Rich Graphics Solution for Embedded Device - GA88 Series IWAYAG -

MIYAUCHI Yoshihito, AIBARA Shinji, KUROKAWA Takafumi, KAMOTANI Ken

Abstract

The "Rich UI" (user interface), so to speak, which features a touch panel and animations, is in use with recent smartphones and tablet terminals. Rich UI offers the benefit of intuitive operability, however, since magnified and reduced screen displays are used heavily, the rendering process capability of the CPUs built into embedded devices at times becomes insufficient. In some cases 3D graphics hardware is implemented on to counter this, however, there have been cases when the amount of hardware becomes so much that the power consumption increases, or the processing capabilities cannot be fully utilized, depending on the type of content. This paper introduces the GA88 series "IWAYAG," which optimally realizes graphics necessary for the Rich UI of embedded devices, using vector graphics technology.

Keywords

vector graphics, rich UI, embedded device
OpenVG, intellectual property

1. Introduction

A large number of Rich UIs, featuring a touch panel and utilizing magnification and reduction functions, as well as animations, is starting to be used on smartphones and tablet terminals, as well as car navigation systems, digital cameras and a variety of embedded devices.

The rendering of images, such as magnification and a reduction in the Rich UI requires an extremely large number of calculations to change the shapes and colors of graphic forms, in comparison with conventional GUI (Graphical User Interfaces). In order to display animations in a smooth manner, it is also necessary to render images at a high speed. Since the CPUs of embedded devices have power and price constraints, this can lead to the occurrence of problems, such as delayed rendering due to insufficient performance, resulting in inconvenient use.

The GA88 series "IWAYAG" is a graphics IP (Intellectual Property) product that utilizes graphics technologies that have been nurtured over many years and one which has promptly responded to the latest technology trends. We offer graphics intended for an embedded device that has as the core this IP, as well as related solutions that combine the product with services.

2. Graphics of Embedded Devices

Once bitmap displays became popular as the displays of user interface for embedded devices, GUIs became more common. Since the characters and icons of the same size are arranged with GUIs, the amount of information that can be displayed on a screen is limited. It is for this reason that windows are used to overlap the information, switch screens with tags or the information is displayed in a hierarchical manner. There are also disadvantages to this, however, such as the difficulty of knowing if the information is hidden under overlapped windows or which of the tagged window contains it, or where in the hierarchy it is contained, making it necessary for the user to keep track of the position of the information.

The overall information can be magnified and reduced in a scalable manner and feature operations that are easily understood, such as scrolling to all directions, achieves an overview of the entire information and provides a close up of the necessary information with Rich UI, which recently is starting to be used. A characteristic of the UI is the use of a touch panel, which makes it possible to operate the embedded device in an intuitive manner.

The graphics processing method used with Rich UI varies in realization methods according to the types of graphics hardware mounted on the embedded device. Such differences and features are described below.

(1) 2D graphics

The CUP must perform rendering calculations, such as magnifying and reducing graphical images, with 2D graphics hardware that has no hardware engine for rendering graphical images. Extremely large amounts of calculations and writing onto the memory occur, relying on CPU processing results in a slow rendering speed and poorer operability. Furthermore, CPU execution times may be taken up by graphics processing, resulting in delays to other processes.

(2) 3D graphics

A large amount of logic and memory are required with 3D graphics for sophisticated functions and high performance geometric engines, rendering engines and Z buffering for the depth coordination of rendering overlapped objects. The realistic display of animations for 3D content becomes possible with the implementing of 3D graphics hardware. Since the graphics process can be passed over to the hardware, the load on the CPU can also be reduced. Menus and the web, as well as character displays of Rich UI content, are often so called vector rendered images that are not 3D but are magnified or reduced in a two-dimensional coordinate system. In such cases, the objects of vectors must be converted into 3D objects before being rendered.

Although it is possible to render at high speeds, in some cases the conversion to 3D, such as the rendering of vector objects by dividing them into polygons, can become bottlenecks or in some cases hardware resources for 3D graphics cannot be fully utilized since light source processing and depth processing are not available.

