Digital Logic
Design with FPGA

Design Flow with Quartus

Foundations

Outline

- Modeling digital systems
- Typical design flow with HDL
- Intel PLD design flow
- Quartus Prime software
  - Project flow
  - Design entry
  - Compilation tools
  - Verification of results

Digital system vs. Digital device

- Digital circuit…
  equipment that performs functions using an appropriate structure of electronic components

- Digital system ...
  a set of digital circuits, contains at least one programmable processor, performs a significant part of its functions by means of software executed by this processor

Embedded system

An embedded system is a microprocessor system built into any technical device, performing specific control, measurement, signal processing functions, etc.,

- Basic elements of an embedded system:
  - Microcontroller / microprocessor
  - Application-specific hardware (ASIC/FPGA)
  - Application software
  - Real-time operating system (RTOS)
- New design considerations:
  - system level design,
  - co-creation of hardware and software (H/S Codesign)

Evolution of ASIC to SoC/SoPC

Technological possibilities allow you to fit in one integrated circuit:

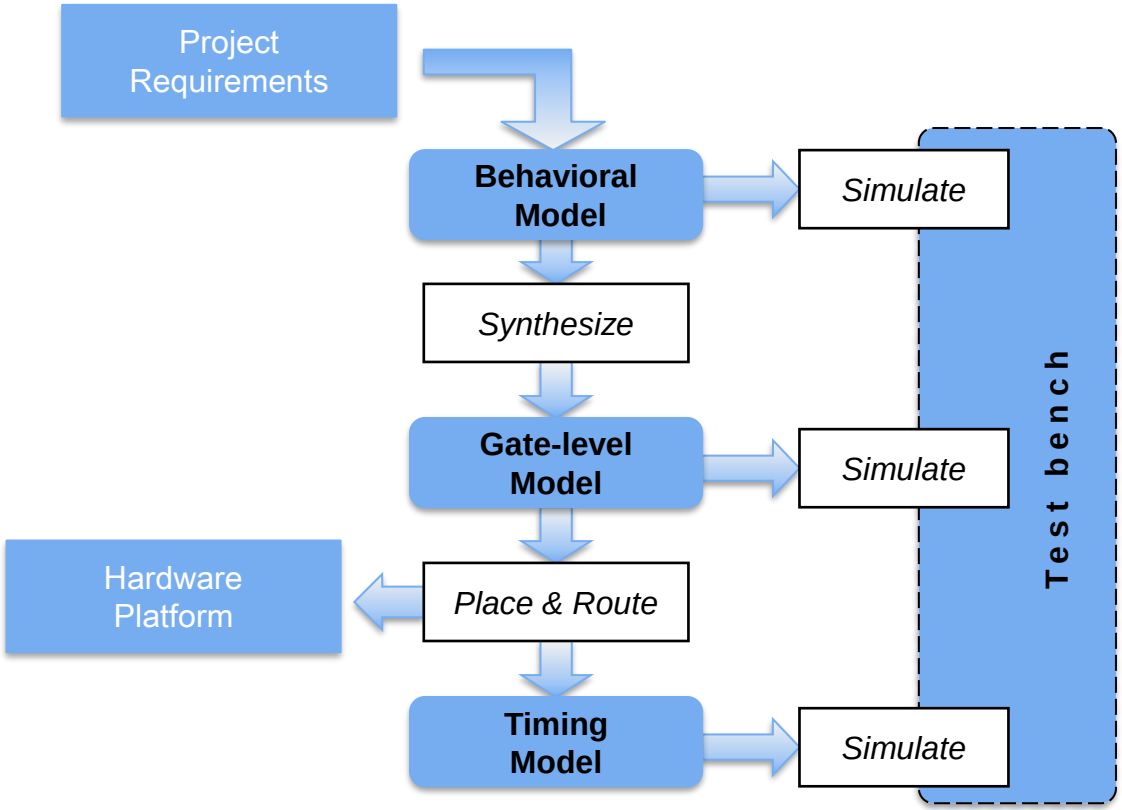
- processor (several processors, e.g. signal processor, microcontroller, many cores with different parameters)
- memory (ROM & SRAM, in specific processes Flash memory, "e"-DRAM)
- standard communication blocks (e.g. I2C or USB controller)
- application-specific logic subcircuits (ASSP)
- mixed and analog circuits (e.g. ADC and DAC converters)
- mechanical microsystems (e.g. MEMS sensors)

Programmable devices also now contain logical resources enabling the implementation of systems in a single package (including embedded processors) -> SoPC, PSoC, APSoC

Modeling Digital Systems

- Simulation and modeling at the system level
  - performance analysis
- Behavior specification at the algorithmic level
  - preliminary functional verification of algorithms
  - division into hardware and software
  - high-level synthesis
- Simulation behavioral models of standard elements
- Functional simulations at the system/package level
  - full
  - bus
- Synthesizable models at the RTL (register transfer level)
  - full functional specification of the project
- Model of the system environment (testbench)
- Simulation models of library cells from integrated circuit manufacturers (VITAL standard) –
  - time verification of ASIC/FPGA systems

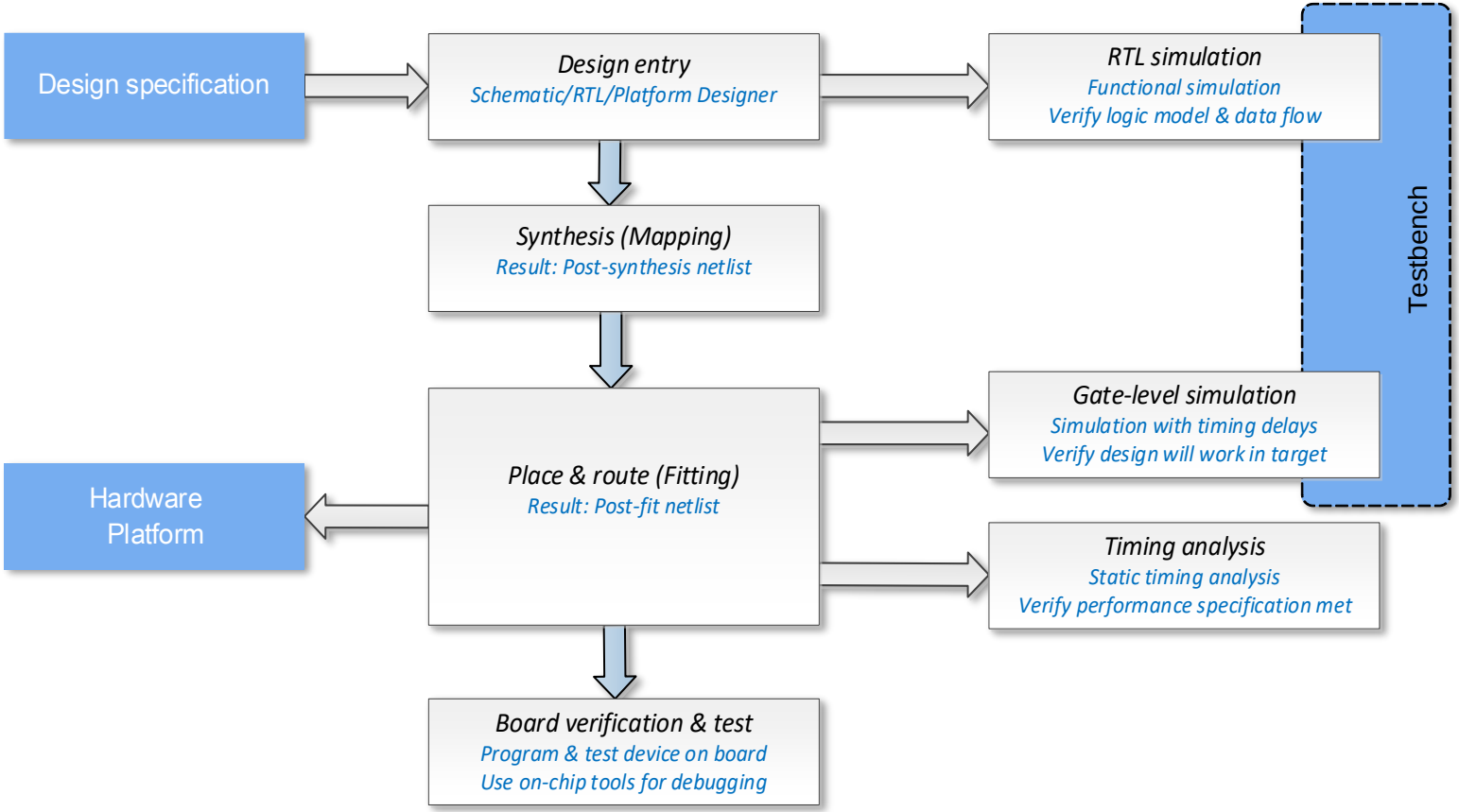
Typical design flow with HDL



Design stages

- Specification / design entry
- Synthesis
  - implementation of functions based on component (library) elements
  - optimization (e.g. logical minimization)
  - generation (physical implementation of the structure)
- Analysis
  - simulation – validation, regression tests
  - formal verification (equivalence of two representations)
  - verification of design rules
- Implementation
  - from SSI / MSI / LSI / VLSI catalog items (memory / ASSP)
  - programmable systems (FPGA / CPLD)
  - specialized systems (ASIC)
  - embedded systems (traditional and integrated - SoC)

