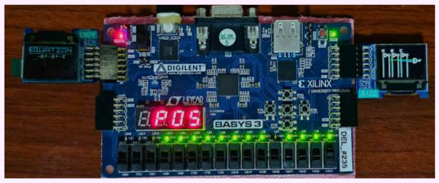
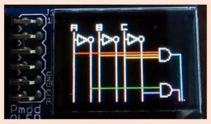
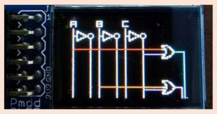
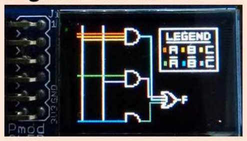
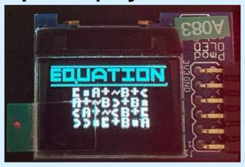
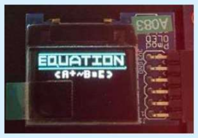
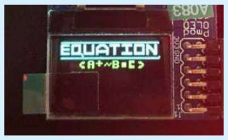
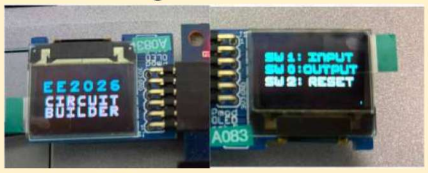
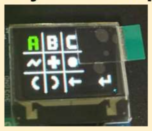
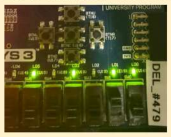
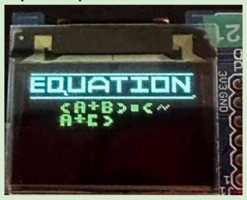
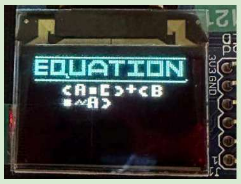
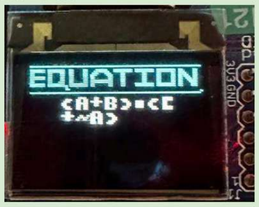
User guide (2 pages)
Group ID: S3_02
Team Members: Si Thu Lin Aung [redacted],Joe Tien You [redacted], Liang Xuanyin Glenda [redacted],Goh Aik Haw [redacted]

PERSONAL AND TEAM IMPROVEMENTS		
Student and Improvement Name	Improvement Description	Images / Photos
Team: S3_02 Project Title: Circuit Builder	The Circuit Builder project utilizes a Basys3 FPGA board to help users seamlessly compute and visualize the Minimum Sum of Products (MSOP) and Minimum Product of Sums (MPOS) for combinational logic equations of up to 32 characters. Main Features: - **Input (keyboard display)**: A 3 x 3 keyboard which uses the 5 buttons for navigation and entering an equation. - **Input (Display of inputs)**: Display the current string of characters. - **Output (Circuit)**: Draws the circuit of the MPOS / MSOP. - **Output (MSOP / MPOS Equation)**: Displays the MPOS / MSOP generated from the input of the user. Screen Selection: - Input Screens: sw[1:0] = 2 - Output Screens: sw[1:0] = 1 Restriction: Only 3 variables and each gate accepts up to 3 inputs.	
Student A: Si Thu Lin Aung "Output: Circuit Realisation"	To see the circuit drawing, the user needs sw[1:0] = 1 **Circuit Drawing** - Draws the circuit based on the MPOS / MSOP. - Wires are color coded for ease of visualisation - A legend is available so that users can easily map the wiring when the variable segments (A,B,C) cannot be seen. - Legend will not show the color coding of the wires from gate to gate. Color coding of such wires is purely for ease of identification - Covers all cases where there are 3 variables and each gate accepts up to 3 inputs. Cases leading to always T / F are covered. - The final gate is marked with an 'F' **Segment Display: MPOS / MSOP** - Circuit can be toggled between MPOS / MSOP by pressing btnC. - The segment display shows the current state of the circuit diagram **Virtual Oled Display** - Display size is increased to **142 x 96 pixels** since the normal 96 x 64 oled screen is too small for the circuits to be fully realised. - Navigation can be done using the 4 buttons - btnU, btnD, btnL, btnR (both long presses and short presses works)	Eg. Circuit Realisation of the equation: (A & B & C) | (~A & ~C) | (~B & ~C) **MSOP** (Picture only shows part of the full 142 x 96 screen)  **MPOS** (Picture only shows part of the full 142 x 96 screen)  **Legend on Screen** 
Student B: Joe Tien You "Input: Display Equation Typed"	**Input Display** - Left screen accurately reflects the user input from the keyboard when user presses **btnC** to select a key they want - Added a top **banner** to indicate the purpose of the interface for users to understand the functionality upon viewing. **Input Limitation and Layout** - Redesigned input from 1-row of 10 characters to a 2-row layout of 10 and 6 characters and finally to a cleaner and more aesthetically pleasing **4-row layout** with **8 characters** per row - Setting a **hard cap of 32 characters** for input length to support complex equations especially those with brackets	**Input display + 4-row character display:**  **Validation System:** Valid Input (White to Green): "(A | ~B & C)"

