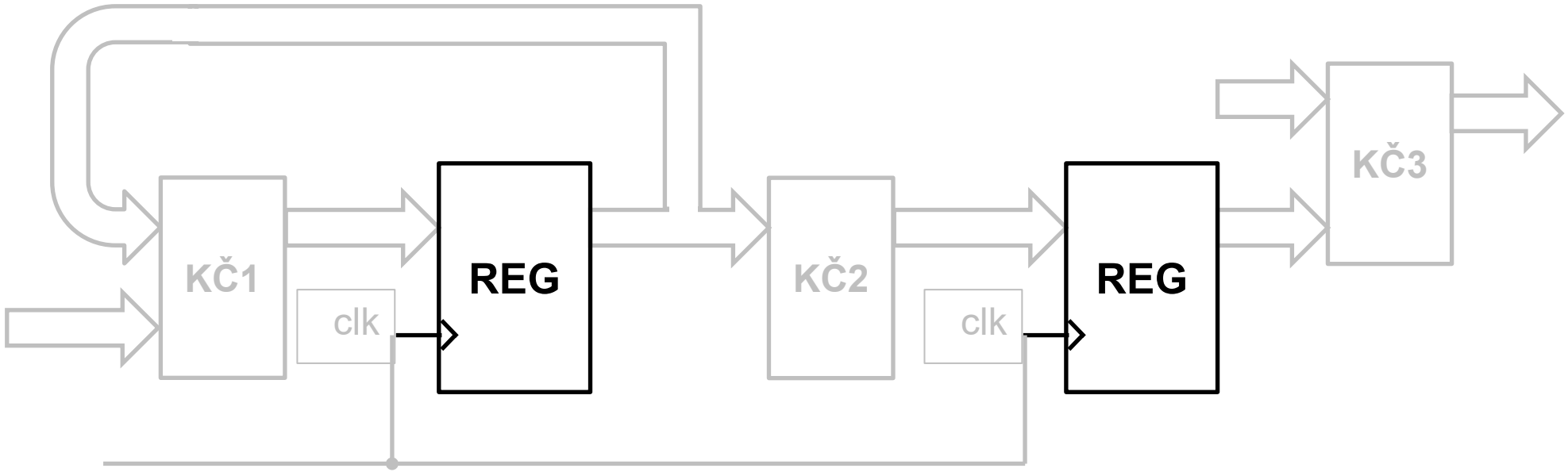
MUNI
FI

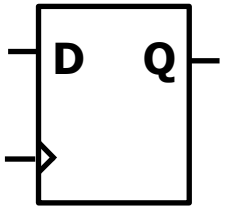
PA221: Timing Analysis, Pipelining and Register Retiming

jan.kral@fi.muni.cz

D Flip Flop

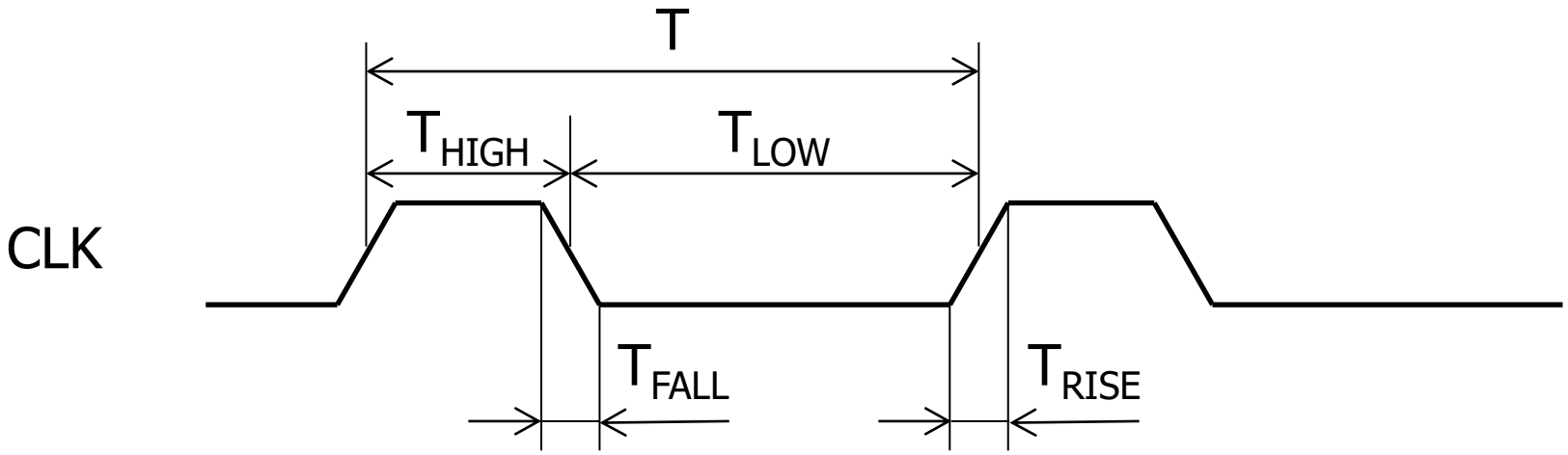


D Flip Flop



Clock signal requirements:

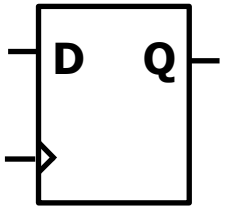
- Minimum period of signal H (T_{HIGH}) (Spartan-3: 0.79 ns)
- Minimum period of signal L (T_{LOW}) (Spartan-3: 0.79 ns)
- Minimum period / maximum frequency (T_{MIN} / F_{MAX}) (Spartan-3: 630 MHz)
- Maximum duration of falling and rising edge (T_{FALL} / T_{RISE})

