FPGA Based Intruder Detection System Using CMOS Sensor

[1]Ashwin Belure, [2]J. S. Rajashekhar, [3]Basheer Ahmed

[1]Student: M.tech in Microelectronics & Control Systems
Dayananda Sagar College of Engineering, Bangalore-560078, Karnataka, India

[2]Asst. Professor, Dept of IT
Dayananda Sagar College of Engineering, Bangalore-560078, Karnataka, India

[3]Head, Operations,
SION Semiconductors, Bangalore-560078,Karnataka, India

Abstract

Humans have always felt very possessive of their belongings. During ancient times, inhabitants used to live in secure caves so that they would be protected from unwanted intrusion and from deadly animals. The advancement of civilization witnessed better and improved means of security system being implemented by humans. Technological revolution of modern age has resulted in concept of security finding widespread popularity. The intruder detection system that is described in this report aims to enhance the security in not only closed spaces like offices and homes but also in open areas.

This thesis proposes to develop an intruder detection system in which a continuous monitoring of the area under surveillance is done by taking the video of the place. The video is captured using a CMOS sensor in the form of frames. These frames are stored and compared and the error between any two frames is calculated. When there is no intrusion, the error is zero. When there is an intrusion, the error is displayed in an LCD display.

Keywords: *CMOS Sensor, Intruder Detection.*

1. INTRODUCTION

Surveillance systems with image recording functions become vital devices both in private and public places. The reason is that, according to the recorded image data, the police have resolved many serious robberies and criminal cases which are difficult in the past and the responsibilities of traffic accident cases have been clarified in terms of the saved image data as well. Therefore, the government has put the construction of image recording systems on top of the list of public security infrastructure. However, the surveillance system has usually been designed and installed in a way that constraints the system itself that can only monitor a fixed direction. Consequently, it would require many monitor systems in order to keep a large area under surveillance.

There have been two major approaches on real-time image tracking. The first approach is to track objects according to some special or predefined features. As study in [1], vehicles were tracked basing on images captured by traffic monitors; and in [2], in accordance with local binary pattern and skin color, human face could be tracked.

The second approach is to distinguish objects and background from images. From [3], a method was proposed to estimate the traffic flow by computing differences of images for the purpose of extracting object edges for vehicles tracking. A block-based motion estimation method was used to tracking a moving object in [4]. There have been many developments in the security systems in the recent past, especially in home security systems due to the ever increasing crime rate. The most common types of home alarm systems have certain devices which are installed on particular access points such as the doors and windows. Once the alarm system is activated, opening these access points will set off the alarm. But these are of limited use as they can only detect the intruder only when the particular access point is opened. It cannot detect if the intruder enters through any other point.

There are many moving object tracking algorithms. In [5], for instance, a particle filter concept was employed for object tracking according to similarity of color density function to predict the object position. A histogram based method was developed to process consecutive images for object tracking in [6]. In study [7], an adaptive block matching algorithm was proposed for tracking, and in [8], multiple objects tacking was examined by combining feature extraction and moving properties of objects. These methods emphasize high positioning rate and recognizing rate, but on the opposite side they need complex computation so they usually implemented only by PCs or embedded systems.

This article adopts the second approach of above mentioned two approaches to track an object by comparing images. The system will be developed by following the hardware and software co-design concept, and the resulting design is going to be implemented on a FPGA chip. The system not only equips with basic surveillance function, but also possesses of function of tracking objects. By controlling horizontal-vertical rotation machinery on which the camera being mounted, the system can track and capture suspected object in multi-angle, alarm in real-time, and provide the police with saved image data for further detecting reference.
The organization of this paper is as follows: insight of various blocks used section II. Section III deals with the i2c communication protocol between the master and the slave so as to configure the CMOS sensor. The description of CMOS image sensor configuration that is used to capture image is shown in Section IV. Section V explains the image dataflow control and Bayer's pattern used to convert color image to black and white and gives the description of image dataflow control and Bayer's pattern block diagram and the equation used to convert color to black and white. Section VI gives the description of SDRAM controller that is used to store image frames. Section VII gives the description of intrusion detection block and different modes that are used. Finally Section VIII deals with the VGA monitor controller with the help of which an image can be displayed on the VGA monitor considering blanking and other parameters and the description of the LCD.

