Dynamic Motion Sensing System

Technical Documentation Report for Electronic Engineering Project Work

Yashodhan Vishvesh Deshpande
Student of Electronic Engineering
Hamm-Lippstadt University of Applied Sciences
Lippstadt, Germany,
yashodhan-vishvesh.deshpande@stud.hshl.de

***Abstract*—The rise of digital technology and firmware development has lead to development of complex hardware and software systems. These systems enable us to create real time systems that have higher sophistication. Thus it allows us to develop , maintain and upgrade electronic systems like multimedia and automobile applications.**

This paper is documentation of a dynamic motion sensing system that was developed for the project work module in the Electronic Engineering course work. This document describes in brief about the motivation, problem space, methodology, system design and Xilinx Vivado- Vitis IDE implementations.

***Index Terms*—Vivado IDE, Vitis, Pmod Module, IP block library. OLED display, accelerometer .**

I. Motivation

The developments in the field of automobile industry and drone manufacturing has led to a demand of smart real time systems that perform dynamic motion sensing. These systems use cutting edge hardware and embedded software technologies in form of sensors and actuators. These sensors nodes and computational nodes form a system that is responsible for detecting motion and location of an object in 3D space.

The motivation behind developing a dynamic motion sensing system is to explore the operational aspects of a Pmod ACL accelorometer, Oled rgb display. The complex integration of backend hardware system in Vivado IDE using microblaze embedded processor and the frontend software application in Vitis IDE.

II. Problem Space

Many dynamic motion sensing systems exist in the current world of technology. After conducting a review of various such systems it was discovered that there are limited number of such FPGA based systems.

III. Methodology

In order to realize this, a dynamic motion sensing a real time system was proposed that would measure acceleration in 3D space and display it on the OLED display.

The proposed system would consist of both hardware and embedded software parts that would make the entire system. This would give a global view of the entire system from bottom up approach.

Identify applicable funding agency here. If none, delete this.

IV. System Design

This section defines proposed system specifications and an overview of the system.

A. Hardware Requirements

The hardware components used in this project are as follows

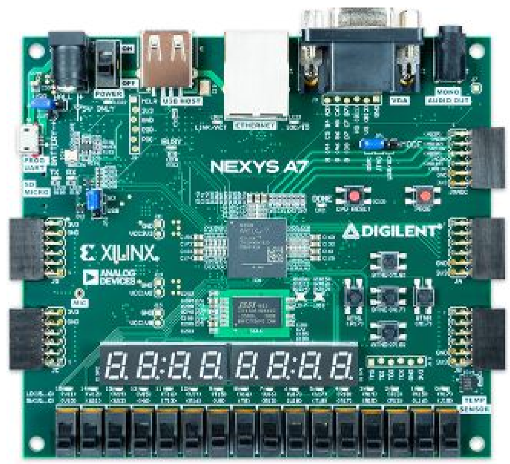

Fig. 1. Digilent Xilinx Nexys A7 100T [1]

1) Xilinx Nexys A7 100T Board: Figure 1 shows the Xilinx Nexys A7 100T board that is used for this project. This board is responsible for the required digital processing of the entire system. This particular board is chosen beacuse it has significant number of Pmod Ports which would be needed for sensor integration.

2) Pmod ACL 3 Axis Accelerometer: Pmod ACL 3 Axis Accelerometer shown in figure 2 is used in this system for measurement of acceleration and motion of an object in 3 dimensional space. This particular device is chosen because

Fig. 2. Pmod ACL 3 Axis Accelerometer [2]

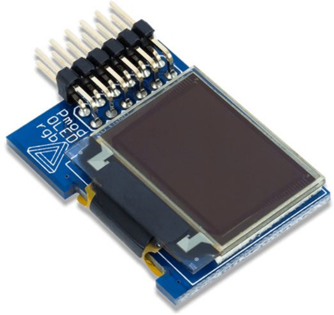

Fig. 3. Pmod OLED rgb display [4]

of it high resolution. It uses SPI communication protocol over Pmod interface. [2]

3) Pmod OLED rgb display : The Pmod OLED rgb display shown in figure 3 is chosen for displaying pourposes needed in this project. This device has been chosen as it has 16 bit color resolution. It uses SPI communication protocol over Pmod interface for communication purposes. [4]

