EE2026: DIGITAL DESIGN

Academic Year 2020-2021, 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 [VENUE: Digital Electronics Lab E4-03-07]:

- 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.

- All lab sessions require that you have **carefully reviewed the relevant lecture and tutorial materials before attendance**. Contents taught during the theory classes, with emphasis on the Verilog language and structure, will directly be applied to solve practical problems during the lab sessions.

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 [LUMINUS SUBMISSION: MONDAY 1st FEBRUARY 2021, NOON]:

- To display a specific character on the 7-segment display(s) of the Basys 3 development board if a password is correct

Further 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.

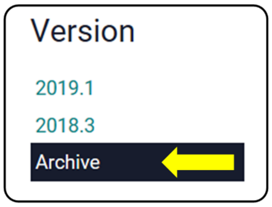
Software Weblink

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

Software Name [Mac version is not supported – Use Bootcamp for Windows]

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.*

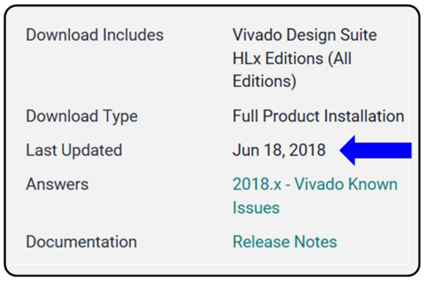


Version

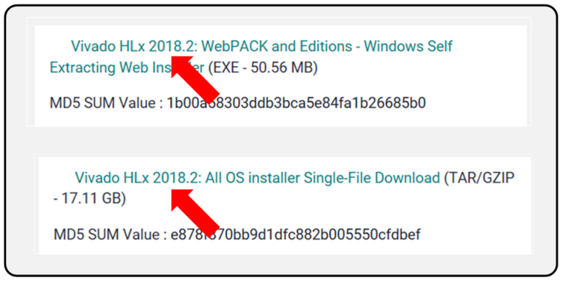
2019.1

2018.3

Archive



Download Includes	Vivado Design Suite HLx Editions (All Editions)
Download Type	Full Product Installation
Last Updated	Jun 18, 2018
Answers	2018.x - Vivado Known Issues
Documentation	Release Notes



Vivado HLx 2018.2: WebPACK and Editions - Windows Self Extracting Web Installer (EXE - 50.56 MB)

MD5 SUM Value : 1b00a68303ddb3bca5e84fa1b26685b0

Vivado HLx 2018.2: All OS installer Single-File Download (TAR/GZIP - 17.11 GB)

MD5 SUM Value : e878f870bb9d1dfc882b005550cfdbef

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.

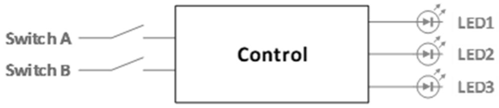
Post-Installation

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				$\bar{A}\bar{B}$
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**:

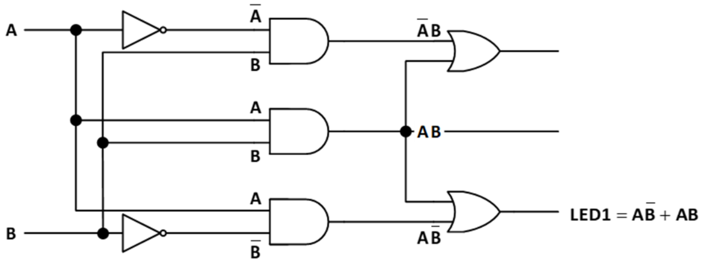
$$\mathbf{LED1} = \boldsymbol{A\bar{B} + AB}$$

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

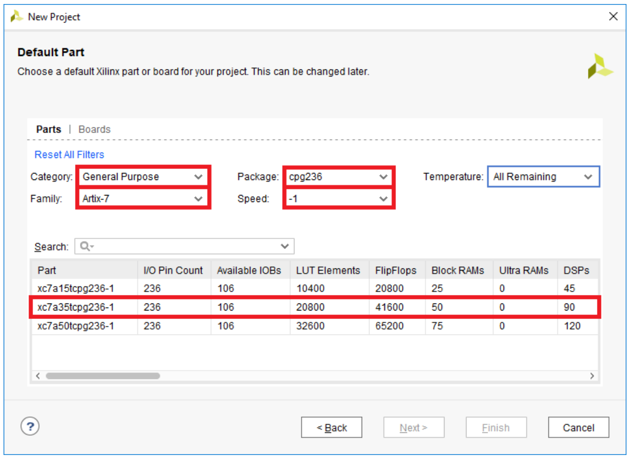
Project Type: Select **RTL Project**, and uncheck "**Do not specify sources at this time**"

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



Default Part

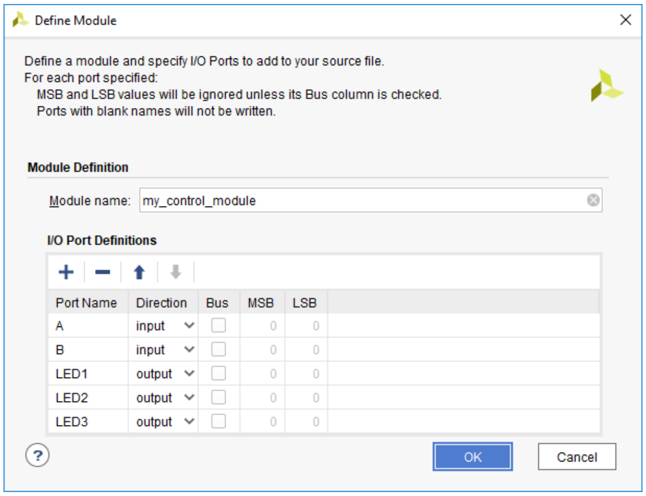
Choose a default Xilinx part or board for your project. This can be changed later.

Category: General Purpose
Family: Artix-7
Package: cpg236
Speed: -1
Temperature: All Remaining
Search:

Part	I/O Pin Count	Available IOBs	LUT Elements	FlipFlops	Block RAMs	Ultra RAMs	DSPs
xc7a15tcpg236-1	236	106	10400	20800	25	0	45
xc7a35tcpg236-1	236	106	20800	41600	50	0	90
xc7a50tcpg236-1	236	106	32600	65200	75	0	120

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.



Define a module and specify I/O Ports to add to your source file.
For each port specified:
MSB and LSB values will be ignored unless its Bus column is checked.
Ports with blank names will not be written.

Module Definition

Module name: my_control_module

I/O Port Definitions

Port Name	Direction	Bus	MSB	LSB
A	input	□	0	0
B	input	□	0	0
LED1	output	□	0	0
LED2	output	□	0	0
LED3	output	□	0	0

Using Vivado Text Editor to Write Verilog HDL Code

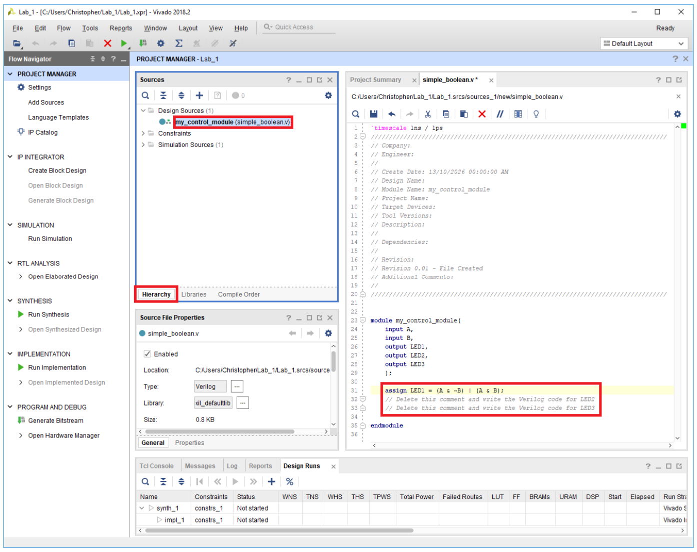
Open the module that has been created by double clicking on it in the Sources window

