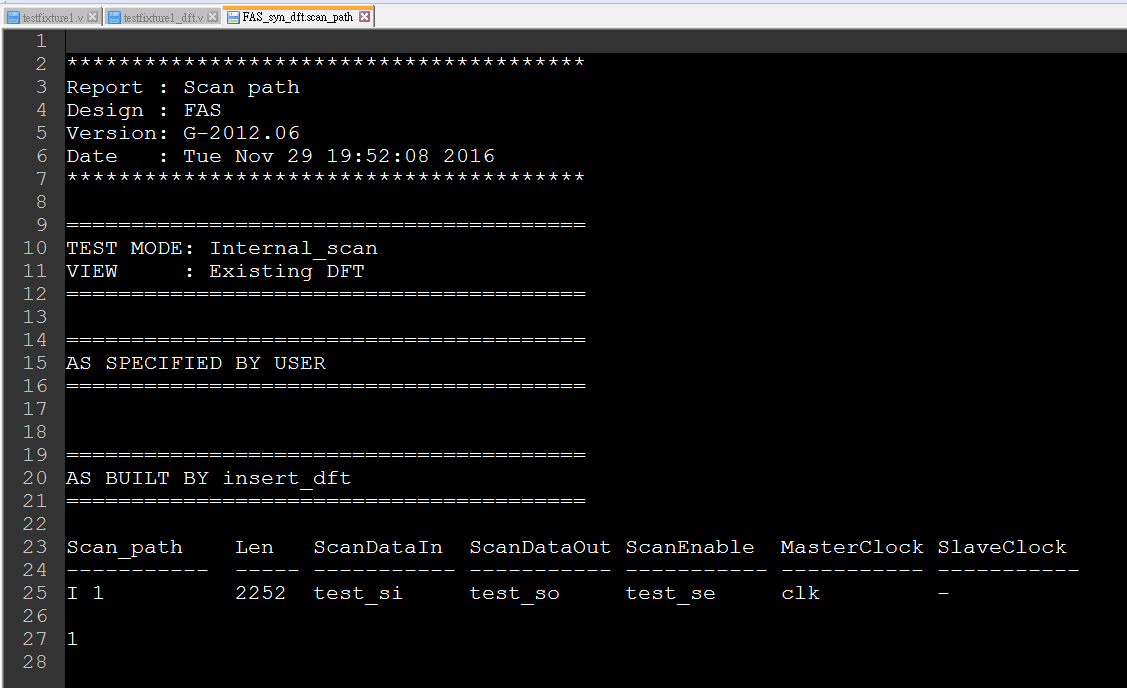


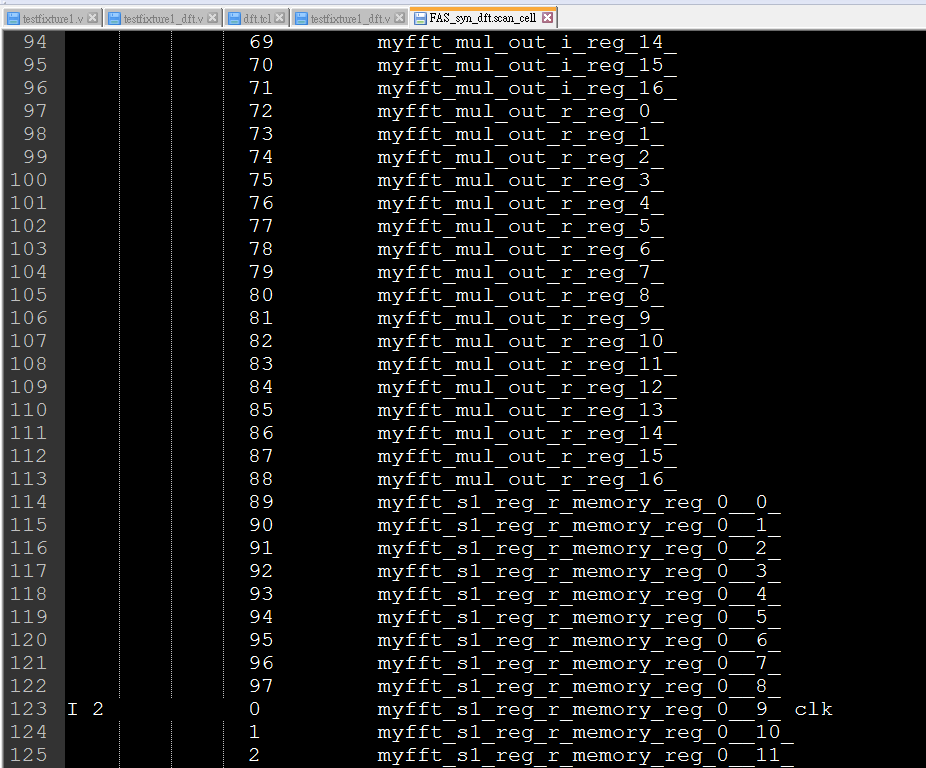
A-1. How many flip-flops are chained? 2252-1=2251