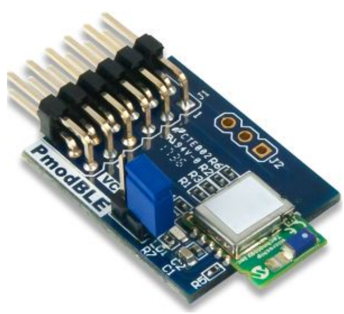

Fig. 4. Pmod BLE Bluetooth Module [3]

4) Pmod BLE Bluetooth Module: The Pmod BLE bluetooth module shown in figure 4 is used for data transfer process with other devices. It has UART communication protocol over pmod interface. [3]

5) USB A to micro USB cable: The USB A to micro USB cable shown in figure 5 is used for communication between the Windows PC applications to the Xilinx Nexys A7 board. The cable is used to connect the USB- UART Bridge communication interface. The USB A end of the cable is attached to a Windows PC while the micro USB end of the

Fig. 5. USB A to micro USB cable [5]

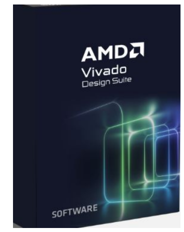

Fig. 6. Vivado Design Suite 2023.1 [7]

cable is attached to the UART Bridge PORT on the Nexys A7 board.

B. Software Requirements

The software components used in this project are as follows

1) ***Vivado Design Suite 2023.1*** *:* The Vivado Design Suite version 2023.1 shown in figure 6 is an Integrated Development Platform (IDE) used for hardware synthesis and hardware discription language (HDL) based designs. This IDE is used for the hardware synthesis of hardware block designs in this project. This IDE is choosen as it is best suited IDE for the hardware development when working with the Nexys A7 board as it has all required feature integrated. For example it has block design feature which allows to use IP block and repositories, it can run synthesis and generate bitstream which will be used to create a HDL wrapper for the project.

Fig. 7. Vitis Development Platform [6]

2) Vitis Unified Software Development Platform: The Vitis Unified development platform shown in figure 7 is an IDE that is used for software development in this project. The vitis IDE has a crtitical feature like compatability with integrating backend hardware wrapper file. This allows for developing the complete project including hardware synthesis integration.

3) C programming language: The C programming langugae as been choosen for the software development. This is because C programming language has features of embedded C and is closely related to low level languages like assembly language.

C. System Overview and Block Diagram

The block diagram shown in figure 8 give an overview of the entire system. It shows that the pmod ACL module is the input sensor node, the Nexys A7 100T board module is the central processing node while the pmod OLED diaplay and bluetooth module are the actuators in the system. The communication protocols used are Serial peripheral interface SPI.

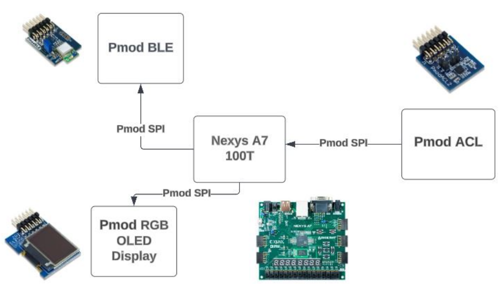

Fig. 8. System Block diagram

V. Hardware Synthesis and Implementation

This section focuses on the hardware implementation in the Vivado IDE.

A. Microblaze Processor

Fig. 9. Microblaze Processor

The microblaze processor as shown in Figure 9 is a soft processor developed by AMD Xilinx. It has 32 bit processor core and has RISC Harvard architecture. This processor is used in vivado Block design as the main microcontroller to perform all the necessary computations of the system. The processor has AXI peripheral interface and other feature like memory and DDR register.

B. System Clock

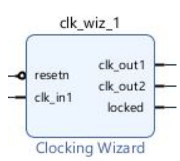

Fig. 10. System Clock

The Clocking Wizard LogiCORE ™ IP is used for providing clocking requirements for the system. The clock wizard has the ability to provide multiple clock frequency to various IP blocks. The mircoblaze IP processor, Mig 7 serires memory and the Pmod IP block modules have been provided a clock speed of 100 MHz.

