EE2026: DIGITAL DESIGN

Academic Year 2020-2021, Semester 1

LAB 2: Combinational Circuits in Verilog

OVERVIEW:

A combinational circuit is one where the outputs depend only on the current inputs. In this lab, we will be designing some combinational circuits that are able to perform addition.

The pre-requisite for this lab requires one to be able to:

- Create a Verilog project and design source in Vivado.
- Create a testbench to simulate the design source.
- Generate the RTL schematic and the synthesised circuit schematic of design source.
- Map and implement a design on the Basys 3 development board.
- Understand well the contents of Lectures 1, 2, and especially 3: Introduction to Verilog.

This lab will cover the following:

- Designing a one-bit full adder circuit using the dataflow modelling method.
- Designing a two-bit parallel adder circuit using the structural modelling method.

Tasks for this lab include:

- Designing the Verilog code of a one-bit full adder.
- Designing, simulating and implementing a four-bit parallel adder on the FPGA.

GRADED ASSIGNMENT [LUMINUS SUBMISSION: TUESDAY 15th SEPTEMBER 2020, NOON]:

- Designing, simulating and implementing an n-bit subtractor on the Basys 3 development board

Further details are available at the end of this lab manual

ONE-BIT-FULL ADDER:

Consider the binary addition shown in ***Figure 2.1***. To design a circuit that would perform the addition of two one-bit values, the circuit would need to have three input bits and two output bits.

Figure 2.1: *Binary Addition and functional block diagram of the one-bit full adder*

Such a circuit is called a one-bit full adder. It adds two bits (**A**, **B**) and the carry ($\mathbf{C_{in}}$) from a previous stage of addition, and produces a sum (**S**) and a carry ($\mathbf{C_{out}}$), as illustrated through a truth table in ***Figure 2.2***. By simplifying the truth table, the Boolean expressions for **S** and $\mathbf{C_{out}}$ can be obtained.

A	B	C_{in}	S	C_{out}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

$$S = A \oplus B \oplus C_{in}$$

$$C_{out} = AB + C_{in}\,(A \oplus B)$$

Figure 2.2: *Truth table and boolean expressions of the one-bit full adder*

Note: A half adder, in contrast to a full adder, does not involve a carry input. Thus, for a half adder, $S = A \oplus B$, *and* $C_{out} = AB$

UNDERSTANDING | TASK 1

Using the dataflow method, complete the Verilog code for the one-bit full adder. Verify that the RTL schematic is as shown.

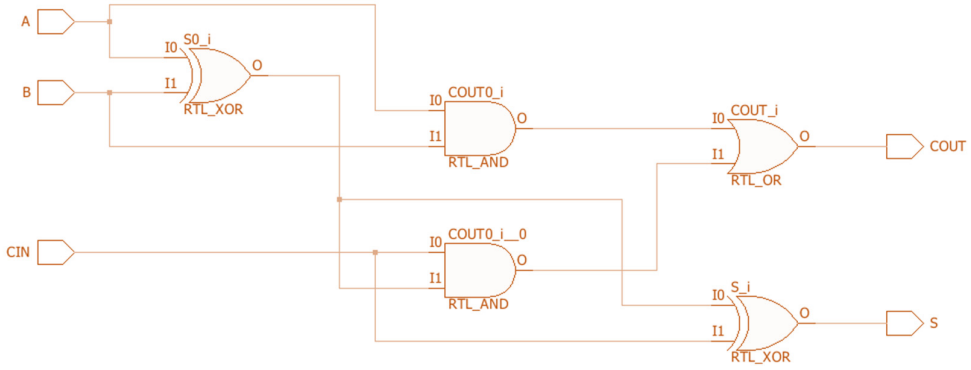
Verilog skeleton code for the one-bit-full adder:

```
module my_full_adder(input A, B, CIN, output S, COUT);

    assign S =
    assign COUT =

endmodule
```

RTL schematic for the one-bit full adder:



TWO-BIT FULL ADDER:

By cascading one-bit full adder blocks, the one-bit adder can be reused and parallel adders that add multiple bits can be created through the structural modelling method. A two-bit ripple-carry adder is illustrated in ***Figure 2.3***.

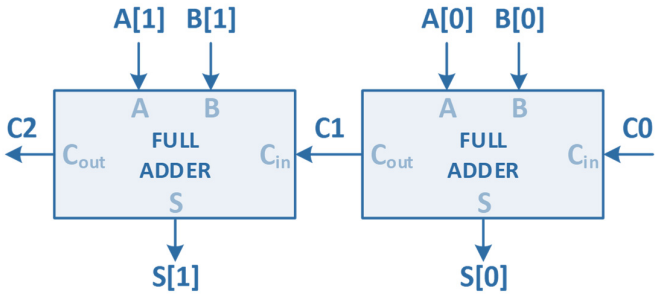


Figure 2.3: *Functional block diagram of the two-bit ripple-carry adder*

With the code for a one-bit full adder, a new module is created and two full adder blocks (fa0, fa1) are instantiated. By specifying the inputs and outputs to these blocks, the connection **C1** between them is made. Note that the order of signals during instantiation should respect the order in which they were declared in the one-bit full adder module **my_full_adder**.

