

题目：智能差分数据处理器（Differential signal Processor）

题目要求：

设计一个差分数据处理模块，集成以下功能：

1. 将差分信号 DP 和 DN 转换为 8-bit parallel data，差分信号以 MSB first 输入，差分信号输入与 clock 是同步关系。
2. 根据 parallel data 解析出正确的 syncword (8' h1D), syncword 不考虑出现在 2T 的情况，syncword 之后为需要输出的 data。
3. Syncword 之前会有一段时间的 data-0。
4. 输出：
 - ※输出 syncword 之后的 8-bit data，并输出 valid 信号
 - ※输出 syncword_vld 信号

解释：

1*. MSB first 为最高有效位，与此相对的是最低有效位，这种一般都出现在传感器和信息传输中。例如对数据的传输和接收，当以二进制传输数据时，接收和发送的数据方式必须一样。

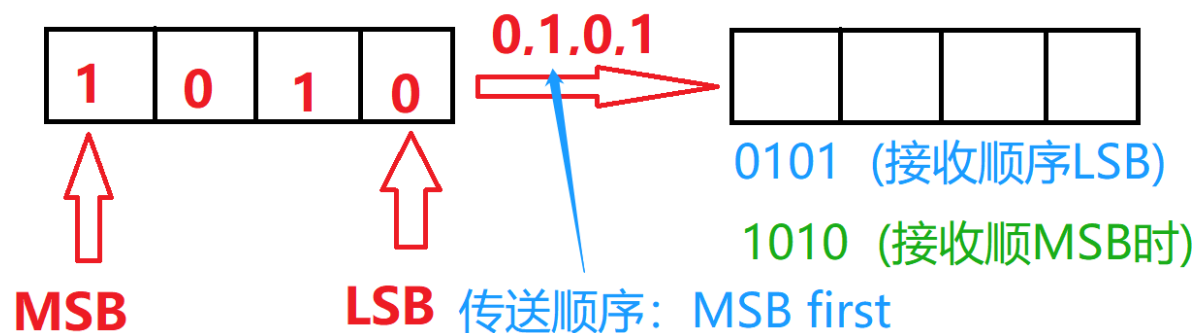


Figure 1: MSB 与 LSB

2*. Syncword (同步字): 是一种特殊的比特序列，在数字通信中用于接收机与发射机之间的同步。通过识别这个特定的比特序列，接收设备能够确定数据帧的开始位置，并与发送设备的时钟同步。

3*. 2T: 这里的 "T" 是指基本的时间单位。

4*. DP (data positive) 和 DN (data negative) 是常用于传输数据的差分信号对，用于减少噪声干扰，提高数据传输可靠性。DP=1&DN=1 表示 stop state，线上不传输数据；DP=0&DN=0 表示 idle state，线上不传输数据；DP=0&DN=1，传输 logic-0；DP=1&DN=0，传输 logic-1。

例：传输 data 为 1D (0001_1101) 时，DP 和 DN 值如下表所示：

| | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|--------|------|------|------|------|------|------|------|------|
| Binary | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 |
| DP | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 |
| DN | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 |

Figure 1: DP, DN 与 data 对应关系

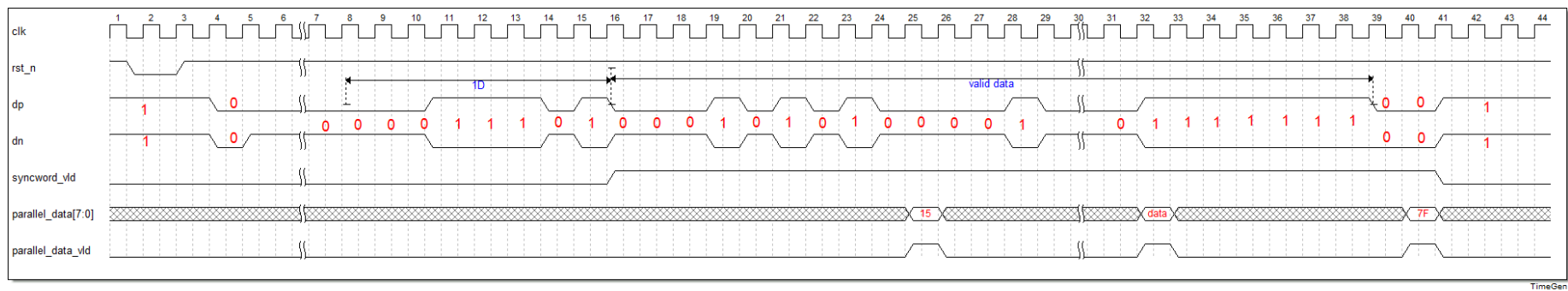
本题时序图请参考图表 1。图中，

※stop state 之后，发送 idle state (DP=0, DN=0)，idle state 只持续 1T，之后为 N 个 byte 的 data-0 (N 为随机值)。

