EE2026: DIGITAL DESIGN
Academic Year 2023-2024, Semester 1

FPGA Design Project

Mouse & OLED Digital System (MODS)

GROUPING

Students are required to form their own team of 4. Depending on class size, we may need to create one to three teams of 3.

We will arrange for all students without a team to meet each other during the lab session in week 7. Then you can form your own team after checking with each other.

ABSTRACT

A Mouse & OLED Digital system (MODS) that makes use of the Basys3 development board and the following devices will need to be created:

Device	Description
Computer Mouse *1 piece / team*	Clicks and movements on this device will provide digital signals to the FPGA
Pmod OLED *1 piece / member*	Information can be displayed on this 96x64 pixels display screen with 16-bit colour resolution

You will need to replace the devices at your own cost if damaged due to negligence.
Deadline extensions are not given for damaged components. Use them with upmost care!

This manual and the first two project sessions **briefly guides** (Exact steps are not provided, and you are expected to code, observe, and make conclusions!) you in getting started with a basic system consisting of the Mouse and OLED.

It is **required to be very familiar and at ease with the dataflow, structural and behavioural modelling** that have been practiced through the previous three lab assignments.

After the basic MODS has been implemented, you will need to use your own technical skills, logic, creativity, and resourcefulness to create an enhanced MODS (E-MODS), while deciding what is best implemented within the resource limitations of the FPGA.

PLAGIARISM WARNING

This is a teamwork project. Discussions are encouraged. However, 'discussion' is not a valid excuse if your codes turn out to be uncomfortably similar. NUS and the EE2026 teaching team take plagiarism very seriously.

- Warning from the NUS Code of Student Conduct:
  "Any student found to have committed or aided and abetted the offence of plagiarism may be subject to disciplinary action" https://www.nus.edu.sg/celc/programmes/plagiarism.html

- Both the source and recipient of the project solutions (codes) or reports will be **unconditionally penalised** in such cases.

- Students will also be reported to BoD (Board of Discipline).

1. SCHEDULE AND PROJECT COMPONENTS

The MODS includes both **individual** and **teamwork** components. Some milestones that you are requested to achieve will be detailed in subsequent sections of this project manual.

Week	Suggested Task to Complete for Each Week	Individual Marks	Team Marks	Evaluation
7	**GETTING STARTED WITH THE OLED**			Individually evaluated at the start of the **week 8 lab session only**. CANVAS submission is not required
	Student A: Basic task 4.A1	1 %		
	Student B: Basic task 4.B1	1 %		
	Student C: Basic task 4.C1	1 %		
	Student D: Basic task 4.D1	1 %		
8	**INDIVIDUAL BASIC TASK**			Evaluated at the start of the **week 9 lab session**, in **ONE** bitstream for the team. CANVAS **submission is required** during the evaluation in week 9. The grader will provide submission details during the grading. A penalty of up to 35% is applicable on the individual basic task and/or group task if they are evaluated, or re-evaluated, after your official lab session in week 9.
	Student A: Basic task 4.A (All Parts of 4.A)	3 %		
	Student B: Basic task 4.B (All Parts of 4.B)	3 %		
	Student C: Basic task 4.C (All Parts of 4.C)	3 %		
	Student D: Basic task 4.D (All Parts of 4.D)	3 %		
	GROUP BASIC TASK			
	Group: Mouse Drawing on OLED with Neural Net module instantiation for digit recognition. Integrating all the individual and group basic tasks into 1 bitstream		7%	
9	**IMPROVEMENT TASK - PROPOSAL**			CANVAS submission required by the end of week 9
	Group: Improvements proposal - Indicate group theme and task responsibilities		2%	
10/11 **No Lab in Week 11**	**INDIVIDUAL IMPROVEMENT IMPLEMENTATION**			Evaluated in week 12, in **ONE** bitstream for the team. Multiple bitstreams are not allowed. CANVAS **submission is required** at the end of Week 11. The basic tasks that have already been assessed in week 9 are not required in the bitstream and are not assessed in week 12. ***Note: There is also an in-lab Verilog evaluation (11%) in Week 10***
	Student A: Component A of group improvement	6 %		
	Student B: Component B of group improvement	6 %		
	Student C: Component C of group improvement	6 %		
	Student D: Component D of group improvement	6 %		
	GROUP IMPROVEMENT IMPLEMENTATION			
	Group: Full integration of component A, B, C and D to create the group improvement		6 %	
12	**INDIVIDUAL ASSESSMENT**			Evaluated in week 12
	Individual: Q&A, Execution, Understanding	5 %		

