OpenEmulator 硬件设计方案

（版本 1.0）

OpenTSN 开源项目组

2022 年 2 月

版本历史

版本	修订时间	修订内容	文件标识
1.0	2021.12.16	初版编制	OpenEmulator

目录

1.概述

TSN 网络在部署前需要对网络进行验证以确保可靠性，而在真实的 TSN 网络环境中对各种情况进行模拟（如时钟同步过程中的故障情况）的难度较大。基于该问题构建了基于仿真软件面向 TSN 的跨域仿真验证工具 OpenEmulator。

2.术语定义

pkt_in	输入报文
pkt_out	输出报文
osc	时钟晶振信号
ticker_counter	位宽 48bit，仿真域维护的同步时钟计数器
sim_enable	晶振信号输出控制，当该信号为高，被干预的晶振会停止输出
sync_clock	同步时钟计数器在某一时刻的值，周期性出现
app_clock	软件域维护的同步时钟
sync_pkt	携带 sync_clock 的同步报文
syncack_pkt	携带 app_clock 的同步响应报文

3.平台方案

3.1.实现思路

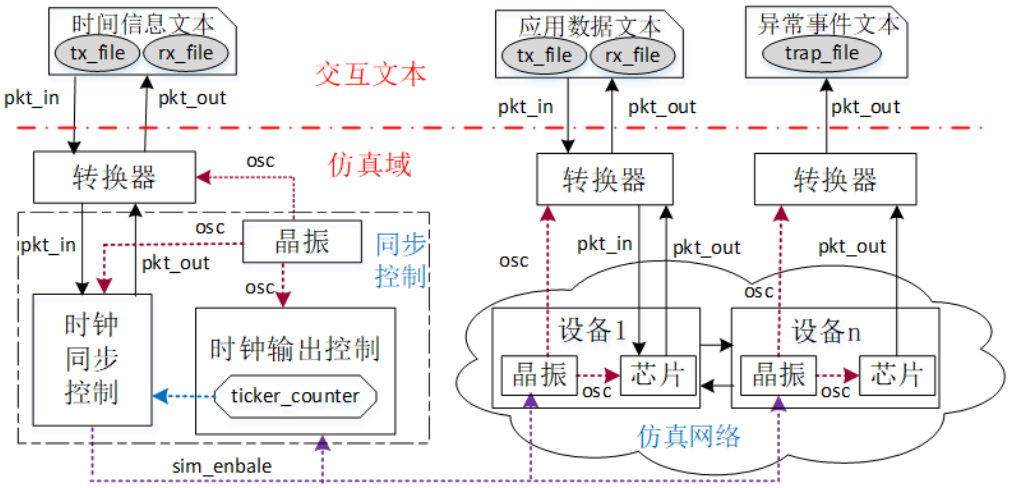


图 3-1OpenEmulator 模型图

跨域仿真验证平台主体分为三部分：软件域、数据交换文件以及仿真域。仿真域在仿真软件中使用硬件描述语言来仿真硬件环境，软件域中的应用中可以是具有实际功能的应用程序或者故障模拟程序。仿真域与软件域通过对数据交换文件的读写完成报文的传递。

仿真域包括“仿真时间敏感网络”、“同步控制模块”、“转换器”以及“工作时钟源模块”。其中转换器主要用于满足向 TSN 网络注入各类业务流量的需求，负责向交换文件中读写报文消息。“工作时钟源模块”为仿真域中所有模块提供工作时钟频率；“同步控制模块”协同“工作时钟源模块”实现软件域和仿真域之间的时钟同步。

