1D Project Group Report

Electronic Game Prototype

Team Number: 13-3

Table of Contents

Introduction

Simon’s Windows is based on the popular game Simon Says, but with a fun twist! Packed in a portable Gameboy-esque cardboard casing, Simon’s Windows (named after the 4 RGB LEDs) tests players’ memory and processing skills by changing things up with 2 differently-pitched buzzers, requiring players to process both visual and audio input at the same time.

Design

Description of the Game

While our game is based on Simon Says, the concept no longer revolves around purely following the visual pattern, but also having to process the possible inversion of buttons given by the two different buzzer beeps.

Every level has a randomly-generated sequence of 8 LED flashes, with each being accompanied by either buzzer beeping. When the lower-pitched buzzer sounds, the appropriate button corresponding to the flashing LED should be pressed. However, when the higher-pitched buzzer sounds, the correct button to be pressed is the one opposite instead.

There is no time limit, so players can take their time to enter the correct answer, and get to as high a level as possible!

Game Design

- Users would have to listen out for the pitch of the buzzer and input the correct sequence of positions to win the game.
- When the LED flashing sequence is occurring at the start of the game, either the lower-pitched or higher-pitched buzzer will sound during each flash, determining if players have to click the actual button, or the inverted button.
- Providing the correct answer sequence (taking into account both actual and inverted input) will allow the user to win.

Regular

- Triggered by lower-pitched buzzer sound
- Player will have to input **actual** position of the LED that flashed

Inverted

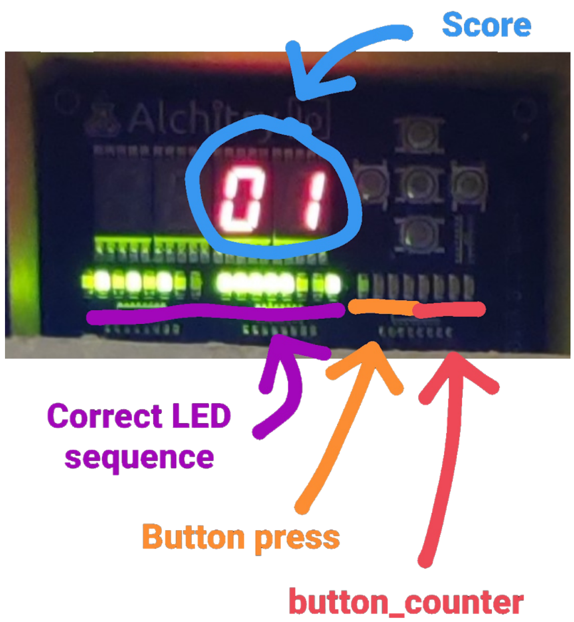
- Triggered by higher-pitched buzzer sound
- Player will have to input **inverted** position of the LED that flashed
  (top left flash → bottom right button pressed)

Design Inspirations

After a consultation with Prof Natalie, we realised that our initial idea was too complex and were advised to play around with the 4 elements of memory, reflex, rhythm and sound. Hence we had another sharing of ideas a few days after to rethink our game idea. It was then when we realised that all of us had at least 1 variant of Simon Says among our game ideas. In light of our favouritism towards the game, we managed to combine all 4 ideas into a single elegant game, which is simple to understand, but also complex to master. The end result was a unique take on the old classic, which combined the elements of not only memory, but also visual and audio cues too.

Going along with the old classic theme, we decided for the product itself to be looking like a handheld, portable Gameboy-ish concept, as a reminiscent of our childhood times. We designed the hardware around this idea of players being able to hold the entire case in their hands, with their fingers being able to comfortably press the game buttons while doing so.