2. SOFTWARE FILES

A project template (**MODS.xpr.zip**) can be downloaded from CANVAS. The template consists of the following:

Design Sources:

Top_Student.v: The top-module of the design. This will be your main module, or typically called the Top Level module, where you instantiate the sub-modules and make the necessary links between these sub modules. **You will need to modify this module, as well as create other design sources for use in this module.**

MouseCtl.vhd: An interface module between the computer mouse, Ps2Interface.vhd and your design on the FPGA. This module signals when clicks and movements are detected on the computer mouse connected to the Basys3 USB port. A wired mouse is recommended, as not all wireless mice will work. **You are NOT required to make changes to this module. Optionally, you should read the description inside if you want to know more about the input and output signals.**

Oled_Display.v: An interface module between your design on the FPGA and the OLED display. This module works to send serial SPI data into the OLED display for initialization and drawing. **You are NOT required to make changes to this module. Optionally, you should read the description inside if you want to know more about the input and output signals.**

Paint.v: A module that connects the computer mouse and OLED, so that digits created by the mouse on the OLED can be recognised, and the results shown on the seven segments display and LEDs on the Basys3. **You are NOT required to make changes to this module.**

Constraint Sources:

Basys3_Master.xdc: A master constraints file that defines the I/O constraints for the Basys 3 Development Board

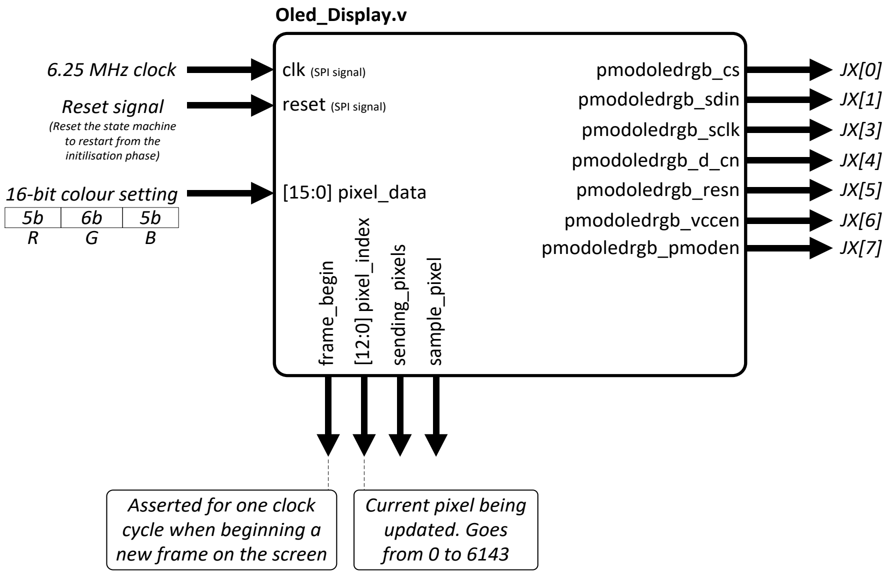
MODULE WIKI:

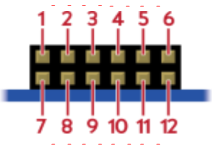
tiny.cc/ee2026wiki : The project wiki includes a Verilog error troubleshooting guide, VHDL vs verilog code comparison and useful resources when you are working on additional features.

