FPGA Based Camera Control System

Omkar Chitturi[1] A.Rajasekaran[2]
[1]M.Tech Scholar [2]Assistant Professor
[1,2]Department of Electronics & Communication Engineering
[1,2]SCSVMV University, Kanchipuram

Abstract— Video surveillance is being widely utilized in the world for various applications like crowd control, visitor monitoring, prevent theft and visual evidence for investigation. To increase the overall safety and security high-speed real-time monitoring systems are essential. FPGAs are suitable for these tasks as they have massive parallel processing capabilities. This paper presents the design and implementation of a video surveillance system which uses Zedboard and OV7670 camera. Zedboard contains Zynq-7020 All Programmable SoC which is a combination of ARM Cortex-A9 Processing System (PS) and Atrix-7series Programmable Logic (PL). The camera is mounted on two servo motors for pan and tilt movement to represent a surveillance camera. Motor control is done in the Processing System and the video footage captured by OV7670 camera is processed in the Programmable Logic and displayed on a VGA monitor. To implement this system, Xilinx Vivado and SDK 2015.2 are used.

Key words: Zynq AP SoC, Zedboard, Vivado, SDK, OV7670, servo motor

I. Introduction

Most common data and video processing devices are microprocessors, microcontrollers, Digital signal processors, Field Programmable Gate Arrays. The choice of selection depends on the application, and other considerations include processing performance, low Non Recursive Engineering (NRE) cost, energy consumption. [1]

Among those options, FPGAs present an good choice with regard to performance and production level.[2] As the number of logic elements per device, and operating clock frequencies are increased, along with the massive parallel processing strategy, the processing power of FPGA's came close to ASIC. Also the FPGA gives the flexibility of reconfigurable hardware, algorithm efficiency can also be increased.[3]

In this paper, the video footage is captured by a OV7670 camera module which is processed in the PL section and the PWM signals are generated using the PS section of APSoC.

A. Zynq SoC :

The Zynq-7000 All Programmable SoC is a integrated product of Dual Core ARM Cortex-A9 processor called processing system and Atrix-7 based programmable logic. The functional blocks of the processing system are application processor unit(APU) , memory interfaces , I/O peripherals , Interconnect to communication between the programmable logic and processing system