A-2. there are 23 scan chain, each is 97 /98 cell

1-20=>98

21-23=>97



A-3.

FAS syn dft.v 
CLK8UFX6 131343 ( .A(n332), 
. Y(n323) ) ; 
BUFX4 131349 ( .A(n332), 
. Y(n319) ) ; 
BUFX16 131353 ( .A(n326), 
. Y(n332) ) ; 
MX12X4 131354 ( .A(n456), 
. 801455) , 
. SO(n315) , 
. Y(n15) ) ; 
MX12X4 131356 ( .A(n462), 
. 801461) , 
. SO(n316) , 
. Y(n22) ) ; 
NAND2X4 131358 ( .A(n316), 
.B(freq[2]), 
. Y(n469) ) ; 
INVX6 131443 ( .A(n239), 
. Y(DP op 1633 124 4490 11270) 
NAND2X4 131444 ( .A(n198), 
.B(done BAR), .Y(n165) 
INVXL 131445 ( 
. Y(n462) ) ; 
INVX20 131446 ( . A(in datal 10]), 
. Y(DP 
INVXL 131447 ( 
.Y(n416) ); 
INVXL 131448 ( 
. Y(n422) ) ; 
NAND2X1 131449 ( .A(DP op 16J3 124 4490 n271), 
. 150) , 
DP op 1633 124 4490 1117) 
NAND2X6 131450 ( .A(n411), 
. Y(n292) ) ; 
INVXL 131451 ( 
. Y(n412) ) ; 
BUFX12 131452 ( .A(n208), 
.Y(n217) ) ; 
NAND2X6 131453 ( .A(DP op 16J3 124 4490 n546), 
. Y(DP 
INVX6 131454 
INVXL 131455 
INVXL 131456 
end•odute 
16J3 124 4490 (1190) 
.A(abs[30]), 
. Y(n414) ) ; 
. Y(n458) ) ; 
. A(n83) , 
. Y(n408) ) ; 
•odute FAS ( data _ valid, data, c 1k, rst, fir _ d, 
.B(DP , 
fir valid, f ft valid, done, 
freq, fft_dl, fft_d2, fft_d3, 
f ft d9, f ft dlO, f ft dll, f ft 
test 
test 
test 
test 
test 
test 
test 
test 
sil, 
s04, 
S18, 
sill, 
si14, 
sill, 
si20, 
si23, 
test sol, test si2, 
test si5, test s05, 
test s08, test si9, 
f ft d4, f ft d5, f ft d6, f ft d7, f ft d8, 
d12, f ft d13, f ft d14, f ft d15, f ft do, 
test s02, test si3, test s03, test si4, 
test si6, test s06, test si7, test s07, 
test s09, test silo, test solo, 
test 
test 
test 
test 
test 
sol 1, 
s014, 
5017, 
5020, 
5023, 
test 
test 
test 
test 
test 
si12, 
si15, 
si18, 
si21, 
se ) , 
test 
test 
test 
test 
so 12, 
5015, 
5018, 
5021, 
test 
test 
test 
test 
si13, 
si16, 
si19, 
si22, 
test 
test 
test 
test 
so 13, 
5016, 
5019, 
5022, 
input 
[15.0] data; 

