Digital Logic
Design with FPGA

Design Flow with Quartus

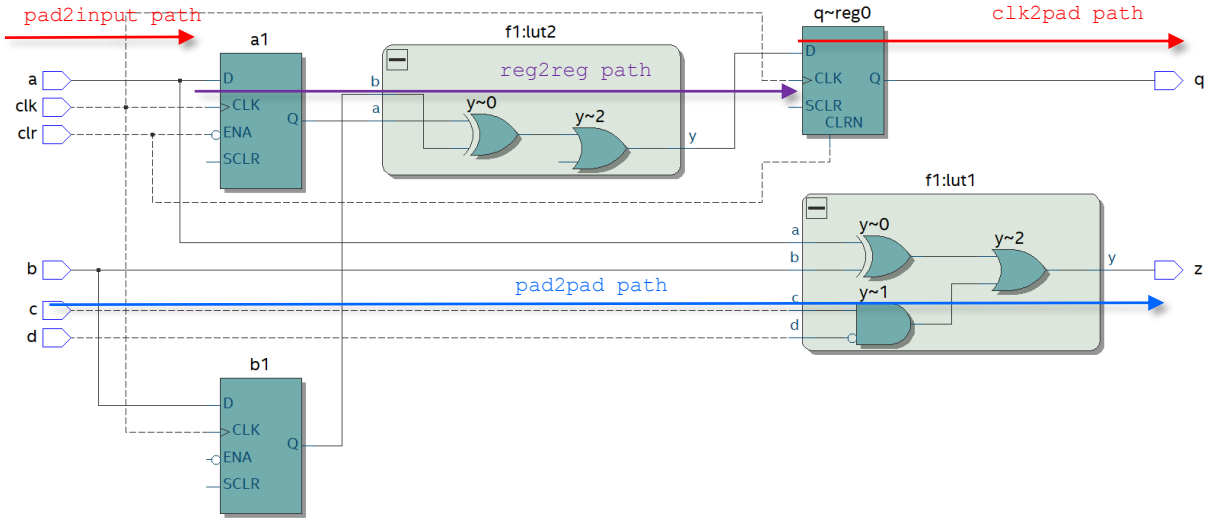
Timing & Power Analysis

Outline

- Purpose of timing analysis
- Synchronous analysis
- Asynchronous analysis
- Timing Analyzer
- Timing constraints
- Power analysis tools
  - Early Power Estimations
  - Power Analyzer

Purpose of Timing Analysis

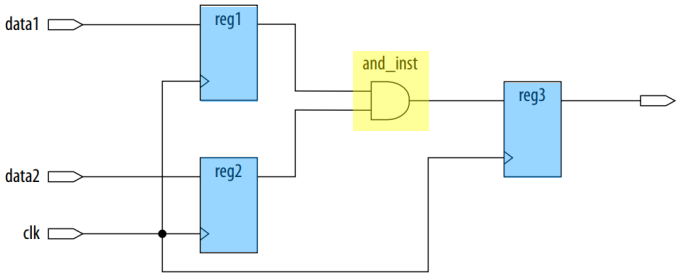
- To analyze design paths against required performance
- Catch timing-related errors faster and easier than gate-level simulation or board testing
- Designer must enter timing expectations
  - Used to guide Fitter during place & route task
  - Used to compare against actual results (post-fit)

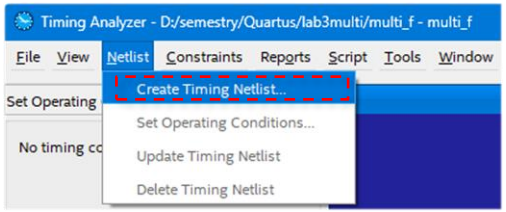


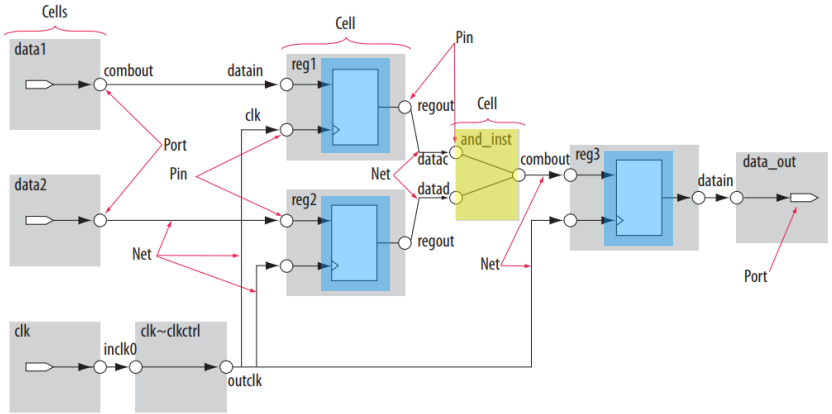
source: https://www.intel.com/content/www/us/en/docs/programmable/683068/18-1/timing-analysis-introduction.html

Timing Netlist

- The Timing Analyzer uses the timing netlist data to determine the data and clock arrival time versus required time for all timing paths
- The timing netlist is generated in the Timing Analyzer any time after running the Fitter or full compilation



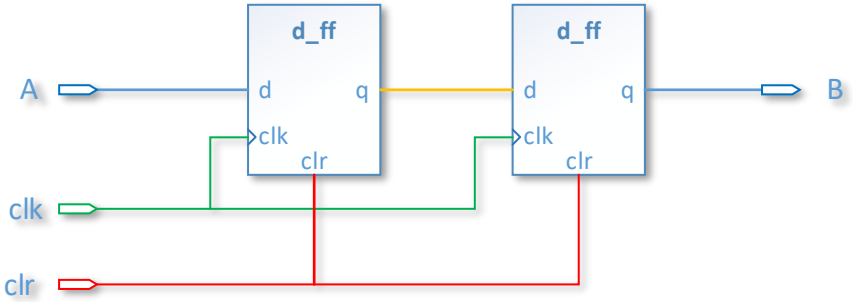




source: https://www.intel.com/content/www/us/en/docs/programmable/683068/18-1/timing-analysis-introduction.html

Timing Paths

- Timing paths connect two design nodes, such as the output of a register to the input of another register
  - Clock paths—connections from device ports or internally generated clock pins to the clock pin of a register
  - Data paths—connections from a port or the data output pin of a sequential element to a port or the data input pin of another sequential element
  - Asynchronous paths—connections from a port or asynchronous pins of another sequential element such as an asynchronous reset or asynchronous clear



- Types of analysis:
  - Synchronous: clock & data paths
  - Asynchronous: clock & async paths

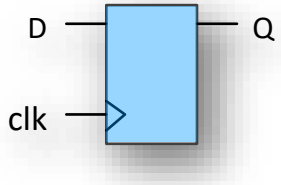
source: https://www.intel.com/content/www/us/en/docs/programmable/683068/18-1/timing-analysis-introduction.html

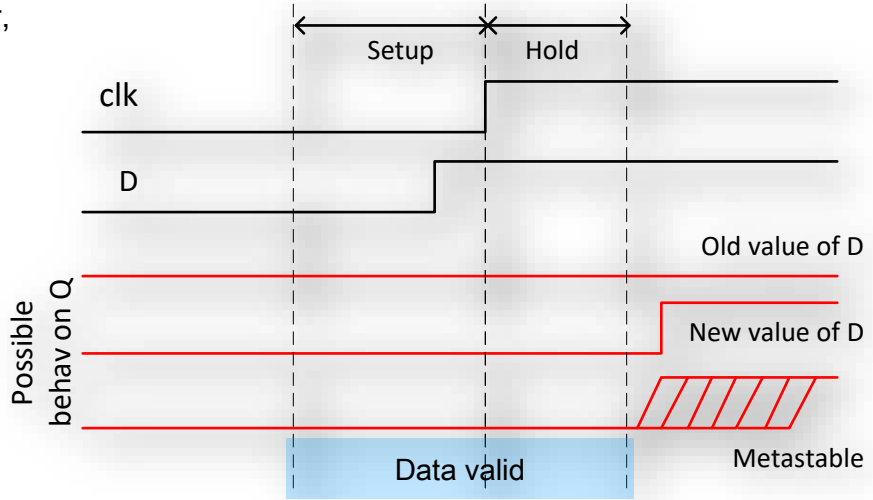
Synchronous Analysis

- Check if:
  - Clock and synchronous signals do not arrive at register input at the same time
  - Register output does not become metastable
  - Register transfer data in a known or guaranteed behavior