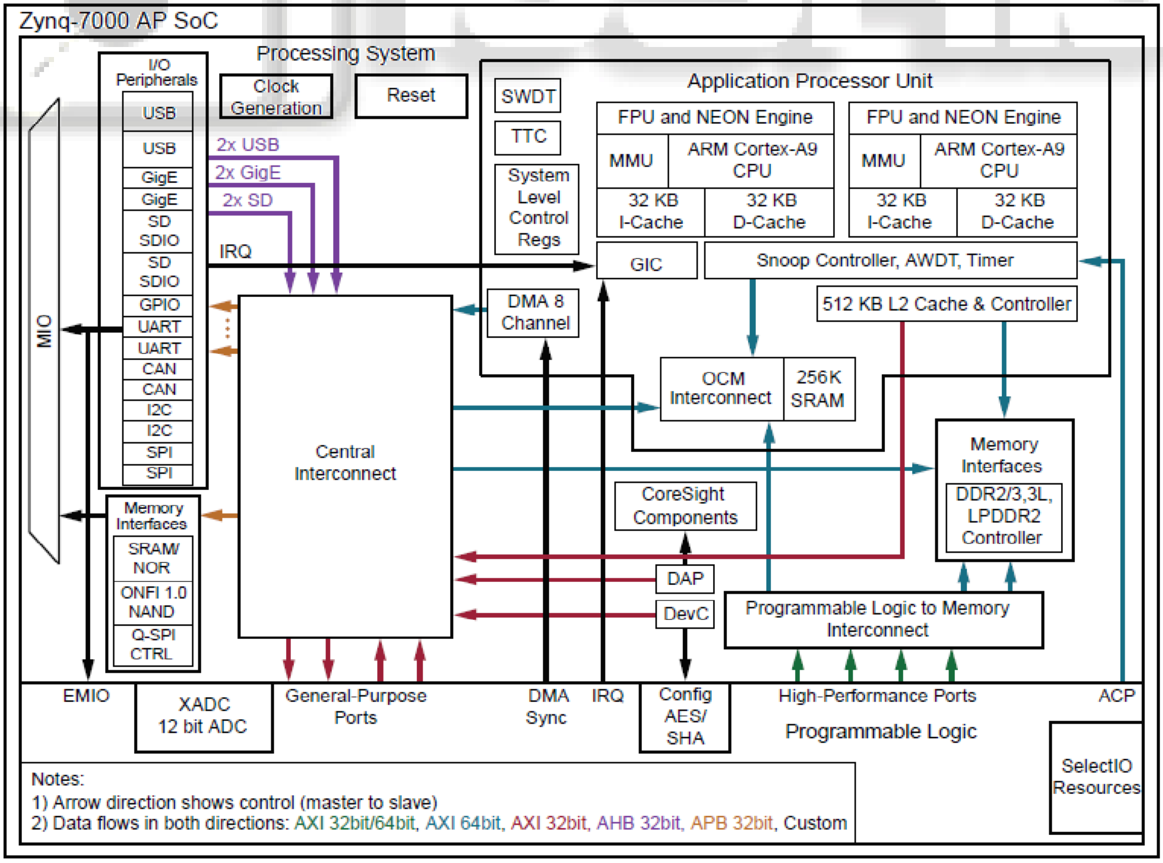

Figure 1: Zynq AP SoC Architecture [4]

With the Dual core ARM Cortex A9, processing can be done in different types like running application on a single processor, symmetrical multiprocessing, asymmetrical multiprocessing. There is Neon floating point engine for performing the floating point operations. Various timers like private timers, watchdog timer, triple timer/counter. Along with Accelerator Coherency port from the PL to PS making high speed data transfers and DMA controllers are also present. Different memories like DDR3, SPI Flash are available.

The PS I/O peripherals are up to 54 pins and called as Multiplexed I/O pins. To extend the functionality of the peripherals from PL to used by the PS, a feature called Extended Multiplexed I/O interface (EMIO) is available. Gigabit Ethernet controller, USB controller as Host, Device or OTG modes, SD/SDIO controllers enabling to boot the board. Others controllers include SPI, CAN, UART, I2C are also present.

On the side of programmable logic, there are Configurable logic blocks containing 6-input LUT, memory capabilities , 36 KB block RAM , DSP slices with 25 × 18 bit two's complement multiplier and a 48-bit accumulator, configurable I/O’s. Various PS-PL interfaces are present like the AXI_ACP, AXI_HP, AXI_GP. [4]

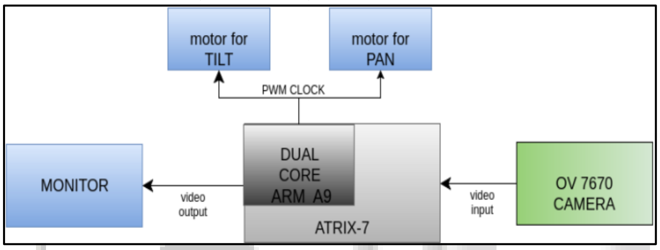

Fig. 2: System Setup

B. Zedboard:

Zedboard contains the Zynq-7020 AP SoC with 512MB of DDR3 RAM and 256Mb of QSPI Flash. Various interfaces like 10/100/1G Ethernet, USB OTG 2.0, SD card, five Pmod headers, seven push buttons eight dip/slide switches and nine user leds along with HDMI and VGA connectors are present for displaying the output.[5]

C. OV7670 :

OV7670 camera is CMOS sensor producing wide range of formats through Serial Camera Control Bus (SCCB). These formats include YUV/YCrCB 4:2:2, RGB565/555, Raw RGB data. Image control features like colour saturation, hue, gamma, sharpness are present. Some features like noise reduction and defect correction are present. The sensor has an image array of 656 x 488 pixels of which 640 x 480 are active giving about 0.3MP images. An analog signal processor along with a Digital Signal Processor is present in the camera module. Analog signal processor performs the functions like Automatic Gain Control and Automatic White Balance while the Digital signal processor performs Edge enhancement, Colour space converter, Hue and saturation control, White/Black pixel correction, Lens shading correction and transfer of 10-bit data to 8-bit.[6]

D. SYSTEM SETUP

This is done in 3 stages. First would be capturing the footage from the OV7670 camera module. Second stage would be obtaining Pan and Tilt movement of the motors. In the third stage, both the above steps would be combined.