2. BLOCK DIAGRAM

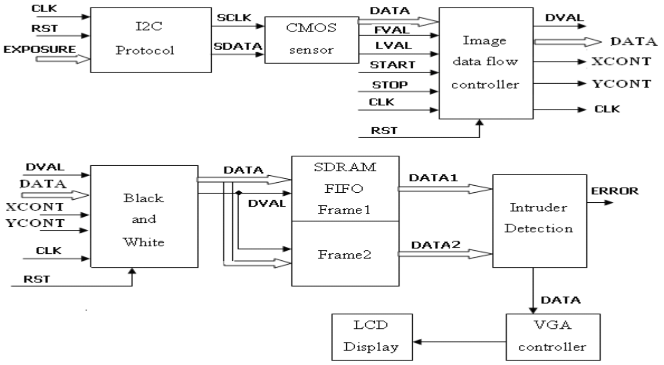


Fig 1: System Design.

In this system, the area under surveillance is continuously monitored by taking the video of the place. This is accomplished using the CMOS sensor, which captures the image frames. The CMOS sensor is configurable to the desired resolution and exposure settings. These settings are done by programming the mode register of the sensor. In order to program the mode register, I2C protocol is used. Using this, the programming data is transferred to the sensor. The parallel data available is converted into serial data and is transferred to the sensor. The programmed sensor captures data continuously.

Since the sensor captures video continuously, it is necessary to control the flow of data to the memory where it is stored. Hence the Image Data Flow Controller is used. It specifies the start and stop to store data. It regulates the flow of data to the memory. The data is stored only when start signal is activated. It is continuously done till stop signal is encountered. The data is stored in the form of black and white image. For this, the black and white converter module is used.

The captured image frames are stored in SDRAM. This is available on the ALTERA development board. It is of size 8MB. The memory can store two frames at a time. Hence it is partitioned into two and the data is read from or written in to the memory. To monitor these operations, the SDRAM controller is used. It generates read and write signals and also tracks which frame is being read or written.

The image frames stored in the SDRAM are compared and the error signal is generated when any two frames differ. When there is no intrusion, the error is zero. When there is an intrusion, the error is displayed in an LED display on the development board.

3. I2C COMMUNICATION PROTOCOL

Communication is established and 8-bit bytes are exchanged, each one being acknowledged using a 9th data bit generated by the receiving party, until the data transfer is complete. The bus is made free for use by other ICs when the 'master' releases the SDA line during a time when SCL is high. Apart from the two special exceptions of start and stop, no device is allowed to change the state of the SDA bus line unless the SCL line is low. If two masters try to start a communication at the same time, arbitration is performed to determine a "winner" (the master that keeps control of the bus and continue the transmission) and a "loser" (the master that must abort its transmission). The two masters can even generate a few cycles of the clock and data that 'match', but eventually one will output a 'low' when the other tries for a 'high'. The 'low' wins, so the 'loser' device withdraws and waits until the bus is freed again. There is no minimum clock speed; in fact any device that has problems to 'keep up the pace' is allowed to 'Complain' by holding the clock line low. Because the device generating the clock is also monitoring the voltage on the SCL bus, it immediately ‘knows’ there is a problem and as to wait until the device releases the SCL line.

Data transfers follow the format shown in below Fig. After the START condition (S), a slave address is sent. This address is 7bits long followed by an eighth bit which is a data direction bit (R/W) - a ‘zero’ indicates a transmission (WRITE), a ‘one’ indicates a request for data (READ). A data transfer is always terminated by a STOP condition (P) generated by the master. However, if a master still wishes to communicate on the bus, it can generate a repeated START condition (Sr) and address another slave without first generating a STOP condition. Various combinations of read/write formats are then possible within such a transfer.

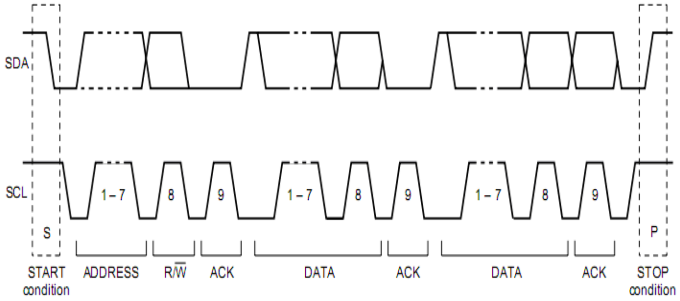


Fig 2: A complete data transfer.

