Logic and Computer Design
Project Report

Whack-A-Mole
打 地 鼠

Group Member:

李海鹏

费昊天

陆嘉铖

Jan 5th, 2017

Content

Background

Xilinx Spartan-3 experimental platform provides sufficient buttons, toggle switches, LED display modules, buzzer and VGA interface for us to input and output. After a semester of experimental study and self-study after school of VGA and other related knowledge, we have a certain understanding of how to use these interfaces, and we also hope to be able to operate a reasonable completion of our own design. In this design project, out of our team's love on games, we decided to design a "whack the mole" game with the experimental platform to output a VGA form interactive interface, Spartan-3 experimental board button to input operation, Digital tube to output scores, buzzer LED to output feedback on whether hit the hamster or not.

Design Specification

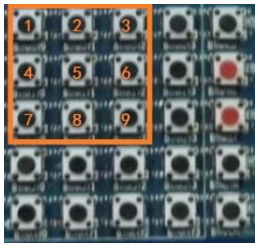
Environment

Hardware: Sword Kintex7

IDE：Xilinx ISE 14.7
Language：Verilog HDL

Input and output

- Clock
- Buzzer
- Button (1 to 9) corresponds to the nine mole holes from left to right, from up to down



- 8 digit 7-Segment LED

- Three color LED light

- VGA monitor

Key module design

- The initial state of this system
  - Map[17:0]=0 (All the hole is empty)
  - Time1[17:0]=0 (No time counter for mole disappearance)
  - Time2[17:0]=0 (No time counter for visual and sound effects)
- Moles generator

- Use LFSR to implements a random number generator
- According to the generated random number, we set register variable Time1[i+1:i] to “11”, which means the counter of disappearance of a mole with decrements by 1 every second.

- Key_input
  - To detect the behavior of buttons
  - If the hole corresponding to the button has a mole (namely the “Time1[i+1:i]” variable is not equal to “00”), we reset Time1[i+1:i] to “00”, set Time2[i+1:i] to “11”, which means the counter of the sound effect and visual effects of “hitting a mouse”, and set Map[i+1:i] to “10”, which means “right hit”.

  - If the hole corresponding to the button has no mole (namely the “Time1[i+1:i]” variable is equal to “00”), we reset Time1[i+1:i] to “00”, set Time2[i+1:i] to “11”, which means the counter of the sound effect and visual effects of “hitting an empty hole”, and set Map[i+1:i] to “11”, which means “wrong hit”.

- Clock cycle
  - In every clock cycle, we refresh the state of each hole (namely we change the key of register variable of “Map[17:0]” according to the keys of variable Time1[17:0] and Time2[17:0])
    - If both Time1[i+1:i] and Time2[i+1:i] equal to zero, the hole should be empty and the Map[i+1:i] should be “00”.
    - Else, if Time1[i+1:i] is not zero while Time2[i+1:i] equal zero, the hole should contains a mole and Map[i+1:i] should be “01”.
    - And if a button is pressed, do the actions as described as above (set Map[i+1:i] to “10” or “11”).
- Give visual and sound effects (feedback)
  - Buzzer
    - Activated (“buzzer=0”) when hit a right mole (Map[i+1:i]= “10”).
  - Three-color LED light
    - Gives off Green light (“LEDG0=0, LEDR0=1, LEDB0=1”) when hit a right mole (Map[i+1:i]= “10”).
    - Gives off Red light (“LEDG0=1, LEDR0=0, LEDB0=1”) when hit an empty hole (Map[i+1:i]= “11”).

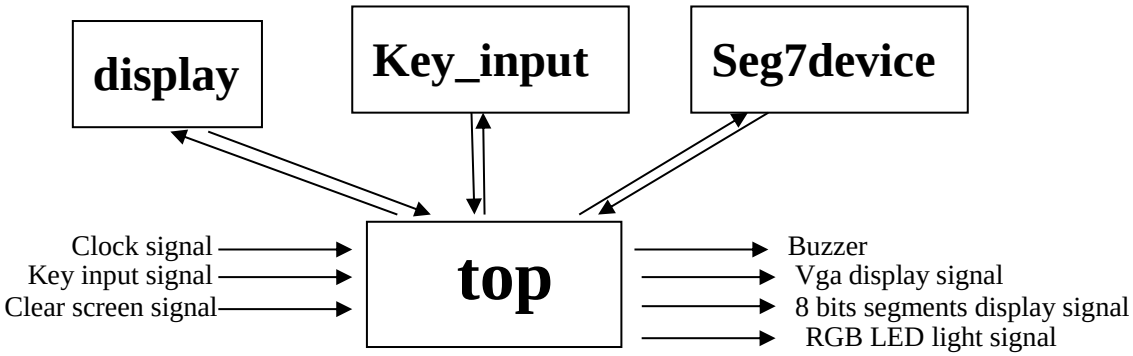
  - 8 digit 7-Segment LED number displayer
    - “Score” increments by 1 when hit a right mole
    - Convert “Score” to BCD code
    - Display register variable “Score” on it.

