International Journal of Multimedia and Ubiquitous Engineering
Vol.9, No.1 (2014), pp.143-152
http://dx.doi.org/10.14257/ijmue.2014.9.1.14

MDDI Protocol Packet Generation Method of Mobile System

Jong-Moon Kim[1], Chang-Su Kim[1] and Hoe-Kyung Jung[1*]

[1]*PaiChai University, Doma2-Dong, SeoGu, DaeJeon, Korea*

elcomtech@elcomtech.co.kr,ddoja@pcu.ac.kr, hkjung@pcu.ac.kr

Abstract

In this study, it is proposed to use software to create MDDI protocol packet which is required for the display device. The previously proposed method to create MDDI protocol packet was implemented in FPGA.

In the previous MDDI interface design method, data and synchronization signal were created inside of the FPGA chip. FPGA chip creates data received from CPU into serial data and creates packets and then controls signals. This design method is intended to reduce CPU load.

In this paper, the method of creating data in software, which was notpresent in the previous FPGA is proposed for design of data transmitter using the MDDI interface method.

Keywords: *MDDI, Protocol, Mobile system, Hibernation, FPGA*

1. Introduction

Recently, the display of mobile device has high resolution and the mobile device becomes so popular so that the needs for more effective high speed interface than previous method have arisen [1-4]. Due to these needs, Qualcomm's MDDI protocol and GSM-based MIPI protocol were introduced.

Most of the previous mobile devices used CPU interface and RGB interface, and these interfaces are still frequently used in the field which does not require high resolution.

MDDI method [5, 6] is one of the standard transmission methods proposed to implement previous parallel data transmission into serial data transmission in the display and multimedia data transmission. This method has advantages including high bandwidth, less number of connection lines, less power consumption and better EMI characteristics than previous parallel interface [4-7].

In this study, the method to create data in software, not in FPGA as previously proposed [8] was proposed in the design of data transmitter using this MDDI interface method [8].

2. Hardware and Software Designs

2.1. Hardware design

2.1.1 Hardware structure and functions

The mobile system consists of a processor for CPU, 512MB DRAM, 8MB NOR Flash, and 256MB NAND Flash. For OS, it is embedded with Linux which operates applications for the packet transmission, control and creation. The MDDI packet transmission block diagram is shown in Figure 1.

* Corresponding Author

ISSN: 1975-0080 IJMUE

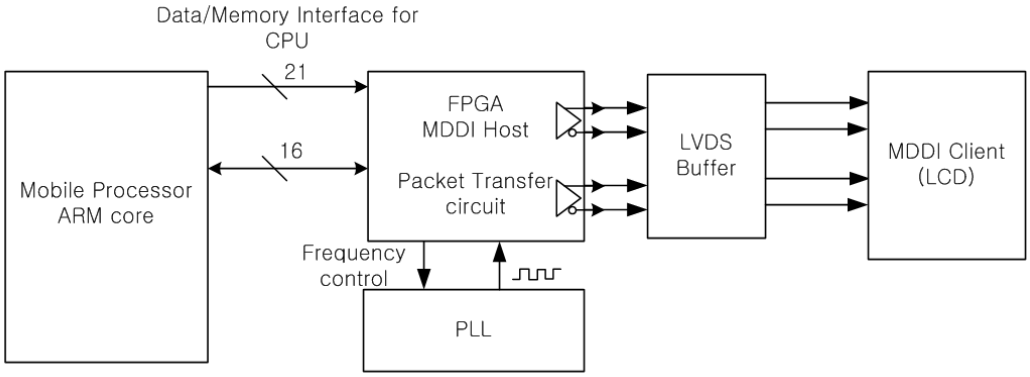

Figure 1. The MDDI packet transmission block diagram

While data delivered from the mobile process is being stacked up on FIFO, the LVDS TX Module inside of the packet transmission circuit reads data in 16 bits from FIFO, serializes them, and outputs data in 8 bits to LVDS Buffer. At this time, the conversion process is carried out in LVDS TX Module to output one byte data received from FIFO to MDDI Data/Stb. Figure 2 shows an example of synchronization signal creation to transmit this order of data "11110001011".

