

EE2026: DIGITAL DESIGN

Academic Year 2025-2026, Semester 1

FPGA Design Project (FDP)

GROUPING

Teams that consist of students who have not already submitted ALL THREE assignments of the fundamental labs will not be automatically approved. Contact the lab instructors for a written approval before proceeding.

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 FPGA Design Project (FDP) that makes use of the Basys3 development board and the following devices will need to be created:

Device	Description
Computer Mouse (Optional) 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

Deadline extensions are not given for damaged components. Use them with utmost care!

This FPGA project requires student to apply their fundamental programming skills in higher-order thinking tasks. The week 7 lab session briefly guides you in getting started with a basic system consisting of the OLED. Exact steps are not provided, and you are expected to code, observe, and make conclusions!

After some basic tasks have been implemented, you will need to use your own technical skills, logic, creativity, and resourcefulness to create an Interactive FDP (I-FDP), 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.

- Each individual carries responsibility for **both** the team and individual portions of the project. Claiming for work done by, and not limited to, senior students, family members, relatives, paid or unpaid service providers, etc. are serious offences.
- All members of the team of the project solutions (codes) or reports will be **unconditionally penalised** in such cases.
- 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>
- Students will also be reported to BoD (Board of Discipline).

1. SCHEDULE AND PROJECT COMPONENTS

The FDP includes both **individual** and **teamwork** components:

Week	Suggested Task for Each Week	Marks	Evaluation
7	INDIVIDUAL BASIC TASK (4.P, 4.Q, 4.R, 4.S, 4.T)		All the basic tasks of the FDP are evaluated at the start of the week 8 lab session . CANVAS submission is required before your official week 8 lab session. (See section 4.T)
	Student P: Basic task 4.P (All Parts of 4.P)	4% (Individual)	
	Student Q: Basic task 4.Q (All Parts of 4.Q)	4% (Individual)	
	Student R: Basic task 4.R (All Parts of 4.R)	4% (Individual)	
	Student S: Basic task 4.S (All Parts of 4.S)	4% (Individual)	
8, 9, 10, 11 (Some weeks do not have labs)	I-FDP PROPOSAL DOCUMENT		CANVAS submission is required before your official week 9 lab session. (See Section 6)
	Group: I-FDP proposal - Indicate group theme, task responsibilities and timeline	2% (Group)	
	I-FDP IMPLEMENTATION		I-FDP is evaluated in week 12, 13 in ONE bitstream for the team. Multiple bitstreams are not allowed. CANVAS submission is required in week 12. (See Section 7) The basic tasks that have already been assessed in week 8 are not required in the bitstream and are not re-assessed.
	Student A: Component A of I-FDP	8% (Individual)	
	Student B: Component B of I-FDP	8% (Individual)	
	Student C: Component C of I-FDP	8% (Individual)	
	Student D: Component D of I-FDP	8% (Individual)	
	Group: Full integration of I-FDP components A, B, C and D in 1 bitstream	8 % (Group)	
WEEK 7 TO 11 PROJECT MARKS DISTRIBUTION – INDIVIDUAL: 12 %, GROUP: 10%			
12, 13	OVERALL EXECUTION, Q&A, UNDERSTANDING, COMPLETION OF PEER FEEDBACK	8 % (Individual)	Evaluated during presentation and through peer feedback

You are encouraged to plan a significant amount of your project time for integration!

Reminder: There is an in-lab Verilog evaluation (12%) in week 10. Details will be sent closer to that week.

2. SOFTWARE FILES

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

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.**

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.**

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. This module is optional, and its usage will not be explained.**

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. This module is optional, and its usage will not be explained.**

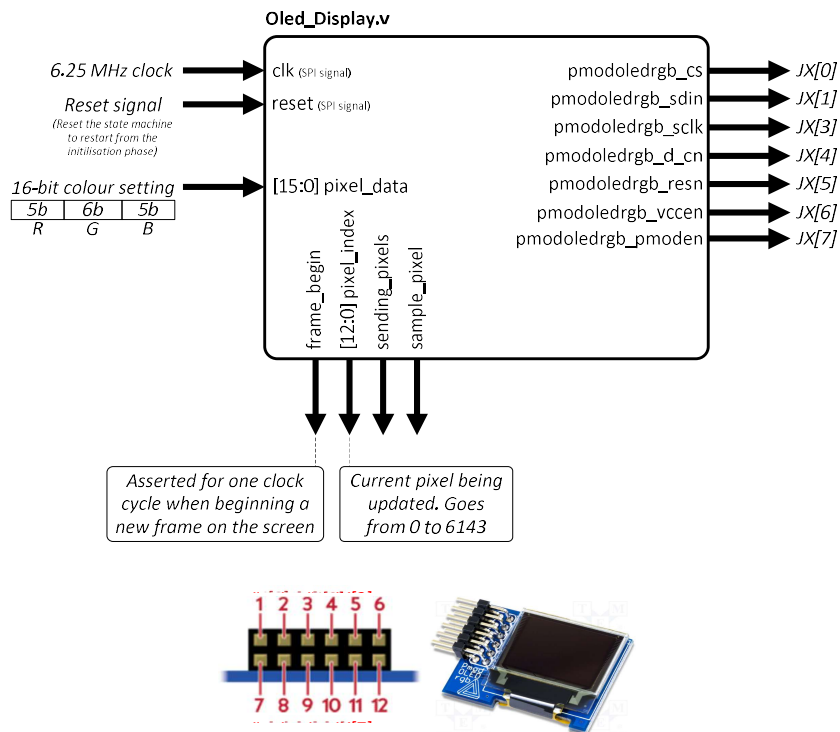
Constraint Sources:

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

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:

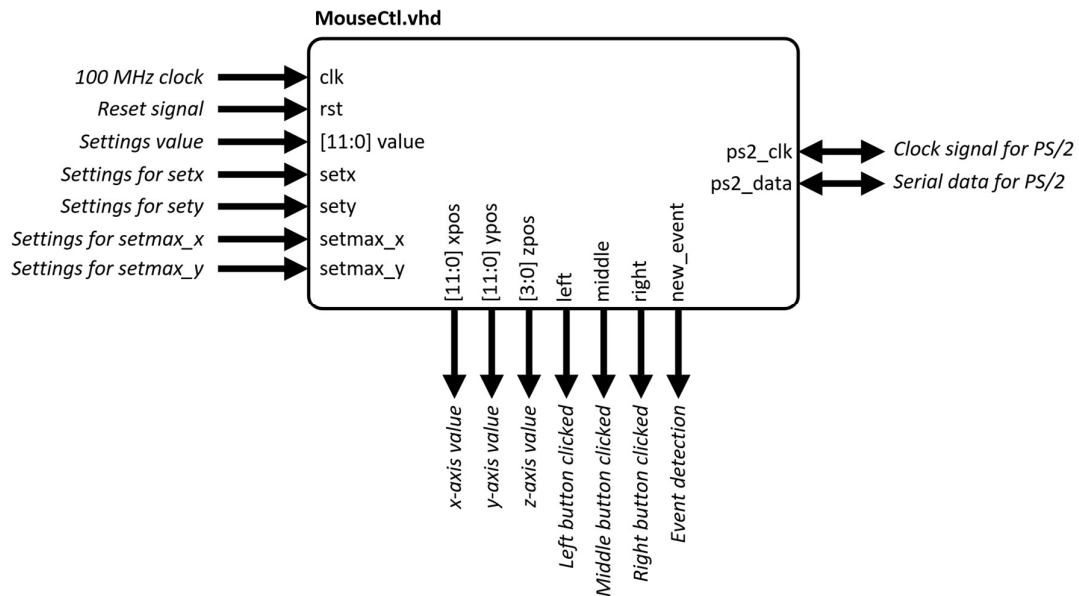
- `sw[4]` is OFF: The background colour is green.
- `sw[4]` is ON: The background colour is red.

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 [This optional (And not explained) component may be used in section 5: I-FDP]

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_y`.

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.

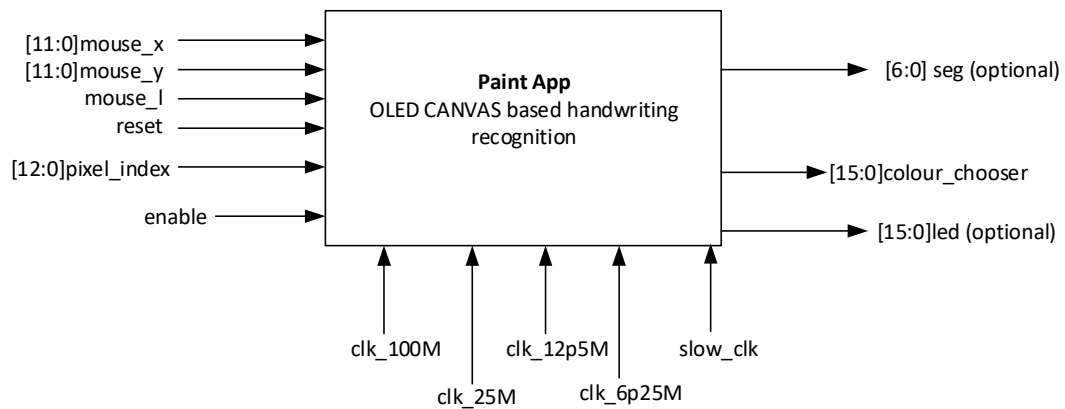
`led[13]` must turn ON when the signal from “`right`” port is high.

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 [This **optional** (And not explained) component may be used in section 5: I-FDP]

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



The Paint App module is a CANVAS drawing app with a built-in handwritten-digit recognition using the neural network algorithm. It contains 11 inputs, which are mouse_x, mouse_y, mouse_l, reset, pixel_index, clk_100M, clk_25M, clk_12p5M, clk_6p25M, slow_clk, and enable, and 3 outputs which are seg, colour_choser and led. Students to take the following procedure to properly instantiate the Paint App.