This approach to hardware description is called structural modelling, whereby a more abstract module (for example, **my_2_bit_adder**) is built from simpler components describing gate-level hardware (such as **my_full_adder**).

Verilog code for two-bit ripple-carry adder, using structural modelling:

```
module my_2_bit_adder(input [1:0] A, input [1:0] B, input C0,
                      output [1:0] S, output C2);

    wire C1;

    my_full_adder fa0 (A[0], B[0], C0, S[0], C1);
    my_full_adder fa1 (A[1], B[1], C1, S[1], C2);

endmodule
```

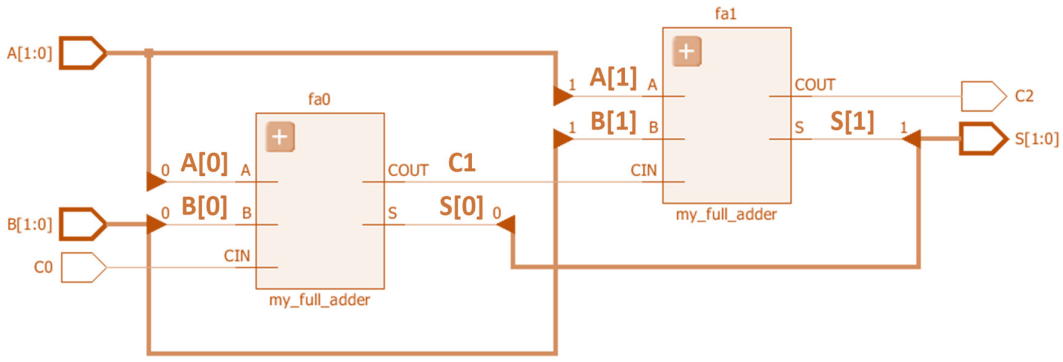
[Note] Two-bit port declaration for A, B, S:

Instead of having multiple input and output ports (A1, A0, B1, B0, S1, S0), multi-bit vector ports are defined.

Example: `input [5:0] apple`

Input port name is **apple**, with size of 6 bits. apple[5] refers to the MSB, apple[0] refers to the LSB.

RTL schematic for the two-bit ripple-carry adder:



FOUR-BIT FULL ADDER:

The functional block diagram of a four-bit ripple-carry adder is shown in ***Figure 2.4***.

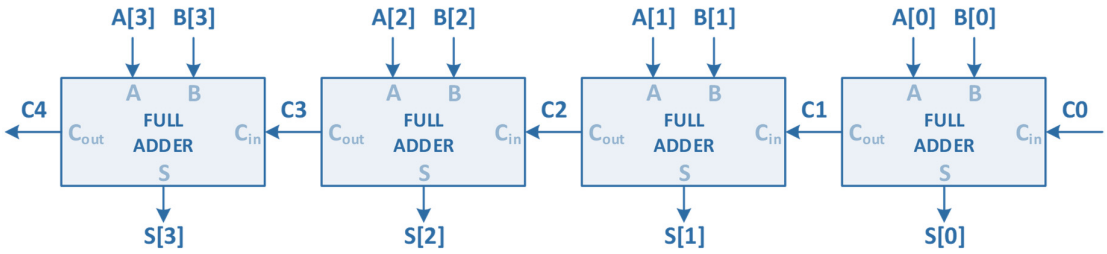


Figure 2.4: *Functional block diagram of a four-bit ripple-carry adder*

UNDERSTANDING | TASK 2

Start by adding a new design source for the four-bit full adder.

1. By using the structural modelling method, design a four-bit ripple-carry adder.
2. Generate the RTL schematic and check that connections between the blocks are correct.
3. Simulate your code with the following three sets of input values, and check that the simulation outputs are correct:

A:		0	0	1	1	A:		1	0	1	1	A:		1	1	1	1
B:	+	0	0	1	1	B:	+	0	1	1	1	B:	+	1	1	1	1
		□	□	□	□			□	□	□	□			□	□	□	□

[Note] Brief guidelines for multi-bit vector ports in the testbench:

- The port size should be indicated when declaring the signals. Inputs to the module being tested are declared using **reg**, whereas outputs from the module being tested are declared using **wire**:
  `reg [3:0] A; wire [3:0] S;`

- The parameters for the module being tested do not need the port size of the signals:
  `my_4_bit_adder module_alias (A, B, CARRY_IN, S, CARRY_OUT);`

- The testbench stimuli for multi-bit vector ports can be written as:
  `A = 4'b0011; B = 4'b0011; CARRY_IN = 1'b0;`

4. Synthesise and implement your code on the FPGA, using appropriate switches and LEDs on the Basys 3 development board to represent the inputs and outputs.