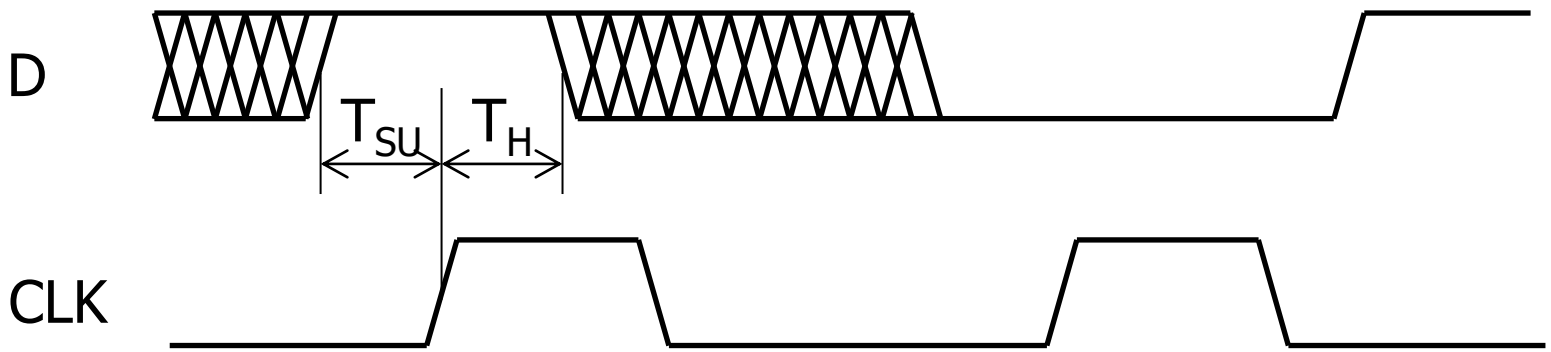


D Flip Flop

Requirements on input data signal:

- T_{SU} = **setup time** (Spartan-3: 0.53 ns)
- T_H = **hold time** (Spartan-3: 0 ns)

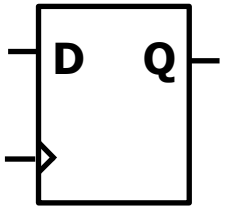


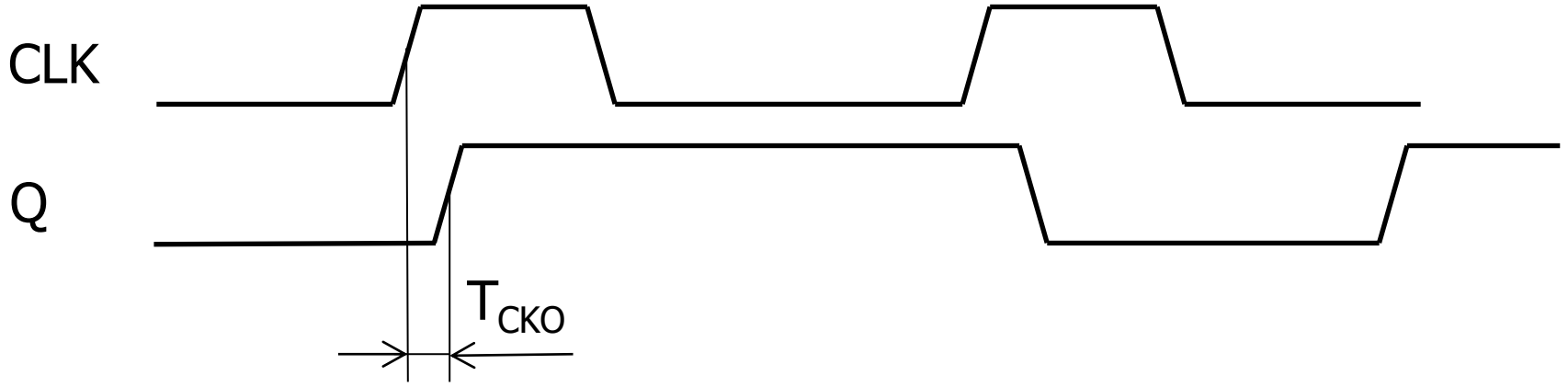


D Flip Flop

Time parameters of output signal

- T_{CKO} = **Clock to Output** (Spartan-3: 0.24 – 0.72 ns)

