

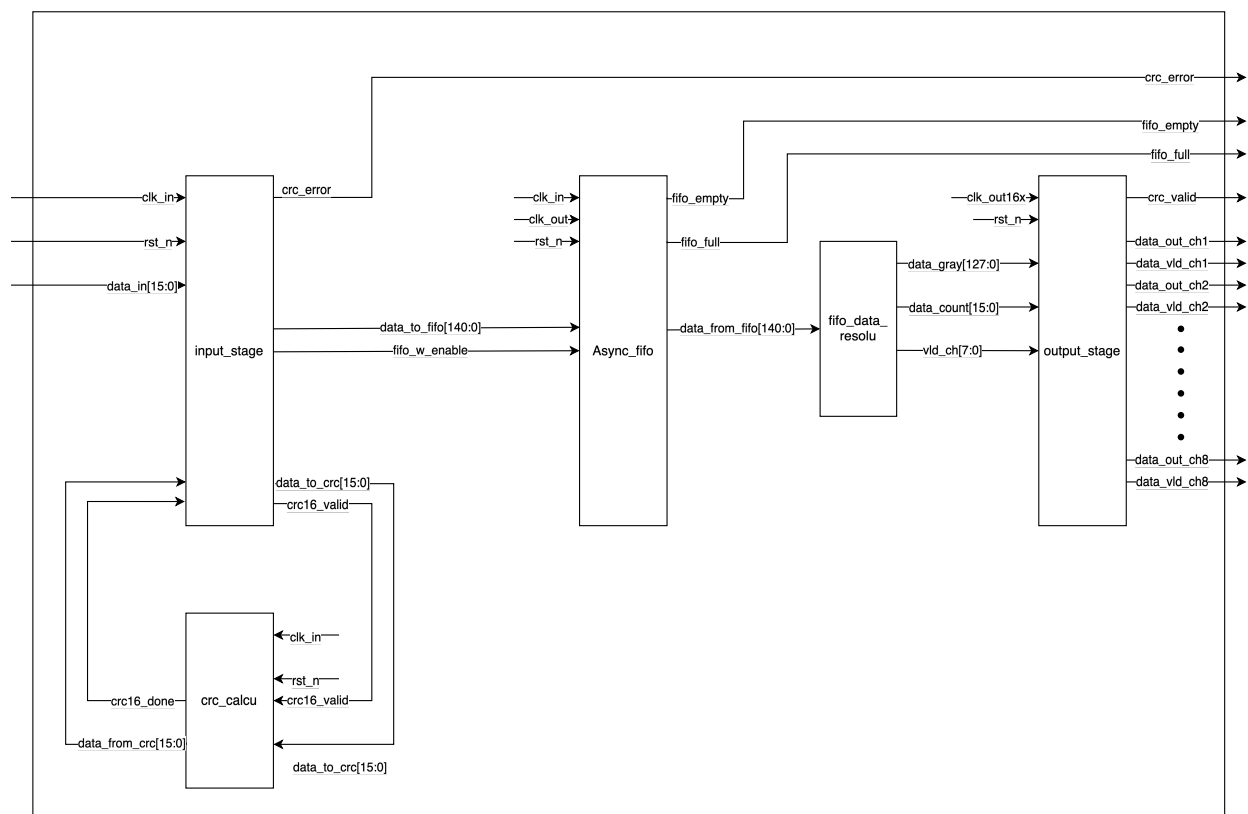
# 帧格式序列检测生成模块设计规范

## Contents

帧格式序列检测生成模块设计规范	1
设计概述 . . . . .	2
子模块概述 . . . . .	3
input_stage . . . . .	3
async_fifo . . . . .	5
fifo_data_resolu . . . . .	6
output_stage . . . . .	7
crc_calcu . . . . .	8

## 设计概述

本规范文件描述了帧格式序列检测与生成模块（Frame Format Sequence Generator）的设计规范和特性要求，该模块旨在检测输入数据流中的特定帧格式，并在检测到目标帧时进行串行输出。