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	$\mathbf{\bar{A}}$	~
XOR	$\mathbf{A \oplus B}$	^



Lab_1 - [C:/Users/Christopher/Lab_1/Lab_1.xpr] - Vivado 2018.2

Sources: Design Sources (1) — my_control_module (simple_boolean.v); Constraints; Simulation Sources (1)

Source File Properties: simple_boolean.v — ☑ Enabled; Location: C:/Users/Christopher/Lab_1/Lab_1.srcs/source; Type: Verilog; Library: xil_defaultlib; Size: 0.8 KB

simple_boolean.v * — C:/Users/Christopher/Lab_1/Lab_1.srcs/sources_1/new/simple_boolean.v

```
1   `timescale 1ns / 1ps
2   //////////////////////////////////////////////////////////////////////////////////
3   // Company:
4   // Engineer:
5   //
6   // Create Date: 13/10/2026 00:00:00 AM
7   // Design Name:
8   // Module Name: my_control_module
9   // Project Name:
10  // Target Devices:
11  // Tool Versions:
12  // Description:
13  //
14  // Dependencies:
15  //
16  // Revision:
17  // Revision 0.01 - File Created
18  // Additional Comments:
19  //
20  //////////////////////////////////////////////////////////////////////////////////
21
22
23  module my_control_module(
24      input A,
25      input B,
26      output LED1,
27      output LED2,
28      output LED3
29      );
30
31      assign LED1 = (A & ~B) | (A & B);
32      // Delete this comment and write the Verilog code for LED2
33      // Delete this comment and write the Verilog code for LED3
34
35  endmodule
36
```

Design Runs:

Name	Constraints	Status	WNS	TNS	WHS	THS	TPWS	Total Power	Failed Routes	LUT	FF	BRAMs	URAM	DSP	Start	Elapsed	Run Stra
synth_1	constrs_1	Not started															Vivado S
impl_1	constrs_1	Not started															Vivado I

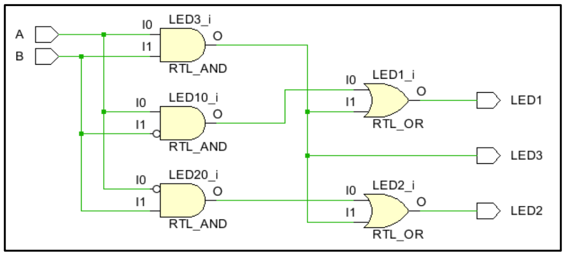
The **assign** statement causes the left hand side of the expression to be updated *every* time there is a change on the right hand side of the expression. It is therefore called a *continuous assignment* statement, describing combinational logic whereby the output on the left is a function of current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

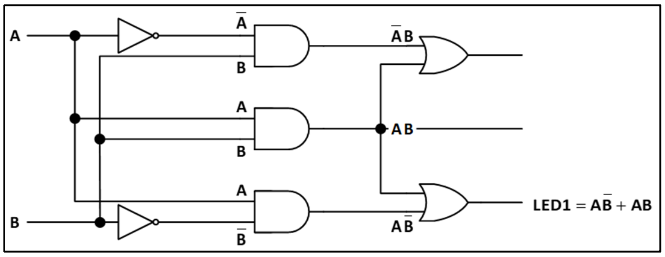
The statements on line 31 till line 33 execute concurrently. This is in contrast to sequential execution of statements in a computer programming language such as C, or procedural assignment that will be taught in subsequent lab sessions.

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?



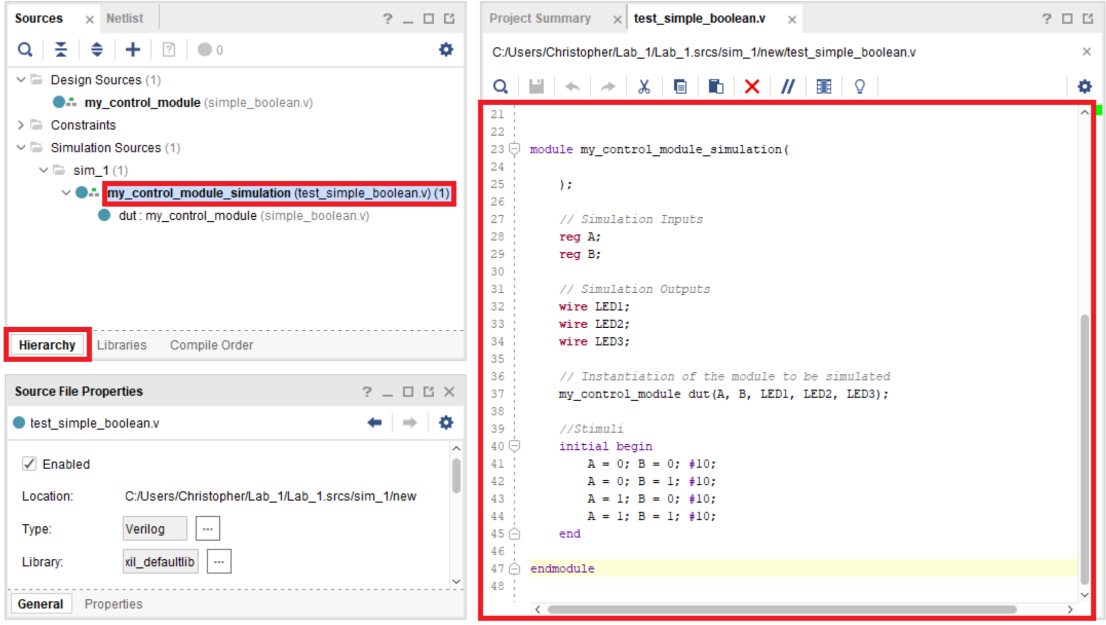


Testbench and Behavioural Simulation

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:



Sources
- Design Sources (1)
  - my_control_module (simple_boolean.v)
- Constraints
- Simulation Sources (1)
  - sim_1 (1)
    - my_control_module_simulation (test_simple_boolean.v) (1)
      - dut : my_control_module (simple_boolean.v)

Source File Properties

test_simple_boolean.v

☑ Enabled

Location:	C:/Users/Christopher/Lab_1/Lab_1.srcs/sim_1/new
Type:	Verilog
Library:	xil_defaultlib

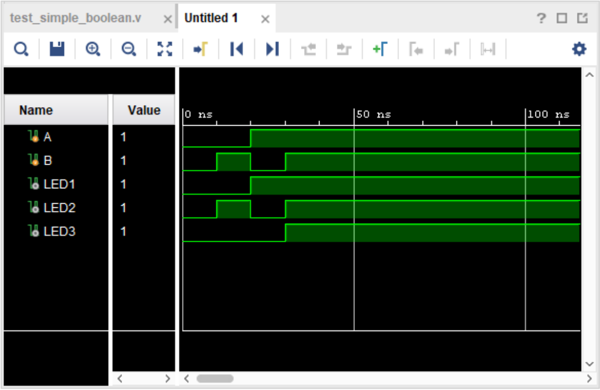
C:/Users/Christopher/Lab_1/Lab_1.srcs/sim_1/new/test_simple_boolean.v

```verilog


module my_control_module_simulation(

    );

    // Simulation Inputs
    reg A;
    reg B;

    // Simulation Outputs
    wire LED1;
    wire LED2;
    wire LED3;

    // Instantiation of the module to be simulated
    my_control_module dut(A, B, LED1, LED2, LED3);

    //Stimuli
    initial begin
        A = 0; B = 0; #10;
        A = 0; B = 1; #10;
        A = 1; B = 0; #10;
        A = 1; B = 1; #10;
    end

endmodule

```

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.



Name	Value
A	1
B	1
LED1	1
LED2	1
LED3	1

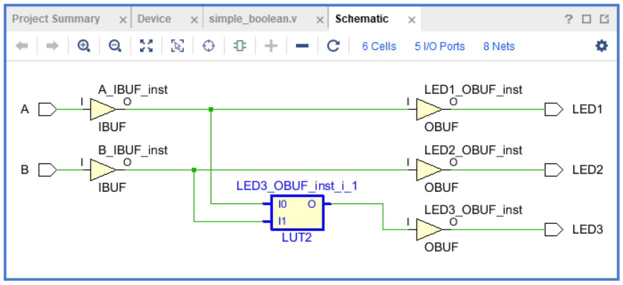
Synthesis

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.