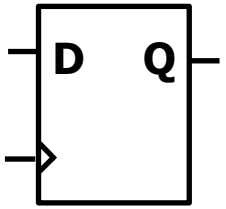


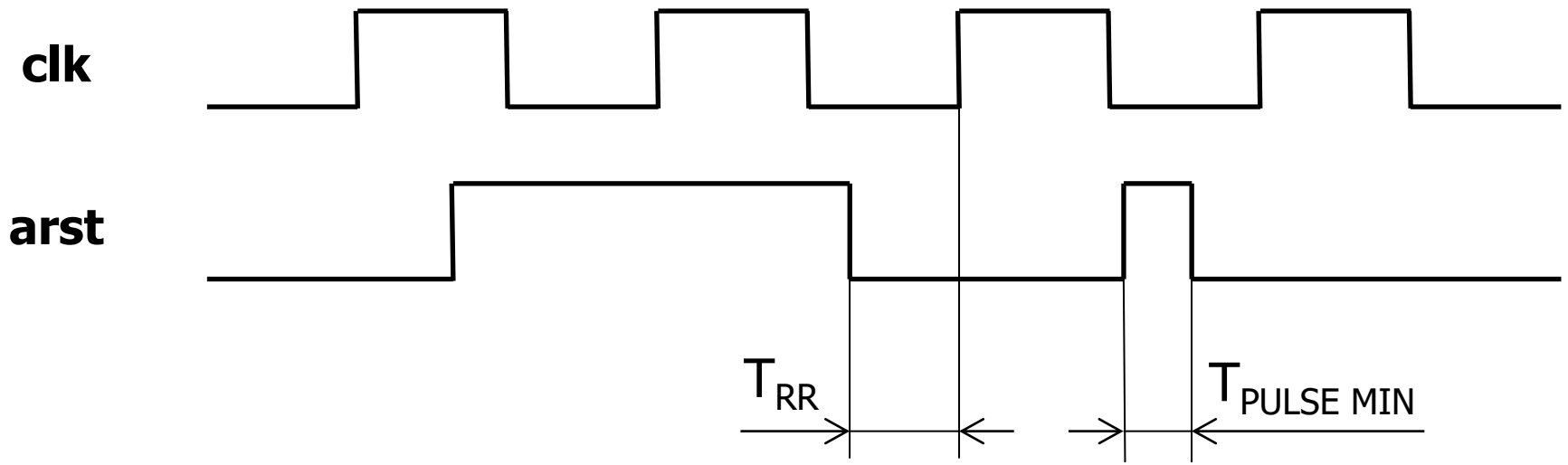


D Flip Flop

Requirements on **asynchronous reset**

- T_{RR} = **reset recovery time**
- $T_{PULSE\ MIN}$ = **minimum pulse width** (Spartan-3: 0.87 ns)

