Implementation of Remote Adapter for Debugging of Ubiquitous Embedded Software

Myeong-Chul Park, Seok-Wun Ha, *Member, KIMICS*

Abstract – Since ubiquitous embedded software is sensitive to the resources and environment of target system, it should be debugged in the same environment as actual target system. However, existing tools to debug embedded software, in which access to internal signal or resources is limited, are uneconomical. In the thesis, economical and practical USB-JTAG Adapter that can use open GDB is suggested. It can remove existing limitations of environment implementation that have many difficulties in implementing an environment for remote debugging. Hence, the thesis provides economical interfacing environment that can debug ubiquitous embedded software inside remote system.

Index Terms – Embedded Software, SoC, JTAG, USB, Remote Debugging, Ubiquitous

I. INTRODUCTION

To develop ubiquitous embedded software [1][2][3] stably, an effective debugging tool is needed. However, since embedded software is generally sensitive to limited resources and timing and should be debugged in the same environment as target system, there exists a difficulty in making an environment. In particular, in the SoC (System On a Chip)[4][5], which accumulated modules, such as memory or input and output tools, in a processor, access to internal signal or resources is limited, and therefore debugging is more difficult. To solve this problem, the On-Chip Debug (OCD) [6][7][8] technique that implemented debug module inside the processor is employed. Existing commonly used tools using the OCD technique are uneconomical and depend on vendors for the expansion for new target system, and this makes the development of embedded software in new system difficult in college institutes or industries. In the thesis, an adapter which can debug SoC software accessing to a target system in a remote place, based on the standardized JTAG (Joint Test Action Group) [9][10][11] in the industry, is suggested. The suggested adapter uses the USB (Universal Serial Bus) of host system and has a structure connected to the JTAG of target system. This does not influence target system and makės more economical and effective remote debugging possible by using high-speed USB port. Also, it is designed to have the flexibility of debugging environment and the expansion characteristic of existing debugging tool so that the change of users may be easy by implementing the firmware for the JTAG handshaking in the internal memory of the adapter in download way. To examine working conditions of the implemented adapter, JTAG debugging tool EDebugger [12] for Intel XScale is used. The EDebugger was implemented to use only the interface of USB port and JTAG port for expensive ICE (In Circuit Emulator) equipment [13] using GDB[14] and Jelie as GNU debugger. From the experiment results, it was confirmed that it can constitute normal debugging environment. In Paragraph Two, as the background of the study, traditional debugging techniques for existing debugging, JTAG and USB are examined. In Paragraph Three, the implementation of firmware inside adapter for adapter design and handshaking, which are core parts of the suggested environment, and experiment results are described. Finally, conclusions and further studies are suggested.

II. BACKGROUND

In this Paragraph, debugging techniques and JTAG are examined, and USB, which forms the basis of the environment, is also examined.

A. Debugging techniques

ICE among existing techniques makes hardware imitation possible, and therefore as a means convenient for real time development has hardware and software function for the error correction of real time input and output. The logic analyzer is a tool with multiple channels, which catch, record and represent the logic signal of digital system, and it outputs binary forms and a series of pulse forms with rectangle wave shapes. Since these techniques access to the resources of target system directly and inspect or control system conditions, they ar not appropriate for debugging for the SoC program, in which access to internal signal or resources is limited. Also, they have impractical limitations as very expensive equipments. Therefore, the thesis suggests tools that debug the SoC software using debug module provided from the SoC processor itself, based on JTAG.

B. JTAG

The JTAG, known as Boundary-Scan widely, has Boundary Scan Cell connected one-on-one to external pin inside chip and has the function of hardware tests or checking connection conditions by performing all

Manuscript received May 10, 2005.

Myeong-Chul Park is Ph.D. course student in the Dept. of Computer science, Gyeongsang national Univ. Jinju, Korea, (e-mail: africa@gsnu.ac.kr)

Seok-Wun Ha is a professor in the Dept. of Computer science, Gyeongsang national Univ. (a member of RICIC; Corresponding author to provide e-mail : swha@gsnu.ac.kr)

movements, which processor can make, arbitrarily through cell. The whole interface is controlled by five pins (TDI, TMS, TCK, nTRST, TDO) called as TAP (Test Access Port), and using these, all external pins of the device can be driven or the values can be read irrespective of the processor conditions. The JTAG interface on applied processor can be used as hardware interface for the software debugging of Intel/PXA system. Boundary-Scan test circuit is composed of TAP (Test Access Port) pin, TAP controller, command register and a series of test register. These include Boundary-Scan register, bypass register, device identification register, and data dependency register. Fig. 1 shows them.