3. MODULE SETUP AND TESTING GUIDELINES

3.A OLED_DISPLAY.V MODULE

Setting up the OLED by using `Oled_Display.v`





3.A1 Instantiate the `Oled_Display.v` module in `Top_Student.v`

3.A2 Create and provide a 6.25 MHz clock signal named `clk6p25m`, to `Oled_Display.v`

3.A3 Create a 16-bit signal named `oled_data` and initialize it with a value of `16’h07E0`
The `oled_data` defines the colour of each pixel. The RGB OLED screen has 16-bit colour resolution, this is represented through 5 bits for the red colour component, 6 bits for the green colour component, and 5 bits for the blue colour component. Connect `oled_data` to the `pixel_data` input of `Oled_Display.v`

3.A4 Connect the `Jx[7:0]` signals to the PmodOLED device accordingly

3.A5 Attach the OLED display to the Basys3 and update the constraints file accordingly
Very important: To avoid damaging the devices, ensure the Basys3 board is OFF before connecting the OLED
DO NOT touch the OLED ribbon cable when inserting / removing the OLED from the Basys3 board

3.A6 Optionally connect the `reset` port of `Oled_Display.v` to any pushbutton, or alternatively give the `reset` port a constant value of 0

3.A7 Generate the bitstream and download to the FPGA
Verify that the background colour of the screen is green

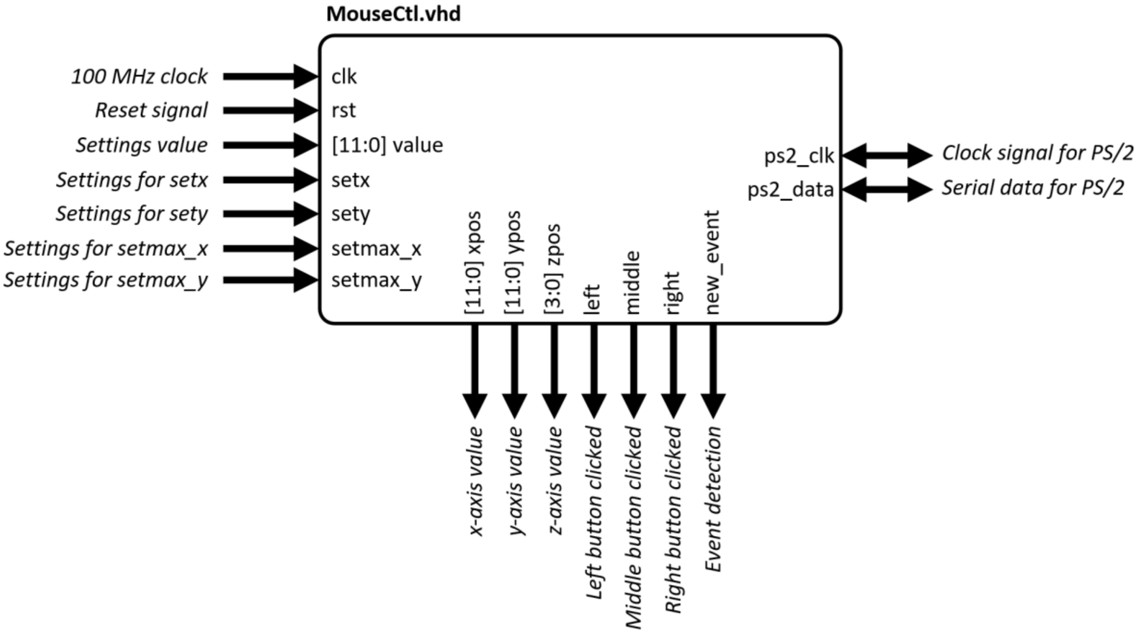
3.A8 Write the codes that uses sw[4] to select between one of these two possibilities:

- sw[4] is OFF: The background colour is green
- sw[4] is ON: The background colour is red

3.A9 Generate the bitstream and download to the FPGA
If the above steps are executed correctly, you would observe a green background on the OLED when sw[4] is OFF, and a red background when sw[4] is ON.

3.B MOUSECTL.VHD MODULE

Setting up the computer mouse by using `MouseCtl.vhd`



3.B1 Instantiate the `MouseCtl.vhd` **module in** `Top_Student.v`

3.B2 Provide a 100 MHz clock signal to `MouseCtl.vhd`

3.B3 Connect ps2_clk port and ps2_data port
`ps2_clk` is an inout signal which is connected to the FPGA PACKAGE_PIN C17.
`ps2_data` is an inout signal which is connected to the FPGA PACKAGE_PIN B17.