Test Scenarios

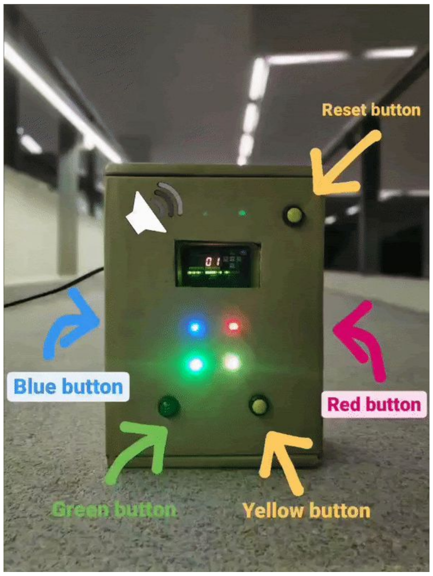


The FPGA LEDs were used to display the answers for every stage for testing purposes. This came out of our frustration for not being able to succeed in completing the levels reliably, allowing us to determine if the output sequence of both the LEDs and Buzzers are correct. The answers displayed also allowed us to test the LED outputs if a wrong sequence was put in by the player while also testing if the seven segments displayed the correct output (the next level) when a correct answer is put in.

The second test was to light up the FPGA LEDs when the corresponding buttons were pressed. As our LED display on the game is hardwired to the buttons, as shown in the stripboard schematic (Fig. 2.2), we were unable to test if the FPGA is detecting a button press correctly. Hence, we needed to output the button press to the LEDs on the FPGA to test the input.

The third test was displaying the button_counter (Fig.1.1) value on the FPGA LEDs. This allowed us to test if the buttons are working properly as they should be registered as only 1 button press. Initially we thought that the buttons were a little faulty as they would sometimes be registered as multiple presses when we only pressed once. However, it turns out that our solder joint was loose, which resulted in random connections and disconnections during the duration of a button press. This test was created because the previous test could not detect such an issue as the led would be switched off and on too fast for our naked eyes to catch.

User Manual



1. Carry the gamebox with both hands
2. Press the yellow start button at the top-right corner of the gamebox

3. Observe the 8 part sequence of the LEDs and buzzers

4. Input your 8 part answer
5. If your answer is correct, your score will increment by 1, and sequence for the next level will begin. Jump to Step 3.

6. If your answer is wrong, all 4 coloured LEDs will light up, signalling that the game is over.
7. To restart the game, jump to Step 2.

Budget

Total Cost for Entire Project: S$27.83

Amount consists of:

- Cost of components used in the **final product**
- Cost of components used for **prototyping**

Components in Final Product

No.	Component	Quantity	Unit Price (S$)	Total Price (S$)
1	12mm Momentary Push Button	5	0.89	4.45
2	RGB 4-pin LED (Clear)	4	0.36	1.44
3	Active Buzzer (w Wire)	1	1.07	1.07
4	Active Buzzer (PCB Mount)	1	0.25	0.25
5	**10**x Male to Male Jumper Wire	1	0.22	0.22
6	**10**x Male to Female Jumper Wire	1	0.22	0.22
7	**10**x Female to Female Jumper Wire	1	0.22	0.22
8	Cardboard Box	1	0	0
	TOTAL			**7.87**

Components Used For Prototyping

No.	Component	Quantity	Unit Price (S$)	Total Price (S$)
1	12mm Momentary Push Button	6	0.89	5.34
2	RGB 4-pin LED (Clear)	4	0.36	1.44
3	RGB 4-pin LED (Diffused)	4	0.36	1.44

4	Active Buzzer (w Wire)	1	1.07	1.07
5	Active Buzzer (PCB Mount)	1	0.25	0.25
6	Passive Buzzer (w Jumper Housing)	1	0.32	0.32
7	Piezo Buzzer (w Wires)	1	0.36	0.36
8	**30**x Male to Male Jumper Wire	1	0.67	0.67
9	**30**x Male to Female Jumper Wire	1	0.67	0.67
10	**30**x Female to Female Jumper Wire	1	0.67	0.67
11	Stripboard (Small)	1	0.57	0.57
12	Finex Solder Lead	1	1.07	1.07
13	Detachable Soldering Iron Stand	1	2.86	2.86
14	Soldering Iron Tip Cleaner	1	8.57	8.57
	TOTAL			**19.96**

Summary

Overall, the project was a very valuable experience as it allowed us to apply the theoretical knowledge gained from the Computation Structures course on a physical product. As such, we were able to better familiarise ourselves with the various intricacies that we may not have thought of while only learning the theoretical portion. In addition, the project incorporates both a hardware aspect, circuitry schematics, as well as a software aspect, FPGA Lucid coding. Thus, we were better able to understand the relations between the hardware portion and the software portion and the various complications that may occur in the process of merging the two aspects. In the end, the creation of a working product left all of us very satisfied as the final game product was the cumulation of all 12 weeks of effort, both physically and mentally. As such, our group is very grateful for the experience we have gained throughout the game creation process.