- Target timing parameters checked
  - Setup time (t_{su})
  - Hold time (t_h)

When you **violate the setup or hold time** of a register, you might oscillate the output, or set the output to an intermediate voltage level between the high and low levels called a **metastable state**





source: https://www.intel.com/content/www/us/en/docs/programmable/683068/18-1/timing-analysis-introduction.html

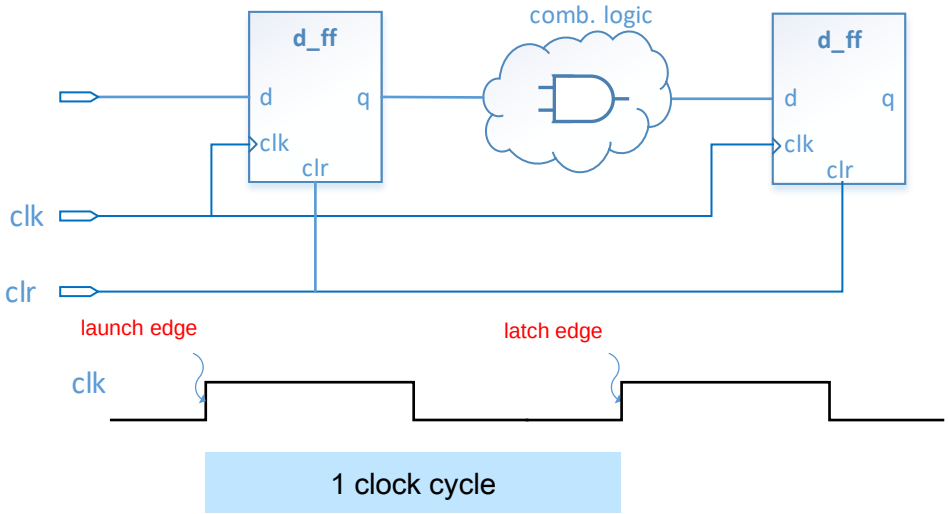
Synchronous Analysis

Launch edge:

- The edge which “launches” the data from the source register

Latch edge:

- The edge which “latches” the data at the destination register (with respect to the launch edge, selected by timing analyzer, typically 1 clock cycle, i.e. rising to rising edge)

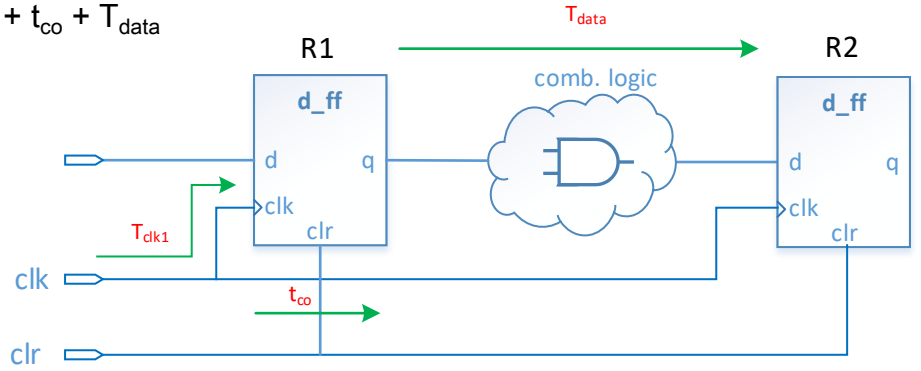


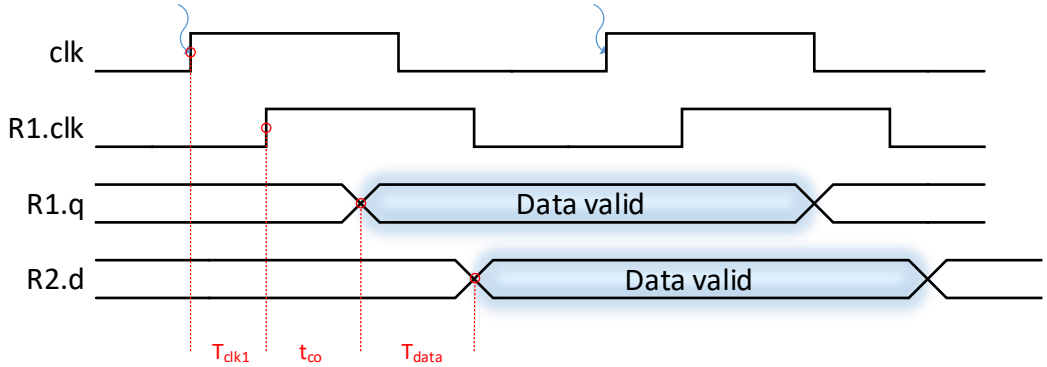
source: https://www.intel.com/content/www/us/en/docs/programmable/683068/18-1/timing-analysis-introduction.html

Synchronous Analysis

Data arrival time:

- The time for data to arrive at the destination register's D input
- Data arrival time = launch edge + T_{clk1} + t_{co} + T_{data}



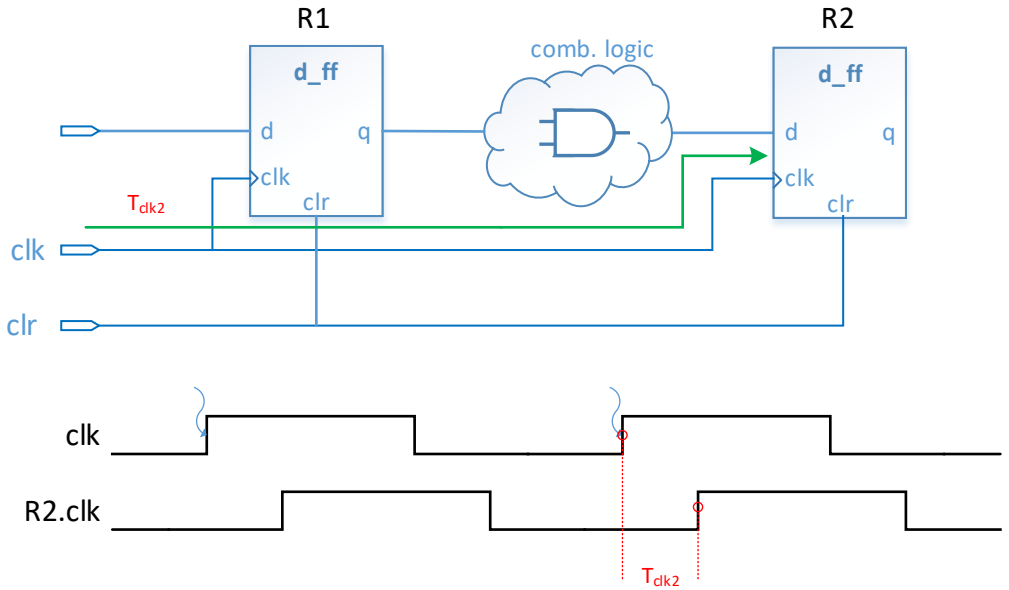


source: https://www.intel.com/content/www/us/en/docs/programmable/683068/18-1/timing-analysis-introduction.html

Synchronous Analysis

Clock arrival time:

- The time for clock to arrive at the destination register's clock input
- Clock arrival time = latch edge + T_{clk2}

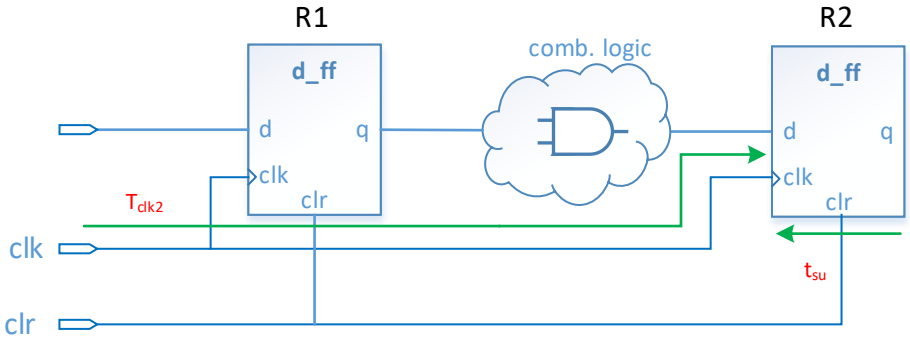


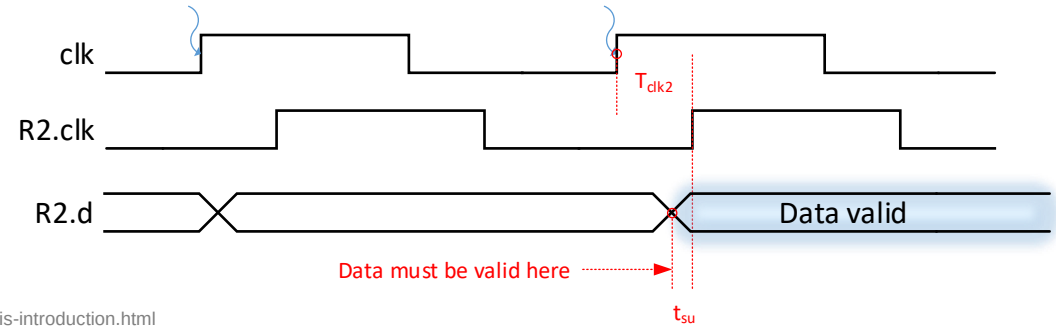
source: https://www.intel.com/content/www/us/en/docs/programmable/683068/18-1/timing-analysis-introduction.html

