LAB VENUE:
DIGITAL ELECTRONICS LAB @ E4-03-07

LAB SCHEDULE:

2022-2023 SEMESTER 2	LABORATORY GROUP		
	B01 (Monday AM)	B02 (Wednesday AM)	B03 (Wednesday PM)
Week 1			
Week 2	Fundamental Lab 1		
Week 3		Fundamental Lab 1	Fundamental Lab 1
Week 4	Fundamental Lab 2	Fundamental Lab 2	Fundamental Lab 2
Week 5	Fundamental Lab 3	Fundamental Lab 3	Fundamental Lab 3
Week 6			
Recess Week			
Week 7	Project Lab 1	Project Lab 1	Project Lab 1
Week 8	Project Lab 2	Project Lab 2	Project Lab 2
Week 9	Project Lab 3 & In-Class Evaluation	Project Lab 3 & In-Class Evaluation	Project Lab 3 & In-Class Evaluation
Week 10	Project Lab 4	Project Lab 4	Project Lab 4
Week 11			
Week 12	Project Assessment	Project Assessment	Project Assessment
Week 13			

EE2026: DIGITAL DESIGN

Academic Year 2022-2023, Semester 2

LAB 1: Quick Start Guide to Vivado 2018.2, Basys 3 Development Board, and Verilog HDL

FOR ALL EE2026 LAB AND PROJECT SESSIONS:

- You are **strongly encouraged to bring to lab, your own laptop with Vivado 2018.2 already installed**. You may still use the desktop PC in lab if you do not have a laptop that can be brought to lab.

- Use the **D:\MyWork** folder for your work if you are using the lab PC. You are required to **delete** all folders within the **D:\MyWork** folder before starting your lab session.

- **Delete** your work folder from the laboratory's computers after your session is over. You are responsible to **safeguard** your confidential programs. For assessable programs, you will be penalised if two programs with similarities beyond empirical evidence are detected. Both the source(s) and recipient(s) of plagiarised programs are equally penalised.

OVERVIEW:

Using a simple Boolean design problem, an introductory approach to the Vivado software used in EE2026 will be covered. Quick instructions on downloading and installing the Vivado software on your personal computer are provided. The Vivado software is a comprehensive integrated development environment (IDE) for FPGA design flow.

In this lab:

- An introduction to very basic Verilog HDL (Hardware Description Language) is provided.
- The overall process flow of designing, synthesising, simulating and implementing a program is covered.
- Programming Digilent's Basys 3 development board, which features an FPGA from Xilinx's Artix-7 family, is illustrated.

GRADED ASSIGNMENT [CANVAS SUBMISSION: SUNDAY 29th JANUARY 2023, 10:00 P.M.]:

Details are available at the end of this lab manual

VIVADO DOWNLOAD AND INSTALLATION:

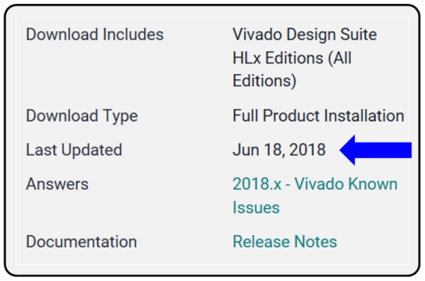
The Vivado 2018.2 software is already installed on the computers in the Digital Electronics Lab, and are ready for immediate usage. **It is also required that you install such software on your own personal computer, preferably before coming for the first lab session.** Some quick guidelines on installing the required software for EE2026 on your personal computer is provided in this section.

Software:

https://www.xilinx.com/support/download/index.html/content/xilinx/en/downloadNav/vivado-design-tools/archive.html

Vivado Design Suite - HLx Editions - 2018.2 [Last Updated: Jun 18, 2018]

Warning: *Do not use other versions of the software. Only the* ***Vivado 2018.2*** *Windows version has been tested. Computer compatibility issues will occur with other versions of the software, and assessment of your project may not be possible. This will lead to loss of project marks if your project cannot be assessed.*



Select either one of the two available installers for download, based on your preference:

- Vivado HLx **2018.2**: WebPACK and Editions - Windows Self Extracting Web Installer (EXE - 50.56 MB)
- Vivado HLx **2018.2**: All OS installer Single-File Download (TAR/GZIP - 17.11 GB)

Registration is required for any downloads from the Xilinx website, but not required for installation and program usage.

Installation

During the installation phase, you will be given an option on the edition to install. The edition to be installed is:

- Vivado HL WebPACK

For subsequent customisation options, you can leave it to the default settings.

Post-Installation

Restart your computer before using the Vivado 2018.2 software. You may wish to uninstall the Xilinx Information Centre from the Windows control panel as it is not needed. This will prevent unnecessary pop-up messages by Xilinx from appearing.

DESCRIPTION OF THE SIMPLE BOOLEAN DESIGN TASK

The following task is required to be implemented on the Basys 3 development board:

- When switch **A** turns on, only **LED1** lights up.
- When switch **B** turns on, only **LED2** lights up.
- When both switches **A** and **B** turn on, **LED1**, **LED2**, and **LED3** light up.

UNDERSTANDING | TASK 1

Complete the truth table for the simple boolean design task:

INPUT		OUTPUT			MINTERM
A	B	LED1	LED2	LED3	
0	0	A	A-B +AB	AB	+ A-B-$\bar{A}\bar{B}$
0	1				$\bar{A}B$
1	0				$A\bar{B}$
1	1				AB

Deriving an SOP Boolean Equation for the Design Task

Given any truth table with any number of input variables, the sum-of-products (SOP) or product-of-sums (POS) form may be used to write out a Boolean equation for each output variable. Let us use the canonical SOP form for **LED1**:

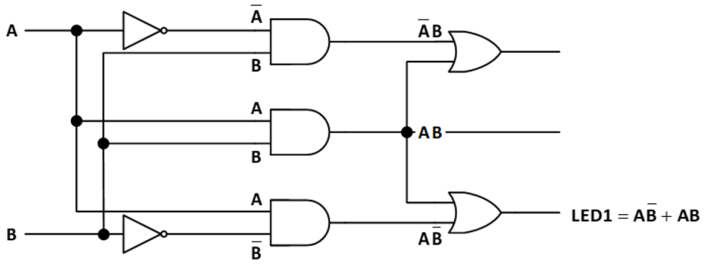
$$\mathbf{LED1} = A\bar{B} + AB$$

UNDERSTANDING | TASK 2

Work out the canonical SOP Boolean equations for **LED2** and **LED3**

Illustrating Logic Expressions by Using a Schematic of Gates

The Boolean equations for **LED1**, **LED2**, and **LED3**, can be implement by using: 2 **NOT** gates, 3 **AND** gates, 2 **OR** gates



Verilog Hardware Description and FPGA Implementation

Xilinx's Vivado software is an integrated design environment that has numerous amounts of advanced features used in the industry, and among which we will be introducing the following:

- Writing and editing HDL codes for digital system designs.
- Simulation of the design's behaviour.
- Synthesis of the codes, in order to convert the design from textual description into logic gates.
- Implementation of the design to map and route the logic to a target FPGA.
- Optimising the synthesis, implementation, and bitstream generation according to the user's strategies. The default optimisation strategies shall be used in EE2026, as changing them is beyond the scope of introductory digital designs.
- Programming an FPGA with the optimised bitstream.

