

國立臺灣大學電子工程學研究所

碩士論文

Graduate Institute of Electronic Engineering

National Taiwan University

Master Thesis

支援 Xilinx AXI DMA 的 Linux UIO 驅動程式

Linux UIO driver for Xilinx AXI DMA

劉宇唐

Yu-Tang Liu

指導教授：鄭振牟 博士

Advisor: Chen-Mou Cheng, Ph.D.

中華民國 107 年 7 月

July 2018

摘要

近年來，由於 AI、VR 產業的崛起，FPGA 產業越來越受到重視。為了簡化 FPGA 的開發流程，使用嵌入式 Linux 會是一個不錯的方法。透過 Linux Kernel 提供的 UIO 驅動程式，我們可以把我們在硬體端設計出來的 IP 視為一個外部裝置，然後在 Linux 使用者空間裡的程式中，輕鬆地開發軟體端的應用。然而，有些硬體端的設計，卻無法透過同樣的方法，利用 UIO 驅動程式，建立裝置節點，而帶有直接記憶體存取 IP 的設計就是其中之一。由於 UIO 驅動程式並無法支援此種設計，我們必須擁有”root”權限，才能使用我們的設計，但是提供”root”給一般使用者並不是一個好方法。在此論文中，我們修改了 Linux 內建的 UIO 驅動程式，使得一般用戶也能在使用者空間中使用帶有 DMA 的硬體設計。

關鍵字: 賽靈思，直接記憶體存取，AXI，Linux UIO 驅動程式

Abstract

In recent year, increasingly importance has been attached to FPGAs with the development of AI,VR. To simplify the development process on FPGAs, embedded Linux on FPGAs will be a good way. With UIO driver provided in Linux Kernel, we can mount our block design , that is, custom IP(Intellectual Property) core in Vivado as a device node, and program it in Linux user space. However, there are some designs that UIO driver cannot recognizes. The design with DMAs(Direct Memory Access) is the one of them. With this kind of design, because UIO driver is not work, we need "root" to controll our IP, and providing root privileges to users is never a good solution. In this thesis, we modify UIO driver so that users can easily use designs with DMA in user-space.

Keywords: *Xilinx, DMA, AXI, Linux UIO driver*

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Contribution	1
2	Preliminaries	2
2.1	Embedded Linux	2
2.1.1	Device Tree	2
2.1.2	Linux Kernel Driver	2
2.2	UIO Driver	2
2.3	AXI Bus	2
2.3.1	AXI Stream	2
2.4	DMA	2
2.4.1	DMA Engine	2
3	Proposed solution and evaluation	3
3.1	Motivation	3
3.2	Contribution	3
4	Environment Framework	4
5	Analysis	5
5.1	FIFO	5
5.2	Stream IP	5
5.3	Comparison	5

6 Conclusion	6
6.1 Motivation	6
6.2 Contribution	6
References	7

List of Figures

List of Tables

Chapter 1

Introduction

FPGA(Field Programmable Gate Array) is

1.1 Motivation

1.2 Contribution

Chapter 2

Preliminaries

In this chapter, we introduce the background technology for our work.

2.1 Embedded Linux

2.1.1 Device Tree

2.1.2 Linux Kernel Driver

2.2 UIO Driver

2.3 AXI Bus

2.3.1 AXI Stream

2.4 DMA

2.4.1 DMA Engine

m

Chapter 3

Proposed solution and evaluation

3.1 Motivation

3.2 Contribution

Chapter 4

Environment Framework

Chapter 5

Analysis

5.1 FIFO

5.2 Stream IP

5.3 Comparison

Chapter 6

Conclusion

6.1 Motivation

6.2 Contribution

References