Digital synthesis is the automated process that translates a high-level design (RTL) into a detailed gate-level netlist. This translation relies on a standard cell library, which contains various "flavors" of each logic gate, primarily differing in speed. This variety is crucial for meeting critical timing requirements: **fast cells** are used to reduce delays in long logic paths to meet performance goals (setup time), while **slow cells** are strategically used to add delay and prevent data corruption in paths that are too fast (hold time).

The physical difference between these cells lies in their transistor sizes. Faster cells use wider transistors to source more current, allowing them to drive capacitive loads more quickly. However, this speed comes at the direct cost of increased power consumption and a larger silicon area. Slower cells, with their narrower transistors, are far more efficient in both power and area but have a higher delay.

This creates a fundamental design trade-off. A circuit with too many fast cells will be power-hungry and large, while one with too many slow cells will fail to meet performance targets. To manage this balance, designers provide the synthesis tool with "constraints," such as the target clock frequency and area limits. The tool uses these guidelines to select an optimal mix of fast and slow cells, creating a final design that is both functional and efficient.

SYNTHESIS Steps  
  
read\_liberty -lib sky130\_fd\_sc\_hd\_\_tt\_025C\_1v80.lib- Loads the SKY130 standard cell library for technology mapping.

read\_verilog ../verilog\_files/good\_mux.v - Reads the high-level RTL Verilog design file for the multiplexer.

synth -top good\_mux - Performs a generic synthesis, converting RTL into basic logic gates

abc -liberty sky130\_fd\_sc\_hd\_\_tt\_025C\_1v80.lib - Maps the generic logic to specific, optimal cells from the loaded SKY130 library.

Show- Displays a graphical schematic of the final, technology-mapped circuit.

write\_verilog good\_mux\_netlist.v - Saves the synthesized gate-level netlist to a file (with extra tool attributes).

!gvim good\_mux\_netlist.v - Opens the generated netlist file in the GVim editor for viewing.

write\_verilog -noattr good\_mux\_netlist.v- Saves a cleaner version of the netlist, removing tool-specific attributes.

!gvim good\_mux\_netlist.v- Re-opens the cleaned-up netlist in GVim to see the final version.