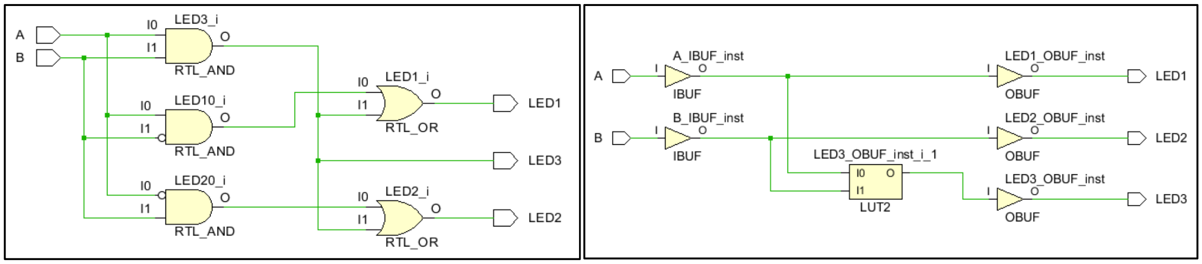
After the synthesis has been successfully completed, in the **Flow Navigator** window, under **SYNTHESIS**, expand **Open Synthesized Design**, and select **Schematics**. The schematic of the synthesised design will be generated and this synthesised circuit is an optimised version of the RTL schematic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

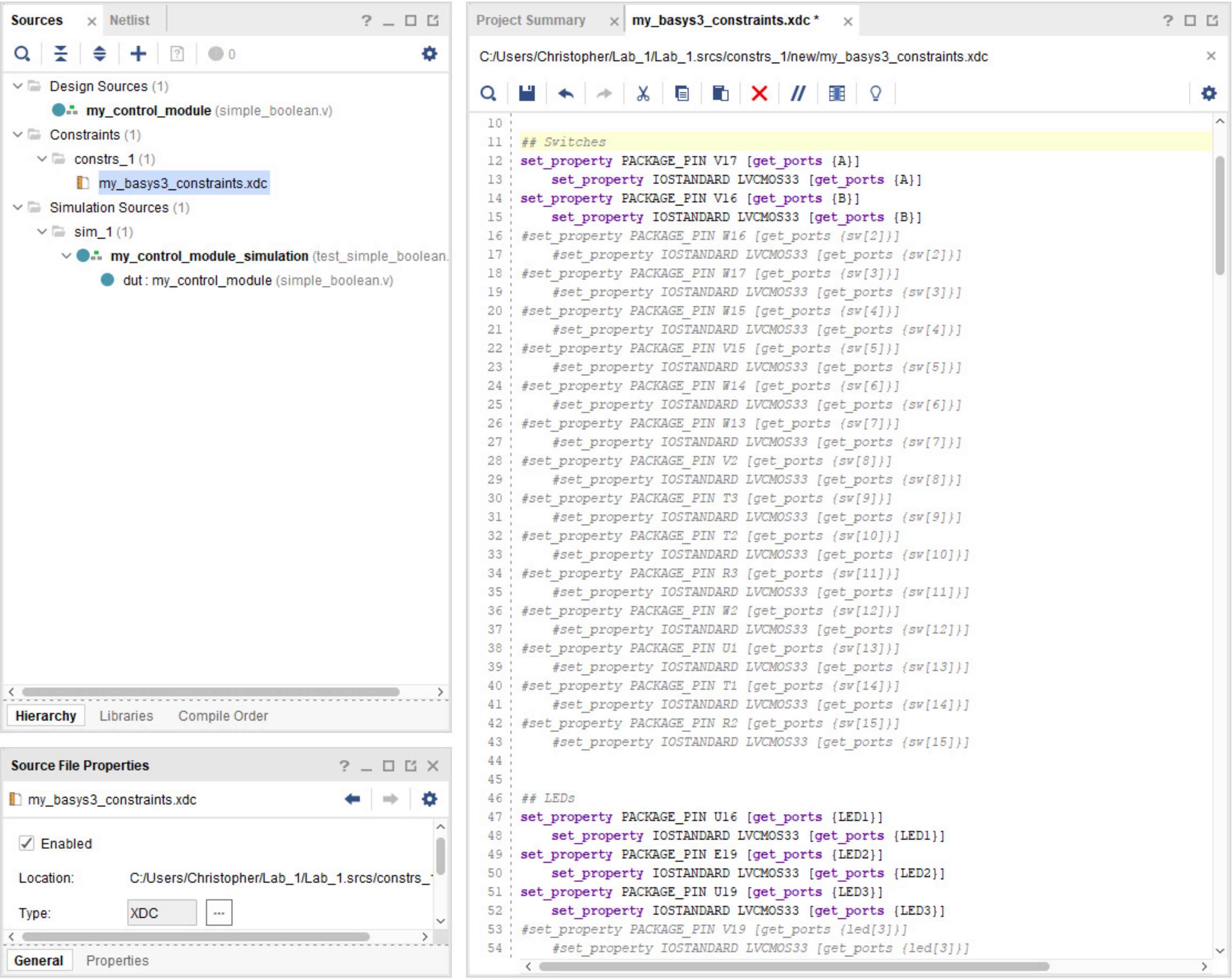


Design Constraints

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:


Sources
Netlist
Design Sources (1)
my_control_module (simple_boolean.v)
Constraints (1)
constrs_1 (1)
my_basys3_constraints.xdc
Simulation Sources (1)
sim_1 (1)
my_control_module_simulation (test_simple_boolean.
dut : my_control_module (simple_boolean.v)
Hierarchy
Libraries
Compile Order
Source File Properties
my_basys3_constraints.xdc
Enabled
General
Properties
Project Summary
my_basys3_constraints.xdc *

Location: C:/Users/Christopher/Lab_1/Lab_1.srcs/constrs_

Type: XDC

C:/Users/Christopher/Lab_1/Lab_1.srcs/constrs_1/new/my_basys3_constraints.xdc

```

## Switches
set_property PACKAGE_PIN V17 [get_ports {A}]
    set_property IOSTANDARD LVCMOS33 [get_ports {A}]
set_property PACKAGE_PIN V16 [get_ports {B}]
    set_property IOSTANDARD LVCMOS33 [get_ports {B}]
#set_property PACKAGE_PIN W16 [get_ports {sw[2]}]
    #set_property IOSTANDARD LVCMOS33 [get_ports {sw[2]}]
#set_property PACKAGE_PIN W17 [get_ports {sw[3]}]
    #set_property IOSTANDARD LVCMOS33 [get_ports {sw[3]}]
#set_property PACKAGE_PIN W15 [get_ports {sw[4]}]
    #set_property IOSTANDARD LVCMOS33 [get_ports {sw[4]}]
#set_property PACKAGE_PIN V15 [get_ports {sw[5]}]
    #set_property IOSTANDARD LVCMOS33 [get_ports {sw[5]}]
#set_property PACKAGE_PIN W14 [get_ports {sw[6]}]
    #set_property IOSTANDARD LVCMOS33 [get_ports {sw[6]}]
#set_property PACKAGE_PIN W13 [get_ports {sw[7]}]
    #set_property IOSTANDARD LVCMOS33 [get_ports {sw[7]}]
#set_property PACKAGE_PIN V2 [get_ports {sw[8]}]
    #set_property IOSTANDARD LVCMOS33 [get_ports {sw[8]}]
#set_property PACKAGE_PIN T3 [get_ports {sw[9]}]
    #set_property IOSTANDARD LVCMOS33 [get_ports {sw[9]}]
#set_property PACKAGE_PIN T2 [get_ports {sw[10]}]
    #set_property IOSTANDARD LVCMOS33 [get_ports {sw[10]}]
#set_property PACKAGE_PIN R3 [get_ports {sw[11]}]
    #set_property IOSTANDARD LVCMOS33 [get_ports {sw[11]}]
#set_property PACKAGE_PIN W2 [get_ports {sw[12]}]
    #set_property IOSTANDARD LVCMOS33 [get_ports {sw[12]}]
#set_property PACKAGE_PIN U1 [get_ports {sw[13]}]
    #set_property IOSTANDARD LVCMOS33 [get_ports {sw[13]}]
#set_property PACKAGE_PIN T1 [get_ports {sw[14]}]
    #set_property IOSTANDARD LVCMOS33 [get_ports {sw[14]}]
#set_property PACKAGE_PIN R2 [get_ports {sw[15]}]
    #set_property IOSTANDARD LVCMOS33 [get_ports {sw[15]}]


## LEDs
set_property PACKAGE_PIN U16 [get_ports {LED1}]
    set_property IOSTANDARD LVCMOS33 [get_ports {LED1}]
set_property PACKAGE_PIN E19 [get_ports {LED2}]
    set_property IOSTANDARD LVCMOS33 [get_ports {LED2}]
set_property PACKAGE_PIN U19 [get_ports {LED3}]
    set_property IOSTANDARD LVCMOS33 [get_ports {LED3}]
#set_property PACKAGE_PIN V19 [get_ports {led[3]}]
    #set_property IOSTANDARD LVCMOS33 [get_ports {led[3]}]
```

