EasyChair Preprint

№ 5033

ROS on ARM Processor Embedded with FPGA for Improvement of Robotic Computing

Tapas Kumar Maiti

February 25, 2021

ROS on ARM Processor Embedded with FPGA for Improvement of Robotic Computing

Tapas Kumar Maiti
DA-IICT, Reliance Cross Rd, Gandhinagar, Gujarat 382007, India
Email: tapas_kumar@daiict.ac.in

***Abstract*—This paper presents the development of an embedded computing system with the implementation of communication between ROS (Robot Operating System) -on-ARM processor and FPGAs (Field Programmable Gate Arrays) for improvement of robotic computing. A DE0-Nano Terasic Altera Cyclone IV development board is used in FPGA side which is programmed with Verilog code to define GPIO (General Purpose Input/Output). On the other hand, GPIO parallel communication python library is developed for RPi3 (Raspberry Pi3) board integrated with ARM Cortex-A53 processor. Implementation of parallel communication between FPGA and RPi3 is verified. As ROS is widely used for different types of robots development, we considered ROS on RPi3 for general purpose use in robotic computing system. As a case study, we performed humanoid robot fall simulation on ROS, controlled via 3-axis accelerometer embedded with DE0-Nano board.**

***Keywords*—ROS, FPGA, ARM Processor, Humanoid Robot, Sensor, Robotic Computing**

I. Introduction

Over the past few years, numerous types of sensors and their applications have been reported to realize autonomous and intelligent humanoid robots [1], [2]. These intelligent robots required high-performance processor for processing and calculation of huge amount of sensor data. These robots are under the limitations of power supply due to battery function, computation, and weight because of their size. Heavy-weight humanoid robots such as Korean HUBO, HONDA ASIMO, MIT Cog and AIST HRP are equipped with high-power battery and high-performance computer which is large in size [1]. On the other hand, small-size and light-weight humanoid robots such as SoftBank NAO [1] and KONDO KHR-3HV [3], which are commercialized for entertainment, education, and human efficiency analysis purposes, are also equipped with similar computing platform. However, these light-weight humanoids required small-size, low-power, and high-performance computing resources to solve the intelligent functions.

This paper focused on the FPGAs (Field Programmable Gate Arrays) to speed-up the light-weight-robot application specific computing function with low-power consumptions at circuit level. The direct implementation of robot function in FPGA is quite challenging, and time consuming compared to software level implementation due to complexity FPGA coding of robot mathematical formulations [4]. We proposed HW/SW co-design method to solve these issues. We used FPGA for the implementation of complex computing part which takes longer time in computation, and remaining part is implemented in ROS (Robot Operating System) robotic software. We installed ROS on ARM processor which communicates with FPGA through GPIO (General Purpose Input/Output) to solve robot computing functions in an efficient way.

In this work, we developed an embedded system for high-performance light-weight robotic computing. We used FPGA and ARM processor development boards (see Fig. 1) to configure the computing system. Robot-computing functions are implemented in FPGA using Verilog programming code at register-transfer-level. Communication between FPGA and ARM processor is established through GPIO pins. We programmed a virtual light-weight humanoid-robot on ROS using Python language. We performed the humanoid-robot falling simulation using developed computing system, controlled with 3-axis accelerometer to verify the use of developed prototype system.

