# EE2026: DIGITAL DESIGN

## Academic Year 2025-2026, Semester 1

## LAB 3: Sequential Circuits in Verilog

## OVERVIEW

A sequential circuit is one where the outputs depend on the current inputs and the sequence of past inputs. As a result, a sequential circuit has memory, also called states. In this lab, some basic sequential circuits will be designed to make an LED blink at various speeds.

**The pre-requisites for this lab are:**

- A very good understanding and application of dataflow modelling and structural modelling in designing modules.
- Knowing how to use the Vivado IDE well.
- Familiarity and knowledge on how to use "*Set as Top*", "*reg*" and "*wire*".

**This lab will cover the following:**

- Using a signal that inverts itself periodically, which shall be called **CLOCK**.
- Making a physical LED blink by using the Basys 3 development board clock signal.

**Tasks for this lab include:**

- Creating a slower clock from a faster clock.
- Having a physical LED blink noticeably on the Basys 3 development board, by using the slower clock.
- Using switches to make a physical LED blink at different speeds on the Basys 3 development board.

## GRADED ASSIGNMENT [CANVAS SUBMISSION: THURSDAY 25th SEPTEMBER 2025, 6 A.M.]:

Further details are available at the end of this lab manual.

## THE BLINKING LED

A simple blinking LED is required to be implemented on the FPGA. To do this, a new signal, **CLOCK**, will be introduced.

The **CLOCK** signal is an external input signal that resembles a square wave of 50% duty cycle. If this **CLOCK** signal is connected directly to a physical LED, the latter will light up when the signal is HIGH, and will switch off when the signal is LOW, as illustrated in **Figure 3.1**.

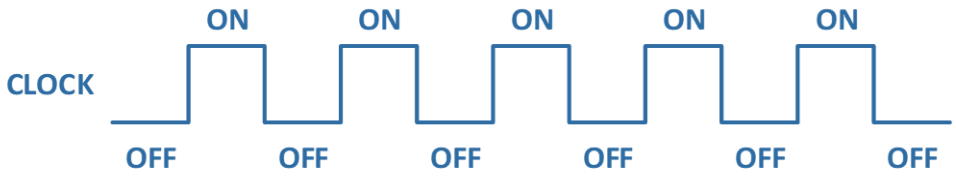



***Figure 3.1:*** *A **CLOCK** signal with 50% duty cycle*

A simple dataflow description in Verilog for a blinky module is written first, followed by a simulation source to verify the design. To create the square wave, or **CLOCK** signal, in the simulation source, a new section of codes will now be introduced:

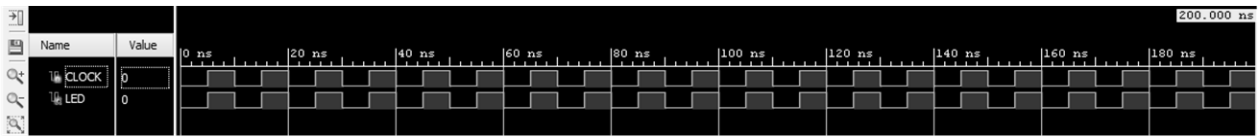
**Verilog code for blinky, using the dataflow method**

```
module blinky (input CLOCK, output LED);

    assign LED = CLOCK;

endmodule
```

**Simulation source code to test the blinky design**

```
`timescale 1ns / 1ps

module test_blinky( );

    reg CLOCK; wire LED;

    blinky dut (CLOCK, LED);

    initial begin
        CLOCK = 0;
    end

   always begin
       #5; CLOCK = ~CLOCK;
   end

endmodule
```

**Expected simulation waveform for the blinky design**




## UNDERSTANDING | TASK 1

Based on the Verilog code and simulation results, check your understanding by answering the following questions:

1. Every 5 units of time, the value of **CLOCK** is being inverted. What is the clock frequency being used in this simulation?

   _____

2. What would happen if the testbench code **CLOCK = 0** is removed?

   _____

For the hardware implementation, instead of using an external signal generator for the **CLOCK** signal to the Artix-7 FPGA, the Basys 3 development board includes a single 100 MHz clock generator connected to pin W5 of the Artix-7 FPGA.

Using the original contents of the ***Basys3_Master.xdc*** in your constraint file, follow these steps:

1. Uncomment lines 7 and 8 to use the 100 MHz on the Basys3 development board. If required, rename the signal to the name used in your **blinky** code. In our example, the name **CLOCK** was used, and the final changes may look similar to **Figure 3.2**.

2. Configure the output signal **LED** that is present in your **blinky** code (or the name chosen by you while writing the code) by linking it to any physical LED on the Basys3 development board.

```
1  ## This file is a general .xdc for the Basys3 rev B board
2  ## To use it in a project:
3  ## - uncomment the lines corresponding to used pins
4  ## - rename the used ports (in each line, after get_ports) according to the top level signal names in the project
5
6  # Clock signal
7  set_property -dict { PACKAGE_PIN W5   IOSTANDARD LVCMOS33 } [get_ports CLOCK]
8  create_clock -add -name sys_clk_pin -period 10.00 -waveform {0 5} [get_ports CLOCK]
9
10
11 ## Switches
12 #set_property -dict { PACKAGE_PIN V17   IOSTANDARD LVCMOS33 } [get_ports {sw[0]}]
13 #set_property -dict { PACKAGE_PIN V16   IOSTANDARD LVCMOS33 } [get_ports {sw[1]}]
```

***Figure 3.2:*** *Modifying your constraint file, based on the contents of the Basys3_Master.xdc*

## UNDERSTANDING | TASK 2

You may optionally generate the bitstream and download your code to the Basys3 development board. What do you "*notice*" about the "*blinking*" LED?

