Proceedings of the 2nd International Conference on Functional Materials and Civil Engineering
DOI: 10.54254/2755-2721/72/20241022

FPGA-based Sokuban-like fun logic learning program with combinatorial logic blocks

Xinyuan Wang

College of Integrated Circuit Science and Engineering, Nanjing University of posts and telecommunications, Nanjing, China

b21021606@njupt.edu.cn

Abstract. There are problems such as a high threshold of entry, difficulty understanding the architecture, and boring learning in electronics applications. In order to improve the readability of the structure and learning efficiency, the study proposes a Sokuban-like fun logic learning program based on a Field Programmable Gate Array (FPGA) with combinational logic blocks displayed by VGA. Based on the logic of Sokuban, the program adds "subject", "predicate" and "object" logic blocks, so that the attributes of each object will change with the different combinations of logic blocks. This allows the properties of each object to change with the different combinations of logic blocks, ultimately realizing the process of achieving the same purpose in different ways. The study visualizes the process of writing code by using a specific code framework to transform syntax and entities into different logic blocks that can be changed in different locations. The design solves the problem of the learning process of logical thinking in electronics which is too abstract leading to difficulty in understanding in a relatively easier and more interesting way and can be used as a new way to represent dynamic changes in circuits. In the future, it will not only assist in learning, but can also be used as another new way to represent the output of circuits similar to schematic diagrams, Verilog and other codes, providing a new way of thinking about electronic design.

Keywords: Combinatorial logic blocks, VGA display, Visualisation.

1. Introduction

The effective and rapid dissemination and application of electronic knowledge is one of the most important needs in the society now. With the development of chip technology and smart technology applications, opportunities to use electronic products will only become more and more, accompanied by the emergence of more and more jobs related to electronic technology. FPGAs have been emphasized as the basis for electronic development, but it is still not enough to meet the demand. Therefore, developing a kind program and methodology that makes it easier to learn has become an important need nowadays.

Nowadays there is a very large number of learning programs for learning purposes all over the world, and the market has proved its possibilities. It shows that, based on electronic development, it is possible to combine games with learning.

Electronic engineering documents for most people are a large string of abstract data, which makes the promotion of the application of technology is still difficult. Only a schematic can be a relatively

DOI: 10.54254/2755-2721/72/20241022

intuitive representation of its structure. There are very few visualization aids in the zone, and only modular programming environments such as Scratch exist, which merely simplify the process and do not change it. Therefore, the development and promotion of fun logic learning programs are meaningful.

In this study, an FPGA-based Sokuban-like fun logical learning program has been developed [1]. It is written in Verilog HDL and uses a top-down design methodology to analyze the necessary sub-modules that are to make up the top-level module, and then further decompose and design the individual modules until it reaches the underlying functional blocks that cannot be further decomposed. The program is expected to achieve the goal of visualizing the process of compiling the program and the output, and expanding the design ideas, under the condition of ensuring a basically smooth and error-free operation.

2. Relevant Technologies and Tools

In this study, Visual Studio Code is used as the development environment and Quartus and ModelSim are used as synthesizing tools.

This design uses Visual Studio Code to edit the code after completing the installation of the Verilog HDL plug-in. The code contains the framework to build the function of VGA to complete the visualization goal, and on this basis, the library included in Quartus provides the IP core of a phase-locked loop to complete the module design, and finally, ModelSim is used to perform a comprehensive verification of the whole program to avoid problems such as syntax and structure.

3. Demand Analysis

3.1. System target

The system utilizes the logic of Sokuban. System functions include map, movement, collision detection and mission completion condition detection.

3.2. User analysis

The system is aimed at 3 main groups of users. The program can be used by students of all ages in their free time, for making the most of their time to improve their electronic skills and logical skills. College students and other students in professional fields can also use the program's ideas to enhance their understanding and application of relevant designs and so do electronic engineers and FPGA enthusiasts in their free time.

3.3. System structure analysis

The program is divided into several modules, including the PLL module, main control module, VGA display module, key signal input module, operation logic module, and data storage module.

The PLL module is responsible for accomplishing the conversion of the clock frequency to the target frequency [2].

