

高层次综合技术在遥测地面站数据处理中的应用

迟东明, 刘兆辉, 王燕玲

(上海航天电子有限公司, 上海 201800)

摘要: 针对遥测地面站所接收的遥测数据的数据结构复杂, 使用 FPGA 实现遥测数据处理难度较大的问题, 提出了基于高层次综合技术的地面站遥测数据处理方法; 相对于传统的 RTL 实现方式, 利用高层次综合技术, 在实现同等功能的条件下, 能够有效缩短 FPGA 设计、开发周期; 经测试, 使用高层次综合技术对遥测数据进行处理, 其代码量仅为 RTL 实现方式的 1/10 至 1/20, 代码开发周期缩仅为 RTL 实现方式的 1/4 至 1/10, 且代码运行效率并无明显降低。

关键词: FPGA; 高层次综合; 遥测数据处理; 浮点计算

Application of High-Level Synthesis in Data Processing of Telemetry Ground Station

Chi Dongming, LIU Zhaohui, WANG Yanling

(Shanghai Aerospace Electronics Co. Ltd, Shanghai 201800, China)

Abstract: The data structure of telemetry data received by telemetry ground station is complex, there are difficulties in processing telemetry data based on FPGA. Thus the High-Level Synthesis technology is introduced. High-Level Synthesis can shorten FPGA design and development cycle compared to traditional RTL implementation when implementing identical functions. Comparing to RTL implementation, the code size about telemetry data processing is only 1/10 to 1/20, the development cycle is only 1/4 to 1/10, and the code execution efficiency is identical.

Keywords: FPGA; high-level synthesis; telemetry data processing; floating-point calculation

0 引言

遥测地面站所接收的火箭、卫星等飞行器的原始遥测数据中包含大量的重要遥测信息, 如姿态信息、油量信息、位置信息等。地面站需要将这些关键遥测参数从原始数据中提取出来并进行综合、计算, 最后将处理的结果显示在终端设备上, 以便操作人员进行分析、决策。随着火箭、卫星等飞行器的结构日益复杂, 地面站所接收的遥测数据不仅数据量更大, 且数据结构更也为复杂, 这就对地面站的数据处理能力提出的更高的要求。

地面站对于遥测信息的处理一般采用“FPGA+DSP+通用 CPU”或“FPGA+通用 CPU”的方式。

“FPGA+DSP+通用 CPU”的实现方式中, FPGA 负责数据帧检索、重组帧等密集型计算, DSP 负责遥测数据计算、数据判断等复杂度较高的计算, 而通用 CPU 负责数据决策、人机交互等。这种实现方式适应面广, 但集成度低且功耗较大, 不利于地面站的小型化设计。

“FPGA+通用 CPU”的实现方式中, 如果 FPGA 负责数据帧检索、重组帧等密集型计算, 通用 CPU 负责其他计

算, 则 FPGA 与通用 CPU 之间需要传输大量数据, 对于总线的吞吐量及通用 CPU 的性能提出了极高的要求, 不利于地面站低成本设计。虽然目前 FPGA 内部可以集成通用 CPU 的 IP, 但由于通用 CPU IP 性能较弱, 无法满足高速率的遥测数据处理任务。

而如果 FPGA 负责数据帧检索、重组帧、遥测数据计算、数据判断等, 通用 CPU 负责其他计算, 虽然降低了总线及通用 CPU 的负担, 但由于遥测数据处理往往涉及到复杂的数据检索、浮点计算以及数据分析, 如果使用传统的基于寄存器传送级 (register transfer level, RTL) 的设计流程实现遥测数据处理, 其代码的设计、调试难度会大大增加, 且代码的维护成本也会加大。

本文将 FPGA 的高层次综合技术 (high-level synthesis, HLS) 引入到地面站的遥测数据处理中, 利用该技术可以将遥测数据处理功能用抽象层级较高的 C 代码实现, 这样就有效降低了 FPGA 进行遥测数据处理的设计难度, 提高了开发效率, 从而缩短了产品的研制周期。

1 地面站遥测数据处理流程介绍

地面站的遥测数据处理流程一般是如图 1 所示的方

收稿日期: 2021-03-05; 修回日期: 2021-03-17。

作者简介: 迟东明(1981-), 男, 吉林四平人, 高级工程师, 主要从事自动化测试方向的研究。

引用格式: 迟东明, 刘兆辉, 王燕玲. 高层次综合技术在遥测地面站数据处理中的应用[J]. 计算机测量与控制, 2021, 29(9): 210-214, 219.