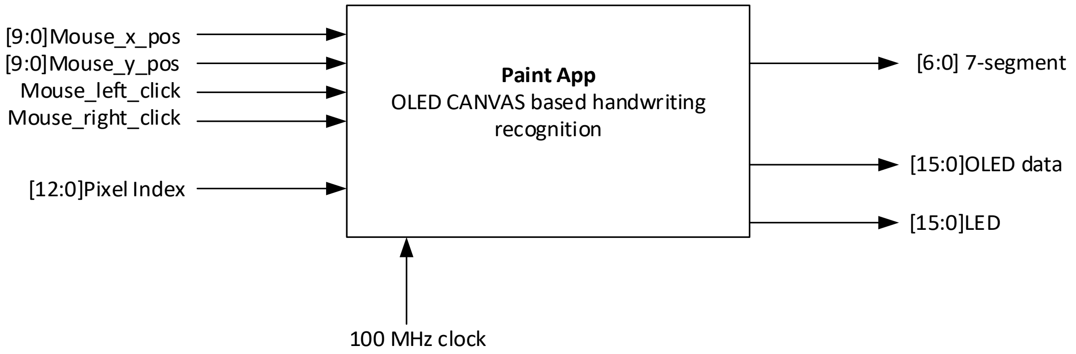
3.B4 Set default values for `value`, `setx`, `sety`, `setmax_x`, `setmax_x`
There is no need to offset any of these values, and we can assume the origin to be the default.

3.B5 Write the code that turns on certain leds based on the mouse clicks
led[15] must turn ON when the signal from "`left`" port is high
led[14] must turn ON when the signal from "`middle`" port is high
led[13] must turn ON when the signal from "`right`" port is high

3.B6 Generate the bitstream and download to the FPGA
Ensure that the mouse is connected to the (upper right) USB port of your Basys3 board. A wired mouse is recommended for testing purposes. Download your program to the FPGA. Press any of the three mouse buttons and observe which LED lights up on the Basys3 development board

3.C PAINT.V MODULE

Setting up the OLED based canvas handwritten recognition by using `Paint.v`



3.C1 Instantiate the `paint.v` **module in** `Top_Student.v`

3.C2 Instantiate both the `MouseCtl.vhd` **and** `Oled_Display.v` **modules in** `Top_Student.v`

3.C3 Connect mouse_x_pos, mouse_y_pos, **and** mouse_left_click **between** MouseCtl.vhd **and** Paint.v **ports.**

3.C4 Assign reset **to mouse_right_click then connect it between between** MouseCtl.vhd **and** Paint.v **ports.**

3.C5 Connect pixel_index **and** colour_chooser (or pixel_data) **between** Oled_Display.v **and** Paint.v.

3.C6 Provide a 100 MHz clock signal to Paint.v.

3.C7 Connect LED **and** seven segment display **to** Paint.v**.**

3.C8 Generate the bitstream and download to the FPGA

Ensure that the mouse is connected to the (upper right) USB port of your Basys3 board, and OLED is connected to corresponding pmod pin. A wired mouse is recommended for testing purposes. Download your program to the FPGA. Use mouse to choose the color on OLED and write any number (and only 1) by holding down "LEFT MOUSE BUTTON" on the white CANVAS. Observe the output from seven segment display and LED on basys 3 board. "RIGHT CLICK" to reset the CANVAS.

4. BASIC TASKS

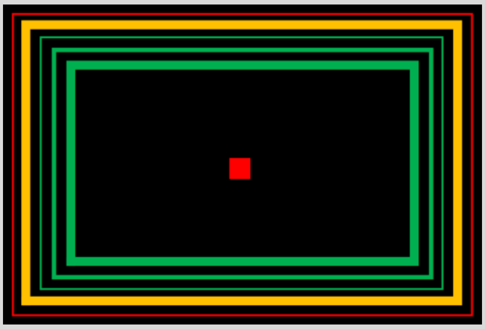
If your team consists of 3 persons, ignore any one of the four basic tasks (4.A or 4.B or 4.C or 4.D)

OLED COORDINATES SYSTEM:

Work on creating an **x** and **y** coordinate system derived from `pixel_index` to make the pixel drawings on the OLED screen less cumbersome (*Hint: You may employ the modulus % and / operators*)

4.A STUDENT A: BASIC TASK A

DESCRIPTION: To create multiple borders and shapes on the OLED. It would look approximately as:



4.A1 Create a **red border** near the edge (Around 2 to 5 pixels away from the edge) of the OLED screen. The thickness of the border should be **1 pixel** and should appear automatically when this task starts.

