

0 Pre-CompileUltra

This section is just for some sanity check and getting familiar with the workspace and tools. First we load the libraries and elaborate our design, then we link them together and afterwards write our constraints and report timing before and after loading the constraints.

```
*****
Report : timing
  -path full
  -delay max
  -max_paths 20
Design : wembley_88
Version: S-2021.06-SP5
Date   : Tue Jun  7 16:29:35 2022
*****  

Operating Conditions: ss0p75v125c  Library: saed32hvt_ss0p75v125c
Wire Load Model Mode: enclosed  

Startpoint: eric/reg_out/q_reg
             (rising edge-triggered flip-flop)
Endpoint: Yout (output port)
Path Group: (none)
Path Type: max  

Des/Clust/Port      Wire Load Model      Library
wembley_88          ForQA                saed32hvt_ss0p75v125c  

Point           Incr     Path
-----  

eric/reg_out/q_reg/clocked_on (**SEQGEN**)
eric/reg_out/q_reg/Q (**SEQGEN**)
eric/reg_out/q_(reg_0)
eric/Yout (eric_clapton)
Yout (out)
data arrival time
-----
```

Figure 1: before constraints

As we can see from the figure above since we don't have any constraints everything is ideal and all of the timing delays are zero.

```
*****
Report : timing
  -path full
  -delay max
  -max_paths 3
Design : wembley_88
Version: S-2021.06-SP5
Date   : Sun Jun 12 13:34:45 2022
*****  

Operating Conditions: ss0p75v125c  Library: saed32hvt_ss0p75v125c
Wire Load Model Mode: enclosed  

Startpoint: reset (input port clocked by clk)
Endpoint: eric/regA/q_reg[2]
             (rising edge-triggered flip-flop clocked by clk)
Path Group: clk
Path Type: max  

Des/Clust/Port      Wire Load Model      Library
wembley_88          ForQA                saed32hvt_ss0p75v125c
reg4                ForQA                saed32hvt_ss0p75v125c  

Point           Incr     Path
-----  

clock clk (rise edge)          0.00    0.00
clock network delay (ideal)    0.00    0.00
input external delay           0.00    0.00 f
reset (in)                    0.00    0.00 f
eric/reset (eric_clapton)      0.00    0.00 f
eric/regA/reset (reg4)         0.00    0.00 f
eric/regA/I 0/2 (GTECH NOT)    2.87    2.95 r
eric/regA/B 0/2 (GTECH BUF)    0.00    2.95 r
eric/regA/C14/2_2 ("SELECT_OP_2.4_2.3.4") 0.00    2.96 r
eric/regA/q_reg[2]/next_state (**SEQGEN**)
data arrival time               0.00    2.96  

-----  

clock clk (rise edge)          0.00    0.00
clock network delay (ideal)    0.00    0.00
clock uncertainty              -0.08   0.72
eric/regA/q_reg[2]/clocked_on (**SEQGEN**)
library setup time              0.00    0.72
data required time              0.00    0.72  

-----  

data required time              0.72
data arrival time               -2.96
-----
```

Figure 2: after constraints and before Compile_Ultra

From the figure above we can see that after sourcing our constraints file we got a negative slack of -2.24, this doesn't worry us since it's before compiling and mapping to a technology specific library, from the report we can see that we have generic-technology cells, again this is just for checking that our constraint work and have the right syntax for the tool.

For our constraints we set the clock period $0.8ns$ as requested, and all the input/output delays and clock transition/uncertainty are chosen to be 10% of the clock period hence $0.08ns$, afterwards we set our 3 false paths as they can be seen from the eric-clapton diagram from the moodle.

```
my_constraints.con (/project/advvlsi/users/omarbishara/ws/pro_1/project_root/inputs)
File Edit Tools Syntax Buffers Window Help
reset_design
create_clock -name "clk" -period 0.8 [get_ports clk]

## This applies a delay for all inputs except the clock
set_input_delay 0.08 -clock clk [remove_from_collection [all_inputs] [get_ports clk]]

set_output_delay 0.08 -clock clk [all_outputs]
set_clock_uncertainty 0.08 [get_clocks clk]
set_clock_transition 0.08 [get_clocks clk]

## here we set our false paths
set_false_path -through [get_pins eric/mux4to1_a[0]] -through [get_pins eric/mux_as/a[0]]
set_false_path -through [get_pins eric/mux4to1_a[1]] -through [get_pins eric/mux_as/a[0]]
set_false_path -through [get_pins eric/mux4to1_a[2]] -through [get_pins eric/mux_as/a[0]]
```

Figure 3: Constraints file screenshot

1 Compile_Ultra -no_autoungroup

We found 21 violating paths in the design.