式进行^[1-2], 主要分为 5 个步骤:

1) 遥测解调。地面站对接收到的无线遥测信号进行解调, 并将解调出的遥测数据以数据流形式发送给后端的遥测数据处理模块做进一步处理。

2) 遥测帧数据重组。遥测数据处理模块对解调后的遥测数据流进行检索, 并根据遥测帧同步头检索出完整的遥测帧, 最后根据遥测帧的数据结构进行遥测帧数据重组, 从而得到连续的遥测帧。

3) 遥测参数提取。遥测数据处理模块根据一定的提取规则, 对遥测帧内的遥测参数进行提取, 从而得到需要的遥测参数。

4) 遥测参数计算。遥测数据处理模块对提取出的遥测参数进行计算及分析, 最终得出计算结果。

5) 计算结果组帧。遥测数据处理模块将计算结果打包并根据一定的成帧规则形成新的数据帧发送给后端的数据处理模块或终端设备进行后续数据处理或数据显示。

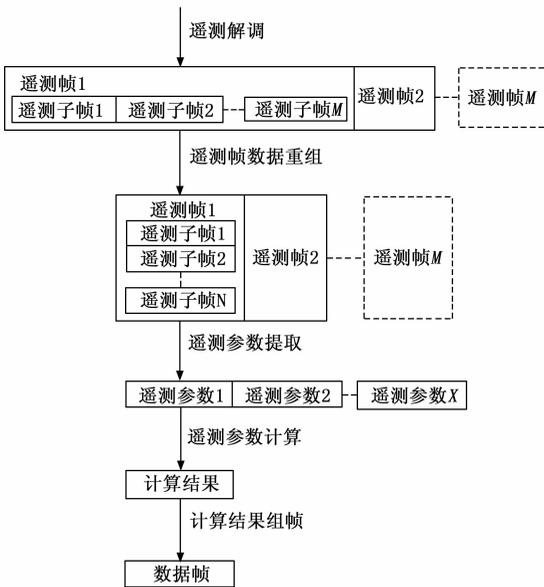


图 1 地面站遥测数据处理流程

一般情况下遥测数据是以一系列遥测帧的形式进行存储和传输的, 如果遥测数据的数据结构较复杂, 则每一个遥测帧又可分为一系列不同的遥测子帧, 子帧内部的不同区域存放有各类遥测参数或其他数据。换言之, 遥测帧一般以二维数据表的形式存放各类参数及数据, 而遥测参数则以某一点、某一行(列)或多行(列)的形式分布于该二维表中, 其数据结构示意图如图 2 所示。

由于火箭、卫星等飞行器内部包含数量众多的传感器和各类设备, 因此其遥测数据的数据结构可能会变得非常复杂。以图 2 为例, 遥测参数集 1 是遥测帧中子行(列)抽取出来的二维数组, 数组中数据可以代表飞行器总线中的某些总线指令, 而总线指令在遥测参数集 1 中的出现的时间以及相对位置可能都不固定, 需要遥测数据处理模块通过总线指令的同步头对接收到的每一个遥测帧进行搜索。