Intel PLD Design Flow



Intel Quartus Prime

- Fully-integrated development tool
  - Multiple design entry methods
  - Logic synthesis
  - Place & route
  - Device programming
- Simulation
  - Supports standard HDL simulation tools
  - Includes ModelSim(Questa)-Intel FPGA Starter Edition tool
  - Optional upgrade to ModelSim(Questa)-Intel FPGA Edition tool

Intel® Quartus® Prime
Design Software

Lite Edition (LE)

Standard Edition (SE)

Pro Edition (PE)

source: https://www.intel.com/content/www/us/en/support/programmable/support-resources/design-software/fpga-development-tools-support.html

Quartus Prime Software

- Lite Edition (LE)
  - Supported families:
    Cyclone V, IV
    MAX 10, V, II
  - No license file required
- Standard Edition (SE)
  - Supported families:
    Stratix V, IV
    Arria 10, V, II
    Cyclone V, IV
    MAX 10, V, II
  - License file required
- Pro Edition (PE)
  - Supported families:
    Stratix 10
    Arria 10
  - License file required

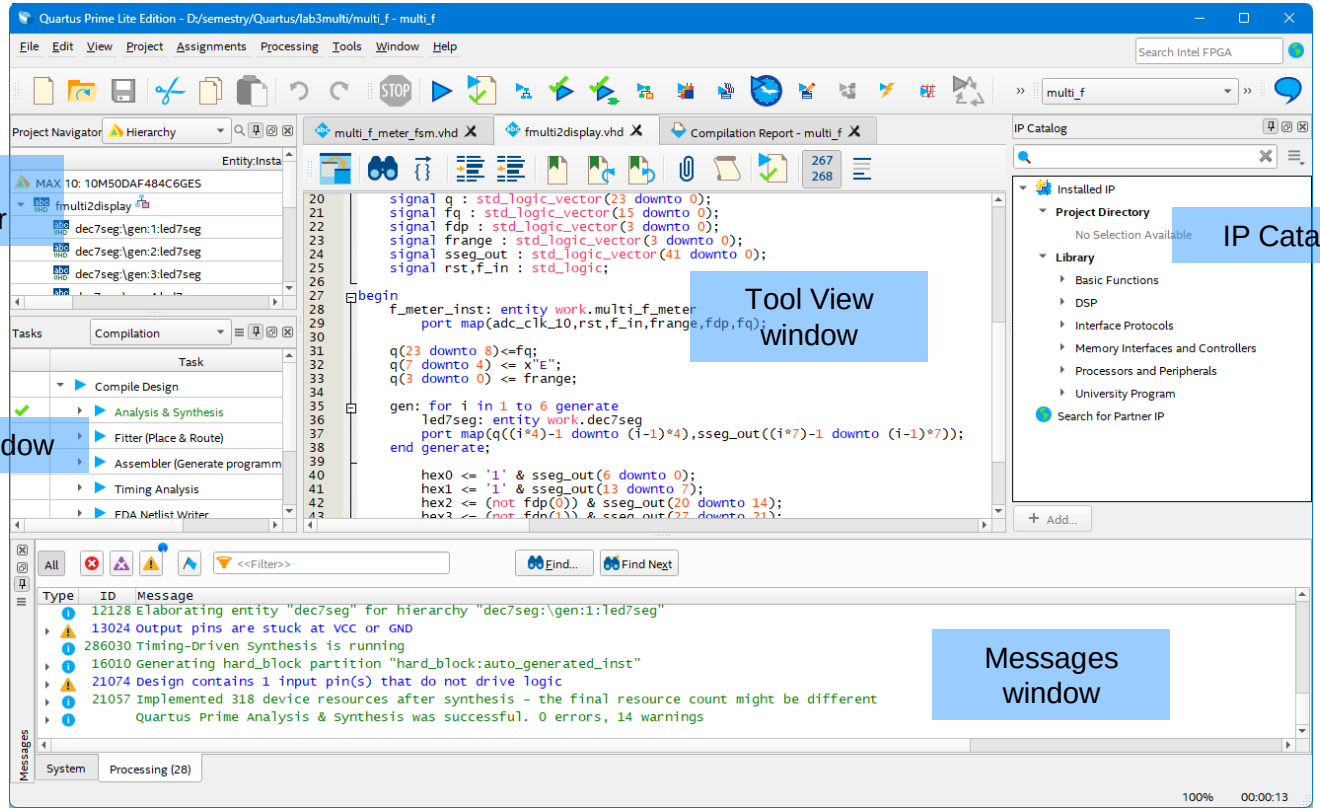
Lite Edition (LE)

Standard Edition (SE)

Pro Edition (PE)

source: https://www.intel.com/content/www/us/en/support/programmable/support-resources/design-software/fpga-development-tools-support.html

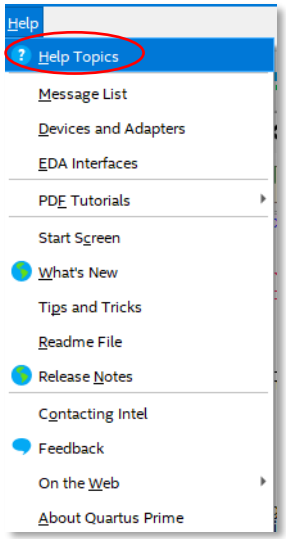
Quartus Prime GUI

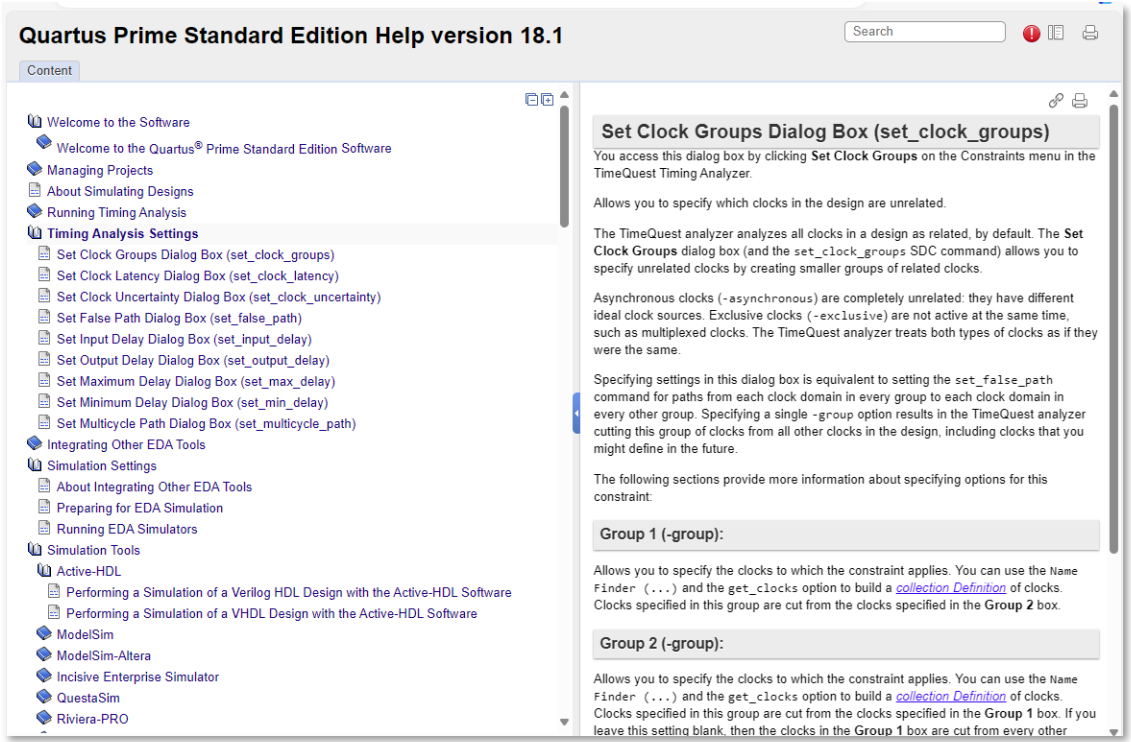


Digital Logic Design with FPGA

source: https://www.intel.com/content/www/us/en/support/programmable/support-resources/design-software/fpga-development-tools-support.html