I2C is used for sending the parallel data which is the input for I2C into serial data at scl. Data is given in parallel which is used to configure CMOS sensor. Ack is the input for the I2C from the CMOS sensor which is high after receiving data from I2C to the sensor. After getting an ack next data is passed to the CMOS sensor. Exposure is used for controlling the lighting in room (example in camera we have different modes like night mode, sunlight mode etc). Exposure data is given by using switches in FPGA board.

4. CMOS SENSOR CONFIGURATION

MT9M011 is an SXGA-format, 1/3-inch CMOS active-pixel digital image sensor with an active imaging pixel array of 1,280H x 1,024V. It incorporates sophisticated camera functions on-chip such as windowing, column and row skip mode, and snapshot mode. It is programmable through a simple two-wire serial interface and has low power consumption. An on-chip analog-to-digital converter (ADC) provides 10 bits per pixel. FRAME_VALID and LINE_VALID sig- nals are output on dedicated pins, along with a pixel clock that is synchronous with valid data. A flash out- put signal is also available to synchronize external light sources with sensor exposure time.

The data output of the MT9M011 is synchronized with the PIXCLK output. When LINE_VALID is HIGH, one 10-bit pixel datum is output every PIXCLK period. The PIX-CLK signal is nominally the inverted of the master clock, allowing PIXCLK to be used as a clock to latch the data. It is continuously enabled, even during the blanking period.

The figure 3 shows the timing of pixel data.

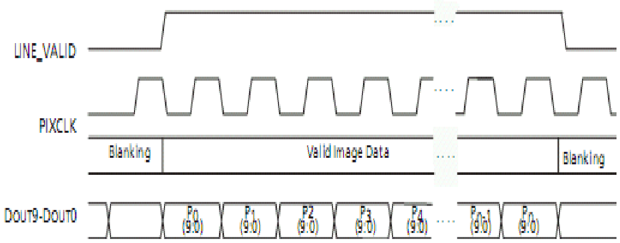


Fig 3: Timing example of pixel data.

The figure 4 gives the block diagram of CMOS configuration which is used for capturing the image by resolution of 640x480.

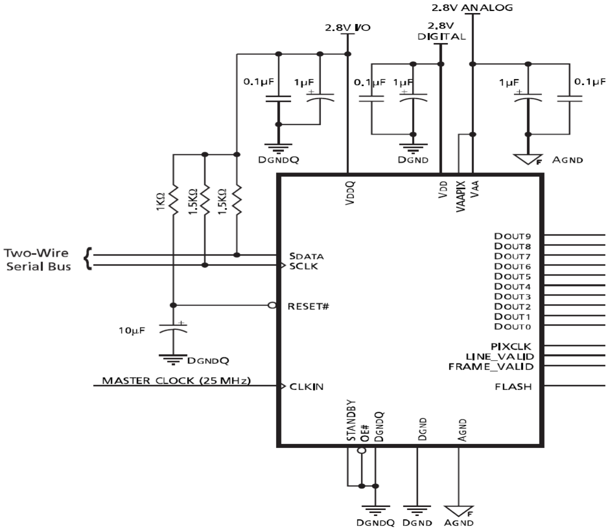


Fig 4: CMOS sensor configuration block diagram.

The sensor configuration data is sent on the sda. First we send the register value and then the data which is to be written into it register. The output clock of the I2C is given as input to the CMOS sensor which is serial clock (scl). First we set the row width by sending the register value as 0x03H the row width as 0x01E0H (480). Next we set the column width by sending the register value as 0x04H the column width as 0x0280H (640). Next we set horizontal blanking B by sending the register value as 0x05H the horizontal blanking B as 0x002DH (45). Next we set vertical blanking B by sending the register value as 0x06H the vertical blanking B as 0x0014H (20). Next we set horizontal blanking A by sending the register value as 0x07H the horizontal blanking A as 0x6400H (25,600). Next we set vertical blanking A by sending the register value as 0x08H the vertical blanking A as 0x2BC0H (11,200).

5. IMAGE DATAFLOW CONTROLLER AND BLACK & WHITE CONVERSION

The CMOS image sensor captures video continuously. It is necessary to control the flow of data to the memory where it is stored and hence we control the dataflow by giving start and end signals.

Data is a 10 bit vector. For an individual pixel initially the R component of the pixel then the G component of the pixel and finally the B component of the image are send), DVAL (this is to indicate the validity of the data), XCNT, YCNT are 11 bit vectors indicates the data corresponds to the pixel of row number equal to XCNT and column number equal to YCNT, finally the frame count which is a 32 bit vector and it counts the number of F_VAL.