References

Lucid Tutorials:
https://alchitry.com/blogs/tutorials/your-first-fpga-project
https://alchitry.com/blogs/tutorials/synchronous-logic-1
https://alchitry.com/blogs/tutorials/io-element
https://alchitry.com/blogs/tutorials/sdram

https://natalieagus.github.io/50002/fpga_1.html
https://natalieagus.github.io/50002/fpga_2.html
https://natalieagus.github.io/50002/fpga_3.html
https://natalieagus.github.io/50002/fpga_4.html

Lucid Documentation:
https://cdn.shopify.com/s/files/1/2702/8766/files/Lucid_Reference.pdf

Appendix

1. ALU Design and Tests

Figure 1.1

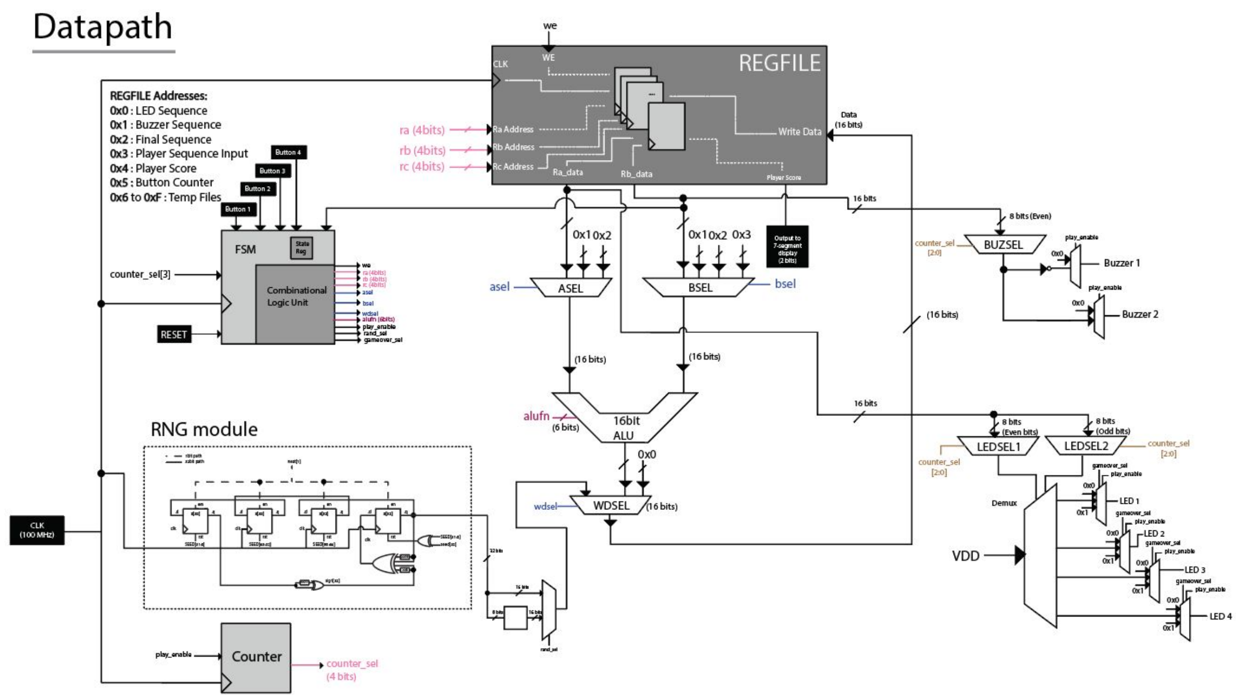
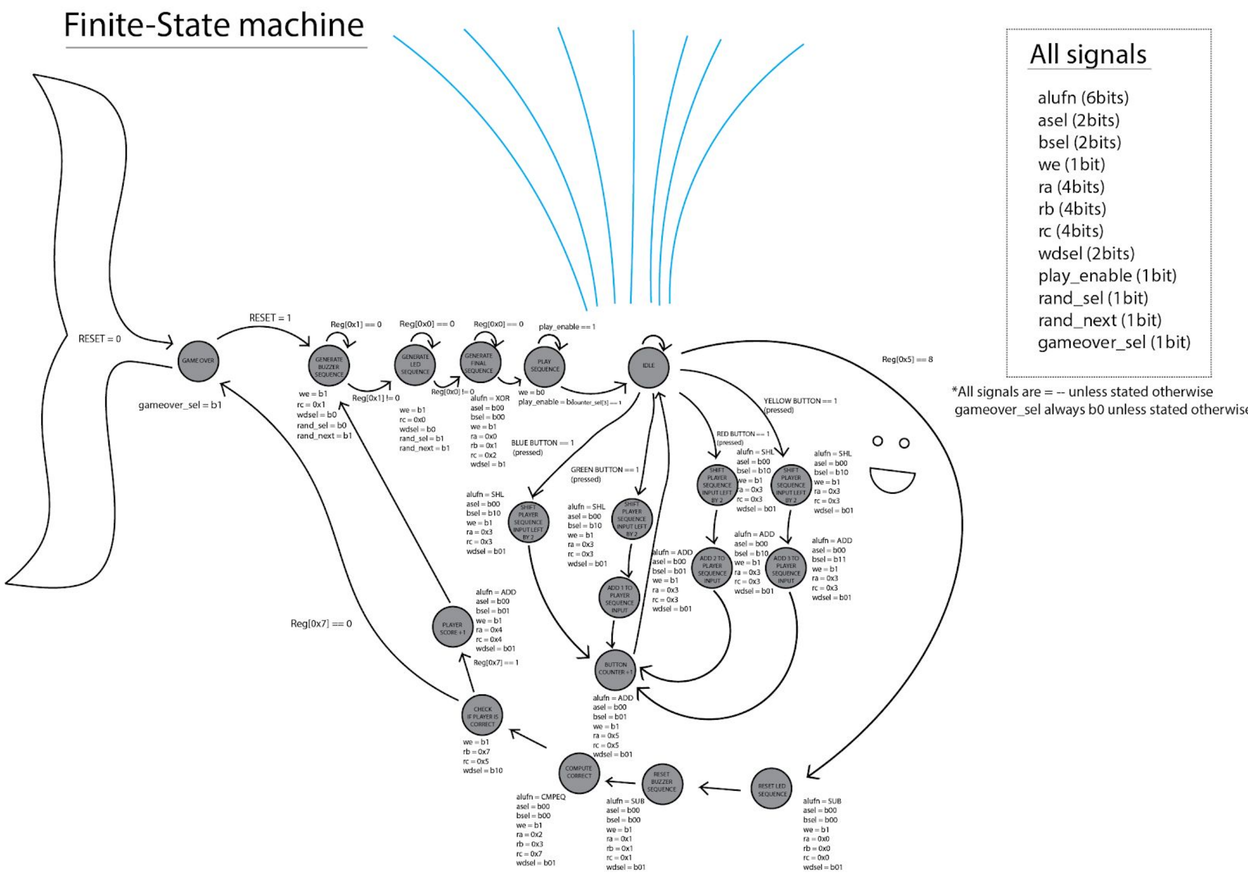


