EE2026 Project Report 2019

Lab Session: Thursday PM

Member 1:
Name: March Phua Hsiao Meng
Matriculation No: A0183219A

Member 2:
Name: Suther David Samuel
Matriculation No: A0182488N

No.	Feature	Owner	FPGA Input Used	Feature Description	Display (Figures)
1	Basic Voice Visualizer	Team	SW0	When SW0 = 0, the VGA displays a 20kHz ramp wave. When SW0 = 1, the VGA will display a voice waveform	Fig. 1
2.A.	Volume Indicator	Marc	N.A.	{LED11, LED10, LED9, LED8, LED7, LED6, LED5, LED4, LED3, LED2, LED1, LED0} is varied based on the amplitude of the of the voice waveform. This is done by passing in the sound wave from the voice capturer module into the counter. Using certain threshold values, we light up the LEDs. {AN0, AN1} is toggled at high frequency to display the intensity of the sound from 0 to 12 7-Segment display is used to display the digits 0 to 12	Fig. 2
2B.	Basic Scope Displays on VGA	Samuel	SW1,btnL, btnR, btnU, btnD, btnC	Grid Lines (crosses) are drawn at every 80th pixel along the horizontal axis and 64th pixel along the vertical axis using VGA_VERT_COORD and VGA_HORZ_COORD Ticks are drawn at every 16th pixel horizontally and every 8th pixel vertically with every vertical coordinate within 508 and 516 and every horizontal coordinate within 652 and 628 using	Fig. 3

				VGA_HORZ_COORD and VGA_VERT_COORD Background is drawn at every pixel that does not satisfy the ticks, axes and grid conditions based on VGA_HORZ_COORD and VGA_VERT_COORD Voice Captured Waveform is drawn in the middle of the screen When SW1 is on, the current view is frozen. When SW1 is off, the task view is unfrozen. When btnL is pressed, the colour of the axes are changed When btnR is pressed, the color of sub-gridlines are changed When btnC is pressed, the background colour is changed When btnU is pressed, the voice capture waveform colour changes When btnD is pressed, the ticks colour changes There are 5 colors for each of the components and hence by pressing the different buttons, different combinations could be achieved. SW2 can be used to toggle the display of gridlines. SW3 can be used to toggle the display of the ticks. SW4 can be used to toggle the display of the axes. SW5 can be used to toggle the display of the background.	

				SW6 can be used to toggle the display of the sound wave.	
3	Menu Screen	Team	SW15	This switch toggles between the task and our special features. It serves to inform the user that our special feature includes a sound visualizer as well as a sound-based game. Help option is also available to display all available options to the user. The text is the menu screen is implemented using Derek-X-Wang/VGA-Text-Generator from GitHub.[1]	Fig . 4
4	Sound amplitude dependent color changing game	Team	SW15 and SW14	This displays the sound waveform. Here, the sound wave changes color depending on the amplitude of the sound detected. The background color, grid color, axes color, ticks colors can be changed or even disabled as usual just like in the task.	Fig. 5
5	Volume Bar	Marc	SW15 and SW14 and SW12	Displays horizontal square bars of gradual gradient colors (green to yellow to red) to display the volume amplitude. This updates at 5Hz, the same frequency as the LEDs and the 7seg.	Fig. 6
6	Piano - square grid waveform	Marc	SW15 and SW14 and SW7	Horizontal color changing piano square grid waveforms are displayed here to represent the sound wave. This is a sound wave visualization. The higher the amplitude, the higher the bars. This feature is implemented by taking the sound amplitude at larger intervals.	Fig. 7
7	Circular Splat waveform	Marc	SW15 and SW14 and SW6	A circular splat waveform representative of the sound is displayed. This is another form of sound wave visualization.	Fig. 8
8	Rainbow Display	Samuel	SW15 and SW14 and SW9	Generates a nice rainbow based on the amplitude of the sampled sound wave. This is implemented by having semi-circles of rainbow colors overlaying one another. Each ring shows up when the sound reaches a particular level. The rainbow is representative of the sound intensity and refreshes at 5Hz frequency.	Fig. 9