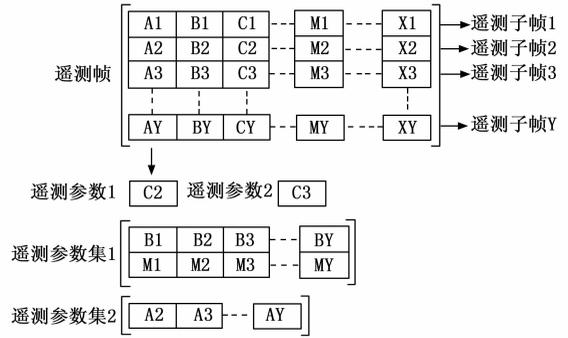


图 2 遥测数据的数据结构示意图

而搜索到总线指令后, 还要对指令进行校验, 校验通过后才能提取其中的遥测参数并进行计算。同时在处理遥测帧时也可能需要对遥测帧进行计数操作, 以便统计通信状态或超时重传等。

由于飞行器内传感器及设备工作状态各异, 其遥测参数的数据类型也千差万别, 通常情况下提取出的遥测参数的数据类型并不一致, 需要经过一系列变换、统计等操作。比如某遥测帧中的遥测参数就可能包含 8 位无符号数、16 位无符号数、32 位单精度及 64 位双精度浮点数等多种数据类型。

综上所述, 在使用 FPGA 进行遥测参数处理时, 会遇到以下两方面的难点:

1) 遥测参数的提取涉及到复杂的搜索、判断、校验过程, 如果使用 RTL 级代码实现, 其描述工作状态的状态机将异常复杂, 相关代码量也将非常大, 不利于代码的维护和调试。

2) 遥测参数的计算会经常遇到浮点计算、数据类型转换以及复杂公式计算(如正弦、对数计算等), 使用 RTL 级代码进行描述较为困难。

针对以上问题, 本文将 FPGA 高层次综合技术引入到遥测数据处理过程中。利用高层次综合技术对遥测数据进行处理具有以下优势:

1) 开发周期短。应用高层次综合技术编写的遥测数据处理相关代码属于行为级描述, 抽象层级较高, 比较符合高级语言的编程习惯。相对于 RTL 级的实现方式, 其代码精简易懂, 也便于修改, 因此可以有效缩短产品的开发周期。

2) 实时性高。相对于 DSP 或通用 CPU 等软件实现方式, 应用高层次综合技术编写的遥测数据处理相关代码仍属于硬件实现, 延时可控且可以多模块并行处理, 实时性高。

3) 支持浮点计算。高层次综合技术支持 IEEE754 标准的浮点数据计算^[7-10], 且支持标准 32 位单精度及 64 位双精度浮点数据类型, 因此非常适合描述复杂遥测参数计算。

2 基于高层次综合技术的遥测数据处理

2.1 高层次综合技术介绍

高层次综合 (High-Level Synthesis) 并不是最近才发

展起来的技术，早在 20 世纪 70 年代就提出了相关的概念，随着软、硬件技术的不断发展，高层次综合的相关技术才逐步完善、成熟^[14]。

对于高层次综合技术的应用，目前国外的相关报道较多，比如基于 FPGA 或 GPU 的计算加速^[13]、图像处理^[11-12]、算法开发^[17-18]等。尤其在异构计算方面，利用高层次综合技术能够有效提升异构计算平台的设计、开发效率^[20-21]。国内目前高层次综合技术相关报道主要集中于图像处理及算法开发上，在卫星测控领域的应用还较少。

简单来说，高层次综合是一种代码的综合技术，其作用是将抽象层级较高的语言（如 C、C++、SystemC、OpenCL 等）编写的行为级描述代码自动转换为 RTL 级代码。而利用高层次综合技术的 FPGA 开发工具^[15-16]（如 Xilinx 公司的 VIVADO HLS），除了提供高层次综合的功能外，还提供了如寄存器分配、I/O 接口生成、操作调度、状态机生成等诸多功能，有效提高了 FPGA 设计、开发效率。

利用高层次综合工具进行 FPGA 设计开发，可以带来以下几方面的优势^[19]：

1) 开发周期短。高层次综合技术可以屏蔽很多的硬件具体实现细节，使设计师将更多精力放在算法设计、产品功能描述上，从而大大缩短开发周期^[5]。根据统计，相对于手工编写 RTL 级代码，采用高层次综合技术的代码设计效率可以提高约 4 倍~10 倍，结果质量（quality of results, QoR）约 0.7 倍~1.2 倍。

