First Prize

Networking Remote-Controlled Moving Image Monitoring System

Institution: **National Chung Hsing University**

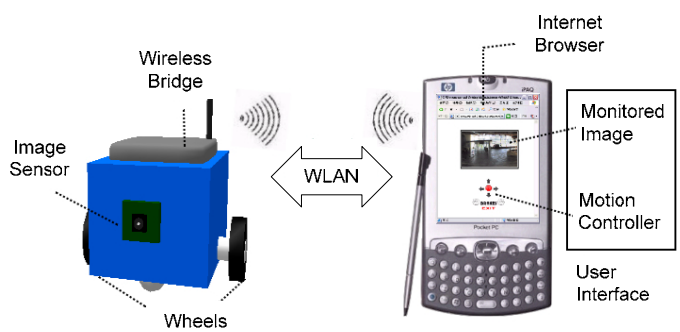
Participants: **Cai Jingtao**

Instructor: **Cai Qingchi**

Design Introduction

Our design focuses on establishing a real-time image monitoring system that can be operated under user commands to achieve real-time network monitoring. Users navigate a browser-like user interface in the manner of web browsing to remote control real-time image inspection. The application scope is security, home usage, and unmanned exploration. See Figure 1.

Figure 1. Motion Image Monitoring System



This design is different from fixed-monitor image inspection systems because it realizes integrated monitoring without any dead angle. For instance, parents can keep an eye on their kids using remote controls and a monitoring interface instead of using computer monitors at fixed locations in their homes.

The system design uses the Nios® II processor for implementing the basic peripheral components. It takes advantage of the powerful flexibility provided by the Nios II processor, as well as the support for real-time operating system (RTOS) to establish an integrated system. Additionally, the Cyclone™ FPGA supports low-cost performance by allowing hardware customization.

Function Description

The functional description of our design is as follows.

- *Real-time image acquisition*—Connect the CMOS image sensor with the programmable I/O (PIO) interface circuit to get the value of picture points (gray scale) and then compose RGB images with a simple de-mosaic algorithm.

- *Real-time image compression*—The algorithm compresses the image into JPEG format to improve the processing efficiency, and then uses hardware/software co-design to implement the Discrete Cosine Transform (DCT) algorithm in the JPEG compression system in hardware. Additionally, we designed a DCT algorithm as a custom peripheral meeting the requirements of the Avalon® bus standard. We used software to implement other algorithms.

- *Graphical user interface (GUI)*—The design uses a web server as a simple GUI in a web-browser format. The GUI monitors picture display and control interface. The common gateway interface (CGI) programs handle control and communication.

- *Motion control*—The mobile part of the design integrates the driver's integrated circuit (IC) and its circuitry with two DC motors on a custom-designed circuit board. The board also supports a pulse width modulator (PWM) component on the Nios II processor to control the motor through the PWM-signal output.

- *Wireless network transmission*—Because the Cyclone FPGA does not provide a wireless network interface, our design uses the available interface with a wireless bridge to accomplish wireless-network transmission.

- *System integration*—The system's motherboard is based on the Nios II development board and the Cyclone FPGA. We designed a circuit board to integrate the drive circuit and power supply. Further, we used a lead-acid battery as the system's power supply to meet embedded system application requirements.

Performance Parameters

The major performance parameters of the design depend on the frame rate of the JPEG encoder. The process of composing a frame includes:

1. Image acquisition: the PIO controls the CMOS image sensor to capture a 320 x 240 gray-level image.

2. Image processing: realize de-mosaic motion in an easy way to synthesize RGB images.

3. Image compression: carried out in accordance with the encoder output, in standard JPEG format.

In the JPEG standard baseline mode, our design uses hardware to implement the DCT algorithm. Software programs fill the other related performance functions. We implemented the JPEG compression in software by migrating the IJG Library. Then, the hardware/software co-design of the JPEG encoder is

complete. Finally, the compressed data is delivered to the remote user through the network transmission by reading and displaying the received JPEG images through a browser interface.