The remaining part of this lab manual will now briefly show the general steps required to go from the design task, to the FPGA implementation on the Basys 3 development board, for EE2026 purposes.

INTRODUCTORY QUICK START GUIDE TO XILINX'S VIVADO 2018.2 SOFTWARE

During your lab session, your EE2026 graduate and lab assistants may provide you helpful hints on the usage of the Vivado 2018.2 software, beyond the most basic things that are described in this section.

Creating a New Verilog Project in Vivado

Start Menu: Open the executable: Vivado 2018.2. You will need to wait multiple seconds before the program opens

Quick Start: Select **Create Project** and continue

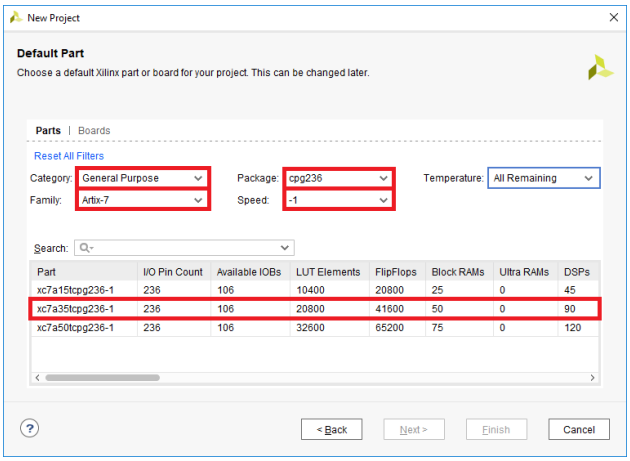
Project Name: Enter a **Project name** and **Project Location**. Ensure that the **Project name** and complete **Project location** for your project folder does not have any spaces or special characters, and that your **Project name** does not start with a number

Project Type: Select **RTL Project**, and uncheck "**Do not specify sources at this time**"

Add sources:
- **Create File.** File type is Verilog. Example: simple_boolean
- **Target language:** Verilog. **Simulator language:** Mixed

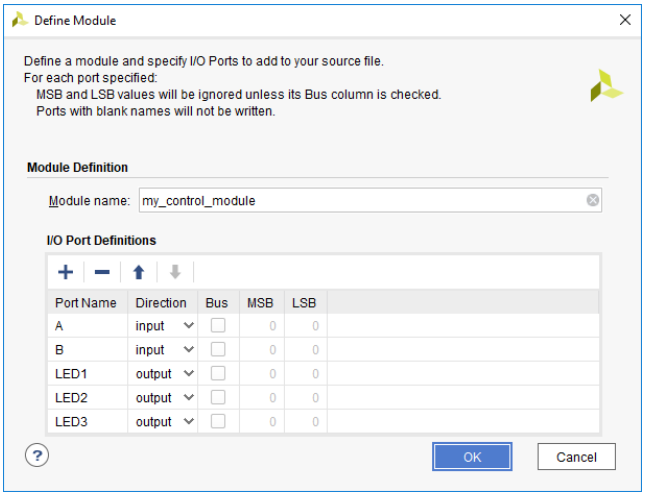
Add constraints (optional): Click on next without any changes

Default Part: Specify the FPGA chip that will be used. The Basys 3 development board uses the **xc7a35tcpg236-1** chip



New Project Summary: To create the project, click **Finish**

Define Module: A module, that is contained within the file, need top be created. Create one based on the inputs and outputs of the simple boolean design task.



Using Vivado Text Editor to Write Verilog HDL Code

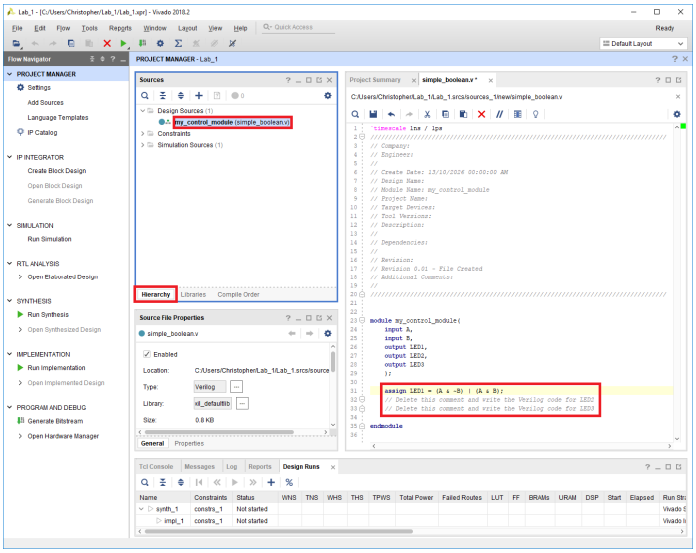
Open the module that has been created by double clicking on it in the Sources window

UNDERSTANDING | TASK 3

Code the behaviour of the module by converting the SOP expressions for **LED1**, **LED2**, and **LED3** to the Verilog equivalent. The codes are to be inserted between the keywords **module** and **endmodule**.

Some Verilog representation of common operators are as tabulated below:

Operators		Verilog Representation
OR	$\mathbf{A + B}$	|
AND	$\mathbf{AB}$	&
NOT	$\mathbf{\overline{A}}$	~
XOR	$\mathbf{A \oplus B}$	^



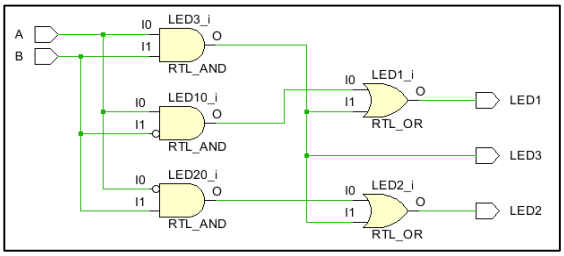
The **assign** statement causes the left hand side of the expression to be updated *every* time there is a change on the right hand side of the expression. It is therefore called a *continuous assignment* statement, describing combinational logic whereby the output on the left is a function of current inputs on the right.

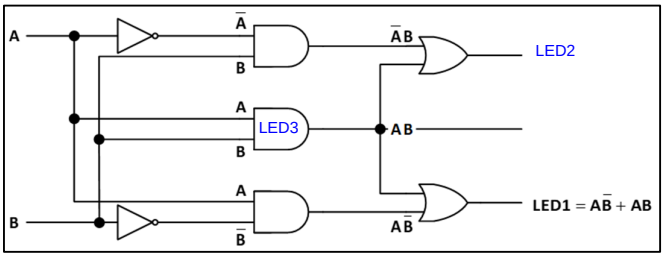
The statements on line 31 till line 33 execute concurrently. This is in contrast to sequential execution of statements in a computer programming language such as C, or procedural assignment that will be taught in subsequent lab sessions.

Save your current file by clicking on **File → Save File**, or by pressing **Ctrl+S**. Each time a file is saved, a syntax check is carried out. After saving, perform the following: In the **Flow Navigator** window, under **RTL ANALYSIS**: **Open Elaborated Design**, select **Schematic.** The schematics window will appear, showing the Register Transfer Level (RTL) schematic of the design.

UNDERSTANDING | TASK 4

What similarities and differences do you notice between the RTL schematic and the schematic obtained from the previous section. How do they compare to the actual schematic obtained on your computer screen?



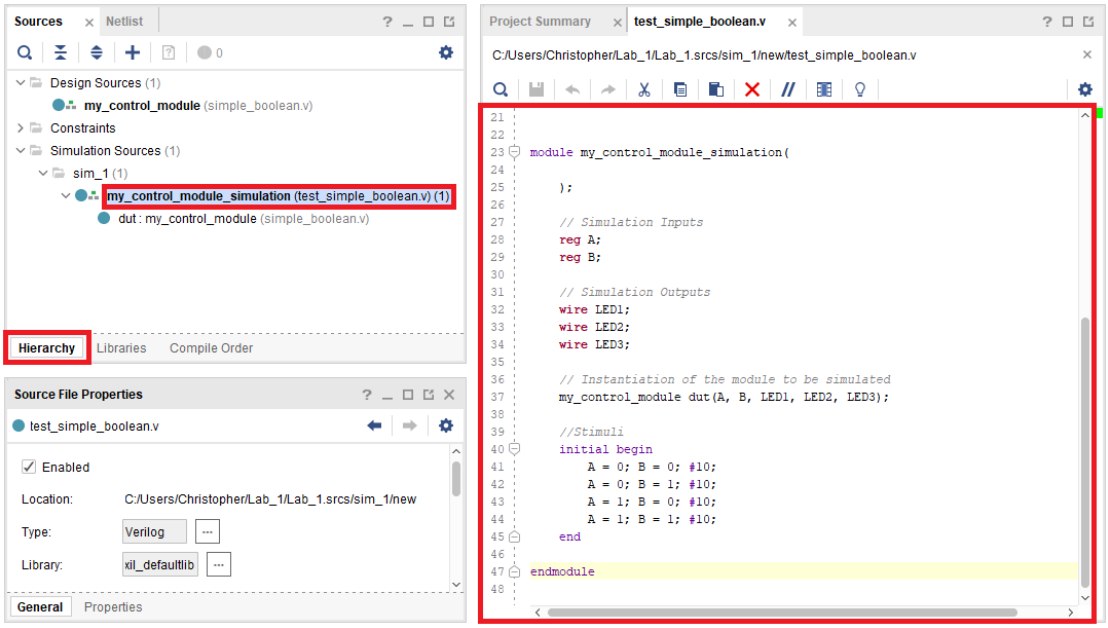


Testbench and Behavioural Simulation

After writing the codes, there is a need to test them to check their behaviours. Inputs are applied to a module, and the outputs are checked to verify whether the module operates as intended. A testbench is an HDL module that is used to test another module. In this example, a testbench will be created to apply inputs to the module to be tested:

- From the **PROJECT MANAGER**, click on **Add Sources**, followed by **Add or create simulation sources**
- **Create File**, and provide a Verilog file name, such as **test_simple_boolean**
- In the subsequent **Define Module** window, provide a **Module name**, such as **my_control_module_simulation**
- Do not input any **I/O Port Definitions**, and click on **OK** to finish creating the simulation module template

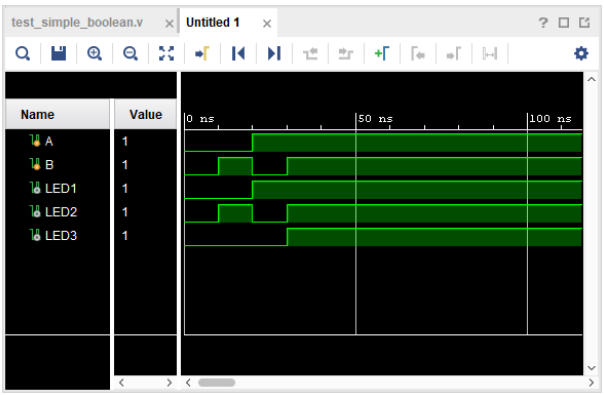
From the Sources window, open the simulation file. Then, within the simulation module, provide the following codes and save them, with the final screenshot looking similar to the image shown below:



If there are no syntax errors, in the **Flow Navigator** window, under **SIMULATION**, select **Run Simulation**, followed by **Run Behavioural Simulation** in order to create the simulation waveform window.

A noticeable waveform pattern may not be seen by default, as the time resolution used in the simulation is very small as compared to the amount of time the simulation is ran. Hence, with the simulation windows being the active window and from the menu, select **View → Zoom Fit**, or press **Ctrl+0**

Look at the simulation results closely. How do the waveforms show that your design is indeed working as desired? Consider trying out the various options provided in the simulation window before going back to the Workspace. Do not save the simulation window waveform, as this consumes a large amount of storage space.



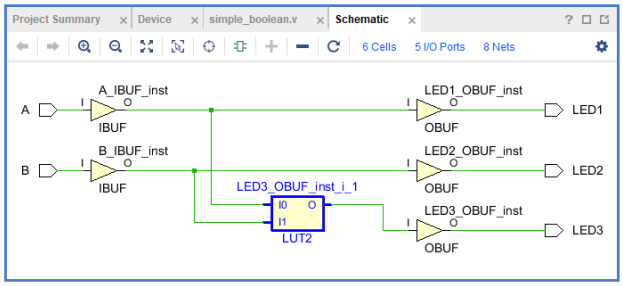
Synthesis

Logic synthesis transforms HDL code into an optimised set of logic gates to reduce the amount of hardware, and to efficiently perform the intended function.

Right-click on your Verilog design source file and select **Set as Top**. This option is disabled if the file is already the top module, and in such a case, proceed directly to the next step. In general, when there are multiple design and simulation modules, the "Set as Top" option selects the design, or simulation, modules to be considered when performing the different stages of the project flow.

In the **Flow Navigator** window, under **SYNTHESIS**, select **Run Synthesis**. While Vivado performs synthesis, the Project Status Bar at the top right provides an indication of the ongoing progress.

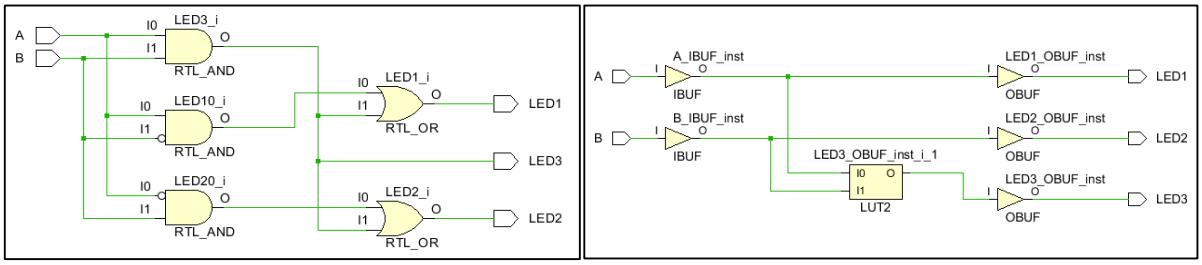
After the synthesis has been successfully completed, in the **Flow Navigator** window, under **SYNTHESIS**, expand **Open Synthesized Design**, and select **Schematics**. The schematic of the synthesised design will be generated and this synthesised circuit is an optimised version of the RTL schematic that was obtained



Click on the Look-up Table (LUT) that defines how the output LED3 behaves. The **Cell Properties** window will appear for that specific LUT. In the **Cell Properties** window for the LUT of **LED3**, open the **Truth Table** tab. Notice how for this simple example, this LUT is behaving as a simple AND gate.

UNDERSTANDING | TASK 5

Compare the optimised and non-optimised schematics. How is this optimised circuit equivalent to the SOP equations of the simple boolean design task?

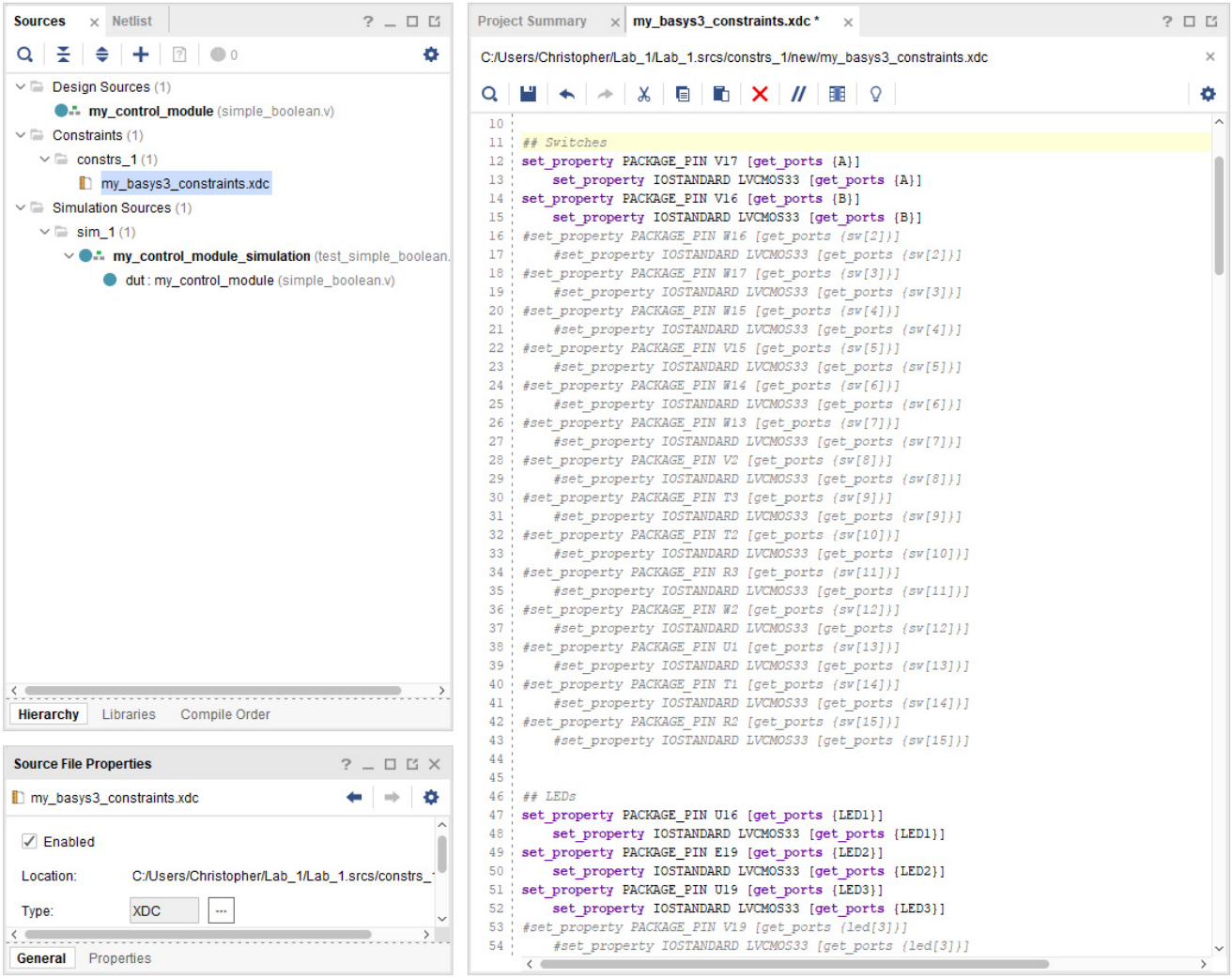


Design Constraints

Design constraints, such as timing and physical I/O pin mapping, must be defined before doing an implementation, following which the program can be downloaded to the FPGA device. Proceed with the following sequence:

- Expand **PROJECT MANAGER** in the **Flow Navigator** panel, and **click Add Sources**
- Select **Add or create constraints** and click **Next**
- Click on **Create File** and give the XDC file a file name, such as **my_basys3_constraints**. The XDC format stands for Xilinx Design Constraints here
- Open the **my_basys3_constraints.xdc** file from the **Sources** window. It will be an empty .xdc file.
- A template, known as the **Basys3_Master.xdc** is provided. Open that template using a basic text editor, such as notepad.
- Copy all the contents from that template to your **my_basys3_constraints.xdc**. All the lines are commented out by default.
- Link the signals (A, B, LED1, LED2, LED3) of your design, to some physical pins of the FPGA, by uncommenting relevant lines. Input signals can be linked to switches, whereas the output signals can be linked to LEDs, on the Basys3 development board.

An example of the above steps is shown below:



Implementation, Bitstream Generation and Program Download

The implementation phase will map the design to available physical resources on the FPGA hardware. In the **Flow Navigator** window, under **IMPLEMENTATION**, select **Run Implementation**. This will make use of the design constraint file that had been created earlier on.

After the implementation phase, there is a need to generate a file that can be downloaded to the FPGA. Such a file is called a bitstream file, and it consists of binary values 0's and 1's that tells the FPGA how to behave. In the **Flow Navigator** window, under **PROGRAM AND DEBUG**, select **Generate Bitstream**. A successful bitstream generation is the last step required before downloading the program to the FPGA.

Before using your Basys 3 development board, and to prevent potential damage to it, take note of the following recommendations and warnings to extend the longevity of the device:

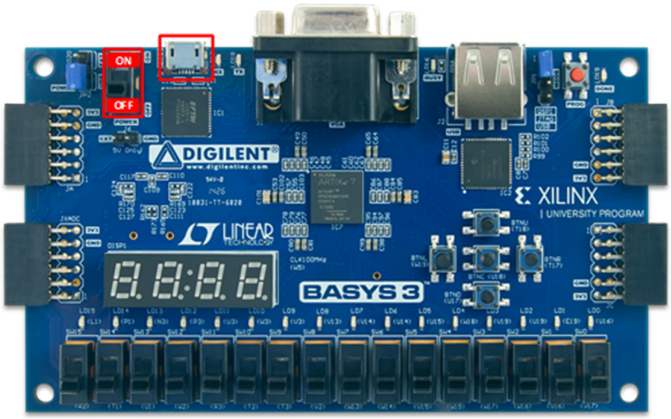
⚠ Make sure the Basys 3 development board is powered OFF by placing SW16 in the OFF position before connection to/removal from the USB port of the computer.

⚠ Carefully connect to the USB cable

xc7a35t_0

⚠ The chips on the board are electrostatic sensitive. Avoid touching them. Handle the board by the edges to prevent damage.

⚠ Make sure the board is not in contact with any metal components, whether above or below. Do not place any liquid sources near the FPGA board.



After connection of the Basys 3 development board to the computer, turn on the power by setting SW16 in the ON position. Test the functionality of your Basys 3 development board before downloading any program to it, according to instructions that will be provided during your lab session. If confirmed to be working, proceed with the following steps:

- Expand **Program and Debug** in the **Flow Navigator Panel**
- Expand **Open Hardware Manager**
- Click **Open Target**
- Select **Auto Connect.** In case connection fails, consider pressing the 'reset' button, or turn your device OFF for a few seconds and ON again, while ensuring that it is detected and installed on your computer. Then try **Auto Connect** again
- If successful, the **Program Device** will be enabled, and you will be able to select **xc7a35t_0**
- By default, if the bitstream was successfully generated, the path name in the **Bitstream file** is automatically provided
- Download the .bit file to the FPGA by clicking on **Program**

UNDERSTANDING | TASK 6

Your program will then be downloaded to the FPGA. Verify the functionality of the design by using the input devices you have assigned to A, B, and observing the output devices assigned to LED1, LED2 and LED3. Check what happens if the 'reset' pushbutton on the Basys 3 development board is pressed, or if power is loss for a short amount of time.

CLOSING NOTES FOR LAB 1

Now that you have successfully completed your FPGA design flow, one final practice task is provided to you for completion before ending the lab session. This practice task is not graded.

FINAL UNDERSTANDING | PRACTICE TASK FOR LAB 1

- **Create a new Vivado project from scratch. Do not reuse your existing project or design**
- The same design as described for the simple boolean design task need to be implemented, with the following exception: There is an additional switch C, and if this switch C is in the OFF state, it forces all the three LEDs to be in the OFF state. If the switch C is in the ON state, the design behaves exactly as described for the simple boolean design task. The switch C is to be mapped to SW[*Your birthday month + 3*] on the Basys 3 development board
- Simulate your design, as well as implement it on the Basys 3 development board

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 (Initialisation and subtask A)
- The five rightmost numerical values of your student matriculation number (Subtask B)
- The 2nd rightmost numerical value of your student matriculation number (Subtask C)
- The rightmost alphabet of your student matriculation number (Subtask C)

INITIALISATION

When the program starts, all 16 active-high switches (SW0 to SW15) are in the OFF position. All 16 active-high LEDs (LD0 to LD15) are also OFF. The seven segment displays must show the following patterns **exactly**, based on the **1st rightmost numerical value of your student matriculation number**:

1st Rightmost Numerical Value	0	1	2	3	4
Required 7-Segments Displays					

1st Rightmost Numerical Value	5	6	7	8	9
Required 7-Segments Displays					

SUBTASK A

Consider the 10 (ten) switches SW0 to SW9. Whenever any of these 10 switches are ON, the corresponding LED LD**X**, where **X** is a number ranging from 0 to 9, must be ON. Examples:

- If SW0 is ON, then LD0 must be ON
- If SW3, SW7 and SW9 are ON, then LD3, LD7 and LD9 must be ON

SW0 to SW9, and LD0 to LD9, must all be constraint.
SW10 to SW15, and LD10 to LD14, must be ignored (Do not put a constraint to switches SW10 to SW15 and LEDs LD10 to LD14).
LD15 requires constraint for SUBTASK B onwards.

SUBTASK B

Continuing from SUBTASK A, create your personal password based on the **five rightmost numerical values of your student matriculation number** (Ignore the alphabet character).

These five digits (May be less than five digits if you have duplicate numbers) will represent the switches that need to be ON, while all the other switches between SW0 to SW9 must be OFF, to be considered a correct password.

If the password entered by the user is the correct password, then LED LD15 must turn ON. LD15 is OFF whenever the password is incorrect.

SUBTASK C

When the password from SUBTASK B is correct and LD15 is ON, it is also required to display the rightmost alphabet of your student matriculation number on some specific anodes of the 7-segment displays. The character must be displayed **exactly** as indicated:

Rightmost Alphabet	A	B	E	H	J	L	M	N	R	U	W	X	Y
Required 7-Segments Character	A	b	E	H	J	L	n	n	r	U	W	H	y

The anode on which the character should be displayed is dependent on the **2nd rightmost numerical value** of your student matriculation number, as indicated in the table below:

2nd Rightmost Numerical Value	Anode AN3	Anode AN2	Anode AN1	Anode AN0
0	**ON**	**ON**	OFF	**ON**
1	**ON**	**ON**	OFF	OFF
2	**ON**	OFF	**ON**	**ON**
3	**ON**	OFF	**ON**	OFF
4	**ON**	OFF	OFF	**ON**
5	**ON**	OFF	OFF	OFF
6	OFF	**ON**	**ON**	**ON**
7	OFF	**ON**	**ON**	OFF
8	OFF	**ON**	OFF	**ON**
9	OFF	**ON**	OFF	OFF

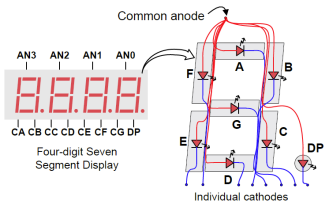
When the password from SUBTASK B is not correct, it is compulsory for the seven-segment displays to show the set of characters indicated in the INITIALISATION phase.

SUGGESTIONS

- Create a new Vivado project for this assignment, instead of continuing from your previous Vivado project
- This assignment can be fully completed by using only what you have learnt throughout lab session 1. It is not recommended to use contents not taught in this lab session, as this is meant to be a warming-up assignment
- The following will be taught in subsequent lectures / tutorials / labs, and are thus **not necessary** in lab 1:
    - if-else functions
    - always blocks
    - multi-bits vector

GETTING STARTED WITH THE SEVEN-SEGMENT DISPLAYS

There are 7 LED segments in each display, with an additional decimal point segment. They are respectively denoted by “seg[0]” to “seg[6]”, and “dp”, in the Basys_Master.xdc constraint file.



There are 4 seven-segment displays on the Basys 3 development board. Each one of the displays is controlled by a common anode pin, thus resulting in a total of 4 common anodes. These active-low pins are denoted as “an[3]” to “an[0]” in the Basys_Master.xdc constraint file. (For more information, you can refer to the Basys 3 reference manual, pages 14 to 16)

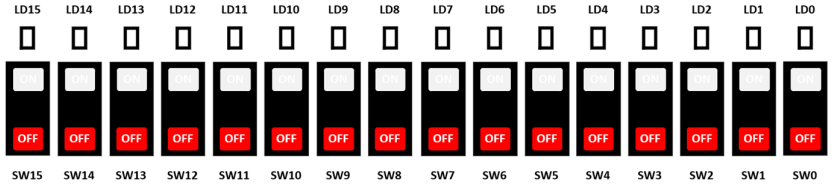
In your constraint file, it is compulsory to put constraints to the 8 segments (7 segments + decimal point) of the seven-segment display, and to the 4 anodes of the seven-segment display.

EXAMPLE:

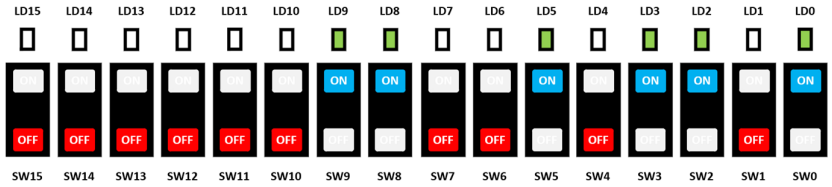
If your student matriculation number is A0159089Y, then:

1st rightmost numerical value:	9
Five rightmost numerical values:	59089
2nd rightmost numerical value:	8
Rightmost alphabet:	Y

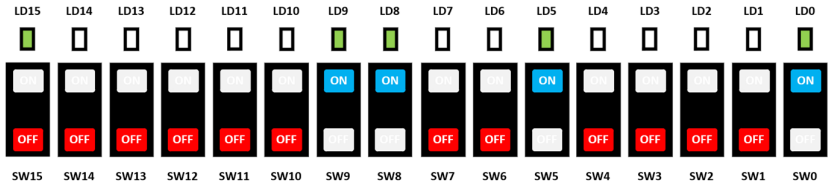
INITIALISATION



WRONG PASSWORD



CORRECT PASSWORD



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. In the uploaded archive, the codes (.v files) are important, not the waveforms (.wdb files). **The archive size should not exceed 4 MB in size for any lab assignments.** Follow the instructions given in the pdf: "Archive Project in Vivado 2018.02"

- **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 1 Graded Assignment -> Lab 1 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 in that extracted folder to see if it can be opened. **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 **Sunday 29th January 2023, 10:00 P.M.** Avoid uploading during the grace period of 2 hours

- A penalty of 25% applies for late submissions of up to 1 week.

- The late submission folder closes 1 week after the original deadline. **Late submissions are not accepted and not graded if a submission is found within the on-time folder, or if grading has already started on an earlier submitted file**. The late submission folder will be located at: CANVAS EE2026 -> Assignments -> Lab 1 Graded Assignment -> Lab 1 Submission (Late)

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 results during the lab sessions in such situations**

APPENDIX (COMPULSORY renaming before just 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.

Name	Archive Naming
ARIF KHALID	L1_Mon_AM_ARIF KHALID_983_Archive
ARTEMIS NGOH	L1_Wed_PM_ARTEMIS NGOH_342_Archive
AUNG PHONE NAING	L1_Mon_AM_AUNG PHONE NAING_363_Archive
BENJAMIN CHRISTOPHER TOH	L1_Wed_PM_BENJAMIN CHRISTOPHER TOH_702_Archive
BENJAMIN LONG WEI MING	L1_Mon_AM_BENJAMIN LONG WEI MING_953_Archive
BIAN RUI	L1_Wed_PM_BIAN RUI_671_Archive
BRANDON OWEN SJARIF	L1_Wed_PM_BRANDON OWEN SJARIF_179_Archive
BUI DUC THANH	L1_Wed_PM_BUI DUC THANH_820_Archive
BUI PHUONG NAM	L1_Wed_AM_BUI PHUONG NAM_451_Archive
CALEB CHAN JIA LE	L1_Wed_PM_CALEB CHAN JIA LE_569_Archive
CHAO YI-JU	L1_Wed_PM_CHAO YIJU_863_Archive
CHEAH HAO YI	L1_Wed_PM_CHEAH HAO YI_666_Archive
CHEN JUNHAN JERALD	L1_Wed_AM_CHEN JUNHAN JERALD_262_Archive
CHENVISUWAT NITA	L1_Mon_AM_CHENVISUWAT NITA_872_Archive
CHERAN LEE YEN-NING	L1_Wed_AM_CHERAN LEE YENNING_991_Archive
CHEW JUN WEI MAX	L1_Wed_PM_CHEW JUN WEI MAX_381_Archive
CHEW SI NING, KACEY	L1_Mon_AM_CHEW SI NING KACEY_433_Archive
CHIA YU XUAN	L1_Wed_AM_CHIA YU XUAN_884_Archive
CHIEW YI XIANG	L1_Wed_PM_CHIEW YI XIANG_756_Archive
CHONG YONG RUI	L1_Wed_PM_CHONG YONG RUI_594_Archive
CHUA ZHONG HENG	L1_Wed_PM_CHUA ZHONG HENG_908_Archive
CHUAH WAN JUIN	L1_Mon_AM_CHUAH WAN JUIN_377_Archive
CLEON CHENG RUI FENG	L1_Wed_PM_CLEON CHENG RUI FENG_796_Archive
CLEON LIEW GE WEI	L1_Mon_AM_CLEON LIEW GE WEI_788_Archive
DEEPANJALI DHAWAN	L1_Wed_PM_DEEPANJALI DHAWAN_498_Archive
DENNIS ONG QINKANG	L1_Wed_PM_DENNIS ONG QINKANG_605_Archive
DEXTER HOON YONG EN	L1_Wed_PM_DEXTER HOON YONG EN_166_Archive
DIVAKARAN MANUSHRI	L1_Wed_PM_DIVAKARAN MANUSHRI_824_Archive
DYLAN CHIA TIAN	L1_Wed_PM_DYLAN CHIA TIAN_729_Archive
DYLAN HO SHU JIE	L1_Wed_AM_DYLAN HO SHU JIE_536_Archive
EE JIA EN JARED	L1_Wed_AM_EE JIA EN JARED_035_Archive
ER JUN ZE	L1_Wed_PM_ER JUN ZE_233_Archive
EU ZHENG XI	L1_Mon_AM_EU ZHENG XI_767_Archive
EUGENE ANG JIA SHING	L1_Wed_PM_EUGENE ANG JIA SHING_189_Archive
FARIS HAMID SIRRAJ	L1_Mon_AM_FARIS HAMID SIRRAJ_265_Archive
FELICIA BEATRICE BUDIAWAN LAUW	L1_Wed_AM_FELICIA BEATRICE BUDIAWAN_855_Archive
FOO YANG WEI, JEROME	L1_Mon_AM_FOO YANG WEI JEROME_012_Archive
GAO YUN FAN	L1_Mon_AM_GAO YUN FAN_187_Archive
GARLAPATI SAI CHAITANYA	L1_Mon_AM_GARLAPATI SAI CHAITANYA_949_Archive
GAU KIAT LOK JERALD	L1_Mon_AM_GAU KIAT LOK JERALD_058_Archive
GOH JING HONG	L1_Wed_PM_GOH JING HONG_722_Archive
GOH YI XUAN	L1_Mon_AM_GOH YI XUAN_250_Archive

