DESIGN AND IMPLEMENTATION OF A PHOTO CARD READER BY A HIGH-INTEGRATION CHIP SOLUTION

Ying-Wen Bai  
Department of Electronic Engineering  
Fu Jen Catholic University  
Taiwan, R.O.C.  
e-mail: bai@ee.fju.edu.tw

Chang-Chih Liu  
Department of Electronic Engineering  
Fu Jen Catholic University  
Taiwan, R.O.C.  
e-mail: a9050522@st2.fju.edu.tw

ABSTRACT

In this paper, we propose a design and implementation of a photography card reader by utilizing a high integration chip solution. The system uses the two machines: one is a popular card reader and the other is a photo player. To reach a higher level of performance and system flexibility, we integrate many independent components into a single chip and provide a system on chip (SOC) solution. As a result of the high pixel count of the digital still camera (DSC), the system can support up to a 4096x4096 picture. The independent components include memory card access modules, an USB2.0 bridge, a microprocessor (8051), a JPEG decoder, and an image processing unit (IPU), respectively. The JPEG decoder module is designed by pipeline architecture that has the advantage of low latency and high throughput. In the practical measurement for the prototype, our design can provide high performance and a low cost solution for potential consumer electronics.

Key Words: Photo Card Reader, Digital Still Camera, Interface and JPEG Decoder

1. INTRODUCTION

As information appliances have developed rapidly, various information devices are quite popular. Recently, the marketing of digital still camera (DSC) has been growing at very high rates [1]. This product has fulfilled new market need such as memory card readers, digital photo printers and photo players. Thus, we propose a design and implementation of a photography card reader by utilizing a high-integration chip solution that allows users to obtain the real-time displaying and browsing of pictures digitally captured and stored on CompactFlash (CF), SmartMedia (SM), Secure Digital (SD) or MultiMedia Card (MMC). In addition, these pictures taken by a DSC can be shown on TVs and computers by use of the corresponding interfaces. Currently, there are no low-cost consumer electronic products for a photo card reader with both TV and PC interfaces [2]. High-resolution pictures require an especially longer period of time than real time processing and displaying for picture zooming by means of a low-cost implementation [3,4].

Our design provides a total solution by means of an Application Specific Integrated Circuit (ASIC) and integrated SOC techniques. It contains five major intellectual properties (IP), in addition to the data-switching module that provides the data transfer between two of the devices by means of a 16-word buffer. The other modules are the memory cards access module, the USB2.0 module, the microprocessor unit (MPU), the JPEG decoder, and the image processing unit (IPU), respectively. Because this design is a prototype of an ASIC, the built-in hardware modules deal with the greater part of the system computation. So it has a higher performance and lower power consumption. In addition, the system is embedded an MPU8051, so it increases the system’s flexibility.

The rest of this paper is organized as follows. In Section 2, a brief overview of the system design is discussed. In Section 3, the performance measurement of the system is presented. The last section draws some conclusions.

2. DESIGN AND IMPLEMENTATION OF THE SYSTEM

Figure 1 shows the block diagram of our design that can be divided into two parts, one is a digital photo player, the other one is a popular card reader which allows a PC to access a memory card through the USB bridge.

The main purpose of a photo player is to show the pictures stored on various memory cards. These pictures are decompressed by a JPEG decoder and processed by an image processing unit according to the user’s setting. The maximum JPEG image size is assumed to be 4096x4096, as a result of universal resolution of the DSC is currently over three million pixels. But the resolution of the TV is only about 640x480, so there is an image-processing machine in the IPU module that can zoom in an input image to a fixed level of the TV resolution. The maximum range of the zoomed-image size of system is 1024x1024. Detailed descriptions of our design are given in Section 2.1 to 2.5.
2.1 Interface module of memory cards

The interface module of the memory cards provides an interface for accessing the SM/CF/SD/MMC flash cards [5-8]. Each memory module complies with the specification of the corresponding memory card interface. Our design supports two internal channels for transferring commands and data from and into the USB bulk-only module and the MPU module, respectively. When the USB cable is plugged in and the devices are selected by the USB host, the memory card’s interface processes only the commands from the USB bulk-only module. In this case, the device acts as the PC SM/CF/SD/MMC card reader. If the device is not selected by the USB host, the device works as a stand-alone. In this case, the interface only processes the commands from the MPU and the device acts as a picture player that can read the JPEG files from the various memory cards, decompress the JPEG files, and then display the pictures on the TV.