Quartus Prime Help System

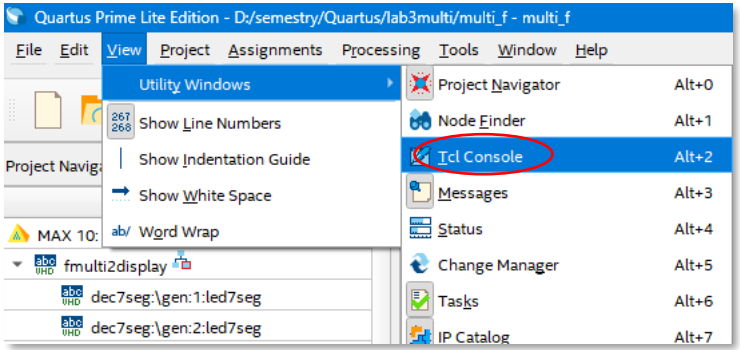


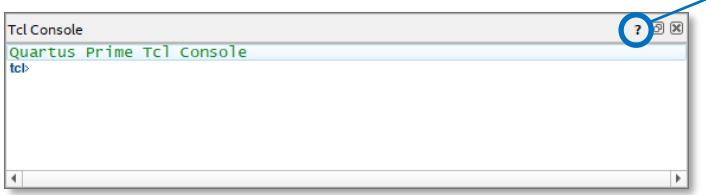


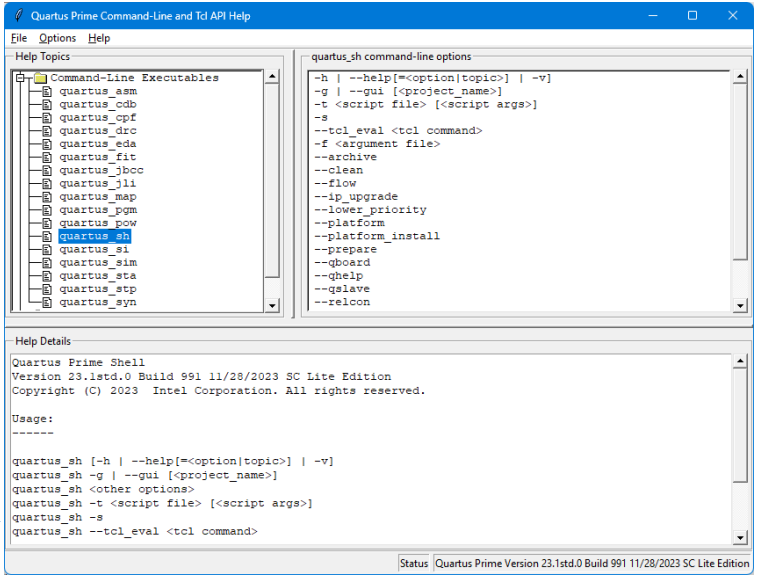
source: https://www.intel.com/content/www/us/en/support/programmable/support-resources/design-software/fpga-development-tools-support.html

Tcl Console

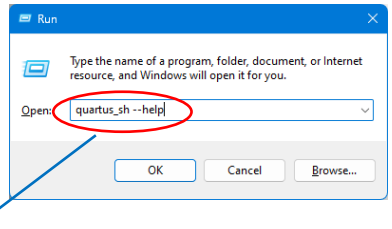
- Enter and execute Tcl commands directly in the GUI
  View menu –> **Utility Windows** -> **Tcl Console**







- Execute from command-line using Tcl shell
  - `quartus_sh -s`

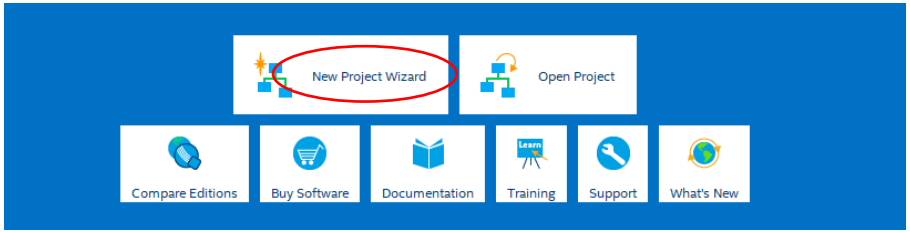


C:\<*install dir*>\quartus\bin64\quartus_sh --qhelp

source: https://www.intel.com/content/www/us/en/support/programmable/support-resources/design-software/fpga-development-tools-support.html

Quartus Prime Design Software

- Quartus project
  - Collection of related design files & libraries
  - Must have a designated top-level entity
  - Target a single device
  - Store settings in settings file (**.qsf**)
  - Compiled netlist information stored in **qdb** folder in project directory
- Create new projects with **New Project Wizard**

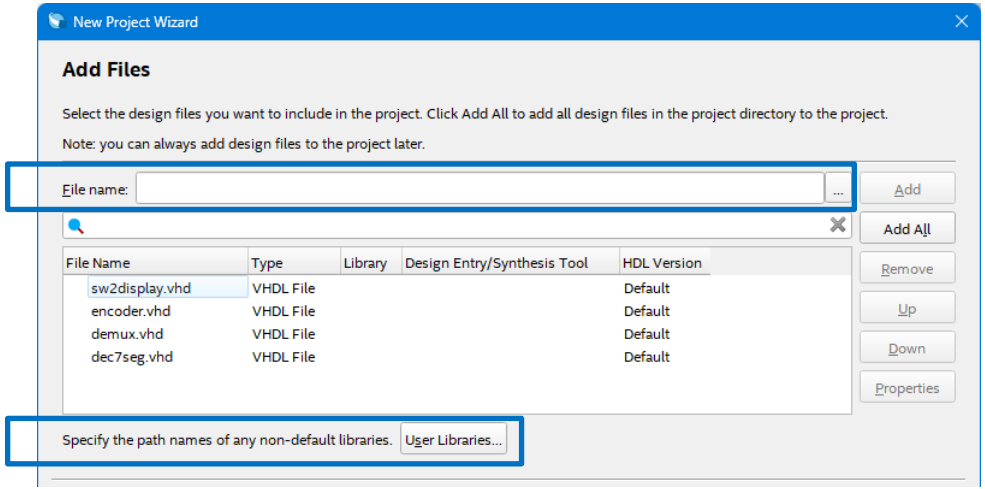


- Can be created using Tcl scripts

Tcl: `roject_new <project_name>`

Add Files in Project

- Add design files
  - VHDL
  - Verilog
  - SystemVerilog
  - EDIF
  - VQM
  - Intel® Quartus® Prime software IP
  - Platform Designer




```
Tcl: set_global_assignment -name VHDL_FILE <filename.vhd>
```

