EE2026: DIGITAL DESIGN

Academic Year 2020-2021, Semester 2

LAB 3: Sequential Circuits in Verilog - Part 1

OVERVIEW

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

The pre-requisites for this lab are:

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

This lab will cover the following:

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

Tasks for this lab include:

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

GRADED ASSIGNMENT [LUMINUS SUBMISSION: FRIDAY 26th FEBRUARY 2021, NOON]:

- Display different LED patterns at different time instants and display periodically changing characters on the 7-segment displays of the Basys 3 development board

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

THE BLINKING LED

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

The **CLOCK** signal is an external input signal that resembles a square wave of 50% duty cycle. If this **CLOCK** signal is connected directly to a physical LED, the latter will light up when the signal is HIGH, and will switch off when the signal is LOW, as illustrated in **Figure 3.2**.

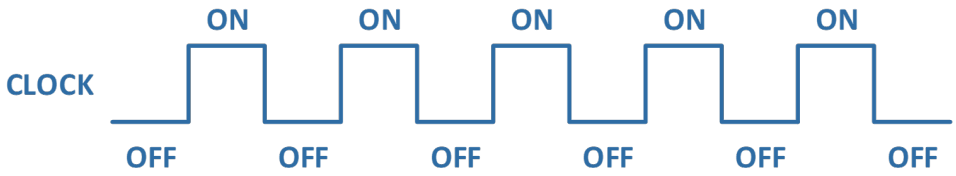


Figure 3.2: *A **CLOCK** signal with 50% duty cycle*

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

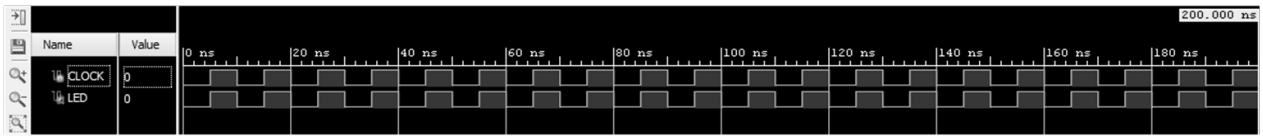
Verilog code for blinky, using the dataflow method

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

    assign LED = CLOCK;

endmodule
```

Simulation source code to test the blinky design

```
`timescale 1ns / 1ps

module test_blinky( );

    reg CLOCK; wire LED;

    blinky dut (CLOCK, LED);

    initial begin
        CLOCK = 0;
    end

   always begin
       #5 CLOCK = ~CLOCK;
   end

endmodule
```

Expected simulation waveform for the blinky design



UNDERSTANDING | TASK 2

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

1. What is the unit of time being used in the simulation source?

   _____

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

   _____

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

   _____

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

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

1. Uncomment lines 7 to 9 to create a clock signal of 100 MHz with 50% duty cycle. If required, rename the signal to the name used in your **blinky** code. In our example, the name **CLOCK** was used, and the final changes may look similar to **Figure 3.3**.

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

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

Figure 3.1: *Modifying your constraint file, based on the contents of the Basys3_Master.xdc*

UNDERSTANDING | TASK 3

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

_____

_____

THE NOTICEABLE BLINKING LED

To be able to observe a blinking LED at a frequency that is visible to the human eyes, modifications need to be done to the Verilog code. Let us introduce a temporary variable **COUNT** that is incremented by 1 at every rising edge (transition from low to high, and also called a positive edge) of the **CLOCK** signal, as shown in **Figure 3.4**. By making use of **COUNT**, a lower frequency signal can be obtained, while the Verilog code for **COUNT** can be created by using the behavioural method of modelling.

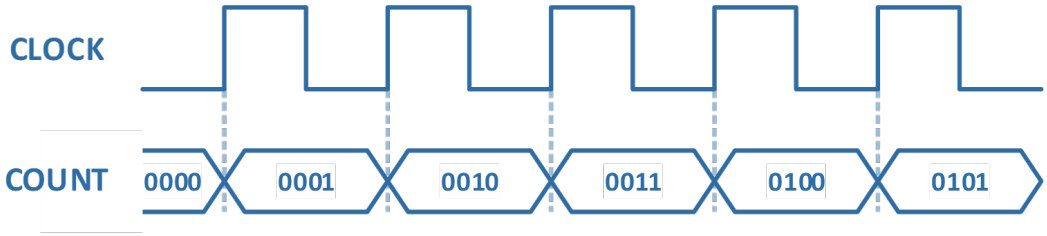


*Figure 3.4: Increasing **COUNT** at each rising edge of **CLOCK***

Verilog code for a slower blinky, using behavioural modelling

```
module slow_blinky_module (input CLOCK);

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

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

endmodule
```

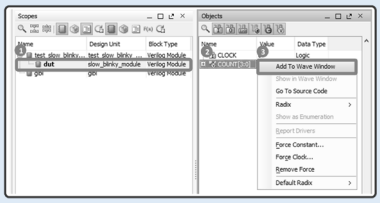
UNDERSTANDING | TASK 4

Create a simulation source for the **slow_blinky_module** design, and observe the waveform of signal **COUNT.**

[NOTE] Analysing a variable in the simulation waveform window

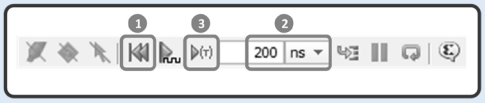
By default, the simulation window only shows the waveforms of input and output signals. To see the waveforms of variables during the simulation, such as the variable **COUNT**:

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



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

1. Restart the simulation
2. Set the simulation time and units
3. Run the simulation for the amount of time set in step 2



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

State the frequency of **COUNT[3]**, **COUNT[2]**, **COUNT[1]** and **COUNT[0]**: _____ , _____ , _____ , _____

UNDERSTANDING | TASK 5 * ***Warning:*** *Do not simulate the code. Why?* *

Using the knowledge and coding obtained from **UNDERSTANDING | TASK 3** and **UNDERSTANDING | TASK 4**, calculate the minimum size of **COUNT** and the bit number which will allow a clock output of around 0.75 Hz.

Minimum of bits required for **COUNT**: _____ Bit number of **COUNT**: _____

Name that waveform with the frequency of 0.75 Hz as **SLOWCLOCK_A**. Modify the **slow_blinky_module** design, and implement the LED blinking at a frequency of around 0.75 Hz on the Basys 3 development board.

UNDERSTANDING | TASK 6

In the **slow_blinky_module**, insert the following line of code in the always block:

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

Do additional modifications in the **slow_blinky_module** and the testbench code, and simulate the **slow_blinky_module** design.

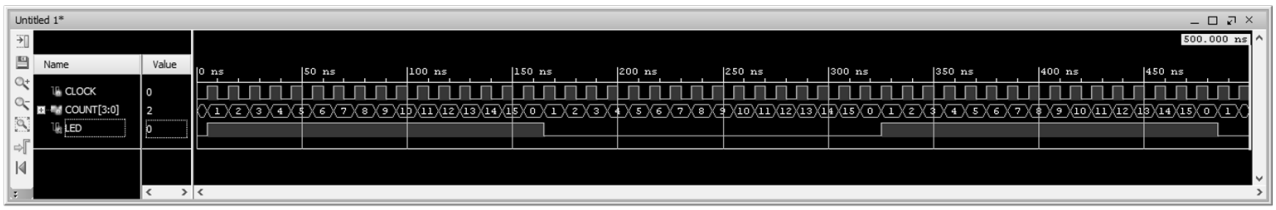
[NOTE] Hints on the modifications related to slow_blinky_module

▶ `LED` is an output signal of the design module

▶ A signal declared as **`reg`** can be reused within the design module. An example is an output signal that needs to be reused as input within the design module

▶ A design module signal declared as **`reg`** can be given an initial value, especially if toggling is involved

If the codes have been correctly written, a simulation waveform similar to what is shown below can be obtained:



Observe the waveform that you have obtained in your simulation window, and calculate the frequency of the signal **LED**:

f = _____ MHz

From your understanding of the multiplexer, state what the following line of code means:

```
LED <= ( COUNT == 4'b0000 ) ? ~LED : LED ;
```

_____

_____

UNDERSTANDING | TASK 7 * ***Warning:*** *Do not simulate the code. Why? **

Using the knowledge and coding obtained from **UNDERSTANDING | TASK 6**, calculate the number of bits for **COUNT** which will allow **LED** to have a frequency of around 0.75 Hz.

Number of bits required for **COUNT**: _____

Name that waveform with the frequency of 0.75 Hz as **SLOWCLOCK_B**. Modify the **slow_blinky_module** design, and implement the LED blinking at a frequency of around 0.75 Hz on the Basys 3 development board.

Finally, modify and implement the Verilog code such that there are two blinking speeds for the LED based on the state of a switch:

- If the switch is in the OFF position, the LED should blink at a frequency of around 0.75 Hz
- If the switch is in the ON position, the LED should blink at a frequency of around 1.50 Hz

One way to accomplish this task is to use multiplexers in Verilog. It may also be implemented using other methods that involve conditional statements.

GRADED POST-LAB ASSIGNMENT

SUBTASK A [Status]

This lab 3 graded assignment is divided into two subtasks for ease of understanding. In subtask A, we assume the Basys 3 development board is being used to monitor the water level status in a water reservoir. In your program, all characters appearing on the seven-segment displays must follow ***Table A***, as shown below:

Character	Aa	Bb	Cc	Dd	Ee	Ff	Gg	Hh	Ii	Jj	Kk	Ll	Mm
Seven-segment display													

Character	Nn	Oo	Pp	Qq	Rr	Ss	Tt	Uu	Vv	Ww	Xx	Yy	Zz
Seven-segment display													

Character	0	1	2	3	4	5	6	7	8	9	-	.	@
Seven-segment display													

Table A

Throughout your program, the following seven-segment displays must be activated depending on the **1st (First)** rightmost numerical value of your student matriculation number, as indicated in ***Table B***:

Seven-segment displays that are required to be always ON or OFF				
1st (First) rightmost numerical value of your student matriculation number	AN3 Display	AN2 Display	AN1 Display	AN0 Display
0	ON	ON	ON	ON
1	ON	ON	ON	OFF
2	ON	ON	OFF	ON
3	ON	ON	OFF	OFF
4	ON	OFF	ON	ON
5	ON	OFF	ON	OFF
6	ON	OFF	OFF	ON
7	ON	OFF	OFF	OFF
8	OFF	ON	ON	ON
9	OFF	ON	ON	OFF

Table B

Before the start of your program, switches SW0 to SW15 must be in the OFF position. When the program starts, LD0 to LD15 are OFF, and the seven segment displays do not show any characters (All 7 segments and decimal point segment are off).

Based on the **3rd rightmost value** of your student matriculation number, your 'Entry Valve Level (**EVL**)' is as indicated in ***Table C***:

3rd (Third) rightmost numerical value of your student matriculation number	Entry Valve Level (**EVL**)	Corresponding **EVL-LED**	Water Release Interval at **EVL-LED** (in seconds)
0 or 1	15	LD15	1.33 seconds
2 or 3	14	LD14	1.33 seconds
4 or 5	13	LD13	1.33 seconds
6 or 7	12	LD12	2.67 seconds
8 or 9	11	LD11	2.67 seconds

Table C

LD0 represents the bottom of the water reservoir, while the top of your water reservoir is represented by **EVL-LED**.

At regular intervals indicated in ***Table C***, water is released at the **EVL**. That water drop is represented by one LED being ON, and the water is always clearly observed to start from the **EVL-LED** right from the start of the program. It falls towards the bottom of the reservoir and stops one LED above the current reservoir water level. The water drops always fall at a speed of 3 Hz (In other words, the LED which is ON moves to the next adjacent lower value LED every 0.33 seconds)

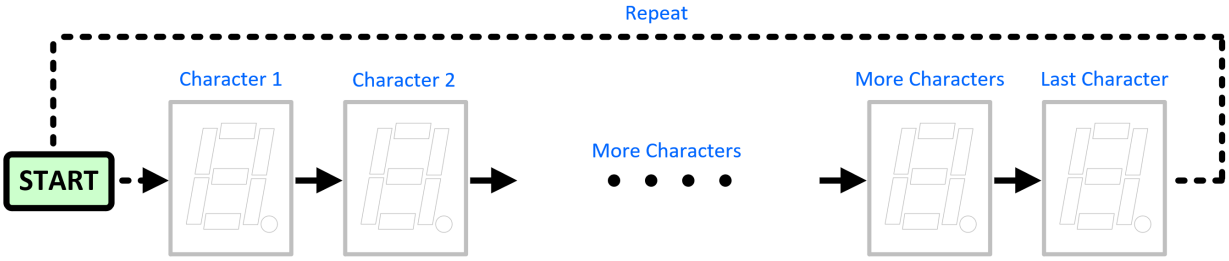
As soon as all LEDs from LD0 to **EVL-LED** are ON, the water reservoir is considered full, water stops flowing into the reservoir, and the following two events (Event 1 and Event 2) are observed at the same time:

Event 1:
EVL-LED blinks at a frequency that is dependent on the **4th rightmost value** of your student matriculation number, and as indicated in ***Table D*** below:

4th (Fourth) rightmost numerical value of your student matriculation number	Frequency, f_{BLINK} *(An error ± 0.1 Hz for the frequency is acceptable)*
0 or 2 or 4	1.49 Hz
1 or 3 or 5	0.75 Hz
6 or 7 or 8 or 9	0.37 Hz

Table D

Event 2:
Characters must be shown on your activated seven-segment displays in a specific sequence, depending the **4th rightmost value** of your student matriculation number, and as indicated in ***Figure E*** and ***Table E***:



All your activated anodes must show the SAME character at a specific clock cycle.
[DO NOT show different characters simultaneously]

Figure E