2) 测试激励开发效率高。对于 FPGA 的设计、开发，测试激励（Testbench）的编写非常重要且非常耗时，而高层次综合技术可以将基于高级语言（如 C/C++）的测试激励直接用于 RTL 级代码的功能验证，这就有效提高了代码验证的效率。

3) 便于代码的修改和维护。由于基于高层次综合技术编写的代码是属于行为级的描述，其抽象层级较高，便于设计师的阅读理解，也便于代码的后期维护。

基于高层次综合技术的特点，其非常适合应用于卫星、火箭等飞行器的遥测数据处理过程中（如遥测参数的提取、计算、重组等过程）。

2.2 遥测数据处理模块的对外接口设计

为了实现对遥测参数的处理，首先需要对遥测数据处理模块的对外接口进行定义。这里使用的开发平台为 Xilinx 公司的 VIVADO HLS^[3-4,8]，且使用 C 语言对遥测数据处理模块的功能及接口进行描述。

假设所有遥测数据均以 1Byte 为单位存储于遥测帧中，而遥测帧是一种 $M \times N$ 的二维表格，经过遥测数据处理模块的处理后会生成一个长度为 L 的一维数据帧输出给后续处理模块。遥测数据处理模块的对外接口定义如图 3 所示。

其中 input_data 为 unsigned char 类型的二维数组，用于存放接收到的遥测帧；output_data 为 unsigned char 类型的一维数组，用于存放处理后的数据帧；tm_process_

```

#define TM_IN_ROW M
#define TM_IN_COLUMN N
#define TM_OUT_LENGTH L
unsigned char tm_process_block(
    unsigned char input_data[TM_IN_ROW][TM_IN_COLUMN],
    unsigned char output_data[TM_OUT_LENGTH])
:
:
:

```

图 3 遥测数据处理模块对外接口定义

block 为该遥测数据处理模块函数名，经过 VIVADO HLS 综合后会转换为相应的 RTL 模块。

由于遥测帧内部数据是按照地址的先后顺序输入到遥测处理模块的，不存在随机访问情况，输出结果也是如此，因此可以将模块的对外接口理解为一个 FIFO，而在 VIVADO HLS 中可以使用约束语句对其进行约束。

为了使 VIVADO HLS 能够正确识别对外接口的输入输出关系和具体接口形式，需要在代码中添加相应的约束语句，具体如图 4 所示。

```

set_directive_interface -mode ap_ctrl_chain "tm_process_block"
set_directive_interface -mode ap_fifo "tm_process_block" input_data
set_directive_interface -mode ap_fifo "tm_process_block" output_data

```

图 4 遥测数据处理模块的外接口约束

其中，第一条约束语句用于指导 VIVADO HLS 为遥测数据处理模块添加模块级握手信号。模块级握手信号用于控制和监测模块的工作状态，以便于不同模块之间的数据交换；第二、三条约束语句用于通知 VIVADO HLS 对 input_data、output_data 采用 FIFO 接口形式。

图 5 为经过 VIVADO HLS 综合后生成的遥测数据处理模块的对外接口，可以看出主要分为三部分：输入接口、输出接口以及相关的握手信号。

RTL Ports	Dir	Bits	Protocol	Source Object	C Type
ap_clk	in	1	ap_ctrl_chain	tm_process_block	return value
ap_rst	in	1	ap_ctrl_chain	tm_process_block	return value
ap_start	in	1	ap_ctrl_chain	tm_process_block	return value
ap_done	out	1	ap_ctrl_chain	tm_process_block	return value
ap_continue	in	1	ap_ctrl_chain	tm_process_block	return value
ap_idle	out	1	ap_ctrl_chain	tm_process_block	return value
ap_ready	out	1	ap_ctrl_chain	tm_process_block	return value
ap_return	out	8	ap_ctrl_chain	tm_process_block	return value
input_data_dout	in	8	ap_fifo	input_data	pointer
input_data_empty_n	in	1	ap_fifo	input_data	pointer
input_data_read	out	1	ap_fifo	input_data	pointer
output_data_din	out	8	ap_fifo	output_data	pointer
output_data_full_n	in	1	ap_fifo	output_data	pointer
output_data_write	out	1	ap_fifo	output_data	pointer