(3) Vector graphics

Vector graphics maintain the shapes of graphical images with two dimensional coordinates, therefore expansions, reductions and other such renderings are performed by converting the coordinates in two dimensions. If in some cases 2D bitmap images are magnified, the edges of the graphical images become rough as the dots are magnified, but no deterioration to the graphics quality occurs with the magnifications or reductions of the vector graphics, since the calculations are performed with a rendering scale for each rendering of the lines between the coordinates. Furthermore, a feature of vector graphics is the function to render images using smooth curves between coordinates, referred to as the Bezier curve. Although the flow of processing for vector graphics is similar to that of 3D graphics, the logic and memory usage is relatively low and power consumption is also low, as calculations or verifications for depth directions, light source calculations or sophisticated texture processing, are not performed.

3. Graphics Optimum for Embedded Devices

Methods for realizing graphics diverge depending on the type of graphics hardware available, as described above. Which of these is best suited for the embedded device must be determined. First of all, 2D graphics places a high load on the CPU and since it is significantly inferior in terms of calculating performance, rendering performance, individual processing stage parallel processing capabilities and other such performances, in comparison with other dedicated hardware, such as 3D graphics or vector graphics, it is quite clear that this is unsuitable. The remaining candidates are 3D graphics and vector graphics. When 3D content is required, 3D graphics would naturally be necessary. The circumstances for which 3D graphics are necessary are limited to games that involve the realistic movement of 3D characters and 3D objects, scenery simulations or a specially prepared 3D user interface. There is a lot of other content that can be processed by vector graphics, such as Rich UI that does not use 3D graphics, web pages, maps or character displays using vector fonts. Furthermore, vector graphics have a function that can transform images as if they are projected in perspective, which can simulate 3D graphics. Techniques, such as blending that involves the overlapping of layers to synthesize images to express shadows and depth, are also available. Vector graphics are often more advantageous in terms of cost and power consumption, depending on the grade and purpose of the products (**Fig. 1**).

Fig. 1 Examples of rich expressions using vector graphics.

In addition, 3D graphics and vector graphics are significantly different in the ways in which the content is created. Modeling, motion and view processes, as well as projection simulations from 3D to plane surfaces, must be assumed in order to create content with 3D graphics. On the other hand vector graphics require the depiction of a start and end using a content preparation tool, whereas all of the images in between can be generated automatically. A characteristic of this method is that a designer can easily create contents he or she imagines.

4. IWAYAG

GA88 series IWAYAG is a graphics IP product for embedded devices, developed by NEC System Technologies. The GA88 series is a series of graphics IP products, whereas IWAYAG is the second phase of products that succeeded ISHITEGI. This is configured by a graphics hardware IP embedded inside an LSI or FPGA, as well as a driver for controlling this IP as a set (**Fig. 2**). The product features an IP that is dedicated to vector graphics, with the vector graphics standard API OpenVG process formulated by the standardization organization, Khronos, accelerated by the following methods and hardware:

- Each processing stage of the vector graphics is processed in parallel.
- A high-speed numerical data calculator with optimum accuracy is used for calculation processes.
- An image cache is used for improving the rendering performance and inhibiting access conflict.
- An outlining process is used for efficiently filling graphic forms.
- A Bezier curve rendering engine is used for the neat and high-speed rendering of curves.
- An Antialias process is used for the smooth rendering of lines and borders.

Other than the above, responsive action is provided for porting the product to a customer's existing systems, for changes in specifications or additions to functions that are considered necessary locally. Furthermore, the software tuning of a customer's systems utilizing know-how, such as eliminating wasteful copying between memories or abbreviating redundant processes, as well as batch processing, are also provided.

Moreover, it is also possible to provide solutions that are comprised of product developments, applications and content in response to the requirements of customers through tie-ups with technologies and products of the NEC Group.