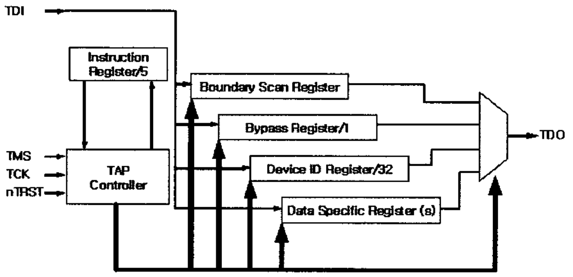


Fig. 1 Block Diagram of JTAG

C. USB

The USB was developed by Compaq, Intel, Microsoft, Nec as the leading figure in 1994, for the convenience to use, the wide expansion characteristic and the connection of PC with telephone. Largely two kinds of chips take charge of the connection with the USB in the side of the Host side, and they are provided by Intel and Compaq. Presently, the USB Spec 2.0 is released, and supports maximum 12 Mbps in the Spec 1.1, and maximum 120 Mbps in the Spec 2.0. The USB is constituted by stratified structure and communicates with USB device using master/slave protocol. That is, in the Device side, communication line to the Host side can not be made. Hence, communication can be started only in the Host side. Since the USB Host Controller Driver is mostly implemented and stabilized, the USB Client Driver of the relevant adapter is implemented in the study.

III. REMOTE ADAPTER

In this Paragraph, the design, implementation and test of USB-JTAG Adapter for remote debugging using the USB port of host system and the JTAG interface of remote target system are examined.

A. Design of the Adapter

The EDebugger, a hardware-based ubiquitous embedded software debugging tool, employed in the thesis is designed as debugging module for GDB and Intel/PXA processor[15]. Also, the EDebugger consists of host part and target part, separately, to remove the resources limitations of embedded system.

The host part has the function of converting GDB command into debugging command for Intel processor, such as PXA210, PXA250 and PXA260, and the function of controlling the USB for high-speed data transmission and the JTAG of target system. And, the target part controls the executed programs and is equipped with debug handler, which is a software module that returns the results to the host system. Its overall structure is shown in Fig. 2.

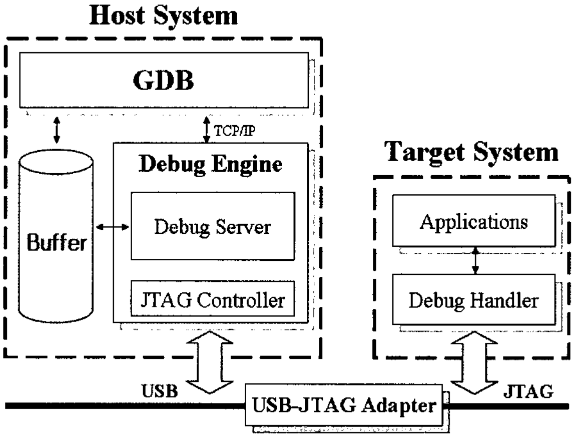


Fig. 2 Structure of EDebugger

In the thesis, economical USB-JTAG Adapter for smooth connection between the host of EDebugger and the target system was designed and implemented.

As the USB chip set which plays a central role of the adapter, the AN2131QC [16] of Cypress, which is widely known as EZ-USB, was employed. Fig. 3 shows the structure of AN2131QC. As can be seen from the figure, it is necessary to control the USB signal properly which is input and output in SIE (Serial Interface Engine), that is, USB signal to JTAG signal.