This task is considered completed and understood if you have the following items related to the four-bit full adder:

- The RTL schematic of your design (item 2 of this task)
- The simulation waveform results of the three testbench stimuli (item 3 of this task)
- The four-bit ripple-carry adder on the Basys 3 development board (item 4 of this task)

UNDERSTANDING | TASK 3

Instead of using four one-bit adder blocks, can you think of an alternative way (still using structural modelling) in creating the four-bit ripple-carry adder? Consider doing the same things as indicated in **UNDERSTANDING | TASK 2** by using such alternative way. This task is meant as practice for you to improve your Vivado skills and Verilog understanding, and will not be explained.

GRADED POST-LAB ASSIGNMENT – THE SUBTRACTOR:

ASSIGNMENT

Based on your student matriculation number, you are required to create an n-bit parallel subtractor by combining several n-bit parallel adders. The RTL schematics of the n-bit parallel subtractor will need to be produced, the system simulated, and finally implemented on the Basys 3 development board.

REQUIREMENT BASED ON YOUR STUDENT MATRICULATION NUMBER

Based on the **second** rightmost numerical digit of your student matriculation number, create the following n-bit parallel adders:

Second rightmost numerical digit of your student matriculation number	Required n-bit parallel adders. Each one of these two b-bits parallel adders must be made up of multiple 1-bit adders	
0	2-bit parallel adder	3-bit parallel adder
1	3-bit parallel adder	2-bit parallel adder
2	2-bit parallel adder	4-bit parallel adder
3	4-bit parallel adder	2-bit parallel adder
4	3-bit parallel adder	4-bit parallel adder
5	4-bit parallel adder	3-bit parallel adder
6	3-bit parallel adder	5-bit parallel adder
7	5-bit parallel adder	3-bit parallel adder
8	2-bit parallel adder	5-bit parallel adder
9	5-bit parallel adder	2-bit parallel adder

After doing so, **make use of both n-bit parallel adders** to create your n-bit parallel subtractor, based on the **second** rightmost numerical digit of your student matriculation number. If required, you may assume that the user will not input any carry-in value to the n-bit parallel subtractor.

The bits given to the subtractor must be divided between the two n-bit parallel adders in the following manner:

Second rightmost numerical digit of your student matriculation number	**Most significant bits of the subtractor** must be given to the following adder	**Least significant bits of the subtractor** must be given to the following adder
0	2-bit parallel adder	3-bit parallel adder
1	3-bit parallel adder	2-bit parallel adder
2	2-bit parallel adder	4-bit parallel adder
3	4-bit parallel adder	2-bit parallel adder
4	3-bit parallel adder	4-bit parallel adder
5	4-bit parallel adder	3-bit parallel adder
6	3-bit parallel adder	5-bit parallel adder
7	5-bit parallel adder	3-bit parallel adder
8	2-bit parallel adder	5-bit parallel adder
9	5-bit parallel adder	2-bit parallel adder

Furthermore, it is compulsory to use the specified complement representation of the binary values, as indicated in the table below:

Second rightmost numerical digit of your student matriculation number	Required n-bit parallel subtractor
0	5-bit subtractor. Bits are in **2's complement**
2	6-bit subtractor. Bits are in **2's complement**
4	7-bit subtractor. Bits are in **2's complement**
6	8-bit subtractor. Bits are in **2's complement**
8	7-bit subtractor. Bits are in **2's complement**
1	5-bit subtractor. Bits are in **1's complement**
3	6-bit subtractor. Bits are in **1's complement**
5	7-bit subtractor. Bits are in **1's complement**
7	8-bit subtractor. Bits are in **1's complement**
9	7-bit subtractor. Bits are in **1's complement**

Example 1: If a student has matriculation number A1234567N, the second rightmost numerical digit is '6'. Thus, the student is required to create an 8-bit parallel subtractor, whereby all the bits are considered to be in 2's complement representation. That 8-bit subtractor must consist, at minimum, of these two parallel adders: A 3-bit parallel adder for the most significant bits, and a 5-bit parallel adder for the least significant bits.