2.2 Microprocessor Unit (8051)

The system has an embedded turbo 8051 processor which performs the following tasks: initializes the device, sets internal registers according to the operation modes, provides GPIO for supporting the remote control, realizes the file system in the memory cards, reads JPEG file data from the memory cards and sends them to the JPEG decoder, and, if needed processes the commands from the USB interface of a PC.

2.3 JPEG Decoder Module

JPEG, the Joint Photographic Expert Group, is a standardization body that produces standards for continuous tone image coding. It is a commonly used digital image compression algorithm officially known as ISO Standard 10918-1. The JPEG Decoder module serves to decompress a JPEG file interchange format (JFIF) file that is read from the memory cards by the MPU (8051). The module works on the simplest and most widely used algorithm known as baseline JPEG [9].

Figure 2 shows the block diagram of the JPEG decoder module, which is the kernel of the system. To reach not only low latency but also high throughput in this design, we use the implementation method of the rapid Inverse Discrete Cosine (IDCT) [10]. The 2-D DCT processes can be regarded as a separable transform and it can be expressed in matrix notation (the row-column decomposition) as two 1-D DCT computations. The JPEG PARSER sub-module is responsible for recognizing the header marker in decompressed data stream. The Huffman table extraction module is responsible for re-building the Huffman table that could be changed by extracting the useful form of the Huffman tree from the data stored in the file.

The VLD sub-module will reconstruct image data by use of the variable-length decoding method. The de-quantize module multiplies reconstructed data by the corresponding value in a quantization table. The inverse “ZigZag scan module” can rearrange the processed data into 8 pixel by 8 pixel data blocks. Finally the IDCT approximately recovers the original 8x8 data block. The output of the JPEG decoder module is YUV 4:2:2 or YUV 4:2:0 or gray format and sends them to the “strip module” in 8x8 block sequences.

2.4 USB 2.0 Bridge

USB provides an expandable, hot-pluggable plug and play serial interface that ensures a standard and a low-cost connection for peripheral devices. The external USB2.0 PHY is a high-speed USB2.0 transceiver macrocell interface (UTMI). The USB 2.0 standardization can support up to 480 Mbps for high-speed devices. The USB2.0 module of our design is also a bridge that allows a PC to access data stored in memory cards by using an USB interface. The module conforms to USB2.0 specification [11], and is also compliant to USB mass storage class bulk only protocol and USB storage class specification Version 1.0. In addition to the standard function of the endpoint 0 (control pipe), it supports two vendor commands for USB mass storage class and
provides one bulk-input pipe and one bulk-output pipe for the mass data transfer.

### 2.5 Image Processing Unit (IPU)

The image processing unit is to zoom a decompressed picture of any resolution based on the setting of the zoom factor. It consists of a zoom-out module and a zoom-in module. The zoom-out ratio can be 1/128, 2/128, 3/128, ..., 128/128 of the original image size. After a picture is zoomed-out, it can be enlarged at a ratio of 2 by the zoom-in module.

The image data flowchart is shown in Figure 3, in which the data flow segment (1) shows the JPEG decoder output decompressed data flowing into the strip module. The strip module rearranges the order of block-oriented data that is generated by the JPEG decoder into a line-by-line order. The output of JPEG decoder can be YUY 4:4:4 or YUV 4:2:2 or YUV 4:2:0 or gray format, but the strip module will convert the data format into YUV 4:2:2 before writing any data to the strip buffer. The data flow segment (2) in Figure 3 shows that the mem_ctrl module doesn’t send the decompressed data into the zoom-out module until the strip module has written a 16-line data in YUV 4:2:0 or written a 8-line data in the other format.