1) Stage 1:

A design was made which takes input from the OV7670 camera and displays on the VGA monitor. Through the serial clock and serial data lines, the control signal are sent to the camera while the lines D0-D7 are used for the data transfer. Through the XCLK pin, clock is sent at which the camera operates and through the PCLK the pixel clock output is taken. The VSYNC and HREF are sync pulses for VGA.

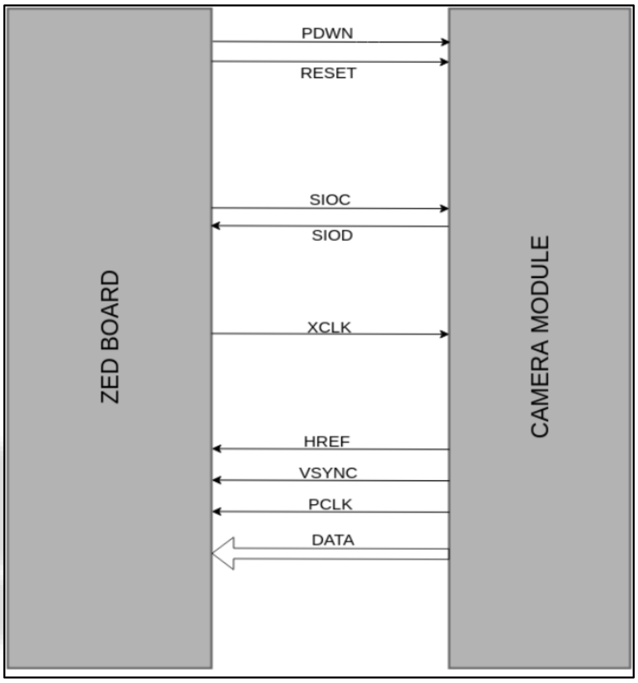

Fig. 3: Connections

A VGA monitor is connected to view the captured footage from the OV7670 camera module. After the PL is programmed, the camera starts capturing and can be viewed on the monitor.

2) Stage 2:

Two servo motors (SG-90) were used for pan and tilt movement. For this, PWM input of 50 Hz is necessary to rotate. A 1ms on time makes the motor rotate left, 1.5 ms to centre and 2 ms to right.

For this, PWM generation, PS of the SoC is being used which has private timer operating at 200MHz. On time for certain period and 20ms on time gives the 50 Hz signal for the motor to operate. During the on time, the output pin should be high and during the off time, the output pin to be made low.

In PS section, the private timer is clocked at half of the CPU frequency (CPU_3x2x).

The CPU clock frequency ratio is 6:2:1 and at CPU_3x_2x, it would be 400MHz and the private timer works at half of it being 200MHz.Using that 200MHz clock, PWM signal is generated from the Pmod JE1 and JE2.