- 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 to mouse_x, mouse_y_pos to mouse_y, and mouse_left_click to mouse_l between MouseCtl.vhd and Paint.v ports.**
- 3.C4 Assign mouse_right_click to reset then connect it between between MouseCtl.vhd and Paint.v ports.**
- 3.C5 Connect Pixel Index to pixel_index, and colour_choser to pixel_data between Oled_Display.v and Paint.v.**
- 3.C6 Provide the clock signal with follow frequencies to Paint.v**
 - 100 MHz via clk_100M
 - 25 MHz via clk_25M
 - 12.5 MHz via clk_12p5M
 - 6.25 MHz via clk_6p25M
 - Between 0.5 to 2.0 Hz to slow_clk
- 3.C7 Connect LED to led and seven segment display to seg to Paint.v.**
- 3.C8 Check the following files are available under "Coefficient Files": 00.coe and weights.coe. If any of these files is unavailable, add these files via ".\MODS.srcs\sources_1\new" using add design sources wizards.**
- 3.C9 Click the "IP sources" under Source Tab (where you browse the Verilog file). Check the following memory blocks are available: blk_mem_gen_0, blk_mem_gen_const, blk_mem_gen_img and blk_mem_gen_inter. If any of the memory block is missing, add the *.dcp memory block file via ".\MODS.srcs\sources_1\ip" using add design sources wizards, and right click to choose "Generate Output Products"**
- 3.C10 Instantiate both mouse and OLED, 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. WEEK 7 BASIC TASKS – ASSESSED DURING WEEK 8 LAB SESSION

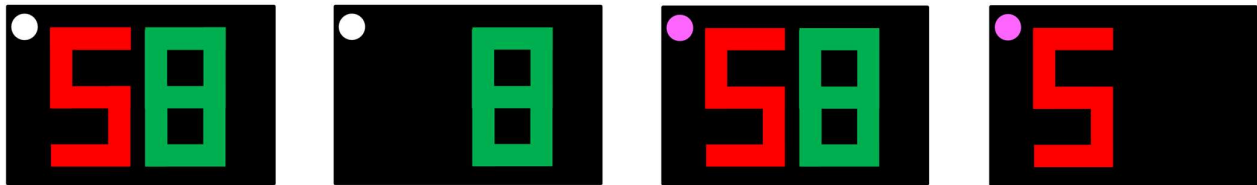
The basic tasks help students to get familiarised with the usage of the OLED screen. Several requirements are similar between the different basic tasks, so **team members can learn by sharing knowledge between themselves!**

If your team consists of 3 persons, ignore any one of the four individual basic tasks (4.P or 4.Q or 4.R or 4.S)

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.P STUDENT P: BASIC TASK P



Consider the **2nd Rightmost Number**, **1st Rightmost Number** of your student number:

7 th Rightmost Number	6 th Rightmost Number	5 th Rightmost Number	4 th Rightmost Number	3 rd Rightmost Number	2 nd Rightmost Number	1 st Rightmost Number	Rightmost Alphabet
A	0						

4.P1 When the program starts, the following appears on the black OLED screen:

- The **2nd Rightmost Number** appears on the left side of the OLED in **red colour**
- The **1st Rightmost Number** appears on the right side of the OLED in **green colour**
- A solid **white circle**, with radius **between 4 to 8 pixels**, appears near the top left of the OLED screen

Hint: Consider using the equation of a circle to draw the solid circle

Use the number template shown below as the requirement for a readable character for this basic task:

0 123456789	Height of each character: At least 45 pixels Width of each character: At least 25 pixels Lines thickness: At least 6 pixels thick
-------------	--

4.P2 Whenever the left pushbutton, or right pushbutton, is being pressed and held, the **solid white circle** on the top left corner must responsively (within 1 millisecond) change to a **solid magenta/pink circle**. Whenever none of the pushbuttons are being pressed, the solid circle responsively reverts to its original white colour.

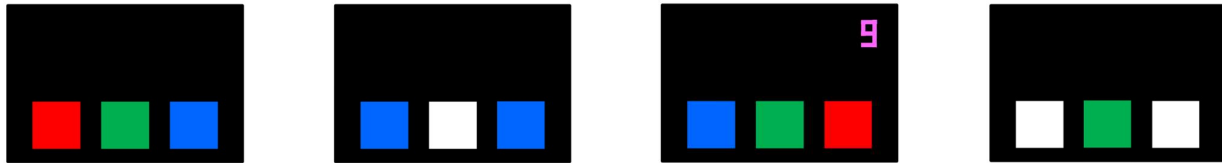
If the **left red character** is present on the OLED and the left pushbutton is pressed, that character should disappear from the OLED. If the left character is not on the OLED, and the left pushbutton is pressed, then it should reappear in **red** colour. This process can happen an infinite number of times.

If the **right green character** is present on the OLED and the right pushbutton is pressed, that character should disappear from the OLED. If the right character is not on the OLED, and the right pushbutton is pressed, then it should reappear in **green** colour. This process can happen an infinite number of times.

Very important note regarding the pushbutton presses controlling the left and right characters:

- The user may press and release the pushbutton within 200 milliseconds (No need to count the actual duration the user pressed the pushbutton, or if 200 milliseconds have been exceeded). The pushbutton signal detection must be **very responsive** to this quick pressing (Example: Detection done every 1 millisecond) and **must apply debouncing to avoid multiple presses detection**. For debouncing, create a counter that counts for 200 milliseconds at the start of the pushbutton pressing, and during that time, bouncing pushbutton signals must be ignored.
- The user may also press and hold for more than 200 milliseconds. In such case, **the pushbutton press should be detected as being just one press**, no matter how long the pushbutton is pressed and held. It may be held for as long as 5 seconds.