4.A2 If the user presses the **centre pushbutton** an additional border should appear within the red border. The thickness of the border should be **3 pixels** and it should be **approximately orange** in colour.

4.A3 When the **orange border** of **3 pixels** is present, the following must happen automatically:

(i) Wait 2.0 seconds. A **1-pixel thick green border** appears within the orange border.

(ii) Wait another 1.5 seconds. A **2-pixels thick green border** appears within **4.A3(i)**.

(iii) Wait another 1.0 second. A **3-pixels thick green border** appears within **4.A3(ii)**.

(iv) Wait another 1.0 second. The **green borders** created in **4.A3(i)**, **4.A3(ii)** and **4.A3(iii) disappears**.

(v) **Anytime during 4.A3**, if the user presses and releases the **up pushbutton**, the following happen:

- If there is no **red solid square** near the centre of the screen, a **red solid square** appears.
- If there is a **red solid square** near the centre of the screen, it disappears.

This is independent of whether the borders described in **4.A3(i)** to **4.A3(iii)** are present or not.

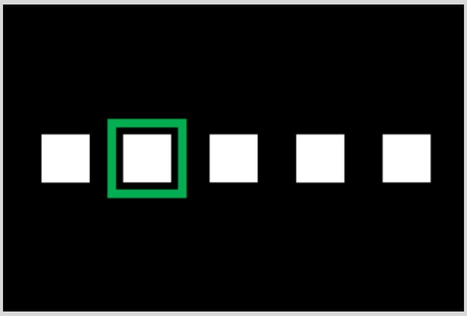
(vi) Steps **4.A3(i)** to **4.A3(vi)** repeats non-stop.

Note 1: Assume that the user never presses the pushbutton for more than 50 milliseconds. The pushbutton signal detection must be ***very responsive*** *(Example: Detection done every 1 millisecond) and* ***must apply debouncing*** *to avoid multiple presses detection. For debouncing, create a counter that* ***counts for 100 milliseconds*** *at the start of the pushbutton pressing, and* ***during that time, the pushbutton signal must be ignored.***

Note 2: The exact locations of the borders and distances between them are not assessed but should be reasonable.

4.B STUDENT B: BASIC TASK B

DESCRIPTION: To select and change the colour of a specific box. It would look approximately as:



4.B1 Create a **green square border** near the centre of the OLED screen. The thickness of the border should be **3 pixels** and should appear automatically when this task starts.

4.B2 After 3 seconds of waiting, 5 solid white squares of **8x8 pixels** must appear horizontally on the screen, of which the middle solid white square should be within the green border created in **4.B1**.

4.B3 The following can now happen:

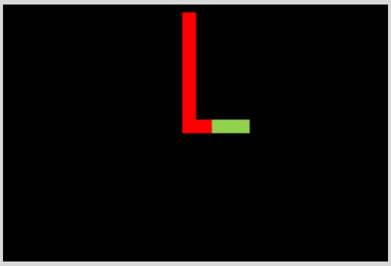
(i) If the user presses the **right pushbutton**, the green border moves to the adjacent square on the **right**.
(ii) If the user presses the **left** pushbutton, the green border moves to the adjacent square on the **left**.
(iii) The green border must not move any further to the left if it is already on the leftmost square.
(iv) The green border must not move any further to the right if it is already on the rightmost square.

Note 1: Assume that the user never presses the pushbutton for more than 50 milliseconds. The pushbutton signal detection must be ***very responsive*** *(Example: Detection done every 1 millisecond) and* ***must apply debouncing*** *to avoid multiple presses detection. For debouncing, create a counter that* ***counts for 100 milliseconds*** *at the start of the pushbutton pressing, and* ***during that time, the pushbutton signal must be ignored****.*

Note 2: The exact locations of the square boxes and distance between the boxes and border are not assessed but should be reasonable.