The three reports with worst negative slack are attached below each with it's WNS.

	Incr	Path
Point		
clock clk (rise edge)	0.00	0.00
clock network delay (ideal)	0.00	0.00
sultan/B_out_reg[0]/CLK (DFFSSRX2_LVT)	0.00	0.00 r
sultan/B_out_reg[0]/QN (DFFSSRX2_LVT)	0.19	0.19 f
sultan/B_out[0] (sultans_of_swing)	0.00	0.19 f
dire/B_out[0] (dire_streets)	0.00	0.19 f
dire/adder_1/b[0] (unsigned_adder4)	0.00	0.19 f
dire/adder_1/U4/Y (AND2X1_LVT)	0.19	0.38 f
dire/adder_1/intadd_0/U4/CO (FADDX1_LVT)	0.15	0.53 f
dire/adder_1/intadd_0/U3/CO (FADDX1_LVT)	0.16	0.69 f
dire/adder_1/U2/CO (FADDX1_LVT)	0.16	0.84 f
dire/adder_1/C_Out (unsigned_adder4)	0.00	0.84 f
dire/carry_Xor/S_5 (Bitwise_Xor4_carry)	0.00	0.84 f
dire/carry_Xor/U5/Y (NBUFFX2_LVT)	0.15	0.99 f
dire/carry_Xor/U3/Y (XOR2X2_LVT)	0.24	1.24 r
dire/carry_Xor/C_s[3] (Bitwise_Xor4_carry)	0.00	1.24 r
dire/B_e[3] (dire_streets)	0.00	1.24 r
eric/B_e[3] (eric_clapton)	0.00	1.24 r
eric/regB/d[3] (reg4_1)	0.00	1.24 r
eric/regB/q_reg[3]/RSTB (DFFSSRX1_LVT)	0.02	1.25 r
data arrival time		1.25
clock clk (rise edge)	0.80	0.80
clock network delay (ideal)	0.00	0.80
clock uncertainty	-0.08	0.72
eric/regB/q_reg[3]/CLK (DFFSSRX1_LVT)	0.00	0.72 r
library setup time	-0.15	0.57
data required time		0.57
data required time		0.57
data arrival time		-1.25
slack (VIOLATED)		-0.69

Figure 4: $WNS = -0.69ns$

Point	Incr	Path
clock clk (rise edge)	0.00	0.00
clock network delay (ideal)	0.00	0.00
sultan/A_out_reg[0]/CLK (DFFX1_LVT)	0.00	0.00 r
sultan/A_out_reg[0]/Q (DFFX1_LVT)	0.23	0.23 r
sultan/A_out[0] (sultans_of_swing)	0.00	0.23 r
dire/A_out[0] (dire_straits)	0.00	0.23 r
dire/adder_1/a[0] (unsigned_adder4)	0.00	0.23 r
dire/adder_1/U4/Y (AND2X1_LVT)	0.13	0.36 r
dire/adder_1/intadd_0/U4/C0 (FADDX1_LVT)	0.16	0.51 r
dire/adder_1/intadd_0/U3/C0 (FADDX1_LVT)	0.16	0.67 r
dire/adder_1/U2/C0 (FADDX1_LVT)	0.16	0.84 r
dire/adder_1/C_Out (unsigned_adder4)	0.00	0.84 r
dire/carry_Xor/S_s (Bitwise_Xor4_carry)	0.00	0.84 r
dire/carry_Xor/U6/Y (NBUFFX2_LVT)	0.15	0.98 r
dire/carry_Xor/U2/Y (XOR2X2_LVT)	0.25	1.24 f
dire/carry_Xor/C_s[1] (Bitwise_Xor4_carry)	0.00	1.24 f
dire/B_e[1] (dire_straits)	0.00	1.24 f
eric/B_e[1] (eric_clapton)	0.00	1.24 f
eric/regB/d[1] (reg4_1)	0.00	1.24 f
eric/regB/U6/Y (AND2X1_LVT)	0.09	1.33 f
eric/regB/q_reg[1]/D (DFFX1_LVT)	0.01	1.35 f
data arrival time		1.35
clock clk (rise edge)	0.80	0.80
clock network delay (ideal)	0.00	0.80
clock uncertainty	-0.08	0.72
eric/regB/q_reg[1]/CLK (DFFX1_LVT)	0.00	0.72 r
library setup time	-0.06	0.66
data required time		0.66
data required time		0.66
data arrival time		-1.35
slack (VIOLATED)		-0.68