  - VGA monitor
    - Change the picture displayed according to the state of “Map” variable
      - Empty hole (Map[i+1:i]= “00”), display empty hole
      - Hole with Mole (Map[i+1:i]= “01”), display a hole with mole in it
      - Whack a mole (Map[i+1:i]= “10”), display a mole with a hammer above it
      - Hit an empty hole (Map[i+1:i]= “11”), display an empty hole with a hammer

above it.

Modules Structure

Relation between modules



Top

INPUT:

clk,

clear_screen, Switch to turn on or off the screen

[3:0] KeyX, button array on SWORD

[4:0] KeyY, button array on SWORD

OUTPUT:

buzzer,

SEGLED_CLK, 8 digit 7segment LED

SEGLED_CLR, 8 digit 7segment LED

SEGLED_DO, 8 digit 7segment LED

SEGLED_PEN, 8 digit 7segment LED

LEDR0,triLED Red color (Left one)

LEDG0,triLED Green color(Left one)

LEDB0,triLED Blue color(Left one)

LEDR1,triLED Red color(Left one)

LEDG1,triLED Green color(Left one)

LEDB1,triLED Blue color(Left one)

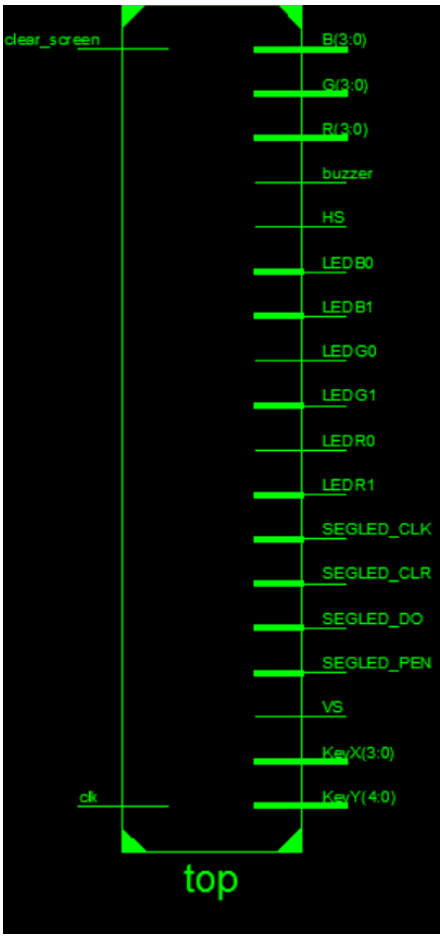
wire HS,horizontal scanner

wire VS,vertical scanner

wire [3:0] R,red colod number

wire [3:0] G,green color number

wire [3:0] Bblue color number



Behavior:

Generate a mole in a random hole

Call Key_input module to detect button

Do proper work(right hit, wrong hit, miss)

Call display and Seg7Device modules to show the result

Key_input—input the key operation

INPUT：

Clk：Clock pulse signal

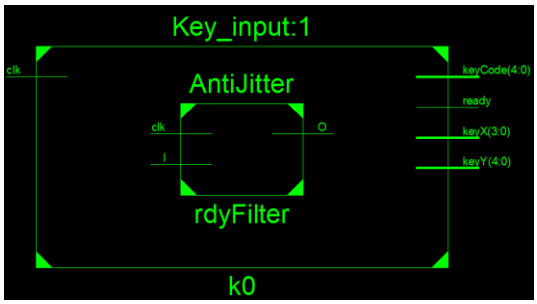
KeyX[3:0]: The X coordinate of the key entry