图 5 遥测数据处理模块的硬件接口

2.3 基于高层次综合技术的遥测数据处理的设计与实现

前文简要介绍了对卫星、火箭等飞行器的遥测数据的处理过程，而使用高层次综合技术对遥测数据进行处理，主要的技术难点在于如何使用高级语言描述遥测数据的数据结构，以及如何使用高级语言描述遥测数据的处理过程。

从前面的分析可以看出, 遥测数据的数据结构可以由高级语言的一维或二维数组描述, 并且可以很方便地对数组中的指定数据进行寻址访问, 而遥测数据的处理过程可以使用高级语言的循环、条件判断等常用的语句描述, 其工作流程如图 6 所示。具体描述如下:

1) 对 input_data、output_data 以及其他用于存放中间结果的临时数据缓冲区进行初始化, 可以使用 For 循环完成。如输出结果中还要包含帧计数功能, 则初始化帧计数相关变量。

2) 将 input_data 中的遥测帧数据拷贝到临时数据缓冲区中。由于遥测帧是二维结构, 存放遥测帧的缓冲区可以声明为 unsigned char 类型的二维数组, 并使用二级 For 循环实现数据拷贝。

3) 校验遥测帧内部数据 (如奇偶校验、CRC 校验), 如果校验正确则进行下一步处理, 如果校验不正确则返回 FALSE, 表明本次遥测帧处理失败。

4) 遥测参数提取。该过程实际上就是对二维数组的访问过程, 将二维数组中指定遥测数据拷贝到临时数据缓冲区, 或按照一定规则在二维数组中指定区域搜索指定的遥测参数集再进行进一步处理。

5) 遥测参数计算。对提取出的遥测参数进行计算, 并将计算结果保存到临时缓冲区。

6) 将遥测参数计算结果保存到临时缓冲区, 并根据一定的规则重新组帧, 同时将计算结果填入帧内。如输出结果中需要有帧计数功能, 则将帧计数变量加 1 并填入帧内相应位置。

7) 根据一定规则计算输出帧的校验和并填入输出帧的相应位置。

8) 将临时缓冲区中的输出帧拷贝到 output_data 中, 该过程可用 For 循环实现, 最后返回 TRUE, 表明本次遥测帧处理成功。

遥测地面站在输出处理结果时经常需要标记当前输出帧的帧序号, 以便于后端设备根据帧序号进行通信握手或误码统计, 这就要用到帧计数功能。帧计数变量需要在不同的遥测帧处理过程中持续保持更新, 而不像其他变量那样每次处理过程都置零或恢复初始状态, 因此帧计数变量需要设置为 static 类型, 这样帧计数变量就可以保持上一次调用遥测数据处理模块时的状态。

对于浮点计算, VIVADO HLS 支持标准 IEEE754 浮点数 (32 位单精度及 64 位双精度浮点数据类型) 及相关浮点计算。因此使用高层次综合技术实现遥测参数的浮点计算非常简洁、高效。以标准 32 位单精度浮点数为例, 配合 C 语言的强制类型转换及位操作便可以实现复杂的浮点计算。

这里假设提取遥测帧中的 11 行 85 列的遥测参数 (unsigned char 类型), 将该参数进行浮点计算 (float 类型) 并将计算结果存储到输出帧中的 [10: 13] 位置, 其实现代码如图 7 所示。