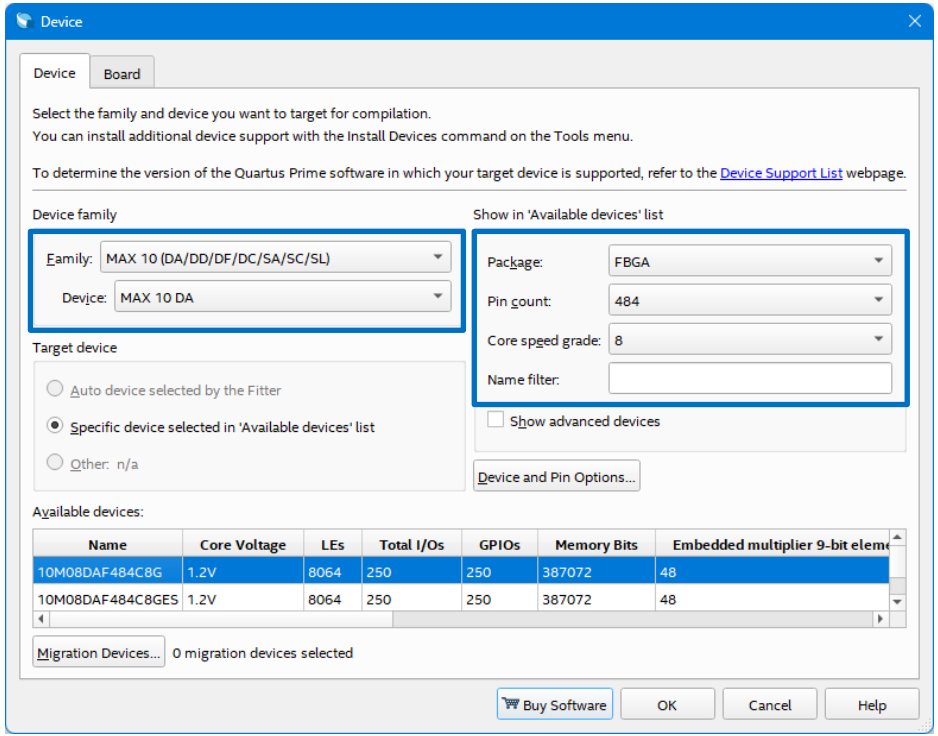
- Add library paths
  - User libraries
  - Intel® FPGA Intellectual Property (IP) library
  - Pre-compiled VHDL packages

```
Tcl: set_global_assignment -name USER_LIBRARIES <library_path_name>
```

source: https://www.intel.com/content/www/us/en/support/programmable/support-resources/design-software/fpga-development-tools-support.html

Device & Family Selection

- Choose device family & family category

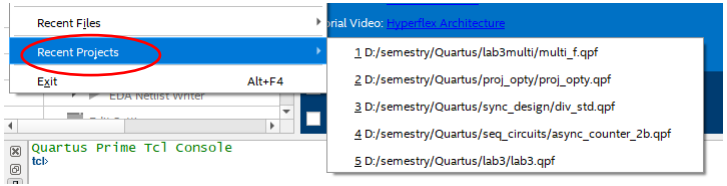


Tcl: `set_global_assignment -name FAMILY "device family name"`
Tcl: `set_global_assignment -name DEVICE <part_number>`

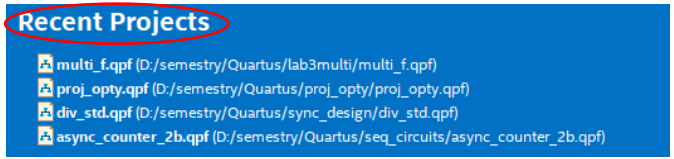
source: https://www.intel.com/content/www/us/en/support/programmable/support-resources/design-software/fpga-development-tools-support.html

Opening a Project

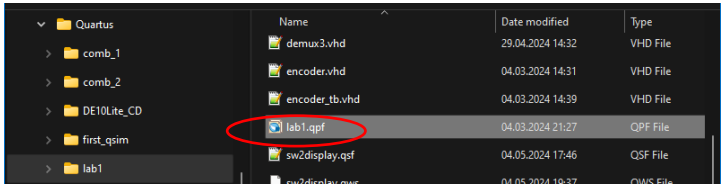
- From File menu



- From Recent Project



- Double-click .qpf file



- Tcl script

```
Tcl: project_open <project_name>
```


source: https://www.intel.com/content/www/us/en/support/programmable/support-resources/design-software/fpga-development-tools-support.html

Quartus Prime Project Files

- Project File (.**qpf**)
  - Intel Quartus Prime software version
  - Time stamp
  - Active revision(s)

```
26  QUARTUS_VERSION = "23.1"
27  DATE = "21:27:21  March 04, 2024"
28
29  # Revisions
30
31  PROJECT_REVISION = "sw2display"
```

- Defaults File (.**qdf**)
  - Stores project setting & assignment defaults for new project revisions
  - <revision_name>_assignment_defaults.qdf
- Settings File (.**qsf**)
  - Stores all settings & assignments except timing
  - Uses Tcl syntax
  - Can be edited manually by user
- Synopsys Design Constraints (.**sdc**)
  - Contains timing constraints
- **qdb** folder
  - Contains compiled design information
- **output_files** folder (customize location/name in project settings)
  - Generated compilation report files
  - Programming files generated by the Assembler

source: https://www.intel.com/content/www/us/en/support/programmable/support-resources/design-software/fpga-development-tools-support.html

Constraint Files & Assignment Priority

- Settings File (.**qsf**)
  - Highest priority
  - Assignments always used from here first
- Revision-specific .**qdf** file located in project directory
  - *<revision_name>_assignment_defaults.qdf*
  - Created automatically in the project directory when a revision is opened in another version of Quartus Prime software
- .**qdf** located in project directory
  - *assignment_defaults.qdf*
  - Created automatically in project directory when project archived & restored
- .**qdf** located in Intel Quartus Prime Design Software **bin64** directory
  - Lowest priority
  - Assignments only used if not found in higher priority files

source: https://www.intel.com/content/www/us/en/support/programmable/support-resources/design-software/fpga-development-tools-support.html

Project Archive & Restore

- Archive Creates 2 files
  - Compressed Quartus Prime Design Software Archive File (.**qar**)
    - Includes design files, .**qpf** file, & .**qsf** file(s)
    - Option to include databases
    - Creates local .**qdf** file for archive
  - Archive activity log (.**qarlog**)

Tcl: `project_archive <project_name>`

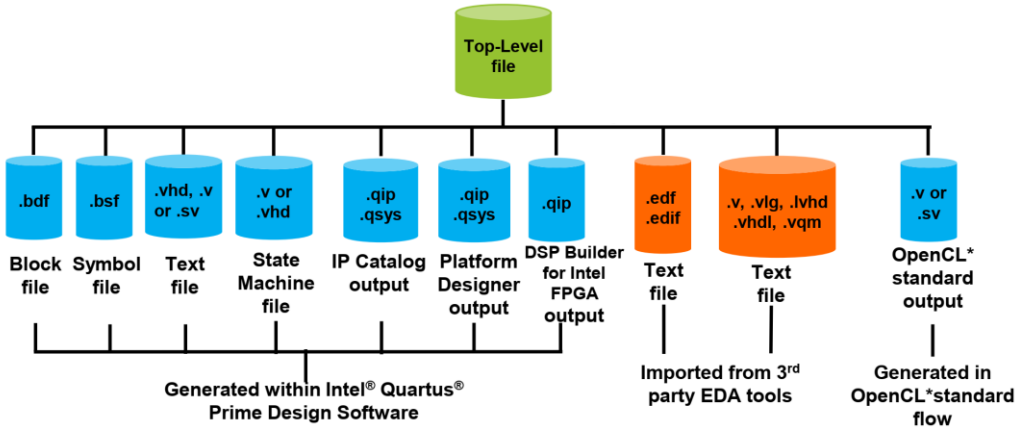
- Restore decompresses .**qar** into specified directory
- Paths/directory structures to referenced files/libraries *outside* project directory must also be restored
  - Recreated in restore location based on nearest common *parent directory*
  - Example of referenced file paths in restored project destination:
    - *<destination folder>**/drive/C/**<entire path to all files in project directory>*
    - <destination folder>***/drive/H/***<path to file(s) referenced on original H drive>

Tcl: `project_restore <archive_file>`

source: https://www.intel.com/content/www/us/en/support/programmable/support-resources/design-software/fpga-development-tools-support.html

Design entry methods

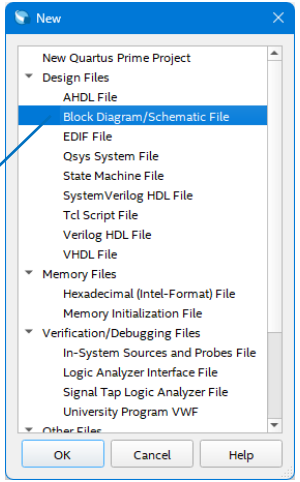
- Quartus Prime design entry
  - Text editor
    - VHDL
    - Verilog or SystemVerilog
  - Schematic editor
    - Block Diagram File
  - System editor
    - Platform Designer
  - State machine editor
    - HDL from state machine file
  - Memory editor
    - HEX
    - MIF
- 3rd-party EDA tools
  - EDIF 2 0 0
  - Verilog software Mapping (.vqm)