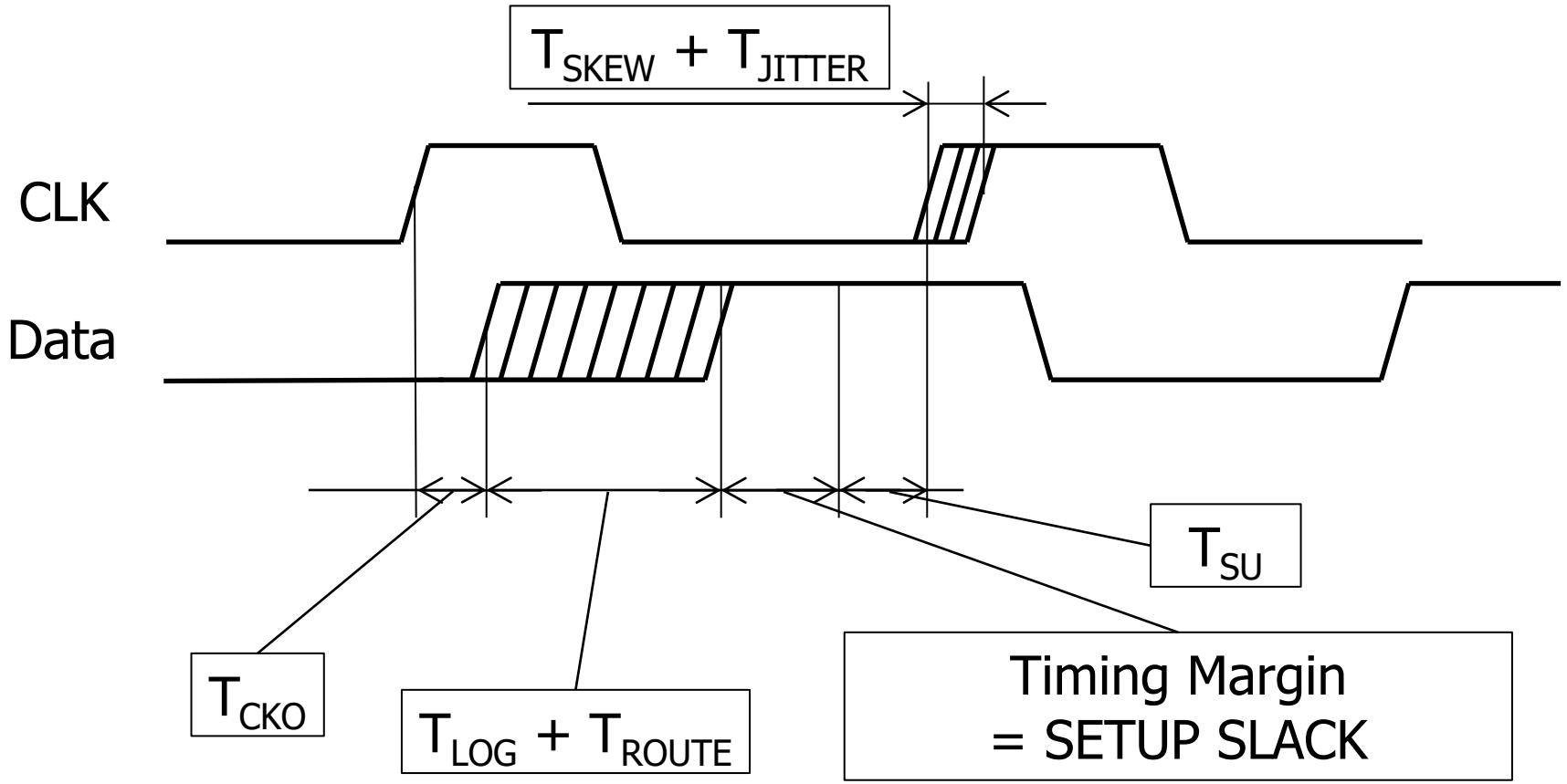




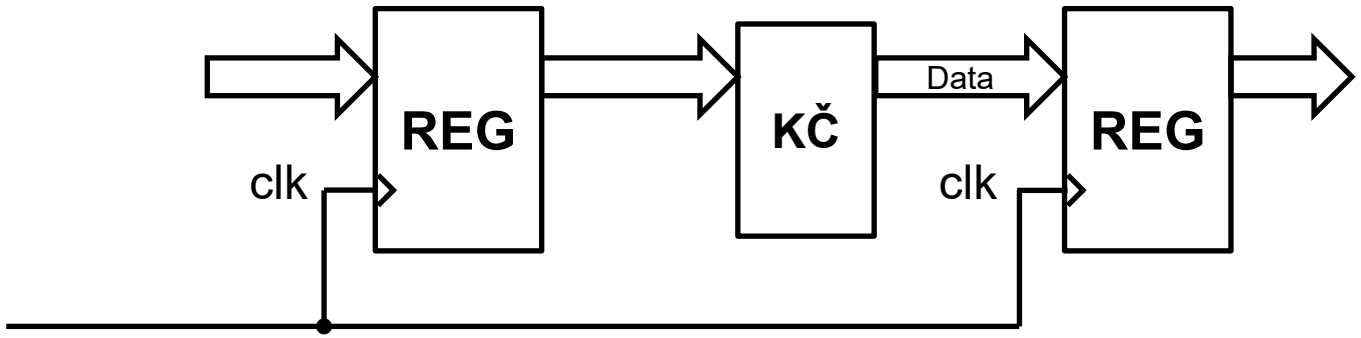
Static Timing

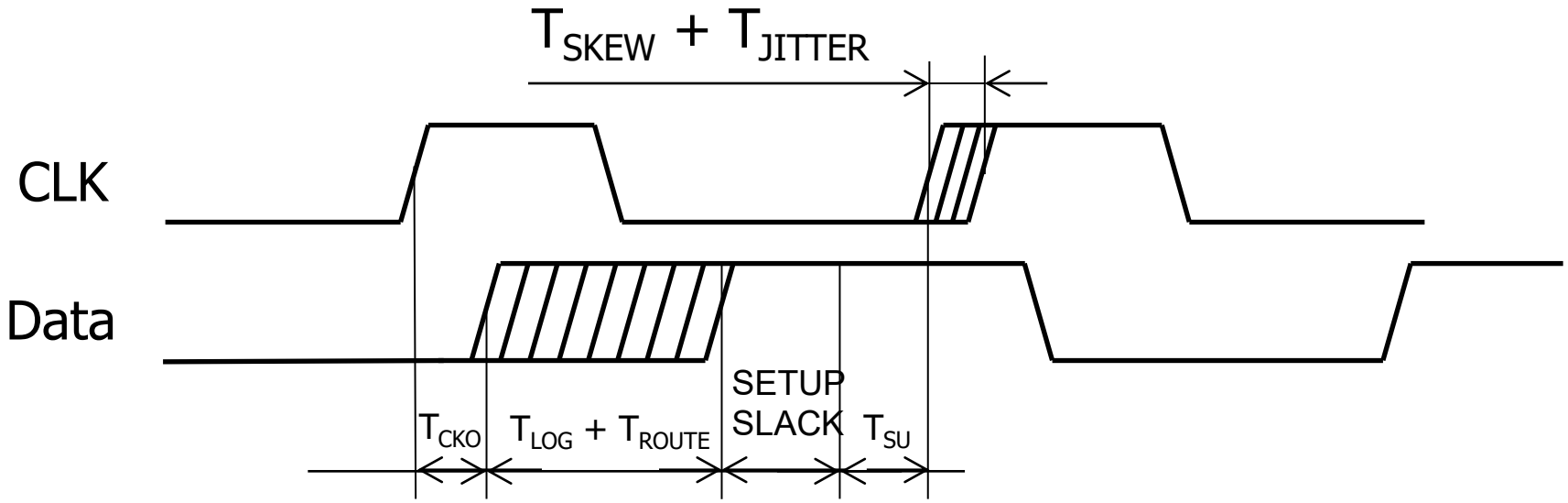
Timing Margin

Setup slack = Data required time – Data arrival time
Hold slack = Data arrival time – Data required time



Static Timing





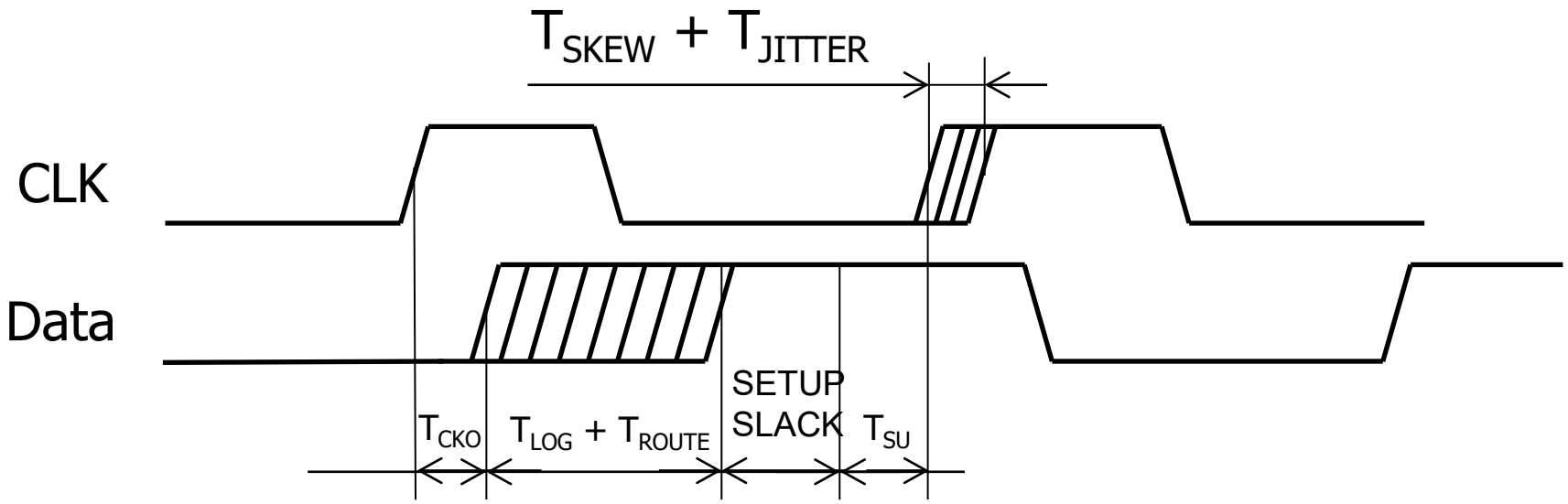
Timing Example

Clock period ($\mathbf{F_{MAX}}$ **= 630 MHz**): T_{MIN} = 1.59 ns

$(T_{LOG} + T_{ROUTE})_{MAX} = T_{MIN} - (T_{CKO} + T_{SU})$ = 1.59 ns – (0.72 ns + 0.53 ns) = **0.34 ns**