4.C STUDENT C: BASIC TASK C

DESCRIPTION: To create the character ‘L’ in an animated way on the OLED. It would look approximately as:



4.C1 Create a **red solid square** approximately near the top centre of the screen.
The square should be **3x3 pixel** and should appear automatically when this task starts.

4.C2 If the user presses and releases (Do not hold) the **centre pushbutton** the square should start moving downwards for around 30 pixels. It should take approximately 1.5 seconds to cover 30 pixels. While moving, it should leave behind a red trail **3 pixels** in **width**.

4.C3 After **4.C2**, the red square should stop its downward movement, and instead move to the right for around 15 pixels. It should take approximately 0.75 seconds to cover 15 pixels. A red trail **3 pixels** in **width** is still left behind, such that by the time the movement stops, the character “L” can be seen on the OLED in red colour. Then the following happens:

(i) 1 second after the movement has stopped, the **3x3 red solid square** changes to green colour.
(ii) After another 1 second, the green solid square moves in reverse, while leaving a green colour trail **3 pixels** in **width**. The reverse movement speed is the same as that of the forward speed described earlier. It should stop at the original location indicated in **4.C1**. At that stage, the character “L” can be seen on the OLED in green colour.
(iii) 1 second after the movement has stopped, the **3x3 green solid square** changes to red colour.
(iv) Steps **4.C2** and **4.C3** repeats when the user presses the **centre pushbutton** again.

*Note 1: Assume that the user never presses the pushbutton for more than 20 milliseconds. The pushbutton signal detection must be **very responsive** (Example: Detection done every 1 millisecond)*

Note 2: The movement animation should be reasonably smooth. It should not appear as if the square is skipping multiple pixels to move.

4.D STUDENT D: BASIC TASK D

DESCRIPTION: To create a moving object on the OLED. At the start, it would look approximately as:

4.D1 Create a **blue solid square** near the **top left corner** of the screen. The square should be **5x5 pixels** and should appear automatically when this task starts.

4.D2 If the user presses the **centre pushbutton**, the **blue solid square** should disappear from the top left corner of the screen and then re-appear **near the centre of the screen in green colour instead of blue**.

4.D3 The user can press and release (**Do not hold**) certain pushbuttons to make the square box move automatically in a certain direction till the edge. It stays stationary at the edge, until certain pushbuttons are pressed. The directions and speeds are as indicated below:

(i) **Right pushbutton** for **right** movement of the green square. The movement speed is approximately 45 pixels per second.

(ii) **Left pushbutton** for **left** movement of the green square. The movement speed is approximately 45 pixels per second.

(iii) The user can change the direction of the green square while it is moving, or when it is stationary at the edges. Assume, the **centre pushbutton** will not be pressed while the square box is moving.

The square box should automatically stop moving approximately near the edge of the screen. After it has stopped, the user can then press the **centre pushbutton** to move the green solid square back to the centre of the screen, and the square must stay stationary. Step **4.D3** can repeat again.

*Note 1: Assume that the user never presses the pushbutton for more than 20 milliseconds. The pushbutton signal detection must be **very responsive** (Example: Detection done every 1 millisecond)*

Note 2: The movement animation should be reasonably smooth. It should not appear as if the square is skipping multiple pixels to move.

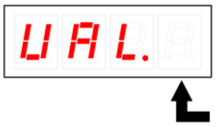
4.E GROUP TASK

DESCRIPTION: The original outputs of `paint.v` are to be used to create slightly modified outputs (**4.E1** and **4.E2**). The team will also need to work together to ensure that all their basic tasks and group task run in **ONE** single bitstream.

Integration is NOT straightforward, so reserve sufficient time for this very important part!

4.E1 Instead of having a solidly lit LED based on the digit detected by `paint.v`, the LED should blink at a frequency of 5 Hz.
*Note: If your team consists of **3 persons**, this part **4.E1** is not graded and can be skipped*

4.E2 Using the knowledge from Lab 3 assignment - Subtask C, improve the 7-segments display, such that the output follows this format:
- AN3, AN2 and AN1 are fixed and always show the fixed characters V A L.
  (Note that there is a decimal point for AN1)
- AN0 has variable values, as detected by `paint.v`



The digit detected by paint.v is shown on AN0

4.E3 Use the Basys3 switches to allow the user to choose which student's basic task, or group task, to run.