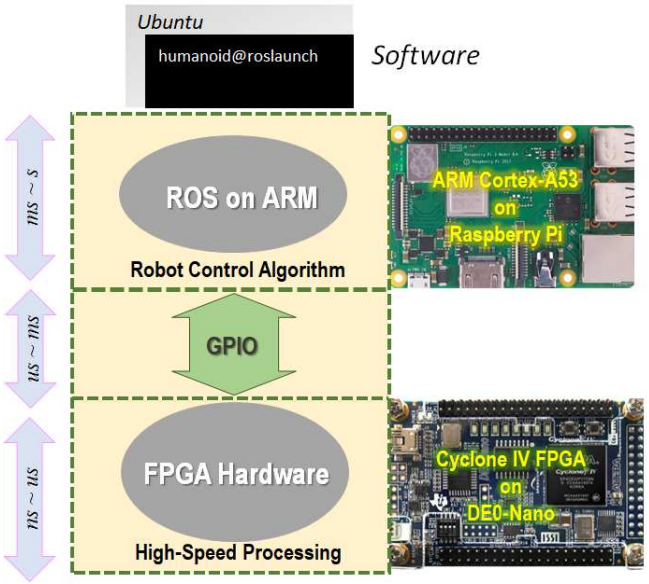

Fig. 1. Robotic computing system configuration. Implementation of FPGA connections with ROS installed on ARM processor is illustrated. Single task computing time in FPGA, GPIO, and ROS, is also depicted.

II. Develepoment of Robotic Computing Systen

This section describes the development phases of robotic computing system which is illustrated in Fig. 1. Figure 1 schematically shows an FPGA hardware, an ARM processor integrated in RPi3 (Raspberry Pi3) board, ROS software installed on ARM processor, and GPIO communication ports. It depicted the single task computing time required in FPGA, GPIO, and ROS. Several FPGA-based robot computing systems are reported [5], [6]. For examples, Yamashina *et al*., reported cReComp tool for the development of ROS-compliant FPGA component [4], Nitta *et al*., illustrated a programmable SoC-based ROS framework to develop autonomous driving systems [5]. However, development of robotic computing system with those outcome, is very difficult and slow due to the complexity in integration of communication interface between FPGA and ROS. There are also difficulty in ROS installation on ARM based SoC framework due to rapid increase of number robot-software packages [6]. In this section, we proposed a simple and efficient HW/SW co-development method for the high-speed robotic computing system, is described below.

A. *Implementation of Customized Circuits on FPGA*

A DE0-nano development education board [7] is used to implement FPGA part (see Fig. 1). This board is integrated with Cyclone IV EP4CE22F17C6N FPGA, on-board 50MHz clock oscillator, ADXL345 3-axis accelerometer, two 40-pin external GPIO headers, etc. For detail descriptions, we refer to Intel FPGA design solution network [8]. We configured accelerometer with customized FPGA circuits at RTL level using Verilog code. The Verilog code for accelerometer is supplied with the FPGA board. However, we modified the code with the 2-bit input register which is called dimension. The dimension 0, 1 and 2 are corresponding to x, y, and z axis respectively. We measured the acceleration in three direction x, y, and z in physical space. The acceleration components are used for ROS-based light-weight virtual-humanoid-robot falling simulation.

B. *ROS on ARM Processor*

A Raspberry Pi 3 (RPi3) microcomputer board [9] with a 1.2 GHz 64-bit quad core ARM Cortex-A53 processor, consumes power 1.4W (at current consumption 260 mA) which has overall dimensions of 85mm×56mm×17mm is used (see Fig. 1). Note that our focus is on the development of light-weight humanoid robot which requires small dimension of microcontroller to fit it with robot mechanical structure, thus the size of RPi3 is suitable for this application. RPi3 is also compatible with Ubuntu operating system which provides a flexible framework to ROS. Ubuntu MATE 16.04 is chosen to install the upgraded version of ROS Kinetic Kame for software level robotic computing. RPi3 is equipped with GPIO, is configured with our developed driver based on RPI.GPIO Python library. The pseudo-code of driver that is used to configure GPIO in RPi3 is given below,

```
#! user/bin/env python
import rospy
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BCM)
gpio_pins = [18, 19, 20, 21, 22, 23, 24, 25]
gpio_clk = 16
gpio_clr = 17
```

We considered 10 GPIO pins for parallel communication purpose for high-speed computing. We described the details of pin configuration in section II.C. We included GPIO.cleanup() syntax before the computation exits. This helps to clean the buffer data from ROS hardware which increases the efficiency of hardware during re-execution of ROS program.