软件域使用高级程序描述语言来编写应用或故障模拟程序。软件域通过维护一个进程来读写交换文件，以向应用程序提供报文读写接

口；同时维护了一个“时钟同步管理程序”来维护软件域的时钟并提供获取时钟的接口。

3.2.跨域时间同步过程

跨域仿真验证平台中软件域与仿真域之间通过“互锁机制”实现跨域时间同步。仿真域的时钟封装成报文周期性的上送至软件域，软件域将同步报文中的时间戳设置为软件域时钟后，再以同步确认报文携带时间戳的方式转发至仿真域。仿真域中的同步控制模块获取同步确认报文中的时间戳，根据该时间戳与此时仿真域的时钟差Δt，来决定是否干预仿真域与时间敏感网络中的工作时钟源，以使软件域时钟和仿真域中的时钟同步在一定的精度内。

软件域和仿真域时钟同步的具体过程如下：

（1）仿真域周期性地将当前时钟值 sync_clock 发送给转换器。

（2）转换器将 sync_clock 封装到sync_pkt报文写入文本，并由软件域中的进程从交换文件中读取报文，并交给“时钟同步管理进程”。

（3）“时钟同步管理进程”读取该报文中的时间戳 sync_clock，并以此更新软件域的时钟。由于 sync_clock 从仿真域到软件域中会经历报文创建、文件读写等过程，当“时钟同步管理进程”更新时钟 app_clock 时，时钟与 sync_clock 之间会存在误差。为控制时钟与 sync_clock 之间的误差，继续进行以

下过程。

（4）“时钟同步管理进程”创建携带软件域时钟值的同步确认报文 syncack_pkt，并写入同步报文交换文件。

（5）转换器从交换文件中读取syncack_pkt(app_clock)，转发给“同步控制模块”。

（6）“同步控制模块”取出app_clock，并判断|app_clock − sync_clock| 与 Δt 之间的大小关系。当 |app_clock − sync_clock| > Δt时，“同步控制模块”向“工作时钟源模块”发送高电平的控制信号sim_enable，“工作时钟源模块”则停止向仿真域中的模块提供工作频率。由于“时钟同步管理进程”会继续读取交换文件中的同步报文 sync_clock，并不断更新软件域的时钟，从而不断更新“同步控制模块”中的 app_clock，直至 |app_clock − sync_clock| ≤ Δt；当 |app_clock − sync_clock| ≤ Δt 时，“同步控制模块”向“工作时钟源模块”发送低电平的控制信号sim_enable，“工作时钟源模块”则正常提供工作频率。

通过上述过程，能够将软件域同步时钟app_clock与仿真域时钟sync_clock之间差距始终控制在Δt之内。Δt可以根据实际的需求，如软件域中应用的周期来确定。

