

Design and Development of 3D Motion Sensing Gaming Platform using FPGA

Liu Xinming, Wang Chongsen, Wang Zhiheng

Information Institute, Beijing Jiaotong university

111125094@bjtu.edu.cn, 211120426@bjtu.edu.cn, 309281149@bjtu.edu.cn

Abstract — The most advantage of FPGA is its hardware/software co-design. Based on this advantage, we can make full use of hardware pipeline, multi-core architecture, hardware acceleration technologies to accelerate their applications, in order to mitigate or completely solve high real-time, multi-tasking, big data and a series of other problems. This design aims at completely achieving a 3D Motion Sensing Game System on the FPGA platform. The game system has to support voice and image output, and collects the real-time in-motion data wirelessly through sensors in the mobile phone to produce human limb movement instruction for game control. This system is multi-tasking, massive data and strict real-time. The final results show that the game system designed with FPGA technology, not only runs smoothly, fast response time, but also has good scalability, the ability to easily maintain and upgrade, and therefore has a very good prospect.

Keywords — FPGA, motion sensing game, hardware/software co-design, hardware acceleration, CG

I. INTRODUCTION

It is a big challenge to design 3D motion sensing games for general embedded playform. An excellent game should has efficient and engaging game mechanic, as well as the fluid interactions and real-time response. Therefore, it is a multi-tasking, a large amount of data, real-time system.

1) **Multi-tasking:** Image display, voice output, game scene rendering, sensor data acquisition and analysis. These tasks must be handled in parallel, otherwise it will be difficult to maintain the fluency of the game.

2) **Large amount of data:** Mainly in two points. The first is the need to continuously refresh the screen. Supposing the refresh rate of 60Hz, the data to be read is about $640 * 480 * 4 * 60 = 70M$ bytes per second . The other point is the need to constantly redraw the animation of the game. In order to keep the game smooth animation, the frame rate (FPS) is at least 30 per second and the data to be written is about $640 * 480 * 4 * 30 = 35M$ bytes per second. It is difficult for a normal low frequency embedded systems.

3) **High real-time:** The game system will inevitably require friendly human-computer interaction and quick response and action. Sound and images should not have delay and distortion. Therefore not maintaining high real-time nature, the game will not be able to ensure the quality of the game.

FPGA technology has intrgrated the respective advantages of software and hardware to get the highest fluency game experience. Logic complexity and less time-used task can be considered to be implemented in software. But to the time-consuming task with relatively simple logic module, hardware implementation is better, such as high-volume data transmission.

FPGA hardware acceleration's core meaning is that it can be completely independent of a single CPU to build a multi-core system, that is, to build a multi-tasking real parallel environment, rather than a virtual parallel environment which is a timeshare multitasking system in fact. There are many foreseeable problems in this project, such as multitasking requirements of the system, large data processing, demanding strict real-time systems and so on, but all can be solved one by one.

II. ARCHITECTURE

The system will implement a 3D bowling game. The hardware components have five parts: Android phone, BT5701 (Bluetooth wireless serial communication adapter), DE2-70, monitors, speakers, as shown in Figure 1.

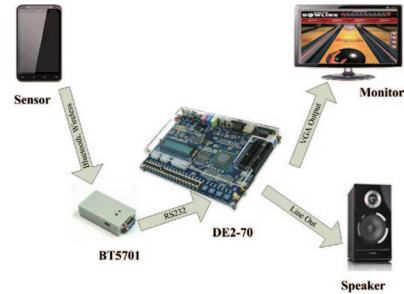


Figure 1. Architecture of game system

This game not only support the basic elements of a bowling game, such as scoring, perception of dynamics, movement trail of the ball, the final status of the bottles, etc, and also offers a variety of game options. For example, you can choose two different game scene, and two-player mode or single player mode. In order to enhance the entertainment value of the game, the system increases the function of pitching rating and pitching tips.