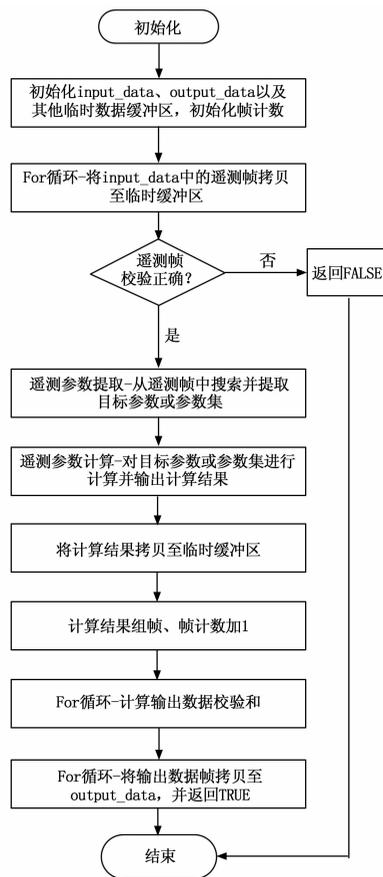


图 6 遥测数据处理模块工作流程图

```

unsigned char temp_u8_data;
float temp_float_data;
int * temp_s16_point;
int temp_s16_data;
:
:
temp_u8_data = tm_array[11][85];
temp_float_data = (float)temp_u8_data*5.0f/255.0f;
temp_s16_point = (int*)&temp_float_data;
temp_s16_data = *temp_s16_point;
out_array[10] = (unsigned char)( temp_s16_data & 0x000000ff);
out_array[11] = (unsigned char)( temp_s16_data & 0x0000ff00)>>8);
out_array[12] = (unsigned char)( temp_s16_data & 0x00ff0000)>>16);
out_array[13] = (unsigned char)( temp_s16_data & 0xff000000)>>24);
    
```

图 7 浮点计算实现代码示例

可以看出基于高层次综合技术的浮点计算代码非常简洁, 遥测数据的寻址、浮点运算等只需一条语句描述, 将计算结果存储到输出帧中也仅需要简单的移位、强制类型转换等操作完成, 而如使用 RTL 方式实现相同功能, 则代码量将大大增加。

3 测试结果与分析

本文将以上描述的遥测数据处理模块使用 VIVADO HLS 进行了设计、仿真, 同时将转换的 RTL 代码进行了硬件测试。代码在 VIVADO 开发环境^[9]中进行综合、布局、布线, 最后在 Xilinx 公司的 XC7A200T^[6]平台上进行了硬件测试。

硬件测试过程中, 遥测数据处理模块对 96 Byte * 64

Byte 的遥测帧进行处理, 输出帧长度为 360 Byte。处理过程涉及到若干次的参数集提取、浮点计算以及校验和计算等操作, 模块工作时钟为 10 MHz。

图 8 为该模块的 FPGA 资源占用情况, LUT 及 DSP 资源占用率分别为 5% 和 1%。可以看出其 FPGA 资源占用率是非常低的。

Utilization Estimates

- Summary

Name	BRAM_18K	DSP48E	FF	LUT
DSP	-	-	-	-
Expression	-	-	0	3238
FIFO	-	-	-	-
Instance	-	8	817	1883
Memory	7	-	0	0
Multiplexer	-	-	-	2503
Register	-	-	849	-
Total	7	8	1666	7624
Available	730	740	269200	129000
Utilization (%)	~0	1	~0	5

图 8 遥测数据处理模块 FPGA 资源占用情况

图 9 是遥测数据处理模块的时钟及延时情况。可以看出, 该模块最高工作频率可以达到 11.2 MHz, 延时最高为 24 367 个时钟周期, 约 2.4 ms, 因此理论上达到 400 帧/s 的处理速度。由于遥测数据处理模块使用的是 FPGA 的基本逻辑资源, 因此也可以多个模块并行处理遥测数据, 以适应更高速的遥测数据处理需求。

Performance Estimates

- Timing(ns)

- Summary

Clock Target	Estimated	Urcertainty
ap_clk	100.00	88.949 1.00

- Latency(clock cycles)