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.

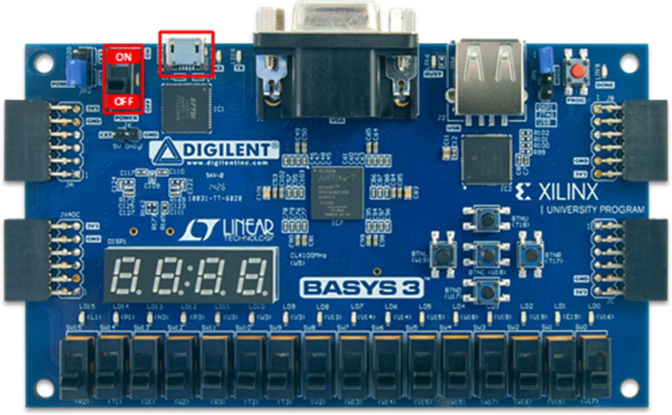
Before using your Basys 3 development board, and to prevent potential damage to it, take note of the following recommendations and warnings to extend the longevity of the device:

⚠ 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.

⚠ Do not force in the micro-USB cable upside down to the Basys 3 development board, as this will damage your micro-USB port and device. **Carefully connect to the micro-USB cable in the correct orientation.**
[**Common cause of board damage in EE2026 labs – Students are required to buy replacements in cases of negligence**]

⚠ 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, but you need to inform your guiding G.A. of the task completion.

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

ASSIGNMENT

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, and all other leds are OFF
- If SW3, SW7 and SW9 are ON, then LD3, LD7and LD9 must be ON, and all other leds are OFF
- If SW0 to SW9 are ON, then LD0 to LD9 must be ON, and all other leds are OFF

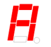
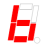
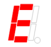
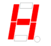
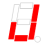
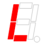
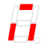
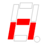
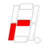
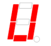
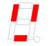
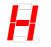
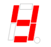
SW10 to SW15, and LD10 to LD15, must be ignored. Do not put a constraint to switches SW10 to SW15 and leds LD10 to LD15.

Continuing from the above task of lighting up the leds corresponding their respective switches, you are required to display a specific character on specific seven-segment displays based on your student matriculation number.

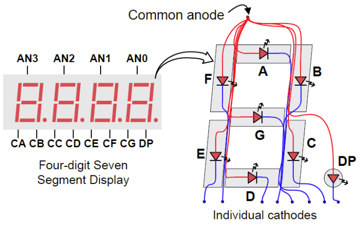
After you have done so, modify your code such that the latter only happens when a 10-bit value (password) is correctly set through the usage of these exact 10 switches: SW0 to SW9. If the 10-bit value is wrong, all the seven-segment displays do not show anything.

REQUIREMENT BASED ON YOUR STUDENT MATRICULATION NUMBER

The character to display is tabulated below:

Last character of your student matriculation number	A	B	E	H	J	L	M	N	R	U	W	X	Y
Required character on the seven-segment display													

There are 7 LED segments in each display, with an additional decimal point. 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.

Based on the rightmost numerical value of your student matriculation number, hardcode the four anode values according to the table below:

Last numerical value of your student matriculation number	AN3	AN2	AN1	AN0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

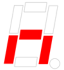
The correct password that allows the seven-segment displays to show character(s) is determined as follows:

- From your student matriculation number, consider the 5 rightmost numerical digits (Ignore the alphabet).
- These 5 digits will represent the active-high switches that need to be ON, while all the other active-high switches between SW0 to SW9 must be OFF, in order to be considered a correct password.

If the password is incorrect, it is compulsory for all the seven-segment displays to not show anything.

EXAMPLE 1 OF REQUIRED DISPLAYS BASED ON YOUR STUDENT MATRICULATION NUMBER

For example, if your student matriculation number is A3456789N, then it means that the character to appear on the display is:

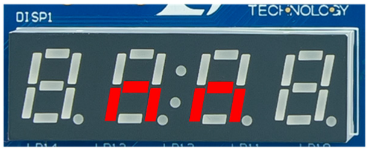


And since the rightmost numerical value of the student matriculation number is 9, AN3 will be 1, AN2 will be 0, AN1 will be 0, and AN0 will be 1. A value of 0 to a common anode makes the specific display active. Hence, with a value of 1001 to the 4 common anodes, the two middle displays will be active (on).

Furthermore, the five rightmost numerical values of the student matriculation number are: 5, 6, 7, 8, 9. Hence, the correct password that allows the seven-segment displays to be on would be:

LD9	LD8	LD7	LD6	LD5	LD4	LD3	LD2	LD1	LD0
ON	ON	ON	ON	ON	OFF	OFF	OFF	OFF	OFF
SW9	SW8	SW7	SW6	SW5	SW4	SW3	SW2	SW1	SW0
ON	ON	ON	ON	ON	OFF	OFF	OFF	OFF	OFF

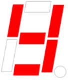
The following value should appear on DISP1 (4 seven-segment displays) when the password is correct:



When the password is wrong (not having the exact 10 switches as tabulated above to be ON/OFF) nothing is displayed on DISP1.

EXAMPLE 2 OF REQUIRED DISPLAYS BASED ON YOUR STUDENT MATRICULATION NUMBER

For example, if your student matriculation number is A0125522Y, then it means that the character to appear on the display is:

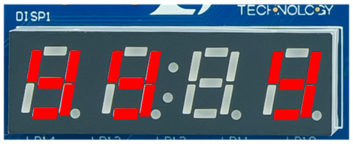


And since the rightmost numerical value of the student matriculation number is 2, AN3 will be 0, AN2 will be 0, AN1 will be 1, and AN0 will be 0. A value of 0 to a common anode makes the specific display active. Hence, with a value of 0010 to the 4 common anodes, the second display from the right will be inactive (off).

Furthermore, the five rightmost numerical values of the student matriculation number are: 2, 5, 5, 2, 2. Hence, the correct password that allows the seven-segment displays to be on would be:

LD9	LD8	LD7	LD6	LD5	LD4	LD3	LD2	LD1	LD0
OFF	OFF	OFF	OFF	ON	OFF	OFF	ON	OFF	OFF
SW9	SW8	SW7	SW6	SW5	SW4	SW3	SW2	SW1	SW0
OFF	OFF	OFF	OFF	ON	OFF	OFF	ON	OFF	OFF

The following value should appear on DISP1 (4 seven-segment displays) when the password is correct:



When the password is wrong (not having the exact 10 switches as tabulated above to be ON/OFF) nothing is displayed on DISP1.