4.E4 When the student basic task is running, the **7-segment displays must not show anything**, while the LEDs should follow this rule:
- LD0 is ON when student A is demonstrating basic task A. All other LEDs are OFF.
- LD1 is ON when student B is demonstrating basic task B. All other LEDs are OFF.
- LD2 is ON when student C is demonstrating basic task C. All other LEDs are OFF.
- LD3 is ON when student D is demonstrating basic task D. All other LEDs are OFF.

4.E5 When one student's basic task, or group task, is running and currently being demonstrated, **none of the other tasks are allowed to run in the background**. In other words, running one task must not affect the other tasks in the background.

4.E6 Each time a student's basic task is selected to run, it **must always start from the beginning** (Either 4.A1, 4.B1, 4.C1 or 4.D1).

5. IMPROVEMENTS FOR AN ENHANCED MODS (E-MODS)

Using the skills learnt from the basic tasks of MODS, select a group improvement theme that your team want to implement on the Basys3 board, and divide the responsibilities among the team members. A good group theme for the E-MODS should be interactive, interesting, functional, and user-friendly. Evaluation consists of the individual improvements, as well as these criteria for the group theme:

(1) Functionality: Improvement(s) should provide useful function(s)
(2) Complexity: Complexity in implementation of the improvement(s).
(3) Quality: Whether the improvement(s) or user experience(s) is designed well or thorough.
(4) Creativity: Creativity or uniqueness in improvement(s) implementation(s).

It is required that the group theme is strongly related to the individual improvements, and not something different. A good individual improvement must include a reasonable amount of Verilog coding and should not have too much overlap with the other members of the team.

A description of the group theme your group has chosen, as well as the individual improvements need to be submitted to CANVAS by the end of week 9. The description should be no more than 1 page, and should make use of the template below:

Group ID: *This will be available in week 9*	**Description (Pictures / Drawings / Sketches are optional and not necessary)**
Group Theme: *Title of the group improvement theme*	*Describe the overall group theme that your team wants to implement on the Basys3. The description is recommended to be between 200 to 300 words. Minimum of 100 words required.*
Group Member A: *Student A who did Basic Task A*	*Describe the responsibilities / improvements of this group member with respect to the group theme. The recommended description should be between 100 to 200 words and can be in bullet points format. Minimum of 50 words required.*
Group Member B: *Student B who did Basic Task B*	*Describe the responsibilities / improvements of this group member with respect to the group theme. The recommended description should be between 100 to 200 words and can be in bullet points format. Minimum of 50 words required.*
Group Member C: *Student C who did Basic Task C*	*Describe the responsibilities / improvements of this group member with respect to the group theme. The recommended description should be between 100 to 200 words and can be in bullet points format. Minimum of 50 words required.*
Group Member D: *Student D who did Basic Task D*	*Describe the responsibilities / improvements of this group member with respect to the group theme. The recommended description should be between 100 to 200 words and can be in bullet points format. Minimum of 50 words required.*

Your team can still make reasonable changes to the actual implementations in week 10 and 11. There is no need to update the above document after submission in week 9.

If you need to connect other external devices to the Basys 3 development board, please seek the approval of your lab assistant before making the connection. You can also use / request for the additional devices listed below, but you will be required to implement them on your own. They will not be explained:

- PMOD for audio input
- PMOD for audio output
- Computer Keyboard
- VGA output to HDMI converter
- Connecting cables to join multiple Basys3 boards

6. PROJECT SUBMISSION INSTRUCTIONS

There are two items (ITEM A, and ITEM B) to submit to CANVAS:

Official Lab Session	ITEM A	ITEM B
	Project Archive	Report (Quick Start / User Guide)
Monday	**Deadline:** Saturday, 4th November 2023, 10:00 P.M.	**Deadline:** Monday, 6th November 2023, 10:00 P.M.
Tuesday		
Wednesday		
Thursday		
Friday		

ITEM A: Project Archive Submission

- Only **ONE** Vivado project archive submission **PER team**. **The archive should not exceed 200 MB in size.** (Excess of 200 MB in size is only allowed with prior approval from the lab assistant)

- All functions must be present in **one single bitstream**. Ensure that your **one single bitstream** has been successfully generated and tested on your Basys 3 development board **BEFORE** archiving your Vivado workspace for CANVAS upload. Download your CANVAS archive after uploading. Unzip it and check if you can run your **one single bitstream** correctly.