Clock period ($\mathbf{F_{MAX}}$ **= 200 MHz**): T_{MIN} = 5 ns

$(T_{LOG} + T_{ROUTE})_{MAX} = T_{MIN} - (T_{CKO} + T_{SU})$ = 5 ns – (0.72 ns + 0.53 ns) = **3.75 ns**



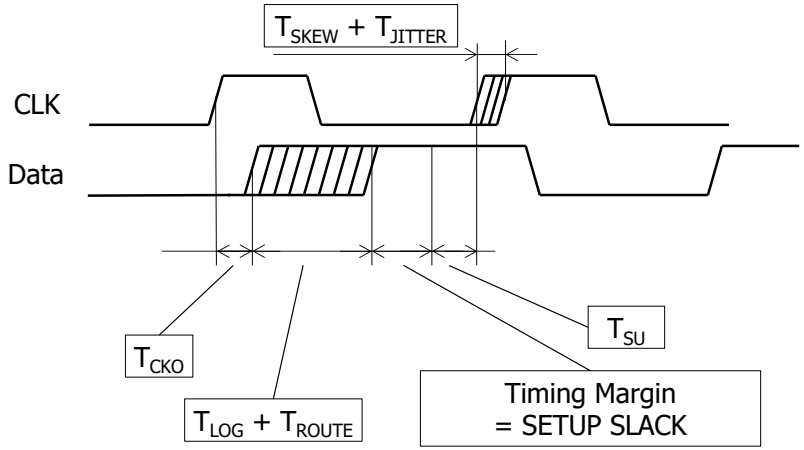
Timing Margin

Setup slack = Data required time – Data arrival time

Setup slack > 0 (**positive slack**) – systém is correctly placed & routed

Setup slack < 0 (**negative slack**) = **SETUP TIME VIOLATION**; system is not correctly placed & routed, it will (might) work incorrectly at the required clock frequency

HOLD TIME VIOLATION is usually caused by improper design or design constraints, i.e. violation of basic design rules and recommendation for FPGA systems.

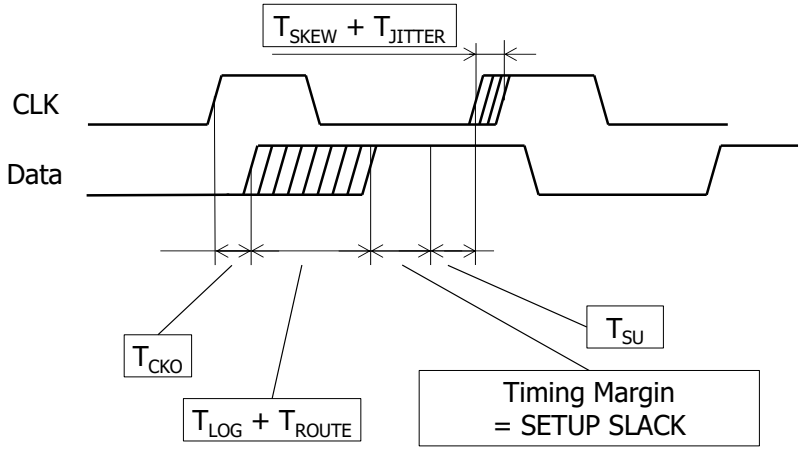


Static Timing Analysis (STA)

Checking requirements on timing of D Flip Flops and other blocks of the whole design. In the case of the maximum working frequency (designer specifies the maximum clock frequency in .sdc file in Quartus) STA especially verifies SETUP TIME.

To determine maximum working frequency (**Fmax**), SETUP TIME SLACK = 0.

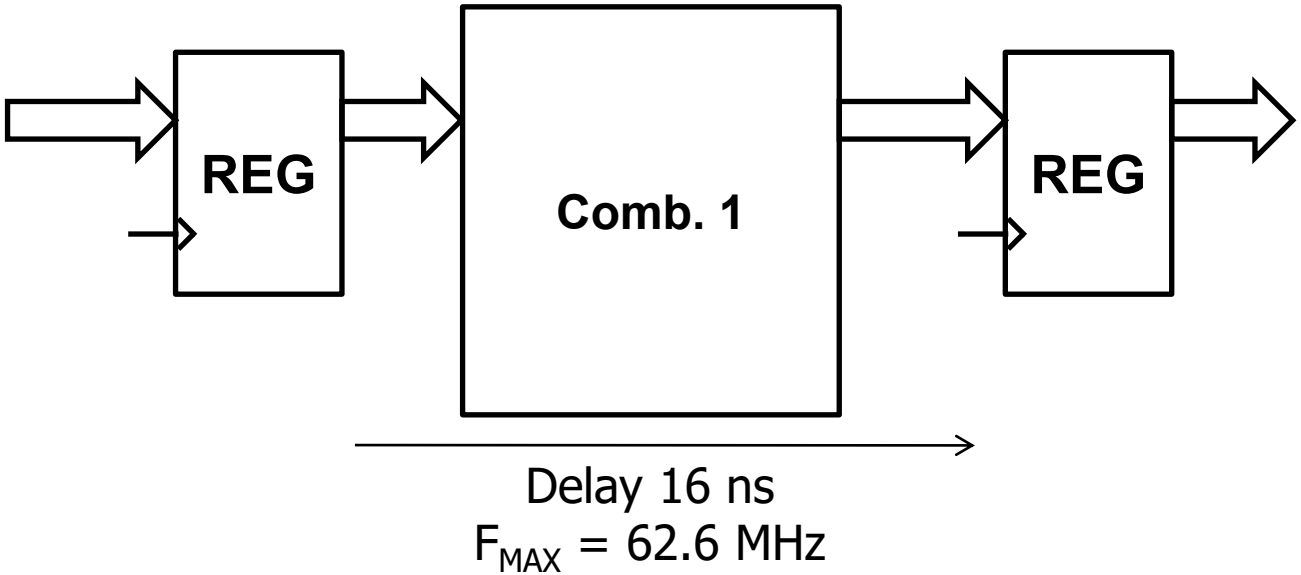
Maximum working frequency of FPGA design is determined by a path with the minimum setup time slack (**worst case path**).