C. AXI Uartlite communication interface

The AXI Uartlite interface shown in figure 11 is designed to provide asynchronous serial data transfer using the Advanced

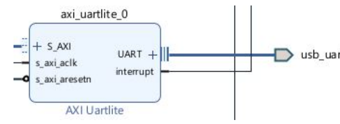

Fig. 11. AXI UARTLITE

eXtensible Interface (AXI) that the UART communication protocol. This IP block creates a communication channel between the Mircoblaze central processor and the Vivado IDE on Windows PC using the USB- UART Bridge communication interface on the NEXYS A7 board.

D. Pmod IP Blocks

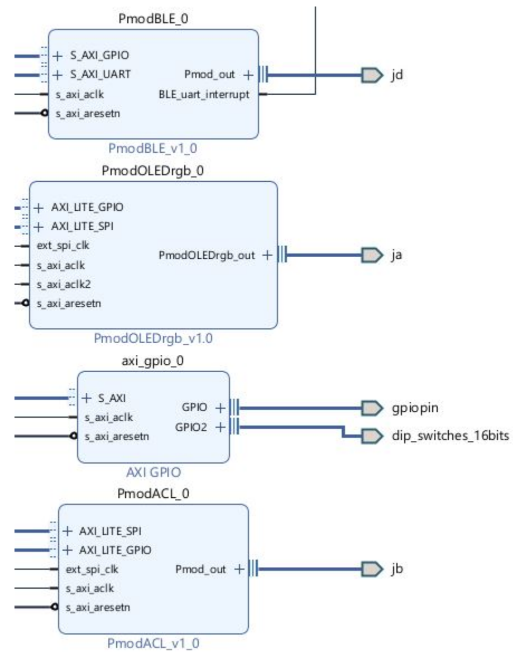

Fig. 12. Pmod IP block

The pmod IP blocks shown in figure 12 are IP blocks of Bluetooth module OLED display module and Pmod ACL accelerometer sensor. The Pmod IPs are the official IPs for pmod sensors provided by Digilent.

1) Pmod BLE Bluetooth IP: The Pmod bluetooth module is responsible for providing the bluetooth connectivity for the system. It has an interrupt pin that is connected to the interrupt handler of the microblaze processor system. The pmod BLE IP is connected to pmod port D on the NEXYS A7 board.

2) Pmod OLED rgb IP : The pmod oled ip is used to form a hardware connection for the pmod oled display module on the pmod port A of the NEXYS A7 board. The module is connected to the microblaze AXI peipheral ip to form connection with the microblaze processor.

3) Pmod ACL accelerometer IP: The pmod ACL ip is used to form hardware connection with the pmod port B on the NEXYS A7 board and the pmod ACL module.

A hardware wrapper file is formed after successfully generating the bitstream of the whole block design integration. The HDL wrapper is acts as an underlaying file with hardware informations for further development of the software top layer. The HDL wrapper is exported to the Vitis IDE.

VI. Software Implementation

This section focuses on the software implementation.

A. C code Libraries

```c
#include "PmodBLE.h"       //pmod bluetooth library
#include "bitmap.h"
#include "PmodOLEDrgb.h"  // pmod oled library
#include "sleep.h"        // c code delay library
#include "xil_cache.h"
#include "xparameters.h"   // vitis primary library
#include "xil_types.h"
#include "xil_io.h"
#include "PmodACL.h"      // pmod acl library
#include <stdio.h>
#include <stdlib.h>
```

Fig. 13. Libraries included

The libraries used for the software c programme development are shown in figure 13. The pmod bluetooth, ACL and OLED display libraries are offcical libraries provided by digilent for the respective pmod modules. The sleep.h and xparameters.h libraries are Vitis IDE specific libraries used for delay functions and board specifications respectively.

B. Initialization and Setup

```c
void Initialize()
{
    EnableCaches();          //enable cache data from microblaze
    SysUartInit();           // initialize uart for bluetooth
    BLE_Begin (              // blue tooth initialization
        &bluetooth,
        XPAR_PMODBLE_0_S_AXI_GPIO_BASEADDR,
        XPAR_PMODBLE_0_S_AXI_UART_BASEADDR,
        BLE_UART_AXI_CLOCK_FREQ,
        115200
    );
    OLEDrgb_begin(&oledrgb, XPAR_PMODOLEDRGB_0_AXI_LITE_GPIO_BASEADDR,  // initialize oled display
            XPAR_PMODOLEDRGB_0_AXI_LITE_SPI_BASEADDR);

       ACL_begin(&ACL, XPAR_PMODACL_0_AXI_LITE_GPIO_BASEADDR,XPAR_PMODACL_0_AXI_LITE_SPI_BASEADDR);
        SetMeasure(&ACL, FALSE);                    // initialize ACL module
        SetGRange(&ACL, PAR_GRANGE_PM4G);
        SetMeasure(&ACL, TRUE);
        CalibrateOneAxisGravitational(&ACL, PAR_AXIS_ZP);
}
```