HINTS

- 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
- **Do not use if-else functions. This is not covered in lab 1**
- Consider each one of the seven segments as being one active-low LED
- Consider each one of the four displays as being an active-low “enable command”
- It is a good practice to code and test each functionality separately, before combining them all to achieve all requirements
- **You will need DISP1 for all remaining labs sessions. Ensure you understand this part well**

LUMINUS SUBMISSION INSTRUCTIONS

- Complete as much required functionalities as possible within the given deadline, and ensure that your bitstream has been successfully generated and tested on your Basys 3 development board **BEFORE** archiving your Vivado workspace for LumiNUS upload. No working bitstream is equivalent to no marks (It is best to have some working functionalities / requirements, instead of not having any bitstream at all while trying all requirements)

- It is compulsory to archive your project in a compressed form without any simulation waveforms. In the uploaded archive, the codes (.v files) are important, not the waveforms (.wdb files). **The archive size should not exceed 1 MB in size for lab 1.** 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 LumiNUS EE2026 -> Files -> Lab and Project - Materials and Submissions -> Lab 1 Submission

- Download your LumiNUS archive after uploading. **Unzip the project folder within, open the program, and check if you can run your bitstream correctly**. No project files and no working bitstream is equivalent to losing all marks

- The LumiNUS upload must be completed by **Monday 1st February 2021, 12:00 P.M. (Noon).** Do not plan to upload 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: LumiNUS EE2026 -> Files -> Lab and Project - Materials and Submissions -> Lab 1 Submission (Late Submission)

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 ***LUMINUS 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 LumiNUS upload):

It is **compulsory to rename your project archive**, just before the LumiNUS upload, as indicated 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
Aaron Chan Siang Joo	L1_Tue_PM_Aaron Chan Siang Joo_426_Archive
Aditi Chadha	L1_Tue_AM_Aditi Chadha_898_Archive
Aiden Thaw @ Aung Kham Thaw	L1_Tue_PM_Aiden Thaw Aung Kham Th_186_Archive
Alvin Lim Jun	L1_Tue_AM_Alvin Lim Jun_284_Archive
Alvin Pang Zi Xiong	L1_Mon_PM_Alvin Pang Zi Xiong_475_Archive
Ameer Khalid	L1_Tue_PM_Ameer Khalid_328_Archive
Anderson Leong Ke Sheng	L1_Tue_PM_Anderson Leong Ke Sheng_152_Archive
Andy Chua Xun Ze	L1_Tue_PM_Andy Chua Xun Ze_384_Archive
ANG JIA JUN, DARYL	L1_Tue_PM_ANG JIA JUN DARYL_485_Archive
Ang Qi Xuan	L1_Mon_PM_Ang Qi Xuan_052_Archive
Ang Yong Siang Alwin	L1_Tue_AM_Ang Yong Siang Alwin_771_Archive
Anna Zhang Runyu	L1_Tue_PM_Anna Zhang Runyu_999_Archive
Antriksh Verma	L1_Tue_AM_Antriksh Verma_149_Archive
Anvitha Rajaram	L1_Tue_AM_Anvitha Rajaram_434_Archive
Ariel Ong Xing Er	L1_Tue_PM_Ariel Ong Xing Er_560_Archive
Bellakka Krishnamurthy Prajwal	L1_Tue_AM_Bellakka Krishnamurthy Pr_862_Archive
Boo Qian Wei, Adeline	L1_Tue_AM_Boo Qian Wei Adeline_207_Archive
Braden Teo Wei Ren	L1_Tue_AM_Braden Teo Wei Ren_238_Archive
Brendan Lau Siew Zhi	L1_Tue_AM_Brendan Lau Siew Zhi_596_Archive
Bryan Elmer Mulijono	L1_Tue_AM_Bryan Elmer Mulijono_447_Archive
Bryan Wong Hong Liang	L1_Tue_AM_Bryan Wong Hong Liang_051_Archive
Bui Quang Huy	L1_Tue_PM_Bui Quang Huy_362_Archive
Chai Zong Lun	L1_Tue_AM_Chai Zong Lun_784_Archive
Chan Keng Jit	L1_Tue_AM_Chan Keng Jit_501_Archive
Chan Yew Kun	L1_Tue_PM_Chan Yew Kun_985_Archive
Chan Zhao Yong	L1_Thu_PM_Chan Zhao Yong_518_Archive
Chan Zhi Jie Ryan	L1_Tue_PM_Chan Zhi Jie Ryan_415_Archive
Chen Hsin	L1_Tue_PM_Chen Hsin_602_Archive
CHEN SILIN	L1_Tue_AM_CHEN SILIN_445_Archive
CHEN YUHAN	L1_Tue_PM_CHEN YUHAN_520_Archive
Cheng Siyuan	L1_Tue_AM_Cheng Siyuan_728_Archive
Chew Yi Jie	L1_Tue_AM_Chew Yi Jie_632_Archive
Chia Shao Xian	L1_Wed_PM_Chia Shao Xian_447_Archive
Chong Jia An	L1_Tue_PM_Chong Jia An_190_Archive
Chong Xuan Liang	L1_Thu_AM_Chong Xuan Liang_171_Archive
Christopher Nge Jing Qi	L1_Thu_AM_Christopher Nge Jing Qi_145_Archive
Christopher Tze-Wen Langton	L1_Thu_AM_Christopher Tze-Wen Langt_265_Archive
CHUA WAN NING	L1_Tue_PM_CHUA WAN NING_189_Archive
Chua Xiong Wei	L1_Thu_AM_Chua Xiong Wei_484_Archive
Chun Min Gyu	L1_Wed_PM_Chun Min Gyu_249_Archive
Chung Ying Qiao Winnie	L1_Tue_PM_Chung Ying Qiao Winnie_324_Archive
Conrad Ephraim Wee Cher Jae	L1_Thu_AM_Conrad Ephraim Wee Cher J_123_Archive
Cordell Chan Yi Hng	L1_Tue_PM_Cordell Chan Yi Hng_253_Archive