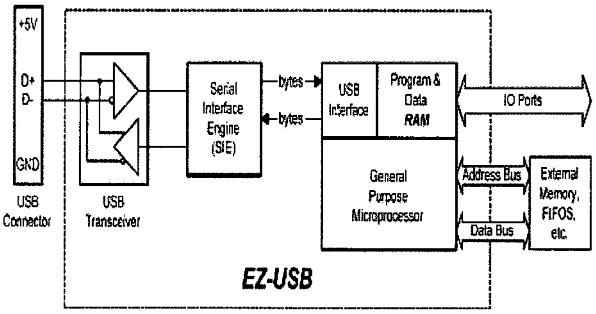


Fig. 3 AN2131QC(EZ-USB) Chipset

For this, the firmware is required in the internal memory. The adapter has internal memory of 8KB, and to send the signal transmitted from the USB port to the JTAG end of the target system, proper firmware should be downloaded to the internal memory. The firmware is written in C code, and it is converted into Hex code and then is downloaded to the internal memory through the device drive of the host. Table 1 shows the summary of important modules that firmware should have. Of course, additions and changes are possible according to user's requests.

Table 1 Important Modules (Firmware)

Module	Description of Function
initPorts	Initialize the I/O port of USB chip
jtagReset	Initialize the JTAG port
cpuReset	Initialize the Processor of Target System
ireg	Access of Instruction Register in JTAG
dreg	Access of Data Register in JTAG
jtagReg	Access of Register in JTAG

For the firmware to make handshaking between connections smooth normally, the motive finite state machine to control the procedure of 16 test logic operations of TAP (Test Access Port) controller should be understood. The TAP is controlled by the bus master. The bus master interfaces the test access port by automatic test equipment or logic device which can program. The TAP controller changes the state only for the positive edge or power-up of TCK. The values of input signal of test mode sate (TMS) in the positive edge of TCK control the procedure of state changes.

The TAP controller is initialized automatically at the time of power-up. Also, the TAP controller can be initialized by applying 1 signal as the TMS input during five TCK periods.

Fig. 4 shows the state transition which occurs in the TAP controller. the value for each transition (0 or 1) means the TMS value. It should be observed that the digital signals of all application processors take part in Boundary Scan except the PWR_EN pin. This keeps scan operation from shutting off the power of application processor.

Since the AN2131QC chip employed in the study uses 8051 core, the development tool for 8051 can be used when programming. Also, it has an advantage that it does not need separate ROM writer since the firmware can be downloaded like software through Re-Numeration. Since the hardware of chip includes the basic function related to basic USB, the firmware consists of conversion modules related to the JTAG. Also, it supports Full Speed and has enough number of input and output pins. The input and output are made by three 8-bit ports of A, B and C, and even each pin in a port can select and establish the input and output. Of course, part of the input and output pins can be used for special purpose, and this can be handled with the firmware.

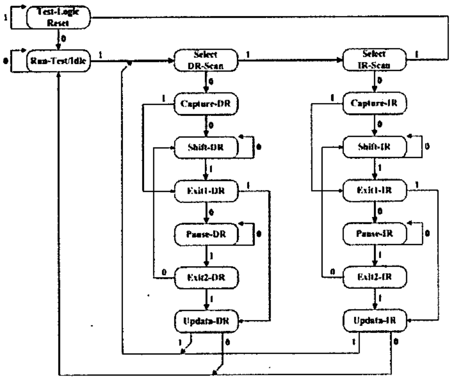


Fig. 4 TAP Controller State Transition Diagram

For reference, Fig. 5 shows not only the part to initialize the input and output port (B port) of USB chip, but also module initializing JTAG, among firmware modules, in C code.

```
void initPorts(void)
{
OUTC=0x00;
OEC=0xff;
PORTCCFG=0x00;

OUTB=0x00;
OEB=JTAG_TMS|JTAG_CLK|JTAG_TDI|JTAG_PWR;
PORTBCFG=0x00;

OEA=0x00;
PORTACFG=0x00;
}

 void jtagReset(void)
{
unsigned char loop_count;
OUTB=JTAG_PWR;
OUTB=JTAG_PWR | JTAG_CLK;
for(loop_count = 0; loop_count < 5; loop_count++)
{
OUTB=JTAG_PWR | JTAG_TMS;
OUTB=JTAG_PWR | JTAG_TMS | JTAG_CLK;
}
OUTB=JTAG_PWR;
OUTB=JTAG_PWR | JTAG_CLK;
OUTB=JTAG_PWR;
OUTB=JTAG_PWR | JTAG_CLK;
OUTB=JTAG_PWR;
jtagRestarted = 1;
}
```

Fig. 5 The part of C code to Firmware

B. Implementation of the Adapter

Table 2 shows the mapping between the input and output pin (PB) of the chip set employed in the implemented adapter and the signal to be transmitted to the JTAG port of the target system.