※Data-0 后为 syncword(1D)，此题中 syncword 只会出现在 1T 中，此值不需要通过 parallel_data 输出，但需要拉起 syncword_vld 信号，此信号在检测到第一个 1D 后，在第 1T 拉高，直到出现 idle state (DP=0, DN=0)，在第 2T 拉低。

※Parallel_data 为有效数据（除 data-0 及 syncword 外）输出，在 DP 和 DN 发送完整 byte 数据后，在第 2T 内输出所接收的 parallel_data，同时拉起 parallel_data_vld 信号。

※传输的 data-0, syncword, valid data 均以 byte 为单位，不考虑 shift 的情况（1byte data 出现在 2T 内，比如 1D，1 出现在前一个 byte，D 出现在后一个 byte，这种情况不考虑）。



图表 1: 时序图

接口定义请参考图表 2，请不要改变文件名称，module 名称以及接口名称。

```

1  module differential_signal_processor(
2      input      clk          ,      //system clock
3      input      rst_n        ,      //async reset
4      input      dp           ,      //data positive
5      input      dn           ,      //data negative
6      output reg  syncword_vld ,      //syncword valid
7      output reg [7:0] parallel_data ,      //output data except for syncword
8      output reg  parallel_data_vld ,      //output data valid
9  );
10
11
12  endmodule

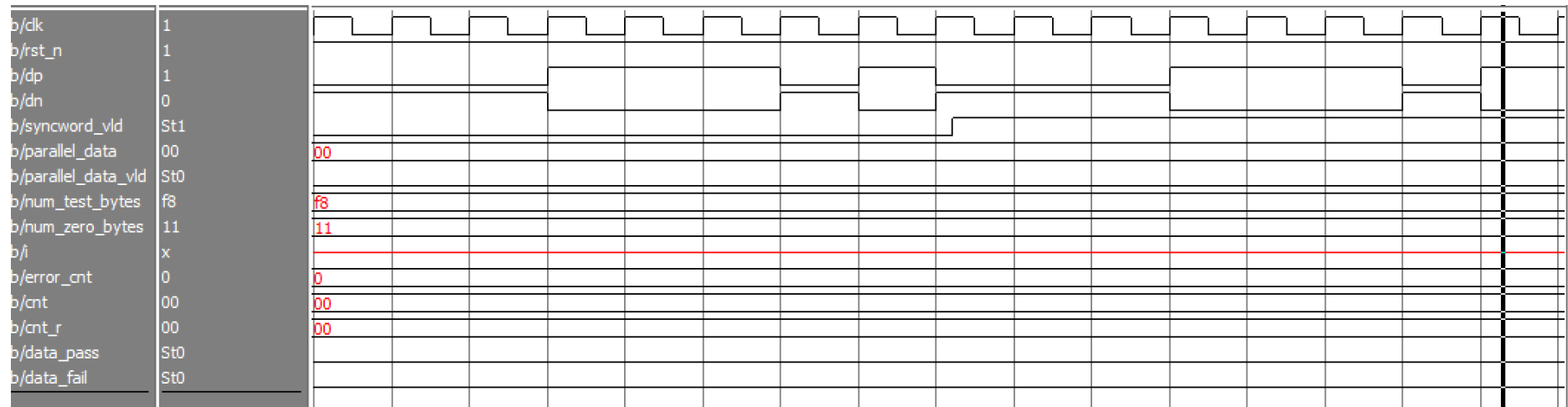
```

