



2019 年 SEU-Xilinx 国际暑期学校团队项目设计文档

(Project Paper Submission Template)

作品名称	<u>SDAceel 环境下的 CRC 算法加速</u>
组员姓名	姚汉梅、朱泳波、陆志成、刘中元
房间号及桌号	715 第三组



第一部分

小组成员分工

(各成员任务分配)

	姓名	任务分配
组长	朱泳波	SDAccel 环境下 build 项目
组员 1	陆志成	CRC 算法的实现
组员 2	刘中元	CRC 算法的 IP 核设计
组员 3	姚汉梅	尝试使用亚马逊云，查阅资料

第二部分

设计概述 /Design Introduction

1. 功能说明

该项目属于使用 FPGA 进行算法加速的领域，对文件进行加解密运算、校验等任务，CRC 校验在网络、存储领域都有广泛应用，对实时性有较高要求。该项目选取一种加密或校验算法进行实验和仿真，并在 SDAccel 开发环境中运行，实现算法的 FPGA 云加速。

2. 所用到的设备清单

PC

第三部分

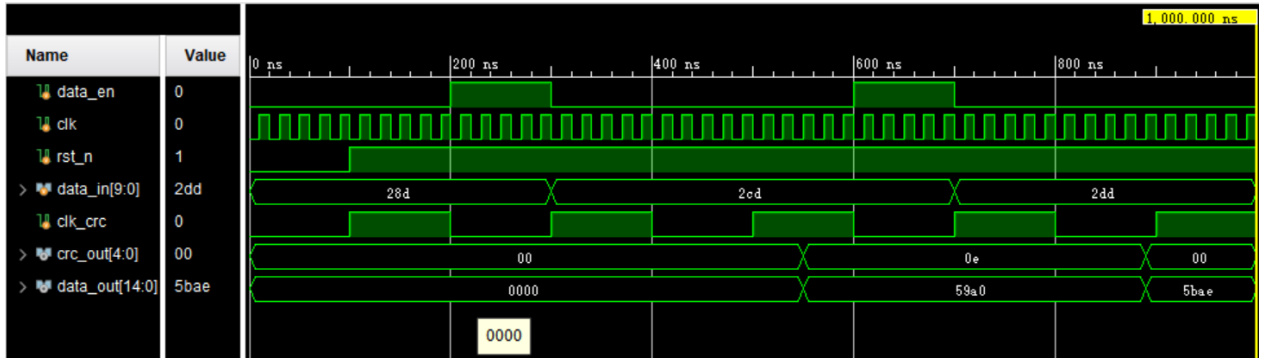
详细设计 /Detailed Design

(请详细说明你作品要实现的所有功能以及如何组建系统以实现该功能，还包括为实现该功能需要用到的所有参数和所有操作的详细说明，必要的地方多用图表形式表述)

如因文档篇幅限制无法详述，可提供附件。

计划功能为以一个同时具备 ARM 和 FPGA 的片上系统，结合 ARM 开发和 FPGA 开发的优势实现 CRC 的算法加速。原计划采用 SDAccel 云加速平台进行一站式加速效果检测，直接在该平台上显示加速效果，形成与只采用 ARM 开发的对比，以彰显 FPGA 与 ARM 组合开发的优势。但因为云平台账号未能成功申请，又改为在本地平台上进行试验，须采用 SDx 软件进行开发。而该软件必须在 Linux 环境下运行，所以在试验设备上安装了虚拟机进行试验。

在编写可以实现 CRC 算法的 verilog 代码之后进行了仿真，其效果图如下：



可以看到在输入信号为 12'h2dd, 输出信号为 16'h5bae。实现了 CRC 算法, CRC 算法的运算公式如下:

```
1010, 1101, 0011, 1001
1001 ) 0000, 0110, 0001, 0111, 0000, 0010
      100, 1
      ----
        1, 100
        1, 001
        ----
          111, 0
          100, 1
          ----
            10, 11
            10, 01
            ----
              1011
              1001
              ----
                10, 000
                1, 001
                ----
                  1110
                  1001
                  ----
                    101, 0
                    100, 1
                    ----
                      1010
                      1001
                      ----
                        0001
```

接下来配 SDAccel 环境。

- 1、首先需要有 linux 的环境, 我用的是虚拟机上的 Ubuntu 系统。
- 2、安装 SDAccel 开发环境, 再把 xrt, deployment shell, development shell 都装一下(xilinx 官网都有)。
- 3、在 source 环境下打开 sdx 进行开发: 先根据 RTL kernel wizard 生成 xml 文件, 再在 vivado 中完成 RTL kernel 的生成, 此时会在 SDx 中生成 xo 文件, 设置好用于 PL 和 PS 通信的 cpp 文件后 build 项目, 选择 simulation



