



An Independent VGA Controller Based on SOPC with Three Pixel-Mapped Schemes

Zerun Li^(✉), Yande Jiang, and Yang Guo

College of Computer, National University of Defense Technology,
Changsha, China
li_zr@foxmail.com

Abstract. As a standard display interface, VGA (Video Graphics Array) has been widely used. In this paper, we propose an independent VGA controller, and the CPU (Central Processing Unit) does not need to control and transmit data, which can save hardware resource and enhance data processing speed, compared to the regular VGA design. Specifically, the controller consists of a synchronizing module, a memory module, and a palette module. We implement three pixel-mapped schemes, including bit-mapped scheme, block-mapped and object-mapped scheme, compared with the traditional mapping scheme. Their signal activities are 5.00×10^7 , 1.67×10^7 , 3.33×10^7 and 5.67×10^7 respectively which measure the display efficiency of the VGA controller. Their synthesized registers are 2348, 2412, 2560 and 2072, which reflect different resource utilization. Our functional simulations and logic syntheses prove that the proposed VGA controller design has strong flexibility, short design cycle, and low production cost under the provided circumstances of application.

Keywords: VGA · FPGA · SOPC · Embedded system

1 Introduction

SOPC (System on Programmable Chip) can realize the main function of the system on an independent FPGA (Field Programmable Gate Array) development board [1]. SOPC integrates memory, logic unit, I/O interface, central processor, bus interface, and other system design necessary functional modules into a PLD (Programmable Logic Device), and consists of a programmable logic system [2]. The design of SOPC is very flexible and its processor system can also be programmed, cut, upgraded, expanded, maintained easily and can be programmed for both software and hardware [3].

The VGA interface is one of the most important interface to a computer monitor and has been used since the age of the bulky CRT (Cathode Ray Tube) display. It is still used today and also known as the D-sub interface. Apart from HDMI (High Definition Multimedia Interface), VGA can also display images of

the 1080P to achieve higher resolution [4]. The VGA video signal is decomposed into RGB (Red, Green, Blue) three primary colors and HV (Horizontal and Vertical) line field signals, so the transmission loss is quite small [5]. Although liquid crystal displays can receive digital signals directly, many products use VGA interfaces in order to match the VGA interface graphics cards. There is more and more demand for displaying the process result in real time as the fast development of embedded system, especially as the development of high speed image processing [6–8].

Compared to the regular VGA design, the proposed design does not require central processing unit to control and transmit the data, which can save the expense of hardware and enhance the speed of data processing. It can be widely used in the domain of video display. FPGA is free to design, so we choose to use FPGA to control VGA interface. The controller consists of a synchronizing circuit, a memory module and a palette circuit. The synchronizing circuit can support three different mapping schemes to accelerate the graphic display [9]. According to the Avalon bus control protocol and VGA synchronous timing standards, RGB signals, a horizontal synchronization signal and a vertical synchronizing signal are allocated in a reasonable way. The design is able to not only control display data easily and efficiently, but also be capable of functions extensions.

The remaining of this paper is structured as follows. Section 2 presents the basic components of the designed VGA controller. Section 3 demonstrates the results of function simulation and performance comparisons. Section 4 discusses the whole design of the modules. Finally, Sect. 5 gives some concluding remarks.

2 System Components

The designed VGA controller generates the synchronization signal and outputs the data of pixels to screen continuously. We use Avalon bus protocol to transfer the data controlling color display. This bus allows users to connect the peripherals easily and configure the SOPC easily. We develop a VGA controller and software driver with the display resolution of 640×480 . The hardware part consists of a VGA synchronization circuit, a controller based on SRAM (Static Random Access Memory) dual-port video memory, and a palette circuit. The software section includes the driver for reading and writing video memory, as well as the basic program to get the image from the BMP (Bitmap) format file. Figure 1 depicts the controller and the coupling relation with the colorful displayer via control signals and color signals.

There are three different mapping schemes to generate the pixel units from the memory: bit-mapped scheme, block-mapped scheme, and object-mapped scheme.