Example 2: If a student has matriculation number A6543210Y, the second rightmost numerical digit is '1'. Thus, the student is required to create a 5-bit parallel subtractor, whereby all the bits are considered to be in 1's complement representation. That 5-bit subtractor must consist, at minimum, of these two parallel adders: A 3-bit parallel adder for the most significant bits, and a 2-bit parallel adder for the least significant bits.

SIMULATION REQUIREMENT:

You are also required to simulate any ten (10) unique test cases as chosen by you, for your n-bit subtractor circuit, to verify that the outputs are as desired. If the number of unique test cases is less than ten (10), full marks will not be achievable.

A screenshot (PrintScreen) of the simulation waveform pattern from the Vivado simulation window must be obtained, and then pasted on a 1-page landscape page. Marks are only given if the simulation waveforms are clear, and **all the values** are **clearly visible in binary representations**.

RTL ANALYSIS SCHEMATIC REQUIREMENT:

A screenshot (PrintScreen) of the RTL analysis schematic from Vivado must be obtained, and pasted on a 1-page landscape page. You are not allowed to do any editing to the RTL analysis schematic screenshot.

For marks to be awarded, the following conditions must be met for the RTL analysis schematics (The images being shown here is for a 9-bit subtractor system, where a 4-bit parallel adder was used for the 4 most significant bits, and a 5-bit parallel adder was used for the 5 least significant bits):

(1) None of the modules are expanded (That is, screenshot the updated and properly zoomed schematics directly after clicking on RTL Analysis -> Schematics). Furthermore, **a minimum of two n-bit parallel adder modules** (Based on your second rightmost numerical digit of your student matriculation number) **must be seen without any modules in the schematics being expanded.**

(2) All inputs of the subtractor must be clear, including the numerical values between the square brackets. For example:

(3) All outputs of the subtractor must be clear, including the numerical values between the square brackets. For example:

(4) All inputs to, and outputs from, the n-bit parallel adder modules must be clear, including the numerical values between the square brackets. For example, in a 9-bit subtractor, there are at least two n-bit parallel adders (Based on your second rightmost numerical digit of your student matriculation number) being used:

DOCUMENT UPLOAD REQUIREMENTS:

(1) Your name and matriculation number on the first page of the document.
(2) The screenshot (PrintScreen) of the simulation waveform pattern from the Vivado simulation window on the first page of the document.
(3) The screenshot (PrintScreen) of the RTL analysis schematic from Vivado on the second page of the document.

Print the two pages landscape document as a **single PDF** for LumiNUS upload. The PDF file **must follow the naming template** indicated in the LumiNUS submission instruction at the end of this lab manual.

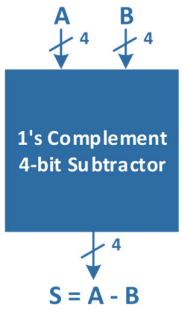
EXAMPLE OF OUTPUTS FOR A 4-BIT PARALLEL SUBTRACTOR

Assume that the two inputs to the subtractor are A and B. The output from the subtractor is S. In this example, we assume all the inputs and output are 4 bits.

Since $A - B = A + (-B)$, by adding A to $-B$, the answer to $A - B$ can be obtained.
1's or 2's complement can be used to find $-B$ from B.

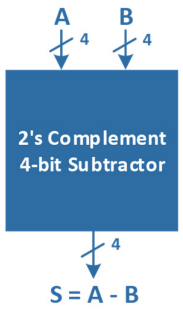
Examples on the expected output S, for some input values of A and B

One's Complement



A	B	S
0101 (+5)	0001 (+1)	0100 (+4)
0101 (+5)	1110 (−1)	0110 (+6)
1110 (−1)	0101 (+5)	1001 (−6)
1110 (−1)	1010 (−5)	0100 (+4)

Two's Complement



A	B	S
0101 (+5)	0001 (+1)	0100 (+4)
0101 (+5)	1111 (−1)	0110 (+6)
1111 (−1)	0101 (+5)	1010 (−6)
1111 (−1)	1011 (−5)	0100 (+4)