keyY[4:0]: The Y coordinate of the key entry

OUTPUT：

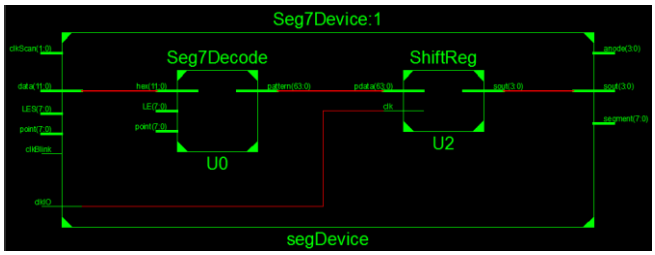
keyCode [4:0]: Output the key code.（[4:2]: The Y coordinate [1:0]: The X coordinate）

ready: Present Whether there is a key input.



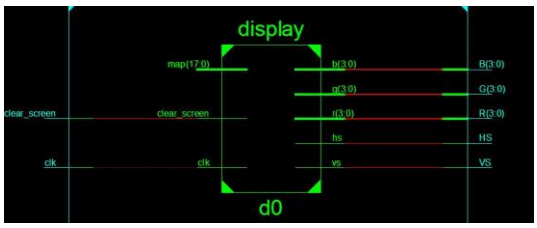
segDevice—output the game score

INPUT:
ClkIO: Clock pulse signal
ClkScan[1:0]: Clock pulse signal
clkBlink: Clock pulse signal
data[11:0]: Game score
point[7:0]: Unnecessary
LES[7:0]: LED
OUTPUT:
sout[3:0]: Output the score
segment[7:0]: Unnecessary
anode[3:0]: Unnecessary



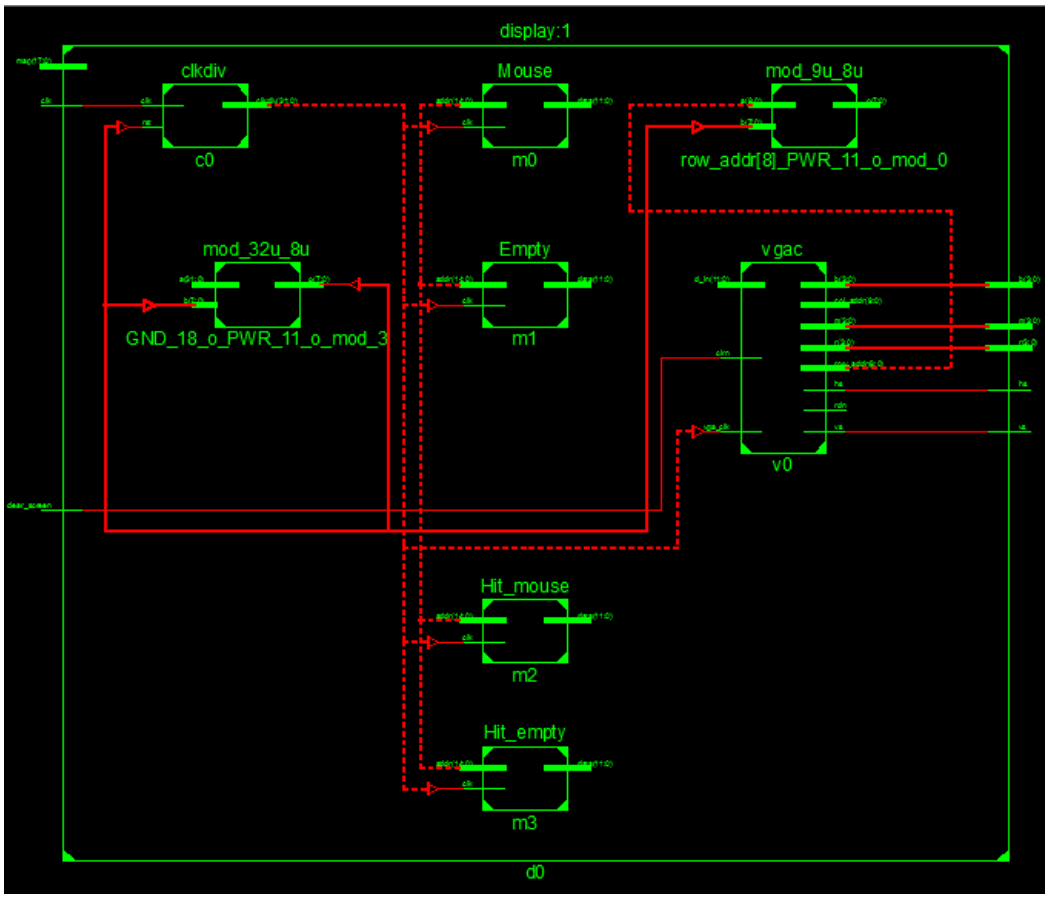
Display module ---- generate vga display signal