GRACE ZHU XING YU	L1_Wed_PM_GRACE ZHU XING YU_936_Archive
GUAN XIAO	L1_Mon_AM_GUAN XIAO_837_Archive
HAMADA MASAHIRO	L1_Mon_AM_HAMADA MASAHIRO_120_Archive
HANG TIAN	L1_Wed_PM_HANG TIAN_725_Archive
HO ZHAN RUI, GLENN	L1_Wed_AM_HO ZHAN RUI GLENN_662_Archive
HONG LIN SHANG	L1_Wed_PM_HONG LIN SHANG_890_Archive
HOO TENG JUAN	L1_Mon_AM_HOO TENG JUAN_494_Archive
HUI YU CONG	L1_Mon_AM_HUI YU CONG_397_Archive
HUNG HIN WANG, CLEMENT	L1_Mon_AM_HUNG HIN WANG CLEMENT_615_Archive
JAVIENNE YEO MYN	L1_Wed_AM_JAVIENNE YEO MYN_441_Archive
JOHN TOH JIA JUN	L1_Mon_AM_JOHN TOH JIA JUN_603_Archive
KHAIRUL AIZAT B MD HALIM	L1_Wed_PM_KHAIRUL AIZAT B MD HALIM_315_Archive
KHOO YOU RUN	L1_Mon_AM_KHOO YOU RUN_353_Archive
KIM TAE WON	L1_Wed_AM_KIM TAE WON_392_Archive
KISHORE S/O ASOKAN	L1_Wed_PM_KISHORE SO ASOKAN_812_Archive
KOH CHEE HENG	L1_Wed_PM_KOH CHEE HENG_333_Archive
KOH JING JIE MARCUS	L1_Mon_AM_KOH JING JIE MARCUS_625_Archive
KOH NGIAP HIN	L1_Mon_AM_KOH NGIAP HIN_229_Archive
KWUA CHUN REN	L1_Wed_AM_KWUA CHUN REN_262_Archive
LEE JUN HAO BRYAN	L1_Wed_PM_LEE JUN HAO BRYAN_090_Archive
LEE JUN HUI, ANSENN	L1_Wed_PM_LEE JUN HUI ANSENN_865_Archive
LEE ZHI XUAN	L1_Wed_AM_LEE ZHI XUAN_363_Archive
LEE ZHI ZHONG, MOSES	L1_Wed_PM_LEE ZHI ZHONG MOSES_669_Archive
LEI HAO	L1_Wed_PM_LEI HAO_291_Archive
LEONARDO ONG DINGCHAO	L1_Wed_AM_LEONARDO ONG DINGCHAO_944_Archive
LEONG HOI MING, JOSHUA	L1_Wed_AM_LEONG HOI MING JOSHUA_256_Archive
LEONG HUEN WENG	L1_Mon_AM_LEONG HUEN WENG_012_Archive
LEONG HUNG REY	L1_Wed_PM_LEONG HUNG REY_590_Archive
LEONG SONG ZHU, OWEN	L1_Wed_PM_LEONG SONG ZHU OWEN_408_Archive
LEONG YAT PANG	L1_Wed_PM_LEONG YAT PANG_470_Archive
LI MINGYUAN	L1_Wed_AM_LI MINGYUAN_670_Archive
LIM HONG YAO	L1_Mon_AM_LIM HONG YAO_619_Archive
LIM ZHENG RONG	L1_Wed_AM_LIM ZHENG RONG_047_Archive
LINUS PUAH JIA HE	L1_Wed_AM_LINUS PUAH JIA HE_811_Archive
LIONG WEI YONG, DEEN	L1_Wed_AM_LIONG WEI YONG DEEN_930_Archive
LIU XIAOGE	L1_Mon_AM_LIU XIAOGE_254_Archive
LOH JOO HOE	L1_Mon_AM_LOH JOO HOE_456_Archive
LOH YUAN LONG, KEDRIAN	L1_Mon_AM_LOH YUAN LONG KEDRIAN_084_Archive
LU BINGYUAN	L1_Mon_AM_LU BINGYUAN_325_Archive
MAN JUNCHENG	L1_Mon_AM_MAN JUNCHENG_844_Archive
MANOJ DORAIRAJAN	L1_Wed_PM_MANOJ DORAIRAJAN_508_Archive
MATTHEW LIU ZHEN JIE	L1_Wed_PM_MATTHEW LIU ZHEN JIE_439_Archive
MITCH MALVIN	L1_Wed_PM_MITCH MALVIN_911_Archive
MUSTAFA ANIS HUSSAIN	L1_Wed_AM_MUSTAFA ANIS HUSSAIN_072_Archive
MUTHUKRISHNAN NAVYA	L1_Wed_PM_MUTHUKRISHNAN NAVYA_130_Archive
MUTHYA NARAYANACHARY AKHIL	L1_Mon_AM_MUTHYA NARAYANACHARY AKHI_509_Archive
NAM SANGJUN	L1_Wed_PM_NAM SANGJUN_260_Archive
NAZRUL SYAHMI BIN MURAD	L1_Wed_PM_NAZRUL SYAHMI BIN MURAD_711_Archive
NG DE QI	L1_Mon_AM_NG DE QI_171_Archive
NG KAI WEN	L1_Mon_AM_NG KAI WEN_144_Archive
NG LIXUAN NIXON	L1_Wed_PM_NG LIXUAN NIXON_667_Archive

