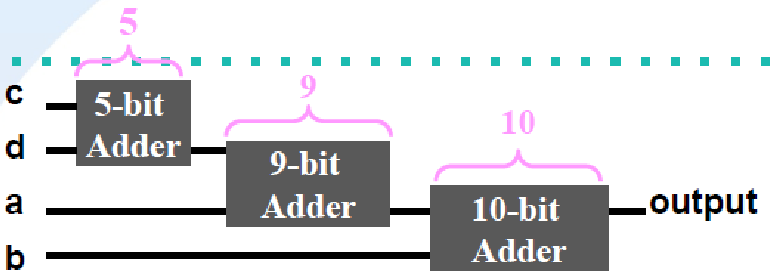
2019 DCS Lab 3

Pipeline

- ✓ a [7:0] , b [7:0] , c [3:0] , d [3:0]
- ✓ Q: (a + b + c + d) x 4 iterations ?

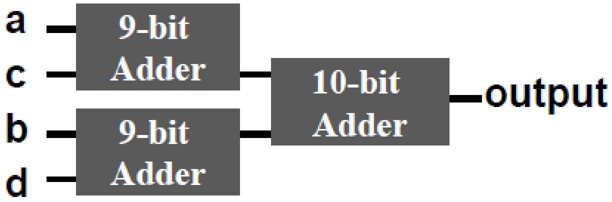
Basic



Area: 24 units

Time: 24 x 4 = 96 units

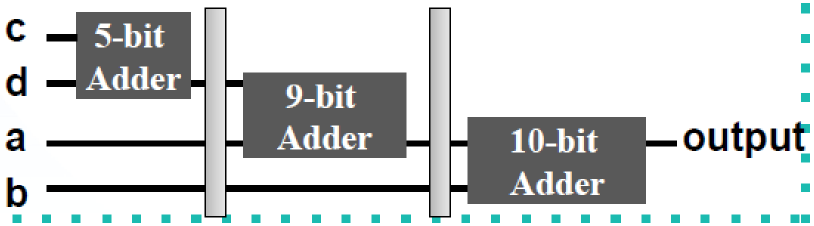
Parallel



Area: 28 units

Time: 19 x 4 = 76 units

Pipeline



Area: 24+ units

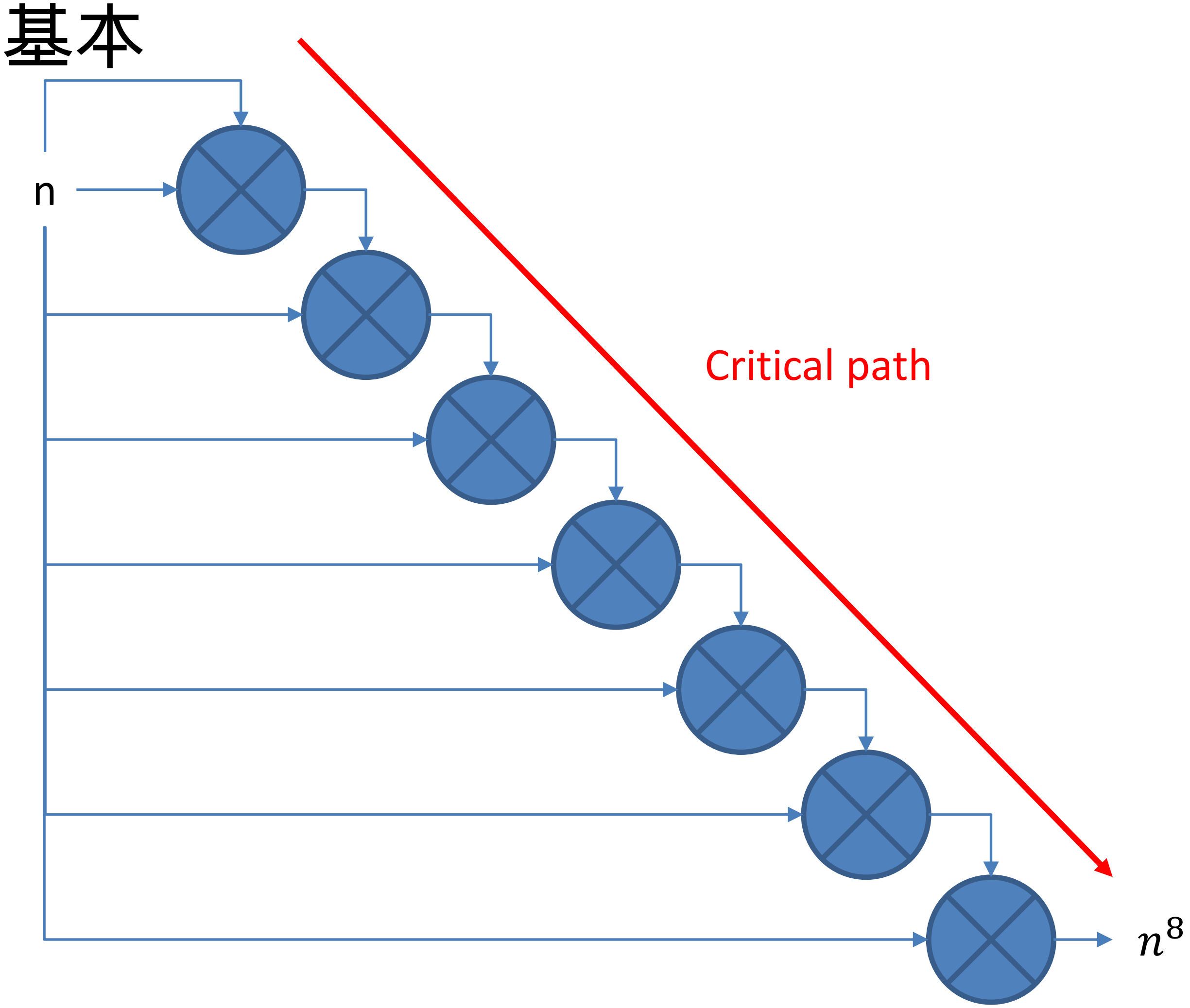
Time: 60 units

Pipeline

- 優點
  - Cycle Time 可以降低，可以增加frequency
  - 增加硬體Utilization與throughput
- 缺點
  - 可能會有hazard
  - 設計困難