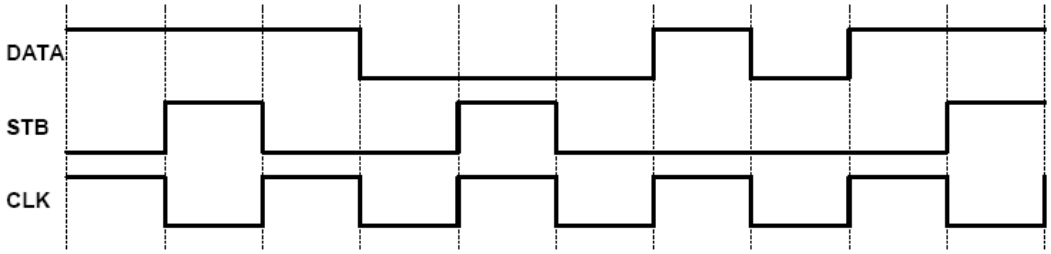

Figure 2. An example of synchronization signal creation to transmit

2.1.2 FPGA configuration

Packets of MDDI protocol created by the packet creation algorithm of mobile processor system are stacked on MDDI FIFO in FPGA. These packets are synchronized with the clock of External PLL Module and transmitted to LVDS TX Module. And then, LVDS TX Module serializes 8 bytes-data and transmits them to MDDI clients with synchronizing signals.

As Figure 3, the packet transmitter consists of an address decoder, register parts for the synchronization and the control of internal modules, a MDDI Hibernation Module for cancellation of maximum power-saving mode of the client, a MDDI FIFO Module, a LVDS TX Module, and a PLL Module .

1) MDDI FIFO

The memory of MDDI FIFO is a total of 8192 words (1word = 2byte). In this paper, I design to support MDDI Type I. The I/O design of FIFO is as follows.


```
component lpm_fifo1
	PORT
	(
		data	: IN STD_LOGIC_VECTOR (15 DOWNTO 0);
		rdclk	: IN STD_LOGIC ;
		rdreq	: IN STD_LOGIC ;
		wrclk	: IN STD_LOGIC ;
		wrreq	: IN STD_LOGIC ;
		q	: OUT STD_LOGIC_VECTOR (7 DOWNTO 0);
		rdempty		: OUT STD_LOGIC ;
		rdusedw		: OUT STD_LOGIC_VECTOR (13 DOWNTO 0);
		wrfull	: OUT STD_LOGIC
	);
END component;
```