With regard to the bit-mapped scheme, the signal *pixel_x* and *pixel_y* can generate the position index for the graphic units. The synchronization circuit constantly updates the screen and writes the relevant data to video memory. A retrieval circuit reads video memory continuously and assigns RGB signal

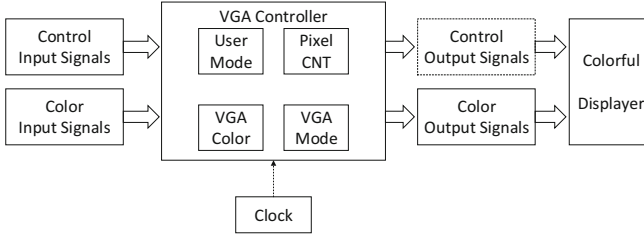


Fig. 1. Controller components and input output signals

data. For a resolution of 640×480 display, there are about 310 kB (640×480) pixel units on a screen to show different pixels [10], which needs 38.4 kB ($640 \times 480 \times 1/8$) space of the memory. The need for 8-bit and 12-bit color display are 310 kB ($640 \times 480 \times 8/8$) and 461 kB ($640 \times 480 \times 12/8$), respectively.

The block-mapped scheme can decrease the demand for the memory performance. In this scheme, a series of data comprise a block and each block is treated as a display unit. For example, an 8×8 pixel square (64 pixels) can be defined as a block. Such a screen, made up of 640×480 pixels, consists of 80×60 blocks and also requires only 4800 words (80×60). The bit width of a word depends on the number of block patterns [11]. For example, if there are 256 block modes, each word should be 8 bits and the block memory size is 4.8 kB ($4800 \times 8/8$), which is called screen memory. Assuming that the VGA uses 8-bit color format, each 8×8 block pattern requires 64 bytes, and all 256 patterns require 16 kB. As a result, the overall memory requirement is about 21 kB, which is much smaller than 310 kB for the bit-mapped scheme.

The object-mapped scheme uses simple object to display patterns. This scheme, combining and implementing the other two schemes together, can generate different parts of a screen. For instance, the bit-mapped scheme can be used to generate the background, and object-mapped scheme is used to display the main objects. We can also use the bit-mapped scheme to provide one part of the screen and the block-mapped scheme to create another part of the screen [12].

The system is mainly composed of a RAM (Random Access Memory) control module, a time synchronization module and a palette module [13]. The interface diagram is demonstrated in Fig. 2. The whole system has no intervene with graphic processing unit, synchronizes the display information and output color data independently.

2.1 The Synchronization Unit

The synchronization circuit (named VGA_sync circuit in this paper) of VGA interface generates timing signals and synchronizing signals. The two signals are decoded from the internal counter, and the two output signals of the counter are *pixel_x* and *pixel_y*. The signal *pixel_x* and *pixel_y* represent the relative position of the scan and actually indicate the location of the current pixel.

The `vga_sync` circuit also produces the `video_on` signal, indicating whether or not to display it.

In the ideal case, the clock rate of the synchronization circuit should be the same as the pixel rate, for a 640×480 VGA display with a pixel rate of about 25 MHz. In this case, the synchronization circuit can be realized by two special counters: the module 800 counter, used for tracking horizontal scanning, and the module 525 counter for tracking vertical scanning.

If the system clock rate is different from the pixel rate, it is usually necessary to create a separate clock domain for the video system. As the design in this paper uses a 50 MHz quartz oscillator on the development board, the system clock rate is twice the pixel rate. Instead of creating a single 25 MHz clock domain, which complicates the timing, it is better to produce a 25 MHz enabling symbol that enables or pauses the count. This symbol can also be sent to the `p_tick` port as an output signal to coordinate the operation of the pixel generation circuit.

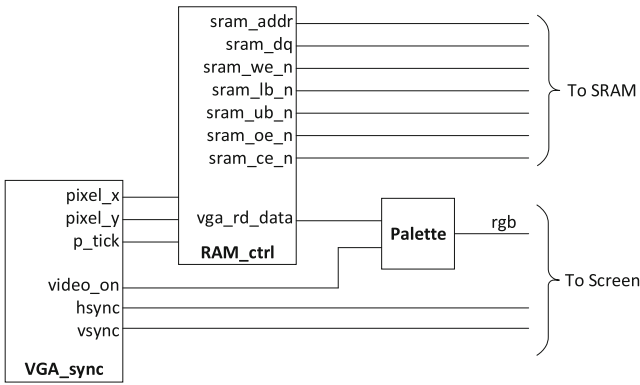


Fig. 2. VGA controller modules interfaces

2.2 The Memory Control Unit

From the bit-mapped scheme point of view, each pixel in the screen is mapped directly to a memory word, and the signal `pixel_x` and `pixel_y` form the pixel address. The system is composed of video RAM memory, and its schematic diagram is shown in Fig. 3. Video memory contains two ports: the VGA port and the host port. The VGA port is marked as `address2` and `data2`, which can be read continuously and beneficial for VGA operation. The pixel address is obtained by the signal `pixel_x` and `pixel_y`, which receive the color information corresponding to the pixel and specify the current coordinate in the screen. The host port, labeled as `address1` and `data1`, connects to the controller. The controller writes information of pixels to the memory and updates the displayed graphics. When it comes to overlapping operations, we may occasionally need to perform reading operations.