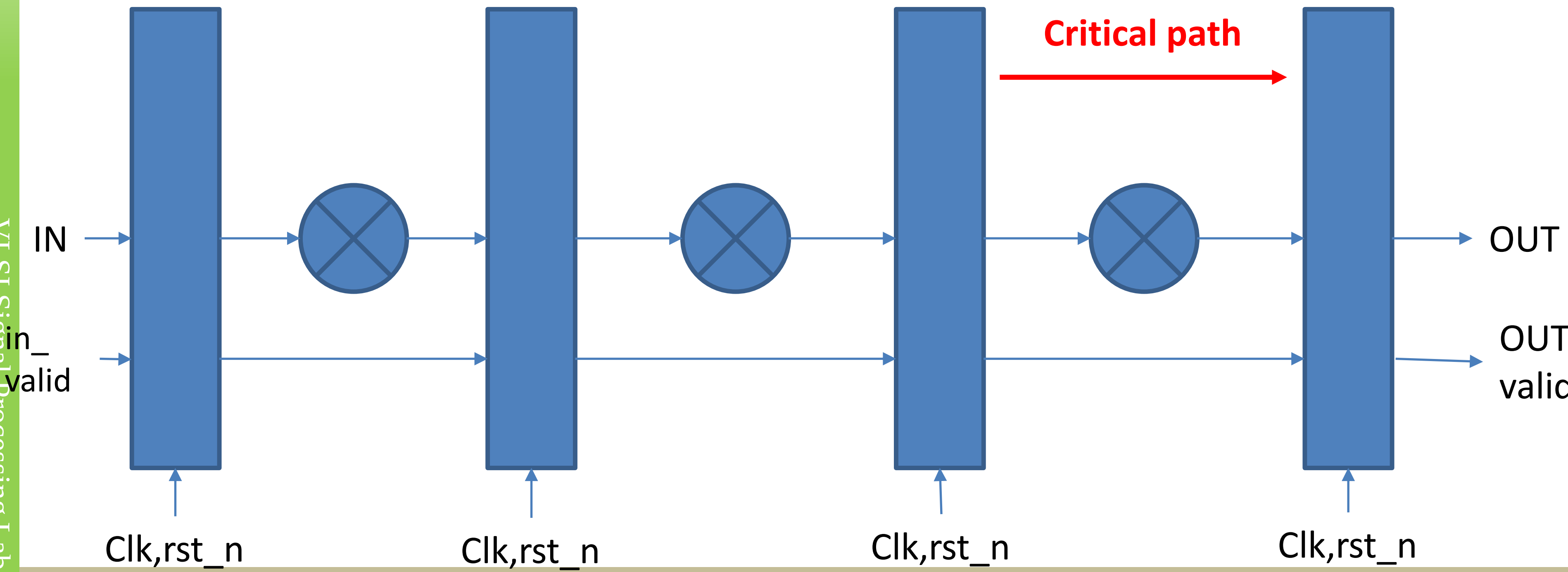
n^8

- 由多個乘法器組成一個n^8
- 基本



Pipeline n^8

- 嘗試去Pipeline這些乘法器
- Hint: 三個stage就可以
- Block diagram如下



Spec

- Pipeline是利用Sequential circuit具有clock跟儲存功能的D Flip-Flop(DFF)。
- Input後跟output前面都必須擋DFF。
- Asynchronous reset.
- Input 會給值，在in_valud = 1時每個clock negative edge。
- Output會檢查，在out_valid = 1時每個clock postive edge。
- 禁止用窮舉方式。
- 02_SYN合成電路，cycle time = 5ns.

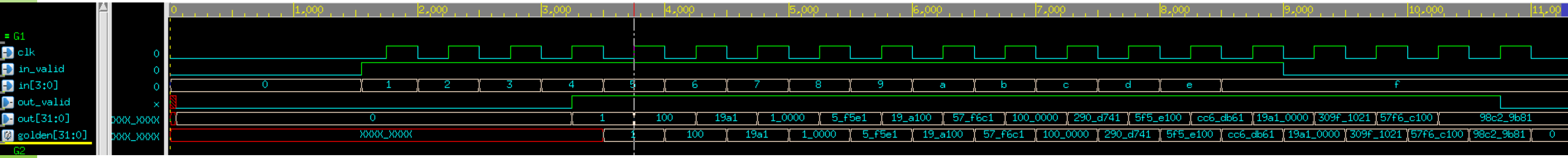
pipeline.sv

Input Signal	Bit Width	Definition
clk	1	5 ns Clock
rst_n	1	非同步reset 當reset negedge時 out及out_valid需為0
in	4	in = 1 ~ 15 且不會為0
in_valid	1	in_valid 拉起為1 時in進入，in最後一筆進入，in_valid降下為0。

Output Signal	Bit Width	Definition
out	32	$out = in^8$
out_valid	1	請參考block digram設計，當out_valid = 1時每個posedge會依”in順序”檢查output。

Output & Waveform

- Waveform


0
1,000
2,000
3,000
4,000
5,000
6,000
7,000
8,000
9,000
10,000
11,00
= G1
clk 0
in_valid 0
in[3:0] 0
out_valid x
out[31:0] XXX_XXXX
golden[31:0] XXX_XXXX
G2
0 1 2 3 4 5 6 7 8 9 a b c d e f
0 1 100 19a1 1_0000 5_f5e1 19_a100 57_f6c1 100_0000 290_d741 5f5_e100 cc6_db61 19a1_0000 309f_1021 57f6_c100 98c2_9b81
XXXX_XXXX 1 100 19a1 1_0000 5_f5e1 19_a100 57_f6c1 100_0000 290_d741 5f5_e100 cc6_db61 19a1_0000 309f_1021 57f6_c100 98c2_9b81 0