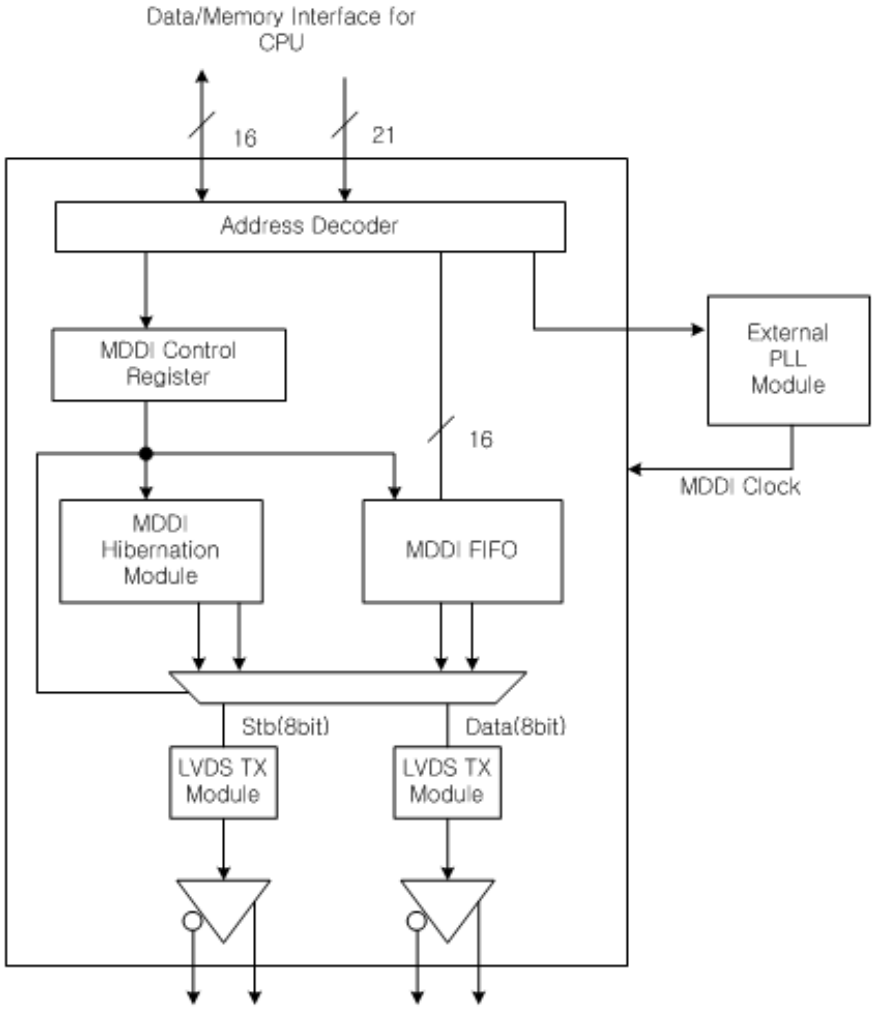

Figure 3. The block diagram of MDDI host packet transmitter

2) MDDI hibernation module

In this thesis, only the condition to wake client in the host was implemented. The flow of converting the client into the Hibernation status corresponds to Zones A~D as shown in Figure 4. When the packet analyzer processes link-shut down packet and notifies it to wake-up block, the host does not enter into the Hibernation status but checks data value while counting the toggle of strobe, and this is the task to confirm whether the host enters into the Hibernation status when CRC error occurs.

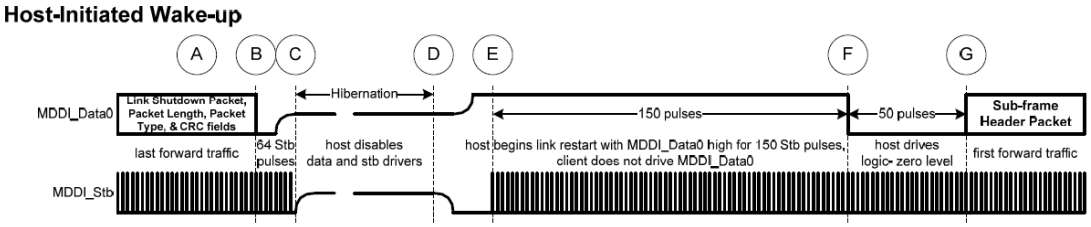

Figure 4. The flow to wake client in the host

3) MDDI PLL module

Figure 5 shows the internal structure of PLL module placed outside of the FPGA. The PLL module used is CDCE62002 of TI, and the frequency is set according to the following conditions and equations.