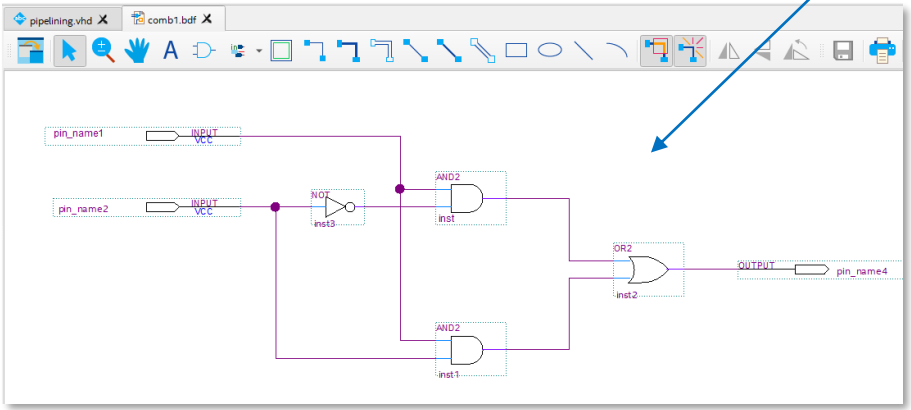


source: https://www.intel.com/content/www/us/en/support/programmable/support-resources/design-software/fpga-development-tools-support.html

Schematic Editor

- Full-featured schematic design capability
- Schematic Editor uses
  - Create simple test designs to understand the functionality of an Intel FPGA IP: PLL, LVDS I/O, memory, etc…
  - Create top-level schematic for easy viewing & connection
  - Convert between schematic .**bdf**, block symbol .**bsf**, and HDL files

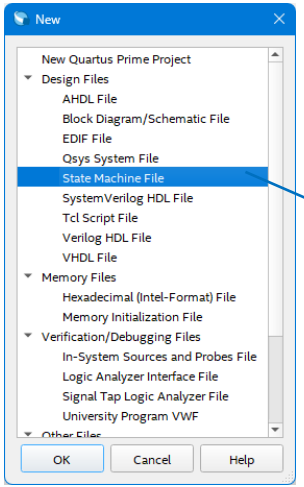


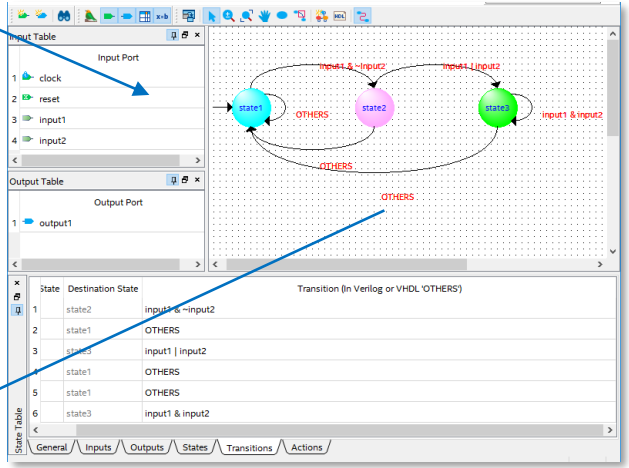


source: https://www.intel.com/content/www/us/en/support/programmable/support-resources/design-software/fpga-development-tools-support.html

State Machine Editor

- Create state machines in GUI
  - Manually by adding individual states, transitions, and output actions
  - Automatically with State Machine Wizard (Tools menu & toolbar)
- Generate state machine HDL code
  - VHDL, Verilog, SystemVerilog
  - Automatically added to project
  - Required for use





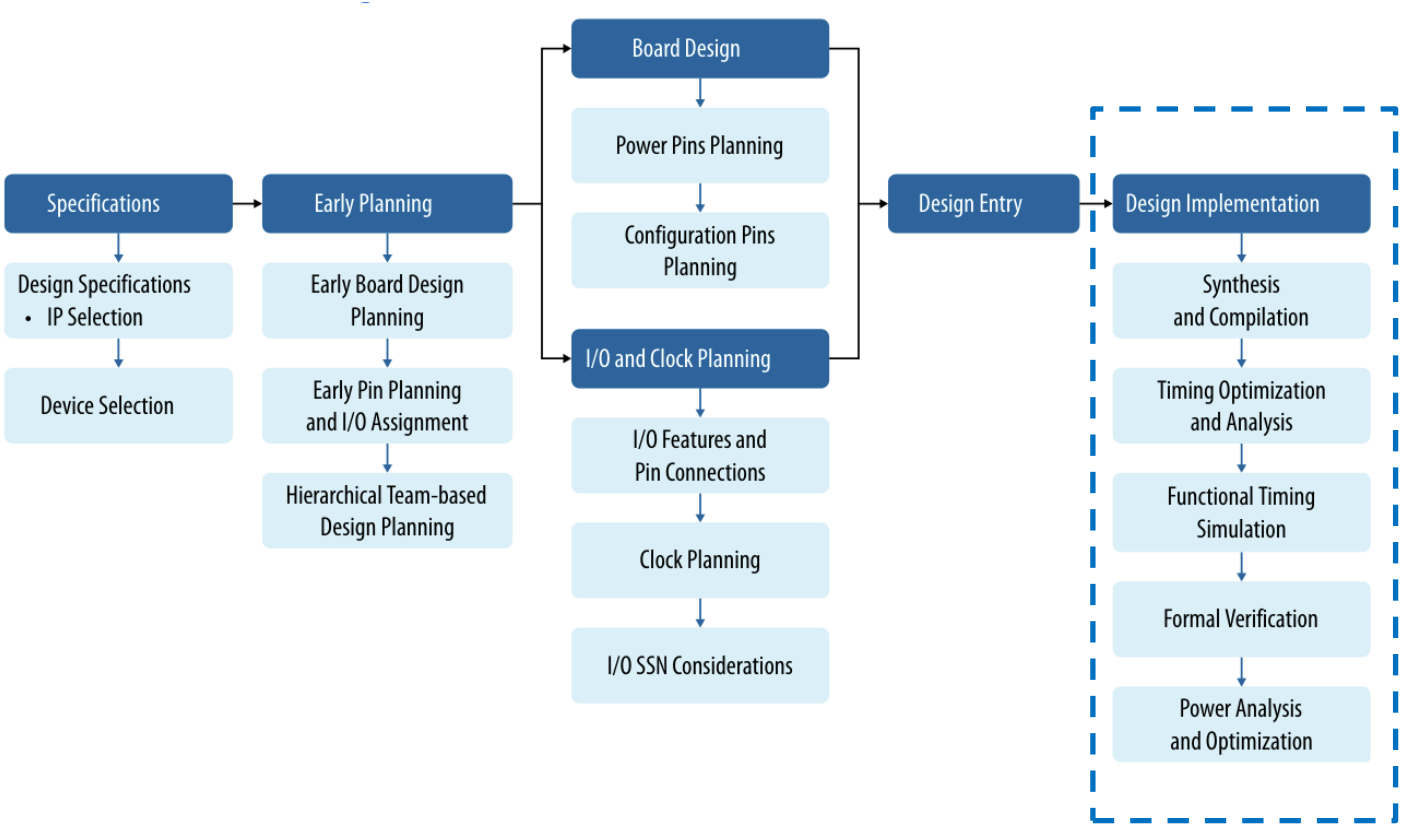

```
always @(posedge clock)
begin
    if (clock) begin
        fstate <= reg_fstate;
    end
end

always @(fstate or reset or input1 or input2)
begin
    if (reset) begin
        reg_fstate <= state1;
    end
    else begin
        case (fstate)
            state1: begin
                if ((input1 & ~(input2)))
                    reg_fstate <= state2;
                else
                    reg_fstate <= state1;
            end
            state2: begin
                if ((input1 | input2))
                    reg_fstate <= state3;
                else
                    reg_fstate <= state1;
            end
            state3: begin
                if ((input1 & input2))
                    reg_fstate <= state3;
                else
                    reg_fstate <= state1;
```