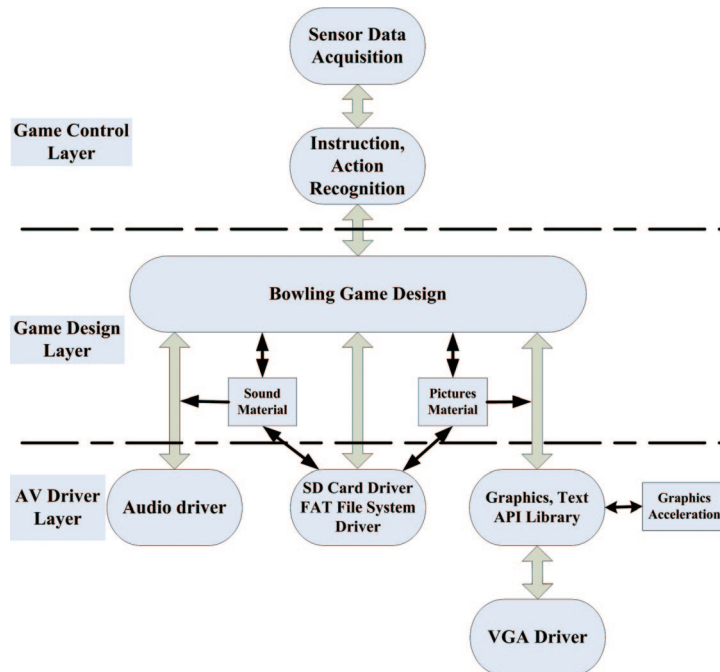


Figure 2. Software structure

This system uses a three-layer structure, the structure diagram is shown in Figure 2.

A. Game control layer

As the control end of the game, this layer acts as the mouse and keyboard. Android phone equipped with a three-axis acceleration sensor and magnetic field sensor collects this information and sends it via Bluetooth to DE2-70. Then our gesture recognition algorithm will give an accurate analysis of gesture on behalf of the instruction, thereby triggering the appropriate response of the game.

B. AV Driver layer

This layer is the basis of the entire game. It not only provides sound driver, VGA driver, SD card driver and FAT file system driver which designed for the SD card read & write operation, but also provides a rich graphics rendering APIs. In order to improve the operating efficiency, we implement the key tasks or key methods in hardware, for example, parts of the methods of the graphics APIs, etc.

C. Game design layer

The part is the core of the entire system to complete the design of the entire bowling game. Its implementation is based on the driver layer, in response to an instruction from the control layer. There has a clear demarcation between it and the control layer, because the driver layer and the control layer provides the interfaces independent of the game. That is to say, we can design a brand new game, without having to change anything of the control layer and the driver layer.

III. SYSTEM DESIGN

The system design is divided into two parts: the hardware and the software. The hardware part of the work is to build a multi-tasking operating environment and provides a control interface for external devices. We design the hardware environment shown in Figure 3.

In the picture, we use SDRAM1 and SDRAM0 to support double buffering mechanism for the video display. The UART module is designed

