ISSN: 2664-9640

Design of High-Performance Storage System Based on Zynq

Yiming Liu, Qingsong Lin, Wensheng Li and Shuai Zhang

Henan University of Science and Technology, School of Information Engineering, Luoyang 471000, China;

Abstract

Aiming at the practical problem of high-speed data stream storage in airborne test, and combining with the system requirements of miniaturization, low power consumption and large capacity, a high-performance storage system design based on Zynq was proposed. The system selects the heterogeneous multi-core chip (XC7Z015) from the Zynq-7000 series launched by XLINX Company as the host core, and uses the SSD based on the M.2 Socket-3 interface as the storage medium.In the paper, the PCIe data interface is designed in the PL to drive the SSD, and then the cross-compilation environment is established in the PS. The SD card startup file is completed by transplanting the Petalinux system.The final experimental results show that the XC7Z015 can recognize the SSD as its own storage device. The average read/write rate of a single disk is 125MB/s, and the chip power consumption is only 2.21W. Compared with the third-party storage system of IP core, the research and development cost on system is greatly reduced. Therefore, with a high practical value, the design provides a useful reference for the future storage solutions with higher speed and larger capacity.

Keywords

Zynq; SSD ;High-performance ;PL ;PS ; Petalinux .

1. Introduction

Commonly, the traditional airborne test storage scheme adopts the architecture of a high-speed acquisition card and a high-performance industrial computer [1], which greatly increases the system complexity and has a large volume when processing the storage task of multiple groups of high-speed data streams. With the rapid development of integrated circuit technology, high speed data storage products with the core of general embedded processor are gradually emerging. The embedded acquisition-storage scheme gets rid of the dependence on PC, and realizes the front-end video acquisition and the direct management of the storage medium by the embedded processor, which is in line with the development trend of miniaturization, lightweight and customization of airborne test equipment [2].

The performance of storage medium directly affects whether the system can meet the storage requirements of high speed and large capacity. In addition to meeting the requirements of storage rate and storage capacity, it is also necessary to consider the impact of harsh application environment such as high-low temperature and strong vibration on the stability of the system. With the continuous progress of storage medium, storage interfaces and protocols are also being updated. Based on PCIE channel, the NVMe SSD breaks through the rate of 6GPS in the traditional SATA protocol [3]. More importantly, the feasibility of establishing communication between the core control devices of the system and the SSD interface should be considered during the demonstration period of the overall design. Meanwhile, the possibility that high licensing fee of storage IP may increase R&D costs should be avoided.

Xilinx is the world's largest supplier of FPGA solutions [4]. Zynq-7000 series, as its key product to enter the field of heterogeneous multi-core architecture, selects the single chip architecture

ISSN: 2664-9640

combining ARM and FPGA. The paper summarizes the chip resources, I/O interface, PCIE channel, operating temperature range, and IP core resources that can be supported. Finally, the XC7Z015-2CLG485I chip is selected. Integrating a dual-core ARM Cortex-A9 processor with a clock speed of 866MHz and Artix-7 series FPGA with a logic resource of 74K, the chip constitutes a programmable on-chip solution of PS (Processing System) and PL (Programmable Logic). The PCIe integrated interface module provided by the PL creates the connection channel for the SSD, and the ARM processor at the PS can run a complete Linux system.

2. Overall scheme design

The system is designed to meet the standard of which the data read-write rate should not be less than 85MB/s,and the storage capacity is not less than 100GB. This system constitutes a high-performance storage system by using a multi-core heterogeneous platform.The overall scheme is shown in the Figure 1 below.