4.OpenEmulator 硬件设计

4.1.同步通路硬件实现架构

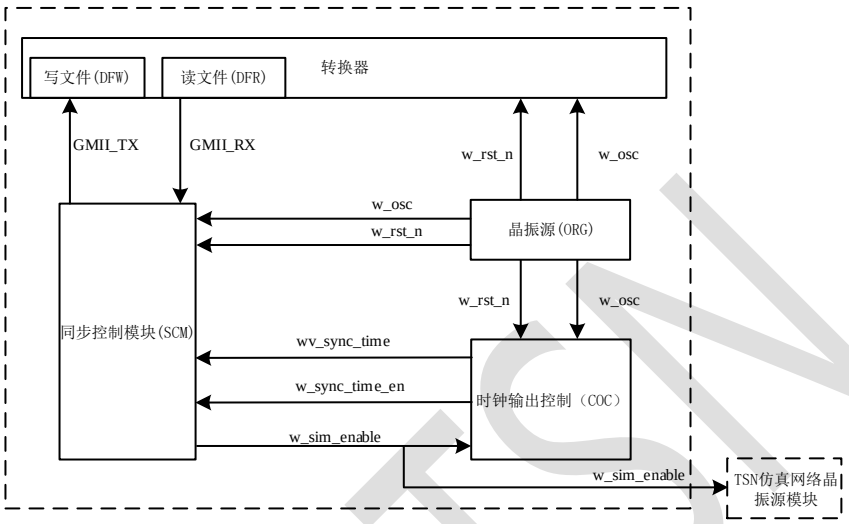


图4-1. 同步通路硬件设计总体架构图

硬件实现总体架构如图 4-1 所示。

时钟源（osc_rst_generate）模块主要负责时钟晶振信号、复位信号的产生。

时钟输出控制(clock_output_control)模块主要功能是根据控制信号 sim_enable 来控制时钟晶振信号的输出，如果 sim_enable 为高电平，输出时钟信号为低电平，如果 sim_enable 为低电平，输出时钟信号正常输出，同时维护一个 48bit 的时钟计数器，以时钟晶振（sim_enable 控制后的晶振输出）的上升沿进行计数，维护一个本地计数器，每当计数满 10us，维护的时钟计数器产生一个有效脉冲信号。

同步控制(sync_control_module)模块主要负责将带有使能有效信号的时钟计数器通过转换器封装成报文写入到文本中，并且根据从

转换器的读文本中读出软件返回的报文，从读取的报文中解析出时间戳，与当前时钟计数器的差值来控制 sim_enable 信号的输出，如果差值大于某个阈值（20000），sim_enable 输出为高，否则 sim_enable 输出为低。

转换器模块主要有两方面的功能，一方面是负责将报文写入到文本中，每当写完一个完整报文，写入尾部判断“1111”，以便软件进行解析；另一方面从文本中读出数据，优先从报文完成写入的状态文本中获取标识信号“1”，只有读到标识信号后，再从文本中读出报文。

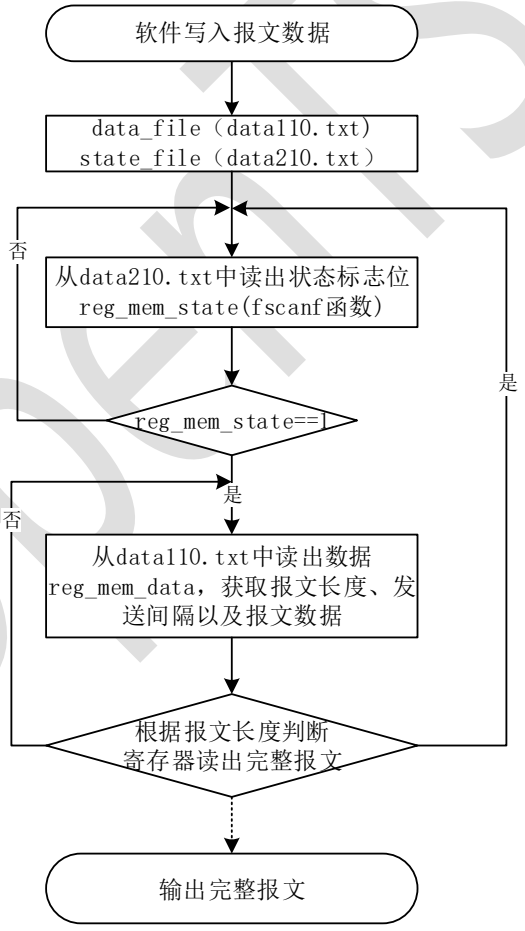
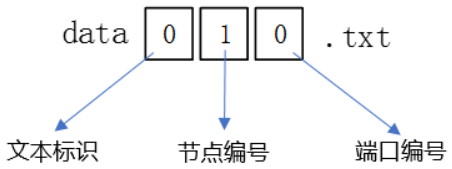


图4-2. 转换器-硬件读取文本流程图

附录 A.转换文本命名规则

文本的命名规则主要体现在三位数字的组合，详细介绍如下：



文本标识：三种状态，分别是 0、1、2。该位置为 0 表示该文本是硬件写入、软件读取的文本，存放报文数据；该位置为 1 表示该文本是软件写入、硬件读取的文本，存放报文数据；该位置为 2 表示该文本是软件写入、硬件读取的文本，存放报文数据写入完成标志信号。

节点编号：与 TSN 仿真网络的组网规模有关，按节点数进行编号，初值从 1 开始。

端口编号：表示每个节点的端口编号。目前端口取值 0-7，表示节点最大支持 8 个端口。