

## Introduction

This document briefly describe the complete CXD9731 chip project and make people easier to adapt this chip for other uses.

The project starts from some request from certain second hand console dealers and repairman from China that they have Sony Playstation 2 console but the DVD / CD laser head is broken and they don't have parts for repair.

To use the Playstation 2 console without CDROM laser head, the other option is using a network card and an IDE harddisk to store programs, with modification in BIOS to boot from the network card.

However, the network adaptor has been out of production for a long time and it is rare to find one, so we need to reverse engineer that network adaptor IC ( CXD9731 ) and look for a viable solution to interface harddisk to Playstation 2 console.

As we don't have any data sheets of that CXD9731IC, we use a logic analyzer to hook up the two interface ports ( 110 pin port to 40 pin IDE port ) and read all transactions to work out the function how Playstation 2 interacts with a harddisk.

Since real networking is not necessary to play games, we didn't work on the RJ45 network part, just the IDE harddisk interface will be sufficient.

## Project Structure

This project was developed using Xilinx Spartan 3 FPGA, an XC3S50A device is good enough ( the real CXD9731 has six blocks of RAM for buffer, however, XC3C50 has resources for only 2 RAM blocks, lucky the handshake signals works well and 2 RAM blocks are OK ).

The development tool used was ISE by Xilinx. The FPGA constraint file is chip.ucf and the top is chip.v

Top.v and TF\_Stub is for testing. Also there is a testbench01.v to simulate the complete circuit.

Special - defend from unlicensed production or design leaking by outsource manufacturer

Xilinx spartan 3 FPGA has a " DNA " port, which is a serial register internal and that ID is unique for each individual device.

The CheckDNA.v module utilize this feature and matches the SPI configuration data, so each configuration data can only drive that particular FPGA.

A JAVA server was setup which when receive a string of FPGA DNA ID, will generate a key-file for that ID and send back to the JAVA client.

The manufacturer was given a base binary configuration code, a JAVA client, Xilinx programming tools with software and a script.

The manufacturer setup a programming station with internet connection, where he puts unit to program.

The script will read that FPGA's unique DNA code and sent this code to a server control by the designer.

The server will calculate a key base on the DNA code and sent back the key-file to the manufacturer programing station.

The station will then combine the key file with the base binary configuration data and write into the SPI configuration EEPROM.

Since the base binary configuration data contains the key comparison algorithm in hardware form, the manufacturer will not know the maths relationship between the DNA code and key-file.

At power-up, each FPGA will checks its own DNA code with the server released key file, if check verified, it can proceed.

As the key generation routine is in the server, which can be located in other parts of the world, designer can control the number of units the factory is producing by counting how many different FPGA DNA ID the manufacturer has sent to the server.

Also even if all parts are stolen from the manufacturer plant, without new key-files, no one can manufacture unlicensed code.

## Conclusion

This project has the same LGPL license and free to modify and distribute for non-commercial use.

You must also include this file in its unaltered form and the whole svn with modification history ( if any modifications has been made ) to entitle you to use this project.

For credits, encouragement or questions, please write to me at [revivechip@gmail.com](mailto:revivechip@gmail.com)

Thanks for your interest in this project and have a good day

Simon Fung

[revivechip@gmail.com](mailto:revivechip@gmail.com)