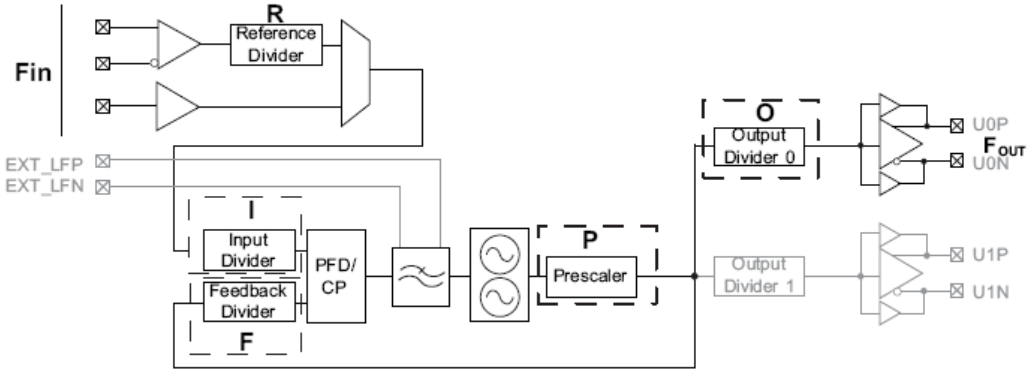

Figure 5. The block diagram of MDDI PLL module

At first, the following is the equation for Feedback Divider frequency setting.

Fout = Fin*F/(R*I*O), R = 1,

The restriction on the frequency setting for Output Divider * Prescaler * Fout is same as 1.750Ghz < O*P*Fout< 2.356GHz.

Fcomp= Fin / (R*I)
If FOUT setting for transmitting in 180Mbps of MDDI data transmission speed are F = 24, R = 1, I = 1, O = 8 and, Fin = 30MHz, then
Fcomp = 30,000KHz

FOUT = 90MHz.
At this time, the Fcomp value should fall within the range of 117Khz<Fcomp<30Mhz.

2.2. Software design

2.2.1 Packet generation program

In order to make MDDI packet data and transmit packet data, the packet generation of the mobile system application is composed of a command analyzer, a MDDI packing environment setter, a register access packet generator, a video stream packet generator, a link-shutdown packet generator, and a hibernation control section of controlling sleep mode clear packet output.

Each packet generator calls the function according to interpreted commands in the script document with the text type, and, video stream packet generates the packet by reading the picture files. Each packet generator uses the packet, which is generated where from the common MDDI Transfer Packet Memory. The packet is written in the above flow, synchronizes the HDDI Hibernation Module or is copied into the FIFO memory of FPGA, which is a packet transmitter through the external address/data bus.