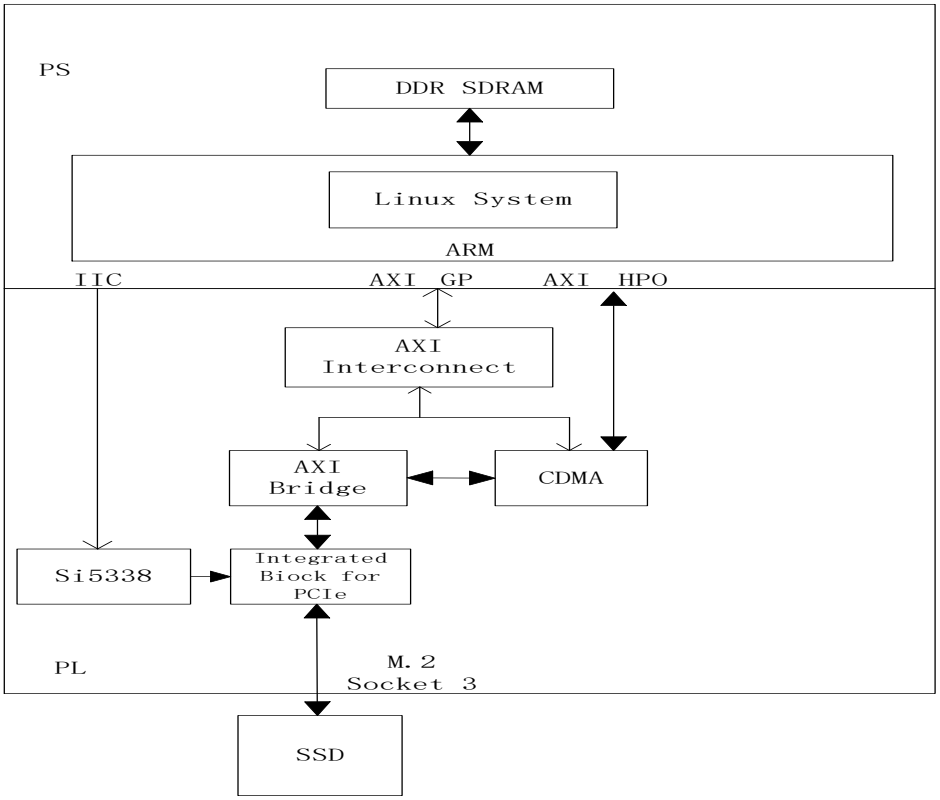

Fig. 1 Overall scheme design

Following the development process of Zynq, the paper divides the system design into two parts, including the firmware logic design in PL and the embedded Petalinux system transplantation in PS . Firstly, the configuration of Zynq and the hardware logic design of PCIe IP are completed by Vivado 2018.4 in PL, and the generated bitstream files are exported to SDK. Secondly, the cross-compilation environment is built in the VM to transplant the Petalinux system, and the necessary files for the startup mode of Zynq SD card are completed. At last, Mabaxterm, a serial port terminal tool, is used to test the performance of the system.

3. PL logic design

3.1. Zynq IP configuration

In the firmware logic design, the Zynq module needs to be customized according to the functional requirements of the design. Create a new project in Vivado, select XC7Z015CLG485-2 as the core device, and add the Zynq core module to the Block Design. Considering that data

ISSN: 2664-9640

needs to be cached in DDR3-SDRAM, the choice of high-performance interface HP and host interface GP0 in PS-PL configuration interface ensures that Zynq can carry out high-speed data transmission with DDR3-SDRAM and access PL peripherals at a high speed. For MIO directly connected with PS, the peripheral required in the design is SD card and UART communication interface, and the power supply voltage corresponding to VCCO_MIO0/1 is set as LVCMO_1.8V. Referring to the official PCIe IP design of Xilinx, the FCLK_CLK0 clock provided by PL is disabled, the 100MHz reference clock provided by AXI PCIe IP is adopted, and the severance from PL to PS is enabled. The designed Zynq module is shown in the following Figure 2 .

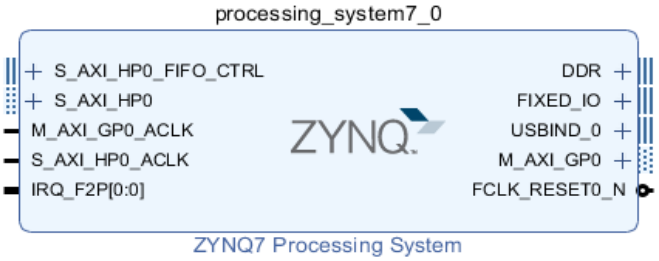

Fig. 2 Designed Zynq moudle

3.2. PCIe IP configuration

Based on the AXI PCIe IP core, the AXI4 interface with address attributes is adopted to complete the design of the PCIe IP core in the paper. The specific process is as follows:

First, the PCIE device should be configured as the Root device, and a single channel PCIe 2.0 protocol should be installed at a rate of 5 gT /s in the Link configuration option. To ensure that the system can correctly recognize M.2 SSD, the value of Class Code is set to 0x06400, the BAR space is set to 1G, and an external high-speed interface is connected to the "PCIE_7X_MGT" port of the IP core. Second, add a Constant IP, set its default value to 0, and connect its output port to the INTX_MSI_REQUEST port. At last, the Utility Buffer IP core is invoked to provide a specific 100MHz high-speed differential clock and its input is connected to the input clock of the system. The designed PCIE IP is shown in the figure 3 below.

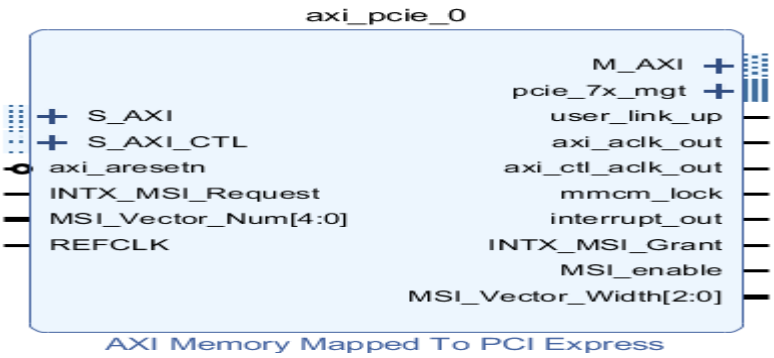

Fig. 3 Designed PCIe IP

3.3. BD design

Two output clock signals of PCIe IP, AXI_CLK_OUT and AXI_CTL_CLK_OUT require the function of Clock Reset, and two Processor System Reset IPs are invoked from the IP library. CDMA IP is added in the design to improve the throughput in the link. Then, the transmit-receive function of the IP is disabled, and the bit width of read-write data is set to 128. At last, Concat IP is invoked to accept the interrupt output of PCIe IP and the input of Zynq IP. AXI Iterconnect IP is invoked to connect the previously designed modules. The axi_0 module is used to access the two memory ports of DDR3-SDRAM, the axi_1 module is used to access the ports of the PCIe IP control interface and the CDMA control port, and the axi_2 module accesses the DDR3-SDRAM memory and the PCIe IP port as CDMA.

ISSN: 2664-9640

After the completion of the BD engineering design, the scheme of automatic address assignment needs to be improved. The address space of Zynq IP, PCIe IP and CDMA should be allocated manually. The PCIe BAR0 is set to 256MB manually and the PCIe control interface is configured to 64MB.

3.4. Resourec and power analysis

Synthesis, one of Vivado's functions, can convert RTL level designs to gate level designs while generating integrated netlist files [5]. In the resource consumption report shown in the Figure 4 below, it can be seen that the utilization rate of the basic unit LUT in the project is 47.93%. In the actual design, a high-speed serial transceiver at the PL terminal is invoked, and the utilization rate is 25%. The above utilization rates reserved sufficient resources for the subsequent iterative design. The estimation results of the chip power consumption obtained in the Report Power are shown in the following Figure 5. It can be seen from the report that the power consumption of the project is 2.21W, which is in the normal power consumption range of the chip. Among them, the power consumption of DDR-SDRAM in dynamic power consumption accounts for 81% of the total power consumption.

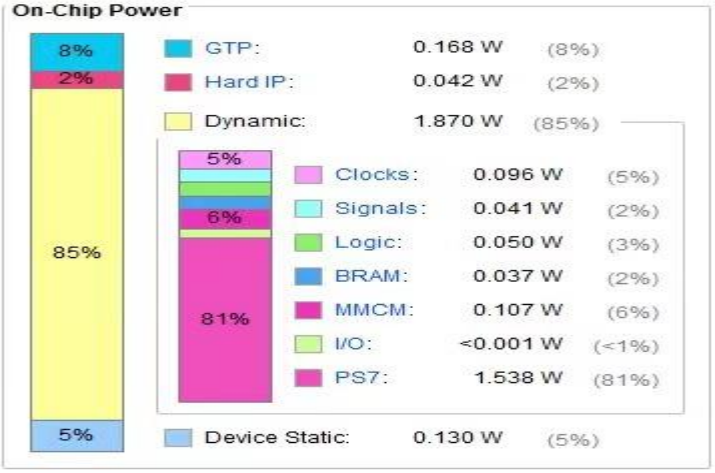

Fig. 4 Resource consumption report

Resource	Utilization	Available	Utilization %
LUT	22144	46200	47.93
LUTRAM	1056	14400	7.33
FF	22962	92400	24.85
BRAM	16	95	16.84
IO	1	150	0.67
GT	1	4	25.00
MMCM	1	3	33.33

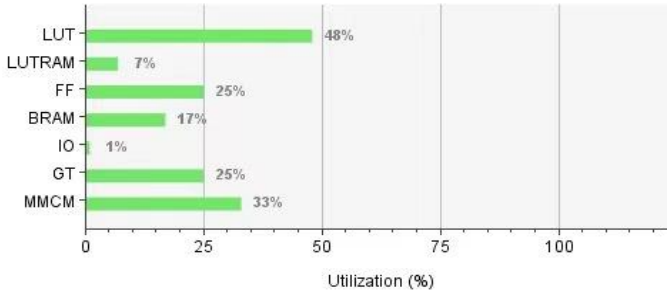

Fig. 5 Estimation results report

ISSN: 2664-9640

4. Petalinux system migration

4.1. Build cross-compilation enviroment

Xilinx offers free access to the Linux operating system for Zynq developers, as well as board-level support packages and device drivers. PetaLinux integrates a complete Linux system and development environment, but a 2GB CPU, a 100GB hard disk, and a stable Linux system are also needed to run properly. The paper simulates a real installation environment by selecting the Ubuntu16.04 version system in the VM, configuring the system installation enhancements to support VBOXSF type mounts, and installing the dependent libraries required by Ptetalinux in Root permissions, as shown in the Figure 6.

```
lym@lym-VirtualBox:~$ sudo -s
root@lym-VirtualBox:~# sudo apt-get install -y gcc git make net-tools libncurses
5-dev tftpd zlib1g-dev libssl-dev flex bison libselinux1 gnupg wget diffstat chr
path socat xterm autoconf libtool tar unzip texinfo zlib1g-dev gcc-multilib buil
d-essential libsdl1.2-dev libglib2.0-dev zlib1g:i386 screen pax gzip gawk
Reading package lists... Done
Building dependency tree
Reading state information... Done
build-essential is already the newest version (12.1ubuntu2).
diffstat is already the newest version (1.61-1).
gcc is already the newest version (4:5.3.1-1ubuntu1).
gzip is already the newest version (1.6-4ubuntu1).
libselinux1 is already the newest version (2.4-3build2).
make is already the newest version (4.1-6).
net-tools is already the newest version (1.60-26ubuntu1).
```

Fig. 6 Dependent libraries

4.2. Make boot file

The first step is to create the XME7015 file in the Termianl to save the Petalinux project of the design, and import the SDK file into the project. After configuring Zynq, generate the image file of the root directory of the operating system.