FURTHER REQUIREMENTS AND RESTRICTIONS: OPERATORS, SWITCHES AND LEDS

- **It is compulsory to use the following switches to represent inputs A and B of the n-bit parallel subtractor:**
  - The input A must use consecutive switches on the Basys 3 Development board, with the **least significant bit A[0] linked to SW0**. Similarly, A[1] should be linked to SW1, A[2] to SW2, and so on.
  - The input B must use consecutive switches on the Basys 3 Development board, with the **least significant bit B[0] linked to SW8**. Similarly, B[1] should be linked to SW9, B[2] to SW10, and so on.
  - **Switches not used by A or B must NOT be used or linked to any other signals under any circumstances.** For example, in a 4-bit parallel subtractor, SW4 to SW7, and SW12 and SW15, should not be linked to any signals, nor used in any circumstances.
- **It is compulsory to use the following LEDs to represent output S of the n-bit parallel subtractor:**
  - The output S must use consecutive LEDs on the Basys 3 development board, with the **least significant bit S[0] linked to LD0**. Similarly, S[1] should be linked to LD1, S[2] should be linked to LD2, and so on.
  - **LD15 must be used to represent the carry bit of final output S**, even if it is always on or off depending on the complement.
  - **LEDs not used by S must NOT be used or linked to any other signals under any circumstances.** For example, in a 4-bit parallel subtractor, LD4 to LD14, should not be linked to any signals, nor used in any circumstances.
- The addition and subtraction operators (+, −) are **not allowed** within the code.
- Pushbuttons may be used as a last resort only if you cannot complete the full subtractor without it. However, note that the subtractor should be easy to use from the user's point of view.

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 2 MB in size for lab 2.** 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.

- Separately from the project archive, **the single PDF document** (Simulation waveform + RTL analysis schematic of the subtractor) is to be named as indicated in the appendix of this lab manual.

- Upload to LumiNUS EE2026 -> Files -> Lab and Project - Materials and Submissions -> Lab 2 Submission

- Download your LumiNUS archive after uploading. **Unzip it / Extract all, 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 **Tuesday 15th September 2020, 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 if you have already submitted on time, 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 2 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 (Renaming submissions just before LumiNUS upload):

It is **compulsory to rename your project archive and report**, just before LumiNUS upload, as indicated in the table below. Copy your respective "Archive Naming" for the archive, and "Report Naming" for the report, and then replace the "xxxxxxxxx" with your student ID number. Do not change any other part of the naming, except the "xxxxxxxxx".

Submission example for project archive: L2_Fri_AM_Alan Turing_Archive_A0131086Z
Submission example for project report: L2_Fri_AM_Alan Turing_Report_A0131086Z.pdf