The following table shows the image of QVGA resolution (320 x 240) processed by our system as well as the time spent in various processes.

Process	Image Fetching	De-Mosaic Algorithm	JPEG Encoder
Time	0.2 sec	0.11 sec	0.83 sec

For testing purposes, the design uses the Cyclone FPGA with an operating frequency of 50 MHz and 0.87 FPS efficiency.

Because the DCT operations are time consuming, the DCT is more easily created in hardware when using the JPEG-compiled code system. However, planar DCT can be realized with current high-speed algorithms such as the IJG Library that provides fixed-point and floating-point DCT functions. However, the DCT is still more compute-intensive than the JPEG operation.

The following table shows time spent on processing one block (8 x 8 pixels) per frequency period. Software refers to the fixed-point fast DCT function provided by the IJG library and hardware is the DCT component that is executed through the Avalon bus interface.

Note: Data processing includes execution time of a single measurement function and accounts for data transfer into memory.

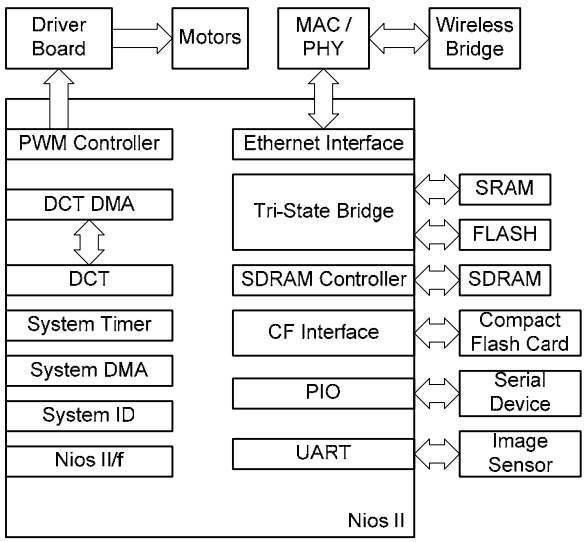
Method	Software	Hardware
Time	4,000 clocks	450 clocks

Using the table above, the design successfully implements the JPEG image compression system on the Nios II processor using hardware/software co-design, freeing up system resources during DCT computation.

Design Architecture

Figure 2 shows the system architecture.

Figure 2. System Architecture

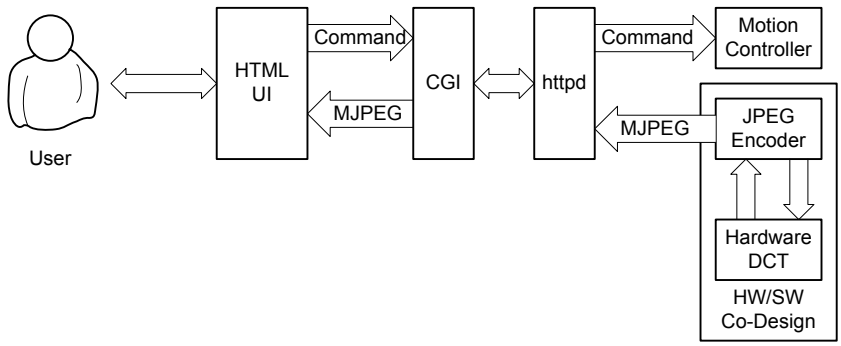


The main system is based on the Nios II Development Board, Cyclone Edition, and we used the peripherals of the development board in this design. The system comprises the following functions:

- It adopts Nios II/f (fast) core and adds hardware multiplier/divider/shifter selectively.
- It stores the hardware and software programming files into flash.
- It adopts SDRAM as memory storage for bigger programs.
- Storage devices of the system use a compact flash (CF) card and two SRAM devices.
- UART interface is used as system-console interface.
- It applies a PHY/MAC chip of the development board to implement wireless-network transmission, connecting with the wireless bridge through an RJ-45 network interface.
- For image acquisition, the system connects to an external CMOS image sensor and uses the PIO to realize a picture-element read interface.
- It uses hardware to implement the DCT, and we designed custom peripherals to work with the Avalon bus and added DMA for data read/write.
- In the motion mechanism, we added a PWM controller to output the PWM signal to the IC motor drive and implement the normal/reverse control for the DC motor.