C. *Connecting FPGA with ROS*

Most important part of our development is the implementation of interface connection between DE0-nao and RPi3 development boards. We started the development of computing system to establish serial communication between DE0-nano and RPi3. We successfully implemented serial port communication of individual board with baud-rate 115200. However, after few seconds of ROS with FPGA via RPI3, both board stop communicating to each-other. As an alternative, we implemented parallel communication between both boards using GPIO pins as presented in Fig. 2.

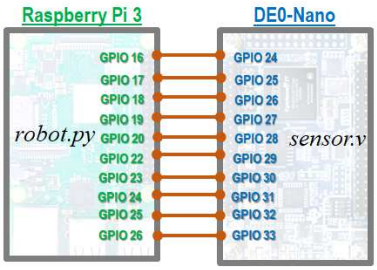

Fig. 2. Implemention of parallel GPIO communication betewwn Raspiberry Pi 3 (RPi3) and DE0-Nano Cyclone IV FPGA. 10 GPIO pins are chosen where one GPIO pin is used for clock, one GPIO pin controlled read/write data, and remaining 8 GPIO pins are used for data transmission.

Here RPi3 functioned as a master device and FPGA as slave device. The communication between two devices is control by master. Note that we considered 10 GPIO pins where one pin *gpio_clk* is used for clock, one pin *gpio_clr* controlled read/write data, and remaining 8 pins *gpio_pins* are used for data transmission. This is implemented using the GPIO python library in RPi3 driver as shown in pseudo code of Section II.B.

III. Light-weight Humanoid Robot Falling Simulation

Figure 3 depicted the developed robot computing system which is used for light-weight humanoid robot simulation. The simulation is running on ROS installed on RPi3 connected with DE0-Nano Cyclone IV FPGA development board via GPIO port. Table 1 shows the list of components which are used for this work. Electronic components such as processor, FPGA, and sensors are supplied by several vendors, however software components such as Ubuntu operating system, ROS, and robot model are open-sources [6]–[10]. Here, ROS is an open source robotic software, released by OSRF (Open Source Robotics Foundation) [6]. It provides communication layers for robot system, runs mainly on Ubuntu OS. Kuroiwasi *et al*., has developed a light-weight humanoid robot model, known as premaidAI model accessible in GitHub [10], is used for robot simulation. The model is not directly usable in ROS as it is a CAD file. Therefore, following method is considered to make the model ROS compatible.

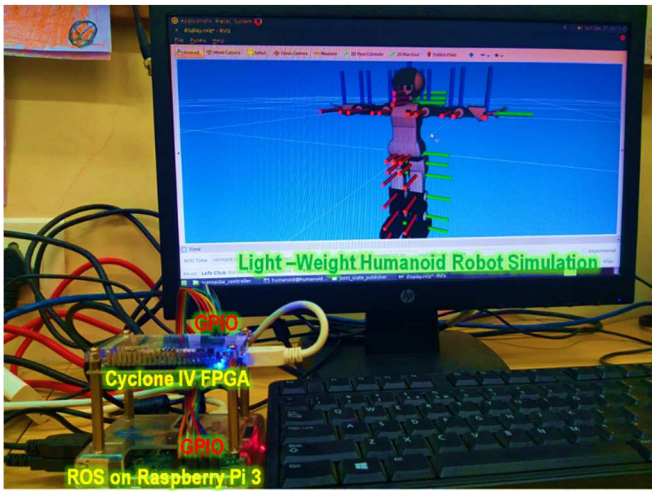

Fig. 3. Developed computing system for light-weight humanoid robot simulation using ROS. The system consists of DE0-Nano Cyclone IV FPGA, Raspberry Pi 3 development board, and virtual humanoid robot simulation environment which is running on ROS. A 3-axis digitial accelerometer is embedded in DE0-Nano board.