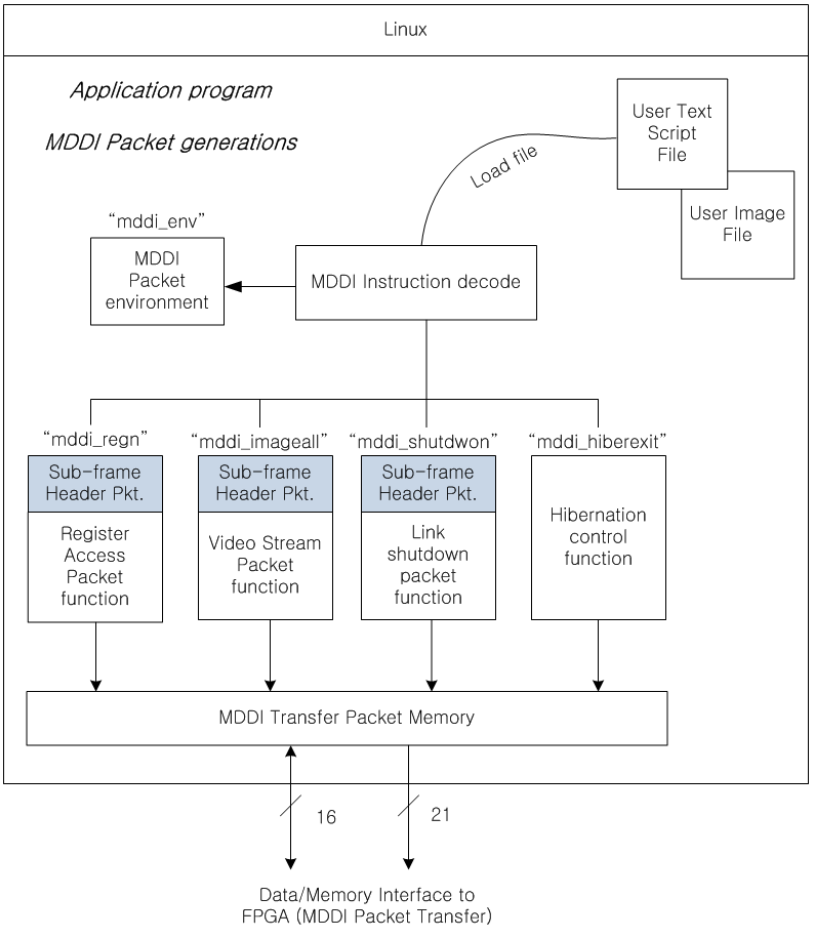

Figure 6. The block diagram of packet generation program

2.2.2 Sub-frame header packet

Sub-frame header packet is the first packet of all sub-frames, and its basic structure is same as shown in Figure 7. Sub-frame header packet is required for the host-client synchronization. All hosts should be able to create this packet, and all clients should be able to receive and analyze this packet.

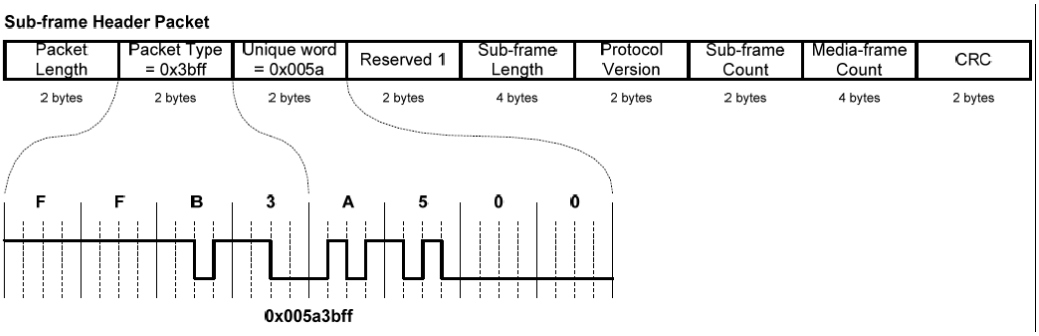

Figure 7. The basic structure of the sub-frame header packet

2.2.3 Link shutdown packet

Link shutdown packet is the packet which is transmitted from the host to client in order to shut down MDDI data and synchronization signal and access the Hibernation mode status. This packet is useful for saving power after the link is shut down and the static image is transmitted to the mobile communication device of client. In order to restore the link normally, the host sends the packet for restoring the link again. The first packet which is sent after hibernation mode is sub-frame header packet.

2.2.4 Video stream packet

Video stream packet should send image data to update the square area to the display. The size of image data may be as small as just one pixel or as large as the whole screen. Therefore, the displayed stream may be almost unlimited. (It is restricted by the system resource.) It's because all the steps required for displaying the stream are included in the video stream packet.

2.2.5 Register access packet

Register access packet accesses the register configured in client which is connected to MDDI link. This packet is mainly used for setting the control function of various displays or devices that perform the client function.

3. Measurement Result and Consideration

3.1 Sub-frame header packet

This sub-frame header packet gives the field data as in Table 1 to the memory of array structure referred as above, and was measured by comparing the output data. In addition, it confirmed normal working to the FIFO works of FPGA and the LVDS output under the hardware function.

Table 1. The field data of sub-frame header packet

Field name	Type	Description & Value
Packet Length	2 bytes	0x0014
Packet Type	2 bytes	0x3BFF
Unique word	2 bytes	0x005A
Reserved	2 bytes	0x0000
Sub-frame Length	4 bytes	0x0000_0009
Protocol Version	2 bytes	0x0000
Sub-frame Count	2 bytes	0x0000
Media-frame Count	4 bytes	0x0000
CRC	2 bytes	

3.2 Video stream packet

In order to test the video data output as shown in Figure 8, a 24-Bit RGB BMP file in 319 wide and 471 long for the static image was created, and setting of main fields for the video stream packet was set for packet and as shown in Table 2.