4.Q STUDENT Q: BASIC TASK Q



Consider the 3rd Rightmost Number, 2nd Rightmost Number, 1st Rightmost Number of your student number:

7 th Rightmost Number	6 th Rightmost Number	5 th Rightmost Number	4 th Rightmost Number	3 rd Rightmost Number	2 nd Rightmost Number	1 st Rightmost Number	Rightmost Alphabet
A	0						
Assign colours to each one of these 3 numbers →				Colour?	Colour?	Colour?	

Use this table to determine the colour to assign to each number

Number →	0, 5	1, 6	2, 7	3, 8	4, 9
Colour →	Red	Blue	Yellow	Green	White

4.Q1 When the program starts, the following appears on the black OLED screen:

- A **red 20x20 pixels solid square** on the bottom left side of the OLED
- A **green 20x20 pixels solid square** on the bottom middle of the OLED
- A **blue 20x20 pixels solid square** on the bottom right side of the OLED

Each one of the three solid squares can change colour independently. Each time a certain pushbutton is pressed, the colour for the solid square changes in this sequence (one colour change per pushbutton press):

Red > Blue > Yellow > Green > White > “Back to Red” (Sequence repeats)

The solid square colours are controlled by:

- **Left pushbutton** changes the colour of the leftmost solid square
- **Centre pushbutton** changes the colour of the middle solid square
- **Right pushbutton** changes the colour of the rightmost solid square

4.Q2 When the three solid squares on the screen matches exactly the colours assigned at the start of this task, a small sized number representing the 4th Rightmost Number of your student number must appear on the top right corner of the OLED screen, in **magenta/pink colour**. (In the pictures at the start of this task, we have assumed that the three numbers of the student are **6-3-5 (Blue-Green-Red)**, and the 4th rightmost number is **9**)

Use the number template shown below as the requirement for a readable character for this basic task:

0 123456789	Height of each character: Between 10 to 15 pixels Width of each character: Between 5 to 8 pixels Lines thickness: Exactly 2 pixels thick
-------------	---

If the user continues to change the colours of the three solid squares and they do not match the assigned colours, the small sized number on the top right corner of the OLED screen must disappear.

The process of the small sized number appearing and disappearing on the top right corner of the OLED, due to changes in the colour of the solid squares, can happen an infinite number of times.

Very important note regarding the pushbutton presses: The user may press and release the pushbutton within 200 milliseconds (No need to count the actual duration the user pressed the pushbutton, or if 200 milliseconds have been exceeded). The pushbutton signal detection must be **very responsive** to this quick pressing (Example: Detection done every 1 millisecond) and **must apply debouncing to avoid multiple presses detection**. For debouncing, create a counter that counts for 200 milliseconds at the start of the pushbutton pressing, and during that time, bouncing pushbutton signals must be ignored.

The user may also press and hold for more than 200 milliseconds. In such case, **the pushbutton press should be detected as being just one press**, no matter how long the pushbutton is pressed and held. It may be held for as long as 5 seconds.

4.R STUDENT R: BASIC TASK R



Arrows are not present on the OLED. They are only for illustration purposes of the movement directions.

Consider the **3rd Rightmost Number**, **1st Rightmost Number** of your student number:

	7 th Rightmost Number	6 th Rightmost Number	5 th Rightmost Number	4 th Rightmost Number	3 rd Rightmost Number	2 nd Rightmost Number	1 st Rightmost Number	Rightmost Alphabet
A	0							

4.R1 When the program starts, the following appears on the black OLED screen:

- The **1st Rightmost Number** appears on the left side of the OLED in **blue colour**
To improve visibility of the number on the OLED, use a lighter blue colour!
- The **3rd Rightmost Number** appears on the top side of the OLED in **orange colour**

Use the number template shown below as the requirement for a readable character for this basic task:

0 123456789	Height of each character: Between 20 to 30 pixels Width of each character: Between 15 to 20 pixels Lines thickness: Exactly 4 or 5 pixels thick
--------------------	--

4.R2 When the user turns ON switch **SW1**, the **blue number** must start moving, such that it moves between the right edge and left edge of the OLED. The movement speed must be such that it takes around **3 ± 0.5 seconds** to move from one edge to the opposing edge. For ease of programming, there is no need to be exactly at the edge (It can be within 3 pixels from the edges)

If **SW1** is turned OFF, the movement of this **blue number** must stop, and if **SW1** is turned ON, it resumes in the same direction that it was going before it stopped. **SW1** can be turned ON / OFF an infinite number of times.

When the user turns ON switch **SW3**, the **orange number** must start moving, such that it moves between the top edge and bottom edge of the OLED. The movement speed must be such that it takes around **5 ± 0.5 seconds** to move from one edge to the opposing edge. For ease of programming, there is no need to be exactly at the edge (It can be within 3 pixels from the edges)

If **SW3** is turned OFF, the movement of this **orange number** must stop, and if **SW3** is turned ON, it resumes in the same direction that it was going before it stopped. **SW3** can be turned ON / OFF an infinite number of times.