module FAS ( data\_valid, data, clk, rst, fir\_d, fir\_valid, fft\_valid, done,

freq, fft\_d1, fft\_d2, fft\_d3, fft\_d4, fft\_d5, fft\_d6, fft\_d7, fft\_d8,

fft\_d9, fft\_d10, fft\_d11, fft\_d12, fft\_d13, fft\_d14, fft\_d15, fft\_d0,

test\_si1, test\_so1, test\_si2, test\_so2, test\_si3, test\_so3, test\_si4,

test\_so4, test\_si5, test\_so5, test\_si6, test\_so6, test\_si7, test\_so7,

test\_si8, test\_so8, test\_si9, test\_so9, test\_si10, test\_so10,

test\_si11, test\_so11, test\_si12, test\_so12, test\_si13, test\_so13,

test\_si14, test\_so14, test\_si15, test\_so15, test\_si16, test\_so16,

test\_si17, test\_so17, test\_si18, test\_so18, test\_si19, test\_so19,

test\_si20, test\_so20, test\_si21, test\_so21, test\_si22, test\_so22,

test\_si23, test\_so23, test\_se );

input [15:0] data;

output [15:0] fir\_d;

output [3:0] freq;

output [31:0] fft\_d1;

output [31:0] fft\_d2;

output [31:0] fft\_d3;

output [31:0] fft\_d4;

output [31:0] fft\_d5;

output [31:0] fft\_d6;

output [31:0] fft\_d7;

output [31:0] fft\_d8;

output [31:0] fft\_d9;

output [31:0] fft\_d10;

output [31:0] fft\_d11;

output [31:0] fft\_d12;

output [31:0] fft\_d13;

output [31:0] fft\_d14;

output [31:0] fft\_d15;

output [31:0] fft\_d0;

input data\_valid, clk, rst, test\_si1, test\_si2, test\_si3, test\_si4, test\_si5,

test\_si6, test\_si7, test\_si8, test\_si9, test\_si10, test\_si11,

test\_si12, test\_si13, test\_si14, test\_si15, test\_si16, test\_si17,

test\_si18, test\_si19, test\_si20, test\_si21, test\_si22, test\_si23,

test\_se;

output fir\_valid, fft\_valid, done, test\_so1, test\_so2, test\_so3, test\_so4,

test\_so5, test\_so6, test\_so7, test\_so8, test\_so9, test\_so10,

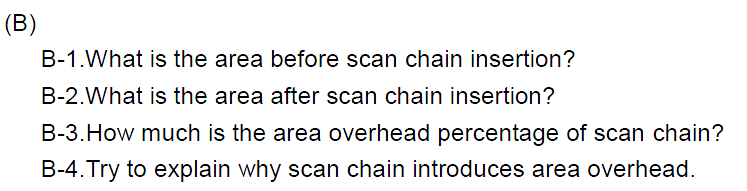
test\_so11, test\_so12, test\_so13, test\_so14, test\_so15, test\_so16,

test\_so17, test\_so18, test\_so19, test\_so20, test\_so21, test\_so22,

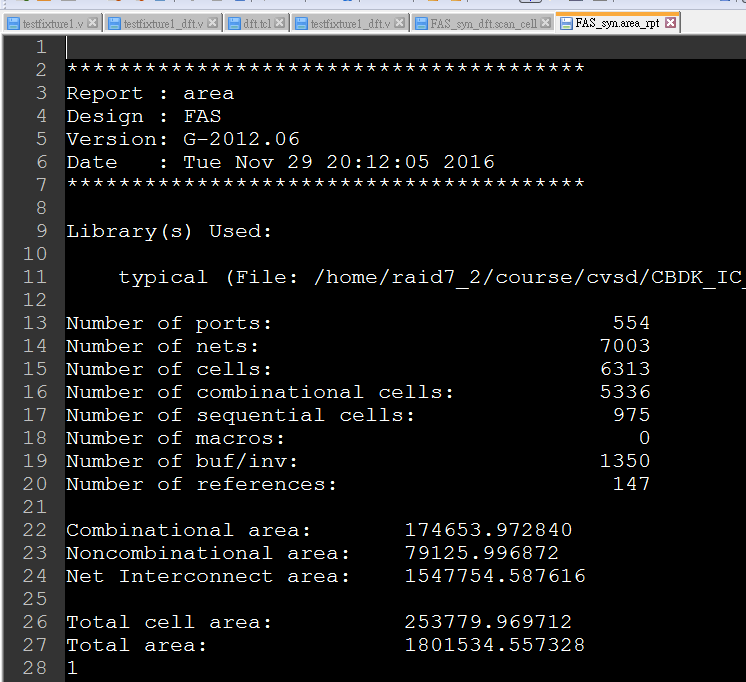
test\_so23;

A-4.