The main control module is the core of the whole program and is responsible for managing the overall logic of the program, state machine control, handling input and output, and other functions [3,4]. It will interact with other modules and control the program process according to the program state.

The VGA display module [5,6] is responsible for displaying the individual elements of the program on the screen. It needs to generate the display content based on the internal storage state information data and output it to the display device.

The key signal input module is responsible for handling key operations. It will pass the key signal input to other modules after stabilization.

The operation logic module is responsible for implementing the core logic of the program, including the movement of objects, interaction rules, processing input and output data, task completion conditions, and IO signal processing. It will update the state and output according to internal data and external input signals, and determine whether the purpose set by the program is completed or not. The data storage

DOI: 10.54254/2755-2721/72/20241022

module is responsible for storing the program's state, data, and other information. It can use memory or registers to store these data and provide read/write interfaces for other modules to access.

The reset function is embedded within each module. The system structure is shown in Figure 1.

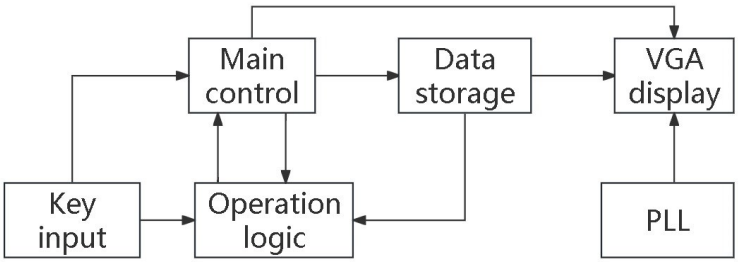

Figure 1. System structure.

4. System Design

4.1. System function module design

4.1.1. PLL module. The PLL module is built using Quartus' altera library functions and transforms the input clock frequency to the desired frequency. A 50Mhz source was used in this study and a 40Mhz source was required for the mode selected for the VGA, then the PLL IP core was generated based on the data as above.

4.1.2. Main control module. This module is a finite state machine, which controls the whole process of the program through the signals fed back from each module, changes the state of the program, and is also responsible for calling and controlling other modules. The program is divided into three states: Idle, Running and Over. The program enters the Running state to run the program after receiving the reset signal at the beginning, and enters the Over state and stays in the Over state to wait for the reset signal to restart the program after receiving the target completion signal endGame. The state transfer diagram is shown in Figure 2.

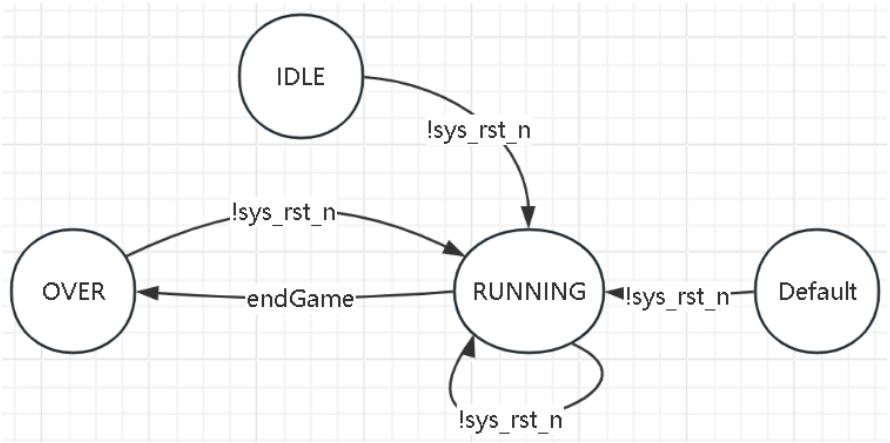

Figure 2. Main control state transfer diagram.

4.1.3. VGA display module. The module is divided into two modules: vga_crtl, the VGA sequence control module, and vga_pic, the VGA image data generation module [7-10]. Vga_crtl is the VGA sequence control module, which serves to drive the VGA monitor and display the graphic pixel information from the input module in accordance with the VGA sequence. The mode called here in this study is 800×600@60, and its parameters are shown in Table 1 and Table 2.

DOI: 10.54254/2755-2721/72/20241022

Table 1. VGA display mode and parameters of the horizon.