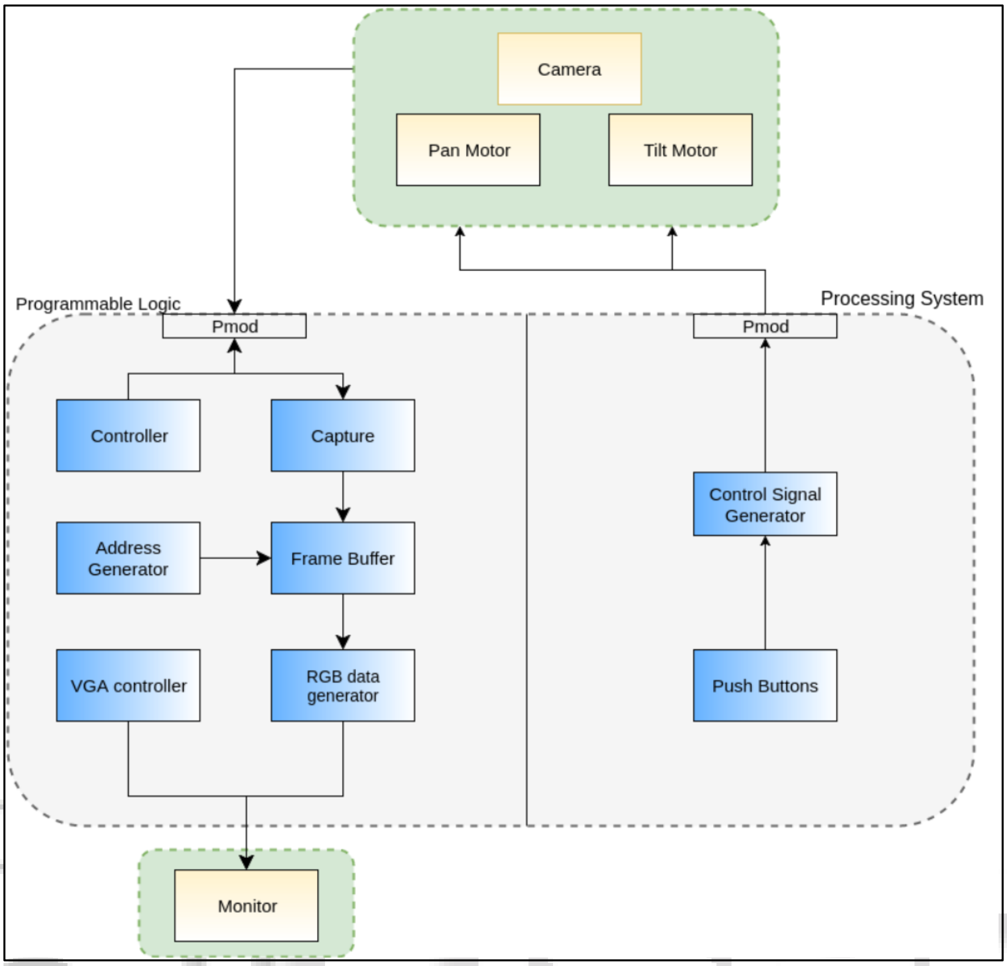

Fig. 4: Block Diagram

The PS is required to use for generation of PWM signal so, a new project is created in Vivado and Zynq IP is added to the block design. And bit stream is generated which is then exported to SDK where the PS is programmed.

1) Step 1: Initialize timer
2) Step 2: Initialize GPIO
3) Step 3: Set pins as output or input
4) Step 4: Enable those pins
5) Step 5: Read the value of that pins
6) Step 6: Generate PWM signal
7) Step 7: If other pin is high, stop the rotation

The output from the Pmod is 3.3V but the servo motors operate with at 5V so, an amplifier is required. For this purpose, an 74HC04 IC is selected[9]. This IC is selected as the input of 3.2V or higher is taken as HIGH so, this would be a suitable option for amplifier. Input is given to pins 1 and 13. as this is an inverter, the output from pins 2 and 12 are given to 3 and 11. Finally the outputs from 4 and 10 are given to the PWM input of motor.

3) Stage 3:

Now both of those have to be combined, to accomplish that an IP is created from the stage 1 and added to the block design of stage 2. The constraints are written and bit stream is generated.

This is exported and the code above is used to generate the .elf file. The PL is programmed first and then the PS is programmed. Thus, the camera footage is displayed on the monitor and the motors below would be performing pan and tilt movement.

II. Experimental Results

After the pins of camera are connected to pmods and VGA monitor to Zedboard, the PL is programmed and the footage being captured is displayed on the monitor.

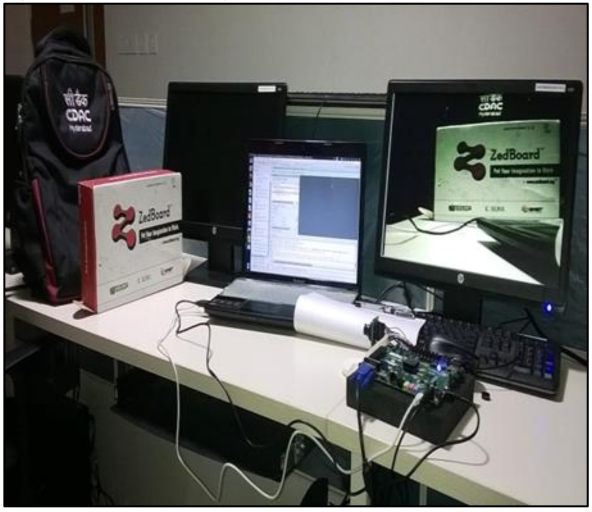

Fig. 5: Output of stage 1

For interfacing the motor, the PWM output is taken from the Pmod JE1 and JE2. Pan and Tilt movement of the motors can be started by pressing the push button on the board