Name	Archive Naming	Report Naming
A AKIL AHAMED	L2_Fri_AM_A AKIL AHAMED_Archive_XXXXXXXXX	L2_Fri_AM_A AKIL AHAMED_Report_XXXXXXXXX
Abdul Hadi Bin Abdul Samad	L2_Fri_AM_Abdul Hadi Bin Abdul_Archive_XXXXXXXXX	L2_Fri_AM_Abdul Hadi Bin Abdul_Report_XXXXXXXXX
Adabelle Lim Ru Leng	L2_Fri_AM_Adabelle Lim Ru Leng_Archive_XXXXXXXXX	L2_Fri_AM_Adabelle Lim Ru Leng_Report_XXXXXXXXX
Alfred Wrong Jia Qing	L2_Fri_AM_Alfred Wrong Jia Qin_Archive_XXXXXXXXX	L2_Fri_AM_Alfred Wrong Jia Qin_Report_XXXXXXXXX
Alvin Goh Jia Hao	L2_Fri_AM_Alvin Goh Jia Hao_Archive_XXXXXXXXX	L2_Fri_AM_Alvin Goh Jia Hao_Report_XXXXXXXXX
Alvinci Merquita	L2_Wed_AM_Alvinci Merquita_Archive_XXXXXXXXX	L2_Wed_AM_Alvinci Merquita_Report_XXXXXXXXX
ANG KENG SIANG	L2_Wed_AM_ANG KENG SIANG_Archive_XXXXXXXXX	L2_Wed_AM_ANG KENG SIANG_Report_XXXXXXXXX
Aryl Ng Shen Le	L2_Wed_AM_Aryl Ng Shen Le_Archive_XXXXXXXXX	L2_Wed_AM_Aryl Ng Shen Le_Report_XXXXXXXXX
Au Yuan Xian	L2_Fri_AM_Au Yuan Xian_Archive_XXXXXXXXX	L2_Fri_AM_Au Yuan Xian_Report_XXXXXXXXX
Bai Xiaoru	L2_Fri_AM_Bai Xiaoru_Archive_XXXXXXXXX	L2_Fri_AM_Bai Xiaoru_Report_XXXXXXXXX
Bryan Yu Cheng You	L2_Fri_AM_Bryan Yu Cheng You_Archive_XXXXXXXXX	L2_Fri_AM_Bryan Yu Cheng You_Report_XXXXXXXXX
Chai Wei Lynthia	L2_Fri_AM_Chai Wei Lynthia_Archive_XXXXXXXXX	L2_Fri_AM_Chai Wei Lynthia_Report_XXXXXXXXX
Cheang Zhi Yi Jordan	L2_Fri_AM_Cheang Zhi Yi Jordan_Archive_XXXXXXXXX	L2_Fri_AM_Cheang Zhi Yi Jordan_Report_XXXXXXXXX
Chee Poh Hock	L2_Wed_AM_Chee Poh Hock_Archive_XXXXXXXXX	L2_Wed_AM_Chee Poh Hock_Report_XXXXXXXXX
Cheng Wei Qiao	L2_Fri_AM_Cheng Wei Qiao_Archive_XXXXXXXXX	L2_Fri_AM_Cheng Wei Qiao_Report_XXXXXXXXX
Cheung Po Rui Bryan	L2_Fri_AM_Cheung Po Rui Bryan_Archive_XXXXXXXXX	L2_Fri_AM_Cheung Po Rui Bryan_Report_XXXXXXXXX
CHONG LEE TENG VALENCIA	L2_Wed_AM_CHONG LEE TENG VALEN_Archive_XXXXXXXXX	L2_Wed_AM_CHONG LEE TENG VALEN_Report_XXXXXXXXX
Davian Chan Sze Peng	L2_Fri_AM_Davian Chan Sze Peng_Archive_XXXXXXXXX	L2_Fri_AM_Davian Chan Sze Peng_Report_XXXXXXXXX
David Michael Woodside	L2_Fri_AM_David Michael Woodsi_Archive_XXXXXXXXX	L2_Fri_AM_David Michael Woodsi_Report_XXXXXXXXX
ELJER CHUA	L2_Fri_AM_ELJER CHUA_Archive_XXXXXXXXX	L2_Fri_AM_ELJER CHUA_Report_XXXXXXXXX
FANG XINJIA	L2_Fri_AM_FANG XINJIA_Archive_XXXXXXXXX	L2_Fri_AM_FANG XINJIA_Report_XXXXXXXXX
Fidel Tan Yan Sheng	L2_Fri_AM_Fidel Tan Yan Sheng_Archive_XXXXXXXXX	L2_Fri_AM_Fidel Tan Yan Sheng_Report_XXXXXXXXX
Foo Fang Kiang	L2_Fri_AM_Foo Fang Kiang_Archive_XXXXXXXXX	L2_Fri_AM_Foo Fang Kiang_Report_XXXXXXXXX
Gao Zhixuan	L2_Fri_AM_Gao Zhixuan_Archive_XXXXXXXXX	L2_Fri_AM_Gao Zhixuan_Report_XXXXXXXXX
Giam Xiong Yao	L2_Fri_AM_Giam Xiong Yao_Archive_XXXXXXXXX	L2_Fri_AM_Giam Xiong Yao_Report_XXXXXXXXX
Gillian Ho Xin Ying	L2_Fri_AM_Gillian Ho Xin Ying_Archive_XXXXXXXXX	L2_Fri_AM_Gillian Ho Xin Ying_Report_XXXXXXXXX