INPUT:
clear_screen: control signal to switch off or switch on the vga display
clk: 25M Hz clock signal for vga display
map[17:0]: denotes the current status of nine parts(two bits control a part)
OUTPUT:
R[3:0]: 4 bits red value for a RGB pixel
G[3:0]: 4 bits green value for a RGB pixel
B[3:0]: 4 bits blue value for a RGB pixel
HS: horizontal synchronization for vga display
VS: Vertical synchronization for vga display.

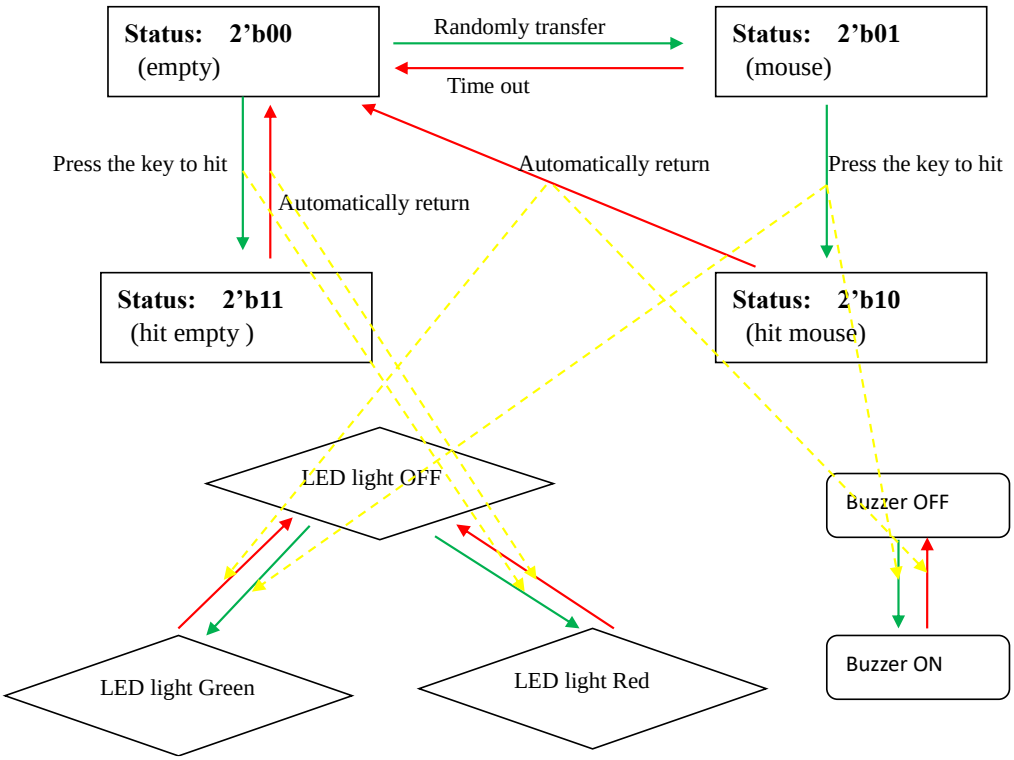


- There are eight main blocks inside display module. For vga display we use only 480*480px in the screen and divide it into 9 parts (each part is a hole where a mole may appear randomly). These 9 parts are separately controlled. Each part has a two-bit register to denotes its states which is passed into display module through map[17:0].
  - clkdiv generates a 25MHz clock signal which is suited for vga display.
  - vgac will implement horizontal synchronization and vertical synchronization. It receives a 12 bits data signal every clock period and changes the RGB value of the pixel being scanned at that clock period and output the address of this pixel.(the design of this module refers to the vgac in VGAdemo)
  - mod_9u_8u receives the address of a pixel to calculate which part currently being scanned by vga and also the offset address in this part.
  - mod_32u_8u will choose a ROM among Mouse, Empty, Hit_mouse and Hit_empty according to the current states and use offset address to fetch the RGB value of the current pixel and pass is to vgac.
- Mouse, Empty, Hit_mouse and Hit_empty are four virtual ROM created by IP core and each

contains the information of a 160*160px picture.