To control the input and output pin, there are 4 registers (PORTBCFG, OUTB, OEB, PINSB). First of all, the PORTBCFG register, which can select the function of port when authorization of the power is applied, should be initialized with 0 as a default value. The 0 means general input and output use, and the 1 means specified specific use. The OEB register is employed for the use selecting the direction of input, and to be employed for the output use, the value of each register in firmware should be initialized with 1. After authorization of the power is applied to chip, the EZ-USB core searches EEPROM connected to I2C port. If EEPROM is detected and the content of the address of 0 is '0xB0', then the EZ-USB core copies Vendor ID, Product ID and Device ID from EEPROM into the internal memory storage. At this time, the EZ-USB core provides these bytes to the host as part of Get_Descriptor-Device requirements. If the content of the address of 0 is '0xB2', the firmware is considered to exist from the address of 7, and then it is downloaded into the internal memory. In the study, only the ID of relevant chip is provided using 24LC00 chip.

In the host system, the driver accorded with this information is loaded by OS, the driver downloads the firmware for 8051 to the RAM of EZ-USB, and then the 8051 core begins action.

Table 2 Pin Connection of USB & JTAG

B Port	JTAG
PB0	TCK
PB1	TDO
PB2	TMS
PB3	TRST
PB4	RESET
PB5	TDI
PB6	Not used
PB7	Not used

Internal modules like Debug Handler are needed additionally so that the JTAG signal transmitted to the target system is recognized by relevant processor.

C. Experiment of the Adapter

Fig. 6 shows the figure of connecting the target system with the host system using the implemented adapter, and Fig. 7 shows the screen that can confirm that the IDCODE of actual target system was read and recognized using EK-Debugger for operation experiments.

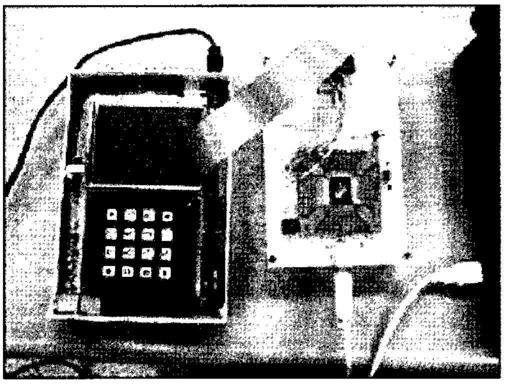
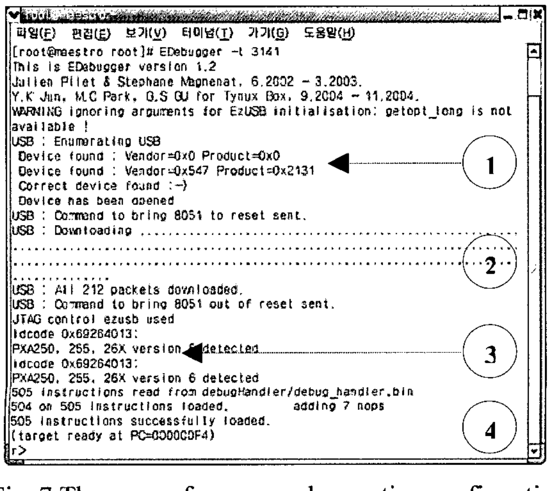


Fig. 6 Figure of Connecting the target system with The Adapter



```
[root@maestro root]# EDebugger -t 3141
This is EDabugger version 1.2
Julien Pilet & Stephane Magnenat, 6.2002 - 3.2003.
Y.K Jun, M.C Park, G.S GU for Tynux Box, 9.2004 - 11.2004.
WARNING ignoring arguments for EzUSB initialisation: getopt_long is not
available !
USB : Enumerating USB
 Device found : Vendor=0x0 Product=0x0
 Device found : Vendor=0x547 Product=0x2131
 Correct device found :-)
 Device has been opened
USB : Command to bring 8051 to reset sent.
USB : Downloading ...............................................................
.................................................................................
.................................................................................
...............
USB : All 212 packets downloaded.
USB : Command to bring 8051 out of reset sent.
JTAG control ezusb used
Idcode 0x69264013:
PXA250, 255, 26X version 6 detected
idcode 0x69264013:
PXA250, 255, 26X version 6 detected
505 instructions read from debugHandler/debug_handler.bin
504 on 505 instructions loaded.        adding 7 nops
505 instructions successfully loaded.
(target ready at PC=0000C0F4)
r>
```