A digital camera uses an electronic photo sensor, usually a CMOS or CCD device, to convert photons, or light energy into electrical signals. CMOS or CCD devices detect minimum light wavelengths and cannot functionally separate and code color information into electronic signals. To capture color information, a mosaic of tiny color filters known as a color filter array (CFA) or a color filter mosaic (CFM) is placed over the pixel sensors of the photo sensor. A widely used CFA is a Bayer filter (see figure 4.5). The Bayer filter pattern is 50% green, 25% red and 25% blue. This arrangement is also called GRGB, meaning that the G channel receives twice the information then the R and B channels each. This is so because 50% and more of the luminance or total light intensity pass through the green filter, which determines the brightness levels for the image.

Black & White conversion block is used to arrange the raw data coming from the image capture. CMOS sensor which send the data in the Bayer's pattern by using the x-coordinate (xcont) and y-coordinate (ycont) data stored in the small buffer because R,G,B are in random by storing the data we extracting the red (R), green (R), blue (B) components of the same pixels by using the Bayer's pattern. We convert the R, G, B image into the black and white image by using the below equation.

$$\mathbf{B.W = 0.25*R+0.5*G+0.25*B} \quad \mathbf{(1)}$$

6. SDRAM CONTROLLER

The SDRAM stores the image frames for comparison. It can store two frames at a time. Hence it is partitioned into two parts which can be read and written simultaneously. The block diagram of the SDRAM is shown in fig. 5. The image frames are written into partitions through port1 and port2 of write side and read into partitions through port1 and port2 of read side. The port1 is used to read/write first frame while port2 is used to read/write second frame.

When input is first frame it is taken on port1 by enabling WR1 and the input data is provided on WR1_DATA. When input is second frame it is taken on port2 by enabling WR2 and the input data is provided on WR2_DATA. If first frame data is to be read, then RD1 is enabled and the data is read on RD1_DATA. If second frame data is to be read, then RD2 is enabled and the data is read on RD2_DATA. WR1_ADDR, WR2_ADDR provide the address to which the data is to be written. RD1_ADDR, RD2_ADDR provide the address from which the data is to be read. The partition information is given by WR1_MAX_ADDR, WR2_MAX_ADDR. It gives the upper bound on the address to which data can be written. Hence the first frame is written within the upper bound of WR1_MAX_ADDR. It is similar in case of second frame.

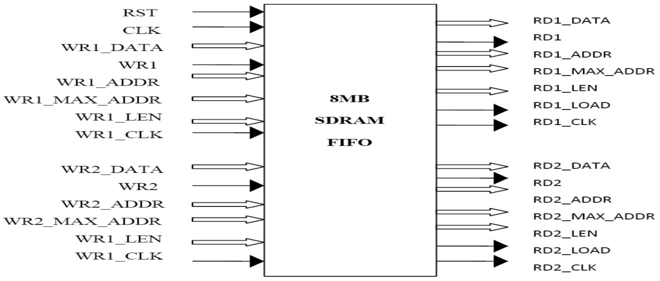


Fig 5: Block diagram of SDRAM

7. INTRUDER DETECTION

The figure 6 shows the block diagram of intrusion which is used for comparing the present frame and the past frame and sends the number of pixels differs in frame 2 from frame1. Here we set three types of thresholds one is for changing of small number of pixels, other is for changing of medium numbers of pixels, and last one is changing of large numbers of pixels.

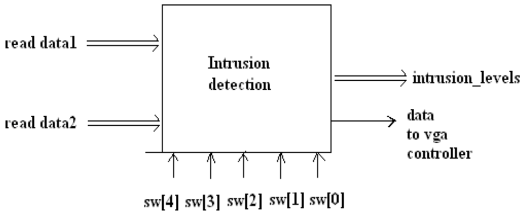


Fig 6: Block diagram of Intrusion Detection

The inputs of intrusion detection block are read data1 and read data2 which are present and past images data stored in SDRAM. Outputs of intrusion detection are intrusion levels which are error numbers of pixels when comparing two image pixels and these are displayed by using 7-segment display in FPGA board. Data to VGA controller will of three modes one is image1 display and second one is image2 display and third one is difference image. This mode setting will be done by using switches 4 and 3 in the FPGA board.