Important note regarding the number movement: The **movement animation should be smooth**. It should not appear as if the moving number is skipping pixels or teleporting itself to move. There is also no restriction on which number is in the foreground, and which number in the background when they cross path near the middle of the screen.

4.S STUDENT S: BASIC TASK S



Arrows are not present on the OLED. They are only for illustration purposes of the movement directions.

Consider the **2nd Rightmost Number** of your student number:

	7 th Rightmost Number	6 th Rightmost Number	5 th Rightmost Number	4 th Rightmost Number	3 rd Rightmost Number	2 nd Rightmost Number	1 st Rightmost Number	Rightmost Alphabet
A	0							

4.S1 When the program starts, the following appears on the black OLED screen:

- A **solid red circle**, with radius **between 9 to 15 pixels**, appears near the centre of the OLED screen
Hint: Consider using the equation of a circle to draw the solid circle
- A **hollow square** appears near the bottom left of the OLED screen in **white colour**
Each side of the square is **between 20 to 28 pixels** in length
The thickness of each side is **exactly 3 pixels** thick
- The **2nd Rightmost Number** appears within the hollow square in **green colour**
To improve visibility of the number on the OLED, use a brighter or lighter green colour!

Use the number template shown below as the requirement for a readable character for this basic task:

0 123456789	Height of each character: Between 12 to 16 pixels Width of each character: Between 6 to 9 pixels Lines thickness: Exactly 3 pixels thick
--------------------	---

4.S2 The user can press (**Do not hold**) certain pushbuttons to make the **solid circle** move automatically (No need to keep pressing or holding the pushbutton) in a certain direction till near the OLED screen edge, or till it hits the sides of the hollow square. It stays stationary at any sides of the hollow square and near the edges of the OLED screen. The movement speed for the **solid circle** is approximately **35 to 55 pixels per second**.

The directions for the solid circle are as indicated below (Pushbuttons are pressed and released immediately):

- **Right pushbutton** for continuous /automatic **right** movement of the solid circle.
- **Left pushbutton** for continuous / automatic **left** movement of the solid circle.
- **Up pushbutton** for continuous / automatic **up** movement of the solid circle.
- **Down pushbutton** for continuous / automatic **down** movement of the solid circle.

The solid circle can change directions while it is moving, but movements cannot be diagonal. For example, if the solid circle is at the bottom left, and the user keeps alternating between **up** movement and **right** movement, the solid circle will move in a staircase-like pattern.

Four points of contact (For collision detection between the **solid circle** and hollow square) are sufficient on the **solid circle**: One on top, one at the bottom, one on the left and one on the right of the **solid circle**. As there are only four points of contacts, it is acceptable if a small part of the **solid circle** overlaps with the hollow square.

Note regarding the pushbutton presses: The user will press and release the pushbutton within 100 milliseconds (No need to count the actual amount of time the user pressed the pushbutton, or if 100 milliseconds have been exceeded). The **pushbutton signal detection must be very responsive to this quick pressing** (Example: Detection done every 1 millisecond). Debouncing is optional here as it does not matter if a pushbutton press is detected as multiple presses.

Important note regarding the number movement: The **movement animation should be smooth**. It should not appear as if the moving number is skipping pixels or teleporting itself to move.

4.T GRADING OF THE BASIC TASKS IN WEEK 8

Reserve ample time to work on integration. Significant time is required if you have not decided with your team members on a common coding format to use before starting on your tasks.

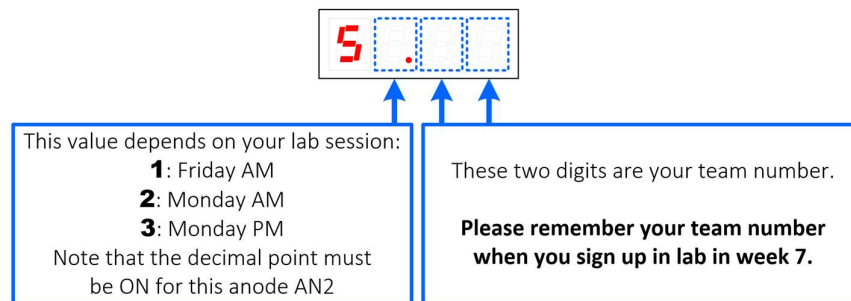
The basic tasks completed by the team members are required to be integrated into **ONE single bitstream**. Team members whose basic task are integrated within that **ONE single bitstream** can obtain up to the full marks of 4% assigned to the basic task component.

To emphasise on the importance of planning for integration, if any of the following scenario occurs, a penalty of up to 50% will apply on the marks obtained for that specific student (Other team members are not affected):

- Student's basic task was not integrated within that **ONE single bitstream**
- Student's basic task was **shown later or shown separately** from the team's presentation during the lab session

If the team was not able to unanimously decide on **ONE single bitstream** to use for the presentation, the penalty will apply to all members within the team.