Fig. 7 The screen for a normal operation confirmation

In Fig. 7, the number 1 represents that relevant adapter was normally found and recognized in the operation system, and the number 2 represents the part of downloading the firmware into the 8051 of the adapter. And, the number 3 represents the kinds of target systems after accessing to the JTAG of relevant target system is accessed and reading the IDCODE. Finally, in the number 4 represents that the Debug Handler was normally downloaded.

IV. CONCLUSION

In the thesis, an economical USB-JTAG Adapter was suggested for remote debugging. The suggested adapter can be employed as a cheap and economical equipment which can substitute expensive ICE equipment. Also, it was confirmed using existing open debugging tools that the operation was smooth. The development of this kind of tool can meet the demand of the embedded software, which will be increased explosively in the coming ubiquitous age, and strengthen the competitiveness. With continuous studies from now on, researches that can maximize the speediness as well as the practicalness should be carried out.

REFERENCES

[1] Rubini. A. and J. Corbert, *Linux Device Drivers Second Edition*, O'Reilly & Associate, June 2001.

[2] Kim. C., Kim. H., and Lim. C., Technology Trends and Development Strategies on Embedded Software for Ubiquitous Computing Era, 2003.

[3] Yaghmour, K., Building Embedded Linux Systmes, O'Reilly & Associate, 2003.

[4] H. Peters K., "Software Development and Debug for System-On-A-Chip," *Embedded Systems Conference*, 1999.

[5] Johnson, M., and N. Puthuff, "Debugging Embedded SoC Systems," *RF Design*, Feb. 2002.

[6] S. J. Nam, J. H. Lee, B. W. Kim, Y. H. Im, Y. S. Kwon, C. M. Kyung, and K. G. Kang, "Fast Development of Source-level Debugging System Using Hardware Emulation," *Asia and South-Pacific Design Automation Conference* , *IEEE*, pp. 401-404, 2000.

[7] ARM Ltd., Real-Time Debug, 2002.

[8] Hot-Debug for Intel XScale Core Debug, 2002.

[9] IEEE, IEEE Standard Test Access Port and Boundary-Scan Architecture, Std 1149.1-1990, 1993.

[10] Asset InterTech, Inc., and R. G. Bennefits, *Boundary-Scan Tutorial*, 2000.

[11] XJTAG Lt d., JTAG-A technical overview, 2003.

[12] Byung-Chul Kim, "A JTAG-Based Debugging Tool for Developing Embedded Softwares," *Proceedings of The 31st KISS Spring Conference*, 31(1), pp. 943-945, 2004.

[13] Signum Systems, JDSnet-ARM/ETM Emulator with Trace, 2003.

[14] Stallman. R., R. Pesch, and S. Shebs, et al., Debugging With GDB, 2003.

[15] Intel Corp., XScale Microarchitecuture for the PXA255 Processor, 2003.

[16] Cypress Semiconductor, *EZ-USB Technical Reference Manual*, 2002.

Myeong-Chul Park
Received B.S. degee in the Dept. of Computer Science from Korea National Open University, Seoul, Korea, in 1999. and M.S degree in the Dept. of Software from Gyeongsang National University(GSNU), Jinju, Korea, in 2002. Since 2003 to now, he has been Ph.D. student in Neuro Vision Lab, GSNU, Jinju, Korea. His research interests include Computer Vision, Image Processing, Visualization, Simulator, Parallel Programs and Debugging. He is a Member of KIMICS, KIPS, KISS, and KMS.

Seok-Wun Ha
He received the B.S., M.S. and Ph.D. degrees in the Dept. of Electrioc Engineering from Pusan National University. Since 1993, has been a professor in the Dept. of Computer Science, GSNU, Jinju, Korea. His research interests include Digital Signal Processing, Neural Network, Image Processing and Computer Vision. He is a Member of KIMICS, KIPS, KISS, and KMS.