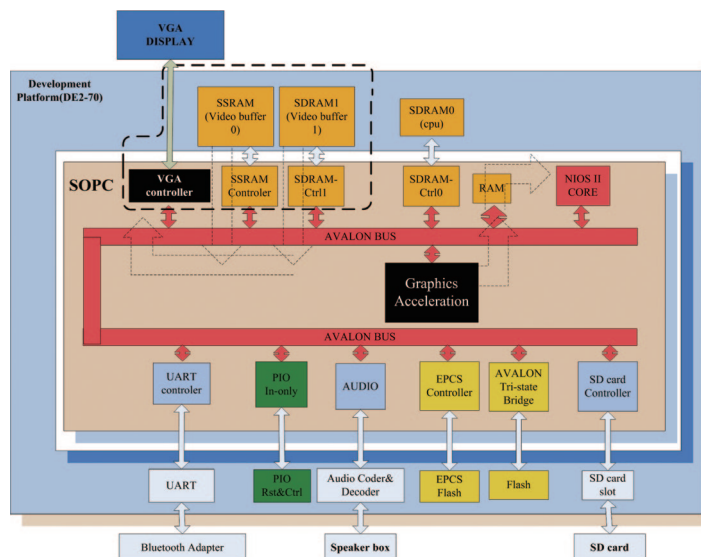


Figure 3. NIOS II system

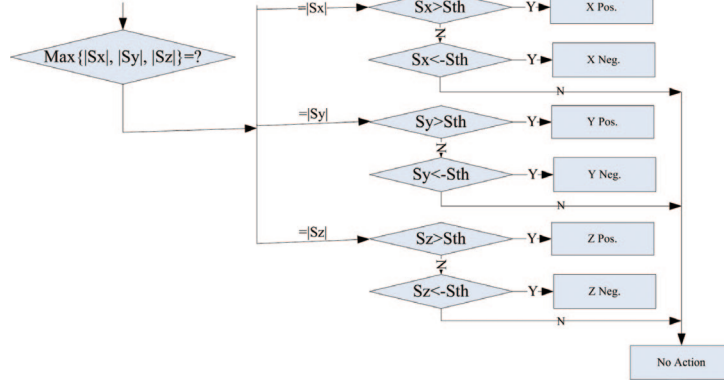


Figure 4. Gestures judge

for receiving the data sent by the phone, and the AUDIO module is for playing the sound effect.

In order to improve the image display and rendering quality, we design a VGA Controller and a Graphics Acceleration IP core, to provide support for the game of high-quality animation.

As described above, the system software uses the three-layer structure design and all details of the respective layers of the system will be elaborated below.

A. The game control layer

The raw data acquired directly by the sensor is, from the physical sense, acceleration, rather than speed, and therefore can not be directly used for determining operation gestures. We denote the speed in the x-direction S_x , and denote two adjacent sampled acceleration A_{x0} , A_{x1} , and the time difference between this two samples is ΔT_x , then

$$S_x = \frac{A_{x1} - A_{x0}}{\Delta T_x} \quad (1)$$

Similarly, we can calculate the speed in the other two directions: S_y and S_z . Then we can identify the gesture command according to the method in Figure 4 and the S_{th} is speed threshold.

So, this method can identify six kinds of operation: the X-axis positive direction shaking, the X-axis negative direction shaking, the Y-axis positive direction shaking, the Y-axis negative direction shaking, the Z-axis positive direction shaking and the Z-axis negative direction shaking.

Not only that, in the bowling games we have to know the direction of motion and the ball speed when tossing. According to the previous calculations, the motion direction and ball speed comes directly from the speed vector (S_x, S_y, S_z) .

B. The AV driver layer

As the bottom of the system, on the one hand, the upper support of sound and image, this layer provides all kinds of drivers and plays a very important role. Another important function is to provide the picture drawing, text display API library. The effect on implementation efficiency the of the graphics library of games is so important, that we must implement a part of the API in hardware, instead of in software, which greatly improves the game frame rate (FPS).

This layer can be divided into 4 parts: audio driver, VGA driver, SD card driver and FAT file system driver, as well as general purpose graphics text API libraries. SD card driver and FAT file

system driver provide support for reading and writing files on the SD card. There are lots of pictures and sound clips locating on the SD card, and we must loading them into memory before using them. The audio driver can be used to play background music and foreground music. The following paragraphs give the details on the VGA hardware design and some graphics API approach by hardware.

(1) VGA driver design

As previously mentioned, if you want to support a resolution of 640 x 480, refresh rate of 60Hz image display, 70M bytes of data will be transferred in one second. If this transmission is implemented by software, it will be a maximum of 4.5M bytes that we can transfer in one second. This shows that we must consider using hardware approach to accomplish this task, and can not affect the execution efficiency of the software, that is to say, we cannot occupy the CPU time too much.

Avalon specifications provide a variety of interface protocols for custom peripherals. There are a large number of data transmission protocols, such as Avalon-MM Master / Slave and Avalon-ST interface. As memory devices provide read and write interfaces via Avalon-MM Slave interface in SOPC, we have to use the Avalon-MM Master to reading and writing memory devices positively. VGA IP core hardware module prototypes we design is shown in Figure 5.