Synchronous Analysis

Data required time (Setup):

- The minimum time required for data to be valid before the latch edge (the data can be successfully latched into destination register R2)
- Data required time (Setup)= (latch edge + T_{clk2}) - t_{su} – Setup uncertainty



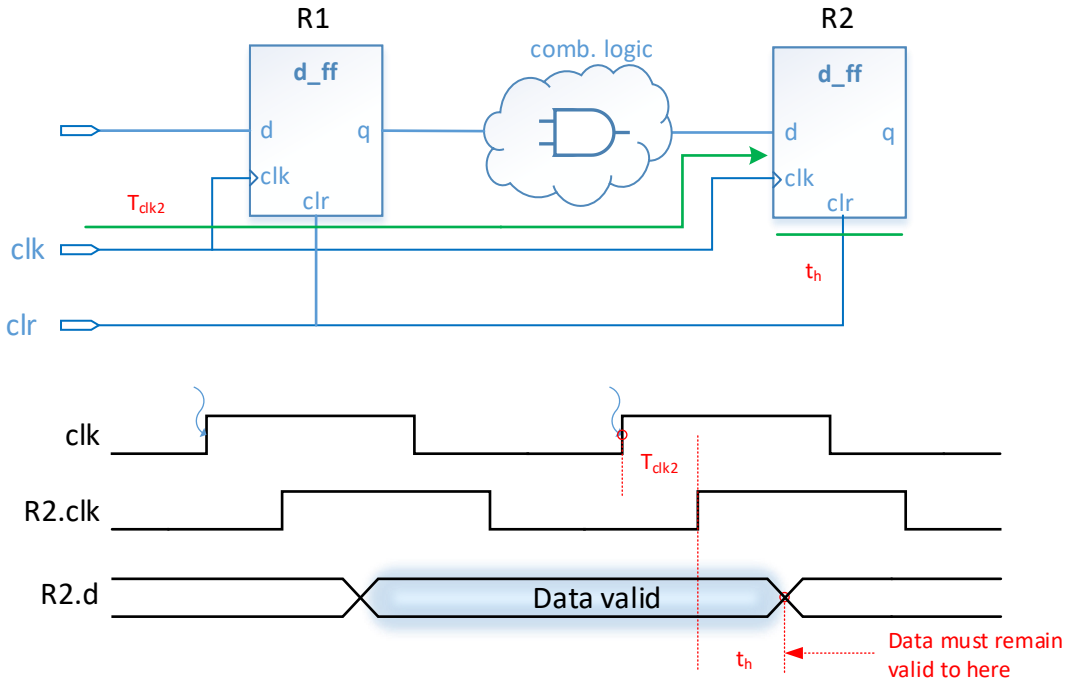


source: https://www.intel.com/content/www/us/en/docs/programmable/683068/18-1/timing-analysis-introduction.html

Synchronous Analysis

Data required time (Hold):

- The minimum time required after the latch edge for data to remain valid (the data can be successfully latched into destination register R2)
- Data required time (Hold)= (latch edge + T_{clk2}) + t_h + Hold uncertainty



source: https://www.intel.com/content/www/us/en/docs/programmable/683068/18-1/timing-analysis-introduction.html

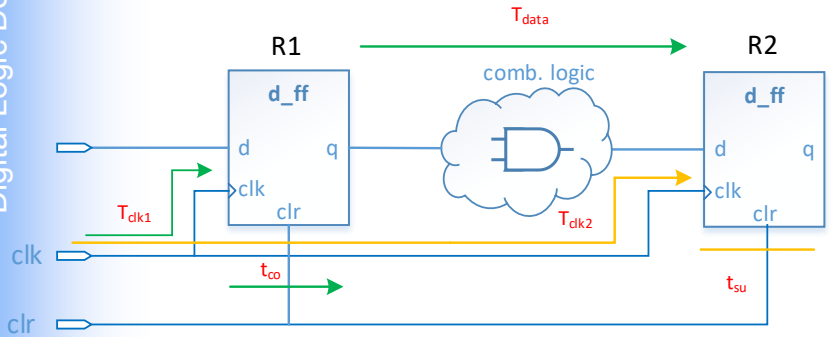
Synchronous Analysis

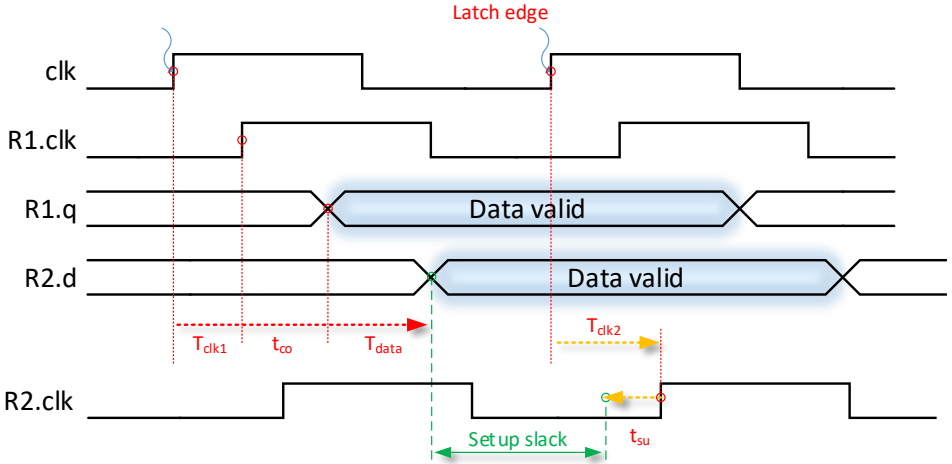
Calculating Setup Slack:

- Margin by witch the setup timing requirement is met
- Ensures launched data arrives in time to meet the latching requirement

Setup slack = Min. Data Required Time (Setup) – Max. Data Arrival Time

- Positive slack – timing requirement met
- Negative slack – timing requirement not met





source: https://www.intel.com/content/www/us/en/docs/programmable/683068/18-1/timing-analysis-introduction.html

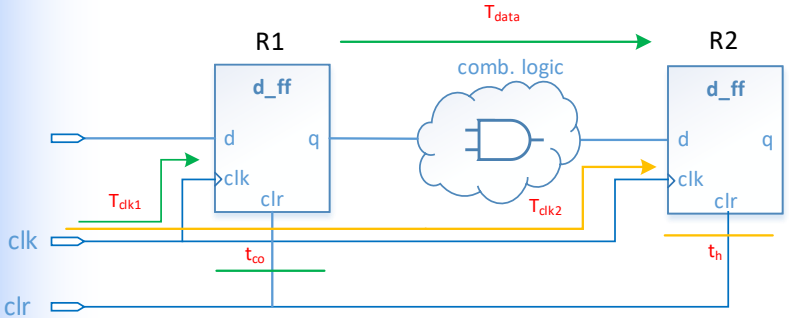
Synchronous Analysis

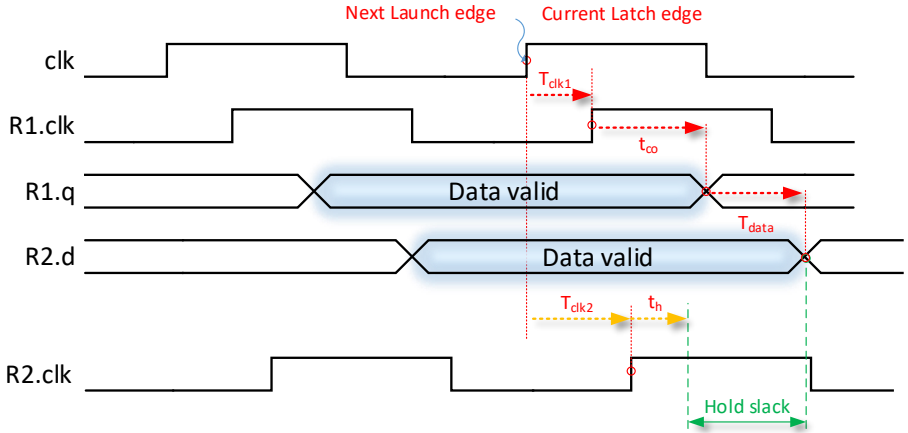
Calculating Hold Slack:

- Margin by witch the hold timing requirement is met
- Ensures latched data is not corrupted by data from the next launch edge (the input data cannot change to quickly after rising edge of the clock)

Hold slack = Min. Data Arrival Time – Max. Data Required Time (Hold)

- Positive slack – timing requirement met
- Negative slack – timing requirement not met



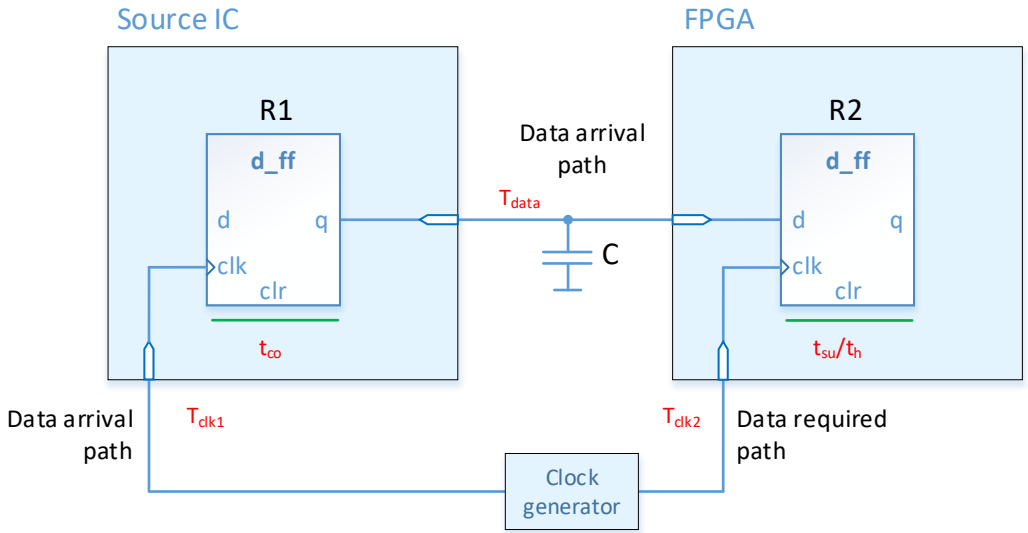


source: https://www.intel.com/content/www/us/en/docs/programmable/683068/18-1/timing-analysis-introduction.html

Synchronous Analysis

I/O Analysis (Common Clock Source):

- Analyze I/O performance in a synchronous design using the same slack equations from the prior slides
- Must include external device & printed circuit board timing parameters

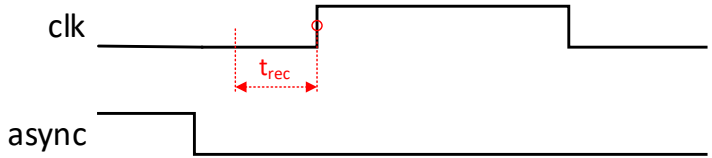


source: https://www.intel.com/content/www/us/en/docs/programmable/683068/18-1/timing-analysis-introduction.html

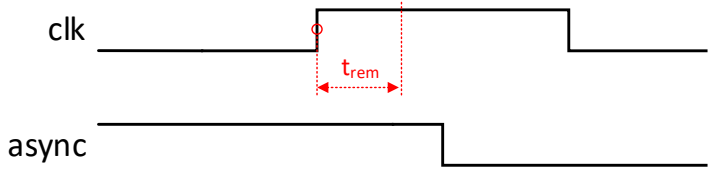
Asynchronous Analysis

- Check if:
  - Clock and asynchronous signals (i.e. clear) do not arrive at register input at the same time
  - Register output does not become unstable
  - Registers come of out asynchronous condition at the same time and in a known state

- Target timing parameters checked
  - Recovery time (t_{rec})
    minimum time an asynchronous signal must be de-asserted before clock edge



  - Removal time (t_{rem})
    minimum time an asynchronous signal must be de-asserted after clock edge



source: https://www.intel.com/content/www/us/en/docs/programmable/683068/18-1/timing-analysis-introduction.html

Asynchronous Analysis

Recovery slack = Min. Required Time (Recovery) – Max. Arrival Time

Removal slack = Min. Arrival Time – Max. Required Time (Removal)

- Positive slack – timing requirement met
- Negative slack – timing requirement not met

source: https://www.intel.com/content/www/us/en/docs/programmable/683068/18-1/timing-analysis-introduction.html

Timing Models

Intel Quartus Prime models device timing at multiple process voltage temperature (PVT)

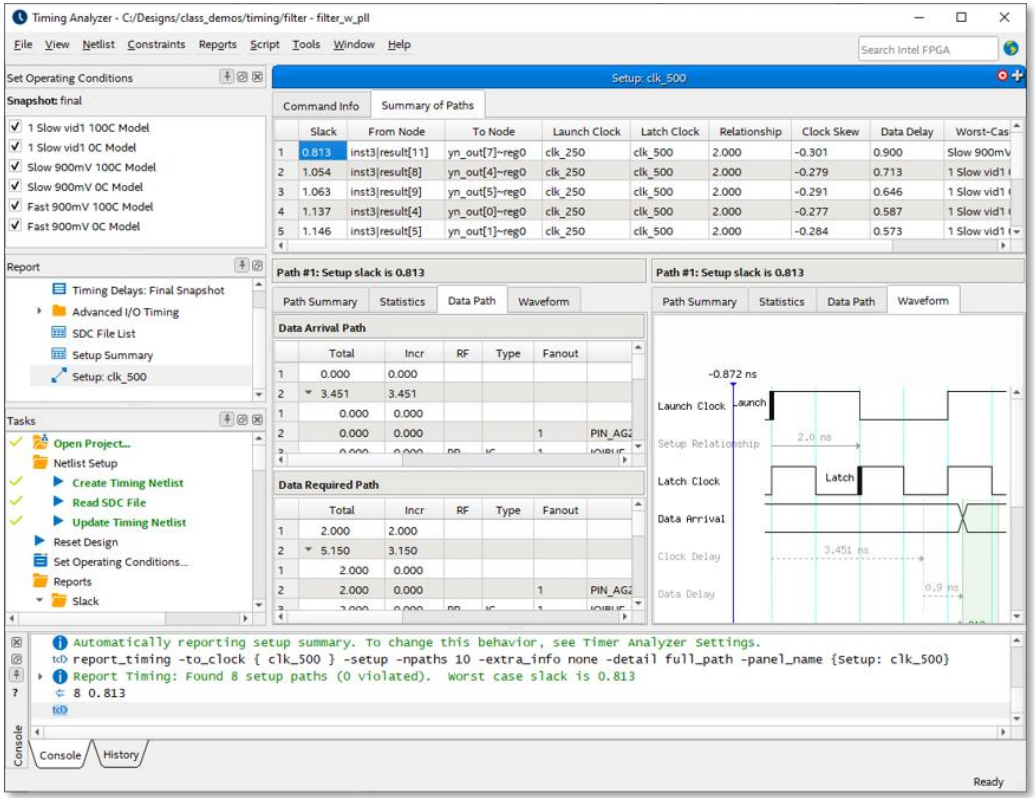
- Slow corner model
  - Indicates slowest possible performance for any single path
  - Timing for slowest device at maximum operating temperature and lowest voltage (VCCMIN)
- Fast corner model
  - Indicates fastest possible performance for any single path
  - Timing for fastest device at minimum operating temperature and highest voltage (VCCMAX)
- 2nd slow and 2nd fast models
  - Slow timing at minimum operating temperature
  - Fast timing at maximum operating temperature

- ! Ensure setup timing is met in the slow models
- ! Ensure hold timing is met in fast model (essential for source synchronous interface)

source: https://www.intel.com/content/www/us/en/docs/programmable/683068/18-1/timing-analysis-introduction.html