from .smf to HDL

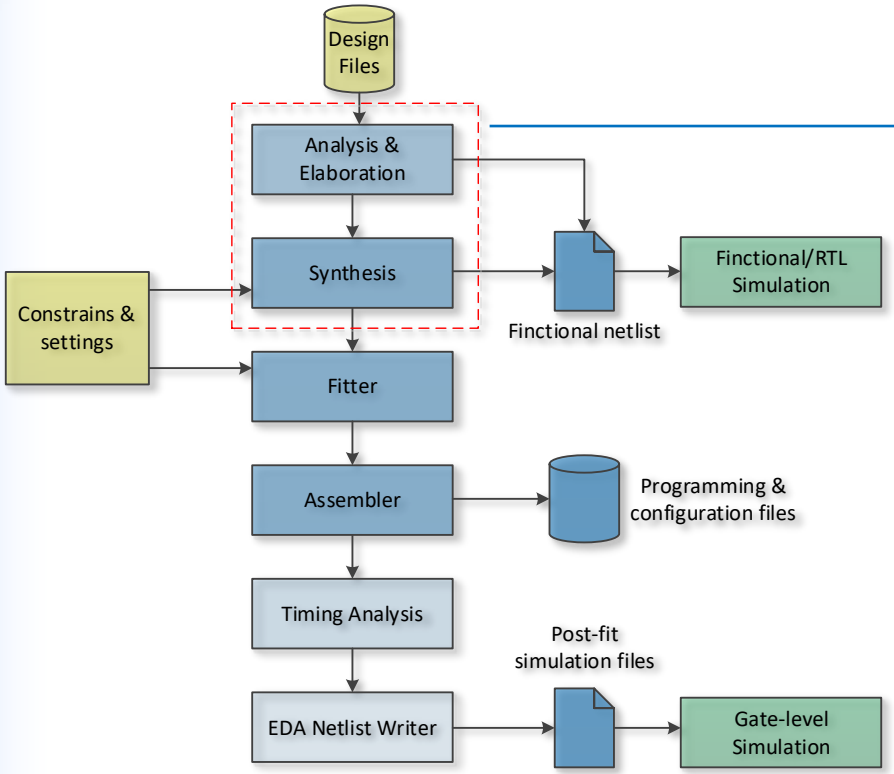
source: https://www.intel.com/content/www/us/en/support/programmable/support-resources/design-software/fpga-development-tools-support.html

Intel MAX10 Design Flow

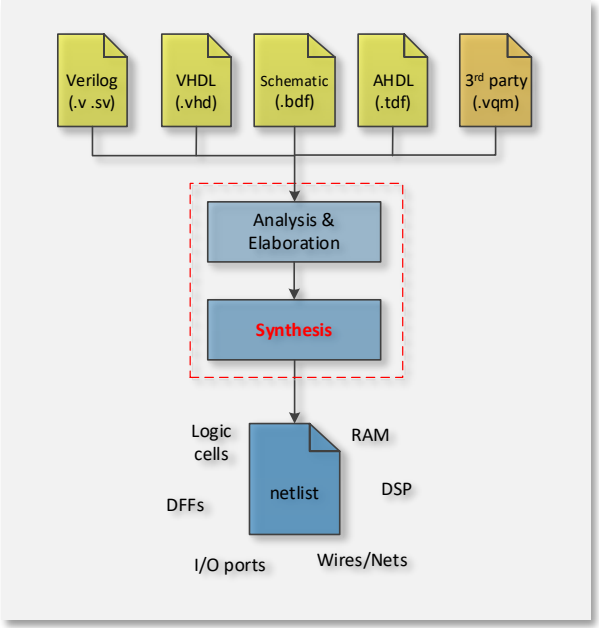


source: https://www.intel.com/content/www/us/en/docs/programmable/683196/current/fpga-design-guidelines-34288.html

Design compilation



Synthesis

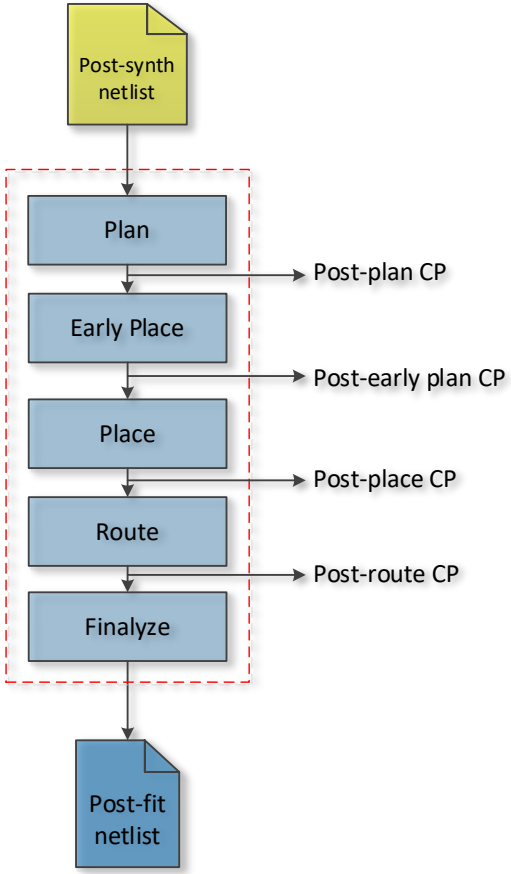


- Translates HDL source files into an atom netlist
- Generates an advanced hierarchical database

source: https://www.intel.com/content/www/us/en/support/programmable/support-resources/design-software/fpga-development-tools-support.html

Design compilation - Fitter

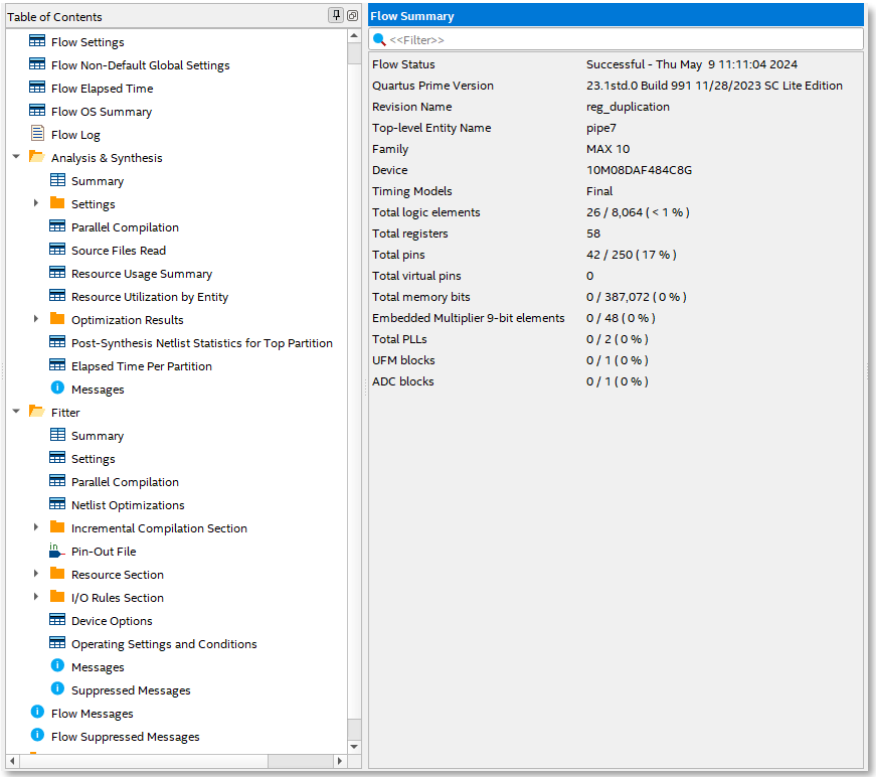
- Advanced place & route engine for finding a valid “solution” in a “reasonable” amount of time
- Consists of 4 stages
  - Plan
    - Periphery (I/O) placement and routing, clock resource selection
  - Early Place
    - Early assignment of core logic to device resources
    - More pessimistic results within 20% of final routed results
  - Place
    - Core resource placement (logic elements, registers, DSP, RAM)
  - Route
    - Core routing connections made
  - Finalize
    - Post-routing optimizations
- Stages can be all run or individually
  - End of running each stage referred to as a checkpoint (**CP**)
- Prior stages must be complete before running later stages



source: https://www.intel.com/content/www/us/en/support/programmable/support-resources/design-software/fpga-development-tools-support.html

Compilation Results

- Quartus Prime Design Software graphical tools available for
  - Understanding design processing
  - Verifying correct design results
  - Debugging incorrect results
- Compilation Report
- Viewers
  - RTL Viewer
  - Technology Map Viewer
  - State Machine Viewer
- Chip Planner