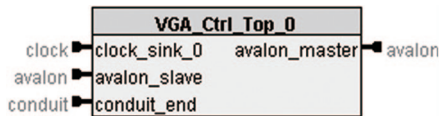


Figure 5. VGA IP core prototypes

1) **Avalon-MM Master:** The working clock of VGA interface is 25M Hz, but that of the Master is 100M Hz. It must take less than 4clocks to complete a read operation (4 bytes), which should

be ensured. We also know that the Avalon system uses the slave-side arbitration mechanism, when multiple Master post a read or write requests to the same Slave at the same time. CPU's performing frequently memory read and write will have greatly effects on the work of the VGA IP core. So this IP core should use a line buffer inside.

2) **Avalon-MM Slave:** This interface is used for CPU to control the IP core. You can start and stop the work of the IP core, and set the VGA buffer memory address, which makes it possible to work with double-buffer mechanism by changing the buffer address anytime.

3) **Conduit:** This interface directly faces at the D15 VGA interface signals. Therefore, the important responsibility of this interface is to generate VGA timing with the RGB data provided by the Avalon-MM Master interface. The VGA interface horizontal timing is shown in Figure 6.

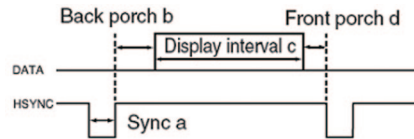


Figure 6. VGA horizontal timing

VGA vertical timing is similar to horizontal timing. Table I gives the length of each part in the horizontal timing and vertical timing.

TABLE I
VGA HORIZONTAL TIMING AND VERTICAL TIMING

Parts	Sync	Back porch	Display interval	Front porch
Horizontal timing (clocks)	96	48	640	16
Vertical timing (lines)	2	31	480	11

(2) Graphics acceleration module design

The IP core module is used to accelerate graphics drawing. Currently supported hardware

operation are displaying images (in transparent or non-transparent way), drawing rectangular boxes(in filled or non-filled way). These operations are the high frequency of use in the scene design. Implementing in hardware way can greatly improve the game frame rate. Figure 7 is the IP core module interface prototype.

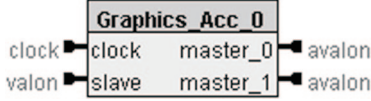


Figure 7. Graphics Acceleration IP core prototype

The IP core uses an Avalon-MM slave interface and two Avalon-MM Master interfaces. Slave interface is used to select the operating mode, and set all the input parameters. There are 10 registers using for storing the input parameters: ParamReg1 ~ ParamReg10. Table 2 shows the registers of the IP core function settings.

TABLE II
GRAPHICS ACCELERATION IP CORE REGISTER TABLE

Address Offset	Name	Access	Display images	Drawing Boxes
0x00	Control	RW	Control and operation mode registers	
0x04	Status	R	Completion status	
0x08	ParamReg1	RW	Image width	Rectangle width
0x0C	ParamReg2	RW	Image Height	Rectangle Height
0x10	ParamReg3	RW	Data address	Start x
0x14	ParamReg4	RW	Start x	Start y
0x18	ParamReg5	RW	Start y	Unused
0x1C~0x2C	ParamReg6~10	RW	Unused	

The IP core module also uses the two master interfaces, one for reading image data from the source address, and another for writing data to memory. This is similar to the DMA functionality, but DMA can only write to consecutive addresses, and therefore is not suitable here. After acceleration, Improved performance is shown in Table III(all operations are on the compiler optimizations: GCC-O3).

TABLE III
GRAPHICS ACCELERATION

Operations	Software way	Hardware Way	Ratio
Display Images (non-transparent, full screen)	97.4ms	3.2ms	30.4
Drawing Boxes (filled, full screen)	53.2ms	3.1ms	17.2

Obviously, Implementation in hardware way will greatly enhance the speed of drawing, and ensure the fluency of the game.

C. The game design layer

This layer is the detailed design on the game. It uses the interfaces provided by control layer, and gets gestures instruction, and then calls graphics and voice APIs provided by driver layer to play music and draw game.