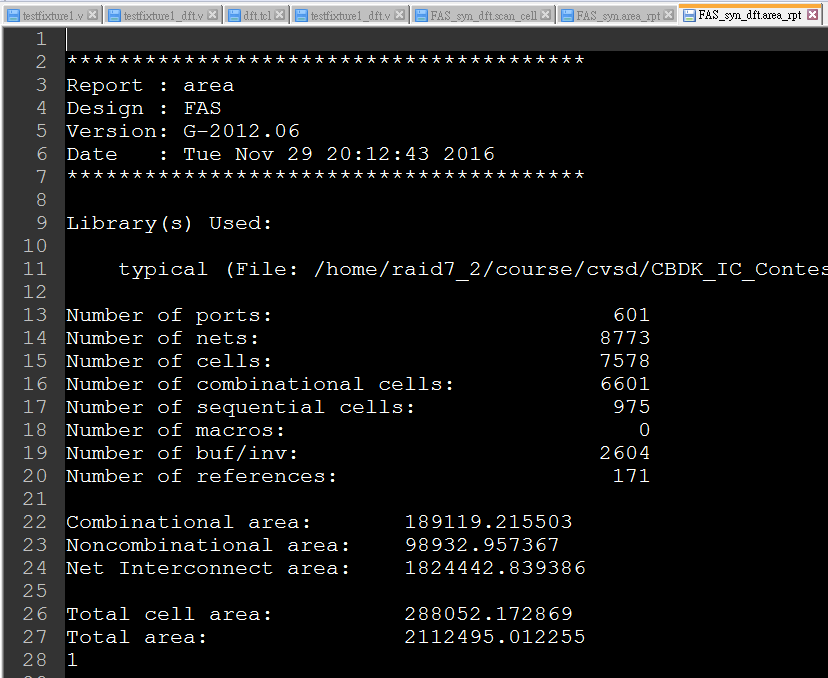
test\_se



B-1. 253779 um^2



B-2. 288052

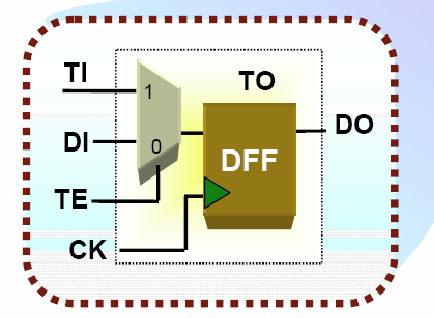


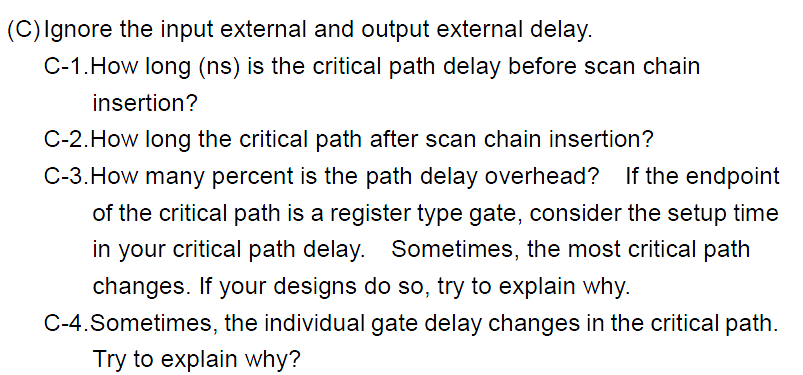
B-3.