Fig. 14. Initialization

The figure 14 shows initialization for microblaze cache data, bluetooth module, ACL module and OLED display. The initialization functions are provides by the libraries, for example the BLE_Begin() function is from the PmodBLE.h library. The module base addresses for example XPAR_PMODACL_0_AXI_LITE_GPIO_BASEADDR is the base address for the pmod acl module and is automatically generated by the HDL wrapper and can be found in the xparameters library.

C. Main function logic

The main function of the program is shown in figure 15. The x , y , z variables are created to store the values of the 3D motion. stringconvertX variable stores the value of

```c
 float x;     // variable for x axis location
 float y;       // variable for y axis location
 float z;      // variable for z axis location
 int i = 0;
char stringConvertX[12];        // convert float x value to a string
   char stringConvertY[12];
   char stringConvertZ[12];

while(1) {
    ReadAccelG(&ACL, &x, &y, &z);     // read x, y, z values from pmod acl
          sprintf(stringConvertX ,"%f", x);   // conversion of float x to string
          sprintf(stringConvertY ,"%f", y);
          sprintf(stringConvertZ ,"%f", z);

          usleep(10000);
      OLEDrgb_SetCursor(&oledrgb, 0, 0);         // setting the location for the cursor on first line, first alphabet
      OLEDrgb_SetFontColor(&oledrgb, OLEDrgb_BuildRGB(0,255, 0));  // choose color, red
      OLEDrgb_PutString(&oledrgb, "X=");
      OLEDrgb_PutString(&oledrgb, stringConvertX);           // print the x component value on the oled display

      usleep(1000);

      OLEDrgb_SetCursor(&oledrgb, 0, 2);         //setting the cursor on the second line, first alphabet position
      OLEDrgb_SetFontColor(&oledrgb, OLEDrgb_BuildRGB(0, 0, 255)); // green color
      OLEDrgb_PutString(&oledrgb, "Y=");
      OLEDrgb_PutString(&oledrgb, stringConvertY);
```

Fig. 15. Main logic

```c
BLE_SendData(&bluetooth, "X= ", 3);
usleep(1000);
BLE_SendData(&bluetooth, stringConvertX, 8);      // send x component value over bluetooth
usleep(1000);
BLE_SendData(&bluetooth, "___", 3);
usleep(1000);

BLE_SendData(&bluetooth, "Y= ", 3);
usleep(1000);
BLE_SendData(&bluetooth, stringConvertY, 8);      // send y component value over bluetooth
usleep(1000);
BLE_SendData(&bluetooth, "___", 3);
usleep(1000);

BLE_SendData(&bluetooth, "Z= ", 3);
usleep(1000);
BLE_SendData(&bluetooth, stringConvertZ, 8);      // send z component value over bluetooth
usleep(1000);
BLE_SendData(&bluetooth, "___", 3);
```

Fig. 16. Bluetooth Transmission

3D motion in a string data type. The ReadAccelG(&ACL, &x, &y, &z) function reads value of the respective x,y,z components of the 3D motion using the pmod ACL sensor. sprintf() function then converts the float data type memory in x,y,z varibale to string data type and stores it in stringconvertX, stringconvertY and stringconvertZ respectively. The OLEDrgb_SetCursor function sets the cursor on the oled display. The OLEDrgb_SetFontColor() function sets the color of the font to be displayed on that specific line. The OLEDrgb_PutString(&oledrgb, stringConvertX) function displays the charachters of the string variable stringConvertX on the oled display.

The figure 16 shows the code for transmitting data via the Bluetooth module. The BLE_SendData() function sends data of the stringConvertX variable of string type over the bluetooth.

VII. Implementation Results

The Serial Bluetooth Terminal application shown in figure 19, from Google Play was used for reading messages transmitted on bluetooth.