There are four big tasks: sensor data acquisition and instruction recognition, image rendering, VGA output, audio output. There is a big problem: these four tasks should concurrent working. Among these tasks, image drawing is our main jobs. Our design combines the advantages of software and hardware to solve the problem.

1) Data acquisition and instruction recognition: Data acquisition is in mobile end, and UART works with interrupt mode to receive data and identify gestures.

2) VGA output: We introduce detailedly the VGA driver design previously, and it is a whole hardware way. It works fully parallelly with CPU.

3) Audio output: Audio output hardware module has a buffer zone, enough to store 5 ms voice data. We trigger a timer interrupt in every 5 ms to fill enough data, so as to ensure the continuous broadcast voice.

4) Image drawing: This as our main task, and it shares time with "command recognition" task and "audio output" task, but is completely parallel with "VGA output" task.

In addition to the above four major tasks, bowling game 3D motion trail modeling is another problem.

We adopt the following physical model : a bowling's initial position is (x_0, y_0, z_0) , and initial velocity (the speed when the ball is out of hand) is (v_x, v_y, v_z) , and initial acceleration (the acceleration when the ball is out of hand) is $(0, 0, g)$, and the reverse friction force acceleration is $(-f_a, -f_b, 0)$. Ground bounce coefficient is k , and we assume that z axis speed denotes 0 when z axis speed is less than v_{zmin} , regardless of the air interference.

According to this physical model, we can easily figure out the motion trail, so the relevant calculation is omitted here. It is proved that this model can completely meet the requirements by using Matlab simulation tool.

CONCLUSIONS

We introduce the game design idea and implementation steps in detail, previously. Eventually we implemented the somatosensory game which demonstrated excellent performance: gesture recognition rate is 85% or more, and game frame rate is average of 40fps, and can play background music continuously at the bit rate of 44.1KHz without interfering with other tasks. These benefits are from the hardware and software combination design. Based on the scalability of the system architecture and FPGA devices' reprogramming, we can easily add new game design or just perform maintenance and upgrade. In short, based on the thinking of hardware and software co-design, the use of FPGA technology can facilitate the construction of multi-parallel processing module, multi-processor core hardware system. This design gives full play to the advantage of FPGA hardware technology, and show the FPGA great advantage and potential in the field of video images, somatosensory games.

PROSPECTS

Game designers must find a balance between the complexity of the system, reliability and low cost. However, to be successful in business, game console must be able to provide a variety of functions, such as high-resolution images, the network connection, motion sensing controllers, and IP-based video control, and all of these must be as low price as possible. In order to maintain low-cost characteristics, usually module reuse of methods is encouraged in designing a game console. FPGA hardware flexibility makes the game system to be easy for different regions, model, component redesign. FPGA replaces the role of ASIC in many applications, perhaps it is the time to use it for the next generation of game design.

REFERENCES

- [1]. Liu Tao, Xinghua Lou. FPGA digital electronic systems design and development. Beijing: people's Publishing House of posts and telecommunications, 2005.6.
- [2]. www.altera.com, Cyclone FPGA family, NIOS soft CPU
- [3]. V.Beller et al, An FPGA based Floating Point Processor Supporting a High Precision Dot Product, FPT'2006, Bangkok
- [4]. A. Hanoun et al, Reconfigurable Cell Architecture for Systolic and Pipelined Data Paths, ReConfig08, Cancun, 2008
- [5]. <http://ptolemy.eecs.berkeley.edu/ptolemyII>
- [6]. F. Mayer-Lindenberg, Dedicated Digital Processors, Wiley Interscience 2004
- [7]. R. Lysecky, K. Miller, F. Vahid, and K. Vissers. Firm-core Virtual FPGA for Just-in-Time FPGA Compilation. Proceedings of the 2005 ACM/SIGDA 13th international symposium on Field-programmable gate arrays, 2005.
- [8]. A.Kumar et.al., Reconfigurable Multi-Processor Network-on-Chip on FPGA, ASCI 2006