Figure 3 shows the software flow chart.

Figure 3. Software Flow Chart

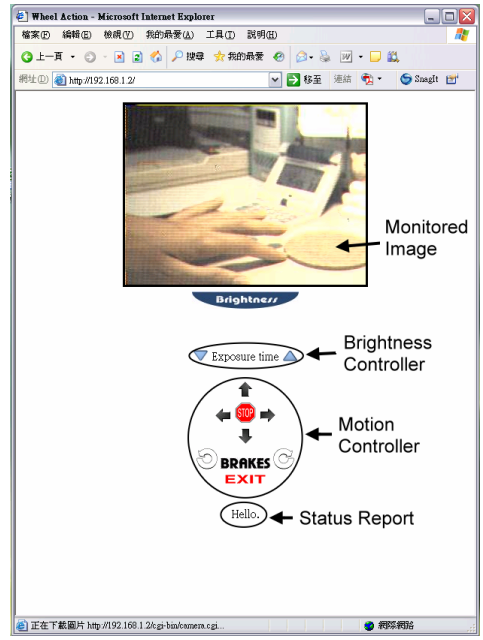


A major feature of the design is its simple user interface. The remote user merely needs to connect to the self-defined IP address of the system using the network browser. The web server then provides service for the user, including JavaScript programs to handle real-time image monitoring of pictures in motion JPEG mode.

Additionally, the mobile control interface can execute forward, backward, and stop motions easily. The software design and development is based on the Microtronix Nios II Linux distribution version 1.3 RTOS, using a CF card to qualify the system as a read/write file.

The communication between web control interfaces is created with CGI programs. When the user executes an image-monitoring function, the system acquires an image immediately, performs JPEG compression with the combination of software and hardware, and delivers the images to the web. At the same time, the user can control the image mobility and adjust the brightness to achieve real-time and interactive motion-monitoring function. Figure 4 shows the GUI.

