

## 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)

### 1. 顶层模块简介

我们基于 Verilog 设计实现一个多通道数据整形模块，作为之后的 DUT 单元，用于后续的验证。整个模块的系统框图如下图所示：

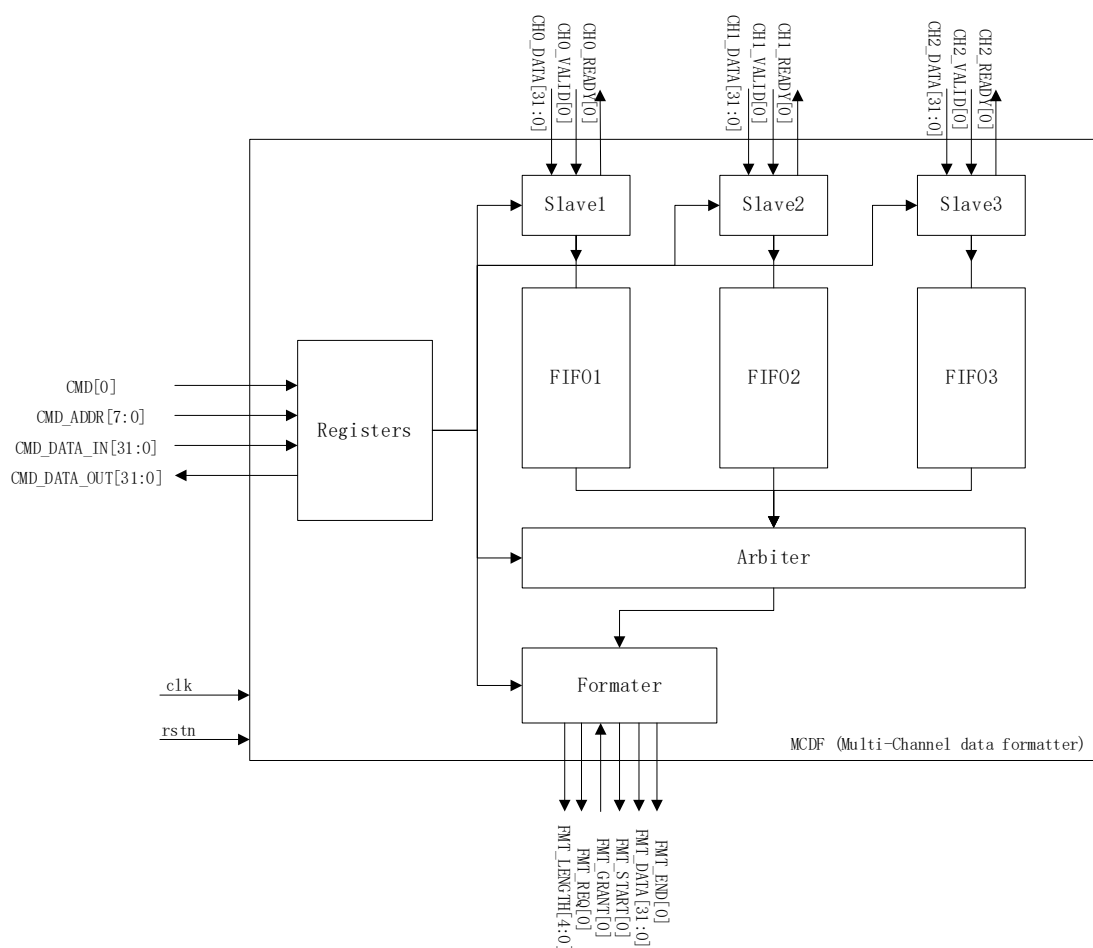


图 1 多通道数据整形器示意图

该设计称为多通道数据整形器 (MCDF, multi-channel data formatter)，它可以将上行多个通道数据经过内部的 FIFO，最终以数据包 (data packet) 的形式送出。同时，上行数据和下行数据的接口协议不同，需要通过 formatter 进行切换。此外，多通道数据整形器也有寄存器的读写接口，可以支持更多的控制功能。

从上图的 MCDF 结构看主要分为如下几个部分：

1. 上行数据的通道从端 (Channel Slave)，负责接收上行数据，并存储在 FIFO 中；
2. 仲裁器 (Arbiter) 可以选择从不同的 FIFO 中读取数据，进而将数据进一步传送至整形器 (formatter)；
3. 整形器 (formatter) 将数据按照一定的接口时序送出至下行接收端；
4. 控制寄存器 (Control Registers) 有专门的寄存器读写端口，负责接收命令对 MCDF 的功能作出修改。