TABLE I: USED LIST OF COMPONENTS TO DEVELOP COMPUTING SYSTEM

Name	Specification
Processor	1.2 GHz 64-bit Quad-Core ARM Cortex-A53 processor [9]
FPGA	Cyclone IV EP4CE22F17C6N FPGA [7]
Sensor	Analog Devices ADXL345, 3-axis accelerometer [8]
Operating System	(i) Ubuntu MATE 16.04 LTS (ii) ROS Kinetic [6]
Robot Model	PremaidAI model [10]

Direct import of any format of CAD file to ROS environment does not allow accurate import of robot shape and texture information in ROS. Therefore CAD file should be transformed to COLLADA (COLLAborative Design Activity) format prior to import in ROS. In this work robot model considered in COLLADA format where the joint part is defined as joint, and the rigid body is defined as link. Then we created a launch file to display the robot structure on robotics graphics which is RVIZ [6]. Here, RVIZ is a robot visualization software in 3D for ROS. Next we created *ros_control* package to establish the communication between Gazebo [6] and ROS for robot dynamic simulation. Here, Gazebo is a 3D robotic simulator integrated with ODE physics engine and OpenGL. We implemented hardware communication with ROS using python GPIO which results a flexible controller established parallel communication between real-world sensors and simulation robots. We controlled servo-motors of a humanoid robot using the developed ROS computing environment. In other words, position, speed, and force of the robot are controlled for robot motion analysis.

Based on the principles of position control in ROS, we proposed a humanoid robot simulation method on human fall analysis inspired by humanoid robot simulation. We fed the digital output of ADXL345 3-axis accelerometer into ROS to perform humanoid robot simulation. We observed different joints and links configurations of the robot as depicted in Figs. 4. Figure 4(a) depicts stable standing condition, and Fig. 4(b) represents falling condition. The main focus is given on the principles of fall simulation, which is based on the change of robot body configuration due to random control of robot joints angles influenced by the change in acceleration of the 3-axis accelerometer.

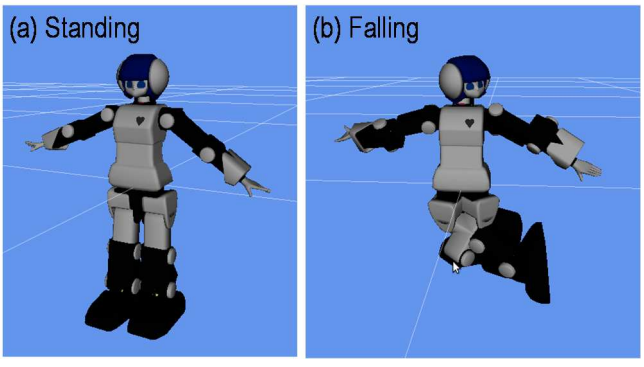

Fig. 4. Sanding and falling conditions of humanoid robot shown in (a) and (b) respectively. Joint and links configuration of the humanoid robot in falling condition is different from stable condition due to influence of output voltage of the accelerometer.

Figure 5 illustrates changes in acceleration during (i) standing and (ii) falling measured from accelerometer (see the marked in blue boxes in Fig. 5). It shows the critical differences in magnitude of accelerometer resonses between (i) and (ii). We created a spike on accelerometer responses as it is similar to a large shock when a body impact on the ground, corresponding robot body configuration shown in Fig. 4. Note that Fig. 4(b) is corrsponfing to the acceleration spike in Fig. 5 marke with bulue box (ii). This illustration confirmed that our developed platform can be used to simultae the virtual robot fall with the influnces of accelerometer responses. Robot fall simulation will guide us to anlysze the serious consequences of human falling such as unconsiousness or inacitivity state for long time after falling. Note that one can use the acceleremeter for the detection of acceleration of a body which responses to 0 *g* (zero gravity) during free fall.

IV. PERFORMANCE EVALUATION

The performance evaluation of our developed computing system is described in this section. We conduction the evaluation in three different conditions which are (i) only FPGA for sensing (ii) Raspberry Pi 3 software only, and (iii) including all components such as FPGA, Raspberry Pi 3, and ROS. We found that a single task computing time in FPGA is 70 µs, RPi3 uses 400ms, and entire computing system uses 500ms. We also compared our work with the previously reported results [4]. We observed that our system is faster for the computing only with FPGA or RPi3. However, entire system computing is slower due to complexity in simulation of humanoid robot on ROS that uses longer simulation time in computing due to existence in several joints and links in humanoid robot model.