For the **ONE single bitstream**, when the program starts, the seven segments displays must show your group ID in this format (It is compulsory for all students to know how to get different characters to appear simultaneously on the seven segments displays):



The basic task to present should be based on the following truth table:

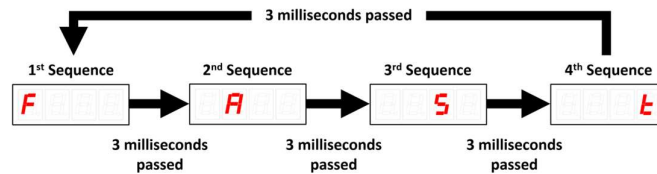
INPUTS				OUTPUTS
SW15	SW14	SW13	SW12	OLED
0	0	0	1	Basic Task P for Student P (If present in bitstream)
0	0	1	X	Basic Task Q for Student Q (If present in bitstream)
0	1	X	X	Basic Task R for Student R (If present in bitstream)
1	X	X	X	Basic Task S for Student S (If present in bitstream)

You will be allowed to re-program the **same ONE single bitstream** to the FPGA to show each student's basic tasks from beginning.

Hint: To be able to show different characters on the seven-segments displays, it is required to:

- Show a character on only 1 anode at a time
- Show another character on another anode a few milliseconds later (around 200 to 500 Hz)
- Repeat the above continuously

An example of the above task is illustrated below:



For humans, this is what they will see:



UPLOAD DETAILS FOR WEEK 7 BASIC TASKS:

- One **single archive** (.xpr.zip) **per group**
- File name format: **Week_7_SX_YZ_Group**, where **SX_YZ** is your group ID (See PDF on CANVAS near submission time for your **SX_YZ**).
- Upload in the CANVAS folder named: **Week 7 FDP Submission**
- Deadline for submission:
 - **Monday Sessions:** Upload to be completed by **Monday 6th October 2025, 6 AM (Morning)**
 - **Friday Sessions:** Upload to be completed the **Friday 10th October 2025, 6 AM (Morning)**
- The team must **submit one hardcopy** of the **completed** "Personalised Basic Tasks Form" during the lab session in week 8. The form is found on the next page of this lab manual.
- Late assessments of week 7 basic tasks are not accepted after week 9 of the semester

Submit one hardcopy of the completed “Personalised Basic Tasks Form” during the lab session in week 8:

PERSONALISED BASIC TASKS FORM:

Each student has a personalised basic task requirement based on the student matriculation number. Carefully write down your number, as mistakes in the student matriculation number used are not accepted.

Team ID (Format is SX_YZ. For Example, S4_07) →	
---	--

BASIC TASK P	Name of <Student P> as written on the Student Matriculation Card →							
	7 th Rightmost Number	6 th Rightmost Number	5 th Rightmost Number	4 th Rightmost Number	3 rd Rightmost Number	2 nd Rightmost Number	1 st Rightmost Number	Rightmost Alphabet
	A	0						

BASIC TASK Q	Name of <Student Q> as written on the Student Matriculation Card →							
	7 th Rightmost Number	6 th Rightmost Number	5 th Rightmost Number	4 th Rightmost Number	3 rd Rightmost Number	2 nd Rightmost Number	1 st Rightmost Number	Rightmost Alphabet
	A	0						
	Assign colours to each one of these 3 numbers →							

BASIC TASK R	Name of <Student R> as written on the Student Matriculation Card →							
	7 th Rightmost Number	6 th Rightmost Number	5 th Rightmost Number	4 th Rightmost Number	3 rd Rightmost Number	2 nd Rightmost Number	1 st Rightmost Number	Rightmost Alphabet
	A	0						

BASIC TASK S	Name of <Student S> as written on the Student Matriculation Card →							
	7 th Rightmost Number	6 th Rightmost Number	5 th Rightmost Number	4 th Rightmost Number	3 rd Rightmost Number	2 nd Rightmost Number	1 st Rightmost Number	Rightmost Alphabet
	A	0						

FOR GRADER'S USE:

I confirm that the student matriculation numbers match the official NUS records.

(Initials of grader: _____)

5. WEEK 8 TO WEEK 11: THE INTERACTIVE FDP (I-FDP)

An Interactive FPGA Design Project (I-FDP) is required to be implemented **within the resource limitations of the FPGA** on the Basys3 board, in **ONE** bitstream. Working together as a group, divide the responsibilities among the team members to achieve the following specifications for the I-FDP:

Group Theme	FPGA Based Smart Graphical Calculator using Verilog Language
Functionality	Compulsory requirement: Able to perform simple additions and subtractions of whole numbers (with single digits for operands, such as 8 + 5). Possible enhancements (non-exhaustive): More digits for the operands, multiplication, division, decimal points, different number representations, trigonometry, graphing, conversions, educational components etc.
Robustness and Reliability	Compulsory requirement: Reprogramming the FPGA is not required for each calculation. System does not crash often due to unexpected behaviours. Possible enhancements (non-exhaustive): Switching between different functionalities always work, reset in-built, reliable communication between multiple Basys3 board if there are multiple boards etc.
User Interface	Compulsory requirement: Usage of switches, LEDs and OLED. Possible enhancements (non-exhaustive): Usage of a seven-segments display, multiple OLEDs, mouse, computer keyboard etc. <i>Note: Usage of unnecessary devices is discouraged (It negatively affects the product)</i>
User Experience	Compulsory requirement: User can input values, and see output values, on the Basys3 board, with ease. Possible enhancements (non-exhaustive): Intuitive usage of the calculator with minimal documentation, adequate on-board guidance where required, brightness/mode of the OLED, elements of fun and excitement when using the product, etc.