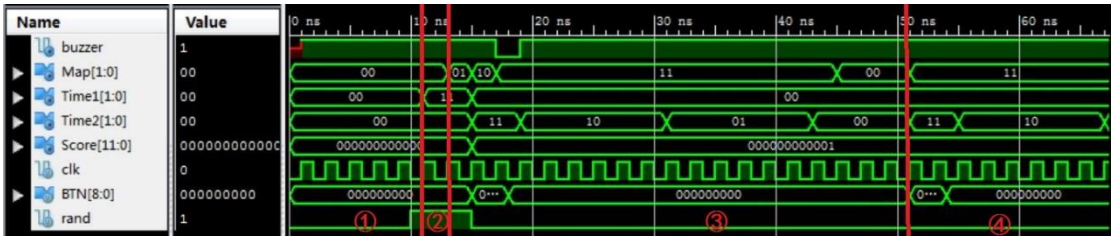


Program Procedure



Note: here we only use one to illustrate because we have same nine parts in the program

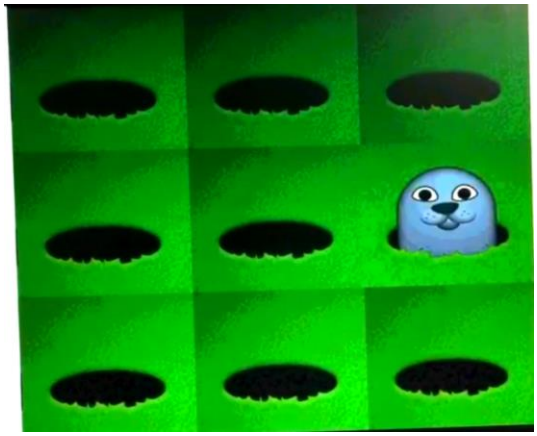
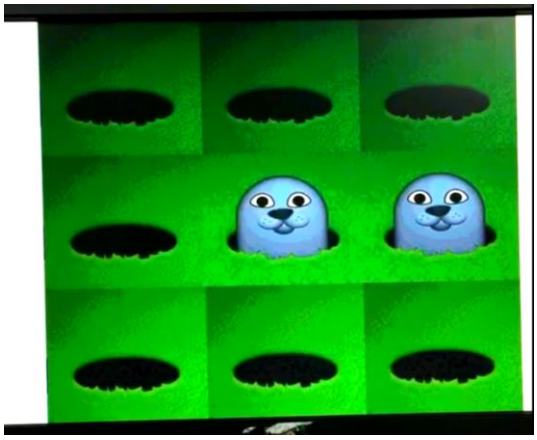
Key Module Simulation



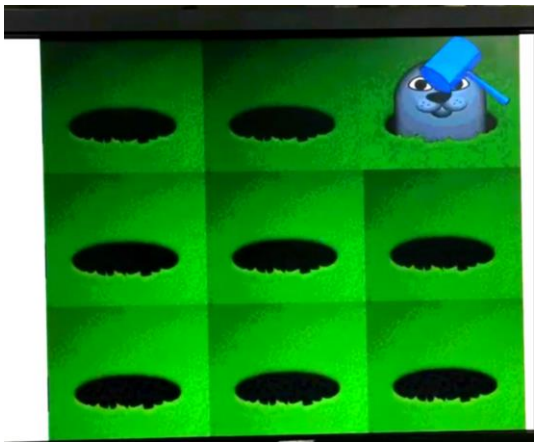
- For simulation, we only need to test the behavior of one particular hole (so I test the hole i==0)
- State specification:
  - State ① is "Empty hole", with Map[i+1:i] equal to "00", Time1[i+1:i] equal to "00" and Time2[i+1:i] equal to "00"
  - State ② is "Hole with a mole", with Map[i+1:i] equal to "01", Time1[i+1:i] is larger than "0" and Time2[i+1:i] equal to "00"
  - State ③ is "right hit", with Map[i+1:i] equal to "10", Time1[i+1:i] equal to "00" and Time2[i+1:i] is large than "0"
  - State ④ is "wrong hit", with Map[i+1:i] equal to "11", Time1[i+1:i] equal to "00" and Time2[i+1:i] is large than "0"
- The four state work fine.