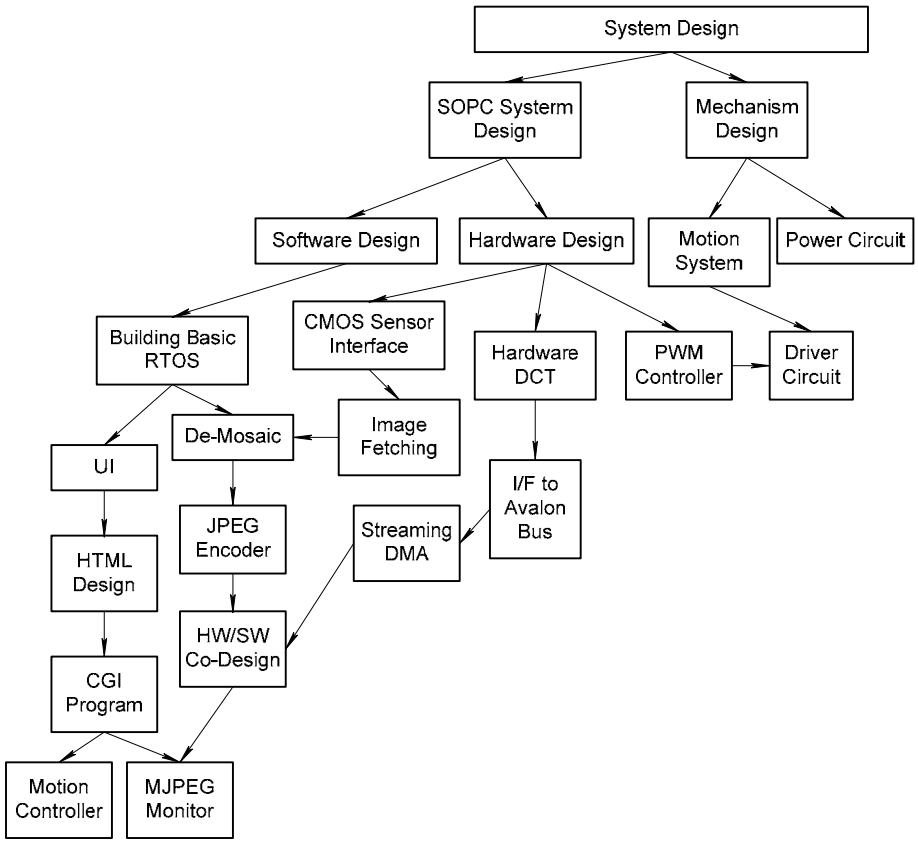
Figure 4. GUI



Design Methodology

Figure 5 shows the detailed design steps, various system tasks, and their inter-working:

Figure 5. Design Steps Flowchart



The flowchart is based on a top-down integrated system design concept, which is divided into two parts: system design and architecture design.

- System-on-a-Programmable-Chip (SOPC) System Design:

  The hardware design includes configuring the basic Nios II SOPC environment and customizing peripherals that are compatible with the Avalon bus. These peripherals include CMOS image interface circuit, hardware DCT, and PWM controller.

  The software module comprises RTOS tasks including the file system routines, the GUI design, and various system tasks. Refer to Figure 5 for more details.

- Architecture Design:

  The main mobile device comprises a chassis with two Maxon DC motors and wheels to handle movements such as forward, backward, and circumrotation (in the manner of differential rotation).

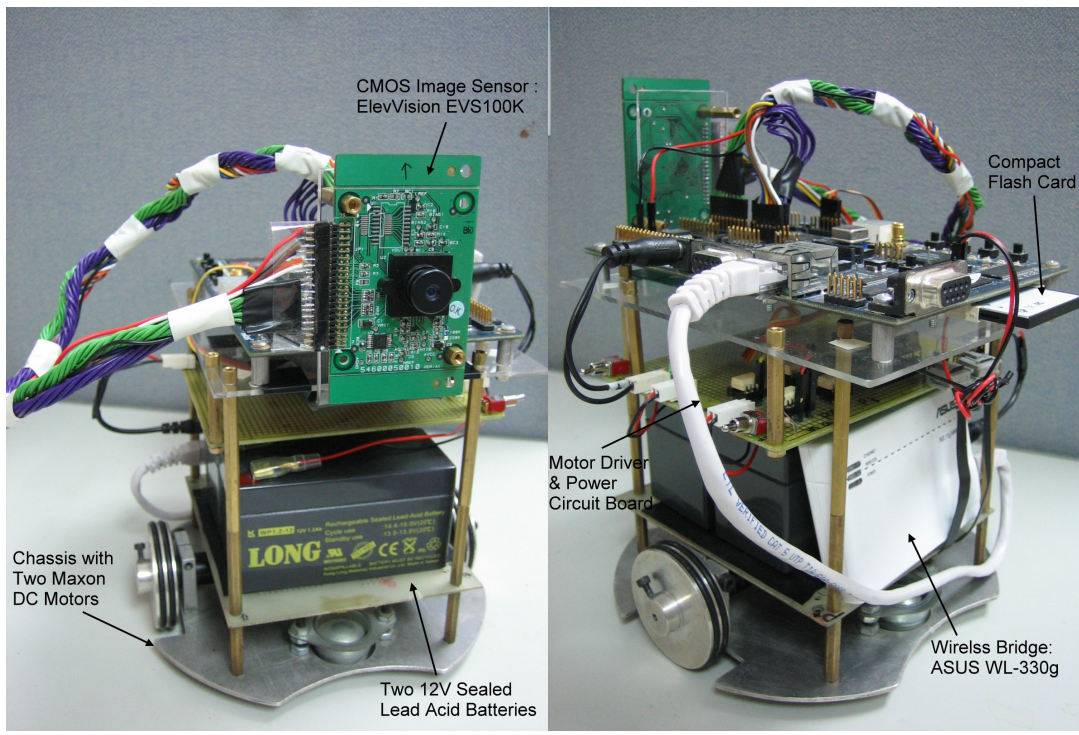
A motor driver and a power circuit comprises driver and optocoupler devices. We designed a board that integrates all IC and driver circuits.