4th (Fourth) rightmost numerical value of your student matriculation number	Character 1 ▸ Character 2 ▸ Character 3 ▸ … ▸ Last Character ▸ Repeat from Character 1
0	.CAUTION-CHECK
1	.WARNING-NOTICE
2	.HIGH-WATER-LEVEL
3	.HIGH-LEVEL
4	.WATER-LEVEL-CAUTION
5	.ATTENTION
6	.RELEASE-WATER-NOW
7	.WARNING-LEVEL
8	.DANGER
9	.DANGER-WATER-LEVEL

Table E

Note that the first character is always a dot (.) and each character must stay on for 0.67 seconds (An error of ±0.05 seconds is acceptable). **Marks are not awarded if any character (coloured in red) is missing, including the dash (-) character where applicable.**

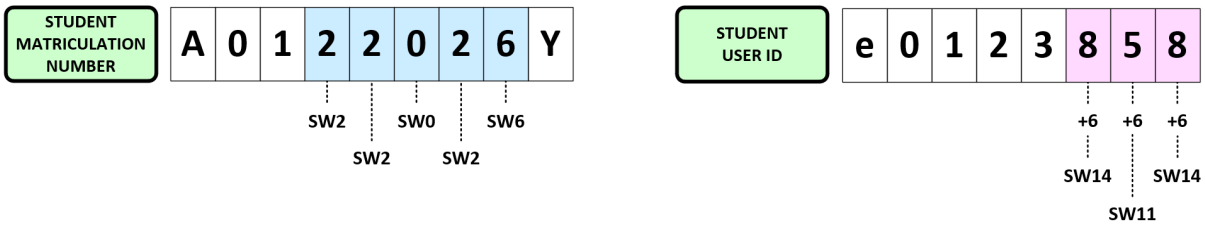
SUBTASK B [Action for Flooding Emergencies]

This Status and Action for Flooding Emergencies (SAFE) system has an additional feature to discharge water when the reservoir is full. The user may start the process to fully discharge water from the reservoir by opening several discharge valves, and this process can only happen when the water reservoir is full. **Discharging water from the reservoir when it is not full must not be allowed.**

The discharge valves are represented by switches SW0 to SW15 on the Basys 3 development board. When the user turns ON a specific switch, **it is assumed that the discharge valve corresponding to that switch will be permanently opened, even if that switch is turned OFF again**.

The correct switches that must be turned ON or were previously ON, are indicated by:

(a) The rightmost 5 individual numerical digits of your **student matriculation number AND**
(b) The rightmost 3 individual numerical digits of your **student user ID**, with each individual digit increased by **+6**



If the correct switches are ON or were previously ON, water starts discharging immediately from the reservoir, with one additional LED being turned OFF at a frequency of f_{OFF} and starting from **EVL-LED** towards LD0. The frequency f_{OFF} is dependent on the **3rd rightmost value** of your student matriculation number, as shown in ***Table F***:

3rd (Third) rightmost numerical value of your student matriculation number	Frequency, f_{OFF} *(An error ± 0.1 Hz for the frequency is acceptable)*
0	6.00 Hz
1	3.00 Hz
2	6.00 Hz
3	3.00 Hz
4	1.50 Hz
5	3.00 Hz
6	1.50 Hz
7	0.75 Hz
8	1.50 Hz
9	0.75 Hz

Table F

When all the LEDs are OFF, the segment displays must then show blank characters (Segment LEDs are OFF). The program ends here.

Assume that the user will not turn on the wrong switches. Even if the wrong switches are turned ON, it can be ignored.

Hint: **Use a stopwatch to manually verify whether you have used the correct frequencies in this SAFE system**

EXAMPLE BASED ON YOUR STUDENT MATRICULATION CARD NUMBER

Fully understand this example before working on your assignment.

If a student has matriculation number "A01282288Y" and user ID "e0123277" then:

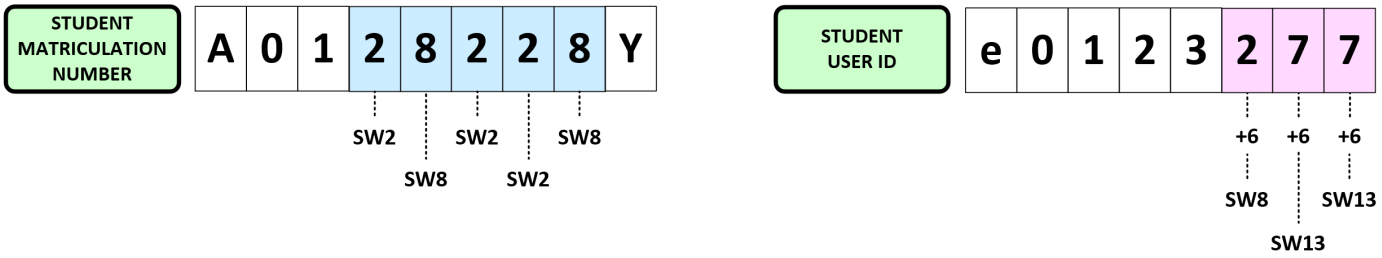
- **1st rightmost digit is 8:** The rightmost 3 seven-segment displays out of the 4 seven-segment displays will be ON wherever required during the running of the program

- **3rd rightmost digit is 2:** The EVL is 14, and the EVL-LED is LD14. A new drop of water (New LED light up at EVL-LED) appears every 1.33 seconds

  Water is discharged from the water reservoir at a frequency f_{OFF} = 3.00 Hz (One LEDs turn OFF every 0.33 seconds)

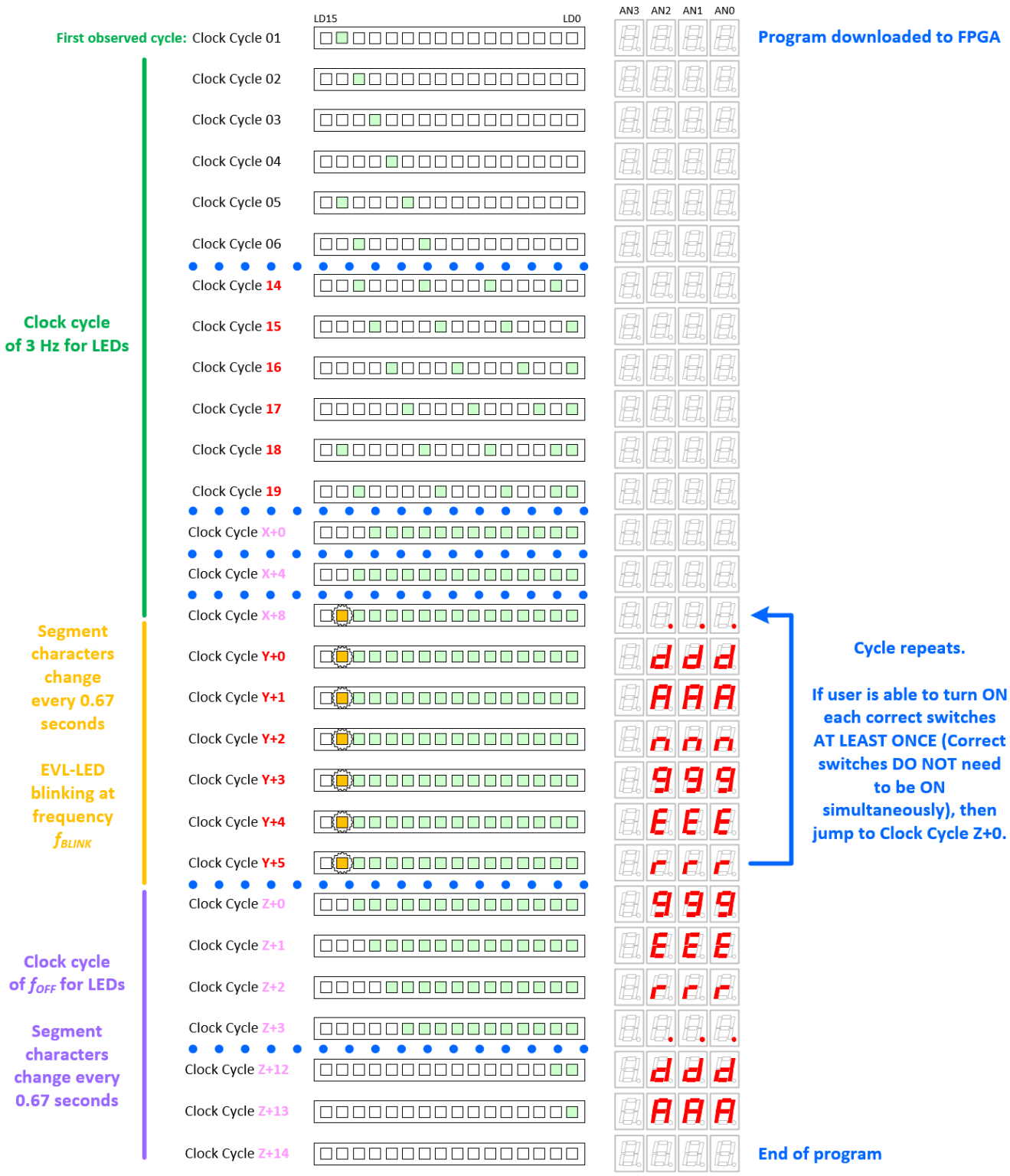
- **4th rightmost digit is 8:** The characters to appear sequentially on the activated seven segment displays are:
  . ▸ D ▸ A ▸ N ▸ G ▸ E ▸ R ▸ . ▸ D ▸ A ▸ N ▸ G ▸ E ▸ R ▸ . ▸ D …

  To discharge water when the reservoir is full, the correct switches that must be turned ON or were previously ON are:
  SW2, **SW8** , **SW13**



(This example continues in the next page)

Example where student has matriculation number "A0128228Y" and user ID "e0123277"



HINTS

- Complete and understand **UNDERSTANDING | TASK 7** clearly before working on this assignment
- Excellent mastery of structural modelling and instantiation is recommended, to properly send signals from one instance to other instances
- Creating different modules for different functions, and then simulating them before instantiating them in a main module will make complex systems easier to debug
- Use multiplexers to select between different clock signals coming from multiple clock dividers
- A counter whose value can change at each clock cycle can be considered. For example, that counter can count 0, 1, 2, 3, 0, 1, 2, 3 etc ... Following that, if-else statements, multiplexers, or case statement that indicates what to do at each specific counter value can be created

  Case statements are recommended here, and the case statement template is given below.