The function of the zoom-out module is to zoom a decompressed picture of any resolution based on the setting of the zoom-factor. The zoom algorithm is shown as Figure 4. If we consider a reduction factor equal to L/M (M=128), with L, M integers and L<M, we will obtain an output picture in which the L pixels contain the same information as the M pixel of the original picture [12]. Figure 4 shows the zooming of an input picture containing 8 pixels, for the reduction relation, which is to 3/8. When considering a lower cost system, one feature of our design is that the system can zoom a picture into a 1024x1024 resolution no matter what size the decompressed picture is. The zoom-out module will process different components (Y, U, V) in both vertical and horizontal direction in accordance with the previous algorithm. So the module has embedded three 1024-byte RAMs for Y component, three 512-byte RAMs for U component, and three 512-byte RAMs for V component, respectively. The data flow segment (3) in Figure 3 shows that the zoom-out module sends the BT.601 format data to the zoom-in module.

The zoom-in module can enlarge the input picture at a ratio of 2. Because the image size that is processed by the zoom-out is no bigger than 1024x1024 pixels, the zoom-in module has an embedded 1024-word RAM. After the zooming process, the zoomed image may be cropped such that the image width and image height is a multiple of an 8-pixel. This zooming process will facilitate the rotation process performed in the next stage.

The data flow segment (4) in Figure 3 shows that the zoom-in module sends the video data into the memory control module. The rotating module can rotate the image right in 0, 90, 180, 270 degree, if set by the users. The data flow segment (5) as shown in Figure 3 indicates that mem_ctrl module sends data to the rotating module, and the segment (6) indicates rotated data stored back to the mem_ctrl module.

The video data flowchart is as shown in Figure 5, in which the video out module is responsible for sending the current frame data in CCIR601 format to the video encoder or in CCIR656 format to a I/O pin direct chip. The module also supports both PAL and NTSC systems. When the video out module will need the video data from a line, it will send a request signal to the mem_ctrl module. The video data is read from one bank of the SDRAM shown in the data flowchart (a), then sent to the video out module shown as the data flowchart (b) as shown in Figure 5.

---

**Image Processing Unit**

The function of the zoom-out module is to zoom a decompressed picture of any resolution based on the setting of the zoom-factor. The zoom algorithm is shown in Figure 4. If we consider a reduction factor equal to L/M (M=128), with L, M integers and L<M, we will obtain an output picture in which the L pixels contain the same information as the M pixel of the original picture [12]. Figure 4 shows the zooming of an input picture containing 8 pixels, for the reduction relation, which is to 3/8. When considering a lower cost system, one feature of our design is that the system can zoom a picture into a 1024x1024 resolution no matter what size the decompressed picture is. The zoom-out module will process different components (Y, U, V) in both vertical and horizontal direction in accordance with the previous algorithm. So the module has embedded three 1024-byte RAMs for Y component, three 512-byte RAMs for U component, and three 512-byte RAMs for V component, respectively. The data flow segment (3) in Figure 3 shows that the zoom-out module sends the BT.601 format data to the zoom-in module.

The zoom-in module can enlarge the input picture at a ratio of 2. Because the image size that is processed by the zoom-out is no bigger than 1024x1024 pixels, the zoom-in module has an embedded 1024-word RAM. After the zooming process, the zoomed image may be cropped such that the image width and image height is a multiple of an 8-pixel. This zooming process will facilitate the rotation process performed in the next stage.

The data flow segment (4) in Figure 3 shows that the zoom-in module sends the video data into the memory control module. The rotating module can rotate the image right in 0, 90, 180, 270 degree, if set by the users. The data flow segment (5) as shown in Figure 3 indicates that mem_ctrl module sends data to the rotating module, and the segment (6) indicates rotated data stored back to the mem_ctrl module.

The video data flowchart is as shown in Figure 5, in which the video out module is responsible for sending the current frame data in CCIR601 format to the video encoder or in CCIR656 format to a I/O pin direct chip. The module also supports both PAL and NTSC systems. When the video out module will need the video data from a line, it will send a request signal to the mem_ctrl module. The video data is read from one bank of the SDRAM shown in the data flowchart (a), then sent to the video out module shown as the data flowchart (b) as shown in Figure 5.

---

**Figure 3. Image data flowchart of the IPU.**

**Figure 4. Relationship of the image zooming (i.e. with a reduction factor = 3/8)**
The video out module will receive the data and transfer into byte-oriented data, then send them to on the screen display (OSD) module shown as the data flowchart (c). The OSD module can integrate the image data with the OSD picture data into a picture. The OSD picture is generated by a sequence of character-images stored into RAM and ROM. The OSD will take four clocks (i.e. latency time is 4), and send the result back to the video out module shown as the data flowchart (d). Finally, the video output will send the standard video data stream to the TV encoder or the external I/O as shown in the data flowchart (e).