CUI MINJING	L1_Thu_AM_CUI MINJING_441_Archive
Cui Xinyu	L1_Tue_PM_Cui Xinyu_116_Archive
Damien Lim Yu Hao	L1_Thu_AM_Damien Lim Yu Hao_933_Archive
Darren Khoo Kah Weng	L1_Thu_AM_Darren Khoo Kah Weng_516_Archive
Darryl See	L1_Tue_PM_Darryl See_317_Archive
Desmond Eng Kian Wee	L1_Wed_PM_Desmond Eng Kian Wee_095_Archive
Donovan Sim Jing Yi	L1_Tue_PM_Donovan Sim Jing Yi_402_Archive
Du Yantang	L1_Tue_PM_Du Yantang_744_Archive
Edly Irsyad B Elham	L1_Thu_AM_Edly Irsyad B Elham_555_Archive
Elumalai Oviya Dharshini	L1_Thu_AM_Elumalai Oviya Dharshini_353_Archive
Eric Bryan	L1_Thu_AM_Eric Bryan_789_Archive
Eugene Chong Zhi Liang	L1_Thu_AM_Eugene Chong Zhi Liang_525_Archive
Fan Shixi	L1_Thu_AM_Fan Shixi_848_Archive
FOONG XIN YU	L1_Thu_AM_FOONG XIN YU_018_Archive
Fu Zhehui	L1_Tue_PM_Fu Zhehui_218_Archive
Gavien Pat Wei Zhuo	L1_Thu_AM_Gavien Pat Wei Zhuo_185_Archive
GOH KAI YAO BRYAN	L1_Tue_AM_GOH KAI YAO BRYAN_690_Archive
Goh Shao Quan	L1_Tue_PM_Goh Shao Quan_422_Archive
GOH SHAU HUI GEORGE	L1_Tue_PM_GOH SHAU HUI GEORGE_353_Archive
Goh Wei Yang	L1_Tue_PM_Goh Wei Yang_254_Archive
Gu Jianqiang	L1_Tue_PM_Gu Jianqiang_514_Archive
Guan Dinghe	L1_Tue_PM_Guan Dinghe_736_Archive
Han Si Yuan	L1_Wed_PM_Han Si Yuan_050_Archive
Hemanth Bangalore Srinivas Murthy	L1_Tue_PM_Hemanth Bangalore Sriniva_135_Archive
Ho Shu Jun	L1_Tue_PM_Ho Shu Jun_198_Archive
Ho Zhen Hong	L1_Thu_AM_Ho Zhen Hong_014_Archive
Hoang Trong Tan	L1_Thu_AM_Hoang Trong Tan_425_Archive
HOE JUN LEONG	L1_Thu_AM_HOE JUN LEONG_182_Archive
Hong Xingwen	L1_Tue_PM_Hong Xingwen_861_Archive
Hossan Goh Xuan Rong	L1_Thu_AM_Hossan Goh Xuan Rong_833_Archive
How Teck Wei	L1_Tue_PM_How Teck Wei_391_Archive
HU JIALUN	L1_Thu_AM_HU JIALUN_521_Archive
Hu Xuefei	L1_Thu_AM_Hu Xuefei_885_Archive
Huang Che Yen	L1_Thu_AM_Huang Che Yen_597_Archive
HUANG HAOFENG	L1_Mon_PM_HUANG HAOFENG_936_Archive
Huang Shanshan	L1_Tue_PM_Huang Shanshan_622_Archive
HUANG YUJING	L1_Tue_PM_HUANG YUJING_485_Archive
Ian Wang Ee En	L1_Thu_AM_Ian Wang Ee En_227_Archive
Imperial Edward Justin Javier	L1_Tue_PM_Imperial Edward Justin Ja_356_Archive
Ishaan Maunil Vyas	L1_Thu_AM_Ishaan Maunil Vyas_479_Archive
Izdiyad Farhan B Zuri	L1_Thu_AM_Izdiyad Farhan B Zuri_407_Archive
Jared Cheang	L1_Mon_PM_Jared Cheang_192_Archive
Jareth Tan Eu Quan	L1_Tue_PM_Jareth Tan Eu Quan_240_Archive
Jason Ong Meng Lee	L1_Tue_PM_Jason Ong Meng Lee_174_Archive
Jasshan Kumeresh	L1_Tue_PM_Jasshan Kumeresh_503_Archive
Jeremiah Jiang	L1_Tue_PM_Jeremiah Jiang_139_Archive
Jeremiah Ong Ray	L1_Tue_PM_Jeremiah Ong Ray_551_Archive
Jeremy Goh Liang Yi	L1_Tue_PM_Jeremy Goh Liang Yi_393_Archive
JIANG QIXIONG	L1_Mon_PM_JIANG QIXIONG_698_Archive
Jiang Xing Kai	L1_Thu_AM_Jiang Xing Kai_564_Archive
Jin Minyue	L1_Tue_AM_Jin Minyue_069_Archive

JIN YIXUAN	L1_Wed_PM_JIN YIXUAN_976_Archive
Joanne Wong Wei Yin	L1_Thu_AM_Joanne Wong Wei Yin_527_Archive
Joel Matthew Chiam Zhi Qiang	L1_Thu_AM_Joel Matthew Chiam Zhi Qi_250_Archive
Jon Lim Yong Kiat	L1_Thu_AM_Jon Lim Yong Kiat_782_Archive
Jonathan Mui Koy Kit	L1_Thu_AM_Jonathan Mui Koy Kit_534_Archive
Joshua Harsha Dass	L1_Thu_AM_Joshua Harsha Dass_714_Archive
Justin Fidelis Wong Jun Wen	L1_Thu_AM_Justin Fidelis Wong Jun W_326_Archive
Kairos Koh Jia Jun	L1_Thu_AM_Kairos Koh Jia Jun_149_Archive
Keh Wen Yang, Rachel	L1_Thu_AM_Keh Wen Yang Rachel_249_Archive
Kevinaldi Dwiastajulio Hunto	L1_Thu_PM_Kevinaldi Dwiastajulio Hu_912_Archive
Khoo Jia Le Isaac	L1_Thu_AM_Khoo Jia Le Isaac_733_Archive
Khor Sheng Hou	L1_Thu_PM_Khor Sheng Hou_443_Archive
Kishor Kumar Haribaskar	L1_Thu_AM_Kishor Kumar Haribaskar_481_Archive
Koh Meng Kiat, Kenneth	L1_Tue_AM_Koh Meng Kiat Kenneth_512_Archive
Koh Qianqi	L1_Thu_AM_Koh Qianqi_746_Archive
Koh Qin Ruo	L1_Thu_PM_Koh Qin Ruo_055_Archive
Koh Ruizhe Jerome	L1_Thu_PM_Koh Ruizhe Jerome_429_Archive
Kom Xing Yuan	L1_Thu_AM_Kom Xing Yuan_993_Archive
Kong Dak Nam	L1_Thu_AM_Kong Dak Nam_171_Archive
Koo Wei De	L1_Tue_PM_Koo Wei De_133_Archive
Krishna R R	L1_Thu_PM_Krishna R R_900_Archive
Kum Wing Ho	L1_Thu_AM_Kum Wing Ho_725_Archive
Kumaravel Vignesh	L1_Thu_AM_Kumaravel Vignesh_585_Archive
Kwok Xiu Sheng Theodore	L1_Thu_AM_Kwok Xiu Sheng Theodore_118_Archive
Kwong Zhi Qian	L1_Thu_AM_Kwong Zhi Qian_569_Archive
Lam Junyu William	L1_Thu_AM_Lam Junyu William_903_Archive
LAM KAI WEN JONATHAN	L1_Thu_AM_LAM KAI WEN JONATHAN_213_Archive
Lau Miang Puang, Glennard	L1_Thu_PM_Lau Miang Puang Glennard_207_Archive
Lee An Sheng	L1_Tue_AM_Lee An Sheng_263_Archive
Lee Cheok Feng	L1_Tue_AM_Lee Cheok Feng_743_Archive
Lee Hung Tien	L1_Thu_PM_Lee Hung Tien_965_Archive
Lee Jia Jun	L1_Thu_PM_Lee Jia Jun_423_Archive
Lee Jing Rui, Evan	L1_Thu_PM_Lee Jing Rui Evan_399_Archive
Lee Jun Wen	L1_Wed_PM_Lee Jun Wen_466_Archive
LEE KENG YONG JOSHUA	L1_Tue_AM_LEE KENG YONG JOSHUA_644_Archive
Lee Qi An	L1_Tue_AM_Lee Qi An_644_Archive
Lee Shi-An, Matthew	L1_Thu_PM_Lee Shi-An Matthew_324_Archive
Lee Sungmin	L1_Tue_AM_Lee Sungmin_490_Archive
Lee Sze Ern, Jeremy	L1_Tue_PM_Lee Sze Ern Jeremy_510_Archive
Lee Yi Kai	L1_Tue_AM_Lee Yi Kai_550_Archive
Lee Yu-Hsueh	L1_Thu_PM_Lee Yu-Hsueh_639_Archive
LEONARD CHUA ZHONG QI	L1_Tue_PM_LEONARD CHUA ZHONG QI_873_Archive
Leong Kah Choong	L1_Thu_PM_Leong Kah Choong_455_Archive
Leow Yuan Yang	L1_Tue_AM_Leow Yuan Yang_574_Archive
Leroy Ong Nai Kiat	L1_Thu_PM_Leroy Ong Nai Kiat_506_Archive
Li Huanda	L1_Thu_PM_Li Huanda_101_Archive
Li Xi Yuan	L1_Tue_AM_Li Xi Yuan_926_Archive
Liang Yuzhao	L1_Tue_AM_Liang Yuzhao_802_Archive
LIM BING SEN	L1_Wed_PM_LIM BING SEN_580_Archive
LIM CHANG QUAN THADDEUS	L1_Thu_AM_LIM CHANG QUAN THADDEUS_210_Archive
Lim Jia Sheng Jackson	L1_Thu_PM_Lim Jia Sheng Jackson_596_Archive