	Character Rendering - Collaborated with Student D to design **9 characters** display and placed them on a bitmap lookup table - Introduced **deletion feature** to allow corrections without the need to reset all inputs subsequently, **multiple deletes** are introduced to allow users to remove larger sections of their inputs **Validation Checker** - Designed and implemented a **custom validation system** to ensure logical syntax rules were upheld during input - Characters are coloured **green** if input is valid upon finalised entry of equation else it will be **white,** indicating invalid entry	  Invalid Input (Remains White) -> “(A | ~B &”
Student C: Liang Xuanyin Glenda “Input: Keyboard and Main Menu”	**Menu Page** - Default screens that are shown when none of the input/output screens are shown (when **sw[1:0] != 1 or 2**) - Left Screen: Project Title - Right screen: Switch instructions for toggling between screens. **Keyboard Display** - Main user interface for equation input. - Navigation is done by pressing **btnU, btnD, btnL, btnR**. - Press btnC for confirming the input and displaying its **4-bit representation** on **LEDs 0-3** - Current selected key is highlighted in **green**. **Input Buffer** - 128-bit buffer (supports up to **32 characters**) - Captures user key selections, storing each as a 4-bit value - **LED 5** indicates whether the buffer is valid when enter is pressed **Reset Input Equation** - Provides quick clearing of all input by toggling **SW2** from off to on.	**Main Menu Page:**  EE2026 CIRCUIT BUILDER SW 1: INPUT SW 0: OUTPUT SW 2: RESET **Keyboard Display:**  **LED** (LD5 marks valid equation and LD0 - LD3 marks the bit value sent to buffer)**:** 
Student D: Goh Aik Haw “Output: MSOP and MPOS computation”	**Equation Parsing** - The input boolean expression by the user is processed and saved in a simpler, readable string of bits **Generation of truth table** - Evaluates the simplified equation for all possible values of the inputs A, B and C. - Generates an 8-bit truth table describing the expression’s output for every possible input combination - Each Bit of the truth table corresponds to the output of the equation for a particular combination of values of A, B and C **MSOP/MPOS computation** - All possible combinations of MSOP/MPOS for variables A,B,C and operators +,*,(,) are pre calculated and stored in memory files - Retrieves the MSOP/MPOS of the equation from memory based on the 8-bit truth table **MSOP/MPOS display** - Colab with student B to display MSOP/MPOS equation of the input equation - By pressing btnC, the user can dynamically swap the displayed equation between MSOP and MPOS.	Ex Equation: (A | B) & (~A | C) Input equation:  MSOP:  MPOS: 

References and feedback

Group ID: S3_02

Team Members: Si Thu Lin Aung [redacted],Joe Tien You [redacted], Liang Xuanyin Glenda [redacted],Goh Aik Haw [redacted]

References Used:

Si Thu Lin Aung	- MSOP and MPOS were pre-computed and stored in a mem file using a python code generated purely by chat gpt. Our team was only responsible for debugging this code and changing the output a little to fit the needs of the project. (https://drive.google.com/file/d/1zC8832j8hF3Dxc7HHT2Epq-3kVMEiuya/view?usp=sharing) - Initially tried to use a python code (also generated by gpt) to create a netlist.(NO LONGER used in the final project)
Goh Aik Haw	
Joe Tien You	- Utilised picture2pixel code to compute screen permutations during proof of concept (https://www.comp.nus.edu.sg/~guoyi/project/picture2pixel/) (NO LONGER used in the final project) - ChatGPT was used to tweak the .py file codes to reduce LUTs of the generated verilog output (NO LONGER used in the final project)
Common to all members	- Utilised an online art drawer to assist in the drawing of components in the project. (https://www.pixilart.com/draw)

Course Feedback:

- No clear distinction of what qualifies as a “good” project, causing the team to be overly ambitious. Suggestion would be to provide a clear rubric on the grading of the project component.
- We believe it would be helpful to set up consultation sessions for the project. There were times where some issues took too long to debug because no one could figure out the problem.
- More could be done to teach the Verilog coding components. Some suggestions include but are not limited to: useful documentations on very useful tools like tasks, functions and generate blocks which helps in reducing “hardcoding” and refactoring of the codes for better readability.
- It might be a good idea to place quiz 2 after the project submission instead of placing it during the crunch period of the project as this may cause students to lose their momentum on the project.
- Labs were closely linked to the project allowing us to easily progress through step by step, however the lab content can feel quite detached from the lecture content this could be due to how sequential logic and FSM usage were introduced too late in the course