图表 2: 接口定义

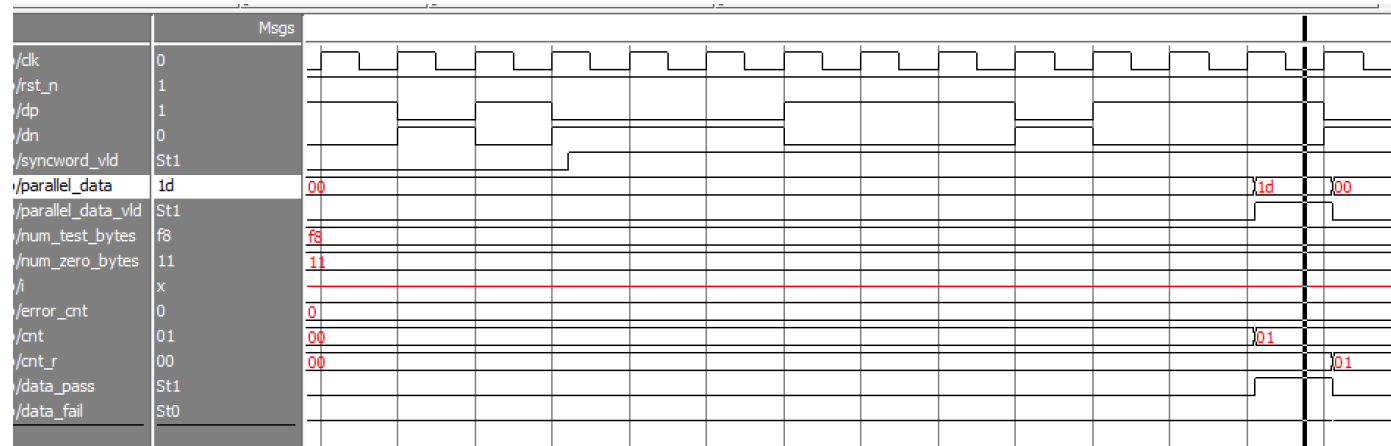
成功运行无错误时, modelsim 窗口打印信息:

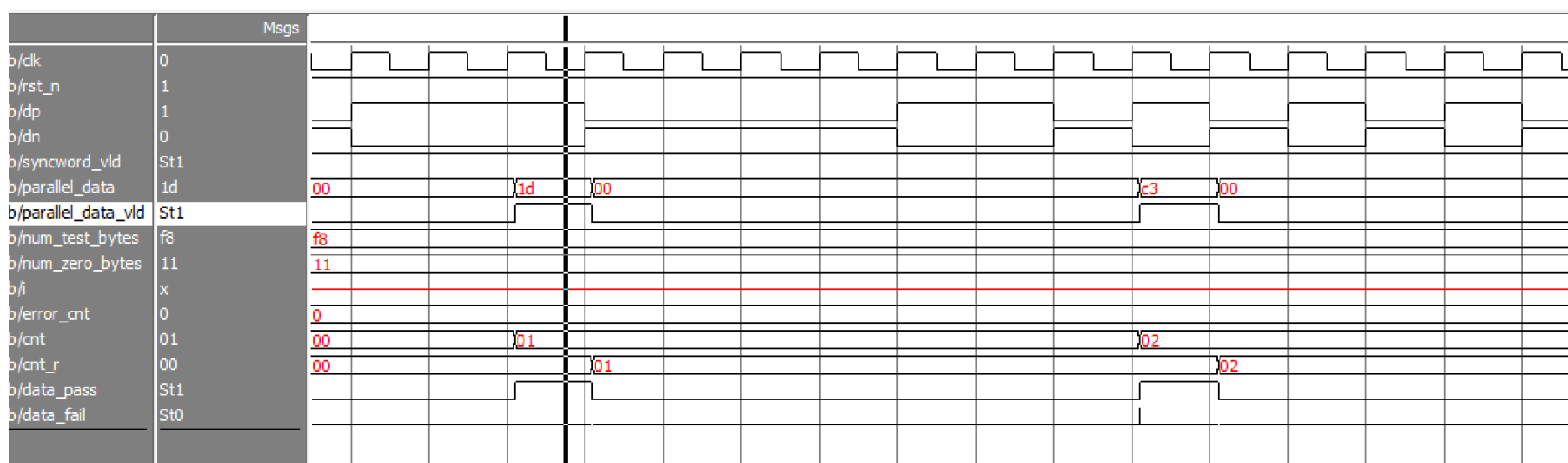
```
#          21040  PASS, parallel_data is ad, expected_data is ad, output MATCH!!!!
#          21100  TX sends data is da
#          21120  PASS, parallel_data is da, expected_data is da, output MATCH!!!!
#          21180  TX sends data is dd
#          21200  PASS, parallel_data is dd, expected_data is dd, output MATCH!!!!
#          21260  TX sends data is e4
#          21280  PASS, parallel_data is e4, expected_data is e4, output MATCH!!!!
#          21340  TX sends data is d5
#          21340  Finished sending 248 bytes...
#          21360  PASS, parallel_data is d5, expected_data is d5, output MATCH!!!!
#          21380  *****
#          21380  *****
#          21380  *****
#          21380  ****Finished sending 248 bytes, total errors are 0****
#          21380  *****
#          21380  *****
#          21380  *****
# ** Note: $finish      : C:/Users/baichuan/Desktop/to_all/differential_signal_processor_tb.v(166)
#    Time: 71380 ns  Iteration: 0  Instance: /differential_signal_processor_tb
# 1
```

syncword_vld 拉高:



parallel data 和 valid 的输出:





末尾 syncword_vld 拉低:

