

Design and Implementation of SoPC with Multi-Bus on a Chip*

Fangjun Jian^{1,2}, Jizhong Han¹, Chengde Han¹, Qin Zhang^{1,2}, Xuhui Liu^{1,2}

¹ Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100080

² Graduate School of the Chinese Academy of Sciences, Beijing 100039

Abstract

SoPC (System on a Programmable Chip) is one important kind of SoC solution based on PLD (Programmable Logic Device). At the same time, PBD (Platform-based Design) has become popular and hot on the SoPC design. However, single-bus architecture has been widely adopted in the design of SoPC using PBD. Therefore, many applications, especially those for real-time data processing, suffer from the bottleneck of bus. In our paper, a novel multi-bus SoPC (MBSoPC) architecture design is provided, which adopts multiple PLBs (Processor Local Bus) in one chip. Moreover, a platform based on MBSoPC is built to verify this architecture, which uses three FPGA chips and one of them using three PLBs to act as the main controller. The other two are connected as coprocessor to speed up some critical processing, such as FFT or DCT used commonly in imaging and video processing. We use the MBSoPC platform for real-time image processing. The results show that this architecture can improve system performance successfully.

Key Words:

SoPC, Platform, FPGA, PLB, MBSoPC

1. Introduction

With the rapid developments of microelectronics and electronics automatically design, the scale of integration circuit is much larger and the chip is more complex. So it is possible to implement a full system on one single chip which is called SoC (System on Chip). The design of SoC covers both hardware and software. Top-down SoC design starts system level design, architecture design including partitions, system implement. What's more, embedded software, embedded operation system and other application software, also play important roles in SoC design.

As one of main solutions of SoC, SoPC (System on a Programmable Chip) is based on PLD (Programmable Logic Device), especially on FPGA (Field Programmable Gate Array). Actually, it is regarded as the product of combination of SoC and PLD technology. SoPC was firstly proposed by Altera Cooperation and now becomes the most important trend of the development of SoC system. As a special embedded system, SoPC possesses the merits of SoC system and PLD. On one hand, acting as SoC, it can finish the main logic functions in on chip. On the other hand, acting as a kind of PLD, SoPC is flexible, scalable, updatable and partly programmable.

To design a SoPC System, buses are usually used to connect different parts in a chip. Among them, IBM Core-Connect buses which include PLB, OPB and DCR, are widely adopted by most of present SoPC systems. However, traditional SoPC designs use only one system bus, so it can perform only one read and one write operation at the same time, even though there are many masters and slaves. Therefore, most present applications, especially those for real-time data processing, suffer from the bus bottleneck since these applications always require multiple read/write transaction events between multiple masters and multiple slaves.

In our paper, a novel multi-bus SoPC (MBSoPC) architecture design is provided, which adopt multiple PLBs (Processor Local Bus) in one chip. Moreover, a platform based on MBSoPC is built to verify this architecture, which uses three FPGA chips and one of them using three PLBs to act as the main controller. The other two are connected as coprocessor to speed up some critical processing, such as FFT or DCT used commonly in imaging and video processing.

* Corresponding author: Fangjun Jian, P.O.Box 2704-25#, P.R.China 100080. E-mail: jian@ict.ac.cn.

We use the MBSoPC platform for real-time image processing. The results show that this architecture can improve system performance successfully.

The rest of the paper is organized as follows: Section 2 gives a brief background about platform-based SoPC design and some popular On-chip buses. Section 3 elaborates the design of the MBSoPC. The implementation is presented in Section 4. Section 5 provides the solution for special application and performance analysis. Finally, conclusion is given in Section 6..

2. Platform-based SoPC Design and On-Chip Bus Architecture

VSIA (Visual Socket Interface Alliance) defined platform and platform-based design as follows[9][10]: A platform comprises a common, integrated, and managed set of features, upon which a set of products or product family can be built. In the SoC context, it is a library of virtual components and an architectural framework, consisting of a set of integrated and pre-qualified software and hardware Virtual Components, models, EDA and software tools, libraries and methodology, to support rapid product development through architectural exploration, integration and verification.

In a word, platform is architecture integrated both hardware and software. Furthermore, its hardware and software could be reused facilely; therefore it deduces time to market greatly. One complete platform is comprised of function IPs and communication IPs, such as embedded MPU, RTOS (Real Time Operation System), peripherals and middle-wares. Now main FPGA suppliers have developed many platform-based devices for different applications.

Platform-based design is an integration-oriented approach, which emphasizes systematic reuse. It is good for developing complex products upon platforms with compatible hardware and software because it has such advantages as decreasing risks, costs, and time to market obviously. PBD has become popular and hot on the SoPC design. Its kernel is maximum reuse.

On-chip buses are like on-board buses, which integrate many function modules in some flexible way. These function modules may be designed by different suppliers. Therefore it is more flexible for design and easier for implementation. The signals of on-chip bus are usually one-way, not tri-state, for it is benefit for low power and high performance. In practice, there are two generics of on-chip bus in the SoPC design, standard on-chip bus and customized on-chip bus. The former is popular for there are many IPs which are encapsulated for these standard bus interfaces, such as AMBA of ARM and Core-Connect of IBM. Once stand on-chip bus is adopted, Designers can pay more attention to inner architecture, including algorithms implementations, arithmetic logic unit, data and control paths, without caring for communication interface.

Three kinds of stand on-chip buses are available in SoPC design, which are AMBA of ARM, wishbone of Silicore and CoreConnect of IBM respectively. These buses have some common features, such as separation of address and data bus, while they have their own characters. AMBA is adopted widely in FPGA of Altera corporation, together ARM architecture[16]. Wishbone is favorable for its flexible interconnects capability and free open source. Therefore it is welcomed by open source organizations and individual. CoreConnect plays an import role in the high performance embedded system with PowerPC processor. Xilinx supplies CoreConnect bus in its high-end FPGA devices. CoreConnect is a group of buses, which includes processor local bus (PLB), On-Chip Peripheral Bus (OPB), Device Control Register Bus (DCR). PLB bus is applied in high performance and low delay connections, which support read and write transactions contemporary and this can maximize the usage of the bus. OPB bus is a second level bus, which is used for connecting low speed peripherals. DCR bus is a separate controller loop with PLB and OPB, which is designed to separate channels of controller signals and of data signals[13][14][15].

In our platform-based multi-bus SoPC, PLBs architecture is provided, because Xilinx supports PLB well in its high-end FPGA devices. It is easy for design and verification by CoreConnect tool kits. There are many reusable IPs with PLB interface offered by Xilinx cooperation and third parties.

3 Multi-PLB Architecture Design

From the above section, PLBs support both read and write transaction simultaneity, address pipeline, multiple masters and slaves. It is a high performance bus and is supported well in FPGA devices of Xilinx. Specially, many reuse IP with CoreConnect interface, such as memory controller, PowerPC and Microblaze microprocessor, are available. MBSoPC is suitable for high performance digital signal processing, which requires high speed data transfer and processing. Therefore PLB bus is adopted in our multi-bus SoPC platform, in which almost custom IPs are encapsulated as PLB interface at the same time.

3.1 Three-PLB Design and Bus load Balance

There are many masters and slaves exchanged data in the application of our SoPC platform. To speed up the data transfer between these function module, read and write data simultaneity must be supported. If single-bus architecture is adopted, there are only two master devices which read or write simultaneously. Therefore multi-PLB architecture is designed in which more than two devices read and write simultaneously. In the multi-PLB architecture, different masters and slaves are allocated to some PLB bus and read and write transaction can be contemporary in every PLB bus. What's more, tactics, global asynchronism and local synchronization, is adopted. Therefore higher performance is obtained for the set of high performance modules work at high frequency. And local synchronization is benefit for design and implementation. Different modules attached to different clock domain exchange data by use of asynchronous handshake mode. Asynchronous FIFO is often used to enhance asynchronous data communication efficiency.

In the multi-Bus architecture, different IP modules are separated according application specification and their performances. What's more, different bus may run at different clock frequency. In every PLB bus, address pipeline is adopted and read/write transactions are simultaneous. Multi-PLB architecture is benefit for IP design and bus efficiency, and speed up communication between multiple masters and slaves modules. Outsidess, multi-PLB architecture is flexible to balance different bus load without leading to change of other modules. A bridge between two pieces of PLB is designed as a common master and slave module of PLB bus.

It is easier for balance load of different buses in multi-PLB bus, because all the PLB buses have the identical signals definition and transaction rules. Therefore, it is smooth for a PLB master or slave to move from one bus to other. Bus load balance may be adopted at any moment according to the requirements of the different bus module, including throughout, frequency and occupancy factor of bus. In the SoPC platform, all the high speed modules are integrated in a bus, which is running at a high frequency. At the same time, the two bus modules are often allocated to different bus if both of them have the high occupancy factors.

3.2 Master-slave FPGA Architecture and Extensive Memory Controller

The target application of MBSoPC is oriented to mass memory application, even more than two gigabytes, which is impossible to be allocated in one piece of FPGA. There are only several megabyte memory, include distribute RAM and block RAM, commonly in a piece of FPGA. An extensive memory of DDR SDRAM module is designed outside FPGA, which volume is two gigabytes. What's more, Flash is added to the SoPC platform for embedded operation system and application firmware.

Processing performance is a key for MBSoPC platform design. To enhance the platform flexibility and applications, there are three pieces of FPGA in SoPC system, as shown in figure 1. One is the main controller, and the other two are coprocessor for some specify application. Coprocessors can speed up critical processing and increase the system performance greatly, such as FFT or DCT used commonly in imaging and video processing. At the same time, there are three pieces of PLB bus in the main control FPGA. One PLB is designed for high performance input and output. Here the input and output interface is designed for high speed applications, therefore I/O is connected with PLB bus, not OPB bus traditionally. The other two are inner processing bus, which is used to exchange data at a high speed between masters and slaves. The two buses have one gigabyte of DDR-SDRAM memory and a coprocessor respectively.

One microprocessor is integrated on the first PLB bus, which is designed to control all the module of three pieces of PLB. Embedded operation system is running on the microprocessor. System status is checked and reported by MPU and real time operation system. In fact, if logic size of two coprocessors is so small and volume of main FPGA is so large that Main FPGA can hold the logic both of two coprocessors, the two slave FPGA could be ignored or canceled.

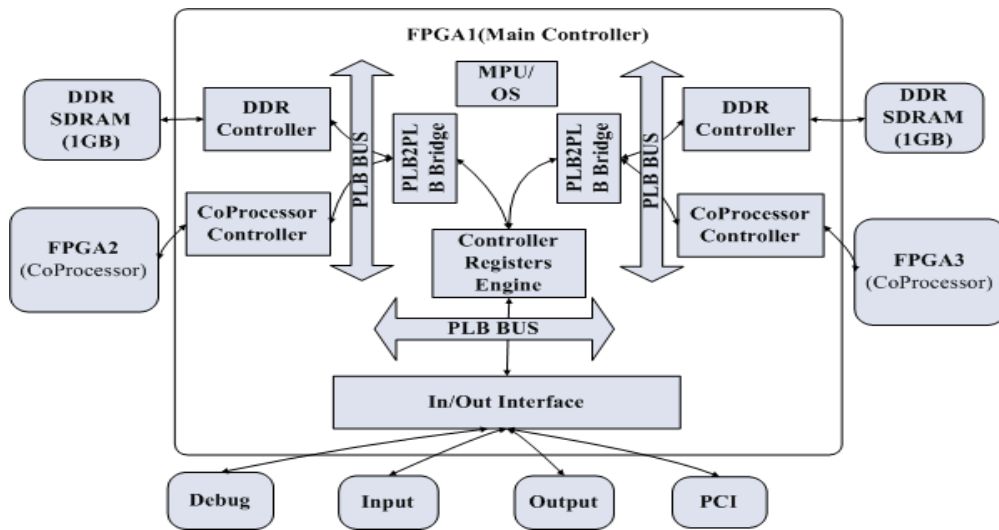


Figure1. Architecture of SoPC board system

4. Implementation of Multi-PLB Architecture

Implementation of multi-PLB architecture includes PLB2PLB bridge, asynchronous circuit design and bus load balance. Furthermore, design for verification is researched and implemented. A fast PLB debugger is designed in the MBSoPC.

4.1 Asynchronous PLB2PLB Bridge

A bus bridge is used to connect two PLB. In this SoPC platform, a single-way bridge is developed, just as following figure2. In other word, one side of this bridge is the slave module of PLB-A, the other side is the master module of PLB-B. For a synchronous design, both sides of the bridge have the same clock domain, and control signals and data is connected with each other directly. For an asynchronous design, control signals and data must be indirectly connected with each other by complex handshake. E.g, NRZ handshake and grey code are adopted to transfer controlling signals.

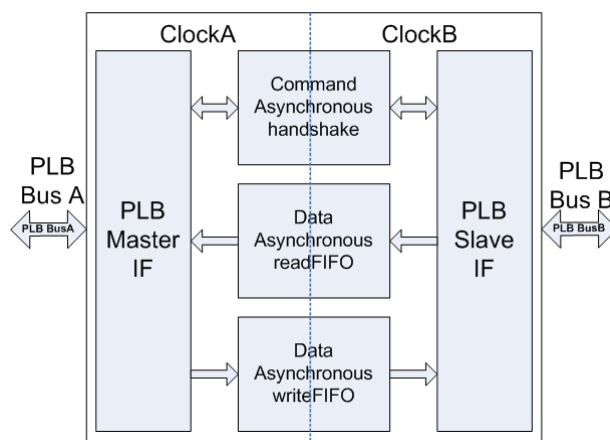


Figure2. A single-way PLB2PLB bridge

For PLB bus is adopted as on-chip bus, BFM[13] is chose to verify the single-channel PLB2PLB bridge. The environment of verification is as shown in figure3. The bridge is used to connect PLB-A and PLB-A. There are two masters, an arbiter and a monitor in PLB-A. On the contrary, there are two slaves in PLB-B. The two masters request respectively to access any slave any time at random under the test environment. Monitor A and B is used to report the

status of PLB2PLB bridge or set it to some state.

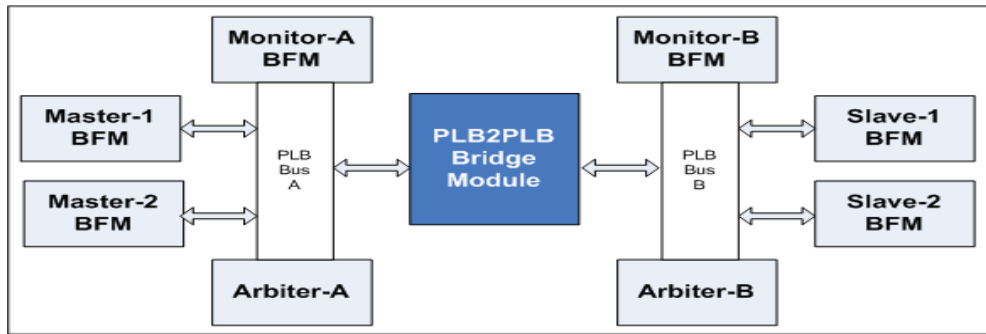


Figure3. The Verification of PLB2PLB Bridge Based BFM

4.2 Debugger Design and Design for Verification

There are many clock domains in the SoPC, such as PCI, USB, DDR SDRAM, JTAG, high speed custom input and output. PCI module is designed for debugging and communication with host PC in the SoPC platform. It has three sub-modules according to space, which are configuration, IO and memory sub-modules. Configuration sub-module is designed for identifying and configuring SoPC as a standard PCI device. IO sub-module is used to access control registers immediately. These control registers describe the command and status of SoPC debug finite state machine, such as DMA between initiator and target of PCI. Memory sub-module is developed for DMA and Debug. Both DMA and Debug start from host PCI module, such as PC, to the PLB slave of some bus, which may be under several buses. A shadow memory is set, which maps a section of PLB memory space to PCI memory space. Using PCI commands, such as IO read, IO write, memory read and memory write, the content of some section of PLB memory space is checked or reset by the shadow memory.

The platform-based SoPC is not a perfect case to verify its function. Therefore, DFV (Design for verification) is inserted into the key finite state machine [11][12]. When any units couldn't work normally during the test, the system will be chose to return to the DFV state. Then DFV could output an error report and change the system state into the other normal state in order to continue verification. In other words, more function verification could be carried out and the verification time is reduced enormously, after DFV is inserted. Otherwise the test system has to wait until the debugs are removed or the test bench is modified.

5. Platform Application and Performance

Jim Grey brought forward a new experimental rule that data volume in 18 months is equal to all data ago under the network environment. Mass data is a huge challenge for storage and transfer. E.g. Video source rate of HDTV is about 1.9Gbps. Data compress is absolutely necessary and key, especially for imaging and video data, in which there is redundance. The real time compress of mass data is commonly as follows. The first step is transform, such as FFT (Fast Fourier Transform, FFT), DCT (Discrete Cosine Transform, DCT) and DWT (Discrete Wavelet Transform, DWT). Transform is often used for mass data from one field to other field, in order to find more data relation and to eliminate redundance. For example, time domain is changed to frequency domain by FFT. Following is determining reserve data or redundance data. Only reserve data is encoded and kept, the other is ignored and discarded. For imaging and video data, much unimportant detail is deserted in which loss compress is adopted. The last is to quantitate and encode data for output. An application of the platform-based SoPC design is implementation of real time compress of mass data. The detail design is as figure 4.

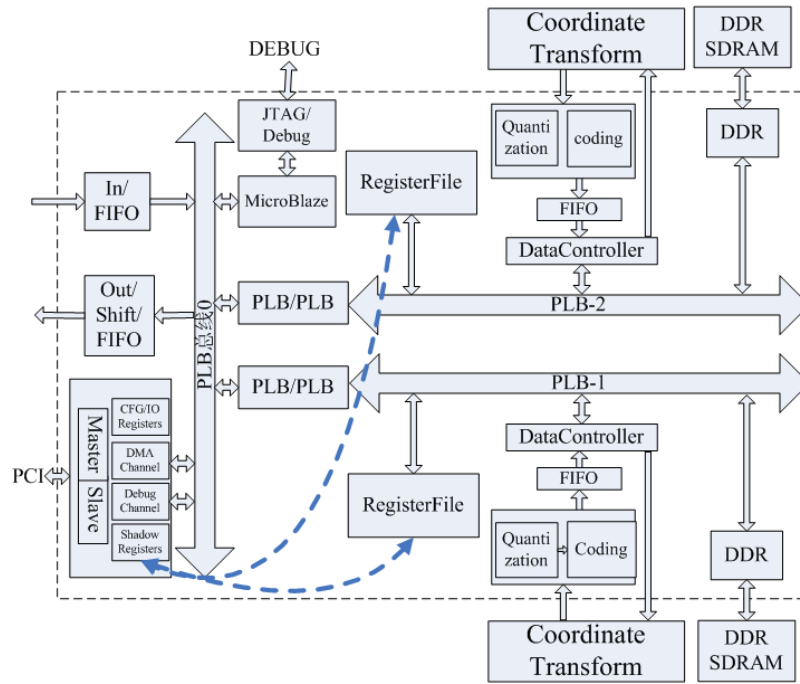


Figure4. Real time compress of Mass data in platform-based SoPC

The platform-based MBSoPC solution supports read and write transaction simultaneously, parallel processing and address pipeline. The platform-based MBSoPC works at 50 Mhz. Total time is constitutive of input, processing and output time. The process time is critical. MBSoPC is design to speed up process time by mean of parallel computing. Two sets of processing cell is working together. In other word, there are two sets of process component which are arranged to two separate pieces of PLB. Speed-up of processing is 2. System is designed by pipeline, including input, processing and output. Processing includes FFT/DCT transform, Quantified, coding. There is an application of geographic map compression. A frame of such map is made of an array of complex pixels of 16384x16384, even double size. Every pixel has a real and an imaginary with float type. In a word, the size of a frame of image is 2G bytes at least. If a 64-bit bus is used as IO interface at the speed of 50 Mhz, it takes 5.36 seconds to input and output each other. In the single-bus architecture, it takes 10.2 seconds for data compression. In MBSoPC, two sets of data compression is deployed to speed up processing, both of them can works at 50 Mhz frequency. System can finish compressing data of two frames of map in 10.2 seconds. In other word, it finishes compressing a frame of map in 5.1 seconds, faster than input/output slightly. Therefore MBSoPC of data compress works with high efficiency pipeline than single bus SoPC.

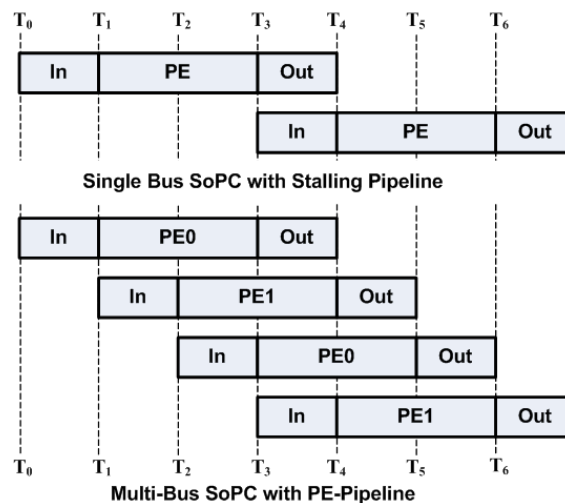


Figure5. Pipeline of single bus SoPC and MBSoPC

6. Conclusion

To get rid of the disadvantage of single-bus design for SoPC system, we provide a novel architecture named MBSoPC. In the design of MBSoPC, three PLBs are used to build a multi-bus SoPC architecture. Moreover, this architecture are implemented in one system including three FPGAs, in which one FPGA acts as the main controller by laying three PLBs, and the other two are designed to speed up some critical processing. By using this architecture, we can achieve obvious performance improvement since we can perform multiple read or write operations at the same time. Therefore, this architecture will satisfy requirements of many applications, especially those for real-time data processing. In our paper, the design and implement the MBSoPC platform for real-time image processing are given. Performance results show that this architecture can improve system performance successfully.

Acknowledgement

This research is partially supported by Chinese 973 Program (2004CB318202), and Faculty Research Grant at Institute of Computing Technology, Chinese Academy of Sciences, and National Nature Science Foundation of China under Grant No.60303017.

Reference

- [1] Henry Chang, Larry Cooke, Merrill Hunt, Grant Martin, Andrew McNelly, Lee Tod, *Surviving the SOC Revolution - A Guide to Platform-Based Design*, Kluwer Academic Publisher, 1999
- [2] Keutzer.K, Newton.A.R, Rabaey.J.M, Sangiovanni-Vincentelli, System-level design: orthogonalization of concerns and platform-based design, *Computer-Aided Design of Integrated Circuits and Systems*, IEEE Transactions on Computer-Aided Design of Integrated Circuits and systems, Volume.19, Issue.12 , Dec. 2000, Pages:1523~1543
- [3] K.Compton, S.Hauck, *Reconfigurable Computing : A Survey of Systems and Software*, ACM Computing Survey, June, 2002:171~210
- [4] Ascia.G, Catania.V, Palesi.M, A GA-based design space exploration framework for parameterized system-on-a-chip platforms, *IEEE Transactions on Evolutionary Computation*, Volume.8, Issue.4, Aug. 2004, Pages:329~346
- [5] Neil D.Jones, Carsten K.Gomard, *Partial Evaluation and Automatic Program Generation*, Prentice Hall International Publisher, 1999
- [6] Thomas H.Cormen, Charles E.Leiserson, etc, *Introduction to Algorithms(Second Edition)*,the MIT Press, 2004
- [7] Xilinx Inc. [Online], Available: http://www.xilinx.com/ise/embedded/bfm_simulation.pdf, 2005
- [8] Wayne Wolf, A Decade of Hardware/Software Design, *IEEE on Computer*, April, 2003
- [9] Alberto Sangiovanni-Vincentelli,Grant Martin, Platform-based design and software design methodology for embedded systems, *IEEE Design & Test of Computer*, Nov. 2001
- [10] Bob Altizer, Platform-Based Design: The Next Reuse Frontier, *Embedded Systems Conference*, San Francisco, 2002
- [11] Michael L. Bushnell, Vishwani D.Agrawa, *Essentials of Electronic Testing for Digital, Memory and Mixed-signal VLSI Circuits*, Kluwer Academic Publisher, 2000
- [12] Alfred L.Crouch, *Design For Test: For Digital IC's and Embedded Core Systems*, Prentice Hall PTR, 1999
- [13] IBM Inc, *Processor Local Bus Functional Model Toolkit User's Manual Version 4.7*, 2002
- [14] IBM Inc. 64-bit Processor Local Bus Architecture Specifications. Version 3.5, May 2001.
- [15] IBM Inc. Processor Local Bus Functional Model Toolkit User's Manual Version 4.7, 2002
- [16] ARM Inc. AMBA® AXI Protocol v1.0 Specification, 2004