Figure 1.2



Finite-State machine
All signals
alufn (6bits)
asel (2bits)
bsel (2bits)
we (1bit)
ra (4bits)
rb (4bits)
rc (4bits)
wdsel (2bits)
play_enable (1bit)
rand_sel (1bit)
rand_next (1bit)
gameover_sel (1bit)
*All signals are = -- unless stated otherwise
gameover_sel always b0 unless stated otherwise
RESET = 0
RESET = 1
GAME OVER
gameover_sel = b1
GENERATE BUZZER SEQUENCE
Reg[0x1] == 0
Reg[0x1] != 0
we = b1
rc = 0x1
wdsel = b0
rand_sel = b0
rand_next = b1
GENERATE LED SEQUENCE
Reg[0x0] == 0
Reg[0x0] != 0
we = b1
rc = 0x0
wdsel = b0
rand_sel = b1
rand_next = b1
GENERATE FINAL SEQUENCE
Reg[0x0] == 0
alufn = XOR
asel = b00
bsel = b00
we = b1
ra = 0x0
rb = 0x1
rc = 0x2
wdsel = b1
PLAY SEQUENCE
play_enable == 1
we = b0
play_enable = b1counter_sel[3] == 1
IDLE
Reg[0x5] == 8
BLUE BUTTON == 1
(pressed)
GREEN BUTTON == 1
(pressed)
RED BUTTON == 1
(pressed)
YELLOW BUTTON == 1
(pressed)
SHIFT PLAYER SEQUENCE INPUT LEFT BY 2
alufn = SHL
asel = b00
bsel = b10
we = b1
ra = 0x3
rc = 0x3
wdsel = b01
ADD 1 TO PLAYER SEQUENCE INPUT
ADD 2 TO PLAYER SEQUENCE INPUT
ADD 3 TO PLAYER SEQUENCE INPUT
alufn = ADD
asel = b00
bsel = b01
bsel = b10
bsel = b11
we = b1
ra = 0x3
rc = 0x3
wdsel = b01
BUTTON COUNTER +1
alufn = ADD
asel = b00
bsel = b01
we = b1
ra = 0x5
rc = 0x5
wdsel = b01
RESET LED SEQUENCE
alufn = SUB
asel = b00
bsel = b00
we = b1
ra = 0x0
rb = 0x0
rc = 0x0
wdsel = b01
RESET BUZZER SEQUENCE
alufn = SUB
asel = b00
bsel = b00
we = b1
ra = 0x1
rb = 0x1
rc = 0x1
wdsel = b01
COMPUTE CORRECT
alufn = CMPEQ
asel = b00
bsel = b00
we = b1
ra = 0x2
rb = 0x3
rc = 0x7
wdsel = b01
CHECK IF PLAYER IS CORRECT
we = b1
rb = 0x7
rc = 0x5
wdsel = b10
Reg[0x7] == 1
Reg[0x7] == 0
PLAYER SCORE +1
alufn = ADD
asel = b00
bsel = b01
we = b1
ra = 0x4
rc = 0x4
wdsel = b01

2. Prototype Schematics

Figure 2.1 - User Interface

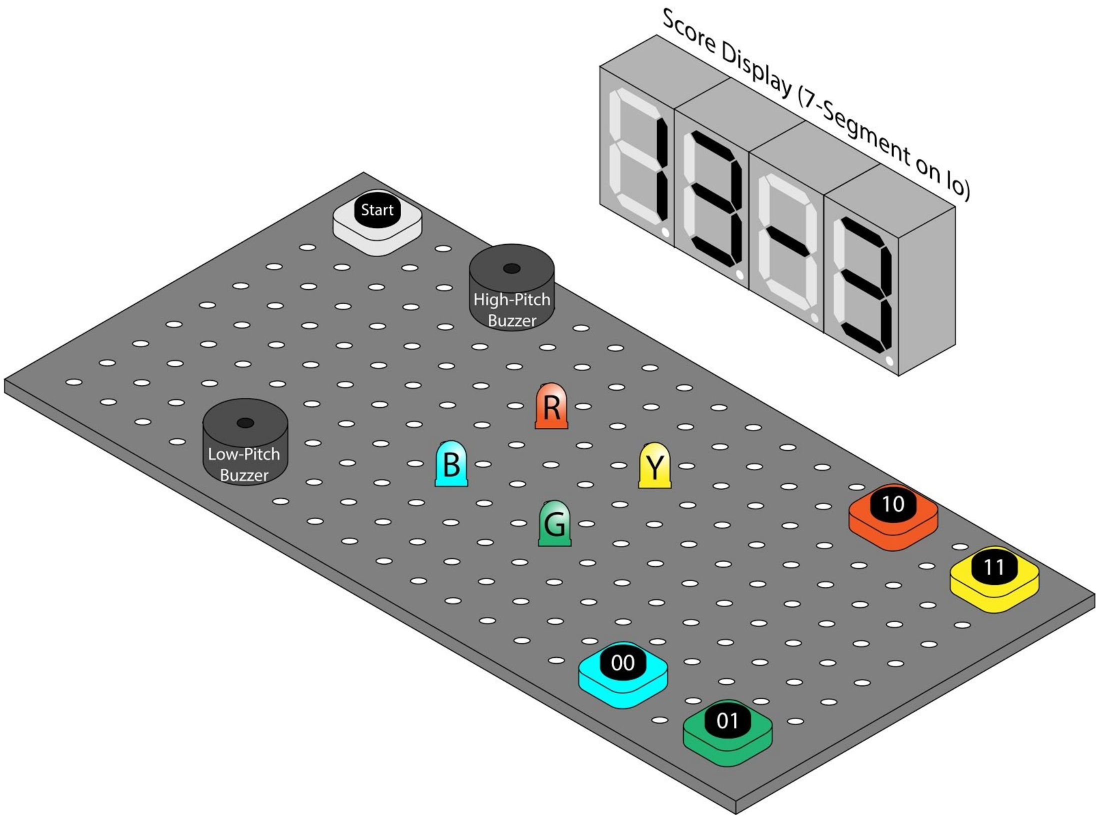
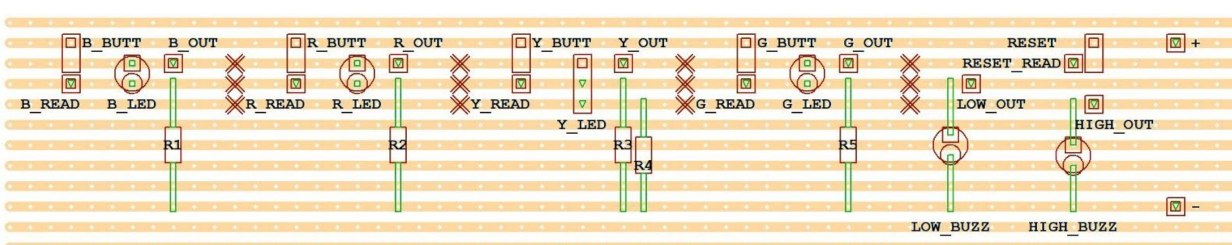