NG YAN ZHEN	L1_Mon_AM_NG YAN ZHEN_909_Archive
NGUYEN DUC THANG	L1_Wed_PM_NGUYEN DUC THANG_361_Archive
NGUYEN QUANG ANH	L1_Wed_AM_NGUYEN QUANG ANH_912_Archive
NICHOLAS H GOH MAOWEN	L1_Mon_AM_NICHOLAS H GOH MAOWEN_475_Archive
NIKHIL SHASHIDHAR	L1_Mon_AM_NIKHIL SHASHIDHAR_992_Archive
NING ZHI YAN	L1_Wed_PM_NING ZHI YAN_096_Archive
OH YI XIU, WILSON	L1_Wed_AM_OH YI XIU WILSON_510_Archive
ONG CHUAN KAI	L1_Mon_AM_ONG CHUAN KAI_208_Archive
ONG HEE JET	L1_Mon_AM_ONG HEE JET_579_Archive
ONG JUN LIN JEREMIAH	L1_Wed_AM_ONG JUN LIN JEREMIAH_679_Archive
ONG SHAO YONG	L1_Mon_AM_ONG SHAO YONG_220_Archive
OONG JIN RONG JARED	L1_Wed_PM_OONG JIN RONG JARED_178_Archive
OU NINGXIANG	L1_Wed_AM_OU NINGXIANG_191_Archive
OW YONG JIN XUAN	L1_Wed_AM_OW YONG JIN XUAN_649_Archive
PARANJAPE INDRANEEL RAJEEV	L1_Wed_PM_PARANJAPE INDRANEEL RAJEE_412_Archive
POOBALAN AATMIKA LAKSHMI	L1_Wed_PM_POOBALAN AATMIKA LAKSHMI_701_Archive
RACHEL FONG RUI YUAN	L1_Mon_AM_RACHEL FONG RUI YUAN_567_Archive
REYNOLD SAMEL LAM	L1_Wed_PM_REYNOLD SAMEL LAM_216_Archive
RICHARD KURNIAWAN	L1_Mon_AM_RICHARD KURNIAWAN_030_Archive
RICHARD LOONG CHENG JUN	L1_Wed_PM_RICHARD LOONG CHENG JUN_496_Archive
RYAN TAN	L1_Mon_AM_RYAN TAN_565_Archive
RYUJI KOW JIE SI	L1_Wed_PM_RYUJI KOW JIE SI_117_Archive
SAMUEL TAN SZE WEE	L1_Wed_AM_SAMUEL TAN SZE WEE_575_Archive
SAUNG NAYCHI MIN	L1_Wed_AM_SAUNG NAYCHI MIN_779_Archive
SEBASTIAN SOEWANTO	L1_Wed_PM_SEBASTIAN SOEWANTO_268_Archive
SEET SZE WEN	L1_Mon_AM_SEET SZE WEN_125_Archive
SENTHILKUMAR SRIRAM	L1_Mon_AM_SENTHILKUMAR SRIRAM_934_Archive
SEOW RUI SHENG	L1_Mon_AM_SEOW RUI SHENG_454_Archive
SHAN YUXUAN	L1_Wed_PM_SHAN YUXUAN_473_Archive
SHANNEN TAN	L1_Wed_AM_SHANNEN TAN_713_Archive
SIM JUSTIN	L1_Mon_AM_SIM JUSTIN_898_Archive
SIM QIAN HUI	L1_Wed_AM_SIM QIAN HUI_289_Archive
SIUT WAI HONG, CLEMENT	L1_Mon_AM_SIUT WAI HONG CLEMENT_175_Archive
SONG ZIJIN	L1_Wed_PM_SONG ZIJIN_699_Archive
STEFAN CHOO BIN HAO	L1_Wed_AM_STEFAN CHOO BIN HAO_098_Archive
STEVEN ANTYA ORVALA WASKITO	L1_Mon_AM_STEVEN ANTYA ORVALA WASKI_459_Archive
SURESH ABIJITH RAM	L1_Wed_PM_SURESH ABIJITH RAM_215_Archive
SYED OMAR ZORAN	L1_Wed_AM_SYED OMAR ZORAN_260_Archive
TAN HSIEN RONG	L1_Wed_AM_TAN HSIEN RONG_011_Archive
TAN JIN SHENG, BRIAN	L1_Wed_AM_TAN JIN SHENG BRIAN_260_Archive
TAN JUN HAO, KENNETH	L1_Mon_AM_TAN JUN HAO KENNETH_519_Archive
TAN TZE LOONG	L1_Wed_AM_TAN TZE LOONG_867_Archive
TAN WAN LIN	L1_Mon_AM_TAN WAN LIN_587_Archive
TAN YI ZHE	L1_Wed_AM_TAN YI ZHE_674_Archive
TAN YU XIANG, GARETH	L1_Mon_AM_TAN YU XIANG GARETH_682_Archive
TAN ZI XI	L1_Wed_AM_TAN ZI XI_046_Archive
TANG JUN MEI, SHANICE	L1_Wed_AM_TANG JUN MEI SHANICE_739_Archive
TAY JIUN YUAN	L1_Wed_PM_TAY JIUN YUAN_398_Archive
TAY YU YANG	L1_Mon_AM_TAY YU YANG_039_Archive
TENG YUAN-KAI	L1_Wed_AM_TENG YUANKAI_862_Archive
TEO JUN YONG ADEN	L1_Mon_AM_TEO JUN YONG ADEN_600_Archive

TEOH JING YANG	L1_Wed_AM_TEOH JING YANG_566_Archive
THANT AUNG HTET NYAN	L1_Wed_AM_THANT AUNG HTET NYAN_654_Archive
TIAN SHIXING	L1_Wed_AM_TIAN SHIXING_830_Archive
TOH HONG FENG	L1_Mon_AM_TOH HONG FENG_218_Archive
TOH MING CHUN	L1_Mon_AM_TOH MING CHUN_573_Archive
TONG ZHENG HONG	L1_Mon_AM_TONG ZHENG HONG_407_Archive
TRICIA BOO KOH WEI PING	L1_Mon_AM_TRICIA BOO KOH WEI PING_477_Archive
TU HUIYU	L1_Mon_AM_TU HUIYU_423_Archive
UDAYAKUMAR NIVETHA	L1_Wed_PM_UDAYAKUMAR NIVETHA_182_Archive
VARATHARAJU VIGNESH	L1_Wed_PM_VARATHARAJU VIGNESH_677_Archive
VARNIKA SRIVASTAVA	L1_Wed_AM_VARNIKA SRIVASTAVA_513_Archive
VISHNU	L1_Wed_PM_VISHNU_211_Archive
VU VAN DUNG	L1_Wed_AM_VU VAN DUNG_825_Archive
WANG HAOYANG	L1_Wed_AM_WANG HAOYANG_534_Archive
WANG SILANG	L1_Wed_AM_WANG SILANG_130_Archive
WANG TINGJIA	L1_Mon_AM_WANG TINGJIA_501_Archive
WANG YONGBIN	L1_Mon_AM_WANG YONGBIN_968_Archive
WILSON LEE JUN WEI	L1_Wed_PM_WILSON LEE JUN WEI_735_Archive
WINSTON LIM CHER HONG	L1_Wed_AM_WINSTON LIM CHER HONG_359_Archive
WONG ZHONG XIANG	L1_Wed_AM_WONG ZHONG XIANG_929_Archive
WOO KAI NING	L1_Mon_AM_WOO KAI NING_128_Archive
WOO WEN JUN	L1_Wed_PM_WOO WEN JUN_636_Archive
WU ZHEN	L1_Wed_PM_WU ZHEN_039_Archive
YAO HE	L1_Wed_PM_YAO HE_485_Archive
YEO MENG HAN	L1_Mon_AM_YEO MENG HAN_360_Archive
YONG SHAN LING	L1_Mon_AM_YONG SHAN LING_973_Archive
ZENG ZIQIU	L1_Wed_PM_ZENG ZIQIU_720_Archive
ZHAI YUXIN	L1_Wed_PM_ZHAI YUXIN_744_Archive
ZHANG WENZE	L1_Wed_AM_ZHANG WENZE_047_Archive
ZHANG ZHITONG	L1_Wed_AM_ZHANG ZHITONG_202_Archive
ZHAO LIXIUQI	L1_Wed_AM_ZHAO LIXIUQI_464_Archive
ZHOU HUIQI	L1_Wed_AM_ZHOU HUIQI_364_Archive
ZHOU KAIWEN	L1_Mon_AM_ZHOU KAIWEN_651_Archive