_____

_____

## COUNTER FOR A SLOWER CLOCK

To be able to observe a blinking LED at a frequency that is visible to the human eyes, modifications need to be done to the Verilog code to have a slower clock. Let us introduce a temporary variable **COUNT** that is incremented by 1 at every rising edge (transition from low to high; also called a positive edge) of the **CLOCK** signal, as shown in **Figure 3.3**.

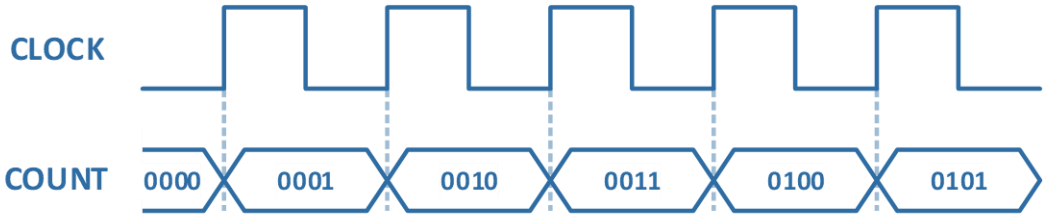



***Figure 3.3:*** *Increasing* ***COUNT*** *at each rising edge of* ***CLOCK***

**Verilog code for an incrementing counter of 4-bit, using behavioural modelling**

```
module slow_blinky_module (input CLOCK);

    reg [3:0] COUNT = 4'b0000;

    always @ (posedge CLOCK) begin
        COUNT <= COUNT + 1;
    end

endmodule
```

Now, use a conditional statement that changes the state of an output signal, called SLOW_CLOCK, whenever COUNT reaches zero.

**Partial Verilog code for toggling SLOW_CLOCK signal, using behavioural modelling**

```
    always @ (posedge CLOCK) begin
        COUNT <= COUNT + 1;
        SLOW_CLOCK <= ( COUNT == 4'b0000 ) ? ~SLOW_CLOCK : SLOW_CLOCK ;
    end
```

Here, since the 4-bits COUNT has 16 possible states, it would mean that SLOW_CLOCK changes state after 16 clock cycles of CLOCK. In other words, SLOW_CLOCK is 1/32$^{th}$ the clock speed of CLOCK.

### UNDERSTANDING | TASK 3

Complete the partial code above for the SLOW_CLOCK signal so that it can be used for implementation / simulation.

*Hint: SLOW_CLOCK is an output signal, as well as a register that holds the SLOW_CLOCK value. The initial state of the SLOW_CLOCK register can be zero or one.*

Simulate the Verilog codes, so that a waveform that toggles every 16$^{th}$ clock cycle can be observed.

### UNDERSTANDING | TASK 4

Explore what happens if COUNT is forcefully toggled before the overflow to 0 occurs. To do so, add a conditional statement to allow COUNT to become 0 at an earlier clock cycle instead of waiting for 16 clock cycles.

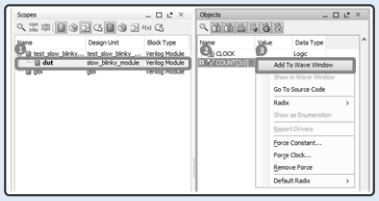
```
    always @ (posedge CLOCK) begin
        COUNT <= (COUNT == 4'b1000) ? 0 : COUNT + 1;
        SLOW_CLOCK <= ( COUNT == 4'b0000 ) ? ~SLOW_CLOCK : SLOW_CLOCK ;
    end
```

Simulate the Verilog codes again and observe if the waveforms have toggled earlier.

# FURTHER ANALYSIS IN THE SIMULATION WAVEFORM WINDOW

By default, the simulation window only shows the waveforms of input and output signals. To see the waveforms of variables during the simulation, such as the variable **COUNT**:

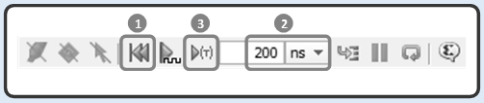
1. Select the **dut** under the simulation module being used
2. In the **Objects** window, right click on the **COUNT** variable
3. Choose **Add To Wave Window**



After adding the variable **COUNT** to the wave window, the current simulation needs to be re-run. Follow these steps:

1. Click on the "Rewind" symbol to remove the waveforms
2. Set the simulation time and unit
3. Click on the "Run" symbol to simulate for the time set in step 2

**This is an important step to remember if you want to simulate for more than the default 1000 ns of simulation!!!**




The **COUNT** variable can then be expanded by clicking on the **+** symbol to the left of **COUNT**. This allows for every individual bit to be observed as independent waveforms:


## UNDERSTANDING | TASK 5

Assume LD15 on the Basys3 development board is the LED to blink.

Consider the three switches SW0, SW1 and SW2, and their behaviours:

- When these three switches are OFF, LD15 must be OFF
- When SW0 is ON, LD15 must blink at 20 Hz
- When SW1 is ON, LD15 must blink at 5 Hz
- When SW2 is ON, LD15 must blink at 1 Hz

Implement the above requirement on the Basys 3 development board. Use the following hints:

- Create a top-level module with the Basys3 clock and 3 switches as inputs, and 1 LED as output
- Create **3 separate Verilog modules**, each one producing a slower clock speed of 20 Hz, 5 Hz and 1 Hz
- The 3 slower clock signals go into a multiplexer. The output of the mux connects to the LED
- The multiplexer has a 3-bits control signal. Based on the switch that is on, one of the slower clock speeds will be connected to the output of the mux. A value of 0 connects to the output if all three switches are off

**Note: For the EE2026 labs, a blinking frequency of 1 Hz means that the LED should be ON for 0.5 seconds, and OFF for around 0.5 seconds, for each blinking cycle**

## UNDERSTANDING | TASK 6

Assume an LED on the Basys3 board needs to turn on every one second:

When 1 second has passed since the beginning, only LD0 turns ON.
When 2 seconds have passed since the beginning, only LD1 turns ON.
When 3 seconds have passed since the beginning, only LD2 turns ON.
When 4 seconds have passed since the beginning, only LD3 turns ON.

To implement the above task on the Basys3 development board, use the following:

- Create a top-level module with the basys3 clock as input, and 4 LEDs as outputs.

- Create a module whose only purpose is to increase a COUNT value every 1 second. A **partial Verilog code** is:

```
always @ (posedge CLOCK_1HZ) begin
    COUNT <= COUNT + 1;
end
```

  In other words, COUNT is keeping track of time, accurate to 1 second.

- Create another module whose only purpose is to control the LEDs. A **partial Verilog code** is:

```
always @ (posedge CLOCK_25MHZ) begin
    if (COUNT == 2'b00) begin
       LEDS <= 4'b0001;
    end
    if (COUNT == 2'b01) begin
       LEDS <= 4'b0010;
    end
    if (COUNT == 2'b10) begin
       LEDS <= 4'b0100;
    end
    if (COUNT == 2'b11) begin
       LEDS <= 4'b1000;
    end
end
```

- Use the top-level module to instantiate and connect the **separate modules** that you have created above. You can now implement that top-level module it on the Basys3 board to see a certain LED turning ON every 1 second.

**IMPORTANT:** It is required to **successfully implement and fully understand the structural modelling and behavioural codes** used in **UNDERSTANDING | TASK 5** and **UNDERSTANDING | TASK 6** **before starting** the graded post-lab assignment.

# BEFORE STARTING ON THE GRADED ASSIGNMENT:

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

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

Students are required to carefully read all the assignment requirements, and to enhance their understanding by looking at the example.

Marks are given for using your correct student matriculation number, and grading is done only once. Re-submissions, late submissions, or updates, even if very simple or minor, are **not accepted** after the grading.

# GRADED POST-LAB ASSIGNMENT

Complete as much as possible, in **ONE working bitstream for this whole assignment**. It is much better to have a working program with some completed functionalities, instead of submitting a program without a working bitstream (No marks given).

## IMPORTANT CHARACTERS

In this assignment, these are the important characters to note from your student matriculation number:

- The 1st rightmost numerical value of your student matriculation number (Subtask A)
- The 2nd rightmost numerical value of your student matriculation number (Subtask B)

## INITIALISATION

When the program starts, switches SW0 to SW15 must be in the OFF position. The seven segment displays are all OFF.

## SUBTASK A

At the start of the program, a set of LEDs starting from LD0 to MAX_LED (Including) are required to turn ON after every TIME_COUNT. Other LEDs, from MAX_LED (Excluding) to LD15, are OFF. MAX_LED is dependent on the **1st rightmost numerical value of your student matriculation number**, as indicated in the table below:

| 1st Rightmost Numerical Value | MAX_LED | TIME_COUNT |
|---|---|---|
| **0** | LD14 | 0.20 seconds (Error of ± 0.05 seconds accepted) |
| **1** | LD13 | 0.36 seconds (Error of ± 0.05 seconds accepted) |
| **2** | LD12 | 0.54 seconds (Error of ± 0.05 seconds accepted) |
| **3** | LD11 | 0.75 seconds (Error of ± 0.05 seconds accepted) |
| **4** | LD10 | 1.00 seconds (Error of ± 0.05 seconds accepted) |
| **5** | LD9 | 1.20 seconds (Error of ± 0.05 seconds accepted) |
| **6** | LD8 | 1.11 seconds (Error of ± 0.10 seconds accepted) |
| **7** | LD7 | 1.00 seconds (Error of ± 0.10 seconds accepted) |
| **8** | LD6 | 0.86 seconds (Error of ± 0.10 seconds accepted) |
| **9** | LD5 | 0.67 seconds (Error of ± 0.10 seconds accepted) |

When all the LEDs from LD0 to MAX_LED are ON, the user can turn ON certain switches. These switches should make certain LEDs blink, as indicated in the truth table below (X is the don't care condition):

| INPUTS | | | OUTPUTS | | | |
|---|---|---|---|---|---|---|
| **SW2** | **SW1** | **SW0** | **MAX_LED to LD2** | **LD2** | **LD1** | **LD0** |
| 0 | 0 | 0 | ON | ON | ON | ON |
| 0 | 0 | 1 | ON | ON | ON | Blink at 1 Hz |
| 0 | 1 | X | ON | ON | Blink at 10 Hz | ON |
| 1 | X | X | ON | Blink at 100 Hz | ON | ON |

(Error of ± 10% accepted for all blinking frequencies)

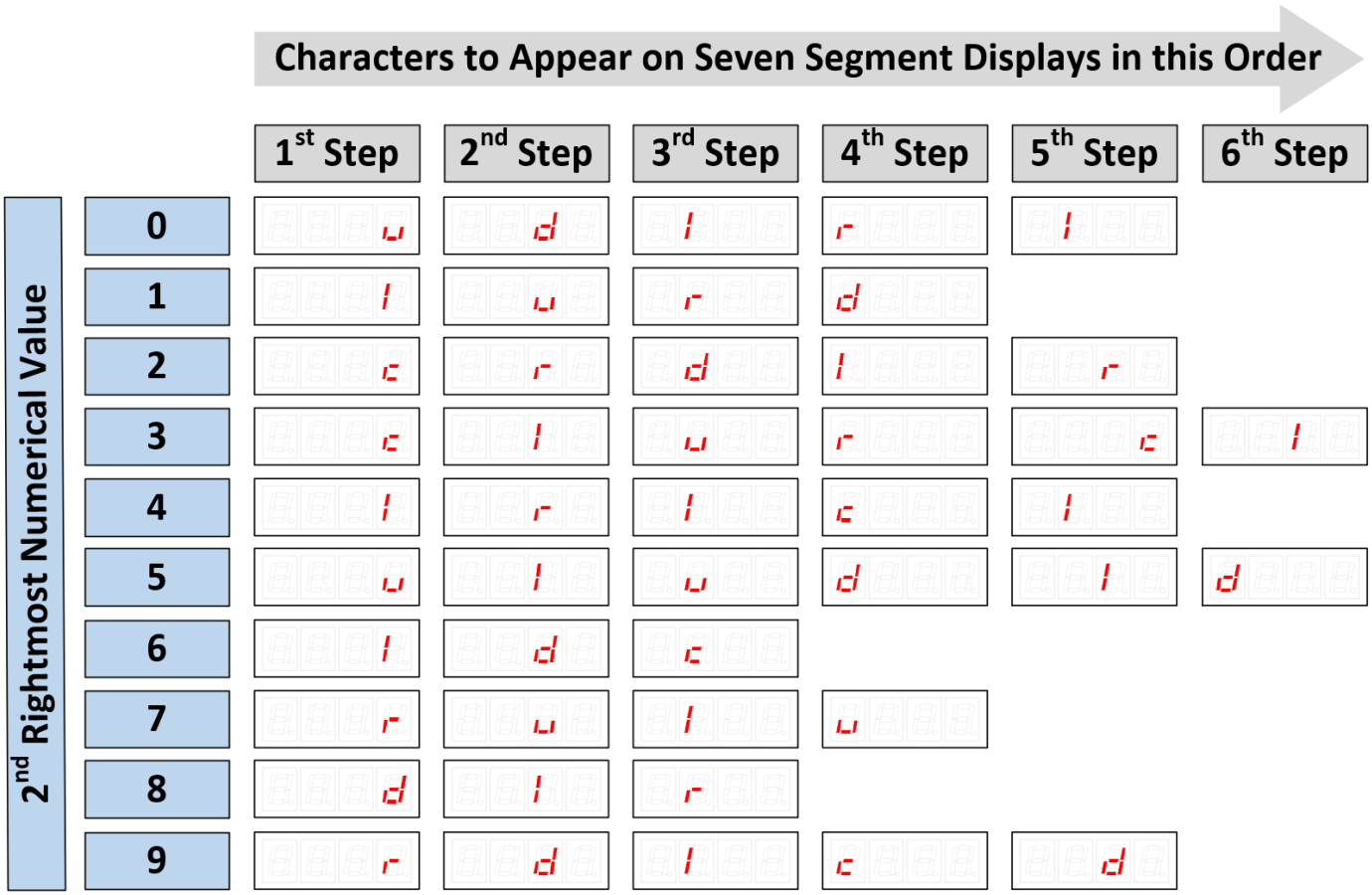
***Note:*** *Assume that the grader will only turn ON the switches after MAX_LED is ON. The switches will not be turned ON while the LEDs are lighting up.*

**SUBTASK B**

The seven segment displays must not show anything before MAX_LED is ON.

At the end of SUBTASK A, the seven segment displays and MAX_LED simultaneously turn ON (Both seven segments and MAX_LED are to be ON within 10 milliseconds of each other).

The seven segment displays start by showing the "1st Step" character. The user will then be required to press certain pushbuttons based on the characters that appear on the seven segment displays. All pushbuttons' presses must always be detected within 10 milliseconds of being pressed. In other words, the system must react to actions very responsively, and by changing to the next step within 10 milliseconds.




| | |
|---|---|
| UP | UP Pushbutton (BTNU) must be pressed before next step |
| RIGHT | RIGHT Pushbutton (BTNR) must be pressed before next step |
| DOWN | DOWN Pushbutton (BTND) must be pressed before next step |
| LEFT | LEFT Pushbutton (BTNL) must be pressed before next step |
| CENTER | CENTER Pushbutton (BTNC) must be pressed before next step |

After the user has pressed the required pushbuttons in the correct order, the system goes into an "UNLOCKED" mode and stays there forever, which is indicated by LD15 being ON. When LD15 is ON, the seven segment displays must show the 1st step again.

***Note:***

- *The system should ignore all incorrect presses (Nothing happens for wrong pushbutton presses)*
- *The pushbuttons will not be pressed after the system goes into "UNLOCKED" mode.*

## IMPORTANT REMINDERS:

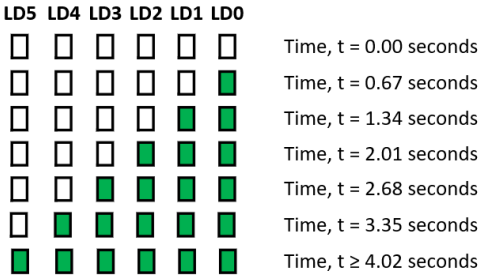
- Case statements and if-else statements must be inside an always block.
- **Be careful of parallel execution of the always blocks. Multi-driven nets indicate that there are conflicting values being given to the same signal from different always blocks.** For example, one cannot tell a signal to increase at a time instant t, and at that same time instant t, telling it to decrease.

## EXAMPLE:

If your student matriculation number is A0159089Y, then:

1st rightmost numerical value: **9** (Subtask A depends on this number: MAX_LED is LD5)
2nd rightmost numerical value: **8** (Subtask B depends on this number)

### SUBTASK A




For this student (A0159089Y), MAX_LED is LD5.
Since MAX_LED is ON when t ≥ 4.02 seconds for this student, LEDs can now blink as indicated in the truth table:

| INPUTS | | | OUTPUTS | | | |
|---|---|---|---|---|---|---|
| SW2 | SW1 | SW0 | MAX_LED to LD2 | LD2 | LD1 | LD0 |
| 0 | 0 | 0 | ON | ON | ON | ON |
| 0 | 0 | 1 | ON | ON | ON | Blink at 1 Hz |
| 0 | 1 | X | ON | ON | Blink at 10 Hz | ON |
| 1 | X | X | ON | Blink at 100 Hz | ON | ON |

### SUBTASK B

For this student (A0159089Y), MAX_LED is LD5:

*All pushbuttons' presses must always be detected within 10 milliseconds of being pressed. In other words, the system must react to actions very responsively, and by changing to the next step within 10 milliseconds.*

LD5 is still OFF.
Therefore, seven segments displays are still OFF.

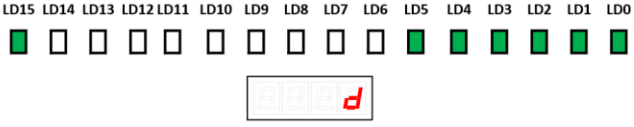
Simultaneously (Within 10 milliseconds) when LD5 turns ON.

User has just pressed BTND in the previous step.

User has just pressed BTNL in the previous step.

User has just pressed BTNR in the previous step.
System now in "UNLOCKED" mode.

In "UNLOCKED MODE", the pushbuttons will no longer be pressed. Switches from SUBTASK A will still be used.



# CANVAS SUBMISSION INSTRUCTIONS

- Ensure that your bitstream has been successfully generated and tested on your Basys 3 development board **BEFORE** archiving your Vivado workspace for CANVAS upload.

- It is compulsory to archive your project in a compressed form without any saved simulation waveforms. Follow the instructions given in the pdf: "Archive Project in Vivado 2018.02". **The archive size should not exceed 5 MB in size for any lab assignments.** If the archive size exceeds, ensure that there is no ".sim" folder in your Vivado project folder before archiving.

- **After** following the instructions in "**Archive Project in Vivado 2018.02**", rename your project archive as indicated in the appendix of this lab manual.

- Upload to CANVAS EE2026 -> Assignments -> Lab 3 Graded Assignment -> Lab 3 Submission (On-Time).

- Download your CANVAS archive after uploading. **Click and drag the single folder within that archive to desktop**, and then open the Vivado project (.xpr) in that **extracted folder** to see if it can be opened. You can ignore "out-of-date" warnings. **Check if you can also run your bitstream correctly.** No project files and no working bitstream is equivalent to losing all marks.

- The CANVAS upload must be completed by **Thursday 25th September 2025, 6:00 A.M. (Morning)**. Avoid planning to upload during the grace period of 2 hours. (Grace period is for last-minute cases, such as due to slow internet, wrong uploads etc.)

- A penalty of 10% on the assignment 3 weight applies for **late submissions of up to 1 week**. Submissions after that 1 week may not be accepted. (Delaying the assignment will make all subsequent labs exponentially more difficult to manage)

- The late submission folder closes 1 week after the original deadline. **Late submissions are not graded if submissions are found in the on-time folder. Do not submit in the late submission folder if you have already submitted in the on-time folder. Inform your lab instructor if you have uploaded in both the on-time folder and late submission folder.** The late submission folder will be located at: CANVAS EE2026 -> Assignments -> Lab 3 Graded Assignment -> Lab 3 Submission (Late).

- Re-submissions, late submissions, or updates, even if very simple or minor, are **not accepted** after your original submission has been graded.

- Submissions through emails, file sharing programs, or links, are **not accepted**. Submissions must be officially submitted to, and downloadable from, Canvas in the correct on-time or late folder.

**Plagiarism is penalised with a 100% penalty for all SOURCES and RECIPIENTS**

All past and future submissions, and marks, will be reviewed in greater detail, for any person found to have plagiarised

**ALL THE SUBMISSION INSTRUCTIONS LISTED ABOVE WILL AFFECT YOUR GRADES!**

---

## GRADING PROCESS

- During subsequent lab sessions, our graders will be providing you updates on the grading of your submission.

- Submissions not following all the ***CANVAS SUBMISSION INSTRUCTIONS*** (listed above) will not be graded immediately, and they will instead be reviewed towards the end of the semester. **You will not be able to see your submission results during the lab sessions in such situations**.

## APPENDIX (COMPULSORY renaming just before CANVAS upload):

It is **compulsory to rename your project archive** just before the CANVAS upload, as listed in the table below.

Do not change any other part of the naming. Simply **copy** the naming from the table below, and **paste** it while renaming your project archive. Penalties will be incurred if your submission cannot be found according to the exact naming template below.

CANVAS may automatically add a suffix (Example. "-1", "-2", "-3" etc.) at the end of the filename if you submit multiple files. That is acceptable and we will grade the latest file submitted within that on-time folder.

| Archive Naming (.xpr.zip and .zip accepted) |
|---|
| Lab3_Mon_AM_AARAV RAJESH_165_Archive |
| Lab3_Mon_AM_ABE FOO QIAN YIN_426_Archive |
| Lab3_Mon_PM_ABHIJIT BALAJEE_045_Archive |
| Lab3_Mon_PM_ABIRAMI BASKARAN_550_Archive |
| Lab3_Mon_AM_ABUDAHEER MOHAMMED IBRAHI_637_Archive |
| Lab3_Fri_AM_ADIT BISWAS_789_Archive |
| Lab3_Mon_PM_AFSHAL GULAM_376_Archive |
| Lab3_Mon_PM_AHMAD TIRMIZI BIN ADAM_523_Archive |
| Lab3_Mon_PM_AIDAN CHIA TONG_306_Archive |
| Lab3_Mon_PM_ALEXANDER YAW KAI MUN_328_Archive |
| Lab3_Mon_PM_ALIYEV ESHGIN_149_Archive |
| Lab3_Mon_PM_ARIFF MUHAMMED AHSAN HUSA_407_Archive |
| Lab3_Fri_AM_AW SHUO JIE_803_Archive |
| Lab3_Mon_PM_AYDEN VAN ETTEN_971_Archive |
| Lab3_Fri_AM_BADRINATH SANDHYA_704_Archive |
| Lab3_Mon_AM_BAE SOOHUN_652_Archive |
| Lab3_Mon_PM_BAJAJ KRISHNA_769_Archive |
| Lab3_Mon_AM_BENJAMIN CHEK JUN KIET_381_Archive |
| Lab3_Mon_PM_BENJAMIN LOH JIAN WEI_455_Archive |
| Lab3_Mon_PM_BHADRA PARV_813_Archive |
| Lab3_Fri_AM_BINDAWALA AARAV_779_Archive |
| Lab3_Mon_AM_BRENDAN TEY SHI HAN_957_Archive |
| Lab3_Fri_AM_BRIEN LIM CHONG_162_Archive |
| Lab3_Mon_AM_CEDRIC TAN SI YU_331_Archive |
| Lab3_Mon_PM_CHAKRABORTY SHRABASTI_685_Archive |
| Lab3_Mon_PM_CHAN YI FENG_824_Archive |
| Lab3_Mon_AM_CHANG YI HERNG_612_Archive |
| Lab3_Mon_AM_CHARNG HE_580_Archive |
| Lab3_Fri_AM_CHEN GUANWEN_978_Archive |
| Lab3_Fri_AM_CHEN HONGYU_122_Archive |
| Lab3_Mon_PM_CHEN XINGTONG_961_Archive |
| Lab3_Mon_PM_CHEN YU HSIN_432_Archive |
| Lab3_Fri_AM_CHEO ZHI XIAN MATHEU_851_Archive |
| Lab3_Mon_PM_CHEONG JIAN HAO_240_Archive |
| Lab3_Fri_AM_CHEW EN WEI_658_Archive |
| Lab3_Mon_AM_CHIA HAO JUN_207_Archive |
| Lab3_Fri_AM_CHIA LE ISAAC_940_Archive |
| Lab3_Mon_PM_CHNG RUI YONG SEAN_435_Archive |
| Lab3_Mon_PM_CHONG KAI JIE_314_Archive |
| Lab3_Mon_AM_CHONG WEIXUAN CYDRIC_871_Archive |

| |
|---|
| Lab3_Mon_PM_CHOY ZHAN HONG_444_Archive |
| Lab3_Mon_AM_CHUA YONG LIANG_014_Archive |
| Lab3_Fri_AM_CUI JIAHAO_085_Archive |
| Lab3_Mon_PM_DANIEL CHUA ZHENG JIE_443_Archive |
| Lab3_Mon_PM_DENG HAOFU_338_Archive |
| Lab3_Mon_PM_DENY HAANS HAZRIEL BIN AR_153_Archive |
| Lab3_Mon_PM_DIWAKAR VIDYA ADHAVAN_335_Archive |
| Lab3_Mon_PM_DYLAN LIM_821_Archive |
| Lab3_Mon_AM_EMRY DANIEL BIN ABDUL LAT_550_Archive |
| Lab3_Fri_AM_FOO KANG_353_Archive |
| Lab3_Mon_AM_GABRA SHUBHAN_258_Archive |
| Lab3_Mon_PM_GABRIEL LEE JING YI_006_Archive |
| Lab3_Fri_AM_GAN ANDREW HOA THIEN_281_Archive |
| Lab3_Fri_AM_GOH ANG LEE_598_Archive |
| Lab3_Fri_AM_GOH SHAO ANN_219_Archive |
| Lab3_Mon_AM_GOH SZE ANH_325_Archive |
| Lab3_Fri_AM_GOH YOU YI_326_Archive |
| Lab3_Mon_PM_GORDON HONG JIA JIE_465_Archive |
| Lab3_Mon_PM_GU MINGYOUJIA_260_Archive |
| Lab3_Mon_AM_GUO ZICHENG_823_Archive |
| Lab3_Mon_AM_HANNAH WESTERHOUT HASAN_413_Archive |
| Lab3_Mon_PM_HAO YIAN_006_Archive |
| Lab3_Fri_AM_HIEW T G_306_Archive |
| Lab3_Fri_AM_HOWIE YEO HAO YU_345_Archive |
| Lab3_Fri_AM_HU LIFAN_030_Archive |
| Lab3_Fri_AM_HU XIRAN_503_Archive |
| Lab3_Mon_AM_HUANG HAU SHUAN_356_Archive |
| Lab3_Mon_AM_HUANG YUANJIN_080_Archive |
| Lab3_Mon_AM_IRWAN AHMED NOOR_184_Archive |
| Lab3_Fri_AM_JAIRUS LEUNG JIE RUI_337_Archive |
| Lab3_Mon_AM_JANSEN KEN PEGRASIO_566_Archive |
| Lab3_Fri_AM_JAVIER YEOH ZHI JI_430_Archive |
| Lab3_Fri_AM_JOEL KU_232_Archive |
| Lab3_Fri_AM_JOEL LIM JUN YI_423_Archive |
| Lab3_Fri_AM_JOHN KENNETH LAYBA AGPAOA_489_Archive |
| Lab3_Fri_AM_JOSHUA TAM KA YUI_315_Archive |
| Lab3_Fri_AM_JOSHUA YEO WEE TZE_878_Archive |
| Lab3_Mon_AM_JOVIAN JOSH_585_Archive |
| Lab3_Mon_AM_KARTHIK KATHIRESH_122_Archive |
| Lab3_Mon_PM_KARTHIKEYAN VETRIVEL_899_Archive |
| Lab3_Fri_AM_KENNETH WONG CUN WI_742_Archive |
| Lab3_Mon_AM_KEVIN LOKE WEI YI_957_Archive |
| Lab3_Fri_AM_KHOO JUNHAO_052_Archive |
| Lab3_Mon_AM_KOH LI TIAN_429_Archive |
| Lab3_Fri_AM_KOTHARI SMEET RONAK_661_Archive |
| Lab3_Fri_AM_KUAH JUN HONG BRYAN_384_Archive |
| Lab3_Mon_AM_LABELLE LEE_619_Archive |
| Lab3_Fri_AM_LAM ZHEN LEI ETHAN_558_Archive |
| Lab3_Mon_PM_LAU EN XIN GRACE_497_Archive |
| Lab3_Fri_AM_LEE KAH HOE BRIAN_300_Archive |
| Lab3_Mon_PM_LEE KUAN YI_201_Archive |
| Lab3_Mon_PM_LEONARD LIM TZE YANG_460_Archive |
| Lab3_Mon_PM_LI LINYING_166_Archive |

| |
|---|
| Lab3_Fri_AM_LI XIAOYANG_853_Archive |
| Lab3_Fri_AM_LIM KAI LER ETHAN_939_Archive |
| Lab3_Mon_PM_LIM SWEE HOW GABRIEL_173_Archive |
| Lab3_Mon_AM_LIM ZE EN MATTHIAS_943_Archive |
| Lab3_Mon_PM_LIU LEKUAN_703_Archive |
| Lab3_Fri_AM_LIU YIHAN_493_Archive |
| Lab3_Mon_AM_LOH HAN XIANG_229_Archive |
| Lab3_Fri_AM_LOU YU_962_Archive |
| Lab3_Mon_AM_LOUIS AGARA PERIN_320_Archive |
| Lab3_Mon_AM_LOW ZEN WEI_635_Archive |
| Lab3_Mon_AM_LU QIANXI_033_Archive |
| Lab3_Mon_PM_LUO HONGXUN_391_Archive |
| Lab3_Fri_AM_MAHESH PURAV_679_Archive |
| Lab3_Mon_PM_MANU DAGUR_861_Archive |
| Lab3_Mon_PM_MAO XIAOHAN_110_Archive |
| Lab3_Fri_AM_MARK NG JIAN XIONG_883_Archive |
| Lab3_Mon_AM_MATHEW SAAYUJ ION_282_Archive |
| Lab3_Fri_AM_MICHAEL SHYAM WILFRED DAV_218_Archive |
| Lab3_Mon_AM_MO HANQI_531_Archive |
| Lab3_Mon_AM_MOHAMED FARAS SO FARIDUL_806_Archive |
| Lab3_Fri_AM_MUHAMMAD AKMAL HANIS BIN _123_Archive |
| Lab3_Mon_PM_NAILA FAIQAH BINTE MUHAMM_406_Archive |
| Lab3_Mon_AM_NAVANEETHAN SANJAI_231_Archive |
| Lab3_Mon_AM_NEERAJ VENGADESSAN_959_Archive |
| Lab3_Mon_AM_NI SHI YONG_596_Archive |
| Lab3_Mon_PM_NICHOLAS LAU HONGYI_324_Archive |
| Lab3_Mon_PM_NOCHUR SIVANSH_332_Archive |
| Lab3_Mon_PM_ONG CHONG YAO_901_Archive |
| Lab3_Mon_AM_ONG XIANG KAI_232_Archive |
| Lab3_Fri_AM_PANG ANG SHENG ASHER_237_Archive |
| Lab3_Mon_PM_PRANAV JANAKIRAMAN_346_Archive |
| Lab3_Mon_AM_PREMIL ROSHAN_805_Archive |
| Lab3_Fri_AM_PRERANA RAVI SHANKAR_092_Archive |
| Lab3_Mon_PM_RAJAN PRAVEEN_146_Archive |
| Lab3_Mon_PM_RAJARAM SUSHMIITHAA_309_Archive |
| Lab3_Mon_PM_RAJESH KUMAR ASWIN_713_Archive |
| Lab3_Mon_PM_REHAAN MAHMOOD_719_Archive |
| Lab3_Fri_AM_RISHABH RAMPRASAD SHENOY_597_Archive |
| Lab3_Mon_AM_ROHAN H_155_Archive |
| Lab3_Mon_AM_ROSHAN ALAGAR PREMKUMAR_245_Archive |
| Lab3_Fri_AM_RUSSELL NG JUN HENG_204_Archive |
| Lab3_Mon_PM_RYAN KOH JUN HAO_601_Archive |
| Lab3_Mon_PM_RYAN PANG ZE XI_453_Archive |
| Lab3_Mon_PM_RYAN TAN RONG CHANG_406_Archive |
| Lab3_Mon_AM_SAMUEL LEE WEN JIN_232_Archive |
| Lab3_Mon_PM_SANGHI NISHCHAY_666_Archive |
| Lab3_Mon_AM_SARAVANAN RAJESHWARI AKSH_513_Archive |
| Lab3_Mon_PM_SEAN LEE WEI SHU_252_Archive |
| Lab3_Fri_AM_SEAN TAN KAI JIE_767_Archive |
| Lab3_Fri_AM_SEAN TAN LIYU_776_Archive |
| Lab3_Mon_PM_SEOW JACKIE JAVIER_092_Archive |
| Lab3_Mon_PM_SHAH JAINAM AMIT_339_Archive |
| Lab3_Fri_AM_SHAH KUSHAL HITESH_473_Archive |

| |
|---|
| Lab3_Mon_PM_SHAUN TAN SHU REN_075_Archive |
| Lab3_Fri_AM_SHENNON TAY_016_Archive |
| Lab3_Mon_AM_SHYAMAL VARNAN VENKATARAM_949_Archive |
| Lab3_Mon_PM_SIM MENG TECK_476_Archive |
| Lab3_Mon_PM_SINGH SIDDHANT NARAYAN_752_Archive |
| Lab3_Mon_AM_SRINIVASAN RANGANATHAN_330_Archive |
| Lab3_Mon_AM_SRIVASTAVA HARSHIT_749_Archive |
| Lab3_Mon_PM_SUWEI SHRESTHA_946_Archive |
| Lab3_Mon_PM_TAN CHUN LIANG_676_Archive |
| Lab3_Fri_AM_TAN KAI CONG_476_Archive |
| Lab3_Mon_AM_TAN PANG_905_Archive |
| Lab3_Mon_AM_TAN WEI HENG_981_Archive |
| Lab3_Mon_AM_TAN YAN HAO MALCOLM_363_Archive |
| Lab3_Mon_PM_TAN YUE YANG_427_Archive |
| Lab3_Fri_AM_TANPRASERTKUL PRAN_173_Archive |
| Lab3_Mon_PM_TAO ENZE_746_Archive |
| Lab3_Fri_AM_TEO YU XIANG ALOYSIUS_442_Archive |
| Lab3_Mon_PM_THEN CHIN KIAT_534_Archive |
| Lab3_Fri_AM_THIA YANG HAN_615_Archive |
| Lab3_Mon_AM_TIRODKAR OM MILIND_212_Archive |
| Lab3_Fri_AM_TJHIN BRIAN_808_Archive |
| Lab3_Fri_AM_TOH EE SEN IZEN_600_Archive |
| Lab3_Mon_PM_TOH YI WEI_295_Archive |
| Lab3_Mon_AM_TONG KAR YUE ABIGAIL_126_Archive |
| Lab3_Mon_PM_VANSH PURI_003_Archive |
| Lab3_Mon_AM_VENKATESH KSHEERABTHI NAT_380_Archive |
| Lab3_Mon_AM_VIHAAN_665_Archive |
| Lab3_Fri_AM_VINAY VANJRE RAVI_336_Archive |
| Lab3_Fri_AM_WAI YAN_450_Archive |
| Lab3_Fri_AM_WANG CHUHAO_566_Archive |
| Lab3_Mon_AM_WANG JIAWEI_498_Archive |
| Lab3_Fri_AM_WANG PENGJIN_104_Archive |
| Lab3_Mon_PM_WANG ZAIXI_940_Archive |
| Lab3_Mon_PM_WEN JUN YU_517_Archive |
| Lab3_Fri_AM_WENG XIAN YANG_191_Archive |
| Lab3_Mon_PM_WESLEY LOW_090_Archive |
| Lab3_Mon_PM_WONG EE SHAWN_244_Archive |
| Lab3_Mon_AM_WU ZI EN ELLIOT JOHN_902_Archive |
| Lab3_Mon_PM_XU HUANGHAO_370_Archive |
| Lab3_Fri_AM_XU ZIHAO_485_Archive |
| Lab3_Fri_AM_XYLON CHAN YI CHONG_514_Archive |
| Lab3_Mon_PM_YADAV ARYAN SUNILKUMAR_694_Archive |
| Lab3_Mon_PM_YAN XIANGYU_416_Archive |
| Lab3_Mon_PM_YAO XIANG_192_Archive |
| Lab3_Mon_PM_YAP JIT EN FAITH_368_Archive |
| Lab3_Mon_AM_YEE JIA JIE_896_Archive |
| Lab3_Mon_PM_YEO CHEN XIAN_213_Archive |
| Lab3_Fri_AM_YEO SI ZHAO_884_Archive |
| Lab3_Fri_AM_YEO TECK IAN BRYAN_832_Archive |
| Lab3_Mon_AM_YEO YEE CHING_332_Archive |
| Lab3_Mon_PM_YEOH SOO LEONG_125_Archive |
| Lab3_Mon_AM_YEUNG KEITH_484_Archive |
| Lab3_Mon_PM_YI YANG_914_Archive |

| |
|---|
| Lab3_Fri_AM_ZHANG YIZE_948_Archive |
| Lab3_Mon_AM_ZHAO ZEYU_093_Archive |
| Lab3_Fri_AM_ZHENG HUIJIE_077_Archive |
| Lab3_Fri_AM_ZHENG KAIWEN_674_Archive |
| Lab3_Fri_AM_ZHU YICHENG_485_Archive |

---

**FINAL CHECKLIST FOR ASSIGNMENT 3:**

**Grading is done only once**. Re-grading due to not meeting the checklist below are not accepted.

- ☐ I have used the **exact and correct numbers and alphabet** from my student matriculation number for the assignment.
- ☐ I have named the submitted archive **according to the given template**.
- ☐ The submitted archive is **less than 5 MB** (Delete ".sim" folder if more than 5 MB. Penalty of up to 10% apply if ≥5 MB).
- ☐ I confirm that the uploaded **submission is the correct / latest version**, and not an older one.
- ☐ I have **downloaded my submission** before the deadline and confirmed that it was properly uploaded to canvas.
- ☐ I confirm that the **grader need not "Generate bitstream"**, as the bitstream is already there during "Program Device".
- ☐ I have carefully verified the bitstream in my **CANVAS submission**, and it works exactly as I expected.