As shown in the Figure 7 and Figure 8, the second step is to configure the Linux system kernel through kernel instructions, select the PCIE Bus in Bus Support, select the NVME protocol in the device number driver, and wait for the completion of the Linux system kernel compilation. At the same time, set up the root file system with the help of the rootfs directive. First, select pciutils, pciutils-ids, and libpci in the configuration window. Second, the terminal command support packages LSBLK, fdisk, MKFS and blkid should be choosed. Finally, select the system tool of EXT2 file, namely, e2fsprogs, e2fsprogs-badblocks, e2fsprogs-mke2fs, libss, libcomerr, libext2fs and libe2p.

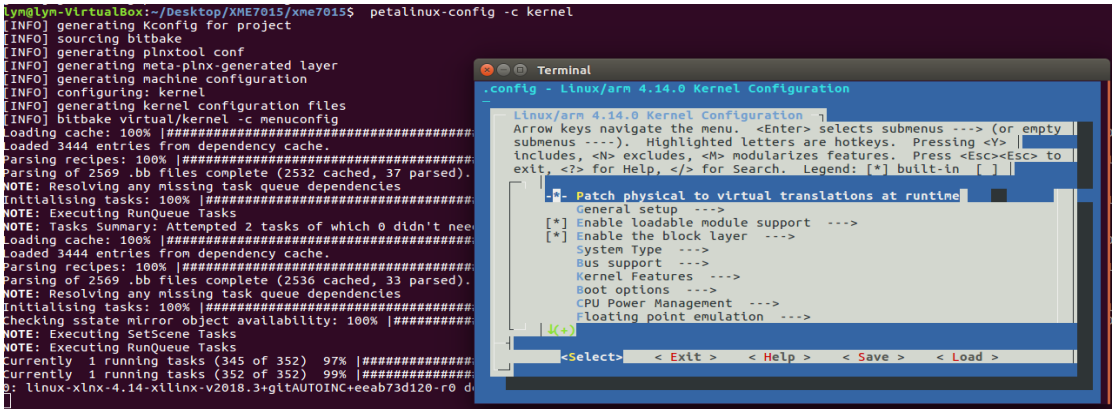

Fig. 7 Kernal configuration

ISSN: 2664-9640

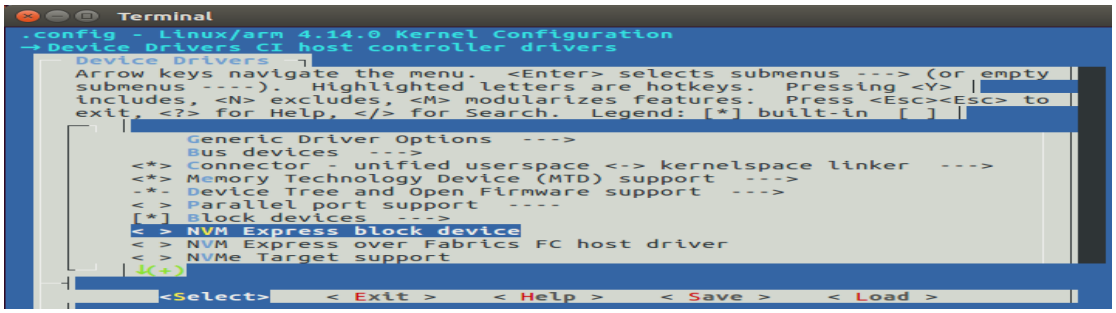

Fig. 8 Device tree configuration

The third step is to compile all the projects with the build directive. Then, package the three necessary Linux BOOT.files, boot.bin, image.ub, and system.dtb, with the package directive.

5. System overall performance test

Firstly, the overall performance test of the system needs to detect the hardware functions of each module. The hardware devices needed in the system include UART serial port, SD card interface, DDR3-SDRAM and M.2 interface. Verify them orderly, and then load the Linux startup file into the SD card of the device, and connect the corresponding power cord and serial port line.At last, the system storage rate needs to be verified. Open the terminal tool Maboxterm which can execute Linux commands in the PC, turn on the power of the device, and the device starts normally from the SD card. The device startup information are printed in the terminal window, and the Petalinux system runs successfully.