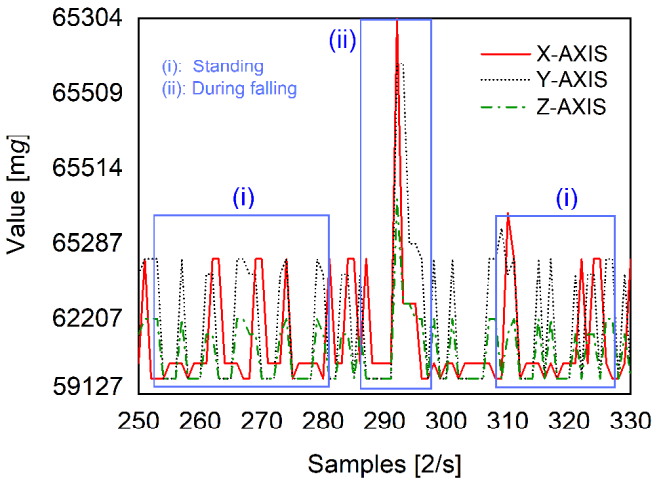

Fig. 5. Human generated accelerometer responses for different types motions such as (i) standing condition, and (ii) falling condtion.

TABLE II: MEASURED COMPUING TIME FOR SINGLE TASK

Work Done	Time		
	FPGA	***Raspberry Pi 3***	***FPGA + Raspberry Pi 3 + ROS***
This work	70µs	400ms	500ms
Yamashina *et al.*, [4]	--	835ms (ARM, Software only)	32ms (ARM + FPGA)

V. CONCLUSION

Ddevelopment of an embedded computing system with the implementation of communication between ROS-on-ARM and FPGA is presented. Entire system functions such as sensing with accelerometer embedded with FPGA board, Communication between ROS and FPGA via Raspberry Pi 3, and robot model implementation on ROS are described. We performed the robot simulation which is influenced by accelerometer responses to demonstrate the use of developed computing system. A comparative study on performance evaluation is illustrated.

REFERENCES

[1] A. Goswami, P. Vadakkepat (eds.), Humanoid Robotics: A Reference, Springer Nature, 2019, pp. 2483-2591.

[2] S. Wang, W. Chaovalitwongse, and R. Babuska, "Machine Learning Algorithms in Bipedal Robot Control," *IEEE Tran. on Systems, Man, and Cybernetics, Part C*, vol. 42, Iss. 5, pp. 728 - 743, Sept. 2012.

[3] T. K. Maiti, Y. Ochi, D. Navarro, M. Miura-Mattausch and H. J. Mattausch, "Walking Robot Movement on Non-smooth Surface Controlled by Pressure Sensor," *Adv. Mater. Lett.*, vol. 9, no.2, pp.123-127, May 2018.

[4] K. Yamashina, T. Ohkawa, K. Ootsu, and T. Yokota, "cReComp: Automated Design Tool for ROS-Compliant FPGA," *Component," IEEE 10th Int. Sym. on Embedded Multicore/Many-core Systems-on-Chip (MCSoC-16)*, pp.138-145, Sept. 2016, Lyon, France.

[5] Y. Nitta, S.Tamura, and H. Takase, "A Study on Introducing FPGA to ROS based Autonomous Driving System," *International Conference on Field-Programmable Technology (FPT)*, pp-424-427, Dec. 2018, Okinawa, Japan.

[6] Robot Operating System (ROS), Dec 2020, http://wiki.ros.org/ Distributions/

[7] DE0-Nano Development and Education Board, Dec 2020: https://www.intel.com/content/www/us/en/programmable/b/de0-nano-dev-board.html

[8] Intel FPGA design solution network, Dec, 2020: https://www.intel.com/content/www/us/en/programmable/solutions/partners/partner-profile/terasic-inc-.html

[9] Raspberry Pi official site, Dec 2020: https://www.raspberrypi.org/ products/ raspberry-pi-3-model-b-plus/

[10] PremaidAI Model, Aug 2019: https://github.com/kuroiwasi/ PremaidAI_Model