The system's power supply uses two 12-V lead-acid secondary batteries. This circuit features a simple filtering-protection circuit, switch, and connection.

- System Integration:

  Figure 6 shows a fully integrated system.

Figure 6. Fully Integrated System



Front View **Back View**

Design Features

Using the Nios II processor, you can simplify the customized SOPC design concept and verify your designs realistically. Furthermore, FPGA resources can be applied creatively to implement various functions depending on user's special requirements. These requirements could include functions such as hardware DCT and a PWM controller. In this way, the designer can integrate most functions into one FPGA.

The core module of the design is the JPEG encoder, which we implemented through hardware/software co-design. This design also showcases the FPGA advantage at its best. We created the DCT in hardware to realize the JPEG encoder and designed a custom interface that followed Avalon bus requirements. Next, we wrote the DCT output data in memory for processing in software with streaming DMA to save operation time.

The general features of the system design are described in the following three points:

Technical Standards

We have successfully used the Nios II processor to fulfill the IP camera function, thus meeting market needs. Additionally, we have added the remote control mobile function for extra benefit.

Our system boasts a highly integrated design and features an easy-to-use GUI and RTOS environment.

We used nearly all peripherals of the Nios II development board. We integrated most functions into the low-cost Cyclone EP1C20 FPGA and the logic element (LE) utilization exceeds 90%.

Design Creativity

Our system integrates a mobile mechanism that can be controlled easily using existing fixed monitoring systems available in the market to produce an innovative design. This system is a balanced product between consumer electronics and a robot.

Adaptability

The no-dead-angle remote monitoring function in this design is flexible and can be used to handle many applications. Potential users include home, factory, and commercial establishments.

Our design's modular architecture is similar to the common IP camera. We implemented our system with the low-cost Cyclone FPGA. Moreover, the mobile system attachment is very economical and can be deployed in consumer applications requiring no-dead-angle remote control. Additionally, the system does not require expensive interface devices, and all its functions can be realized using semiconductor ICs, which can be mass-produced easily.

The outstanding features of the system design:

- *Easy-to-use GUI*—It can monitor the situation on-site through the web.
- *Mobility*—Control the car with a remote network and extend the monitoring scope of the original fixed-monitoring system to any location.
- *Utilize Nios II and FPGA features to implement the functionality, keeping the total cost lower than many competing solutions*—Meet the IP Camera functions based on market expectations with the low cost Nios II processor and Cyclone FPGA.
- *Design uses RTOS*—Adopt Microtronix Nios II Linux Distribution v1.3 as the OS system.
- *Custom peripheral for hardware acceleration*—Added a DCT custom peripheral to accelerate the JPEG encoder.
- *Two custom peripherals without hardware acceleration*— Added Altera's PWM controller for handling motor control functions. Added Microtronix's Compact Flash Component to access the CF card data.
- *Performs graphic acceleration/uses the DSP algorithm*—Through hardware/software co-design, managed to increase the frame rate of motion JPEG.
- *Used DMA as another master device on Avalon bus*—Acquired DCT-output data with streaming DMA.

- Greater than 90% LE/memory utilization.
- Connected over 15 main peripherals on Avalon bus.

Conclusion

When we used the Nios II processor to implement our design, we were very impressed with the convenience it provided. Our design benefited greatly from its features. For example:

- *Complete and coherent development environment*—The SOPC design process, software design, and verification were made easy by using the Nios II processor, thanks to Altera's convenient development tools.
- *Abundant and flexible peripheral support*—Peripherals can be fully integrated and peripheral components customized by designers easily and quickly.
- *Quick verification*—Because the system architecture is based on digital logic design, we were able to get clear and detailed information by simulation software during the system verification.

Altera's Nios II processor platform using FPGAs can be adapted for teaching, experimenting, and quick product development in many diverse applications.