The actual performance of video memory depends on the type of physical memory device used in the system. A true dual port memory chip can be used, or a single port memory chip and a multiplexing circuit are used to simulate dual port access. Considering the cost and availability of dual port storage systems, we often adopt the latter methodology.

In a single port implementation, the VGA port and the host port access the same address and data bus of the memory chip. Additional multiplexing and addressing circuits are used to coordinate operations. In order to avoid the graphical display of burr and noise, VGA ports usually have priorities. When video display is closed, the controller can perform the write operation. The other method is double buffering, in which two memory banks are used, one for VGA ports and the other for the controller to write data. Two memory banks operate concurrently and switch roles when a storage body is filled with new data. The scheme actually multiplies the bandwidth of memory, which acts as a real dual port memory. Other similar methods can also be used to simulate dual-port operations. Figure 3 shows a common construction for a RAM controller and emphasizes the interaction with the address bus and the data bus to transfer the original data into the display screen.

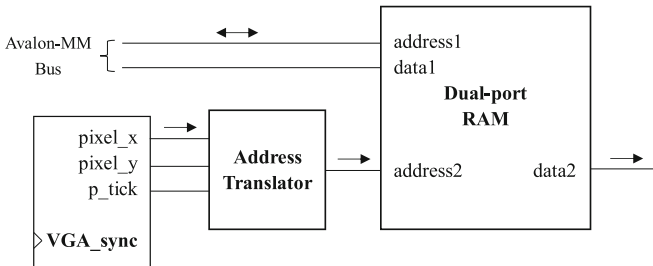


Fig. 3. RAM controller module

2.3 The Palette Unit

The design of VGA controller uses 8 bits for a pixel: 3 for red, 3 for green, and 2 for blue. Assuming the data is $d_7d_6d_5d_4d_3d_2d_1d_0$, the data bits responsible for displaying the red, green and blue are $d_7d_6d_5$, $d_4d_3d_2$ and d_1d_0 respectively. The development board needs 8-bit data bits for each color beam, and three color beams require 24 bits of color depth. We need to convert 8-bit color data of a palette circuit into 24-bit color data output. The easiest way to do this is to use the data bits of the 8-bit color data as the high efficient bit of the 24-bit color output, and copy the low efficient bits. Another more complex solution is to use a color lookup table. In this scheme, the 8-bit input color data is the address of the table which contains 24-bit color data. The 8-bit address corresponds to 256 units in the table, such that the 24-bit colors can catch up to 256. In other words, you can use 24 bits of color in a single image, but we can only use

256 colors in the 4096 possible colors. The size of the lookup table is 256×24 (6144). Many image usage packages contain a palette lookup table that can be invoked according to the actual needs.

3 Function Simulation and Performance Comparisons

The VGA display control system is embedded into the computer in a standardized and effective manner, and the rational circuit conception is realized in the EDA tool. Schematic design input method and hardware description language (HDL) circuit design text is a common design input method in IC design field.

In this design, we use the synchronization circuit, video memory controller, and the module of the palette circuit to realize the RTL (Register Transfer Level) circuit writing of the discrete module [14]. In addition, a package circuit can be added to the video controller to create a SOPC component. It contains three interfaces, one Avalon-MM interface from the device interface functioning as the host port for interacting with the host; a clock input interface for system clock; a conduit interface for the I/O signal of SRAM. The packaged video controller contains additional logics for address decoding, multi-channel data distribution, and appropriate enabling signals.

3.1 Simulation Results

In the Modelsim, the timing sequence of the VGA synchronization signal is simulated in the RTL simulation, and the simulation timing results are shown as Fig. 4. When the row coordinate in the picture above becomes 799, the row coordinate signal will be set to zero and the column coordinate will be 2. At this time, the second row pixel will be scanned. This is a detailed example for the bit-mapped scheme. The length of line synchronization signals sequence is greater than that of the screen opening signals. This phenomenon is consistent with 800 clocks cost by synchronization signal and 640 clocks spent by screen open signal. When the column coordinate becomes 524, the column coordinate is set to zero and the screen completes a refresh process.