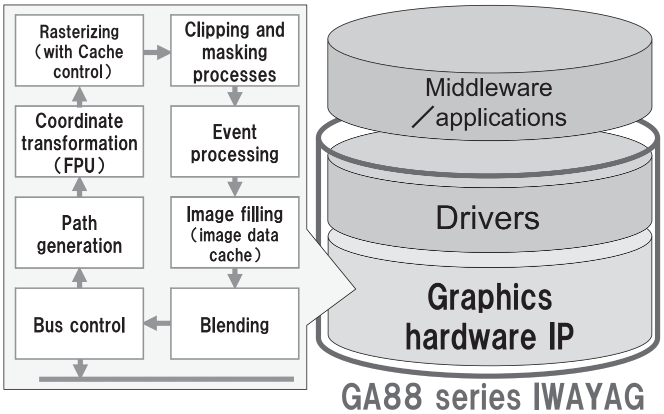


Fig. 2 Configuration of GA88 series IWAYAG.

Effects that can be expected by using IWAYAG are as follows:

- High-speed rendering that is at least double the rates provided by CPUs.
- Reduces the load on CPUs.

The results of the actual measurements on specific content, taken on evaluation devices with an ordinary field programmable gate array (FPGA), with IWAYAG implemented onto the FPGA to verify such effects, are introduced. In this case example, a comparison between rendering using a CPU alone and with IWAYAG indicated an improvement in the rendering speed from 13 frames per second to 43 frames per second and the performance by about 3.3 times, with a 55% reduction of the CPU usage rate realized, from 94% to 39% (**Fig. 3**).

The process of this mechanism that delivers such effects starts with the CPU preparing commands and the data of graphics based on the rendering instructions from applications by loading them onto memory first, when rendering with IWAYAG. Once the loading of commands and data has been completed, the CPU issues instructions for rendering to IWAYAG and IWAYAG sequentially reads the commands and data loaded onto memory and performs rendering automatically. Since the CPU merely prepares commands and data, a margin for processing is gained. Since IWAYAG sequentially reads commands as well as the data and performs processes using several acceleration functions described earlier, high-speed processing is possible.

Since this example is a case when IWAYAG is used on FPGA, IWAYAG runs at a relatively low frequency of 50MHz, but when actually implemented on an LSI the frequency of

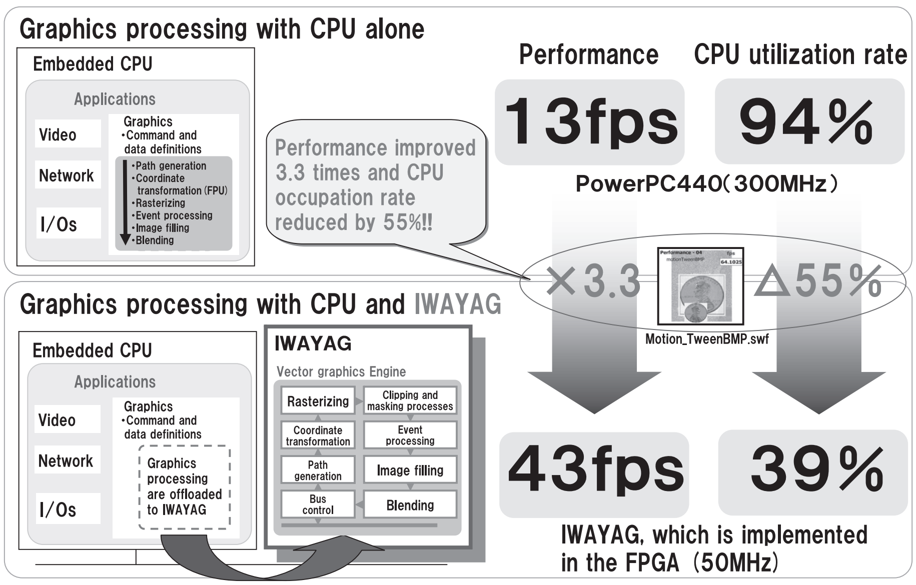


Fig. 3 Effects of IWAYAG.