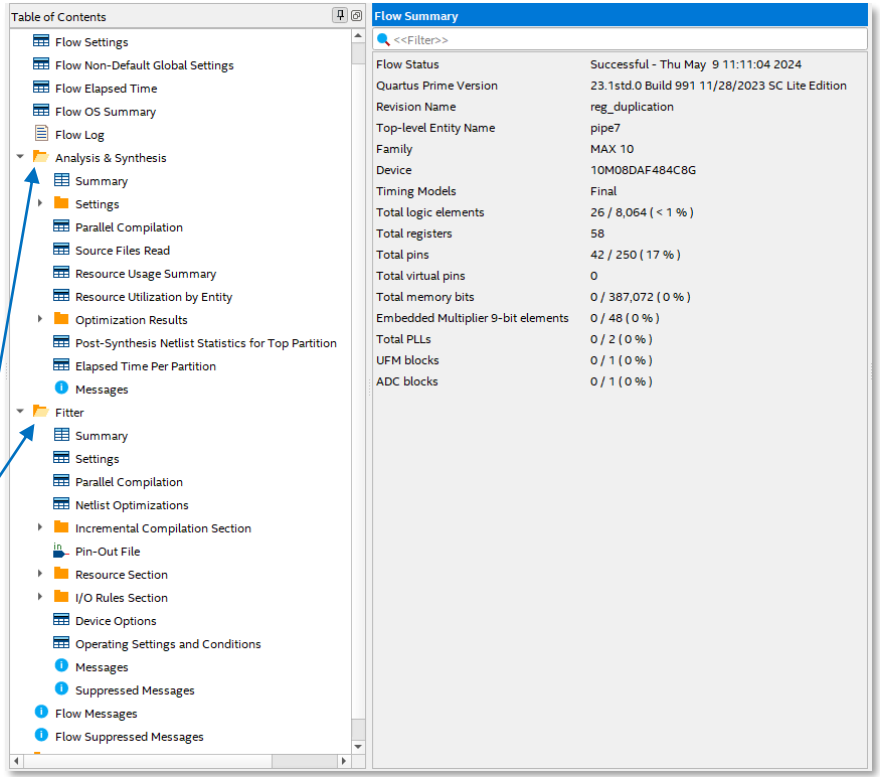


source: https://www.intel.com/content/www/us/en/support/programmable/support-resources/design-software/fpga-development-tools-support.html

Compilation Results

- Compilation Report: contains all compilation processing information
  - Resource usage
  - Device pin-out
  - Settings and constraints applied
  - Messages
- *Recommendation*: Go through report for a design to get sense of information being provided
- Information also available as text files in **output_files** folder in project directory: *<revision_name>.**syn.rpt**, <revision_name>.**fit.rpt**, <revision_name>.**fit.plan.rpt**,* etc.
- Access from **Processing** menu, toolbar, or Compilation Dashboard
- Each compiler process generates separate folder

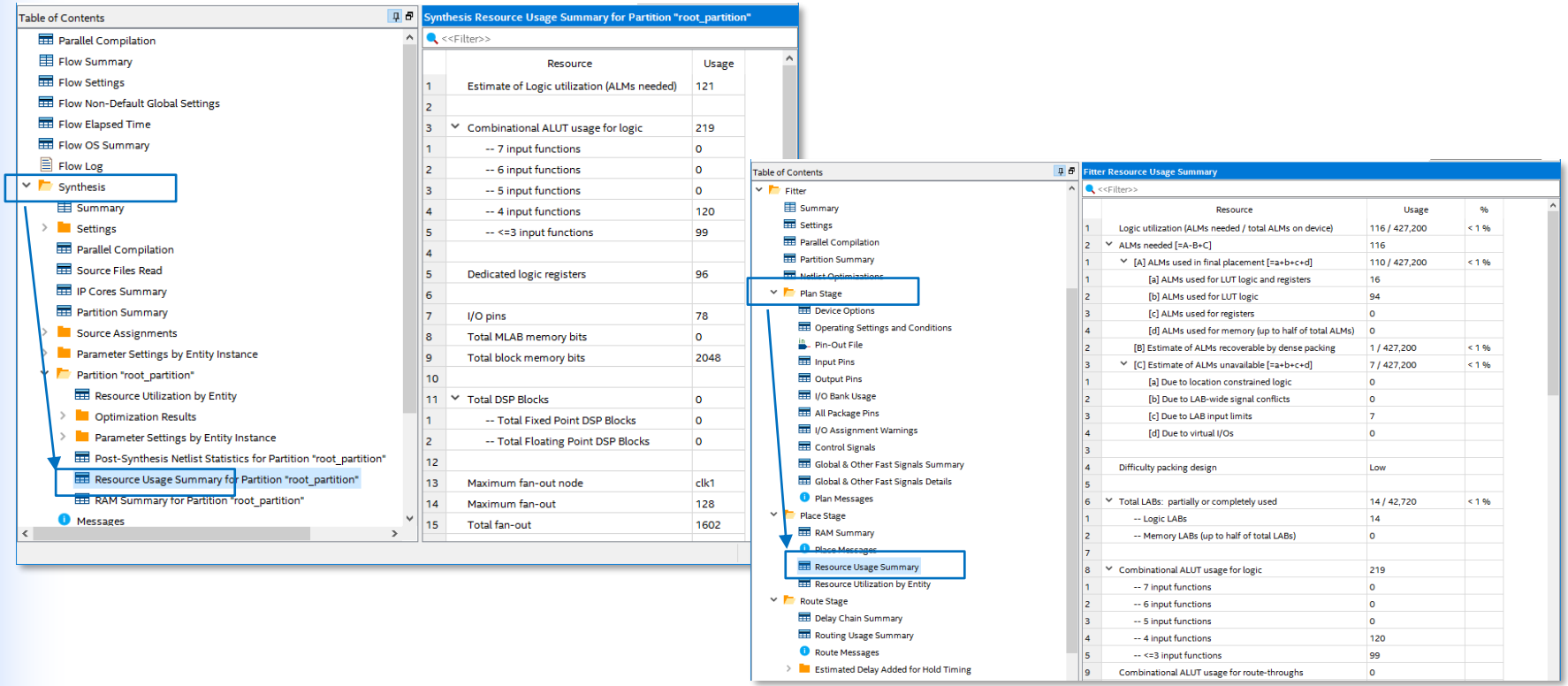
Compilation Report GUI



source: https://www.intel.com/content/www/us/en/support/programmable/support-resources/design-software/fpga-development-tools-support.html

Example: Compilation Results - Resources

- Synthesis resource usage: **estimates of FPGA resources** required to implement design
- Fitter resource usage: **detailed information** on all resources used by design for each stage



source: https://www.intel.com/content/www/us/en/support/programmable/support-resources/design-software/fpga-development-tools-support.html

Netlist Viewers

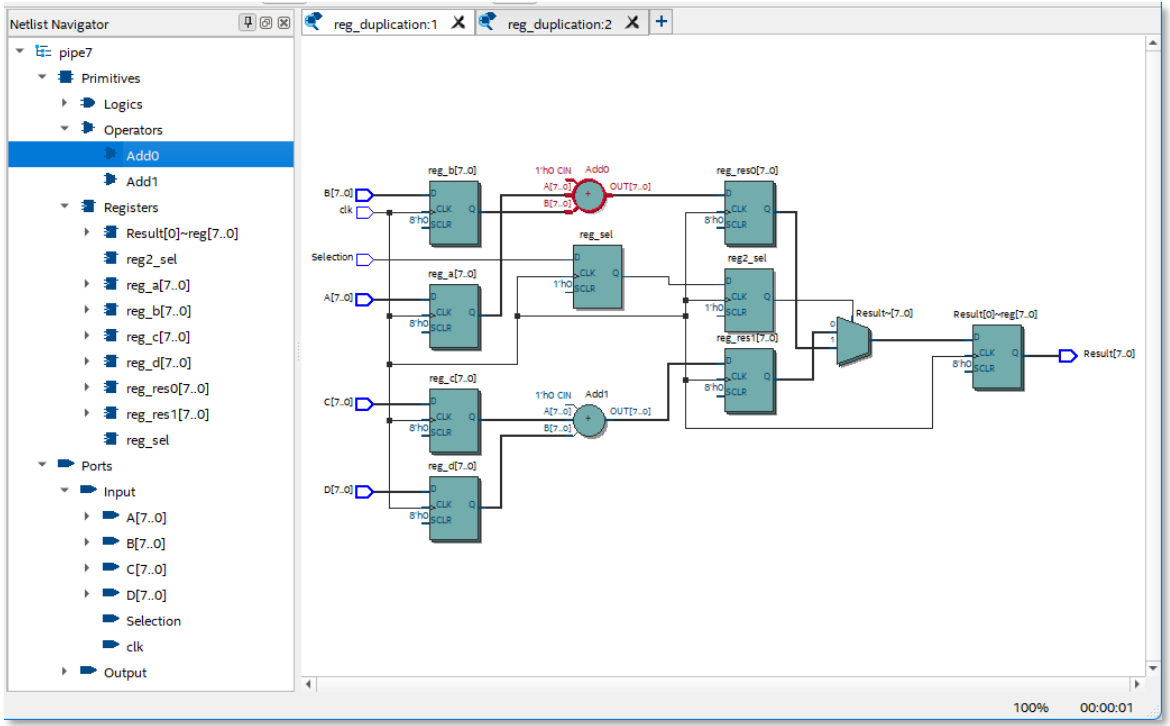
- RTL Viewer
  - Schematic of design after Analysis & Elaboration
  - Visually check initial HDL before synthesis optimizations
  - Locate synthesized nodes for assigning constraints
  - Debug verification issues