9	Pre-set waves	Team	SW15, SW14, SW11 and btnC, btnU, btnL, btnR, btnD	When SW15, SW14 and SW11 are switched on, preset waves are generated. -Triangular waveforms -Jaw shaped waveform (rising) - Square Waveforms -Jaw-shaped (falling) waveforms -btnC can be used to switch among these waveforms -btnU can be used to increase the frequency -btnD can be used to decrease the frequency -btnR can be used to increase the amplitude -btnL can be used to decrease the amplitude This preset wave can also be passed into our sound visualizer to study waves and hence could be used as an education tool for students.	Fig. 10
10	Game	Team	SW15, SW14 and SW13	When SW15, SW14 and SW13 are switch on, our mini game is activated. A spaceship will move left to right and right to left continuously at the bottom of the screen. 4 blocks of different colors are generated at the top of the screen and they start to move towards the bottom towards the player's territory. When the player makes a noise, a bullet of a certain color corresponding to the intensity of the sound is generated. The objective is for the player to hit the targets with the bullets. However, the target is only destroyed if the bullet color matches the target color. If any of the target enters the player's territory, the player loses and the	Fig. 11

				game immediately ends. If the player successfully destroys a block, another block generates at the top of the screen. Each time the player successfully destroys a block, a point is added.	
11	Shaded Wave with amplifier	Samuel	SW15, SW14, SW6, btnU and btnD	The entire displacement of the selected wave (sound wave or preset wave) with respect to the horizontal axis is now shaded. This is another visualization of the wave. -btnU and btnD are used to amplify and diminish the waves respectively. This allows for the user to visualize softer waves in greater detail by amplifying it.	Fig. 12
12	Block Waves	Team	SW15, SW 14, SW8 and SW6 btnU and btnD	-When SW15, SW14, SW8 and SW6 are turned on, a block wave is generated about the horizontal axis. -This effect is achieved by combining Samuel's shaded wave and Marc's Piano wave effects showing that combinations of sound visualizations are possible in our system. The amplification functionality could be applied on this wave as well with btnU to amplify and btnD to diminish the wave. -Behind the scenes, this is achieved by first applying the piano waveform to the sound wave. The shading effect is then applied to this waveform to achieve the block waves.	Fig. 13
13	Circular Waveform	Team	SW15, SW14, SW11, SW7 btnR, btnU, btnD, btnL and btnC	- Another complex example of how sound visualization features can be combined to achieve new features. - When SW15, SW14, SW11 and SW7 are turned on and the square preset waveform is selected, 4 circles appear on the screen. Changing the frequency changes the size of the squares making up the circles.	Fig. 14

				-By changing the amplitude of the preset wave, the circles can be made to move out away from each other or in towards each other	
14	Piano -Image Waveform	Team	SW15, SW14, Sw11 and Sw8	-When SW15, SW14, SW11 and SW8 are switched on, and btnC is pressed until square waveform is selected, a piano kind of shape is displayed on the screen.	Fig. 15
				-When SW15, SW14, SW11 and SW8 are switched on, and btnC is pressed until Jaw-tooth waveform is selected, it creates a jaw-toothed waveform	
				-When SW15, SW14, SW11 and SW8 are switched on, and btnC is pressed until triangle waveform is selected, running mountains waveforms are created on the screen	
				-This is an example of how different wave combinations can result in unique visualizations in our implementation. Also, the amplitude and frequency of these waveforms can be controlled	
15	Sound Visualizations credits overlay screen	Team	SW15, SW14 and SW10	A small overlay screen with green border and black background displays the text “EE2026 Sound Visualizations by Marc and Samuel”.	
16	Help Menu	Team	SW15 and SW12	-When SW15 and SW12 are turned on, a help menu is displayed -Help menu displays the various switches needed to activate the various waveform and the game.	Fig. 16