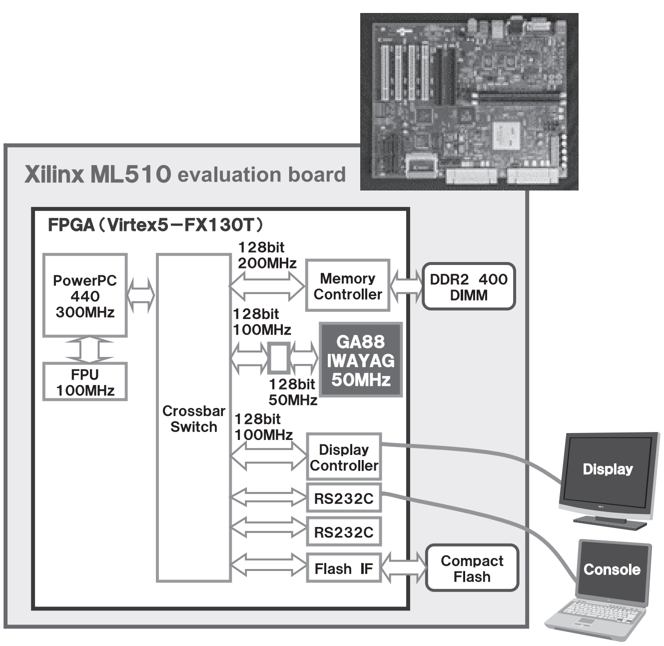


Fig. 4 Configuration of FPGA evaluation device.

200MHz is assumed, thus a far superior performance can be expected (**Fig. 4**).

5. Conclusion

Rich UI, which started with smartphones and tablet terminals, has now spread to car navigation systems and digital cameras. Further spread is expected, as advancements in technologies increase the scale and reduce the costs of LSIs and FPGAs, as well as through mass production effects of the display panels, which will lead to the lowering of prices and other such movements. Moves are being made to implement Rich UI even on embedded devices with lower graphics capabilities in the future, such as home electronics and machinery for industrial use, as well as various dedicated equipment. Vector graphics are suitable for these products, as they are often required to be supplied at moderate prices and with lower power consumption.

Furthermore, in terms of web technologies the formulation of HTML5 is progressing and high performance 2D and vector graphics technologies, such as Canvas and SVG (Scalable Vector Graphics), are included as graphics specifications. Scenes, wherein vector graphics are utilized, are expected in the future also with cloud terminals and embedded devices that support HTML5, which are expected to be produced commercially in the future.

We are aiming to provide solutions that increase added val-

Rich Graphics Solution for Embedded Device - GA88 Series IWAYAG -

ue to embedded devices by providing IP products that respond rapidly to the trends of technologies and markets, as well as the software of upper layers and platforms.

*PowerPC is a trademark and/or a registered trademark of International Business Machines Corporation.

*CompactFlash is a trademark and/or a registered trademark of SanDisk Corporation.

*Virtex is a trademark and/or a registered trademark of Xilinx Inc.

Reference

1) Khronos "OpenVG Specification Version 1.1 December 3, 2008"

Authors' Profiles

MIYAUCHI Yoshihito
Senior Expert
Server Platform Division
Platform Operations Unit
NEC System Technologies, Ltd.

AIBARA Shinji
Manager
Business Promotion Division
NEC System Technologies, Ltd.

KUROKAWA Takafumi
Manager
Server Platform Division
Platform Operations Unit
NEC System Technologies, Ltd.

KAMOTANI Ken
Manager
Embedded Software Division
Platform Operations Unit
NEC System Technologies, Ltd.

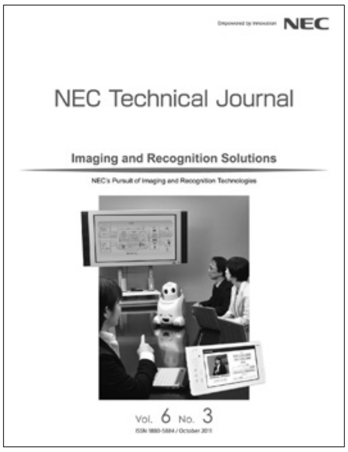
Information about the NEC Technical Journal

Thank you for reading the paper.
If you are interested in the NEC Technical Journal, you can also read other papers on our website.

Link to NEC Technical Journal website

Japanese | English

Vol.6 No.3 Imaging and Recognition Solutions



Vol.6 No.3

October, 2011

Special Issue TOP