Table 2. Setting for the video stream packet

Parameter	Value	Descriptions
Mmbp	180	Clock rate = 90MHz
Subframe	1	Subframe frame length, 20Byte
Pixelmap	3	Video Data Desctiptor, 24BPP(888)
Subheader	0	Tags sub-frame header for every packet.
Reginfoset	0	the availability of reginfo, 0: not used
Reginfo	0	Register access packet 의 register count : 0x1
Videoresolx	319	Video stream packet resolution X value
Videoresoly	471	Video stream packet resolution Y value
Video Format	0x23	Pixel data attributes, Fixed 0x23
Imagestart	1	Image start line number

The parameter value of the script command "mddi_env" referred for setting main data to create the video stream packet is as shown in Table 2 above, and Pixelmap, Videoresolx, Videoresoly and Video Format are the values that correspond to main items which should be defined by the standard.

3.3 Link shutdown packet

The result about the link shutdown packet generation is as in Figure 9 and I could confirm that the designs of packet generation and data transmission are successful.

Link shutdown packet was created in MDDI interface Type 1 where and its Packet Length was an integer 20 (14h) and its Packet type was 69(45h). CRC was created by algorithm, with an output of 16 byte Zero value.

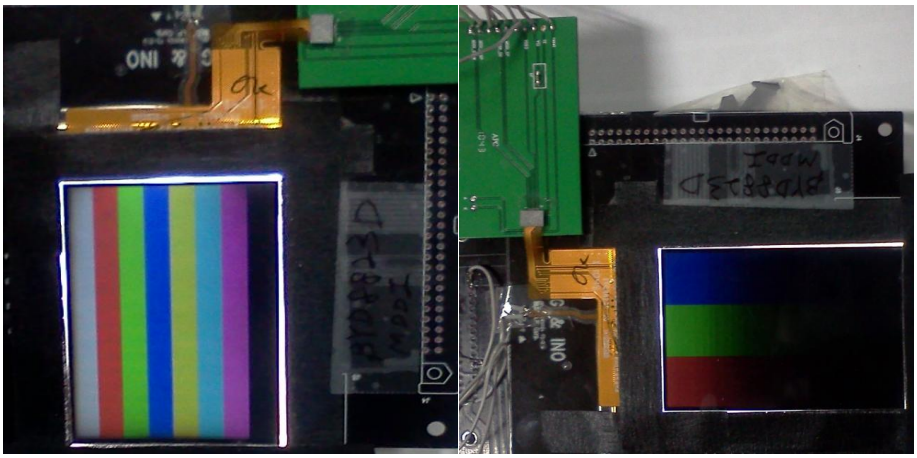

Figure 8. Test the video data output for the static image

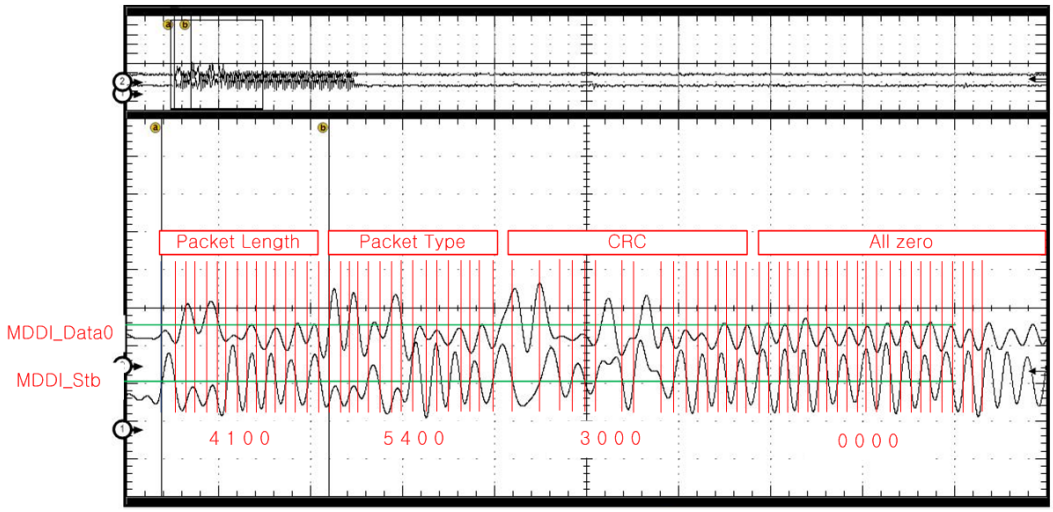

Figure 9. The output waveform of link shutdown packet

3.4 Register access packet

Read/Write Info value was set to 1 by standard. This value is the number of Register Data List which is set to 1 on Bit[13:0] by standard. This setting is to write data to 00 on the client for Read/Write of Bit[15:15].