## 2. 接口描述

### 系统信号接口

- clk: 时钟信号;
- rstn: 复位信号, 低位有效。

### 通道从端接口

- CHx\_DATA[31:0]: 通道数据输入
- CHx\_VALID: 通道数据有效标志信号, 高位有效
- CHx\_READY: 通道数据接收信号, 高位表示接受成功

### 整形器接口

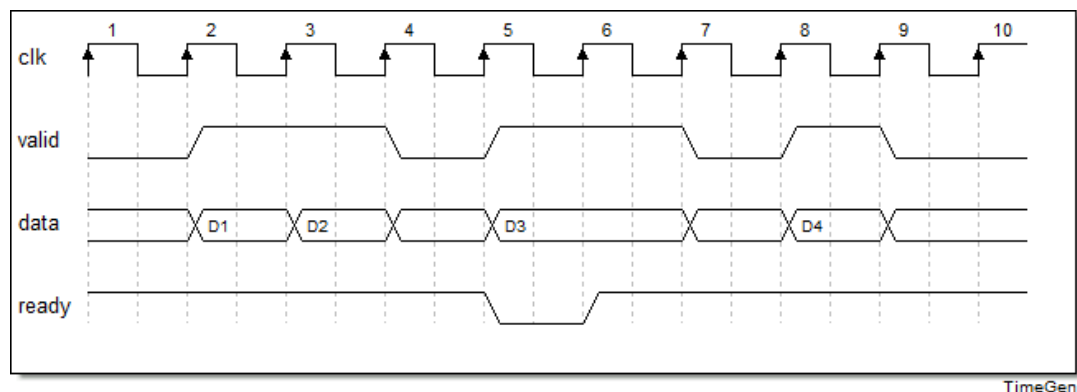
- FMT\_CHID[1:0]: 整形数据包的通道 ID 号
- FMT\_LENGTH[4:0]: 整形数据包的长度信号
- FMT\_REG: 整形数据包的发送请求
- FMT\_GRANT: 整形数据包被允许发送的接受标志
- FMT\_DATA[31:0]: 数据输出端口
- FMT\_START: 数据包起始标识
- FMT\_END: 数据包结束标识

### 控制寄存器接口

- CMD[1:0]: 寄存器读写命令
- CMD\_ADDR[7:0]: 寄存器地址
- CMD\_DATA\_IN[31:0]: 寄存器写入数据
- CMD\_DATA\_OUT[31:0]: 寄存器读出数据

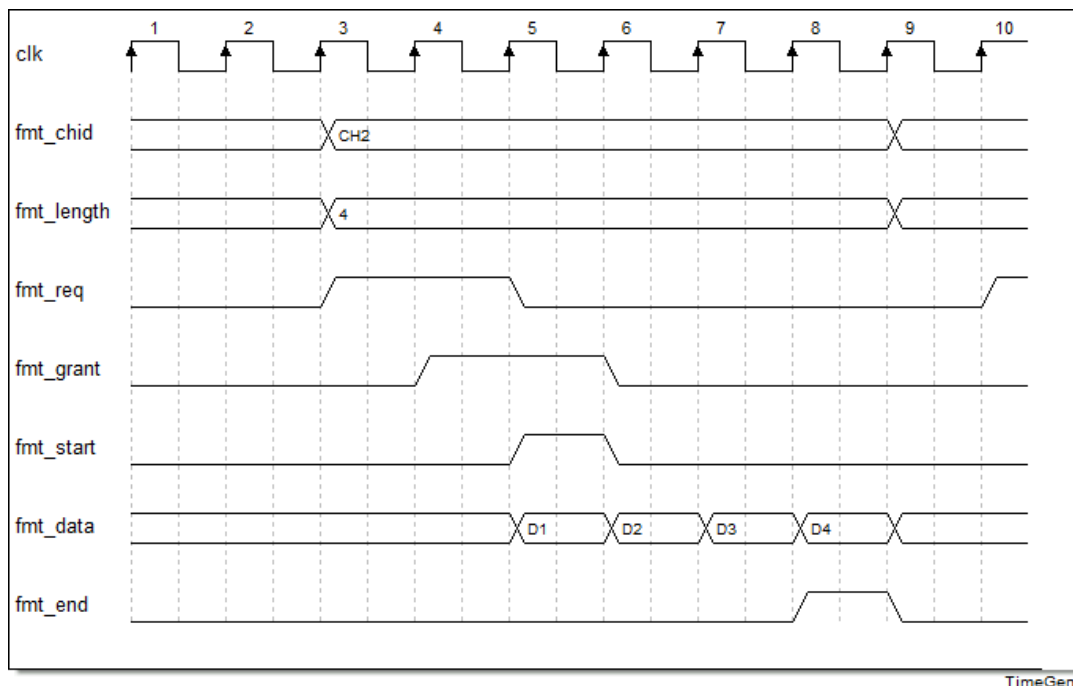
## 3. 接口时序

### 通道从端接口时序



当 valid 为高时, 表示要写入数据。如果该时钟周期 ready 为高, 则表示已经将数据写入; 如果该时钟周期 ready 为低, 则需要等待 ready 为高的时钟周期才可以将数据写入。