Timing Analyzer GUI

- Timing analysis engine in Quartus Prime providing timing analysis solution for all level of experience
- Features:
  - Synopsys Design Constraints (SDC) support
  - Easy-to-use interface
    - Graphical constraint entry
    - Standard, on-the-fly reporting
  - Scripting fully supported



source: https://www.intel.com/content/www/us/en/docs/programmable/683492/18-1/specifying-timing-constraints-in-the-gui.html

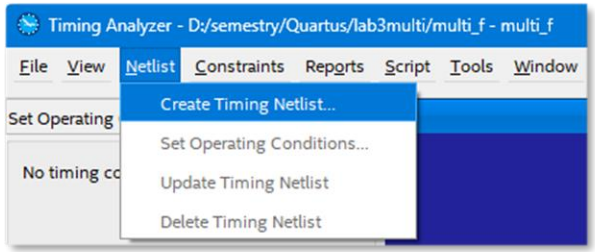
Timing Constraints

Constraining recommendations

- Ensure all design path are constrained
  - Constraints guide the Fitter during placement & routing design
  - Without guidance, Fitter can make incorrect optimization choice
  - Design paths must be constrained to fully analyze design
  - Timing Analyzer only performs slack analysis on constrained paths

Basis timing analysis flow

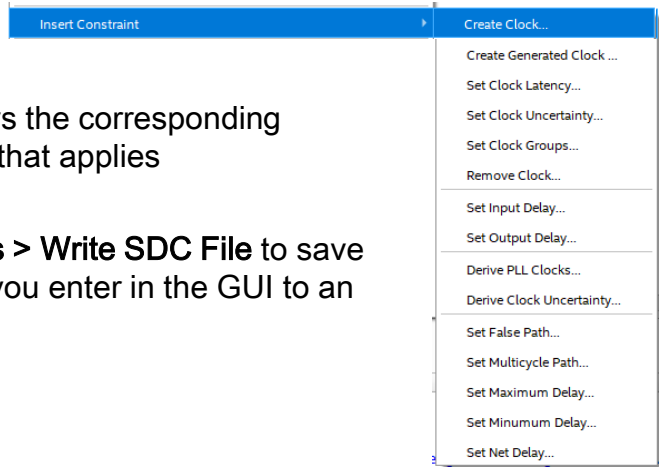
- Create a Project and run the Fitter
- Create Timing Netlist
- Specify timing constraints
- Run timing analysis
- Analyze results



source: https://www.intel.com/content/www/us/en/docs/programmable/683492/18-1/specifying-timing-constraints-in-the-gui.html

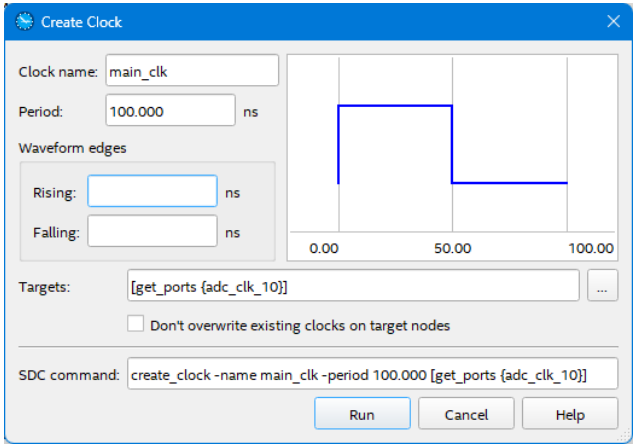
Timing Analyzer GUI

- Enter constraints in the Timing Analyzer GUI
  - Create clock constraint



- The GUI displays the corresponding SDC command that applies

- click **Constraints > Write SDC File** to save the constraints you enter in the GUI to an **.sdc** file

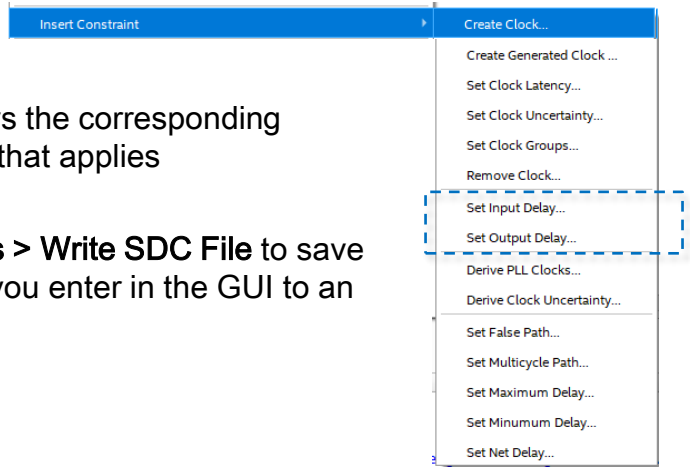



```
create_clock -name {main_clk} -period 100.000 -waveform { 0.000 50.000 } [get_ports {adc_clk_10}]
```

source: https://www.intel.com/content/www/us/en/docs/programmable/683492/18-1/specifying-timing-constraints-in-the-gui.html

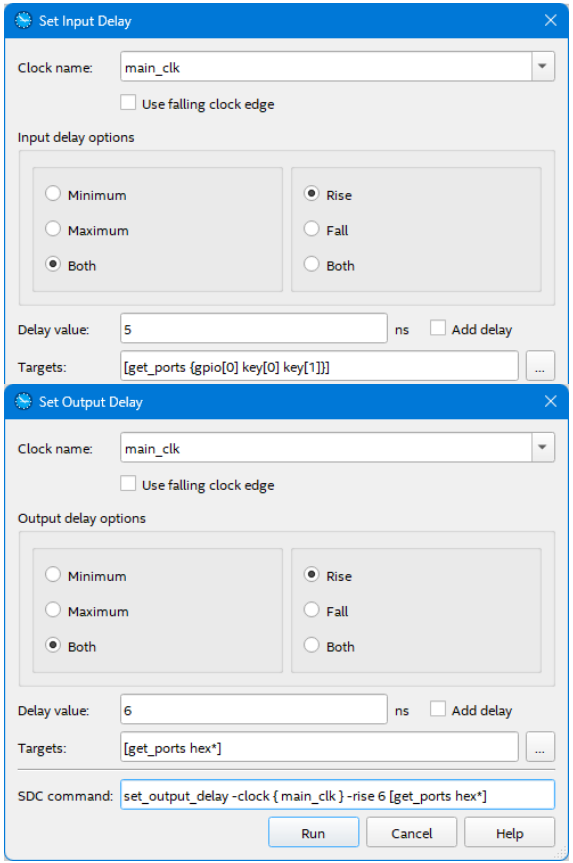
Timing Analyzer GUI

- Enter constraints in the Timing Analyzer GUI
  - Create input/output delay constraints



- The GUI displays the corresponding SDC command that applies

- click **Constraints > Write SDC File** to save the constraints you enter in the GUI to an **.sdc** file