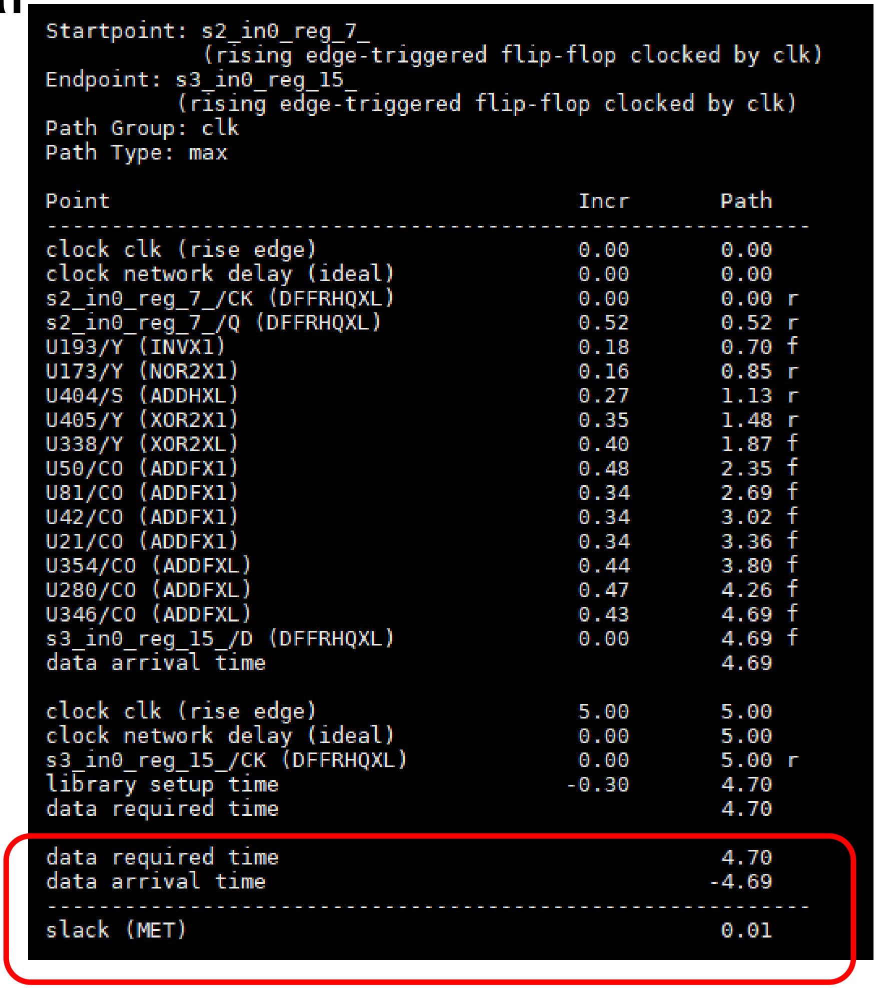
Command

- tar -xvf ~dcsta01/Lab03.tar
- Need 02_SYN/01_run_dc

Synthesis

without any latch, error

timing met



```
Startpoint: s2_in0_reg_7_
            (rising edge-triggered flip-flop clocked by clk)
Endpoint: s3_in0_reg_15_
          (rising edge-triggered flip-flop clocked by clk)
Path Group: clk
Path Type: max

Point                                    Incr       Path
-----------------------------------------------------------
clock clk (rise edge)                    0.00       0.00
clock network delay (ideal)              0.00       0.00
s2_in0_reg_7_/CK (DFFRHQXL)              0.00       0.00 r
s2_in0_reg_7_/Q (DFFRHQXL)               0.52       0.52 r
U193/Y (INVX1)                           0.18       0.70 f
U173/Y (NOR2X1)                          0.16       0.85 r
U404/S (ADDHXL)                          0.27       1.13 r
U405/Y (XOR2X1)                          0.35       1.48 r
U338/Y (XOR2XL)                          0.40       1.87 f
U50/CO (ADDFX1)                          0.48       2.35 f
U81/CO (ADDFX1)                          0.34       2.69 f
U42/CO (ADDFX1)                          0.34       3.02 f
U21/CO (ADDFX1)                          0.34       3.36 f
U354/CO (ADDFXL)                         0.44       3.80 f
U280/CO (ADDFXL)                         0.47       4.26 f
U346/CO (ADDFXL)                         0.43       4.69 f
s3_in0_reg_15_/D (DFFRHQXL)              0.00       4.69 f
data arrival time                                   4.69

clock clk (rise edge)                    5.00       5.00
clock network delay (ideal)              0.00       5.00
s3_in0_reg_15_/CK (DFFRHQXL)             0.00       5.00 r
library setup time                      -0.30       4.70
data required time                                  4.70

data required time                                  4.70
data arrival time                                  -4.69
-----------------------------------------------------------
slack (MET)                                         0.01
```

Appendix

Digital Circuits and Systems(DEE3342)
How to Upload Your Design?