It is confirmed through this test that the Register Data List of this register packet is the value subtracted 14 bytes from the packet length, and as shown in the test, if the register address is 0x11 and there is one data as 0x0002, the packet length will become 18(12h)-14 and the length of this data list will become 4 bytes. That means that all the data placed on the data line are applied in 4-byte data value.

3.5 Link hibernation

The result of the generation of link hibernation packet is as in Figure 10, and the result was successful .

Zone Ⓐ is the link shutdown packet zone. Zone Ⓑ is the zone by standard according to the transmission of shutdown packet, and it is all zero field zone to send 64 Stb pulses after the transmission of CRC value to MSB is finished. . Ⓒ~Ⓓ show that the host deactivates MDDI_Data0 and MDDI_stb driver and enters into the sleep mode status.

Ⓔ shows that after the host driver is completely activated, MDDI_Data0 operates in Logic Level 1. The host sends a pulse to the client during a 150 MDDI_Stb cycle. Continuously, the host is activated Ⓕ zone, and MDDI_Data0 operates in Logic level 0, the host sends a pulse to the client during a 50 MDDI_Stb cycle.

It was confirmed through this test that the most appropriate timing presented by standard for the step to wake up the client in the hibernation mode is to implement with configuration in FPG. It's because the program cannot guarantee accurate delay time.

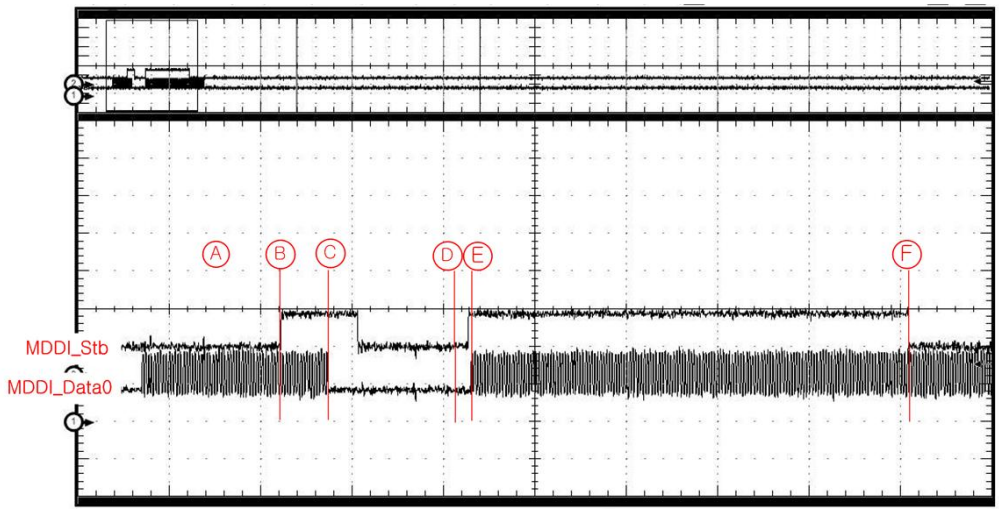

Figure 10. The output waveform of link hibernation

5. Conclusion

In this thesis, the method of implementing with software that could minimize the packet expandability and hardware design rather than the complicated FPGA method which requires many circuits was proposed to creating MDDI protocol packet required for the display device.