Increasing Maximum Working Frequency

Pipelining

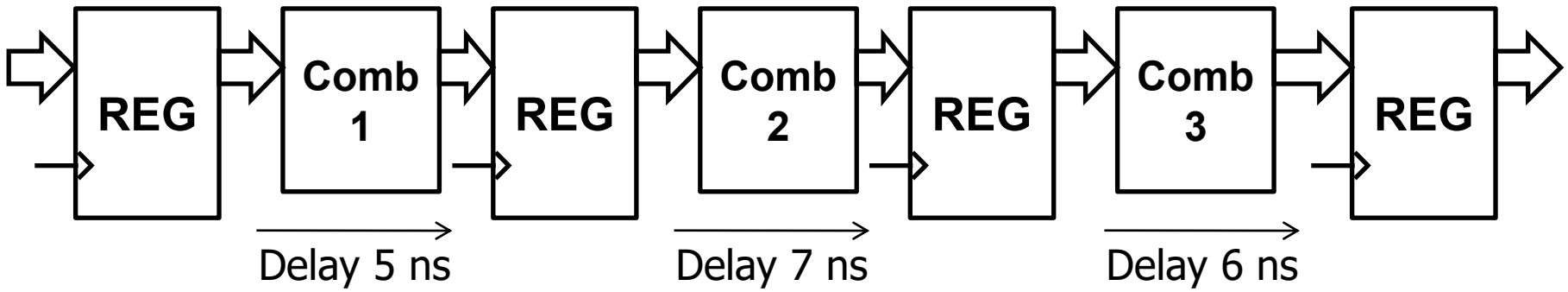
Split complicated tasks to more simpler tasks which can be faster.



Increasing Maximum Working Frequency

Pipelining

Split complicated tasks to more simpler tasks which can be faster.

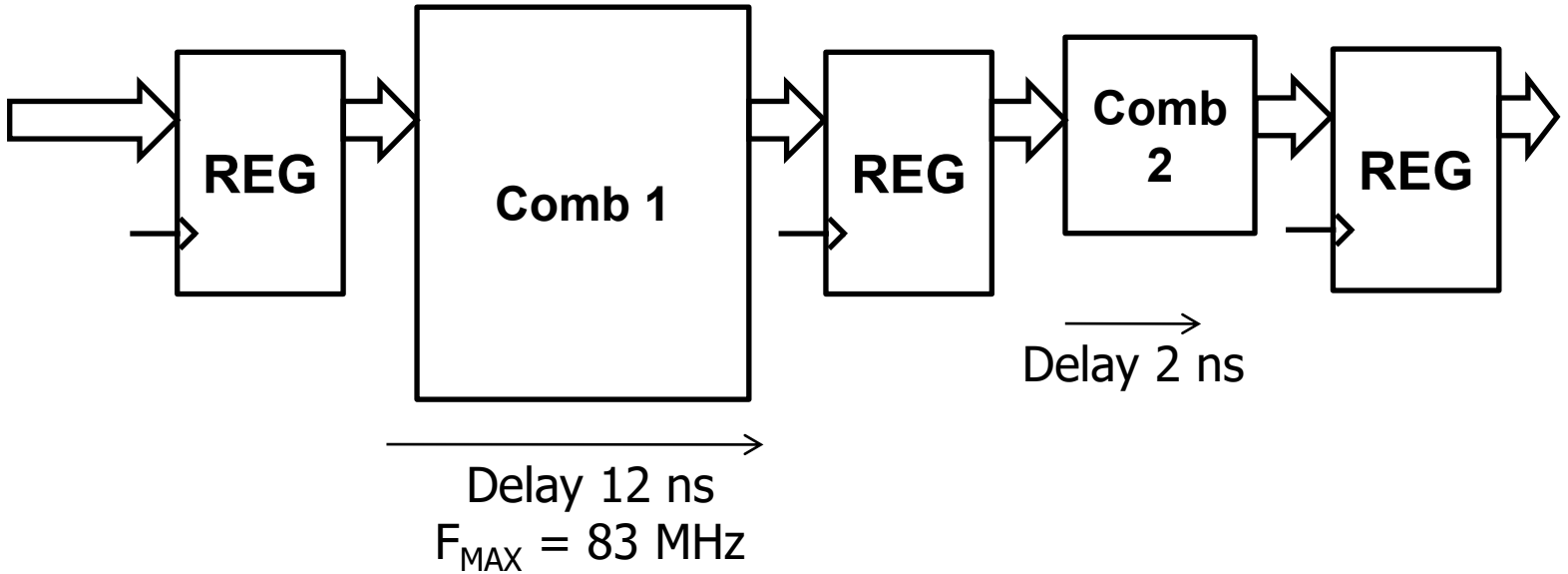


Maximum delay 7 ns => F_{MAX} = 143 MHz
(ideal delay 5.3 ns => F_{MAX} = 187 MHz)

Increasing Maximum Working Frequency

Register Retiming (Balancing)