![MemCtrl Module](image1)

**Figure 5.** Video data flowchart of the IPU.

The memory control module provides three write-ports and three read-ports for accessing the SDRAM. The service priority order is: the video out module, the zoom-out module, the zoom-in, the strip module and the rotate module, respectively. The SDRAM is 4MWords and is accessed in 27MHz with the burst length set to 8 lines (16 bytes). As shown in Figure 6, the layout of the SDRAM is divided into two parts: strip buffer and frame buffer. A strip buffer can store 8 lines of decompressed data in YUV 4:2:2 format and 16 lines of decompressed data in YUV 4:2:0 format. The strip buffer starting address is 0, and its size is 256KWord (in the case of 16 4096-pixel lines). For the frame buffers, the remaining part of the SDRAM is used as frame buffers, each of which occupies 1MWord space. For a 4MWord SDRAM, there are three frame buffers. A frame buffer is for storing the frame data displayed on the TV. The frame image size is no bigger than 1024x1024 pixels. For the simplicity, we allocate 1KWord of space for each line in a frame. Therefore, a frame buffer is allocated with 1Mword of space.

![MemCtrl Module](image2)

**Figure 6.** The layout of the memory configuration

### 3. The Performance Measurement of the System

In the system design, we used a high-level Verilog HDL (Hardware Description Language) to implement every module. We then ran RTL (Register Transfer Level) simulation under the ModelSim simulated environment. In order to demonstrate our design using a Field Programmable Gate Array (FPGA), we had used synthesis, placement and routing tools to generate FPGA download files for testing and verification.

When the system is plugged into in the PC via USB cable, our system works on the popular card reader mode. The memory cards are regarded as four removable disks in the computer. We use HDBENCH tools to analyze the speed of reading and writing as shown in Table 1. Of all memory cards, the reading speed for the CF card is up to 3.6Mbytes per second; the speed for the MMC card is the slowest. The difference in performance depends on a specialized accessing manner for each memory card.

When disconnecting the USB cable, the system looks like a photo player that shows the photo on a TV. As the memory control module must send the video data stream to a TV in real time, the video out module has to use the 78% of the memory bandwidth in both a NTSC and a PAL system. The remaining bandwidth is used for other modules, so the memory bandwidth is a bottleneck of the whole system. In Table 2, the testing condition (full bandwidth) indicates that the system performance works under the condition of the disable TV. The testing condition (remaining bandwidth) indicates that the system is working in a normal condition.
Table 1. The performance measurement for accessing memory cards

<table>
<thead>
<tr>
<th>Memory Card</th>
<th>Test Capacity</th>
<th>Test</th>
<th>Read</th>
<th>Random Read</th>
<th>Write</th>
<th>Random Write</th>
<th>Unit</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>CF</td>
<td>16MB</td>
<td>3612</td>
<td>1201</td>
<td>3567</td>
<td>1194</td>
<td>KBytes/s</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SM</td>
<td>16MB</td>
<td>1484</td>
<td>782</td>
<td>1439</td>
<td>773</td>
<td>KBytes/s</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SD</td>
<td>16MB</td>
<td>1389</td>
<td>482</td>
<td>1278</td>
<td>456</td>
<td>KBytes/s</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MMC</td>
<td>16MB</td>
<td>982</td>
<td>546</td>
<td>534</td>
<td></td>
<td>KBytes/s</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Test software: HDBENCH3.40
USB 2.0 Host Controller: NEC D720100AGM USB2.0
Test system: Windows XP

Table 2. The area allocation of sub-modules.