Lim Kay Yun	L1_Tue_AM_Lim Kay Yun_244_Archive
Lim Shyun Yin	L1_Tue_AM_Lim Shyun Yin_063_Archive
Lim Wen Jie	L1_Thu_PM_Lim Wen Jie_382_Archive
Liu Danfeng	L1_Thu_PM_Liu Danfeng_777_Archive
Liu Ruijun	L1_Thu_PM_Liu Ruijun_371_Archive
LIU ZEHANG	L1_Tue_AM_LIU ZEHANG_491_Archive
LIU ZHIYANG	L1_Wed_PM_LIU ZHIYANG_653_Archive
Loo Keng Leong	L1_Thu_PM_Loo Keng Leong_408_Archive
LOW ZHEN WEI JERRELL	L1_Tue_AM_LOW ZHEN WEI JERRELL_516_Archive
Lu Jingguang	L1_Thu_PM_Lu Jingguang_319_Archive
Lu Sicheng	L1_Tue_AM_Lu Sicheng_634_Archive
LU ZONGHAN	L1_Tue_PM_LU ZONGHAN_452_Archive
MA XUDONG	L1_Tue_PM_MA XUDONG_493_Archive
Ma Zijian	L1_Tue_AM_Ma Zijian_828_Archive
Madhan Selvapandian	L1_Tue_AM_Madhan Selvapandian_482_Archive
Madheswaran Niveytha	L1_Thu_PM_Madheswaran Niveytha_465_Archive
Mah Yuan Jie Alvin	L1_Tue_AM_Mah Yuan Jie Alvin_500_Archive
Mahadevan Svetha	L1_Tue_AM_Mahadevan Svetha_108_Archive
Mahadevan Swati	L1_Tue_AM_Mahadevan Swati_107_Archive
Marcus Goh Xuan De	L1_Thu_PM_Marcus Goh Xuan De_355_Archive
Marcus Lim Sheng Jie	L1_Thu_PM_Marcus Lim Sheng Jie_408_Archive
Marcus Ong Yih	L1_Tue_AM_Marcus Ong Yih_404_Archive
Mathur Aayush	L1_Tue_AM_Mathur Aayush_581_Archive
Mayank Panjiyara	L1_Tue_AM_Mayank Panjiyara_763_Archive
Mehedi Hasan Salim	L1_Thu_PM_Mehedi Hasan Salim_436_Archive
Mohamed Faez Bin Shahlan	L1_Thu_PM_Mohamed Faez Bin Shahlan_297_Archive
Mohammad Shoib Memon Loya	L1_Tue_AM_Mohammad Shoib Memon Loya_487_Archive
Muhammad Aizat Bin Rahim	L1_Thu_PM_Muhammad Aizat Bin Rahim_437_Archive
Muhammad Ashraf B Mohamad J	L1_Tue_AM_Muhammad Ashraf B Mohamad_432_Archive
MUHAMMAD HAZIM BIN ABDULLAH	L1_Thu_PM_MUHAMMAD HAZIM BIN ABDULL_633_Archive
Muhammad Jaish Bin Jamalun Nasir	L1_Wed_PM_Muhammad Jaish Bin Jamalu_287_Archive
Mun Le Zong	L1_Tue_AM_Mun Le Zong_172_Archive
Nan Song	L1_Thu_PM_Nan Song_102_Archive
Ng Andre	L1_Tue_AM_Ng Andre_973_Archive
Ng Cheng Yang, Titus	L1_Tue_AM_Ng Cheng Yang Titus_478_Archive
Ng Jin Loong, Jeremy	L1_Thu_PM_Ng Jin Loong Jeremy_395_Archive
Ng Qi Hao	L1_Thu_PM_Ng Qi Hao_420_Archive
Ng Xinyi	L1_Thu_PM_Ng Xinyi_545_Archive
Ngoi Hui Wen, Vanessa	L1_Tue_AM_Ngoi Hui Wen Vanessa_471_Archive
Nguyen Minh Tuan	L1_Tue_AM_Nguyen Minh Tuan_389_Archive
Nguyen Van Binh	L1_Mon_PM_Nguyen Van Binh_453_Archive
Nigel Loh Weien	L1_Thu_PM_Nigel Loh Weien_416_Archive
Nigel Ng	L1_Mon_PM_Nigel Ng_444_Archive
Nishant Rai	L1_Mon_PM_Nishant Rai_182_Archive
Oh Qi Ren	L1_Thu_PM_Oh Qi Ren_441_Archive
Ong Chor Yew	L1_Tue_PM_Ong Chor Yew_460_Archive
Ong Jun Giat	L1_Thu_PM_Ong Jun Giat_205_Archive
Ong Siying Falicia	L1_Thu_AM_Ong Siying Falicia_566_Archive
Ong Yew Yong, Adrian	L1_Thu_PM_Ong Yew Yong Adrian_401_Archive
Owng Kai Leng Sally	L1_Thu_PM_Owng Kai Leng Sally_331_Archive
Pang Kai Lin	L1_Thu_PM_Pang Kai Lin_036_Archive

Pang Kai Yi	L1_Thu_PM_Pang Kai Yi_236_Archive
Pang Qi Wei, Jenna	L1_Wed_PM_Pang Qi Wei Jenna_685_Archive
Paramita Tejasvi	L1_Mon_PM_Paramita Tejasvi_194_Archive
Peh Li Yan	L1_Thu_PM_Peh Li Yan_267_Archive
PENG FEI	L1_Mon_PM_PENG FEI_518_Archive
Peng Yanjia	L1_Tue_AM_Peng Yanjia_877_Archive
Phoon Pei Zhen	L1_Thu_PM_Phoon Pei Zhen_744_Archive
Phua Keng Wee	L1_Thu_PM_Phua Keng Wee_706_Archive
Phuah Yong Chen Keith	L1_Thu_PM_Phuah Yong Chen Keith_360_Archive
Pichanon Rattanadilok Na Phuket	L1_Tue_AM_Pichanon Rattanadilok Na _545_Archive
Pojcharapol Leenukiat	L1_Wed_PM_Pojcharapol Leenukiat_642_Archive
Poon Jeun Lek	L1_Wed_PM_Poon Jeun Lek_202_Archive
Pow Zhi Xiang	L1_Tue_AM_Pow Zhi Xiang_942_Archive
Pradhan Rachit Manish	L1_Wed_PM_Pradhan Rachit Manish_230_Archive
Pranav Venkatram	L1_Tue_AM_Pranav Venkatram_200_Archive
Qi Tian Cong	L1_Wed_PM_Qi Tian Cong_442_Archive
Ramalingam Saravanamani	L1_Tue_AM_Ramalingam Saravanamani_586_Archive
Ravindiran Rakesh	L1_Tue_AM_Ravindiran Rakesh_010_Archive
Rebecca Chua	L1_Thu_AM_Rebecca Chua_171_Archive
REN TIANLE	L1_Mon_PM_REN TIANLE_446_Archive
Renzo Rivera Canare	L1_Thu_AM_Renzo Rivera Canare_502_Archive
Reuel Teo Lu Wei	L1_Wed_PM_Reuel Teo Lu Wei_435_Archive
Richard Willie	L1_Thu_AM_Richard Willie_368_Archive
Roycius Lim Yuanwei	L1_Thu_AM_Roycius Lim Yuanwei_060_Archive
Samuel Ong Wei Chuan	L1_Tue_PM_Samuel Ong Wei Chuan_462_Archive
Se Sean	L1_Wed_PM_Se Sean_140_Archive
See Jian Hui	L1_Thu_AM_See Jian Hui_737_Archive
Seet Ting Yang Irvin	L1_Mon_PM_Seet Ting Yang Irvin_608_Archive
Seetoh Yit Ching	L1_Mon_PM_Seetoh Yit Ching_154_Archive
Seo Gimin	L1_Mon_PM_Seo Gimin_442_Archive
Seth Teng Shann	L1_Wed_PM_Seth Teng Shann_419_Archive
Shao Yurui	L1_Mon_PM_Shao Yurui_111_Archive
Shawn Chang	L1_Wed_PM_Shawn Chang_151_Archive
Shreshth Sarda	L1_Wed_PM_Shreshth Sarda_424_Archive
Shyam Ganesh Jayagopi	L1_Mon_PM_Shyam Ganesh Jayagopi_484_Archive
Sidharth Premnath	L1_Wed_PM_Sidharth Premnath_502_Archive
Siew Yang Zhi	L1_Thu_AM_Siew Yang Zhi_331_Archive
Sim Le Yee Beatrice	L1_Wed_PM_Sim Le Yee Beatrice_769_Archive
Sin Ren Xiang	L1_Thu_PM_Sin Ren Xiang_865_Archive
Sivakumar Yogarajan	L1_Wed_PM_Sivakumar Yogarajan_505_Archive
Song Chenan	L1_Wed_PM_Song Chenan_797_Archive
Song Min Kyu	L1_Wed_PM_Song Min Kyu_226_Archive
Sridharan Arvind Srinivasan	L1_Mon_PM_Sridharan Arvind Srinivas_477_Archive
Sthitipragyan Samal	L1_Mon_PM_Sthitipragyan Samal_664_Archive
Sun Jiale	L1_Mon_PM_Sun Jiale_853_Archive
SUN JIAWEI	L1_Thu_PM_SUN JIAWEI_496_Archive
SWAMINATHAN VARUN	L1_Mon_PM_SWAMINATHAN VARUN_281_Archive
Swann Tet Aung	L1_Mon_PM_Swann Tet Aung_552_Archive
Tan Haoxuan	L1_Wed_PM_Tan Haoxuan_934_Archive
Tan Hui En	L1_Mon_PM_Tan Hui En_373_Archive
Tan Jun Heng Daren Justin	L1_Mon_PM_Tan Jun Heng Daren Justin_331_Archive