```
case (expression)
    case_item: statement or statement_group
    case_item: statement or statement_group
    default: statement or statement_group
endcase
```

- An example in using case is given below. Note that case statements (if-else statements also) must lie within an always block:

```
always @ (posedge clk_25_mhz)
begin
    case (counter_value)
       2'd0:
           begin
                  my_value_a <= 20;
                  my_value_b <= 40;
           end
       2'd1:
           begin
                  my_value_a <= 100;
                  my_value_b <= 200;
           end
       2'd2: my_value_c <= 5;
       default: my_value_d <= 9;
    endcase
end
```

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

- Refer to **http://tiny.cc/ee2026wiki** for more details on commonly encountered errors.

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 3 MB in size for lab 3.** 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.

- There are no report submissions for this lab assignment

- Upload to LumiNUS EE2026 -> Files -> Lab and Project - Materials and Submissions -> Lab 3 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 **Friday 26th 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 3 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 will instead be reviewed towards the end of the semester. **You will not be able to see your results during the labs in such situations**

APPENDIX (COMPULSORY renaming before just LumiNUS upload):

It is **compulsory to rename your project archive**, just before the LumiNUS 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
Aaron Chan Siang Joo	L3_Tue_PM_Aaron Chan Siang Joo_426_Archive
Aditi Chadha	L3_Tue_AM_Aditi Chadha_898_Archive
Aiden Thaw @ Aung Kham Thaw	L3_Tue_PM_Aiden Thaw Aung Kham Th_186_Archive
Alvin Lim Jun	L3_Tue_AM_Alvin Lim Jun_284_Archive
Alvin Pang Zi Xiong	L3_Mon_PM_Alvin Pang Zi Xiong_475_Archive
Ameer Khalid	L3_Tue_PM_Ameer Khalid_328_Archive
Anderson Leong Ke Sheng	L3_Tue_PM_Anderson Leong Ke Sheng_152_Archive
Andy Chua Xun Ze	L3_Tue_PM_Andy Chua Xun Ze_384_Archive
ANG JIA JUN, DARYL	L3_Tue_PM_ANG JIA JUN DARYL_485_Archive
Ang Qi Xuan	L3_Mon_PM_Ang Qi Xuan_052_Archive
Ang Yong Siang Alwin	L3_Tue_AM_Ang Yong Siang Alwin_771_Archive
Anna Zhang Runyu	L3_Tue_PM_Anna Zhang Runyu_999_Archive
Antriksh Verma	L3_Tue_AM_Antriksh Verma_149_Archive
Anvitha Rajaram	L3_Tue_AM_Anvitha Rajaram_434_Archive
Ariel Ong Xing Er	L3_Tue_PM_Ariel Ong Xing Er_560_Archive
Bellakka Krishnamurthy Prajwal	L3_Tue_AM_Bellakka Krishnamurthy Pr_862_Archive
Boo Qian Wei, Adeline	L3_Tue_AM_Boo Qian Wei Adeline_207_Archive
Braden Teo Wei Ren	L3_Tue_AM_Braden Teo Wei Ren_238_Archive
Brendan Lau Siew Zhi	L3_Tue_AM_Brendan Lau Siew Zhi_596_Archive
Bryan Elmer Mulijono	L3_Tue_AM_Bryan Elmer Mulijono_447_Archive
Bryan Wong Hong Liang	L3_Tue_AM_Bryan Wong Hong Liang_051_Archive
Bui Quang Huy	L3_Tue_PM_Bui Quang Huy_362_Archive
Chai Zong Lun	L3_Tue_AM_Chai Zong Lun_784_Archive
Chan Keng Jit	L3_Tue_AM_Chan Keng Jit_501_Archive
Chan Yew Kun	L3_Tue_PM_Chan Yew Kun_985_Archive
Chan Zhao Yong	L3_Thu_PM_Chan Zhao Yong_518_Archive
Chan Zhi Jie Ryan	L3_Tue_PM_Chan Zhi Jie Ryan_415_Archive
Chen Hsin	L3_Tue_PM_Chen Hsin_602_Archive
CHEN SILIN	L3_Tue_AM_CHEN SILIN_445_Archive
CHEN YUHAN	L3_Tue_PM_CHEN YUHAN_520_Archive
Cheng Siyuan	L3_Tue_AM_Cheng Siyuan_728_Archive
Chew Yi Jie	L3_Tue_AM_Chew Yi Jie_632_Archive
Chia Shao Xian	L3_Wed_PM_Chia Shao Xian_447_Archive
Chong Jia An	L3_Tue_PM_Chong Jia An_190_Archive
Chong Xuan Liang	L3_Thu_AM_Chong Xuan Liang_171_Archive
Christopher Nge Jing Qi	L3_Thu_AM_Christopher Nge Jing Qi_145_Archive
Christopher Tze-Wen Langton	L3_Thu_AM_Christopher TzeWen Langt_265_Archive