Compulsory requirements are the bare minimum required to show your fundamental Verilog capabilities.

Possible enhancements are value-added components that further demonstrate your Verilog capabilities. The enhancement(s) must be strongly related to the group theme. Each team is encouraged to work together on a larger enhancement by appropriately dividing the responsibilities, instead of multiple unrelated / simpler enhancements, for the I-FDP.

Each team member must have a reasonable amount of Verilog coding to do, and their allocated responsibility should not have much overlap with the other members of the team. These individual responsibilities would form the marks for the individual marks of the I-FDP implementation.

The group marks of the I-FDP implementation consider how well the different components from each team member are integrated together into one bitstream to create a more complete project. It must not be a separate component.

ALTERNATIVE GROUP THEME OPTION FOR I-FDP:

To allow teams to further explore other options that they wish to implement using the Basys3 and Verilog as the building block, it is allowed to choose another group theme that matches their interest, instead of the FPGA Based Smart Graphical Calculator described above. In such cases, please note that the same rules apply:

- The individual I-FDP implementation should be strongly related to the group I-FDP implementation, and not something different / unrelated. A project that focuses on one main theme with deeper integration would have higher value, than a project that has many smaller unrelated components and lower integration complexity.
- The same criteria will be used to judge your chosen I-FDP, namely:
Functionality, Robustness and reliability, User Interface and User Experience

Tip: Do not focus only on implementing games! There are many more interesting things that can be implemented for the Smart Graphical Calculator or I-FDP. Spend some time with your team members for some creative thinking and surprise the graders during the presentation!

OTHER OPTIONAL DEVICES (None of these are needed for a very successful project):

If you need to connect other external devices to the Basys3 development board, please seek **explicit email approval** from your lab assistant **before** making the connection. Connections between multiple Basys3 boards require you to **provide a schematic of which ports will be connected via wires** in the approval request email. **Do not request for Basys3 boards connections without providing the exact port connections schematics.** Also, wires will need to be connected to appropriate headers. Do not use wires without headers.

You can also use the additional devices listed below, but you will be required to implement them on your own. You can only request to sign out for these devices **during EE2026 lab sessions** (The lab counter does not distribute EE2026 components outside EE2026 lab sessions hours), and they need to be returned to the lab at the end of the semester. **These external devices will not be explained, and design sources will not be provided:**

- PMOD for audio input
- PMOD for audio output
- VGA output to HDMI converter

Additional marks are not given for using additional hardware components. Hardware is not assessed in EE2026.

If you are not re-using parts from the basic tasks in I-FDP, consider removing them to release memory on the Basys3 board.

On average, the demonstration of the I-FDP in week 12, 13 should have sufficient components for a duration of 2 to 5 minutes for the team, excluding questions asked by the grader. Questions by graders can extend the duration to more than 10 minutes.

6. I-FDP PROPOSAL DOCUMENT (UPLOAD TO CANVAS AT THE START OF WEEK 9)

A description of the group theme your group has chosen needs to be **submitted to CANVAS**. The description should be no more than 2 pages (1 sheet), and should make use of the template below:

I-FDP PROPOSAL DOCUMENT	
Group ID: SX_YZ	Description (Pictures / Drawings / Sketches / Gantt Charts are optional and not necessary)
Group Theme: <i>Title of the group theme</i>	<i>Describe the overall group theme that your team wants to implement on the Basys3. If you are choosing the default FPGA Based calculator, this would involve indicating the overall functions of your calculator. The description for this part is recommended to be between 200 to 400 words. Minimum of 100 words required.</i>
Group Member 1: <i>Name of group member</i>	<i>Member responsibility: "Describe what this group member is required to do with respect to the group theme"</i> <i>Week 8 tasks: "Describe what the student has already done for this week"</i> <i>Week 9 tasks: "Describe what the student will be doing for this week"</i> <i>Week 10 tasks: "Describe what the student will be doing for this week"</i> <i>Week 11 tasks: "Describe what the student will be doing for this week"</i> <i>The description for this part is recommended to be between 200 to 400 words and can be in bullet points format. Minimum of 100 words required.</i>
Group Member 2: <i>Name of group member</i>	<i>Member responsibility: "Describe what this group member is required to do with respect to the group theme"</i> <i>Week 8 tasks: "Describe what the student has already done for this week"</i> <i>Week 9 tasks: "Describe what the student will be doing for this week"</i> <i>Week 10 tasks: "Describe what the student will be doing for this week"</i> <i>Week 11 tasks: "Describe what the student will be doing for this week"</i> <i>The description for this part is recommended to be between 200 to 400 words and can be in bullet points format. Minimum of 100 words required.</i>
... More members	<i>... More members</i>
Group Integration:	<i>Week 9 integration: "Describe what parts will be integrated for this week"</i> <i>Week 10 integration: "Describe what parts will be integrated for this week"</i> <i>Week 11 integration: "Describe what parts will be integrated for this week"</i> <i>The description for this part is recommended to be between 100 to 200 words and can be in bullet points format. Minimum of 50 words required.</i>

Your team can still make reasonable changes to the actual implementations in week 8, 9, 10 and 11. There is no need to update the above document after submission.