Debug and analysis

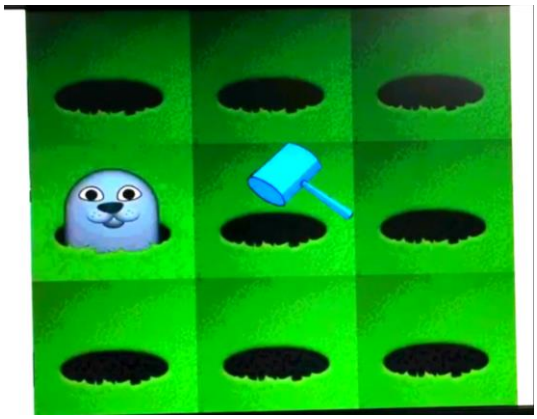
- Mole generating
  - The moles can be generated randomly and generating speed is not uniformed.
  - If it is not hit by player in 2 seconds, it disappear automatically.
  - Works fine.

- “right hit” action
  - For a hole with mole inside it, the picture on screen displayed correctly.
  - The Score on digit displayer plus one
  - The buzzer alerts
  - The LED gives off green light




- “empty hit” action
  - For a hole without mole inside it, the picture on screen displayed correctly.
  - The Score on digit displayer does not plus one
  - The buzzer does not alert.
  - The LED gives off red light



What we learned from this project

By Li Haipeng

- Have deep understanding of two concepts: "behavior description" and "gate-level description".
  - In the design of many previous experiment modules we use "gate-level description", both in schematic design and Verilog code design. However, in this project, most part is designed using "behavior description".
    - Their relation is like the difference between different kinds of programming language. For example, MATLAB and Assembly language. A function call in MATLAB can have the same effect as thousands of lines in Assembly language.
    - Though higher level languages or descriptions are more convenient to use for programmers, it is not what we want to learn from this course.
- The solution of BUGs
  - In the project, especially in top module, which involves the main logic relation of the system. BUGs appeared quite often. The syntax checking can find most of them.
  - But when having BUGs on logic layers, it is hard to find them since we cannot debug on ISE 14.7 like what we can do on other c++ IDEs such as Visual Studio.
  - One substituted method is to change my code a little (sometimes a lot) to conduct a simulation and find out where goes wrong. It is time saving since we can waste a lot of time in synthesizing, translating, mapping and so on.

By Lu Jiacheng

- There project helps me a lot in learning the design of logical circuits, especially the concept of states. The core of our project is about state transferring and display specific content according to current state. Actually when we first had

the idea of design a program of Whack-a-mole, we had thought about several methods. However, we believe using states is the easiest and most clear way and so it does. So I strongly realized the importance of states in computer design.

- Besides, I picked up pieces of understanding about VGA display. It scanned the screen row by row, pixel by pixel. I also get more familiar with the mechanism of ROM after build virtual ROMs on the FPGA platform use IP core.

By Fei Haotian

- When designed the project, my part was to control the output to present the feedback we want to transform to the player and input the player's operation. When designed these modules, I met a lot of problems.
  - The first is to understand how to let the device on the board work as you expected. As every module has its own display rules, I need to figure them out clearly and guarantee not to apply them on an inappropriate one. When I dealt with this problem, I chose to analyze the VGA Demo first to understand the work principles. Then, to confirm some of my hypothesis, I wrote some test modules to test the function. Finally, I wrote the real modules to complete the functions the project needs.
  - The second problem I met is to fit my modules in the whole project. I need to know which form of the data that the programs in the project will give me and what I need to provide for other modules. This step involves interaction with other programmer in my group and understands their expectation for my modules. This one teaches me a lot.
- During designing this project with my partners, I learnt a lot of things from the problems I met and my understanding of the logic circuit is developing rapidly.

Duty Division

李海鹏(40%)：Top module, report, video taking.
费昊天(30%)：Key input, LED light, LED digit display module, report, video composing.
陆嘉铖(30%)：VGA display module, report, video taking.

Reference

1. 《Logic and Computer Design Fundamental》, Fourth Edition, Pearson Education, Inc.
2. Wikipedia, The Free Encyclopedia. 12 Dec. 2016.< https://en.wikipedia.org/wiki/Linear-feedback_shift_register>
3. 《Verilog 数字系统设计教程》 夏宇闻，北京航空航天大学出版社
4. 施青松、董亚波编者．逻辑与计算机设计基础实验教程，杭州浙江大学出版社, 2007