CHUA WAN NING	L3_Tue_PM_CHUA WAN NING_189_Archive
Chua Xiong Wei	L3_Thu_AM_Chua Xiong Wei_484_Archive
Chun Min Gyu	L3_Wed_PM_Chun Min Gyu_249_Archive
Chung Ying Qiao Winnie	L3_Tue_PM_Chung Ying Qiao Winnie_324_Archive
Conrad Ephraim Wee Cher Jae	L3_Thu_AM_Conrad Ephraim Wee Cher J_123_Archive
Cordell Chan Yi Hng	L3_Tue_PM_Cordell Chan Yi Hng_253_Archive
CUI MINJING	L3_Thu_AM_CUI MINJING_441_Archive
Cui Xinyu	L3_Tue_PM_Cui Xinyu_116_Archive
Damien Lim Yu Hao	L3_Thu_AM_Damien Lim Yu Hao_933_Archive
Darren Khoo Kah Weng	L3_Thu_AM_Darren Khoo Kah Weng_516_Archive
Darryl See	L3_Tue_PM_Darryl See_317_Archive
Desmond Eng Kian Wee	L3_Wed_PM_Desmond Eng Kian Wee_095_Archive
Donovan Sim Jing Yi	L3_Tue_PM_Donovan Sim Jing Yi_402_Archive
Du Yantang	L3_Tue_PM_Du Yantang_744_Archive
Edly Irsyad B Elham	L3_Thu_AM_Edly Irsyad B Elham_555_Archive
Elumalai Oviya Dharshini	L3_Thu_AM_Elumalai Oviya Dharshini_353_Archive
Eric Bryan	L3_Thu_AM_Eric Bryan_789_Archive
Eugene Chong Zhi Liang	L3_Thu_AM_Eugene Chong Zhi Liang_525_Archive
Fan Shixi	L3_Thu_AM_Fan Shixi_848_Archive
FOONG XIN YU	L3_Thu_AM_FOONG XIN YU_018_Archive
Fu Zhehui	L3_Tue_PM_Fu Zhehui_218_Archive
Gavien Pat Wei Zhuo	L3_Thu_AM_Gavien Pat Wei Zhuo_185_Archive
GOH KAI YAO BRYAN	L3_Tue_AM_GOH KAI YAO BRYAN_690_Archive
Goh Shao Quan	L3_Tue_PM_Goh Shao Quan_422_Archive
GOH SHAU HUI GEORGE	L3_Tue_PM_GOH SHAU HUI GEORGE_353_Archive
Goh Wei Yang	L3_Tue_PM_Goh Wei Yang_254_Archive

Gu Jianqiang	L3_Tue_PM_Gu Jianqiang_514_Archive
Guan Dinghe	L3_Tue_PM_Guan Dinghe_736_Archive
Han Si Yuan	L3_Wed_PM_Han Si Yuan_050_Archive
Hemanth Bangalore Srinivas Murthy	L3_Tue_PM_Hemanth Bangalore Sriniva_135_Archive
Ho Shu Jun	L3_Tue_PM_Ho Shu Jun_198_Archive
Ho Zhen Hong	L3_Thu_AM_Ho Zhen Hong_014_Archive
Hoang Trong Tan	L3_Thu_AM_Hoang Trong Tan_425_Archive
HOE JUN LEONG	L3_Thu_AM_HOE JUN LEONG_182_Archive
Hong Xingwen	L3_Tue_PM_Hong Xingwen_861_Archive
Hossan Goh Xuan Rong	L3_Thu_AM_Hossan Goh Xuan Rong_833_Archive
How Teck Wei	L3_Tue_PM_How Teck Wei_391_Archive
HU JIALUN	L3_Thu_AM_HU JIALUN_521_Archive
Hu Xuefei	L3_Thu_AM_Hu Xuefei_885_Archive
Huang Che Yen	L3_Thu_AM_Huang Che Yen_597_Archive
HUANG HAOFENG	L3_Mon_PM_HUANG HAOFENG_936_Archive
Huang Shanshan	L3_Tue_PM_Huang Shanshan_622_Archive
HUANG YUJING	L3_Tue_PM_HUANG YUJING_485_Archive
Ian Wang Ee En	L3_Thu_AM_Ian Wang Ee En_227_Archive
Imperial Edward Justin Javier	L3_Tue_PM_Imperial Edward Justin Ja_356_Archive
Ishaan Maunil Vyas	L3_Thu_AM_Ishaan Maunil Vyas_479_Archive
Izdiyad Farhan B Zuri	L3_Thu_AM_Izdiyad Farhan B Zuri_407_Archive
Jared Cheang	L3_Mon_PM_Jared Cheang_192_Archive
Jareth Tan Eu Quan	L3_Tue_PM_Jareth Tan Eu Quan_240_Archive
Jason Ong Meng Lee	L3_Tue_PM_Jason Ong Meng Lee_174_Archive
Jasshan Kumeresh	L3_Tue_PM_Jasshan Kumeresh_503_Archive
Jeremiah Jiang	L3_Tue_PM_Jeremiah Jiang_139_Archive
Jeremiah Ong Ray	L3_Tue_PM_Jeremiah Ong Ray_551_Archive
Jeremy Goh Liang Yi	L3_Tue_PM_Jeremy Goh Liang Yi_393_Archive
JIANG QIXIONG	L3_Mon_PM_JIANG QIXIONG_698_Archive
Jiang Xing Kai	L3_Thu_AM_Jiang Xing Kai_564_Archive
Jin Minyue	L3_Tue_AM_Jin Minyue_069_Archive
JIN YIXUAN	L3_Wed_PM_JIN YIXUAN_976_Archive
Joanne Wong Wei Yin	L3_Thu_AM_Joanne Wong Wei Yin_527_Archive
Joel Matthew Chiam Zhi Qiang	L3_Thu_AM_Joel Matthew Chiam Zhi Qi_250_Archive
Jon Lim Yong Kiat	L3_Thu_AM_Jon Lim Yong Kiat_782_Archive
Jonathan Mui Koy Kit	L3_Thu_AM_Jonathan Mui Koy Kit_534_Archive
Joshua Harsha Dass	L3_Thu_AM_Joshua Harsha Dass_714_Archive
Justin Fidelis Wong Jun Wen	L3_Thu_AM_Justin Fidelis Wong Jun W_326_Archive
Kairos Koh Jia Jun	L3_Thu_AM_Kairos Koh Jia Jun_149_Archive
Keh Wen Yang, Rachel	L3_Thu_AM_Keh Wen Yang Rachel_249_Archive
Kevinaldi Dwiastajulio Hunto	L3_Thu_PM_Kevinaldi Dwiastajulio Hu_912_Archive