288052/253779=1.135 => (after-before)/before= 13.5% overhead

B-4.

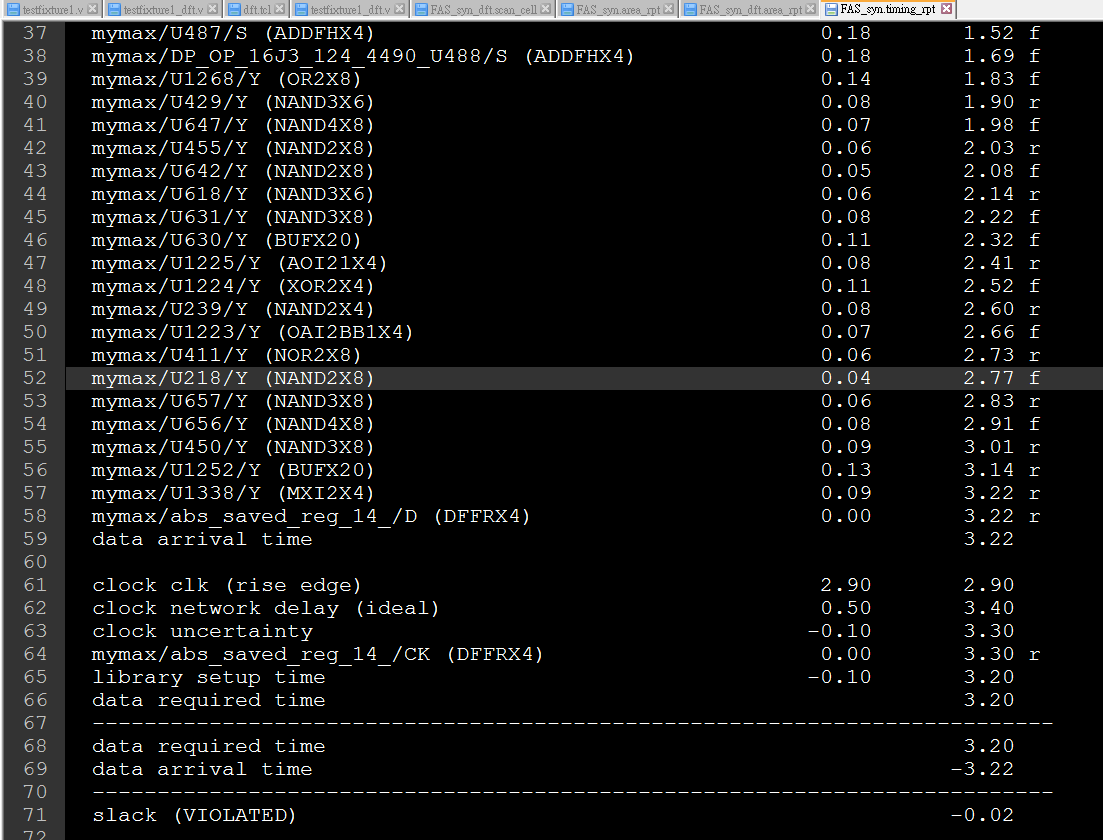
Because we replace original DFF to scan chain DFF(additional 2x1MUX)