Appendix

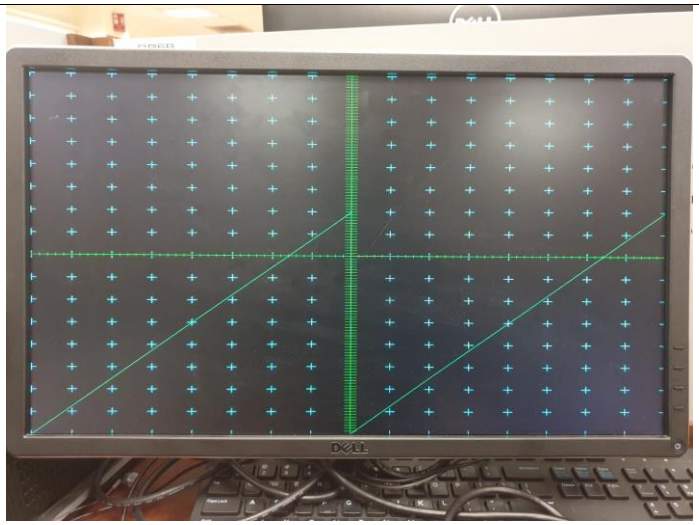

Fig 1

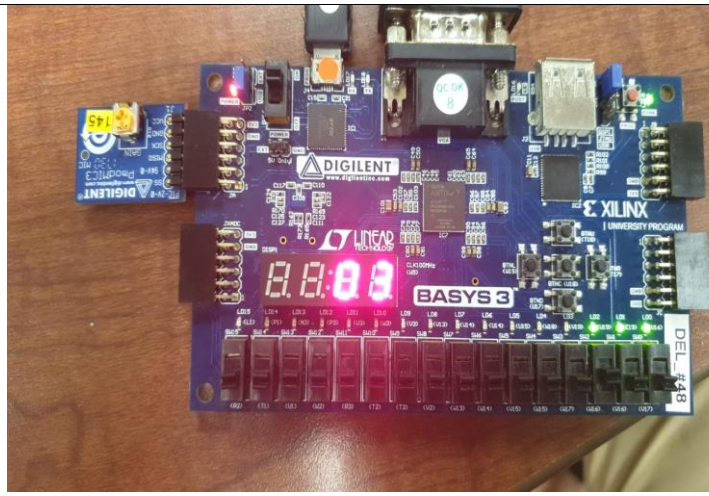

Fig 2

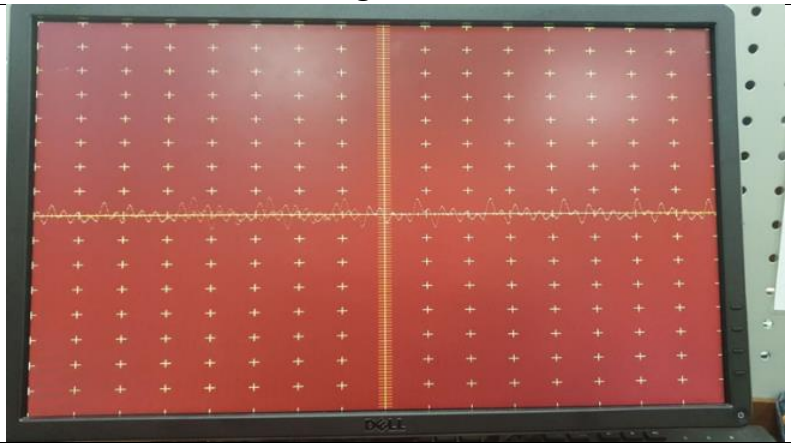

Fig 3

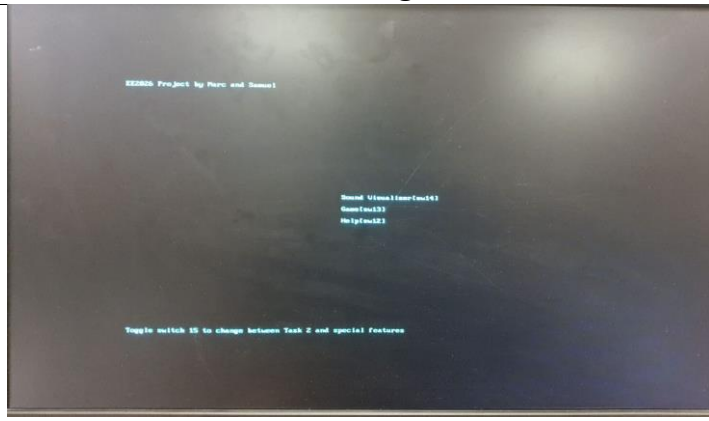

Fig 4

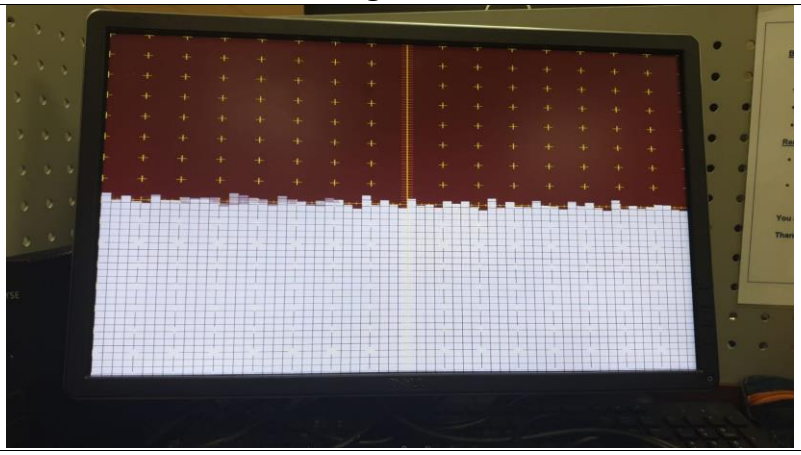

Fig 5

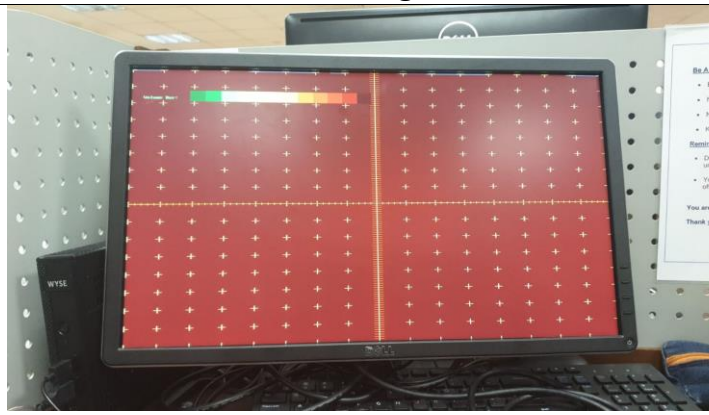

Fig 6

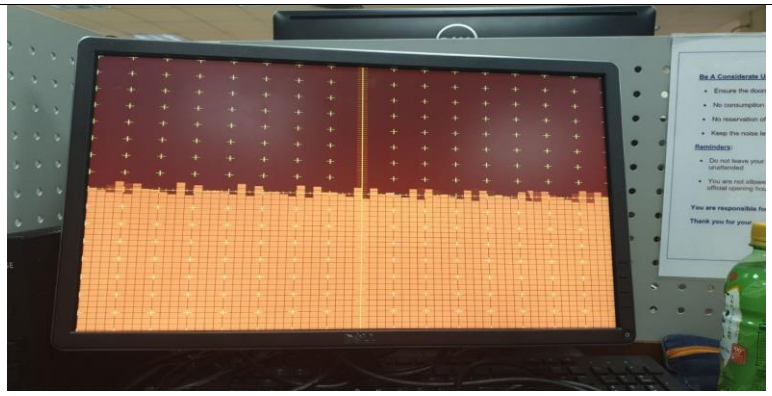

Fig 7

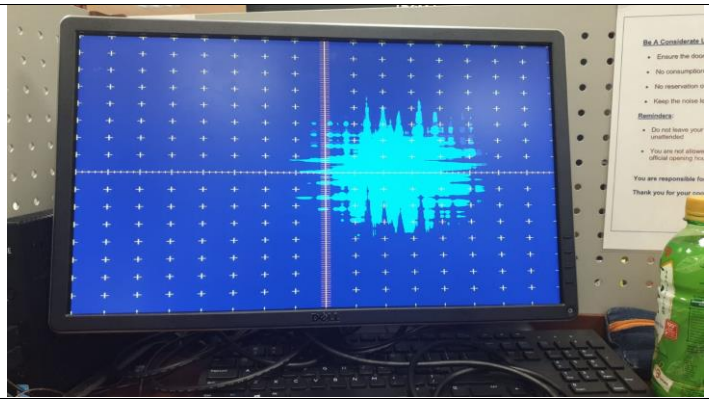