Khoo Jia Le Isaac	L3_Thu_AM_Khoo Jia Le Isaac_733_Archive
Khor Sheng Hou	L3_Thu_PM_Khor Sheng Hou_443_Archive
Kishor Kumar Haribaskar	L3_Thu_AM_Kishor Kumar Haribaskar_481_Archive
Koh Meng Kiat, Kenneth	L3_Tue_AM_Koh Meng Kiat Kenneth_512_Archive
Koh Qianqi	L3_Thu_AM_Koh Qianqi_746_Archive
Koh Qin Ruo	L3_Thu_PM_Koh Qin Ruo_055_Archive
Koh Ruizhe Jerome	L3_Thu_PM_Koh Ruizhe Jerome_429_Archive
Kom Xing Yuan	L3_Thu_AM_Kom Xing Yuan_993_Archive
Kong Dak Nam	L3_Thu_AM_Kong Dak Nam_171_Archive
Koo Wei De	L3_Tue_PM_Koo Wei De_133_Archive
Krishna R R	L3_Thu_PM_Krishna R R_900_Archive
Kum Wing Ho	L3_Thu_AM_Kum Wing Ho_725_Archive
Kumaravel Vignesh	L3_Thu_AM_Kumaravel Vignesh_585_Archive
Kwok Xiu Sheng Theodore	L3_Thu_AM_Kwok Xiu Sheng Theodore_118_Archive
Kwong Zhi Qian	L3_Thu_AM_Kwong Zhi Qian_569_Archive
Lam Junyu William	L3_Thu_AM_Lam Junyu William_903_Archive
LAM KAI WEN JONATHAN	L3_Thu_AM_LAM KAI WEN JONATHAN_213_Archive
Lau Miang Puang, Glennard	L3_Thu_PM_Lau Miang Puang Glennard_207_Archive
Lee An Sheng	L3_Tue_AM_Lee An Sheng_263_Archive
Lee Cheok Feng	L3_Tue_AM_Lee Cheok Feng_743_Archive
Lee Hung Tien	L3_Thu_PM_Lee Hung Tien_965_Archive
Lee Jia Jun	L3_Thu_PM_Lee Jia Jun_423_Archive
Lee Jing Rui, Evan	L3_Thu_PM_Lee Jing Rui Evan_399_Archive
Lee Jun Wen	L3_Wed_PM_Lee Jun Wen_466_Archive
LEE KENG YONG JOSHUA	L3_Tue_AM_LEE KENG YONG JOSHUA_644_Archive
Lee Qi An	L3_Tue_AM_Lee Qi An_644_Archive
Lee Shi-An, Matthew	L3_Thu_PM_Lee ShiAn Matthew_324_Archive
Lee Sungmin	L3_Tue_AM_Lee Sungmin_490_Archive
Lee Sze Ern, Jeremy	L3_Tue_PM_Lee Sze Ern Jeremy_510_Archive
Lee Yi Kai	L3_Tue_AM_Lee Yi Kai_550_Archive
Lee Yu-Hsueh	L3_Thu_PM_Lee YuHsueh_639_Archive

LEONARD CHUA ZHONG QI	L3_Tue_PM_LEONARD CHUA ZHONG QI_873_Archive
Leong Kah Choong	L3_Thu_PM_Leong Kah Choong_455_Archive
Leow Yuan Yang	L3_Tue_AM_Leow Yuan Yang_574_Archive
Leroy Ong Nai Kiat	L3_Thu_PM_Leroy Ong Nai Kiat_506_Archive
Li Huanda	L3_Thu_PM_Li Huanda_101_Archive
Li Xi Yuan	L3_Tue_AM_Li Xi Yuan_926_Archive
Liang Yuzhao	L3_Tue_AM_Liang Yuzhao_802_Archive
LIM BING SEN	L3_Wed_PM_LIM BING SEN_580_Archive
LIM CHANG QUAN THADDEUS	L3_Thu_AM_LIM CHANG QUAN THADDEUS_210_Archive
Lim Jia Sheng Jackson	L3_Thu_PM_Lim Jia Sheng Jackson_596_Archive
Lim Kay Yun	L3_Tue_AM_Lim Kay Yun_244_Archive
Lim Shyun Yin	L3_Tue_AM_Lim Shyun Yin_063_Archive
Lim Wen Jie	L3_Thu_PM_Lim Wen Jie_382_Archive
Liu Danfeng	L3_Thu_PM_Liu Danfeng_777_Archive
Liu Ruijun	L3_Thu_PM_Liu Ruijun_371_Archive
LIU ZEHANG	L3_Tue_AM_LIU ZEHANG_491_Archive
LIU ZHIYANG	L3_Wed_PM_LIU ZHIYANG_653_Archive
Loo Keng Leong	L3_Thu_PM_Loo Keng Leong_408_Archive
LOW ZHEN WEI JERRELL	L3_Tue_AM_LOW ZHEN WEI JERRELL_516_Archive
Lu Jingguang	L3_Thu_PM_Lu Jingguang_319_Archive
Lu Sicheng	L3_Tue_AM_Lu Sicheng_634_Archive
LU ZONGHAN	L3_Tue_PM_LU ZONGHAN_452_Archive
MA XUDONG	L3_Tue_PM_MA XUDONG_493_Archive
Ma Zijian	L3_Tue_AM_Ma Zijian_828_Archive
Madhan Selvapandian	L3_Tue_AM_Madhan Selvapandian_482_Archive
Madheswaran Niveytha	L3_Thu_PM_Madheswaran Niveytha_465_Archive
Mah Yuan Jie Alvin	L3_Tue_AM_Mah Yuan Jie Alvin_500_Archive
Mahadevan Svetha	L3_Tue_AM_Mahadevan Svetha_108_Archive
Mahadevan Swati	L3_Tue_AM_Mahadevan Swati_107_Archive
Marcus Goh Xuan De	L3_Thu_PM_Marcus Goh Xuan De_355_Archive
Marcus Lim Sheng Jie	L3_Thu_PM_Marcus Lim Sheng Jie_408_Archive
Marcus Ong Yih	L3_Tue_AM_Marcus Ong Yih_404_Archive
Mathur Aayush	L3_Tue_AM_Mathur Aayush_581_Archive
Mayank Panjiyara	L3_Tue_AM_Mayank Panjiyara_763_Archive
Mehedi Hasan Salim	L3_Thu_PM_Mehedi Hasan Salim_436_Archive
Mohamed Faez Bin Shahlan	L3_Thu_PM_Mohamed Faez Bin Shahlan_297_Archive
Mohammad Shoib Memon Loya	L3_Tue_AM_Mohammad Shoib Memon Loya_487_Archive
Muhammad Aizat Bin Rahim	L3_Thu_PM_Muhammad Aizat Bin Rahim_437_Archive