Display mode	CLK(Mhz)	SYNC (pix)	BACK	LEFT	VALID	RIGHT	FRONT	TOTAL
640×480@60	25.175	96	40	8	640	8	8	800
640×480@75	31.5	64	120	0	640	0	16	840
800×600@60	40.0	128	88	0	800	0	40	1056
800×600@75	49.5	80	160	0	800	0	16	1056
1024×768@60	65	136	160	0	1024	0	24	1344
1024×768@75	78.8	176	176	0	1024	0	16	1312
1280×1024@60	108.0	112	248	0	1280	0	48	1688

Table 2. Display mode and parameters of the field.

Display mode	CLK(Mhz)	SYNC (pix)	BACK	TOP	VALID	BOTTOM	FRONT	TOTAL
640×480@60	25.175	2	25	8	480	8	2	525
640×480@75	31.5	3	16	0	480	0	1	500
800×600@60	40.0	4	23	0	600	0	1	628
800×600@75	49.5	3	21	0	600	0	1	625
1024×768@60	65	6	29	0	768	0	3	806
1024×768@75	78.8	3	28	0	768	0	1	800
1280×1024@60	108.0	3	38	0	1024	0	1	1056

The block diagram of the module is shown in Figure 3.

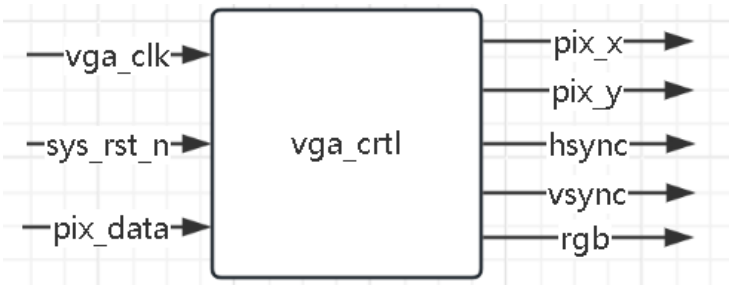

Figure 3. Block diagram of VGA sequence control.

The output signal function is described in Table 3.

Table 3. Display mode and parameters of field.

Signal	Bit Width	Type	Function Description
Vga_clk	1Bit	Input	CLK 40Mhz
Sys_rst_n	1Bit	Input	Reset signal, active low
Pix_data	16Bit	Input	Image pixel color information
Pix_x	10Bit	Output	X-axis pixel point coordinates in valid display area
Pix_y	10Bit	Output	Y-axis pixel point coordinates in valid display area
hsync	1Bit	Output	Horizon synchronization signal
vsync	1Bit	Output	Field synchronization signal
rgb	16Bit	Output	RGB information of pixel

DOI: 10.54254/2755-2721/72/20241022

Vga_pic is a VGA image data generation module that serves to input information from the data storage module into the module's ROM to generate object pixel data and pass it to vga_crtl for display.

The block diagram of the model is shown in Figure 4.

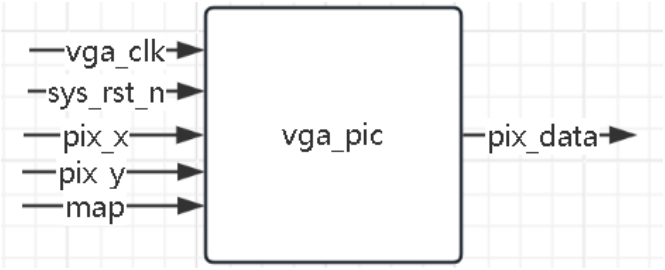

Figure 4. Block diagram of VGA image data generation module.

4.1.4. Key signal input module. Through the key input, part of the signal is stablized and input to other modules. The stabilization possess delays the input signal, and only outputs the signal to other modules if the signal remains unchanged for a certain period of time.

4.1.5. Operation logic module. The operation logic module is divided into the object movement module, the rule interaction module and the IO module.

The object movement module carries out register data changes through the key signal input module, while each signal carries out a collision detection to trigger the corresponding behaviors, such as object coordinate changes, movement being blocked, data input and output, logic blocks forming valid logic statements and calling the rule in rule interaction module.