Goh Jia Hong Edwin	L2_Wed_AM_Goh Jia Hong Edwin_Archive_XXXXXXXXX	L2_Wed_AM_Goh Jia Hong Edwin_Report_XXXXXXXXX
Guinne Teresa Sng Yu Lin	L2_Fri_AM_Guinne Teresa Sng Yu_Archive_XXXXXXXXX	L2_Fri_AM_Guinne Teresa Sng Yu_Report_XXXXXXXXX
Hariharan Hadrian S/O Subramaniam	L2_Fri_AM_Hariharan Hadrian S_Archive_XXXXXXXXX	L2_Fri_AM_Hariharan Hadrian S_Report_XXXXXXXXX
HO MING JUN	L2_Wed_AM_HO MING JUN_Archive_XXXXXXXXX	L2_Wed_AM_HO MING JUN_Report_XXXXXXXXX
Ho Yi Shu Keon	L2_Wed_AM_Ho Yi Shu Keon_Archive_XXXXXXXXX	L2_Wed_AM_Ho Yi Shu Keon_Report_XXXXXXXXX
Hou Yinjiayi	L2_Fri_AM_Hou Yinjiayi_Archive_XXXXXXXXX	L2_Fri_AM_Hou Yinjiayi_Report_XXXXXXXXX
Ian Isaiah Tan Jun Wei	L2_Fri_AM_Ian Isaiah Tan Jun W_Archive_XXXXXXXXX	L2_Fri_AM_Ian Isaiah Tan Jun W_Report_XXXXXXXXX
Jacob Zhang Zhiqiang	L2_Wed_AM_Jacob Zhang Zhiqiang_Archive_XXXXXXXXX	L2_Wed_AM_Jacob Zhang Zhiqiang_Report_XXXXXXXXX
JEROME TEO SZE YONG	L2_Wed_AM_JEROME TEO SZE YONG_Archive_XXXXXXXXX	L2_Wed_AM_JEROME TEO SZE YONG_Report_XXXXXXXXX
Jonathan Ang Xu Wen	L2_Wed_AM_Jonathan Ang Xu Wen_Archive_XXXXXXXXX	L2_Wed_AM_Jonathan Ang Xu Wen_Report_XXXXXXXXX
JONATHAN KHOO TENG YANG	L2_Fri_AM_JONATHAN KHOO TENG Y_Archive_XXXXXXXXX	L2_Fri_AM_JONATHAN KHOO TENG Y_Report_XXXXXXXXX
Kabeta Takuma	L2_Wed_AM_Kabeta Takuma_Archive_XXXXXXXXX	L2_Wed_AM_Kabeta Takuma_Report_XXXXXXXXX
Khoo Wu Jian Samuel	L2_Wed_AM_Khoo Wu Jian Samuel_Archive_XXXXXXXXX	L2_Wed_AM_Khoo Wu Jian Samuel_Report_XXXXXXXXX
KIM JOOHWAN	L2_Fri_AM_KIM JOOHWAN_Archive_XXXXXXXXX	L2_Fri_AM_KIM JOOHWAN_Report_XXXXXXXXX
Lau Wai Kit	L2_Wed_AM_Lau Wai Kit_Archive_XXXXXXXXX	L2_Wed_AM_Lau Wai Kit_Report_XXXXXXXXX
LEE KE HUI	L2_Wed_AM_LEE KE HUI_Archive_XXXXXXXXX	L2_Wed_AM_LEE KE HUI_Report_XXXXXXXXX
Lee Shao Yu	L2_Wed_AM_Lee Shao Yu_Archive_XXXXXXXXX	L2_Wed_AM_Lee Shao Yu_Report_XXXXXXXXX
Lek Ju Ying	L2_Wed_AM_Lek Ju Ying_Archive_XXXXXXXXX	L2_Wed_AM_Lek Ju Ying_Report_XXXXXXXXX
Leong Ka Weng, Rachelle	L2_Fri_AM_Leong Ka Weng Rache_Archive_XXXXXXXXX	L2_Fri_AM_Leong Ka Weng Rache_Report_XXXXXXXXX
LEW POH CHEN, DOUGLAS	L2_Fri_AM_LEW POH CHEN DOUGLA_Archive_XXXXXXXXX	L2_Fri_AM_LEW POH CHEN DOUGLA_Report_XXXXXXXXX
Long Deng Jie	L2_Wed_AM_Long Deng Jie_Archive_XXXXXXXXX	L2_Wed_AM_Long Deng Jie_Report_XXXXXXXXX
Markus Lim Yi Qin	L2_Wed_AM_Markus Lim Yi Qin_Archive_XXXXXXXXX	L2_Wed_AM_Markus Lim Yi Qin_Report_XXXXXXXXX
Mohamad Adam Bin Mohamad Yazid	L2_Wed_AM_Mohamad Adam Bin Moh_Archive_XXXXXXXXX	L2_Wed_AM_Mohamad Adam Bin Moh_Report_XXXXXXXXX
Muhammad Irfan Bin Zakaria	L2_Wed_AM_Muhammad Irfan Bin Z_Archive_XXXXXXXXX	L2_Wed_AM_Muhammad Irfan Bin Z_Report_XXXXXXXXX
Myat Thwe Naing	L2_Wed_AM_Myat Thwe Naing_Archive_XXXXXXXXX	L2_Wed_AM_Myat Thwe Naing_Report_XXXXXXXXX
Ng Etek	L2_Wed_AM_Ng Etek_Archive_XXXXXXXXX	L2_Wed_AM_Ng Etek_Report_XXXXXXXXX