Muhammad Ashraf B Mohamad J	L3_Tue_AM_Muhammad Ashraf B Mohamad_432_Archive
MUHAMMAD HAZIM BIN ABDULLAH	L3_Thu_PM_MUHAMMAD HAZIM BIN ABDULL_633_Archive
Muhammad Jaish Bin Jamalun Nasir	L3_Wed_PM_Muhammad Jaish Bin Jamalu_287_Archive
Mun Le Zong	L3_Tue_AM_Mun Le Zong_172_Archive
Nan Song	L3_Thu_PM_Nan Song_102_Archive
Ng Andre	L3_Tue_AM_Ng Andre_973_Archive
Ng Cheng Yang, Titus	L3_Tue_AM_Ng Cheng Yang Titus_478_Archive
Ng Jin Loong, Jeremy	L3_Thu_PM_Ng Jin Loong Jeremy_395_Archive
Ng Qi Hao	L3_Thu_PM_Ng Qi Hao_420_Archive
Ng Xinyi	L3_Thu_PM_Ng Xinyi_545_Archive
Ngoi Hui Wen, Vanessa	L3_Tue_AM_Ngoi Hui Wen Vanessa_471_Archive
Nguyen Minh Tuan	L3_Tue_AM_Nguyen Minh Tuan_389_Archive
Nguyen Van Binh	L3_Mon_PM_Nguyen Van Binh_453_Archive
Nigel Loh Weien	L3_Thu_PM_Nigel Loh Weien_416_Archive
Nigel Ng	L3_Mon_PM_Nigel Ng_444_Archive
Nishant Rai	L3_Mon_PM_Nishant Rai_182_Archive
Oh Qi Ren	L3_Thu_PM_Oh Qi Ren_441_Archive
Ong Chor Yew	L3_Tue_PM_Ong Chor Yew_460_Archive
Ong Jun Giat	L3_Thu_PM_Ong Jun Giat_205_Archive
Ong Siying Falicia	L3_Thu_AM_Ong Siying Falicia_566_Archive
Ong Yew Yong, Adrian	L3_Thu_PM_Ong Yew Yong Adrian_401_Archive
Owng Kai Leng Sally	L3_Thu_PM_Owng Kai Leng Sally_331_Archive
Pang Kai Lin	L3_Thu_PM_Pang Kai Lin_036_Archive
Pang Kai Yi	L3_Thu_PM_Pang Kai Yi_236_Archive
Pang Qi Wei, Jenna	L3_Wed_PM_Pang Qi Wei Jenna_685_Archive
Paramita Tejasvi	L3_Mon_PM_Paramita Tejasvi_194_Archive
Peh Li Yan	L3_Thu_PM_Peh Li Yan_267_Archive
PENG FEI	L3_Mon_PM_PENG FEI_518_Archive
Peng Yanjia	L3_Tue_AM_Peng Yanjia_877_Archive
Phoon Pei Zhen	L3_Thu_PM_Phoon Pei Zhen_744_Archive
Phua Keng Wee	L3_Thu_PM_Phua Keng Wee_706_Archive
Phuah Yong Chen Keith	L3_Thu_PM_Phuah Yong Chen Keith_360_Archive
Pichanon Rattanadilok Na Phuket	L3_Tue_AM_Pichanon Rattanadilok Na _545_Archive
Pojcharapol Leenukiat	L3_Wed_PM_Pojcharapol Leenukiat_642_Archive

Poon Jeun Lek	L3_Wed_PM_Poon Jeun Lek_202_Archive
Pow Zhi Xiang	L3_Tue_AM_Pow Zhi Xiang_942_Archive
Pradhan Rachit Manish	L3_Wed_PM_Pradhan Rachit Manish_230_Archive
Pranav Venkatram	L3_Tue_AM_Pranav Venkatram_200_Archive
Qi Tian Cong	L3_Wed_PM_Qi Tian Cong_442_Archive
Ramalingam Saravanamani	L3_Tue_AM_Ramalingam Saravanamani_586_Archive
Ravindiran Rakesh	L3_Tue_AM_Ravindiran Rakesh_010_Archive
Rebecca Chua	L3_Thu_AM_Rebecca Chua_171_Archive
REN TIANLE	L3_Mon_PM_REN TIANLE_446_Archive
Renzo Rivera Canare	L3_Thu_AM_Renzo Rivera Canare_502_Archive
Reuel Teo Lu Wei	L3_Wed_PM_Reuel Teo Lu Wei_435_Archive
Richard Willie	L3_Thu_AM_Richard Willie_368_Archive
Roycius Lim Yuanwei	L3_Thu_AM_Roycius Lim Yuanwei_060_Archive
Samuel Ong Wei Chuan	L3_Tue_PM_Samuel Ong Wei Chuan_462_Archive
Se Sean	L3_Wed_PM_Se Sean_140_Archive
See Jian Hui	L3_Thu_AM_See Jian Hui_737_Archive
Seet Ting Yang Irvin	L3_Mon_PM_Seet Ting Yang Irvin_608_Archive
Seetoh Yit Ching	L3_Mon_PM_Seetoh Yit Ching_154_Archive
Seo Gimin	L3_Mon_PM_Seo Gimin_442_Archive
Seth Teng Shann	L3_Wed_PM_Seth Teng Shann_419_Archive
Shao Yurui	L3_Mon_PM_Shao Yurui_111_Archive
Shawn Chang	L3_Wed_PM_Shawn Chang_151_Archive
Shreshth Sarda	L3_Wed_PM_Shreshth Sarda_424_Archive
Shyam Ganesh Jayagopi	L3_Mon_PM_Shyam Ganesh Jayagopi_484_Archive
Sidharth Premnath	L3_Wed_PM_Sidharth Premnath_502_Archive
Siew Yang Zhi	L3_Thu_AM_Siew Yang Zhi_331_Archive
Sim Le Yee Beatrice	L3_Wed_PM_Sim Le Yee Beatrice_769_Archive
Sin Ren Xiang	L3_Thu_PM_Sin Ren Xiang_865_Archive
Sivakumar Yogarajan	L3_Wed_PM_Sivakumar Yogarajan_505_Archive
Song Chenan	L3_Wed_PM_Song Chenan_797_Archive
Song Min Kyu	L3_Wed_PM_Song Min Kyu_226_Archive
Sridharan Arvind Srinivasan	L3_Mon_PM_Sridharan Arvind Srinivas_477_Archive
Sthitipragyan Samal	L3_Mon_PM_Sthitipragyan Samal_664_Archive
Sun Jiale	L3_Mon_PM_Sun Jiale_853_Archive
SUN JIAWEI	L3_Thu_PM_SUN JIAWEI_496_Archive