- Summary

Latency		Interval		Type
min	max	min	max	
6836	24367	6836	24367	rone

图 9 遥测数据处理模块时钟及延时情况

图 10 是遥测数据处理模块接口部分的 RTL 级仿真图。可以看出, 当遥测帧输入到模块后, 经过一定的处理周期, 模块输出处理结果, RTL 级仿真结果与行为级仿真一致。

从开发周期上进行对比, 应用高层次综合技术, 整个模块从代码设计、仿真直到最后的硬件测试仅仅耗费了 1 周的时间, 而如果使用传统的 RTL 开发流程, 保守估计预计要耗费 2 个月的时间。

从代码量上进行对比, 由于模块是基于 C 语言描述的, 其抽象级别较高 (行为级描述), 因此相关代码极为精简, 模块主体代码量仅 300 行左右, 加上 Testbench 的代码量为 500 行左右。与之对比, 如果采用基于 RTL 的设计流程,

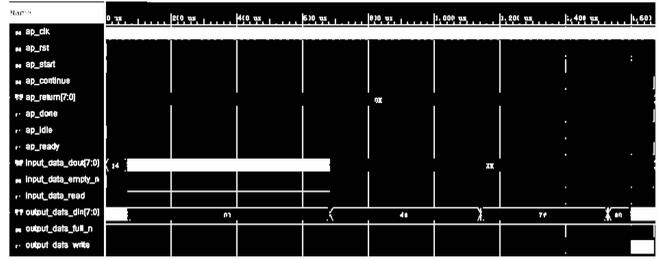


图 10 遥测数据处理模块接口部分 RTL 仿真图

整个设计的代码量预计要增加 10 倍以上。

4 结束语

综上所述, 使用高层次综合技术对遥测数据进行处理, 在同样满足设计要求的情况下, 其设计、开发效率要远远高于传统的基于 RTL 的开发流程, 同时也在很大程度上减轻了设计工程师的工作量。

目前高层次综合技术主要应用在图像处理、数据编解码^[11-12]等领域, 而在卫星测控领域的应用报道还很少。本文将高层次综合技术应用到地面站的遥测数据处理过程中, 证明了利用高层次综合技术可以有效提高卫星测控产品相关 FPGA 的设计、开发效率, 缩短产品的研制周期, 因此具有推广价值和参考价值。

参考文献:

- [1] 严平. FPGA 的实时 PCM 遥测数据传输系统设计 [J]. 单片机与嵌入式系统应用, 2018 (1): 38-40.
- [2] 陈建行, 等. 某机载遥测原始数据处理软件设计与实现 [J]. 微处理机, 2018, 12 (6): 34-37.
- [3] UG902—Vivado Design Suit User Guide: High-Level Synthesis (v2019.1) [Z]. Xilinx, 2019.
- [4] 党宏社, 等. 基于 Vivado HLS 的 FPGA 开发与应用研究 [J]. 陕西科技大学学报, 2015, 33 (1): 1155-1159.
- [5] 孟祥刚, 等. FFT 算法硬件模块的高层次综合实现与优化 [J]. 微电子学, 2017, 47 (2): 217-221.
- [6] DS180-7 Series FPGAs Data Sheet—Overview (v2.6) [Z]. Xilinx, 2018.
- [7] James Hrica. 利用赛灵思 Vivado HLS 实现浮点设计 [J]. 今日电子, 2013 (1): 34-38.
- [8] UG871—Vivado Design Suite Tutorial High-Level Synthesis (v2019.1) [Z]. Xilinx, 2019.
- [9] UG893—Vivado Design Suite User Guide—Using the Vivado IDE (v2019.1) [Z]. Xilinx, 2019.
- [10] James Hrica. XAPP599—Floating-Point Design with Vivado HLS (v1.0) [Z]. Xilinx, 2012: 4-6.
- [11] 林振钰, 等. 基于 ZYNQ 的高清图像显示及检测系统设计 [J]. 计算机测量与控制, 2021, 29 (2): 30-34.
- [12] 王林, 等. 基于 HLS 协议视频监控加密系统优化设计与实现 [J]. 计算机测量与控制, 2019, 27 (7): 169-173.

(下转第 219 页)