- The **one single bitstream** must show only the improvement parts. The basic tasks that have already been assessed in week 9 are not required in the bitstream. If you are not re-using parts from the basic tasks, they are best removed to release more memory on the Basys3 board.

ITEM B: Report Submission in PDF Format (One A4 Size Double-Sided Sheet Strongly Preferred)

- Include the group ID, members, and matriculation, on the top-left corner of the first page.

- The report serves as a quick start / user guide of what your system can do (Use the template provided on the next page):
    - Description of the features that you have designed and successfully implemented.
    - Instructions on how these features can be operated by the user.
    - Instructions may alternatively be described through flowcharts / state transition diagrams.

- Images must be clear and in colour.

- Short feedbacks are encouraged. Feedbacks, whether positive or negative, **DO NOT** have any effects on your grades. 😄
    - What did you like most / least about the project?
    - How would you suggest the overall project assignment be improved?
    - Any other constructive feedbacks / suggestions are welcomed.

- References
    - **Compulsory** to include references to **open-sources codes**.
    - Plagiarism penalties apply for **open-source codes that are not properly referenced**.

INSTRUCTIONS AND TEMPLATE FOR THE QUICK START / USER GUIDE (REPORT)

Column for Student and Improvement Name:

- Each student’s improvement needs to be indicated here, and marks for the improvement is awarded to that student.

Column for Improvement Description:

- Indicate how to use the improvement, and the expected effects that would be observed.
- You are encouraged to indicate the main or interesting capabilities in bold characters, but do not overuse bold characters. (Less than 20% of words for the description are allowed to be in bold)
- The instructions should be clear and should also indicate the input devices used.
  For example: SW0, SW[12:8], PBC, Character “X” on keyboard, Left mouse button.

Column for Images / Photos:

- Photos of your Basys3 / OLED screen to support your improvement. Short descriptions below the images are also allowed.
- Images that properly support your improvement description will demonstrate your efforts and help you explain better.
- You are encouraged to have as much distinct and clear images as possible, within the one A4-size double sided sheet, to describe your improvements. This helps to ensure all marks are properly awarded and accounted for.
- Pictures should show the parts being focused on. For example, do not show the whole Basys3 board if only the 7-segments values are the parts in focus.

PERSONAL AND TEAM IMPROVEMENTS		
Student and Improvement Name	**Improvement Description**	**Images / Photos**
Team “Name of group theme”	*Describe the group theme, indicating what the system can do overall. By reading this description, the user should know what the whole system is capable of and what are the star feature(s) of the system, if any. The group theme MUST be a superset of the individual improvements / components, and not something different*	*Take only one picture of the whole system, showing where the OLED (and other devices) are connected. The system should be running with your bitstream loaded, and the OLED screen / LED / Segments are showing something*
Student A: Alice “Name of Improvement” *One row per student (Preferred to be one row only). Two rows are only allowed if the student made two distinct/non-related improvements*	*Example: When BTNC is pressed, the image moves on the screen and the 7-segment shows …*	
Student B: Betty “Name of Improvement”		
Student C: Charles “Name of Improvement”		
Student D: Darwin “Name of Improvement”		

Colour Requirements to Ensure Accurate Grading:

- Highlight the background colour for **TEAM IN PASTEL (OR LIGHT) PINK** (Do not use dark / bright pink)
- Highlight the background colour for **STUDENT A IN PASTEL (OR LIGHT) RED** (Do not use dark / bright red)
- Highlight the background colour for **STUDENT B IN PASTEL (OR LIGHT) BLUE** (Do not use dark / bright blue)
- Highlight the background colour for **STUDENT C IN PASTEL (OR LIGHT) YELLOW** (Do not use dark / bright yellow)
- Highlight the background colour for **STUDENT D IN PASTEL (OR LIGHT) GREEN** (Do not use dark / bright green)

Bring **TWO printed copies** of your report in **COLOUR**, when you come for your assessment.
ONE A4 size double-sided sheet, with clear texts and images, is strongly preferred.