Name: Jesse Chen
Advisor: Tian-Sheuan Chang

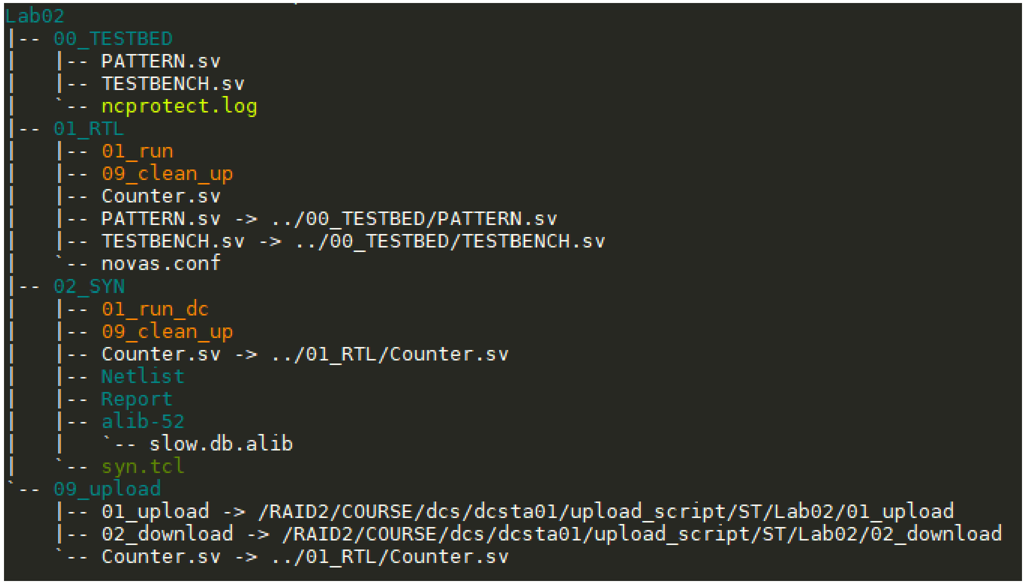
2019.03.20(revised)

Problems

- The servers in ED415 can not connect to external network for security issues
- It takes too long for students to wait for demo
- We will give you a script to upload your file to us

Directory Tree Structure

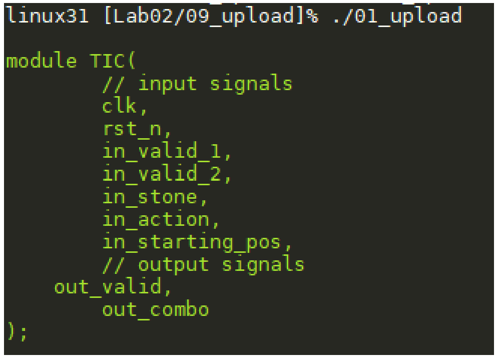
- The following directory of lab and homework will be like this:
  - Add 09_upload with 01_upload and 02_download



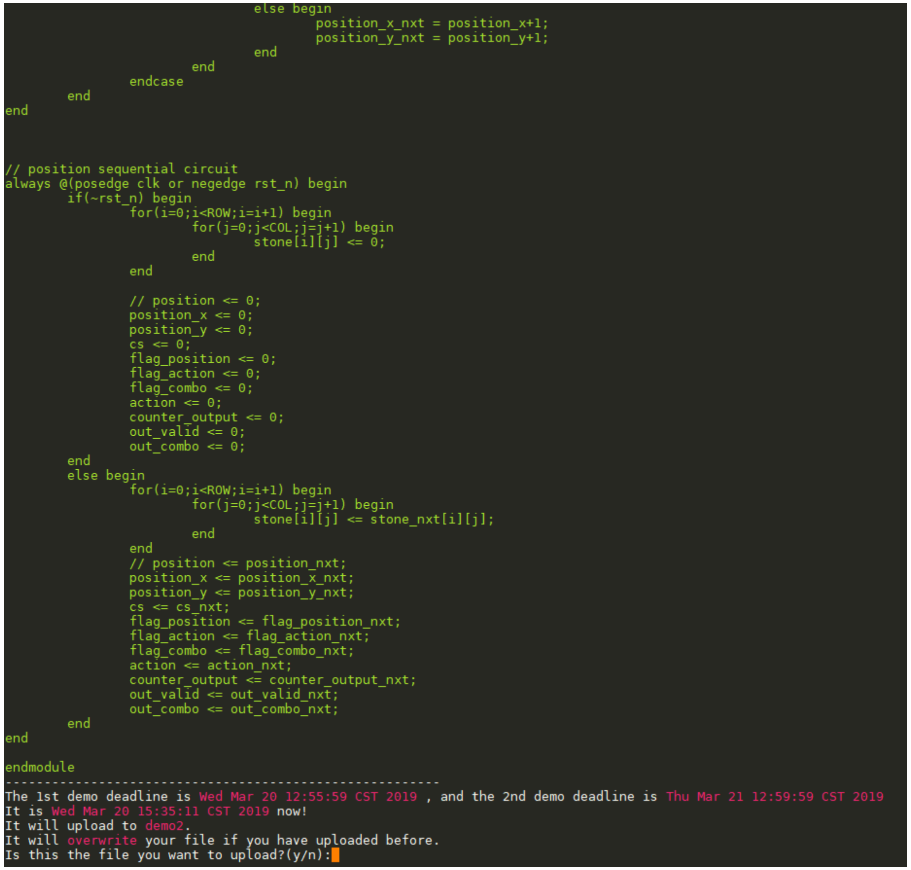
```
Lab02
|-- 00_TESTBED
|   |-- PATTERN.sv
|   |-- TESTBENCH.sv
|   `-- ncprotect.log
|-- 01_RTL
|   |-- 01_run
|   |-- 09_clean_up
|   |-- Counter.sv
|   |-- PATTERN.sv -> ../00_TESTBED/PATTERN.sv
|   |-- TESTBENCH.sv -> ../00_TESTBED/TESTBENCH.sv
|   `-- novas.conf
|-- 02_SYN
|   |-- 01_run_dc
|   |-- 09_clean_up
|   |-- Counter.sv -> ../01_RTL/Counter.sv
|   |-- Netlist
|   |-- Report
|   |-- alib-52
|   |   `-- slow.db.alib
|   `-- syn.tcl
`-- 09_upload
    |-- 01_upload -> /RAID2/COURSE/dcs/dcsta01/upload_script/ST/Lab02/01_upload
    |-- 02_download -> /RAID2/COURSE/dcs/dcsta01/upload_script/ST/Lab02/02_download
    `-- Counter.sv -> ../01_RTL/Counter.sv
```