Noorhakim Bin Jasman	L2_Wed_AM_Noorhakim Bin Jasman_Archive_XXXXXXXXX	L2_Wed_AM_Noorhakim Bin Jasman_Report_XXXXXXXXX
NUR SYADIYAH BTE LUTFI	L2_Fri_AM_NUR SYADIYAH BTE LUT_Archive_XXXXXXXXX	L2_Fri_AM_NUR SYADIYAH BTE LUT_Report_XXXXXXXXX
ONG WEI SHENG	L2_Fri_AM_ONG WEI SHENG_Archive_XXXXXXXXX	L2_Fri_AM_ONG WEI SHENG_Report_XXXXXXXXX
PANG JUN WEN, ADRIC	L2_Fri_AM_PANG JUN WEN ADRIC_Archive_XXXXXXXXX	L2_Fri_AM_PANG JUN WEN ADRIC_Report_XXXXXXXXX
PUN ZE YONG	L2_Wed_AM_PUN ZE YONG_Archive_XXXXXXXXX	L2_Wed_AM_PUN ZE YONG_Report_XXXXXXXXX
Qiang Zhuang	L2_Wed_AM_Qiang Zhuang_Archive_XXXXXXXXX	L2_Wed_AM_Qiang Zhuang_Report_XXXXXXXXX
QIU YI WEN	L2_Fri_AM_QIU YI WEN_Archive_XXXXXXXXX	L2_Fri_AM_QIU YI WEN_Report_XXXXXXXXX
R M RAAJAMANI	L2_Fri_AM_R M RAAJAMANI_Archive_XXXXXXXXX	L2_Fri_AM_R M RAAJAMANI_Report_XXXXXXXXX
RIZAVUR RAHMAN FASLUR RAHMAN	L2_Fri_AM_RIZAVUR RAHMAN FASLU_Archive_XXXXXXXXX	L2_Fri_AM_RIZAVUR RAHMAN FASLU_Report_XXXXXXXXX
Ryan Tan Jun Hao	L2_Wed_AM_Ryan Tan Jun Hao_Archive_XXXXXXXXX	L2_Wed_AM_Ryan Tan Jun Hao_Report_XXXXXXXXX
Saw Wee Kiat	L2_Wed_AM_Saw Wee Kiat_Archive_XXXXXXXXX	L2_Wed_AM_Saw Wee Kiat_Report_XXXXXXXXX
SIM BOWEN	L2_Fri_AM_SIM BOWEN_Archive_XXXXXXXXX	L2_Fri_AM_SIM BOWEN_Report_XXXXXXXXX
TAM LI NA	L2_Fri_AM_TAM LI NA_Archive_XXXXXXXXX	L2_Fri_AM_TAM LI NA_Report_XXXXXXXXX
Tan Javen	L2_Wed_AM_Tan Javen_Archive_XXXXXXXXX	L2_Wed_AM_Tan Javen_Report_XXXXXXXXX
Tan Kai Hao Andrew	L2_Wed_AM_Tan Kai Hao Andrew_Archive_XXXXXXXXX	L2_Wed_AM_Tan Kai Hao Andrew_Report_XXXXXXXXX
Tan Suet Ying	L2_Fri_AM_Tan Suet Ying_Archive_XXXXXXXXX	L2_Fri_AM_Tan Suet Ying_Report_XXXXXXXXX
Tan Yeung Ming Sean Eugene	L2_Wed_AM_Tan Yeung Ming Sean _Archive_XXXXXXXXX	L2_Wed_AM_Tan Yeung Ming Sean _Report_XXXXXXXXX
Teoh Yi Zheng	L2_Fri_AM_Teoh Yi Zheng_Archive_XXXXXXXXX	L2_Fri_AM_Teoh Yi Zheng_Report_XXXXXXXXX
Tey Zi Le	L2_Wed_AM_Tey Zi Le_Archive_XXXXXXXXX	L2_Wed_AM_Tey Zi Le_Report_XXXXXXXXX
Thet Ke Min, Sonia	L2_Wed_AM_Thet Ke Min Sonia_Archive_XXXXXXXXX	L2_Wed_AM_Thet Ke Min Sonia_Report_XXXXXXXXX
Trina Wern Qin Rong	L2_Fri_AM_Trina Wern Qin Rong_Archive_XXXXXXXXX	L2_Fri_AM_Trina Wern Qin Rong_Report_XXXXXXXXX
Venessa Chee Li Lin	L2_Fri_AM_Venessa Chee Li Lin_Archive_XXXXXXXXX	L2_Fri_AM_Venessa Chee Li Lin_Report_XXXXXXXXX
WANG HUA CHEN	L2_Fri_AM_WANG HUA CHEN_Archive_XXXXXXXXX	L2_Fri_AM_WANG HUA CHEN_Report_XXXXXXXXX
Wee Cheng Yuan Andrew	L2_Wed_AM_Wee Cheng Yuan Andre_Archive_XXXXXXXXX	L2_Wed_AM_Wee Cheng Yuan Andre_Report_XXXXXXXXX
Wee Xin Ze	L2_Wed_AM_Wee Xin Ze_Archive_XXXXXXXXX	L2_Wed_AM_Wee Xin Ze_Report_XXXXXXXXX
Yeo Zhong Kang Dennis	L2_Wed_AM_Yeo Zhong Kang Denni_Archive_XXXXXXXXX	L2_Wed_AM_Yeo Zhong Kang Denni_Report_XXXXXXXXX
ZHONG SHUHAO	L2_Fri_AM_ZHONG SHUHAO_Archive_XXXXXXXXX	L2_Fri_AM_ZHONG SHUHAO_Report_XXXXXXXXX