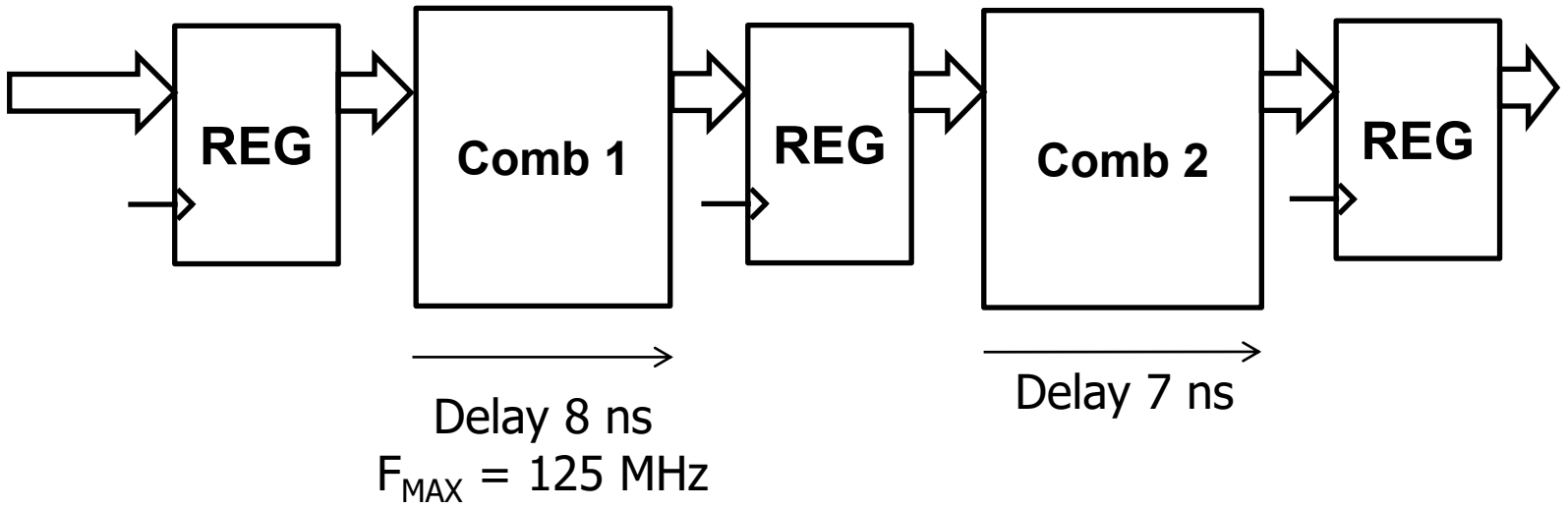
Move part of the large combinational function to a different stage. Leads to equivalent function.



Increasing Maximum Working Frequency

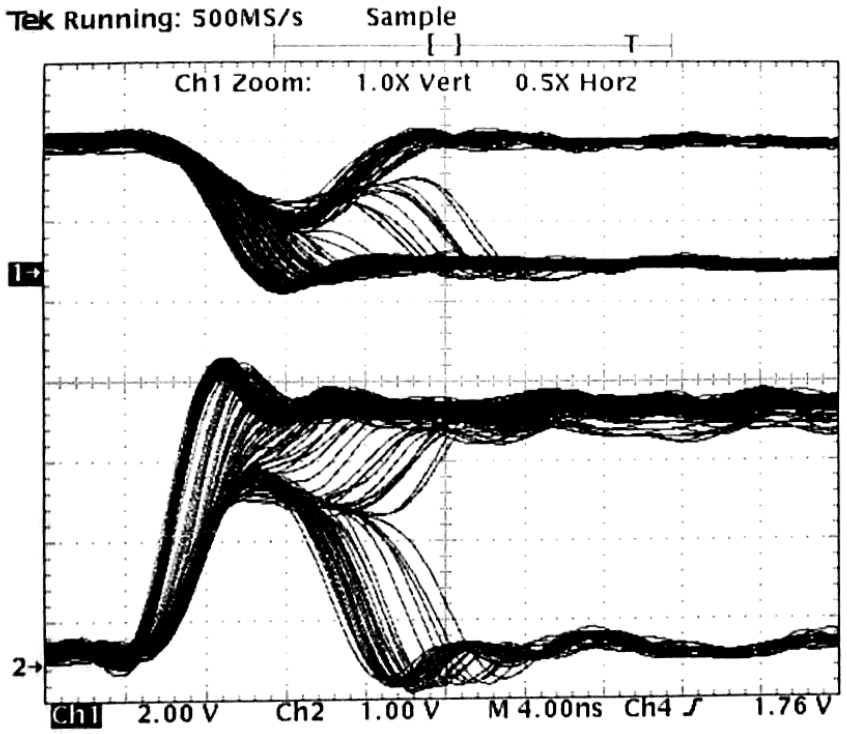
Register Retiming (Balancing)

Move part of the large combinational function to a different stage. Leads to equivalent function.



Metastability

- Temporary incorrect (undefined) output of a digital circuit
- Can be caused by **violating timing requirements** of D Flip Flops
- Can be caused by input data signals violating the specifications (slow rising edge, signal voltage around input comparator's reference voltage)
- Usually, it is only short impulse
- Generally, **metastability is dangerous and we try to avoid it**



Metastability

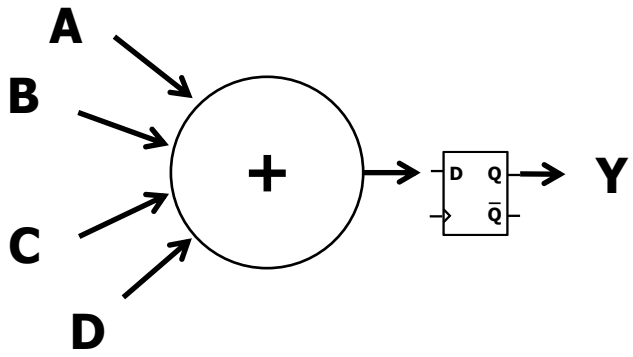
Consequences

- Leads to **random errors without replication possibility**
- Strongly depends on environment conditions : working frequency, power supply voltage, processed data signal, temperature, humidity, …
- Only certain hardware pieces can be affected (depends on the chip fabrication process)