Figure 2.2 - Stripboard Design

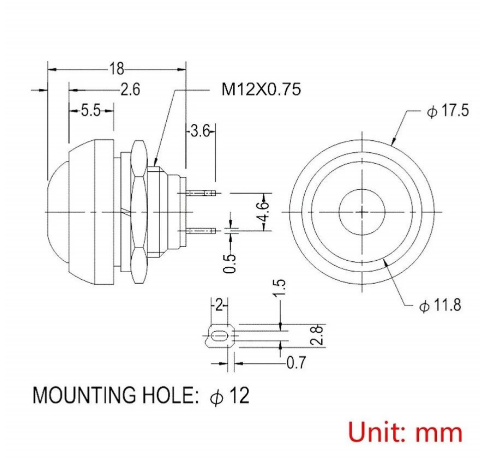


3. Project Management Log: Team Tasks

- Darren
    - REGFILE
    - Project Poster
- Jun Kai
    - LED and Buzzer Sequences
    - Randomiser and au_top
- Le Xuan
    - Product Hardware
    - Electronic Circuitry
- Russell
    - FSM
    - Project Video

4. Components' Specifications

1) Buttons x 5 (RED x 1 , BLUE x 1, GREEN x 1, YELLOW x 2)

a) 

2) RGB LED x 4

Parameter	Symbol	Value	Unit
Power dissipation	PD	180	mW
Forward current	IF	30	mA
Pulse current	IFP	100	mA
Operating temperature	Topr	-25 ~ +80	℃
Storage temperature	Tstg	-30~+100	℃

a)

Part No.	Color	Light intensity (mcd)*3		Forward Voltage (V)		Wavelength (nm)		Reverse Leakage (μA)	
		Typ	Max	Typ	Max	Typ	Max	Max	VR
F51BC9RG B-A	R	3000	4000	1.9	2.2	630	635	10	5
	G	6000	8000	3	3.3	515	525	10	5
	B	2000	3000	3.1	3.4	460	470	10	5

3) High pitch buzzer x 1

a) Operating voltage: 2-6V DC
b) Diameter: 1.2cm

4) Low pitch buzzer x 1

a) Operating voltage: 6-12V DC
b) Diameter: 2.2cm

5) 330Ω Resistor x 5