The bit-mapped scheme requires all the pixels to display based on the clock edge, so the color and coordinate data are incoherent and independent to display every display datum in a single time period, compared to the other two schemes. Otherwise, the block-mapped scheme allows blocks of display data to transfer continuously without the restrain of the clock edge and contributes to coherent display mode with continuous data output. Furthermore, the object-mapped scheme mixes the benefits of the first two schemes as it supports the high-definition pixel display with the bit-mapped scheme and the high-efficiency data transfer with the block-mapped scheme. But this allocation method for display is quite complex of all three display schemes and may result in display irregularity sometimes.

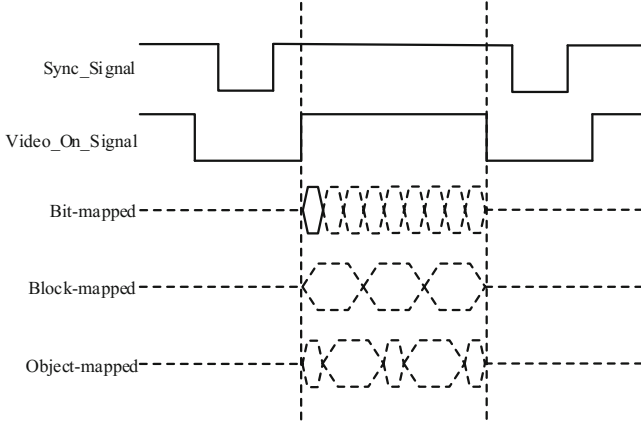


Fig. 4. Synchronization sequence diagram

3.2 Performance and Comparison

The design for three innovated display schemes is independent and reliable for specific applied environment. Compared to primitive VGA controller, the proposed design presents more display mapping modes to appeal to different implementation environment. The VGA controller can be employed in several systems which have video output and support variable mapping schemes to accelerate the display speed compared with the traditional VGA display system [15]. In this section, we propose a FPGA-based system which uses VGA controller as a functional module to display visual data in three mapping display mode. This system plays a role as a remote surveillance system.

The novel VGA controller is not in need of central processing unit to transfer the data for display, which may spare some cost for hardware and increase the speed of data processing. This efficient design can be extensively implemented in the domain of video display without looking up the instant VGA information, while other VGA controllers tend to control the color signals in a monotonic way without the use of independent VGA submodules [5]. FPGA is compatible to various designs and we choose to utilize FPGA to control VGA interface. The design can not only control display data easily and efficiently, but also be capable of functions extensions.

We realize the synthesis of the function modules presented below with Quartus II and observe the overall performance. Moreover, we compare the three mapping schemes with register amounts after compilation, signal amounts and signal activity, as shown in Table 1. The signal amounts affect the complication and the power consumption of the whole system. The block-mapped scheme has two more signals than bit-mapped scheme, including block starts signal and block ends signal. Besides, object-mapped scheme implements bit starts signal and bit ends signal more than block-mapped scheme. The signal activity here indicates the average signal reversal amounts in one second time at 50MHz

clock frequency and the higher signal activity results in greater dynamic power consumption for transistors. The increase in signal amounts can result in more register amounts and synthesized areas.

Table 1. Differences between three display schemes

Scheme	Register amounts	Signal amounts	Signal activity ($\times 10^7$)
Bit-mapped	2348	10	5.00
Block-mapped	2412	12	1.67
Object-mapped	2560	14	3.33
Traditional mapping	2072	10	5.67

4 Discussion

In this paper, we have presented efficient hardware architecture for VGA monitor controller which has a high potential to be used in Altera FPGA-based systems. The highlighted feature makes the design suitable for several FPGA devices and is able to meet different requirements of targeted applications. Using FPGA to control the VGA interface has the advantages of hardware integration and overcoming the disadvantages of large size, difficulty in modifying the architecture, and poor system compatibility. The image information is stored by RAM, which is easy to write RGB trichromatic information, and the image content is displayed according to FPGA. The real-time control of video display based on VGA protocol is realized, which is of great practical significance to the realization of various miniaturized devices and portable embedded systems.

The whole video controller system is composed of a synchronous circuit, a video memory controller circuit, and a color palette circuit. The three basic components can deal with the normal transactions while displaying videos. The resources cost is decreased to a low level and the design achieves low energy use, which relies on the specific design based on SOPC method. The transistors are generated by the experimental platform and cater to many kinds of requirements. This paper presents a practical method to transfer the different color data and synchronize the display lines and rows. However, it fails to be compatible with all FPGA development boards. The compatibility for this design will be under evaluation. Moreover, it may increase the difficulty on setting up the system and selecting the appropriate pixel-mapped schemes. The raise of signal activities causes higher consuming power. And the expansion of synthesized areas may result in more resource utilization. Our experiment results show that the proposed method generates display functions with good subjective quality in terms of the objects' spatial consistency and temporal motion continuity. It also avoids overloaded components and compensates for insufficient available pixels.