Sw[4:3] = 11 or 00 = image1 display (2)

Sw[4:3] = 01 = image2 display (3)

Sw[4:3] =10 = difference image. (4)

After comparison of 2 images the intrusion levels will be displayed based on the threshold values which are given by using switches. If we set switches as shown below on the board based on that values threshold level setting will be done.

Sw[2:0] = 001 set low threshold level (5)

Sw[2:0] = 010 set medium threshold level (6)

Sw[2:0] = 100 set high threshold level (7)

If there is intruder then the intrusion levels will change continuously otherwise it display constant value in 7- segment display it says that there is no intruder or no change in surroundings.

8. VGA CONTROLLER

The VGA monitor is controlled by five signals: red, green, blue, horizontal synchronization, and vertical synchronization. The three color signals, collectively referred to as the RGB signal, control the color of a pixel at a given location on the screen. They are analog signals with voltages ranging from 0 to 0.7V Different color intensities are obtained by varying the voltage. The circuit could treat these three color signals as digital signals, so that one can turn each one on or off.

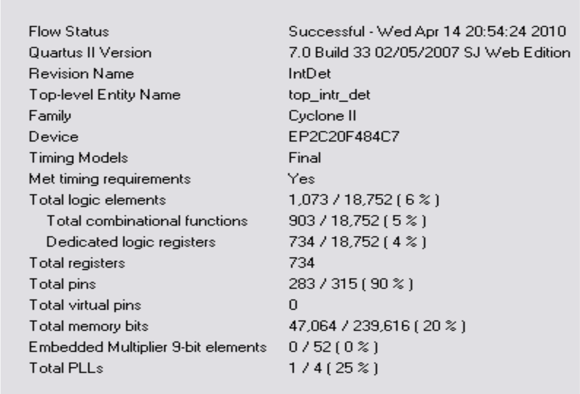
The horizontal and vertical synchronization signals are used to control the timing of the scan rate. Unlike the three analog RGB signals, these two sync signals are digital signals. In other words, they take on either logic 0 or logic 1 value. The horizontal synchronization signal determines the time it takes to scan a row, while the vertical synchronization signal determines the time it takes to scan the entire screen. Understanding how to control a VGA monitor simply boils down to understanding the timings for these two synchronization signals. By manipulating these two sync signals and the three RGB signals, images are formed on the monitor screen.

To obtain the 480x640 screen resolution, use a clock say with a 25.175-MHz frequency. A higher clock frequency is needed for a higher screen resolution. For the 25.175-MHz clock, the period is approximately 0.0397 µs per clock cycle. For region B of the horizontal synchronization signal, needed is 3.77 µs, which is approximately 95 clock cycles (3.77/0.0397). For region C, we need 1.79 µs, which is approximately 45 clock cycles. Similarly, we need 640 clock cycles (region D) for the 640 columns of pixels and 20 clock cycles for region E. The total number of clock cycles needed for each row scan is 800 clock cycles (95 +45 + 640 + 20). With a 25.175-MHz clock, region D requires exactly 640 cycles, generating the 640 columns per row. If we use a different clock speed, we will get a different screen resolution.

9. EXPERIMENTAL RESULTS AND DISCUSSIONS

The resultant system is implemented on an Altera CYCLONE II 2C35 FPGA which has software support for I/O interfaces and a control panel facility for accessing various components. Implementation of FPGA configuration will done on Quartus-II IDE. Verification of Verilog HDL to be performed using ModelSim.

The size of an image frame is 640 x 480 for the image processing unit to process. For the clock rate is 25MHz, image frame rate is around 30 fps.



Flow Status	Successful - Wed Apr 14 20:54:24 2010
Quartus II Version	7.0 Build 33 02/05/2007 SJ Web Edition
Revision Name	IntDet
Top-level Entity Name	top_intr_det
Family	Cyclone II
Device	EP2C20F484C7
Timing Models	Final
Met timing requirements	Yes
Total logic elements	1,073 / 18,752 (6 %)
Total combinational functions	903 / 18,752 (5 %)
Dedicated logic registers	734 / 18,752 (4 %)
Total registers	734
Total pins	283 / 315 (90 %)
Total virtual pins	0
Total memory bits	47,064 / 239,616 (20 %)
Embedded Multiplier 9-bit elements	0 / 52 (0 %)
Total PLLs	1 / 4 (25 %)