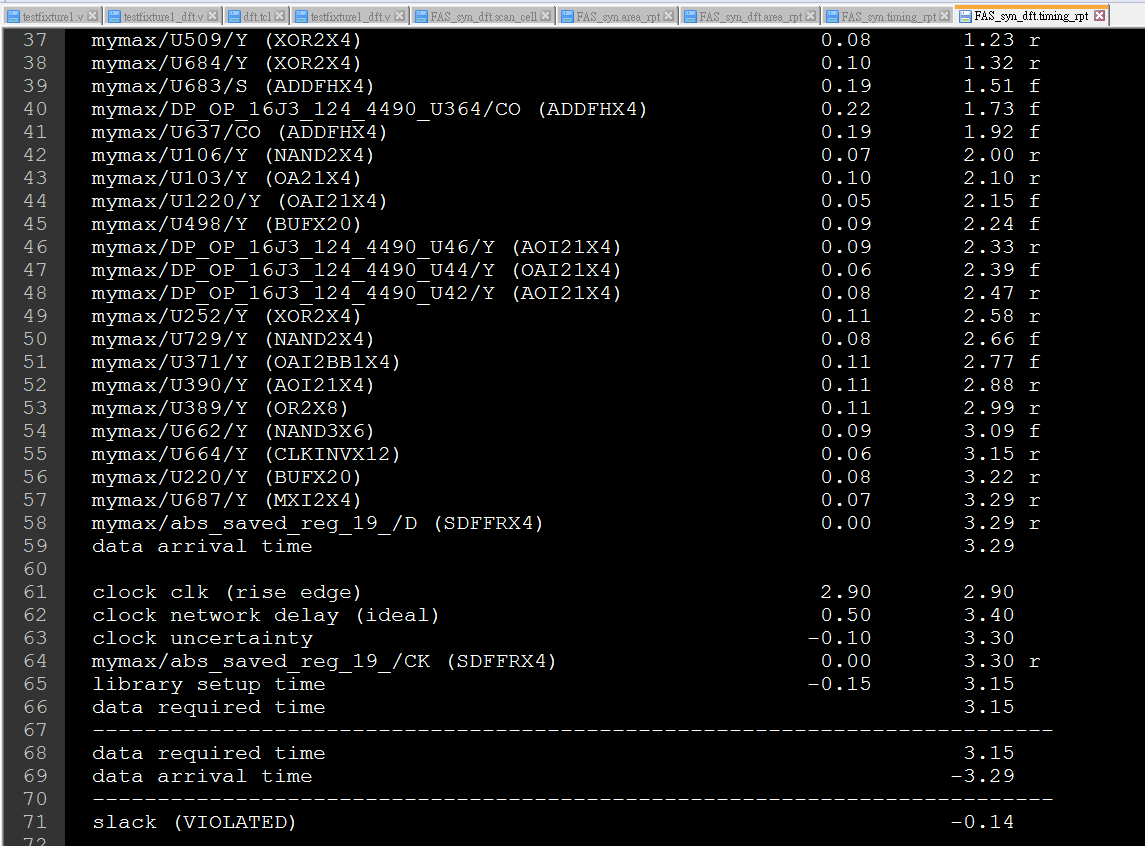
C-1.

data arrival time 3.22 + library setup time 0.1 = 3.23 ns



C-2.

data arrival time 3.29 + library setup time 0.15 = 3.44 ns



C-3.

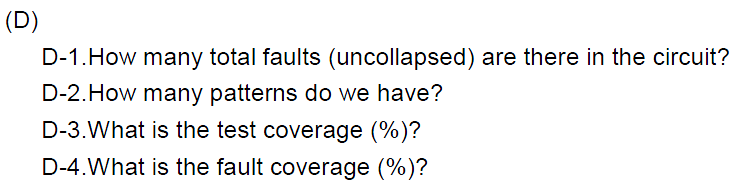
(3.44-3.23)/3.23=6.5%

Yes, the critical path in my design has changed, because mux-DFF change my setup time of critical path.

C-4.

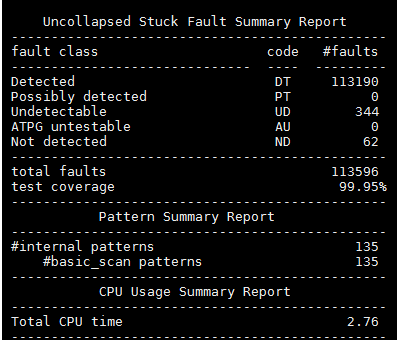
Original: DFF input of critical path without mux.

After DFT: DFF input of critical path is added a mux, with increase critical path delay



D-1.

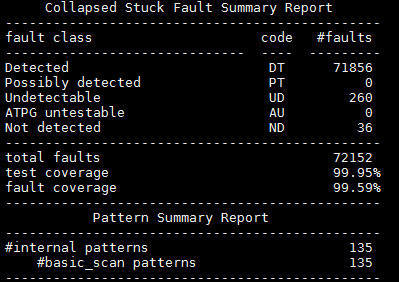
113596



D-2.

135

D-3.



99.95%

D-4.

99.59%