Fig 8

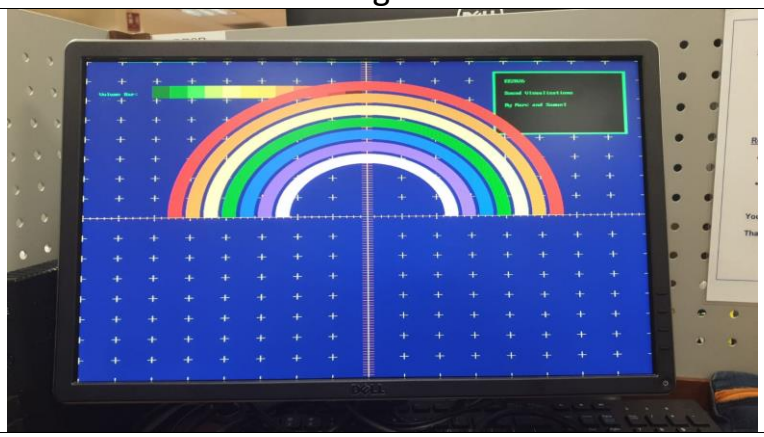

Fig 9

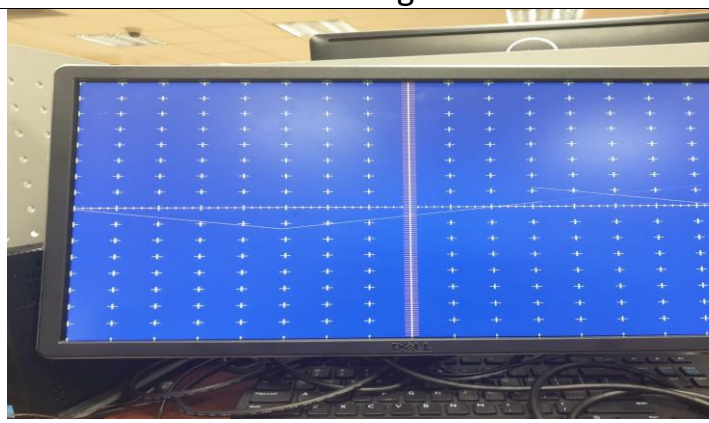

Fig 10

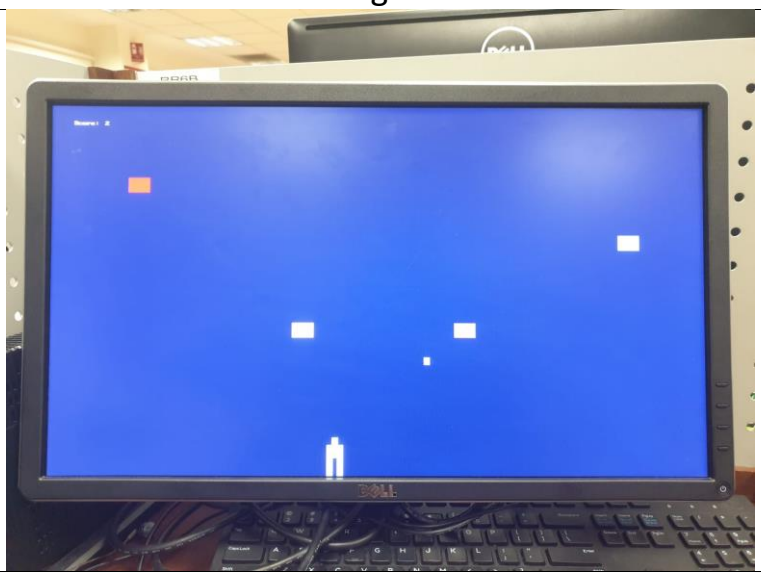

Fig 11

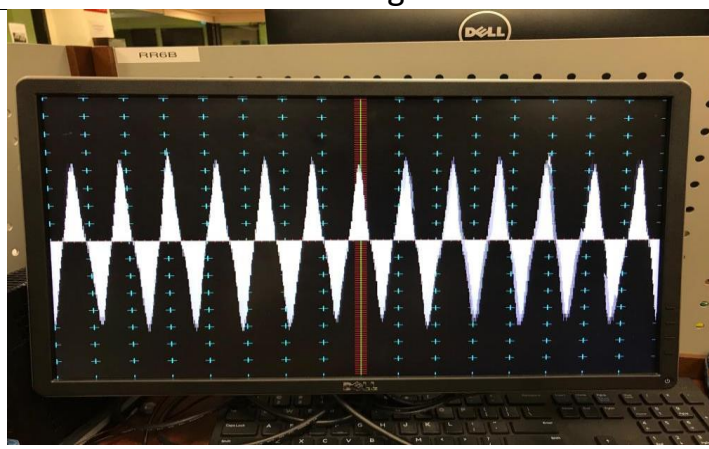

Fig 12

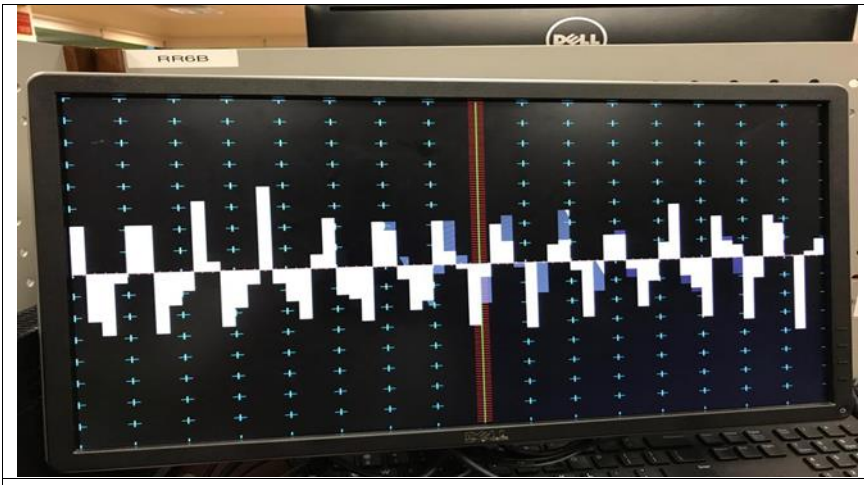

Fig 13

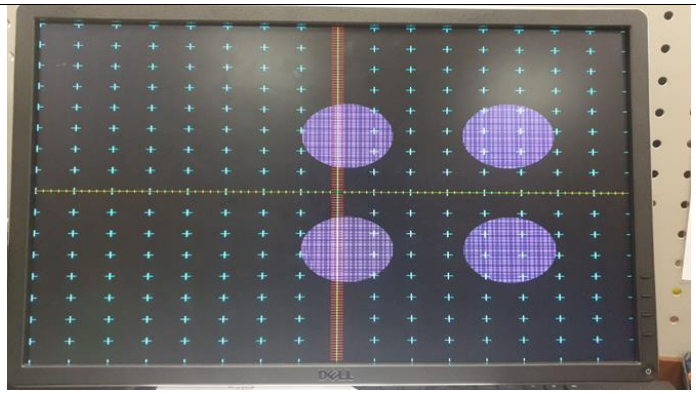

Fig 14

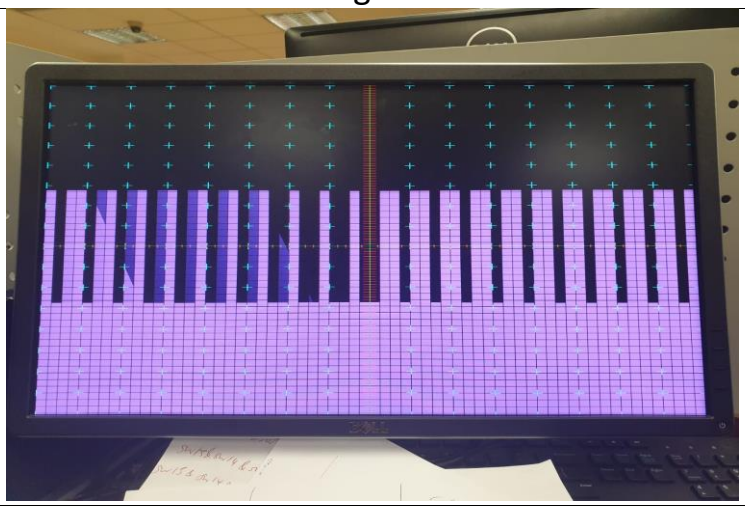

Fig 15

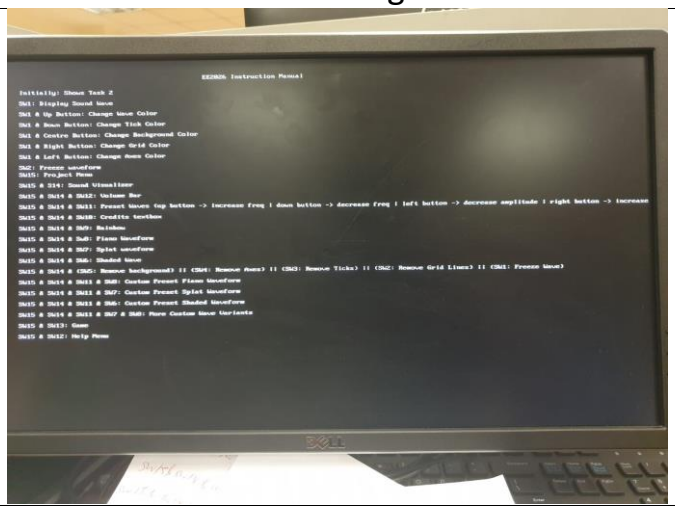

Fig 16

References:

[1] VGA Text Generator: https://github.com/Derek-X-Wang/VGA-Text-Generator was used for all the texts used in our project

Feedback:

Marc Phua	**Suther David Samuel**
1. What did you like most/least about the project	
Like most: Programming a game on Verilog and seeing it work was very satisfying. **Least:** As Verilog's programming methodology is very different from the other computer programming languages such as C, it required a lot of hardcoding for the game.	**Most liked:** Playing around with the sound wave and background to achieve the different effects and also combining different effects to achieve totally new effects. **Least:** A good portion of the time we spent on the project was just wasted waiting for bitstream to be generated.
2. How would you suggest the overall project assignment be improved?	
-	One mic could be given to each person so that both of the team members could work with the soundwave independently for their individual components. Since we only had one mic per team member, for our group, we decided to generate our own waveform so that we could work independently.
3. Any other constructive feedback/suggestions are welcome	
-	**-**