The Rule interaction module is responsible for outputting a signal representing whether the rule is valid or not. When the valid signal is detected, it means that the rule is valid, which generally occurs at the same time as each collision detection.

The IO module is processed by defining specific objects as IO ports, detecting the state of the object through the rule interaction module, and outputting it as a high or low level.

4.1.6. Data storage module. The module is responsible for storing the map data information. The map data includes a gridded map frame and parameterized objects, such as a green block set to 1 and, red block set to 2, while the object parameters will be passed to the VGA image data generation module to complete the processing of the image.

4.2. System conclusion

The system implements an FPGA-based Sokuban-like fun logic learning program with combinational logic blocks. The program conveys key signals to modules after stablization process, and controls the program state with a finite state machine under the control of the clock. In the running state, it first undergoes a series of initialization, and then reads the gridded map data in the data storage module, passes the data to the VGA display module, and completes the process of displaying image by using the data as the address of the location and the rgb information that ROM containing. The movement of some objects can be operated by keys in order to interact with other objects, such as pushing and overlapping. Before the process of each operation, a position detection is carried out to realize the detection of collision and rules interaction. The rules are realized by detecting whether the "subject", "predicate" and "object" are arranged next to each other according to certain sequence.

Finally, the study implements a simple and intuitive fun program which visualizes the internal structure of the code and the output results. It is like Verilog or schematic diagrams, through a different way to achieve the compilation of the program. So that it may assist in carrying out the application and promotion of electronic technology.

DOI: 10.54254/2755-2721/72/20241022

5. Conclusion

In order to solve the problem of difficulties in the application and promotion of electronic technology, aiming to reduce the threshold of application and visualize the operation process, the study based on a top-down approach designs a solution, a FPGA-based Sokuban-like fun logic learning program with combinational logic blocks. It describes the overall system framework and the design approach of each module in turn. The study is the result of upgrading Sokuban after expanding it. The program realizes the change of signals through the collision detection triggered by the change of the position of the object, which in turn indicates the circuit that realizes a specific function. The significance of the completion of this design is to complete a visual, intuitive, and simple framework. A variety of visual combinational logic and sequential logic methods are used through the FPGA program to complete the compilation of the target and reduce the difficulty of the application. So, it may provide a new way of thinking for the future of electronic design. However, it is worth noting that the current program still has many limitations, such as the space utilization problem. For this reason, more kinds of logic blocks and more structural designs are needed in the future to achieve the goal of further facilitating circuit implementation. Therefore, the design methodology described in this paper is of great significance and practical value in the application and promotion of electronic technology.

References

[1] Xu H and Lin F 2019 Design and Realization of FPGA-based Sokuban Game on Practical Electronics vol 19 p 24-26
[2] Yang C 2023 Design and Simulation Study of an All-Digital Phase-Locked Loop on Shanxi Electronic Technology vol 2 p 74-76
[3] Liu H and Guan X 2018 A Study of Verilog HDL Description Methods for Hierarchical State Machines and Coprocessing State Machines on Journal of Heilongjiang Institute of Technology vol 32 no 1 pp34-40+45
[4] Luo X Li J and Tian Zh 2012 Optimized Design of Verilog HDL-Based Finite State Machine on Electronics Quility no 3 p36-38+42
[5] Li Y Lv Zha and Chen S 2019 Design and Realization of FPGA-based VGA Image Display and Control System Journal of Huainan Normal University vol 21 no 2 p112-115
[6] V Sklyarov 2002 Reconfigurable Models of Finite State Machines and their Implementation in FPGAs on Journal of Systems Architecture vol 47 Issues 14–15 p 1043-1064
[7] Caffee Sugar 2023 FPGA-VGA Monitor Displaying Colour Bar(Code enclosed) on blog.csdn.net/m0_72885897/article/details/129900357
[8] Gazi O Arlı A Ç 2021 Video Graphic Array (VGA) and HDMI Interfacing In State Machines using VHDL Springer Cham
[9] Carlos A R A 2012 FPGA Open Architecture Design for a VGA Driver on Procedia Technology vol 3 p 324-333
[10] Gui H 2019 Design of FPGA-Based VGA Image Display on Computer Programming Skills Maintenance vol 8 p 136-138