如上框图所示，模块接收到数据输入 `data_in` 后，在模块进行以下处理过程：

1. 输入数据检测: 模块接收并采样输入数据。
2. 帧格式识别: 检测并识别输入数据中的特定帧格式，包括帧头、通道选择字段、数据、CRC 校验字段和帧尾 5 个部分。
3. 解帧与数据提取: 识别帧头、通道选择和帧尾，提取数据部分。
4. CRC 校验: 对提取的数据进行 CRC 校验，确保数据完整性。
  - 如果 CRC 校验成功，数据将被写入异步 FIFO 进行缓存。
  - 如果 CRC 校验失败，模块报告错误并丢弃错误数据。

5. 异步 FIFO 缓存: 提取的数据通过异步 FIFO 进行缓存, 提供 FIFO 空、FIFO 满信号状态指示。
6. 数据编码: 从 FIFO 中读取数据, 按照格雷码进行编码。
7. 根据通道选择进行串行输出: 编码后的数据根据通道选择的数值, 决定在哪个通道进行串行输出

## 子模块概述

### input\_stage

整体说明: 该模块旨在检测输入数据流中的特定帧格式, 并提取有效数据。每帧输入 16 位数据, 输出 5 种状态的指示信号。具体帧格式如下:

- 帧头: 32 位, 取值为 E0E0E0E0
- 通道选择: 8 位, 独热码, 高位数据丢弃
- 数据: N 位 (可变长度, 限制在 16 位到 128 位之间, 是 16 的整数倍, 按照 Big-Endian 方式输入)
- CRC 校验字段: 16 位
- 帧尾: 32 位, 取值为 0E0E0E0E

### 状态机状态定义:

#### 1.IDLE (空闲状态)

- 初始状态, 等待帧头起始
- 行为: 持续监测输入数据是否为帧头的第一部分 E0E0
- 转换条件: 当 data\_in == 16'hE0E0 时进入 HEAD\_CHECK

2.CHANNEL（通道选择状态）

- 读取通道选择字段
- 行为：存储输入数据的低 8 位（高 8 位丢弃）到 data\_ch 寄存器
- 转换条件：无条件进入 DATA 状态（仅需 1 周期）

3.DATA（数据接收状态）

- 接收数据字段和后续字段
- 行为：  
启动 4 位计数器（data\_counter）从 0 开始计数  
启动 4 位计数器（data\_count）从 0 开始计数，每次增加 16 表示数据位宽将 data\_buffer[47:32] 发送给 data\_to\_crc，同时接收 data\_from\_crc 为 crc，保存 data\_buffer[47:32] 到 128 位整段 data 寄存器当检测到 data\_buffer[31:0] == 32’ h0e0e0e0e 且 crc 校验成功输出 data\_to\_fifo 为 data[15:0],112’ d0,ch\_sel[7:0],data\_count[7:0]同时输出 crc\_err 和 fifo\_w\_enable
- 当检测到 data\_buffer[31:0] == 32’ h0e0e0e0e 时返回 IDLE  
如果 data\_count 大于 128 表示数据超长，丢弃数据返回 IDLE；没有超长保留在 DATA（数据接收状态）

顶层 IO

信号	位宽	I/O
clk_in	1	I
rst_n	1	I
data_in	16	I
crc_err	1	O
data_to_fifo	140	O
fifo_w_enable	1	O
data_to_crc	16	O

信号	位宽	I/O
data_from_crc	16	I
crc16_done	1	I
crc16_valid	1	O

## 信号说明

- clk\_in: 输入时钟
- rst\_n: 异步复位信号，低电平有效
- data\_in: 16 位并行输入数据
- crc\_err: CRC 校验结果，错误时为真
- data\_to\_fifo: 输出到 fifo 的数据
- fifo\_w\_enable: fifo 模块的写使能信号
- data\_to\_crc: 输入到 CRC 校验模块的待校验数据
- crc: 输出到帧格式识别模块 crc 校验初始值
- data\_from\_crc: 从 CRC 校验模块返回的数据校验值

## async\_fifo

整体说明: 传统的异步 fifo，宽度为 140 位，深度为 8，提供 fifo 空、fifo 满信号状态指示

## 顶层 IO

信号	位宽	I/O
clk_in	1	I
clk_out	1	I
rst_n	1	I
fifo_w_enable	1	I
fifo_r_enable	1	I
data_to_fifo	140	I
crc	16	O
data_from_fifo	140	O