Upload Steps

- $ ./01_upload
- Display the file first in green color



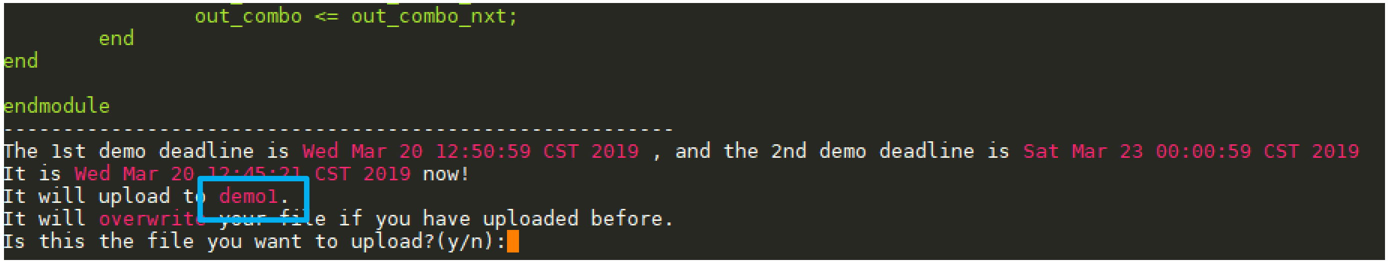
```
linux31 [Lab02/09_upload]% ./01_upload

module TIC(
        // input signals
        clk,
        rst_n,
        in_valid_1,
        in_valid_2,
        in_stone,
        in_action,
        in_starting_pos,
        // output signals
    out_valid,
        out_combo
);
```



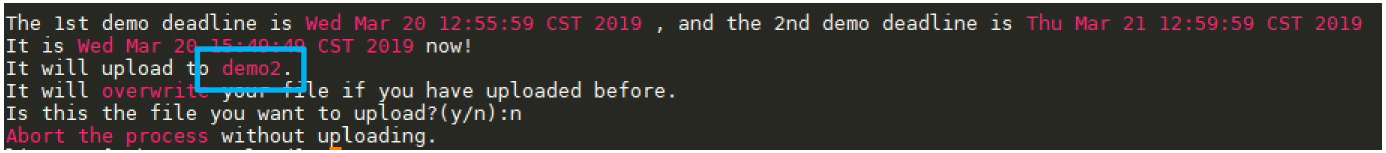
```
                                else begin
                                        position_x_nxt = position_x+1;
                                        position_y_nxt = position_y+1;
                                end
                        end
                endcase
        end
end


// position sequential circuit
always @(posedge clk or negedge rst_n) begin
        if(~rst_n) begin
                for(i=0;i<ROW;i=i+1) begin
                        for(j=0;j<COL;j=j+1) begin
                                stone[i][j] <= 0;
                        end
                end

                // position <= 0;
                position_x <= 0;
                position_y <= 0;
                cs <= 0;
                flag_position <= 0;
                flag_action <= 0;
                flag_combo <= 0;
                action <= 0;
                counter_output <= 0;
                out_valid <= 0;
                out_combo <= 0;
        end
        else begin
                for(i=0;i<ROW;i=i+1) begin
                        for(j=0;j<COL;j=j+1) begin
                                stone[i][j] <= stone_nxt[i][j];
                        end
                end
                // position <= position_nxt;
                position_x <= position_x_nxt;
                position_y <= position_y_nxt;
                cs <= cs_nxt;
                flag_position <= flag_position_nxt;
                flag_action <= flag_action_nxt;
                flag_combo <= flag_combo_nxt;
                action <= action_nxt;
                counter_output <= counter_output_nxt;
                out_valid <= out_valid_nxt;
                out_combo <= out_combo_nxt;
        end
end

endmodule
-------------------------------------------------------
The 1st demo deadline is Wed Mar 20 12:55:59 CST 2019 , and the 2nd demo deadline is Thu Mar 21 12:59:59 CST 2019
It is Wed Mar 20 15:35:11 CST 2019 now!
It will upload to demo2.
It will overwrite your file if you have uploaded before.
Is this the file you want to upload?(y/n):
```