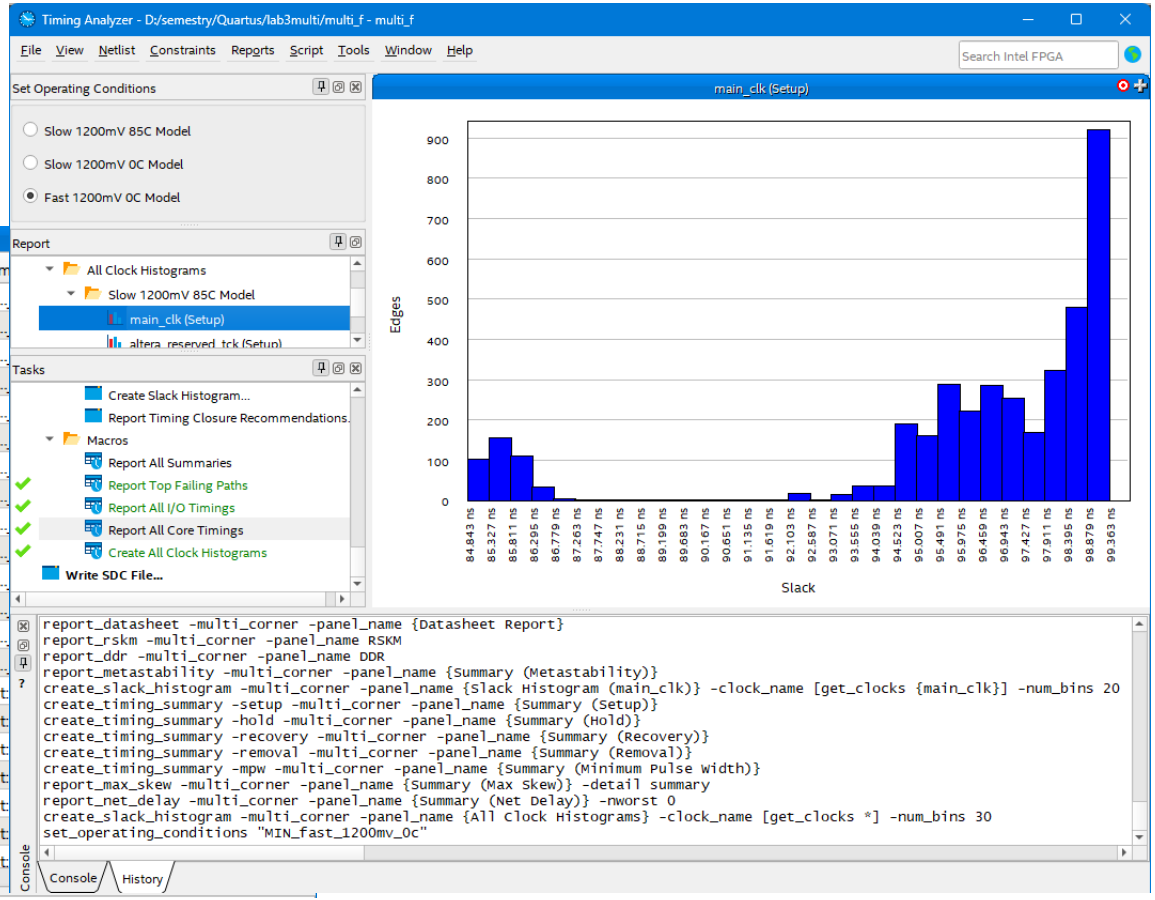



```
set_input_delay -clock { main_clk } -rise 5 [get_ports {gpio[0] key[0] key[1]}]
set_output_delay -clock { main_clk } -rise 6 [get_ports hex*]
```

source: https://www.intel.com/content/www/us/en/docs/programmable/683492/18-1/specifying-timing-constraints-in-the-gui.html

Timing Analyzer GUI

- Analysis of time results



Multi Corner Summary (2/3 corners)

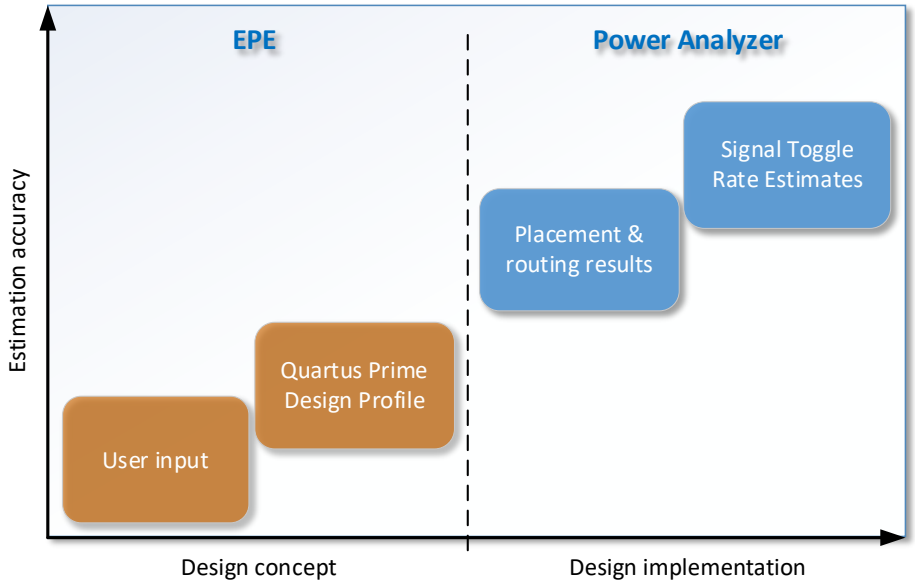
	Corner	Slack	From
951	Slow 1200mV 85C Model	95.502	sld_signaltap:auto_signaltap_0|sld_signaltap...
952	Slow 1200mV 85C Model	95.502	sld_signaltap:auto_signaltap_0|sld_signaltap...
953	Slow 1200mV 85C Model	95.502	sld_signaltap:auto_signaltap_0|sld_signaltap...
954	Slow 1200mV 85C Model	95.502	sld_signaltap:auto_signaltap_0|sld_signaltap...
955	Slow 1200mV 85C Model	95.502	sld_signaltap:auto_signaltap_0|sld_signaltap...
956	Slow 1200mV 85C Model	95.502	sld_signaltap:auto_signaltap_0|sld_signaltap...
957	Slow 1200mV 85C Model	95.502	sld_signaltap:auto_signaltap_0|sld_signaltap...
958	Slow 1200mV 85C Model	95.502	sld_signaltap:auto_signaltap_0|sld_signaltap...
959	Slow 1200mV 85C Model	95.502	sld_signaltap:auto_signaltap_0|sld_signaltap...
960	Slow 1200mV 85C Model	95.502	sld_signaltap:auto_signaltap_0|sld_signaltap...
961	Slow 1200mV 85C Model	95.502	sld_signaltap:auto_signaltap_0|sld_signaltap...
962	Slow 1200mV 85C Model	95.502	sld_signaltap:auto_signaltap_0|sld_signaltap...
963	Slow 1200mV 85C Model	95.502	sld_signaltap:auto_signaltap_0|sld_signaltap...
964	Slow 1200mV 85C Model	95.502	sld_signaltap:auto_signaltap_0|sld_signaltap...
965	Slow 1200mV 85C Model	95.507	sld_hub:auto_hub|alt_sld_fab_with_jtag_input
966	Slow 1200mV 85C Model	95.509	sld_hub:auto_hub|alt_sld_fab_with_jtag_input
967	Slow 1200mV 85C Model	95.509	sld_hub:auto_hub|alt_sld_fab_with_jtag_input
968	Slow 1200mV 85C Model	95.509	sld_hub:auto_hub|alt_sld_fab_with_jtag_input
969	Slow 1200mV 85C Model	95.509	sld_hub:auto_hub|alt_sld_fab_with_jtag_input
970	Slow 1200mV 85C Model	95.509	sld_hub:auto_hub|alt_sld_fab_with_jtag_input
971	Slow 1200mV 85C Model	95.514	sld_hub:auto_hub|alt_sld_fab_with_jtag_input

source: https://www.intel.com/content/www/us/en/docs/programmable/683492/18-1/specifying-timing-constraints-in-the-gui.html

Power Analysis Tools

Quartus Prime provides tools to estimate power consumption in a FPGA design at different stages of the design process

- Early Power Estimator (EPE) to estimate the power consumption before compilation
  - Reliable estimate before design development
  - Spreadsheet-based “what-if” analysis
- Power Analyzer to estimate power consumption for a post-fit design
  - Detailed design power estimation
  - Use actual design place & route and logic configuration
  - High accuracy

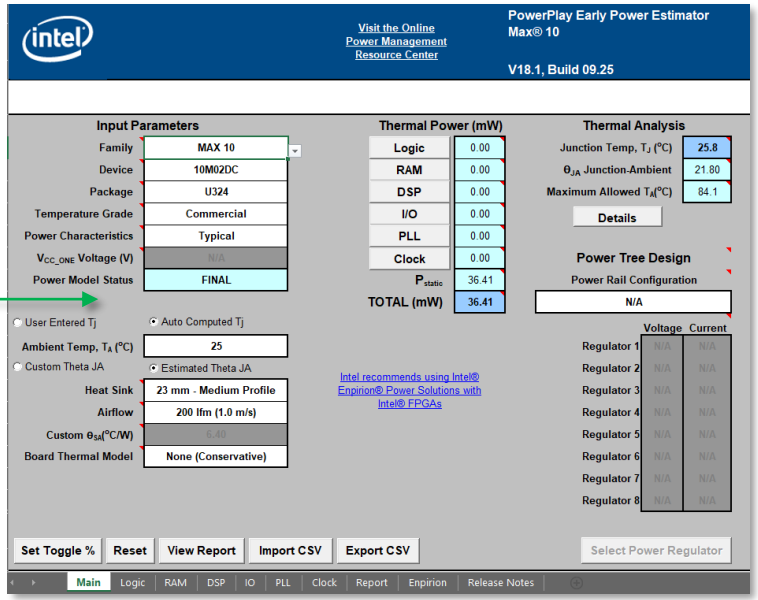


source: https://www.intel.com/content/www/us/en/docs/programmable/683506/18-1/power-analysis.html