## 信号说明

clk\_in: fifo 输入时钟域的时钟

clk\_out: fifo 输出时钟域的时钟，与 clk\_in 为同频异步。

rst\_n: 异步复位信号，低电平有效

fifo\_w\_enable: fifo 写使能信号

fifo\_r\_enable: fifo 读使能信号

data\_to\_fifo: fifo 输入数据

data\_from\_fifo: fifo 输出数据，当 fifo\_empty 为 1 时应读出 140 'b0

fifo\_empty: fifo 空信号

fifo\_full: fifo 满信号

## 时序说明

数据将在 fifo\_w\_enable 被拉高的时钟周期内被写入

数据将在 fifo\_r\_enable 被拉高的时钟周期内被读出

## fifo\_data\_resolu

整体说明：纯组合逻辑模块，将从 fifo 读到的 140 位数据分离为 128 位数据位 +8 位通道选择位 +4 位数据长度表示位。其中高 128 位数据位，中间 8 位通道选择位，低 4 位为数据长度表示位。

4 位数据长度表示位生成 16 位具体数据长度位 data\_count，例如 0-8 分别输出 0 16 32 48

64 80 96 112 128。

数据位高位在前，长度由 `data_count` 给出，不足 128 位的数据将在低位补 0。数据位转为格雷码后，在低位补 0 补齐 128 位，输出信号 `data_gray` 也是高位在前。

8 位通道选择位不做处理，直接输出 8 位信号 `vld_ch`。

## 顶层 IO

信号	位宽	I/O
<code>data_from_fifo</code>	140	I
<code>data_gray</code>	128	O
<code>vld_ch</code>	8	O
<code>data_count</code>	16	O

## 信号说明

`data_from_fifo`: fifo 输出数据 `data_gray`: 输出数据的格雷码表示 `vld_ch`: 通道选择数据  
`data_count`: 具体数据长度位

## output\_stage

整体说明: 本模块根据前面模块提供的信息，将数据发送到对应的通道并转为串行信号输出。根据 `vld_ch` 所输入的独热码，将输入的 `data_gray` 发送到对应的数据串行输出通道 (`data_out_ch1~8`)，同时拉高对应通道的数据有效信号 (`data_vld_ch1~8`)。根据 `data_count`，具体决定输出 `data_vld_ch` 的持续周期数，即 `data_vld_ch` 仅在输出时拉高。

## 顶层 IO

信号	位宽	I/O
<code>rst_n</code>	1	I
<code>clk_out16x</code>	1	I
<code>data_gray</code>	128	I

信号	位宽	I/O
vld_ch	8	I
data_count	16	I
crc_valid	1	O
data_out_ch1	1	O
data_out_ch2	1	O
...	1	O
data_out_ch8	1	O
data_vld_ch1	1	O
data_vld_ch2	1	O
...	1	O
data_vld_ch8	1	O

## 信号说明

- rst\_n: 异步复位信号，低电平有效
- clk\_out16x: 输出串行信号速率的时钟
- data\_gray: 输出数据的格雷码表示
- vld\_ch: 通道选择数据，8 位宽独热码
- data\_count: data\_gray 的具体数据长度
- crc\_valid: 任一 data\_vld\_ch\* 拉高时即拉高
- data\_out\_ch1~8: 数据串行输出通道，输出时高位在前
- data\_vld\_ch1~8: 数据有效信号通道，仅在输出时拉高

## crc\_calcu

设计一个 CRC 校验模块，采用 CRC-16/CCITT 算法，多项式：0x1021，初始值：为输入的 crc 信号，输入不反转，输出不反转，输出异或值：0x0000。

## 顶层 IO



信号	位宽	I/O
data_to_crc	16	I
crc	16	I
data_from_crc	16	O

## 信号说明

data\_to\_crc: 从帧格式识别模块输入的待校验数据

crc: 从帧格式识别模块输入的 crc 校验初始值

data\_from\_crc: 输出数据的 CRC 校验值