The time dd instruction is used to test the storage rate of the system. The write rate of the system is tested by the null character stream generated by the dev/zero virtual device, while the read rate is tested by the dev/null virtual device. The test results are shown in the Figure 8 below.

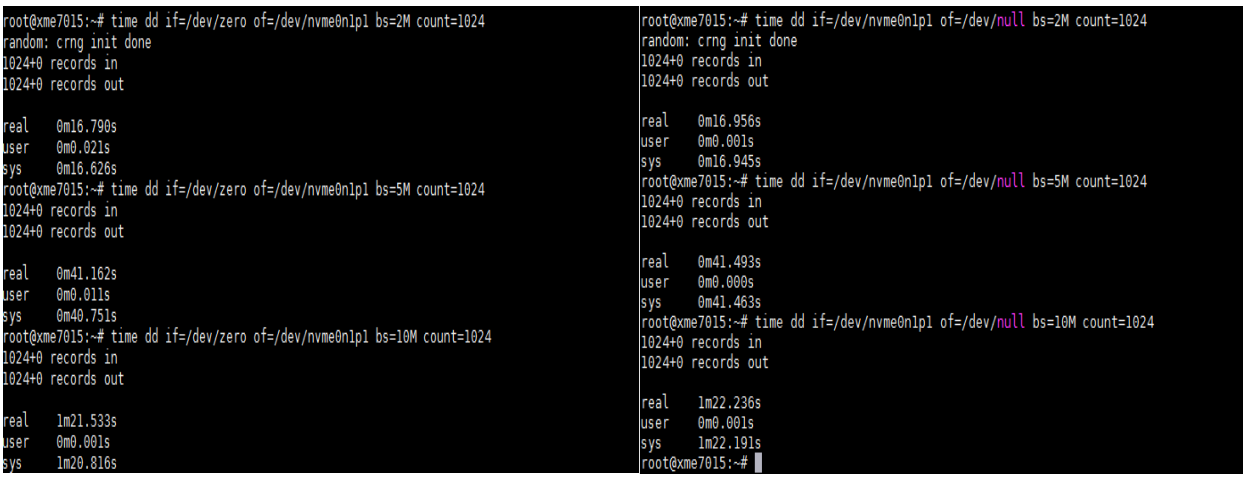

Fig. 8 The test results

The rate information returned by the serial terminal can be used to obtain the three write speeds, respectively, 121.98MB/s, 124.39MB/s and 125.59MB/s. The average write rate is 123.98MB/s. Three reading rates are performed at 120.79 Mb/s, 123.39 Mb/s and 124.51 Mb/s, with an average reading rate of 122.89 Mb/s.

6. Summary

The read-write rate of the system already meets the read-write performance index of the airborne high performance system, but does not inspire the full performance of SSD. The main factors affecting the storage rate are the core controller resources, the PCIe protocol version, the number of channels, and the AXI interface rate. Different from the storage system developed

ISSN: 2664-9640

by using expensive exclusive storage IP core, the storage system of Zynq combined with SSD is designed by transplantation of Petalinux system at a limited cost in the paper, which met the requirements of large capacity, miniaturization and low cost of airborne high-performance storage system.

References

[1]. Yang Ruigeng, Sun Fengqin, Tian Yinqiao, et al. Intelligent Auxiliary Test System Requirements of General Aircraaft Verification Flight. Measurement Control Technology. Vol. 39 (2020) No. 12, p. 126-128.
[2]. Li Shuo, Liu Fang, et al. Design and inplementation of FPGA-based PCIe SSD. China Sciencepaper. Vol. 9 (2014) No. 4, p. 403-405.
[3]. Yang Shian, Wang Zicheng, et al. Design of Data Acquisition and Display System Based on Zynq-7000. Instrument Technique and Sensor. Vol. 9 (2020) No. 8, p. 60-64.
[4]. Information on: https://china.xilinx.com/
[5]. Xu Lei:Development of portable radar testing equipment based on ZYNQ(Master of Engineering, Harbin Institute of Technology, China 2018).p1-16.