UPLOAD DETAILS FOR WEEK 8:

- One **single PDF** submission **per group**
- File name format: **Week_8_SX_YZ**, where **SX_YZ** is your group ID (See PDF on CANVAS near submission time).
- Upload in the CANVAS folder named: **Week 8 I-FDP Proposal Submission**
- Deadline for submission: **Monday 13th October 2025, 6 AM (Morning)**

7. FINAL I-FDP SUBMISSION INSTRUCTIONS – ASSESSED IN WEEK 12, 13

There are two items (ITEM A, and ITEM B) to submit to CANVAS, and ITEM C (More Details in Week 11) to complete on CANVAS:

Official Lab Session	ITEM A (Submit on CANVAS)	ITEM B (Submit on CANVAS)	ITEM C (Complete on CANVAS)
	Project Archive	Report (User Guide)	Peer Feedback
Monday	Deadline: Tuesday 4 th November 2025, 6:00 A.M. (Morning)	Deadline: Thursday, 6 th November 2025, 6:00 A.M. (Morning)	Deadline: Tuesday, 11 th November 2025, 6:00 A.M. (Morning)
Tuesday			
Wednesday			
Thursday			
Friday			

Deadlines cannot be extended. Plan early and responsibly on how to use the 4 weeks allocated for the I-FDP.

ITEM A: Project Archive Submission

- Only **ONE** Vivado project archive submission **PER team**. The archive should not exceed **400 MB in size**. (Excess of 400 MB in size is only allowed with prior approval from the lab instructor)
- 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 I-FDP. The basic tasks that have already been assessed in week 8 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 (Two A4 double-sided sheet):

Sheet 1 (Page 1 and 2) - User guide part:

- Include the group ID, members, and matriculation, on the top of the first page.
- A quick start / user guide of what your system can do (Use the template provided on the next page):
 - Description of the components that you have designed and successfully implemented.
 - Instructions on how these components can be operated by the user.
 - Instructions may alternatively be described through flowcharts / state transition diagrams.
- Images must be clear and in colour.
- All parts of the quick start / user guide must be on this 1st sheet of A4 paper (2 pages)

Sheet 2 (Page 3 and optionally 4) - References and feedback part:

- The 2nd sheet of A4 paper must not contain any part of the quick start / user guide.
- Include the group ID, members, and matriculation, on the top of the third page.
- **Include references used.** (GitHub, Websites, YouTube, Chatbots, AI, etc...)
 - **Compulsory** to include references to **open-sources codes**.
 - Usage of **Artificial Intelligence (AI)** or **automated code generators** must be **clearly indicated and documented**.
 - Full penalties apply for the project if **open-source codes / AI / code generators / chatbots** have been used but **not properly documented on how they have been used in the project**.
- Include **course related feedback** (Ideally, no feedback about teachers in the report. It must be about the course):
 - Feedback, whether positive or negative, **DO NOT** have any effects on your grades.
 - Negative feedback with constructive comments is encouraged in this report, as they will allow two-way communications between students / teachers during the project presentation, to understand what to improve.
 - Anonymous feedback of teachers, or course feedback that don't need two-way communications, can be given on the official NUS end-of-semester feedback.

INSTRUCTIONS AND TEMPLATE FOR THE QUICK START / USER GUIDE

Column for Student and Component Name:


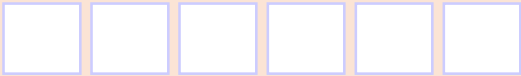

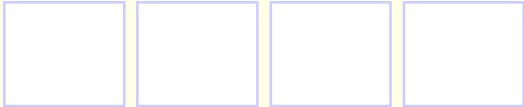
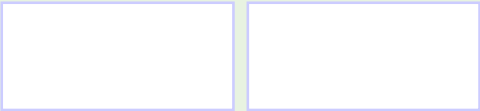
- Each student's component needs to be indicated here, and marks for the component is awarded to that student.

Column for Component Description:

- Indicate how to use the component, and the expected effects that would be observed.
- 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 component. Short descriptions below the images are also allowed.
- Images that properly support your component 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 components. 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.

Student and Component Name	Component Description	Images / Photos
Team "Name of group theme"	<i>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 component(s) of the system, if any. The group theme MUST be a superset of the individual components, and not something different</i>	 <i>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</i>
Student A: Alice "Name of Component" <i>One row per student (Preferred to be one row only). Two rows are only allowed if the student made two distinct/non-related components</i>	<i>Example: When BTNC is pressed, the image moves on the screen and the 7-segment shows ...</i>	
Student B: Betty "Name of Component"		
Student C: Charles "Name of Component"		
Student D: Darwin "Name of Component"		

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** (There are up to two examiners per team: One copy for each examiner) of your report in **COLOUR**, when you come for your assessment.