- Technology Map Viewers (Post-Mapping or Post-Fitting)
  - Graphically represents results of mapping (post-synthesis) & fitting
  - Analyze critical timing paths graphically
  - Locate nodes & node names after optimizations (cross-probing)

source: https://www.intel.com/content/www/us/en/support/programmable/support-resources/design-software/fpga-development-tools-support.html

Netlist Viewers

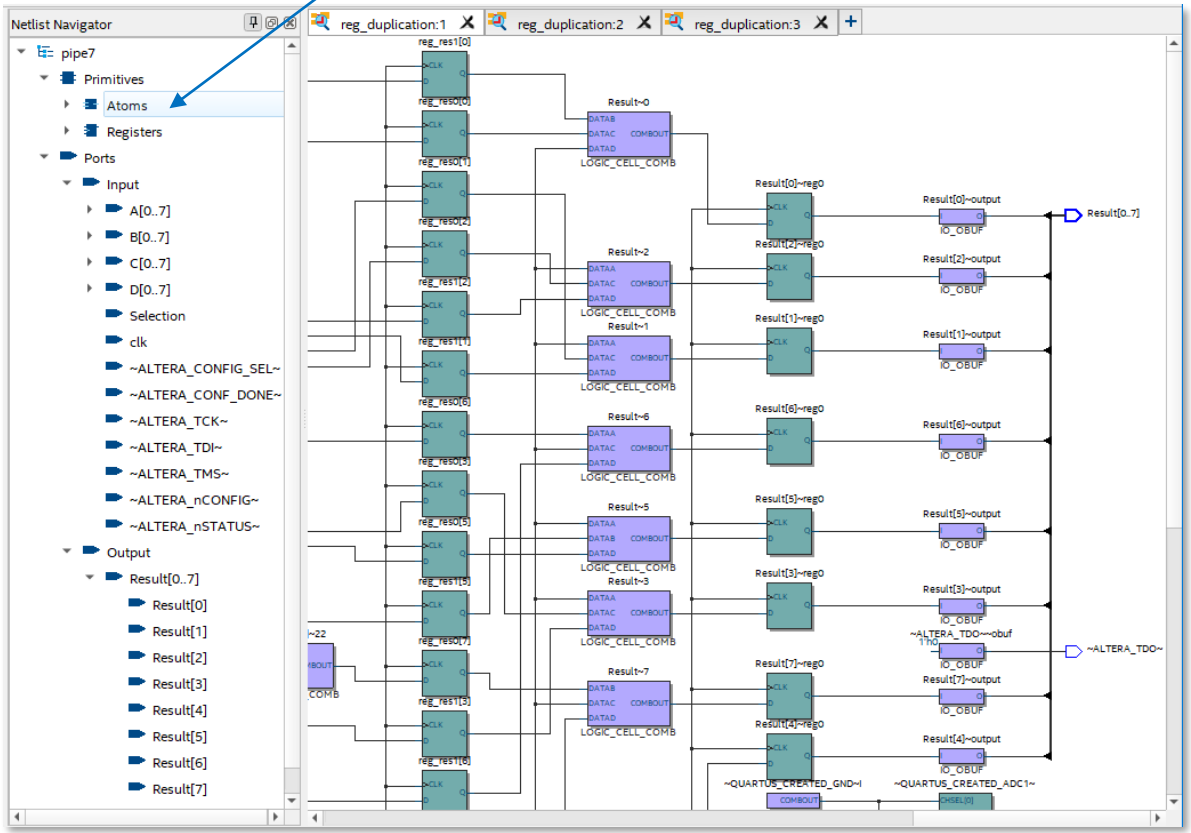
- RTL Viewer - represents design using logic blocks & nets
  - I/O pins
  - Registers
  - Muxes
  - Gates (AND, OR, etc.)
  - Operators (adders, multipliers, etc.)



source: https://www.intel.com/content/www/us/en/support/programmable/support-resources/design-software/fpga-development-tools-support.html

Netlist Viewers

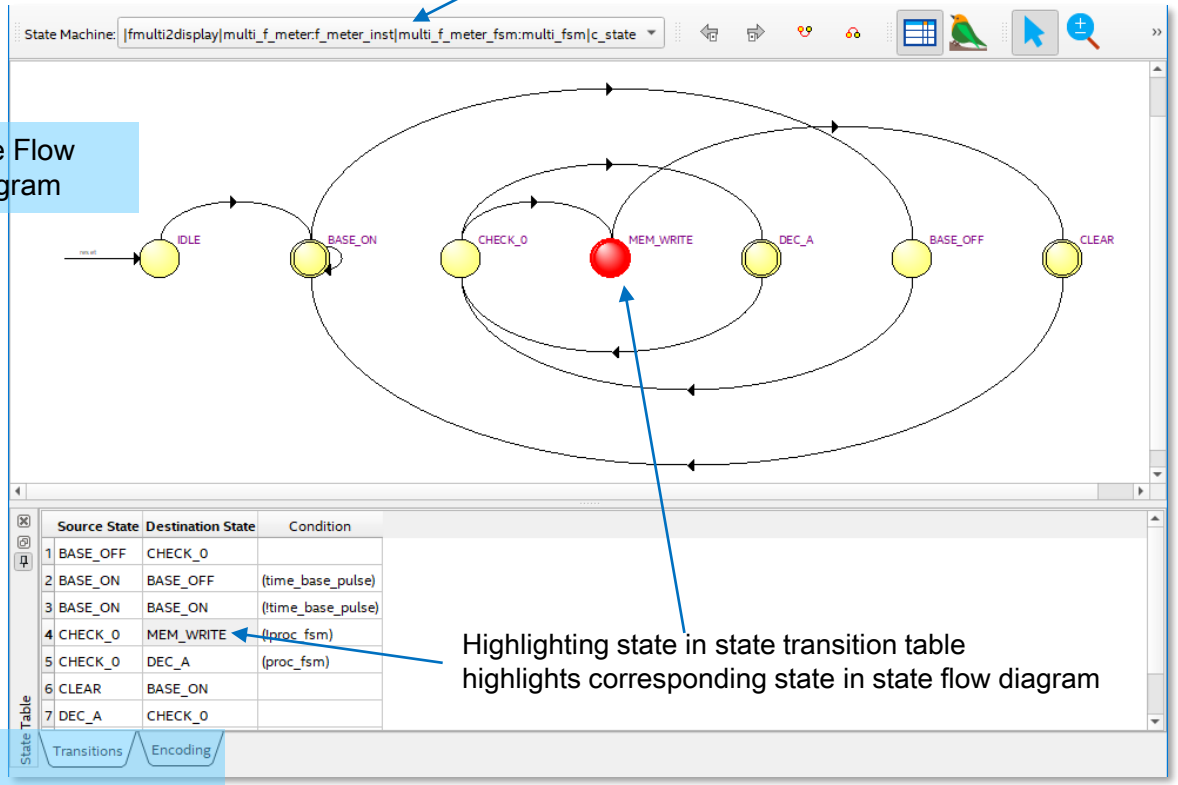
- Technology Map Viewers - represents design using atoms
  - I/O pins & cells
  - Logic cells (Lcells)
  - Memory blocks
  - MAC (DSP blocks)



source: https://www.intel.com/content/www/us/en/support/programmable/support-resources/design-software/fpga-development-tools-support.html

State Machine Viewer

- **Tools** menu → **Netlist Viewers**



source: https://www.intel.com/content/www/us/en/support/programmable/support-resources/design-software/fpga-development-tools-support.html