## 整形器接口时序

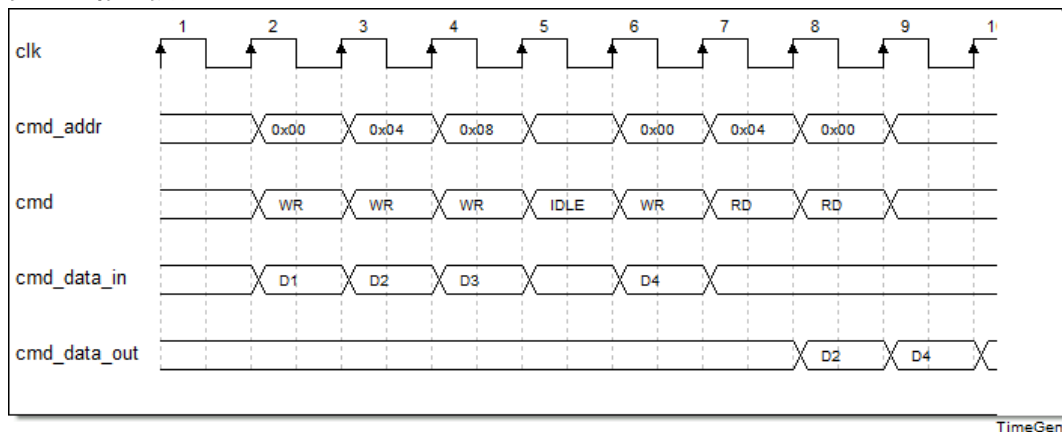


整形器是按照数据包的形式发送数据，可以选择的数据包的长度有 4、8、16、32。整形器必须完整的发送某一个通道的数据包后，才可以准备转而发送下一个数据包，在发送数据包期间，`fmt_chid` 和 `fmt_length` 应该保持不变，直到数据包发送结束为止。

在整形器准备发送数据包时，首先应该将 `fmt_req` 置为高，同时等待接收端的 `fmt_grant`。当 `fmt_grant` 变为高时，应该在下一个周期将 `fmt_req` 置为低。`fmt_start` 也必须在接收到 `fmt_grant` 高有效的下一个时钟被置高，且需要维持一个时钟周期。在 `fmt_start` 被置为高有效的同一个周期，数据也开始传送，数据之间不允许有空闲周期，即应该连续发送数据，直到发完最后一个数据时，`fmt_end` 也应该被置为高并保持一个时钟周期。

相邻的数据包之间应该至少有一个时钟周期的空闲，即 `fmt_end` 从高位被拉低之后，至少需要经过一个时钟周期，`fmt_req` 才可以再次为高。

## 控制寄存器接口时序



在控制寄存器接口上，需要在每一个时钟解析 `cmd`。当 `cmd` 为写指令时，需要把数据 `cmd_data_in` 写入到 `cmd_addr` 对应的寄存器中；当 `cmd` 为读指令时，即需要从 `cmd_addr` 对应的寄存器中读取数据，并在下一个周期，将数据驱动至 `cmd_data_out` 接口。

## 4. 寄存器描述

### 地址 0x00 通道 1 控制寄存器 32bits 读写寄存器

bit[0]: 通道使能信号。1 为打开, 0 为关闭。复位值为 1。

bit[2:1]: 优先级。0 为最高, 3 为最低。复位值为 3。

bit[5:3]: 数据包长度, 解码对应表为, 0 对应长度 4, 1 对应长度 8, 2 对应长度 16, 3 对应长度 32, 其余数值(4-7)均暂时对应长度 32。复位值为 0。

bit[31:6]: 保留位, 无法写入。复位值为 0。

### 地址 0x04 通道 2 控制寄存器 32bits 读写寄存器

同通道 1 控制寄存器描述

### 地址 0x08 通道 3 控制寄存器 32bits 读写寄存器

同通道 1 控制寄存器描述

### 地址 0x10 通道 1 状态寄存器 32bits 只读寄存器

bit[7:0]: 上行数据从端 FIFO 的可写余量, 同 FIFO 的数据余量保持同步变化。复位值为 FIFO 的深度数。

bit[31:8]: 保留位, 复位值为 0。

### 地址 0x14 通道 2 状态寄存器 32bits 只读寄存器

同通道 1 状态寄存器描述

### 地址 0x18 通道 3 状态寄存器 32bits 只读寄存器

同通道 1 状态寄存器描述