Images up to XGA-level can be transmitted in the Type I interface.

The packet created with software on the hardware where microprocessor and FPGA were tested in connection with LCD Driver chip built-in with MDDI Client module (developed by Samsung Electronics).

The image was displayed on LCD using Register Access Packet, video stream packet, Link Shutdown Packet, Sub-Frame Header Packet and Hibernation wake-up implemented by the main test and it was confirmed that all packets were implemented perfectly. Therefore, for the implementation of MDDI protocol for mobile systems, the flexibility to create various packets are proposed by MDDI standard with software any time and the effect of hardware cost reduction are secured.

This study could be applied to ASIC design[4]of MDDI-based high speed serial data transmission, and MDDI driver chips added with keyboard interface or audio interface packets presented by standard will be released in future with improved performance. Studies on various interfaces using these chips should be carried out in future.

References

[1] F. Zarkenshvari, "An Overview of High-speed Serial I/O Trends, Techniques and standards", Proc. IEEE CCECE-CCGEI, **(2004)** May.
[2] Digital Visual Interface DVI, Revision 1.0, DDWG, **(1999)**.
[3] IEEE Standard for Low-Voltage Differential Signals(LVDS) for Scalable Coherent Interface, 1596.3 SCI-LVDS Standard, IEEE std. 1596.3-1996, **(1996).**
[4] C. K. Park, "Experimental Investigation of Differential Line Inductor for RF Circuits with Differential Structure", Journal of Information and Communication Convergence Engineering, vol. 9, no. 1, **(2011)**, pp. 11-15.
[5] S. G. Cho, J. J. Hwang and N. J. Kwak, "Visible Distortion Predictors Based on Visual Attention in Color Images", Journal of Information and Communication Convergence Engineering, vol. 10, no. 3, **(2012)**, pp. 300-306.
[6] VESA, VESA Mobile Display Digital Interface Standard, Ver.1.2, **(2008)**.
[7] SAMSUNG Electronic Co., 5PC210 RISC Microprocessor Users Guide. Revision 0.20, **(2010)**.
[8] ALTERA, LVDS SERDES Transmitter / Receiver Megafunction User Guide, **(2011)**.

Authors

Jong Moon Kim

He received the M.S. degrees from the Department of Computer Engineering of Paichai University, Korea in 2012. From 1992 to 2003, he worked for Elcomtech System CO., Ltd. Inc. as a CEO. Since 2003, he has worked in ELCOMTECH CO., Ltd. Inc. as a CEO. He is currently a Ph.D. in Department of Computer Engineering of Paichai University. His current research interests include Digital Multimedia Broadcasting, Internet Protocol Television, MPEG.

Chang Su Kim

He received the B.S., M.S., and Ph.D. degrees from the Department of Computer Engineering of Paichai University, Korea, in 1996, 1998, and 2002, respectively. From 2005 to 2012, he worked for the Department of Internet at Chungwoon University as a professor. Since 2013, he has worked in the Department of Computer Engineering at Paichai University, where he now works as a professor. His current research interests include multimedia document architecture modeling, web 2.0, and the semantic web

Hoe Kyung Jung

He received the B.S degree in 1987 and Ph. D. degree in 1993 from the Department of Computer Engineering of Kwangwoon University, Korea. From 1994 to 2005, he worked for ETRI as a researcher. Since 1997, he has worked in the Department of Computer Engineering at Paichai University, where he now works as a professor. His current research interests include multimedia document architecture modeling, information processing, information retrieval, and databases.

