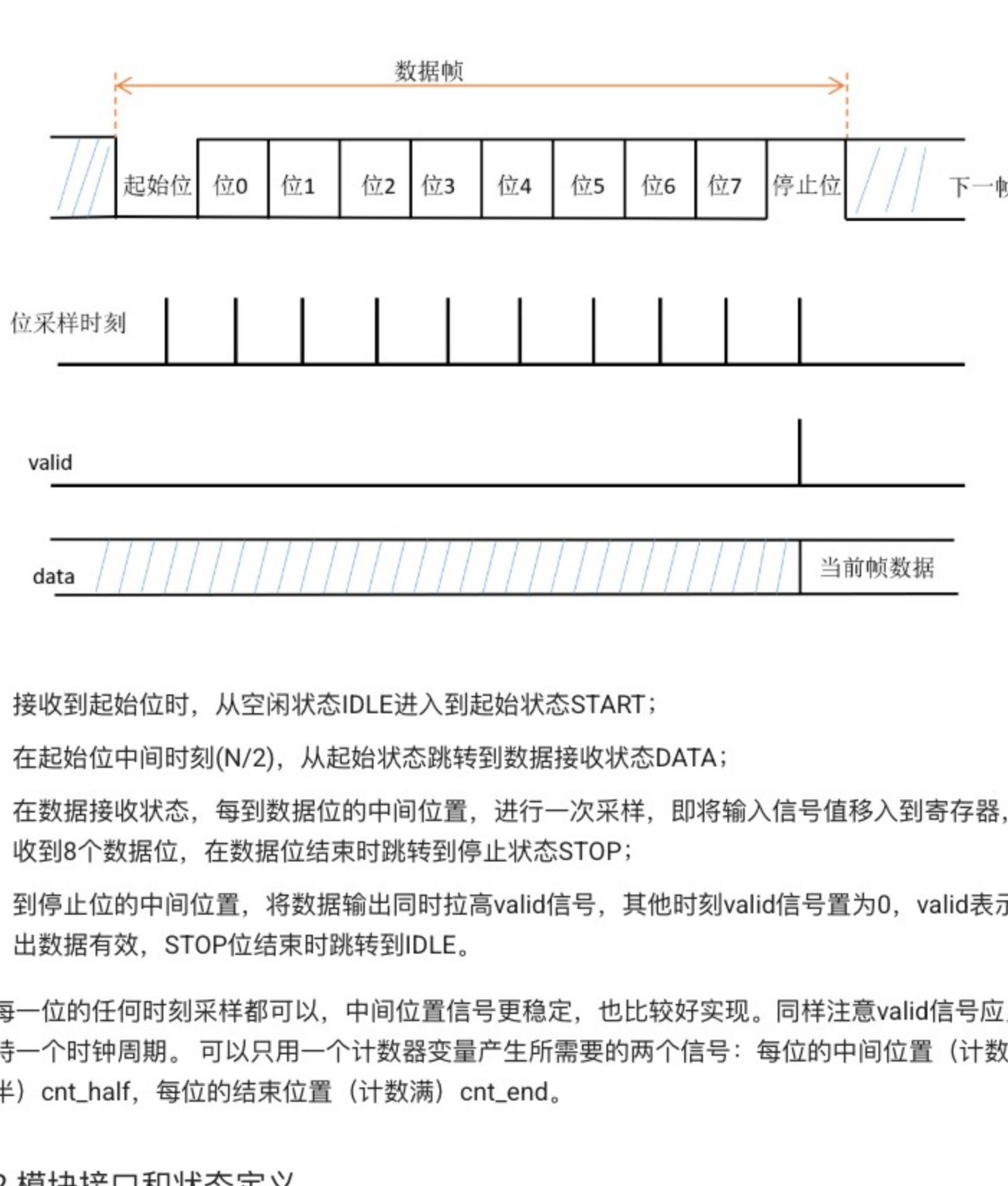
## 1、自顶向下的结构化设计方法

本实验涉及到的功能较多，在编写代码前请按照自顶向下的结构化设计方法根据需求拆解子模块，确认好各个子模块的接口定义，再根据实验指导书实现各个子比如uart\_recv模块，再在顶层模块中例化、连接各个子模块，完成实验要求的收发功能。

自顶向下的结构化设计方法指的是从系统级的功能需求开始，逐步细化为各个子模块或组件的设计，直至最终实现整个数字电路系统。在数字电路设计中，首先根据需求将电路划分为多个组成模块，再分别对每个模块建模，然后将这些模块组成一个总模块。结构化设计的基本单位是模块，每个模块的组成可以遵循以下规则：

- 在设计中被使用两次及以上的逻辑，比如计数器；
- 功能规格明确，封装成模块，比如数码管控制器；
- 单个模块规模过大不利于维护，拆分成若干个小模块；

自顶向下的结构化设计输出最主要的体现是硬件框图，下图是数码管控制器的硬件框图参考示例，通过硬件框图，能够清晰知道整个系统有哪些模块，模块间是如何连接交互。硬件框图要求能正确、清晰地显示模块名、信号名、信号方向和位宽。