5 Conclusion

In this design, the VGA controller based on FPGA development board could be employed to display colorful pixels efficiently. The VGA controller based on SOPC can send data to the screen directly through the used interface. Three pixel-mapped schemes were implemented to accelerate the display rate of the videos. The feasibility of system design, the overall architecture of the hardware description language code, and the development tools affected the performance of the control system. From the perspective of the wide application of VGA interface in flat panel display field, the popularity of this flexible design method combined with FPGA has surged. Our simulation results prove that this system is stable, flexible, short in design period, low in expense, and has certain external expansibility, which clearly suggests the display system with VGA interface opens a promising avenue towards the display market prospect.

References

1. Tormo, D., Monmasson, E., Idkhajine, L., Blasco-Gimenez, R.: Embedded real-time simulator implementations of electromechanical systems using system-on-chip devices. In: *Electrimacs* (2017). <https://hal.archives-ouvertes.fr/hal-01799852>
2. Perez, K.G., Yang, X., Scott-Hayward, S., Sezer, S.: Feature study on a programmable network traffic classifier. In: *IEEE International System-On-Chip Conference*, pp. 108–113 (2017). <https://doi.org/10.1109/socc.2016.7905446>
3. Luo, Z.J., Zhang, W.N., Liu, L.W., Xie, S.T., Zhou, G.F.: Portable multi-gray scale video playing scheme for high-performance electrowetting displays. *J. Soc. Inf. Disp.* **24**(6), 345–354 (2016). <https://doi.org/10.1002/jsid.444>
4. Pfeifle, F.: Real-time signal processing on field programmable gate array hardware. In: Bader, R. (ed.) *Springer Handbook of Systematic Musicology*. SH, pp. 385–417. Springer, Heidelberg (2018). https://doi.org/10.1007/978-3-662-55004-5_20
5. Zhang, Y., Lu, K., Gao, Y.: Fast image matching algorithm based on affine invariants. *J. Cent. South Univ.* **21**(5), 1907–1918 (2014). <https://doi.org/10.1007/s11771-014-2137-7>
6. Zhang, H.B., Pan, J.C., Zeng, D.J.: Video graphics array interface switch apparatus. US, US8321621B2 (2012)
7. Hu, H.: Testing system and method for video graphics array port, US, US20140244195 (2014)
8. Gurevich, K.L.: Apparatus and methods for video graphics array (VGA) virtualization in system exploiting multiple operating systems, US, US20060036775 (2006)
9. Zhao, T., Li, T., Han, B., Sun, Z., Huang, J.: Design and implementation of software defined hardware counters for SDN. *Comput. Netw.* **102**, 129–144 (2016). <https://doi.org/10.1016/j.comnet.2016.03.004>
10. Tissot, J.L., Tinnes, S., Durand, A., Minassian, C., Robert, P., Vilain, M., Yon, J.: High-performance uncooled amorphous silicon video graphics array and extended graphics array infrared focal plane arrays with 17- μm pixel pitch. *Opt. Eng.* **50**(60), 409–421 (2011). <https://doi.org/10.1117/1.3572155>
11. Endoh, T., Tsutomu, S., Yamazaki, T., et al.: Uncooled infrared detector with 12 μm pixel pitch video graphics array. In: *Infrared Technology and Applications XXXIX International Society for Optics and Photonics* (2013). <https://doi.org/10.1117/12.2013690>

12. Waldvogel, B., Schulz, H., Behnke, S.: Dense real-time mapping of object-class semantics from RGB-D video. *J. Real-Time Image Process.* **10**(4), 599–609 (2015)
13. Kim, I.H., Choi, J.S., Yun, J.L., Nam, J.Y., Ha, Y.H.: Design and implementation of multimedia functional module for digital TV. *IEEE Trans. Consum. Electron.* **50**(3), 962–967 (2004). <https://doi.org/10.1109/tce.2004.1341707>
14. Sivasathya, S.: Design of VGA monitor controller in FPGA using on chip embedded array RAM. *Int. J. Technol. Eng. Syst.* **6**(1), 27–31 (2004). <https://www.ijcns.com>
15. Bharathi, M., Yogananth, A.: Design of VGA monitor control using Altera FPGA based system. *Int. J. VLSI Embed. Syst.* **5**, 866–890 (2014). <http://ijves.com>