SWAMINATHAN VARUN	L3_Mon_PM_SWAMINATHAN VARUN_281_Archive
Swann Tet Aung	L3_Mon_PM_Swann Tet Aung_552_Archive
Tan Haoxuan	L3_Wed_PM_Tan Haoxuan_934_Archive
Tan Hui En	L3_Mon_PM_Tan Hui En_373_Archive
Tan Jun Heng Daren Justin	L3_Mon_PM_Tan Jun Heng Daren Justin_331_Archive
Tan Kah Heng	L3_Mon_PM_Tan Kah Heng_677_Archive
Tan Le Yi	L3_Mon_PM_Tan Le Yi_071_Archive
Tan Lindsey	L3_Wed_PM_Tan Lindsey_197_Archive
Tan Qi Xian, Keith	L3_Wed_PM_Tan Qi Xian Keith_397_Archive
Tan Rui Yang	L3_Mon_PM_Tan Rui Yang_472_Archive
Tan Tze Yeong	L3_Wed_PM_Tan Tze Yeong_970_Archive
Tan Wei Li	L3_Wed_PM_Tan Wei Li_336_Archive
Tan Xing Jie	L3_Mon_PM_Tan Xing Jie_747_Archive
Tan Yong Zheng	L3_Wed_PM_Tan Yong Zheng_261_Archive
Tang Zehou	L3_Mon_PM_Tang Zehou_210_Archive
Tay Weida	L3_Mon_PM_Tay Weida_027_Archive
Tay Yi Heng, Atticus	L3_Mon_PM_Tay Yi Heng Atticus_994_Archive
Teh Jiewen	L3_Mon_PM_Teh Jiewen_520_Archive
Teh Zi-Chun	L3_Wed_PM_Teh ZiChun_328_Archive
Teng Yi Shiong	L3_Mon_PM_Teng Yi Shiong_647_Archive
Teo Ziyi Ivy	L3_Mon_PM_Teo Ziyi Ivy_117_Archive
Tham Yang Tze Xavier	L3_Wed_PM_Tham Yang Tze Xavier_256_Archive
TIAN ZHENYU	L3_Thu_PM_TIAN ZHENYU_467_Archive
Tiang Zhang Quan Xavier	L3_Wed_PM_Tiang Zhang Quan Xavier_446_Archive
Tie Zhou Peng	L3_Wed_PM_Tie Zhou Peng_264_Archive
Toh Yi Cheng	L3_Wed_PM_Toh Yi Cheng_421_Archive
Toh Yi Zhi	L3_Mon_PM_Toh Yi Zhi_086_Archive
Tran Nhan Duc Anh	L3_Mon_PM_Tran Nhan Duc Anh_358_Archive
Tran Thi Phuong Thao	L3_Wed_PM_Tran Thi Phuong Thao_438_Archive
Varun Agarwal	L3_Wed_PM_Varun Agarwal_605_Archive
VIKAS HARLANI	L3_Mon_PM_VIKAS HARLANI_376_Archive
Vishal Jeyaram	L3_Mon_PM_Vishal Jeyaram_224_Archive
Wan Haocheng	L3_Wed_PM_Wan Haocheng_780_Archive
Wang Wenxuan	L3_Wed_PM_Wang Wenxuan_649_Archive
WANG YUDA	L3_Thu_PM_WANG YUDA_443_Archive
Wang Zhao Yu, Edward	L3_Mon_PM_Wang Zhao Yu Edward_953_Archive
Wang Zhihuang	L3_Mon_PM_Wang Zhihuang_682_Archive

Wang Zichen	L3_Wed_PM_Wang Zichen_951_Archive
Wang Zihan	L3_Wed_PM_Wang Zihan_361_Archive
Wang Zixi	L3_Wed_PM_Wang Zixi_445_Archive
William Wahyudi	L3_Mon_PM_William Wahyudi_230_Archive
Wong Jun Lin	L3_Tue_AM_Wong Jun Lin_077_Archive
Wong Tze Shan Samantha	L3_Mon_PM_Wong Tze Shan Samantha_672_Archive
Wong Zi Xin, Avellin	L3_Mon_PM_Wong Zi Xin Avellin_073_Archive
Woo Bo Tuan	L3_Mon_PM_Woo Bo Tuan_153_Archive
WU HAO HSUAN	L3_Thu_PM_WU HAO HSUAN_635_Archive
Wu Luoyu	L3_Mon_PM_Wu Luoyu_894_Archive
WU YUWEI	L3_Wed_PM_WU YUWEI_472_Archive
XIAO JUNTIAN	L3_Wed_PM_XIAO JUNTIAN_497_Archive
Xu Yuxing	L3_Wed_PM_Xu Yuxing_183_Archive
Xue Yuxuan	L3_Wed_PM_Xue Yuxuan_250_Archive
Yam Jin Ee Dmitri	L3_Mon_PM_Yam Jin Ee Dmitri_974_Archive
Yang Zikun	L3_Tue_AM_Yang Zikun_313_Archive
Yap Joon Siong	L3_Mon_PM_Yap Joon Siong_925_Archive
YAP WEI XUAN	L3_Mon_PM_YAP WEI XUAN_997_Archive
Yap Zhan Wei	L3_Thu_PM_Yap Zhan Wei_455_Archive
Yeat Nai Jie	L3_Mon_PM_Yeat Nai Jie_613_Archive
Yeo Shi Heng	L3_Wed_PM_Yeo Shi Heng_390_Archive
Yeo Wei Hng	L3_Mon_PM_Yeo Wei Hng_075_Archive
Yeo Zi Hao, Edwin	L3_Mon_PM_Yeo Zi Hao Edwin_710_Archive
YIP WAYNE	L3_Mon_PM_YIP WAYNE_998_Archive
YU HAIHONG	L3_Wed_PM_YU HAIHONG_470_Archive
Yue Junfeng	L3_Thu_AM_Yue Junfeng_802_Archive
Yuk Yeon Soo	L3_Wed_PM_Yuk Yeon Soo_243_Archive
Zeng Jiexiong	L3_Mon_PM_Zeng Jiexiong_052_Archive
ZHANG HAOYU	L3_Thu_AM_ZHANG HAOYU_783_Archive
ZHAO LUOYUANG	L3_Mon_PM_ZHAO LUOYUANG_466_Archive
Zhao Yibo	L3_Wed_PM_Zhao Yibo_863_Archive
Zhao Ziqi	L3_Wed_PM_Zhao Ziqi_275_Archive
ZHONG XINGHAN	L3_Mon_PM_ZHONG XINGHAN_468_Archive
ZHOU CHENGXU	L3_Mon_PM_ZHOU CHENGXU_492_Archive
ZHOU YUHAN	L3_Wed_PM_ZHOU YUHAN_530_Archive
Zhu Shaohan Steven	L3_Wed_PM_Zhu Shaohan Steven_193_Archive
Zhuang Jianning	L3_Mon_PM_Zhuang Jianning_277_Archive
Zubin Jain	L3_Tue_PM_Zubin Jain_990_Archive