根据框图实现各个子模块，在顶层模块例化调用，顶层模块不写功能逻辑，只做例化！！！

## 2、UART接收的状态机实现

### 2.1 接收时序

在实现UART发送的基础上，再来看UART接收，会更容易理解。两者时序处理一样，不同点在于接收端对信号采样的最佳时刻选择了每位的中间，如下图所示，



- 接收到起始位时，从空闲状态IDLE进入到起始状态START；
- 在起始位中间时刻(N/2)，从起始状态跳转到数据接收状态DATA；
- 在数据接收状态，每到数据位的中间位置，进行一次采样，即将输入信号值移入到寄存器，接收到8个数据位，在数据位结束时跳转到停止状态STOP；
- 到停止位的中间位置，将数据输出同时拉高valid信号，其他时刻valid信号置为0，valid表示输出数据有效，STOP位结束时跳转到IDLE。

在每一位的任何时刻采样都可以，中间位置信号更稳定，也比较好实现。同样注意valid信号应只维持一个时钟周期。可以只用一个计数器变量产生所需要的两个信号：每位的中间位置（计数到一半）cnt\_half，每位的结束位置（计数满）cnt\_end。

### 2.2 模块接口和状态定义

实验约定了如下UART接收的接口，根据时序和状态定义，画出对应的状态图，并用三段式状态机实现。

```
'timescale 1ns / 1ps
```

```
module uart_recv(
    input          clk,
    input          rst,
    input          din, // 串口数据输入
    output reg     valid, // 为1表明输出的数据data有效, 只维持一个时钟周期
    output reg [7:0] data
);
```

```
    localparam IDLE = 3'b000;
```

```
    localparam START = 3'b001;
```

```
    localparam DATA = 3'b010;
```

```
    localparam STOP = 3'b011;
```

```
endmodule
```

### 2.3 接收模块仿真验证

实现uart\_recv模块后，下载uart\_recv\_tb.v进行仿真验证，同样会在Tcl Console窗口中输出校验结果，确保10个测试用例都通过。testbench只覆盖了基本的情况，可以自行完善，但不能影响已有 的测试用例。

如果仿真波形中出现clk、rst等为x、z的情况，先检查下uart\_recv\_tb.v是不是被正确设置为 Set as Top。当 Simulation Sources 下有多个仿真文件时，仿真会选择被指定的Top模块运行。具体参考[仿真调试-指定仿真模块](#)进行操作。如果已经正确设置，仿真波形仍然不对，可以尝试重新运行仿真。

## 3、串口软件设置

### 3.1 字符模式和十六进制模式

串口软件发送的时候有字符和十六进制两种模式，点击STR或HEX会进行模式切换。如下输入框左边的模式显示的是STR是字符模式，输入框中的每个字符作为ASCII字符发送出去，比如下图中输入GH，则会先发送G的ASCII值，再发送H的ASCII值。



如下输入框左边的模式显示HEX是十六进制模式，输入框中的每个字符是十六进制的值，不在0-F范围内的不发送，比如下图中输入55，则直接发送十六进制8'h55，等同于STR模式下发送字符U。



下图中的abcd，会当作8'hab, 8'hcd两个8位的十六进制数据发送出去，而非abcd四个ASCII字符发送。



实验只要能在数码管上正确显示接收到的数据，字符或者十六进制模式自行选择。比如字符模式发送5显示5，或者十六进制模式发送8'h55，显示55，或者其他自定义编码方式，只要按编码规则正确的显示接收到的数据。

### 3.2 回车换行

串口调试工具开启“发送时加回车换行”会在每次发送的时候，自动增加LF和CR两个ASCII字符。从ASCII表中可以查到，ASCII值10是LF换行，13是CR回车。对于发送的字符串，可以约定使用回车换行作为一次发送的结束，也可自行约定单个字符比如；作为终结符，用来表示当前的输入序列已经结束。



接收模块通过了仿真，下板测试数码管没任何显示。单从现象无法判断是数码管显示的问题还是串口接收模块的问题，需要构造数据对两个模块单独测试。

- 数码管测试：在代码中，将数码管的输入给固定的值，下板测试是否有数据显示。

- 接收模块测试：将接收模块输出接到开发板上的LED灯，需要考虑将数据保持住，电脑端发送数据，查看LED是否正常显示。LED的控制逻辑简单，容易定位问题是否出在接收模块。

定位到问题所在的模块后，再仔细检查代码，重点查看Messages窗口中有明确提到代码文件和行数的warnings。

## 4、常见问题

接收模块通过了仿真，下板测试数码管没任何显示。单从现象无法判断是数码管显示的问题还是串口接收模块的问题，需要构造数据对两个模块单独测试。

- 数码管测试：在代码中，将数码管的输入给固定的值，下板测试是否有数据显示。

- 接收模块测试：将接收模块输出接到开发板上的LED灯，需要考虑将数据保持住，电脑端发送数据，查看LED是否正常显示。LED的控制逻辑简单，容易定位问题是否出在接收模块。