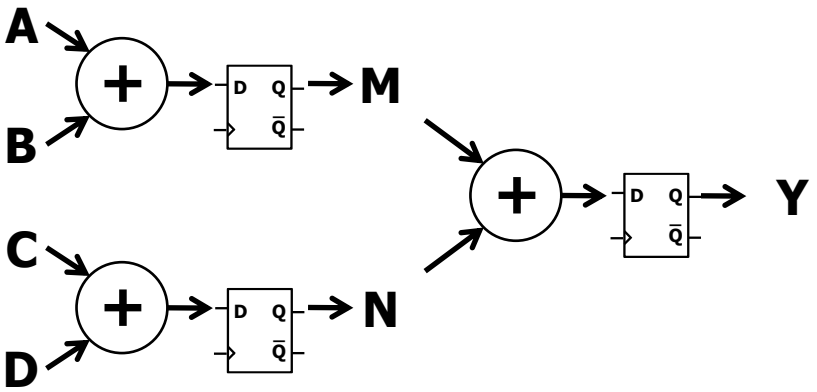
Task Assignment

Pipelining – example

```
always @(posedge clk) begin
  Y <= A + B + C + D;
end
```



```
always @(posedge clk) begin
  Y <= M + N;
  M <= A + B;
  N <= C + D;
end
```



Task Assignment

- ❑ Explore the provided template
- ❑ Set clock constrains in .sdc file
- ❑ Determine the maximum working frequency of the original design
- ❑ Use pipelining to achieve higher working frequency
- ❑ Verify that the function is unchanged (RTL viewer)
- ❑ Determine the maximum working frequency of the pipelined design
- ❑ Determine the maximum working frequency for adding four 16-bit numbers

Clock Constrains

Minimum constrains in .sdc file to analyze clock:

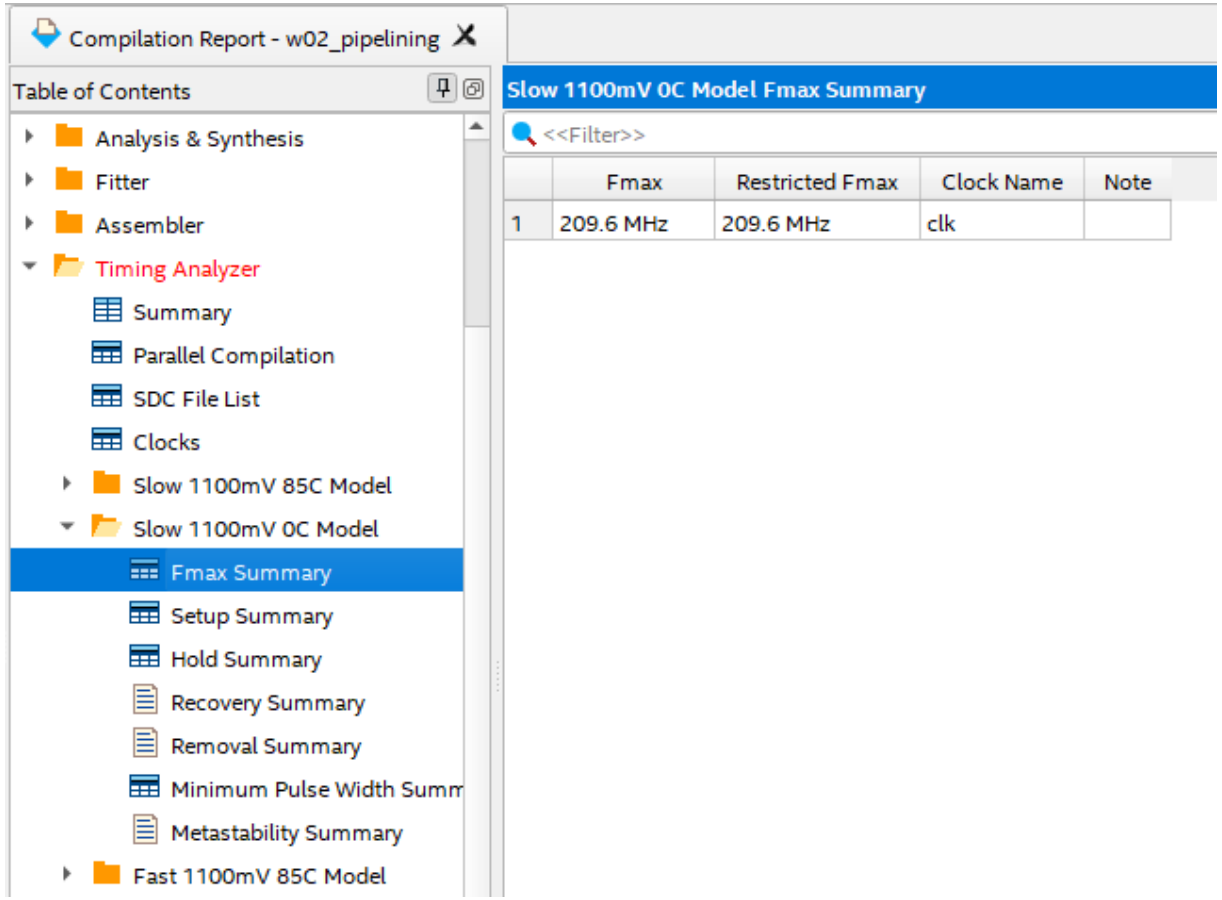
```
# create input clock which is 12MHz
create_clock -name clk -period 4 [get_ports {clk}]


# derive PLL clocks (required if your design instantiates a PLL)
derive_pll_clocks


# derive clock uncertainty
derive_clock_uncertainty
```

Maximum Working Frequency

- ❑ Maximum working frequency is output of Timing analyzer:
- ❑ This is the maximum working frequency for the design as it is currently routed.
- ❑ Placer & Router runs until they find a configuration which satisfies the clock frequency given in .sdc file.



Maximum Working Frequency

- To find the real maximum working frequency, you need to increase the clock frequency in .sdc file until router fails to find a solution.

- Effect of the increasing clock frequency:

Set frequency (.sdc)	Final design frequency (Timing Analyzer)
50 MHz	122 MHz
125 MHz	180 MHz
190 MHz	201 MHz
220 MHz	266 MHz
270 MHz	212 MHz

Note: Sigasi License

- License is available from A415 only.
- IP: 192.168.50.2
- Port: 27000