HW，便可以生成一个可直接运行的安装包。Cpp 文件的接口如下：

- A handle to the kernel is created (line 239).

```
clCreateKernel(program, "vadd_A_B", &err);
```

- Buffers are created to transfer data back and forth between the host and the FPGA (line 256).

```
clCreateBuffer(context, CL_MEM_READ_WRITE, sizeof(int) * number_of_words, NULL, NULL);
```

- Values (A and B) are written into the buffers, and the buffers transferred to the FPGA (lines 266 and 274).

```
clEnqueueWriteBuffer(commands, dev_mem_ptr, CL_TRUE, 0, sizeof(int) * number_of_words,  
host_mem_ptr, 0, NULL, NULL);
```

- After A and B have been transferred to the device, the kernel can be executed (line 299).

```
clEnqueueTask(command_queue, kernel, 0, NULL, NULL);
```

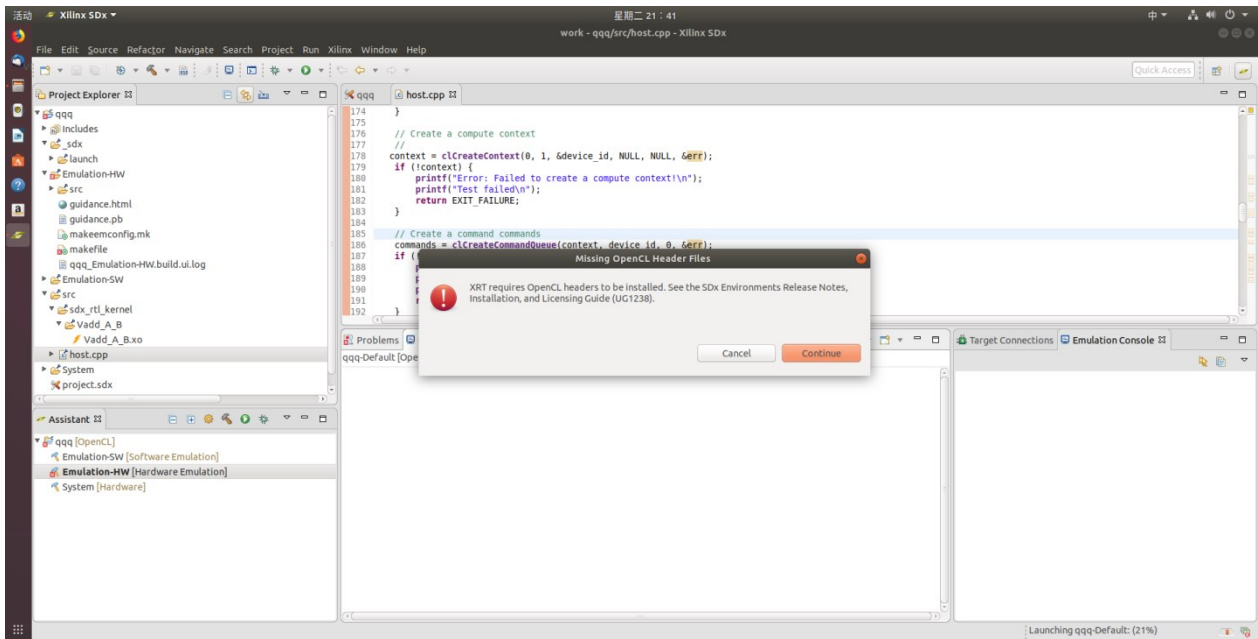
- After the kernel completes, the host application reads back the buffer with the new value of B (line 312).

```
clEnqueueReadBuffer(command_queue, dev_mem_ptr, CL_TRUE, 0,  
sizeof(int)*number_of_words, host_mem_output_ptr, 0, NULL, &readevent );
```

第四部分

完成情况及性能参数 /Final Design & Performance Parameters

最后 build 时报错，改成中文路径仍未解决



第五部分

项目总结 /Conclusions

主要学习了：

CRC 算法的实现与仿真

SDAccel 环境的初步使用

SDAccel 环境下 PL 与 PS 间的通信接口

第六部分

源代码/Source code

(作品源码 Github 链接, github 操作步骤见 *Github 项目上传简单教程*)

<https://github.com/jojo-oop/SDAccel>

第七部分 (技术总结/心得/笔记等分享)

<https://mp.csdn.net/mdeditor/96426726#>