<table>
<thead>
<tr>
<th>Module Name</th>
<th>Logic Cell</th>
<th>Embedded Ram/Rom (Byte)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jpeg Decoder</td>
<td>5790</td>
<td>256x8 + 64x16</td>
</tr>
<tr>
<td>mem_ctrl</td>
<td>1003</td>
<td>(768x8)x2</td>
</tr>
<tr>
<td>zoom-out</td>
<td>2485</td>
<td>(1024x8)x3 (512x8)x3x2</td>
</tr>
<tr>
<td>zoom-in</td>
<td>819</td>
<td>1024x16</td>
</tr>
<tr>
<td>rotate</td>
<td>2216</td>
<td></td>
</tr>
<tr>
<td>video_if</td>
<td>1520</td>
<td></td>
</tr>
<tr>
<td>OSD</td>
<td>1316</td>
<td>3072x12(ROM) 3072x12(RAM)</td>
</tr>
<tr>
<td>register</td>
<td>3893</td>
<td></td>
</tr>
<tr>
<td>SmartMedia Interface</td>
<td>1618</td>
<td></td>
</tr>
<tr>
<td>SD/MMC Interface</td>
<td>1238</td>
<td></td>
</tr>
<tr>
<td>CompactFlash Interface</td>
<td>1048</td>
<td></td>
</tr>
<tr>
<td>mp3(8051)</td>
<td>2631</td>
<td>8192x8</td>
</tr>
<tr>
<td>DataSW</td>
<td>622</td>
<td></td>
</tr>
<tr>
<td>USB2.0 Bridge</td>
<td>2294</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>30132</td>
<td>220,160</td>
</tr>
</tbody>
</table>

Table 2 shows the area of every sub-module, which is about 30,000 logic cells altogether, and the whole chip 220,160 bits of embedded memory, which include SRAM and ROM. In Figure 7, the image processing unit (IPU) occupies 50% area of the system, and JPEG decoder occupies 20%, memory cards interface module occupies 12%, MPU (8051) module occupies 9%, and USB2.0 bridge occupies 8%, respectively.

Table 3. The latency time between the “DataSW” module and the “Rotate” module.

<table>
<thead>
<tr>
<th>Test condition (1): Full Bandwidth</th>
<th>Test condition (2): Remained Bandwidth</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image size</td>
<td>Zoom-down Factor (H/V)</td>
</tr>
<tr>
<td>640x480</td>
<td>0.3</td>
</tr>
<tr>
<td>1024x768</td>
<td>0.78</td>
</tr>
<tr>
<td>1280x960</td>
<td>1.2</td>
</tr>
<tr>
<td>1600x1200</td>
<td>1.92</td>
</tr>
<tr>
<td>2048x1536</td>
<td>3.14</td>
</tr>
<tr>
<td>2272x1704</td>
<td>3.78</td>
</tr>
<tr>
<td>2560x1920</td>
<td>4.92</td>
</tr>
<tr>
<td>3072x2048</td>
<td>6.29</td>
</tr>
</tbody>
</table>

As result of the FPGA testing, the system meets the specifications of our design, and the operation speed of the prototype system can also meet the needs of our design. In the practical measurement as shown in Table 3, we input the various size pictures captured by the DSC, and measure the latency time between the “DataSW module” and the “rotate module” in both condition (1) and (2). The testing condition (1) of the system uses full bandwidth that indicates the system performance when working with the disabled TV. The testing condition (2) that uses the remaining bandwidth indicates that the system is working under a normal condition. We have zoomed the original size into the best size for TV resolution (640x480) by setting both the horizontal zoom-down factor (H/128) and the vertical zoom-down factor (V/128). In Figure 8, we learn that our design takes 4.5 seconds for the popular size of the pictures, which has a resolution of about 3.4 million pixels. It only takes less than 6 seconds for 6 million pixels of a picture.
4. CONCLUSIONS

In this paper, we have designed and implemented a photography card reader by a high integration chip solution. To attain both higher performance and system flexibility, we integrate many independent components into a single chip. So our design is also a system on chip (SOC) solution. As a result increasing pixel count of the digital still camera (DSC), the system can support up to a 4096x4096 picture that will extend the product’s life. The independent components of the system include memory card access modules, USB2.0 bridge, microprocessor (8051), JPEG decoder, and image processing unit (IPU), respectively. The JPEG decoder module has a pipeline architecture design that has the advantage of low latency and high throughput. In the practical measurement for the prototype, our design can provide high performance and low cost solution for potential consumer electronics. As result of the FPGA testing, the system performance meets the specifications of our design, and the operation speed of the prototype system can also meet the need of the design request. In the practical measurement of the latency time of the system, our design can finish the processing and displaying of a picture on a TV in keeping with the user’s habit no matter what the size of the picture is.

5. REFERENCES