VIII. Conclusion

To conclude, every single aspect of the project has been documented in this report. I have explored and learnt various things from this project. The project deepens my understanding about concepts of hardware and software integration, use of Vivado/Vitis design suite and the operation of pmod modules. The project also helped me develop a global view of working

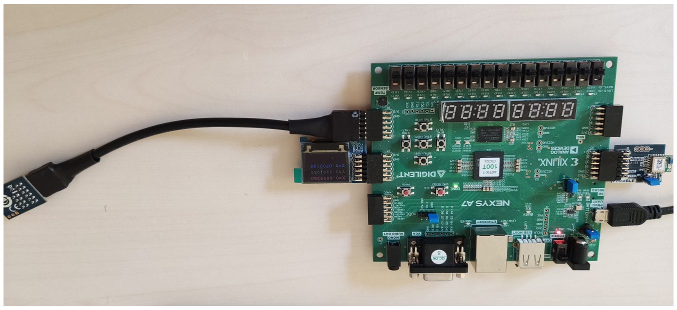

Fig. 17. Image of the whole hardware system

with embedded system and can prove to be crucial for my further career.

IX. Declaration of Originality

I, Yashodhan Vishvesh Deshpande, herewith declare that I have composed the present paper and work by myself and without the use of any other than the cited sources and aids. Sentences or parts of sentences quoted literally are marked as such; other references with regard to the statement and scope are indicated by full details of the publications concerned. The paper and work in the same or similar form have not been submitted to any examination body and have not been published. This paper was not yet, even in part, used in another examination or as a course performance. I agree that my work may be checked by a plagiarism checker.

References

[1] Nexys a7 reference manual - digilent reference.
[2] Pmod ACL - digilent reference.
[3] Pmod BLE - digilent reference.
[4] Pmod OLEDrgb - digilent reference.
[5] USB-a/micro-USB-kabel (1 m, schwarz).
[6] Vitis unified software platform.
[7] Vivado overview.

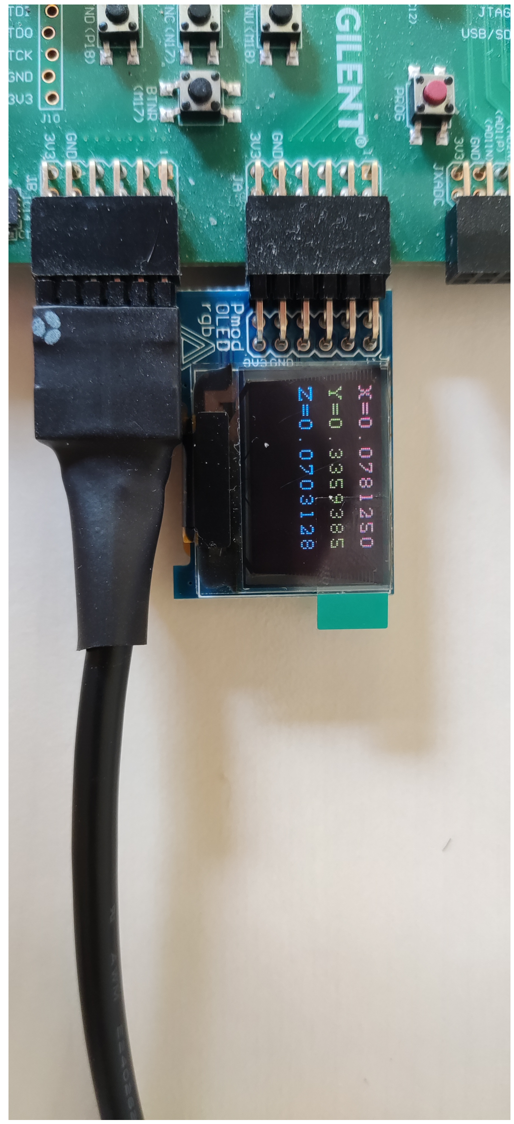

Fig. 18. Pmod OLED display showing results

Fig. 19. Bluetooth data transmission successful

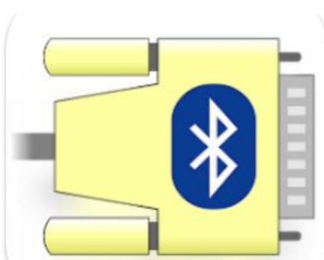

Fig. 20. Serial Bluetooth Terminal app