Upload Steps

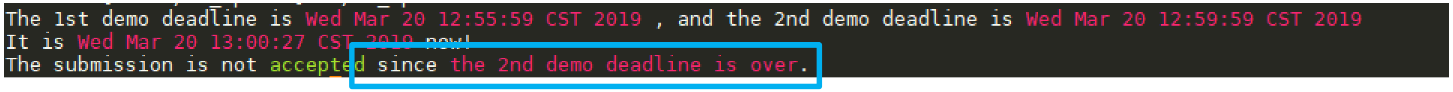
- Display the deadline for the 1st demo and the 2nd demo
  - Get the current time and decide whether it is for 1st demo or the 2nd demo



```
                out_combo <= out_combo_nxt;
        end
end

endmodule
--------------------------------------------------------
The 1st demo deadline is Wed Mar 20 12:50:59 CST 2019 , and the 2nd demo deadline is Sat Mar 23 00:00:59 CST 2019
It is Wed Mar 20 12:45:21 CST 2019 now!
It will upload to demo1.
It will overwrite your file if you have uploaded before.
Is this the file you want to upload?(y/n):
```



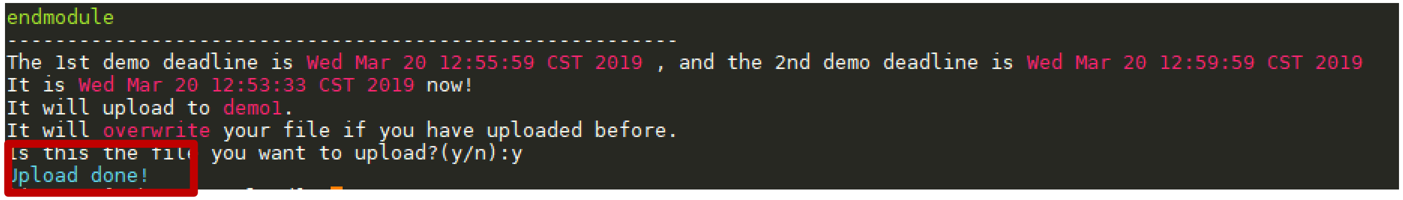
```
The 1st demo deadline is Wed Mar 20 12:55:59 CST 2019 , and the 2nd demo deadline is Thu Mar 21 12:59:59 CST 2019
It is Wed Mar 20 15:49:49 CST 2019 now!
It will upload to demo2.
It will overwrite your file if you have uploaded before.
Is this the file you want to upload?(y/n):n
Abort the process without uploading.
```



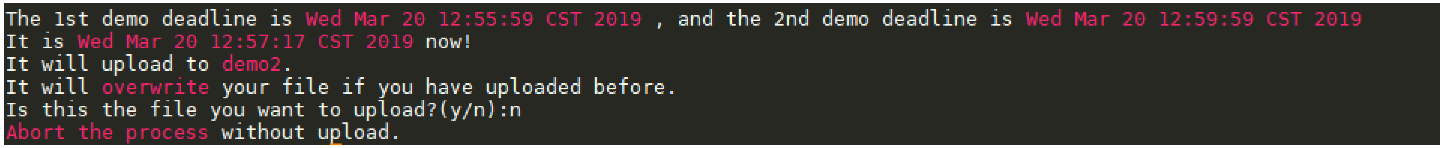
```
The 1st demo deadline is Wed Mar 20 12:55:59 CST 2019 , and the 2nd demo deadline is Wed Mar 20 12:59:59 CST 2019
It is Wed Mar 20 13:00:27 CST 2019 now!
The submission is not accepted since the 2nd demo deadline is over.
```