Early Power Estimator

- For quick “what-if” analysis
- Enter design data manually before design
- Generate design inputs in Quartus Prime for more accurate analysis (<>_early_power.csv)

Input:
- Number of registers
- Transition rates
- Clocks and frequency
- IO and memory
- DSP and PLL
- Temperature, air flow, heat sinking

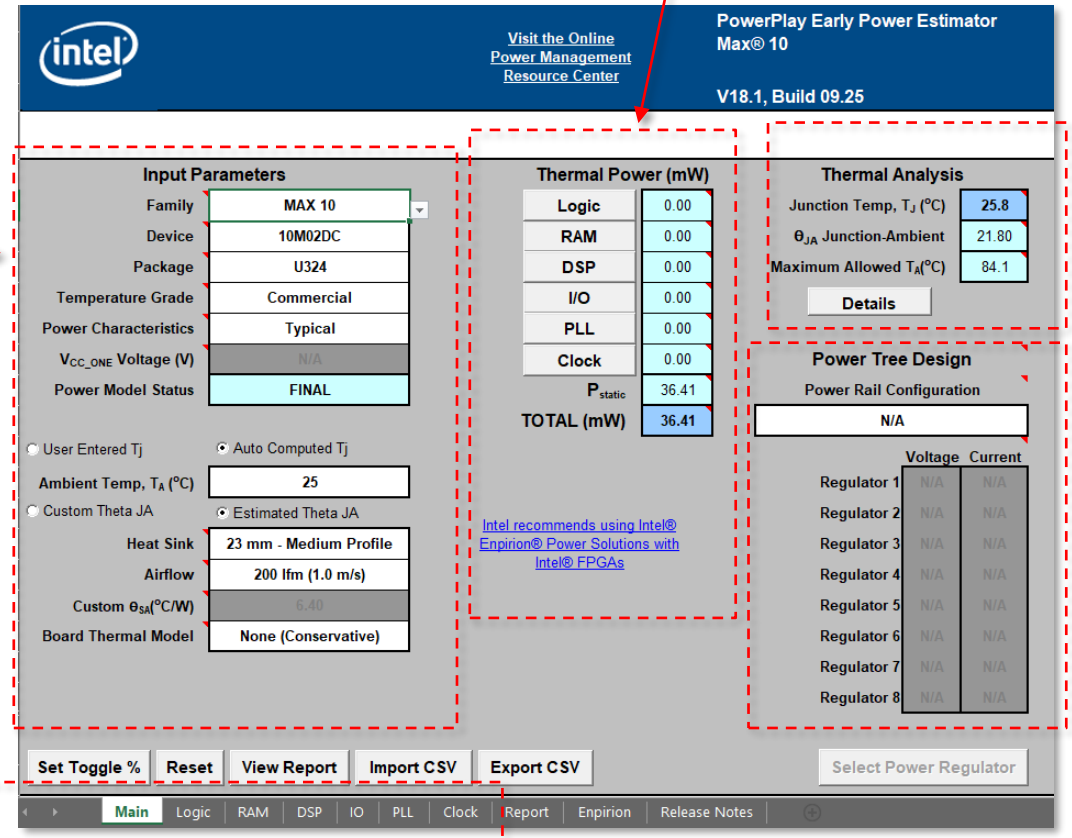


Download form Intel website

- https://www.intel.com/content/dam/altera-www/global/en_US/others/support/devices/estimator/max-10-estimator/max10_epe.xls

source: https://www.intel.com/content/www/us/en/docs/programmable/683506/18-1/power-analysis.html

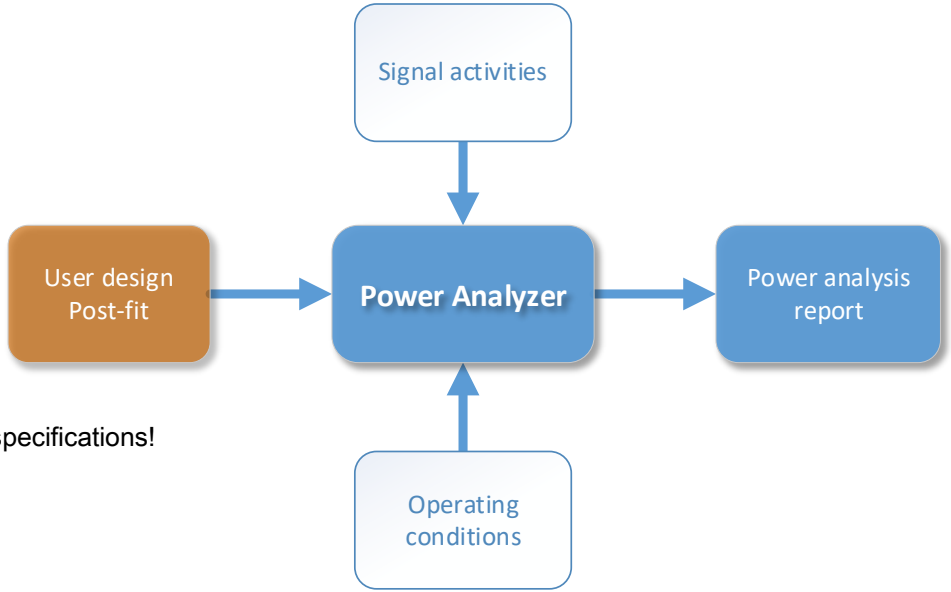
Early Power Estimator



source: https://www.intel.com/content/www/us/en/docs/programmable/683506/18-1/power-analysis.html

Power Analyzer

- Data inputs:
  - Post-fit design
  - Clock requirements
  - Signal activity defaults
  - Environmental conditions
  - Register transfer level (RTL) simulation results (optional)
  - Post-fit simulation results (optional)
  - Signal activities per node or entity (optional)
- Data outputs:
  - Total thermal power
  - Thermal static power
  - Thermal dynamic power
  - Thermal I/O power
  - Thermal power by design hierarchy
  - Thermal power by block type
  - Thermal power dissipation by clock domain
  - Off-chip (non-thermal) power dissipation
  - Device supply currents



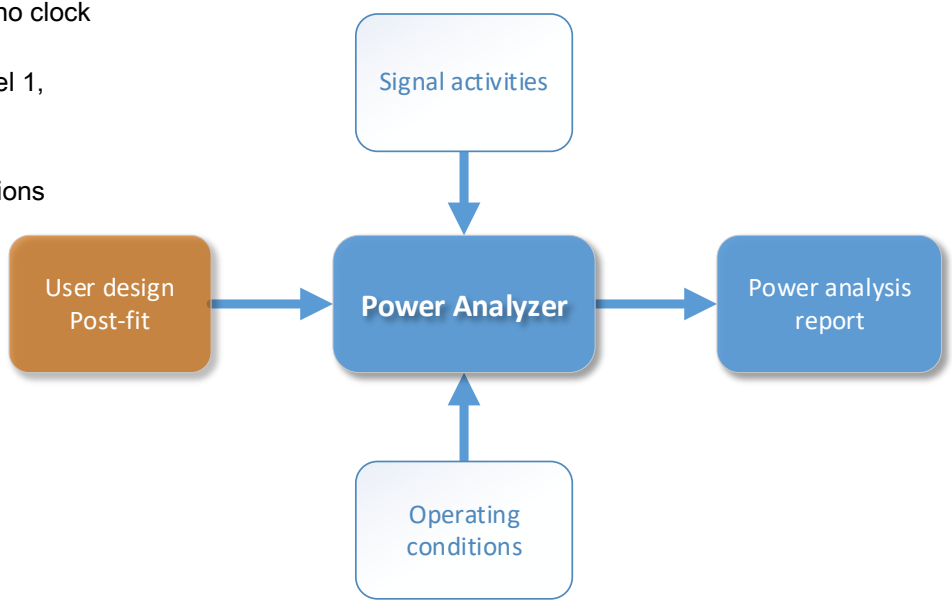
Note: Do not use the results of the Power Analyzer as design specifications!

source: https://www.intel.com/content/www/us/en/docs/programmable/683506/18-1/power-analysis.html

Power Analyzer

Signal activity inputs:

- Value change dump file (.**vcd**)
  - Captures signal toggling information
  - Created by Questa simulator
- Assignments applied to entities in Assignment Editor or script
  - **Power Toggle Rate**:
    absolute toggle rate in transitions/s
  - **Power Toggle Rate Percentage**:
    toggle rate relative to clock driving node; ignore if no clock
  - **Power Static Probability**:
    fraction of analysis time when signal is at logic level 1,
    ranges between 0 and 1
- Default toggle rate (12,5%)
  - Percentage of clock periods in which signal transitions
  - Absolute number of transitions per second



source: https://www.intel.com/content/www/us/en/docs/programmable/683506/18-1/power-analysis.html

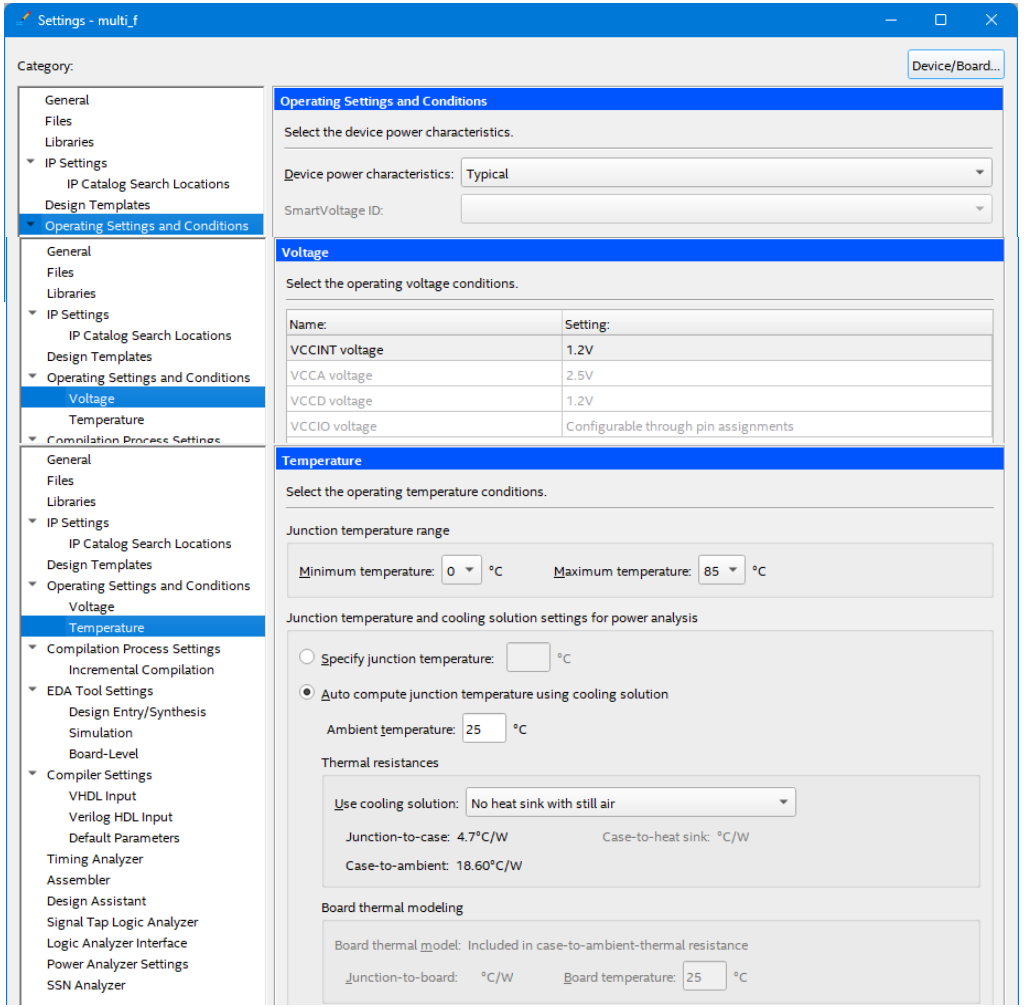
Power Analyzer

Operating conditions

Specifies the device characteristics to be used during power estimation. Estimates are based on average power consumed by typical silicon at nominal operating conditions.

- **Typical** – nominal operating conditions for device
- **Maximum** – worst-case conditions, use for power budgeting and supply design (timing analysis always uses worst-case!)

- **Voltage** settings
  (options differ depending on device selected)
- Operating **temperature**
  (package temperature and cooling solution)

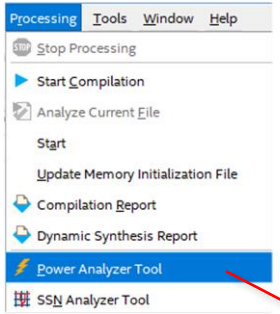


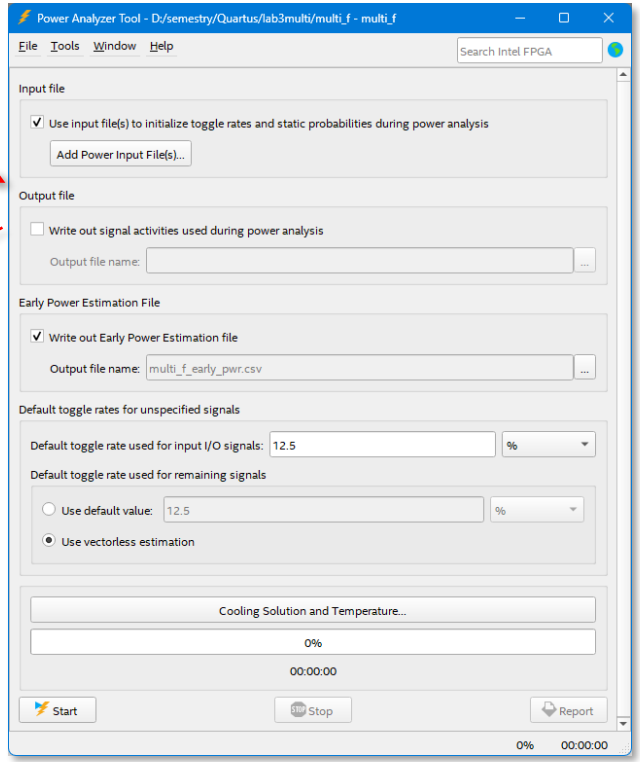
source: https://www.intel.com/content/www/us/en/docs/programmable/683506/18-1/power-analysis.html

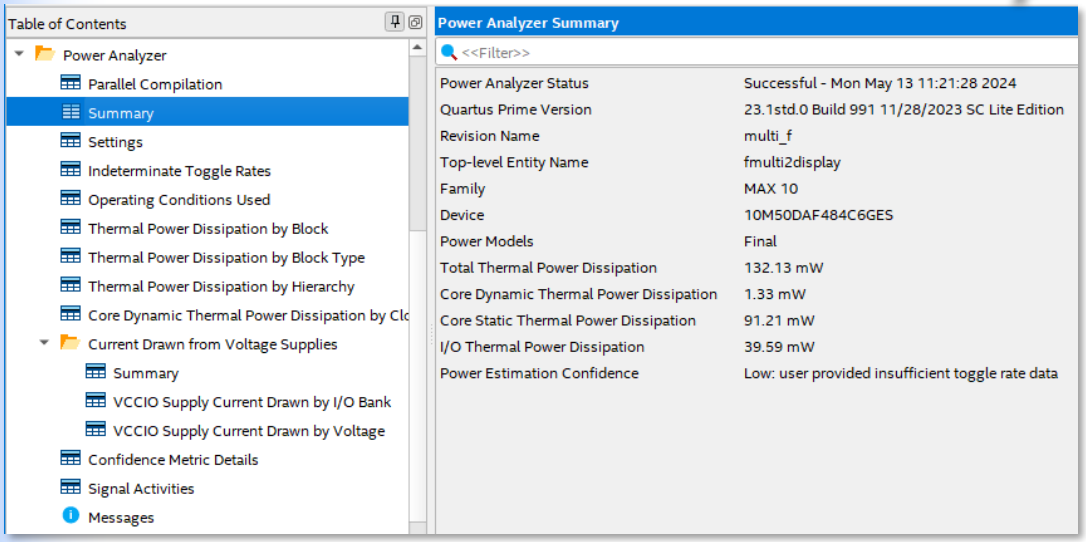
Power Analyzer

Running the Analyzer

- **Processing** menu -> **Power Analyzer Tool**
  - or enable running during full compilation
  - and viewing report







source: https://www.intel.com/content/www/us/en/docs/programmable/683506/18-1/power-analysis.html