Tan Kah Heng	L1_Mon_PM_Tan Kah Heng_677_Archive
Tan Le Yi	L1_Mon_PM_Tan Le Yi_071_Archive
Tan Lindsey	L1_Wed_PM_Tan Lindsey_197_Archive
Tan Qi Xian, Keith	L1_Wed_PM_Tan Qi Xian Keith_397_Archive
Tan Rui Yang	L1_Mon_PM_Tan Rui Yang_472_Archive
Tan Tze Yeong	L1_Wed_PM_Tan Tze Yeong_970_Archive
Tan Wei Li	L1_Wed_PM_Tan Wei Li_336_Archive
Tan Xing Jie	L1_Mon_PM_Tan Xing Jie_747_Archive
Tan Yong Zheng	L1_Wed_PM_Tan Yong Zheng_261_Archive
Tang Zehou	L1_Mon_PM_Tang Zehou_210_Archive
Tay Weida	L1_Mon_PM_Tay Weida_027_Archive
Tay Yi Heng, Atticus	L1_Mon_PM_Tay Yi Heng Atticus_994_Archive
Teh Jiewen	L1_Mon_PM_Teh Jiewen_520_Archive
Teh Zi-Chun	L1_Wed_PM_Teh Zi-Chun_328_Archive
Teng Yi Shiong	L1_Mon_PM_Teng Yi Shiong_647_Archive
Teo Ziyi Ivy	L1_Mon_PM_Teo Ziyi Ivy_117_Archive
Tham Yang Tze Xavier	L1_Wed_PM_Tham Yang Tze Xavier_256_Archive
TIAN ZHENYU	L1_Thu_PM_TIAN ZHENYU_467_Archive
Tiang Zhang Quan Xavier	L1_Wed_PM_Tiang Zhang Quan Xavier_446_Archive
Tie Zhou Peng	L1_Wed_PM_Tie Zhou Peng_264_Archive
Toh Yi Cheng	L1_Wed_PM_Toh Yi Cheng_421_Archive
Toh Yi Zhi	L1_Mon_PM_Toh Yi Zhi_086_Archive
Tran Nhan Duc Anh	L1_Mon_PM_Tran Nhan Duc Anh_358_Archive
Tran Thi Phuong Thao	L1_Wed_PM_Tran Thi Phuong Thao_438_Archive
Varun Agarwal	L1_Wed_PM_Varun Agarwal_605_Archive
VIKAS HARLANI	L1_Mon_PM_VIKAS HARLANI_376_Archive
Vishal Jeyaram	L1_Mon_PM_Vishal Jeyaram_224_Archive
Wan Haocheng	L1_Wed_PM_Wan Haocheng_780_Archive
Wang Wenxuan	L1_Wed_PM_Wang Wenxuan_649_Archive
WANG YUDA	L1_Thu_PM_WANG YUDA_443_Archive
Wang Zhao Yu, Edward	L1_Mon_PM_Wang Zhao Yu Edward_953_Archive
Wang Zhihuang	L1_Mon_PM_Wang Zhihuang_682_Archive
Wang Zichen	L1_Wed_PM_Wang Zichen_951_Archive
Wang Zihan	L1_Wed_PM_Wang Zihan_361_Archive
Wang Zixi	L1_Wed_PM_Wang Zixi_445_Archive
William Wahyudi	L1_Mon_PM_William Wahyudi_230_Archive
Wong Jun Lin	L1_Tue_AM_Wong Jun Lin_077_Archive
Wong Tze Shan Samantha	L1_Mon_PM_Wong Tze Shan Samantha_672_Archive
Wong Zi Xin, Avellin	L1_Mon_PM_Wong Zi Xin Avellin_073_Archive
Woo Bo Tuan	L1_Mon_PM_Woo Bo Tuan_153_Archive
WU HAO HSUAN	L1_Thu_PM_WU HAO HSUAN_635_Archive
Wu Luoyu	L1_Mon_PM_Wu Luoyu_894_Archive
WU YUWEI	L1_Wed_PM_WU YUWEI_472_Archive
XIAO JUNTIAN	L1_Wed_PM_XIAO JUNTIAN_497_Archive
Xu Yuxing	L1_Wed_PM_Xu Yuxing_183_Archive
Xue Yuxuan	L1_Wed_PM_Xue Yuxuan_250_Archive
Yam Jin Ee Dmitri	L1_Mon_PM_Yam Jin Ee Dmitri_974_Archive
Yang Zikun	L1_Tue_AM_Yang Zikun_313_Archive
Yap Joon Siong	L1_Mon_PM_Yap Joon Siong_925_Archive
YAP WEI XUAN	L1_Mon_PM_YAP WEI XUAN_997_Archive
Yap Zhan Wei	L1_Thu_PM_Yap Zhan Wei_455_Archive

Yeat Nai Jie	L1_Mon_PM_Yeat Nai Jie_613_Archive
Yeo Shi Heng	L1_Wed_PM_Yeo Shi Heng_390_Archive
Yeo Wei Hng	L1_Mon_PM_Yeo Wei Hng_075_Archive
Yeo Zi Hao, Edwin	L1_Mon_PM_Yeo Zi Hao Edwin_710_Archive
YIP WAYNE	L1_Mon_PM_YIP WAYNE_998_Archive
YU HAIHONG	L1_Wed_PM_YU HAIHONG_470_Archive
Yue Junfeng	L1_Thu_AM_Yue Junfeng_802_Archive
Yuk Yeon Soo	L1_Wed_PM_Yuk Yeon Soo_243_Archive
Zeng Jiexiong	L1_Mon_PM_Zeng Jiexiong_052_Archive
ZHANG HAOYU	L1_Thu_AM_ZHANG HAOYU_783_Archive
ZHAO LUOYUANG	L1_Mon_PM_ZHAO LUOYUANG_466_Archive
Zhao Yibo	L1_Wed_PM_Zhao Yibo_863_Archive
Zhao Ziqi	L1_Wed_PM_Zhao Ziqi_275_Archive
ZHONG XINGHAN	L1_Mon_PM_ZHONG XINGHAN_468_Archive
ZHOU CHENGXU	L1_Mon_PM_ZHOU CHENGXU_492_Archive
ZHOU YUHAN	L1_Wed_PM_ZHOU YUHAN_530_Archive
Zhu Shaohan Steven	L1_Wed_PM_Zhu Shaohan Steven_193_Archive
Zhuang Jianning	L1_Mon_PM_Zhuang Jianning_277_Archive
Zubin Jain	L1_Tue_PM_Zubin Jain_990_Archive