Upload Steps

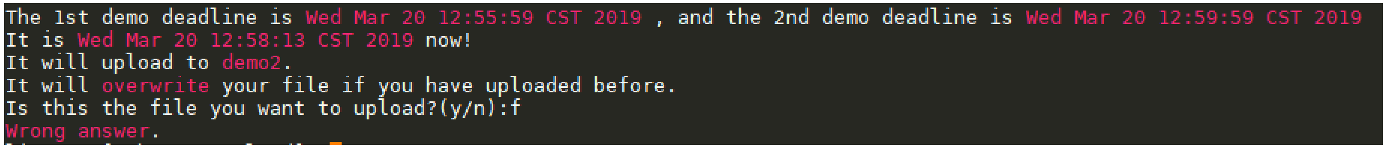
- It will overwrite the file you uploaded before
  - Input y or n



```
endmodule
------------------------------------------------------------
The 1st demo deadline is Wed Mar 20 12:55:59 CST 2019 , and the 2nd demo deadline is Wed Mar 20 12:59:59 CST 2019
It is Wed Mar 20 12:53:33 CST 2019 now!
It will upload to demo1.
It will overwrite your file if you have uploaded before.
Is this the file you want to upload?(y/n):y
Upload done!
```



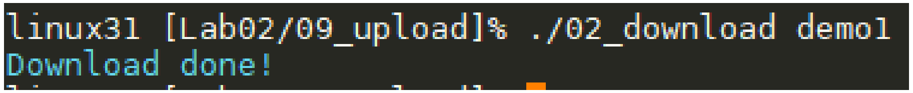
```
The 1st demo deadline is Wed Mar 20 12:55:59 CST 2019 , and the 2nd demo deadline is Wed Mar 20 12:59:59 CST 2019
It is Wed Mar 20 12:57:17 CST 2019 now!
It will upload to demo2.
It will overwrite your file if you have uploaded before.
Is this the file you want to upload?(y/n):n
Abort the process without upload.
```



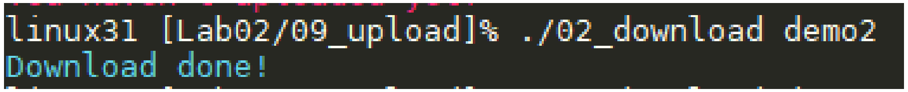
```
The 1st demo deadline is Wed Mar 20 12:55:59 CST 2019 , and the 2nd demo deadline is Wed Mar 20 12:59:59 CST 2019
It is Wed Mar 20 12:58:13 CST 2019 now!
It will upload to demo2.
It will overwrite your file if you have uploaded before.
Is this the file you want to upload?(y/n):f
Wrong answer.
```

Download Steps

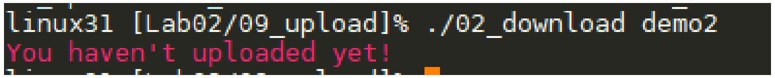
- You can check your file after uploading
- $ ./02_download [Argument]
  - $ ./02_download demo1
  - $ ./02_download demo2



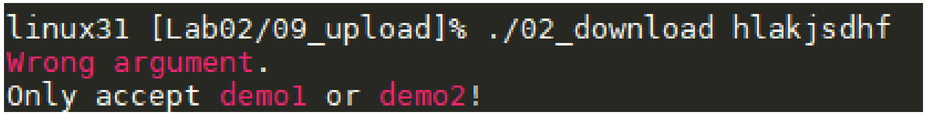
```
linux31 [Lab02/09_upload]% ./02_download demo1
Download done!
```



```
linux31 [Lab02/09_upload]% ./02_download demo2
Download done!
```

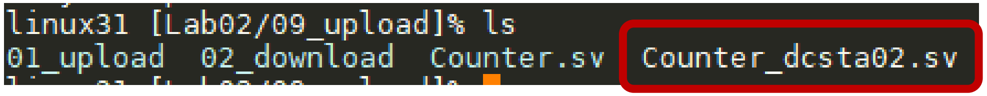


```
linux31 [Lab02/09_upload]% ./02_download demo2
You haven't uploaded yet!
```



```
linux31 [Lab02/09_upload]% ./02_download hlakjsdhf
Wrong argument.
Only accept demo1 or demo2!
```

- Download to 09_upload



```
linux31 [Lab02/09_upload]% ls
01_upload  02_download  Counter.sv  Counter_dcsta02.sv
```