Figure 5: $WNS = -0.68ns$

Point	Incr	Path
clock clk (rise edge)	0.00	0.00
clock network delay (ideal)	0.00	0.00
sultan/A_out_reg[0]/CLK (DFFX1_LVT)	0.00	0.00 r
sultan/A_out_reg[0]/Q (DFFX1_LVT)	0.23	0.23 r
sultan/A_out[0] (sultans_of_swing)	0.00	0.23 r
dire/A_out[0] (dire_straits)	0.00	0.23 r
dire/adder_1/a[0] (unsigned_adder4)	0.00	0.23 r
dire/adder_1/U4/Y (AND2X1_LVT)	0.13	0.36 r
dire/adder_1/intadd_0/U4/C0 (FADDX1_LVT)	0.16	0.51 r
dire/adder_1/intadd_0/U3/C0 (FADDX1_LVT)	0.16	0.67 r
dire/adder_1/U2/C0 (FADDX1_LVT)	0.16	0.84 r
dire/adder_1/C_Out (unsigned_adder4)	0.00	0.84 r
dire/carry_Xor/S_s (Bitwise_Xor4_carry)	0.00	0.84 r
dire/carry_Xor/U6/Y (NBUFFX2_LVT)	0.15	0.98 r
dire/carry_Xor/U1/Y (XOR2X2_LVT)	0.25	1.24 f
dire/carry_Xor/C_s[0] (Bitwise_Xor4_carry)	0.00	1.24 f
dire/B_e[0] (dire_straits)	0.00	1.24 f
eric/B_e[0] (eric_clapton)	0.00	1.24 f
eric/regB/d[0] (reg4_1)	0.00	1.24 f
eric/regB/U7/Y (AND2X1_LVT)	0.09	1.33 f
eric/regB/q_reg[0]/D (DFFX1_LVT)	0.01	1.35 f
data arrival time		1.35
clock clk (rise edge)	0.80	0.80
clock network delay (ideal)	0.00	0.80
clock uncertainty	-0.08	0.72
eric/regB/q_reg[0]/CLK (DFFX1_LVT)	0.00	0.72 r
library setup time	-0.06	0.66
data required time		0.66
data required time		0.66
data arrival time		-1.35
slack (VIOLATED)		-0.68

Figure 6: $WNS = -0.68ns$

From the reports we can see that the biggest time increases are from the MSB in the adder, and the carry-bit in the bit-wise-xor. as we guesses this is the critical path of our design since that these two bits go through a long path of combinational gates and the fact that the carry bit is dependant on the result of the adder of the previous four bits.

Now we'll attach a screenshot of the violating path in schematic view and explain how we might solve the violation.



Figure 7: Violation path with maximum WNS

The WNS of this path (and the design as a whole) is -0.69 which is less than 10% of the clock period, therefore we think that we can solve the violation without changing the clock period, we can change our standard-cell libraries that were implemented in this design and use ultra-low-vt to make the cells faster and compensate for the negative slack, ofcourse this means higher leakage power consumption, or we can add a pipelining stage (add registers to the path) but this means that our throughput of the design is lower, or we can (my favourite solution) buffer the hell out of the design, adding buffers can solve this violation.