When the next push button is pressed, the motors stop moving. Thus pan & tilt movement is accomplished. The camera is mounted on the pan and tilt mounts. And when the APSoC is programmed, the footage is displayed on the monitor when the push button is pressed, the pan and tilt movement starts.

Care should be taken while rotating the motors if the position is extreme left and if the signal is being provided to go to the same, the motor would damage.

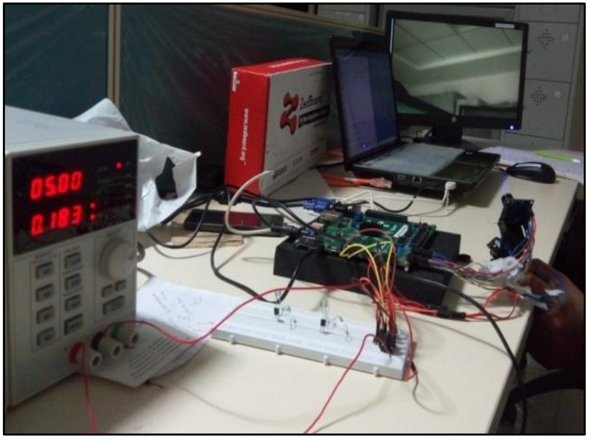

Fig. 6: Output of stage 3

III. Conclusion

Thus, an OV7670 camera module is used to capture video footage and display it on the monitor and pan and tilt movement is achieved using servo motors realizing a CCTV system. This is controlled by the push buttons on the board. As future enhancement, the controlling can be done using gestures and more advanced camera module can be used for capturing the footage. Also the captured footage an be transferred through Ethernet which can be viewed from remote locations.

References

[1] Ignacio Bravo, Javier Baliñas, Alfredo Gardel, José L. Lázaro, Felipe Espinosa and Jorge García "Efficient Smart CMOS Camera Based on FPGAs Oriented to Embedded Image Processing" Sensors2011,2282-2303; doi:10.3390/s11030 2282

[2] Mustafah, YM; Bigdeli, A; Azman, AW; Lovell, BC. Smart Cameras Enabling Automated Face Recognition in the Crowd for Intelligent Surveillance System. Proceedings of Recent Advances in Security Technology (RNSA) 2007, The 2007 RNSA Security Technology Conference, 2007; pp. 310–318.

[3] Bravo, I; Mazo, M; Lázaro, JL; Gardel, A; Jiménez, P; Pizarro, D. Intelligent Architecture Based on FPGAs Designed to Detect Moving Objects by Using PCA.Sensors2010,10, 9232–9251

[4] ug585-Zynq-7000-TRM.pdf

[5] ZedBoard_HW_UG_v2_2.pdf

[6] http://www.voti.nl/docs/OV7670.pdf

[7] http://embeddedprogrammer.blogspot.in/2012/07/hacking-ov7670-camera-module-sccb-cheat.html

[8] Arbab Waheed Ahmad, Heekwon Yang, Gul Shahzad, Chankil Lee "Design and Implementation of FPGA based Automatic Lecture Recording System", 17th International Conference on Advanced Communication Technology (ICACT) (2015)

[9] http://forum.arduino.cc/index.php?topic=19661.0

[10] http://hamsterworks.co.nz/mediawiki/index.php/Zedboard_OV7670

[11] Salvatore Vitabile,Alessandra De Paola and Filippo Sorbello "Bright Pupil Detection in an Embedded, Real-time Drowsiness Monitoring System", 24th IEEE International Conference on Advanced Information Networking and Applications, pp. 661-668 (2010)

[12] Yu Shi, Ronnie Taib and Serge Lichman "GestureCam: A Smart Camera for Gesture Recognition and Gesture-Controlled Web Navigation", International Conference on Control, Automation, Robotics and Vision (2006).