Fig 7: Synthesis report of Intruder detection system

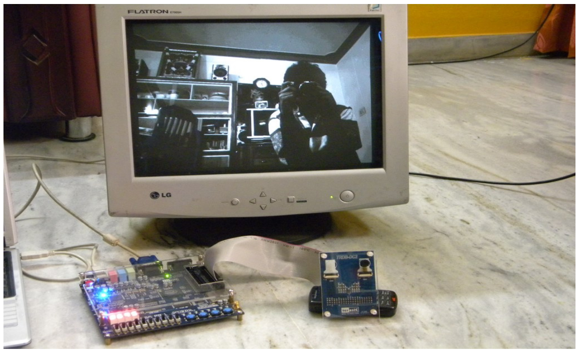


Fig 8: Appearance of the implemented system

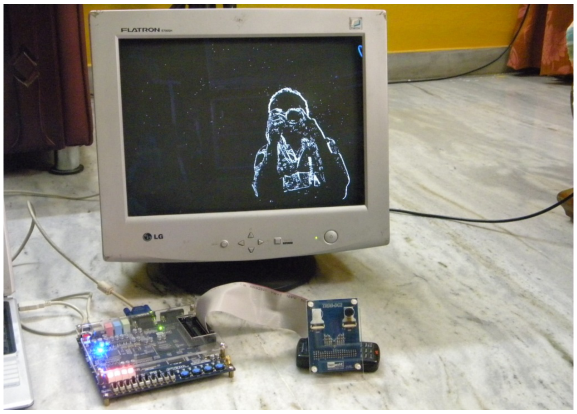


Fig 9: Displaying the moving object

10. CONCLUSION AND FUTURE SCOPE

It can be concluded that this project provides a better security system compared to the existing systems as the presence of minor disturbances can be detected. Also in this system visual monitoring can be done. This makes it a better system as the intruder can be identified too.

It can further be improvised to automatically send a message to the nearest police station and the owners of the house by which the occurrence of offence can be prevented. This can be done by interfacing GSM module to the current system which will automatically send a message to the owners informing them about intrusion. It can also improvise to store the image when the motion is detected in the room.

REFERENCES

[1] K. Robert, "Video-based traffic monitoring at day and night vehicle features detection tracking," Proc. of IEEE Conf. on Intelligent Transportation Systems, October 2009, pp.285-290.

[2] Chuan-Xu Wang and Zuo-Yong Li, "A new face tracking algorithm based on local binary pattern and skin color information," IEEE Transactions on International Symposium on Computer Science and Computational Technology, vol. 2, December 2008, pp.657-660.

[3] D. J. Dailey, F. W. Cathey, and S. Pumrin, "An algorithm to estimate mean traffic speed using uncalibrated cameras," IEEE Transactions on Intelligent Transportation Systems, June 2000, pp.98-107.

[4] Heon Soo Shin, Sung Min Kim, Joo Woong Kim, Ki Hwan Eom, "Novel object tracking method using the block-based motion estimation," IEEE Transactions on SICE Annual Conference, September 2007, pp.2535-2539.

[5] M. Z. Islam, Chi-Min Oh, Chil-Woo Lee, "Real time moving object tracking by particle filter," IEEE Transactions on International Symposium on Computer Science and its Applications, October 2008, pp. 347 -352.

[6] An Zhao, "Robust histogram-based object tracking in image sequences," IEEE Transactions on Digital Image Computing Techniques and Applications, December 2007, pp.45-52.

[7] Wu-Chih Hu, "Adaptive template block-based block matching for object tracking," IEEE Transactions on Intelligent Systems Design and Applications, vol. 1, November 2008, pp.61-64.

[8] Wei You, Hao Jiang, Ze-Nian Li, "Real-time multiple object tracking in smart environments," Proc. of IEEE Conf. on Robotics and Biomimetics, February 2009, pp. 818 -823.

[9] FPGA Implementation of a Real-Time Image Tracking System SICE Annual Conference 2010 August 18-21, 2010, The Grand Hotel, Taipei, Taiwan.

[10]CMOS image sensors: State-of-the-art, Albert J.P. Theuwissen, Delft University of Technology, Mekelweg, 4, 2628 CD Delft, The Netherlands.

[11] http://www.altera.com

[12] http://www.fpga-faq.com

[13] http://en.wikipedia.org/wiki/Bayer_filter

[14] http://en.wikipedia.org/wiki/Image_sensor