2 Compile_Ultra

Now we compile our design without the `-no_autoungroup` option, which means we don't constraint our tool in the sense that it now we don't let the tool optimize each module independently of the other modules BUT let the tool actually optimize the whole design as one module which often gives better results (with the cost of higher computational price).

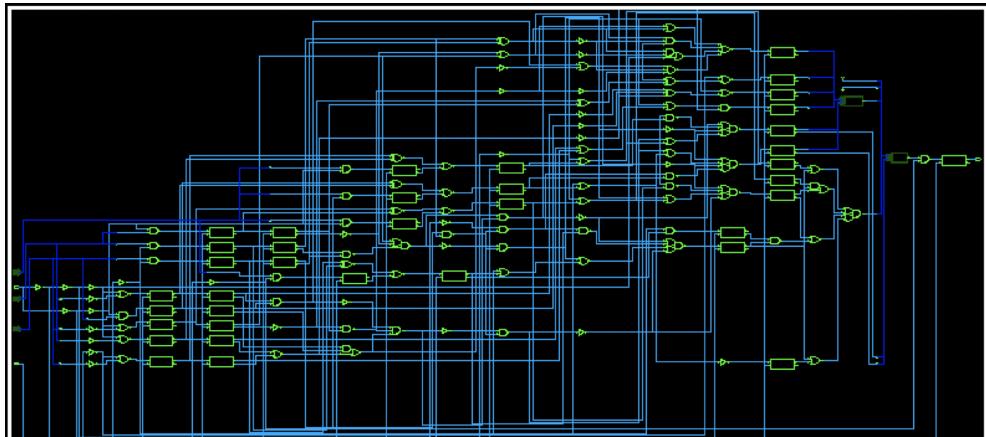


Figure 8: Schematic view of the design after compile ultra with autoungrouping

As we can see the schematic view has changed quite alot, now the worst negative slack is $-0.43ns$ which means the WNS improved by $0.26ns$ which is significant.
Also now we have 20 violating paths instead of 21.

3 Area

The estimated total area of the design is $553.94\mu m^2$

```
*****
Report : area
Design : wembley_88
Version: S-2021.06-SP5
Date   : Sun Jun 12 16:05:58 2022
*****

Library(s) Used:

    saed32lvt_ss0p75v125c (File: /project/advvlsi
125c.db)
    saed32rvt_ss0p75v125c (File: /project/advvlsi
125c.db)

Number of ports:                      29
Number of nets:                       188
Number of cells:                      152
Number of combinational cells:        111
Number of sequential cells:           37
Number of macros/black boxes:         0
Number of buf/inv:                   33
Number of references:                 22

Combinational area:                  246.011395
Buf/Inv area:                      49.558081
Noncombinational area:              244.486536
Macro/Black Box area:               0.000000
Net Interconnect area:              63.447144

Total cell area:                   490.497931
Total area:                         553.945075
```

Figure 9: Area report

4 Power consumption

As we can see registers consume 86.93% of the design's total power and the combinational cells consume 13.07% of the total power. so the ratio between the registers and combinational power consumption is ≈ 6.65

Power Group	Internal Power	Switching Power	Leakage Power	Total Power	(%)	Attrs
io_pad	0.0000	0.0000	0.0000	0.0000	(0.00%)	
memory	0.0000	0.0000	0.0000	0.0000	(0.00%)	
black_box	0.0000	0.0000	0.0000	0.0000	(0.00%)	
clock_network	0.0000	0.0000	0.0000	0.0000	(0.00%)	
register	152.4907	1.5354	5.2105e+06	159.2365	(86.93%)	
sequential	0.0000	0.0000	0.0000	0.0000	(0.00%)	
combinational	13.4254	3.3746	7.1498e+06	23.9498	(13.07%)	
Total	165.9160 uW	4.9100 uW	1.2360e+07 pW	183.1863 uW		

Figure 10: Power report

5 Frequency

The highest frequency our design can work at is the frequency at which there is no negative slack, since our WNS is $-0.43ns$ then we need to enlarge our period by that much.

$$T_{clk}^{new} = 0.8ns + 0.43n = 1.23ns$$

$$f_{max} = \frac{1}{T_{clk}^{new}} = 813MHz$$