



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

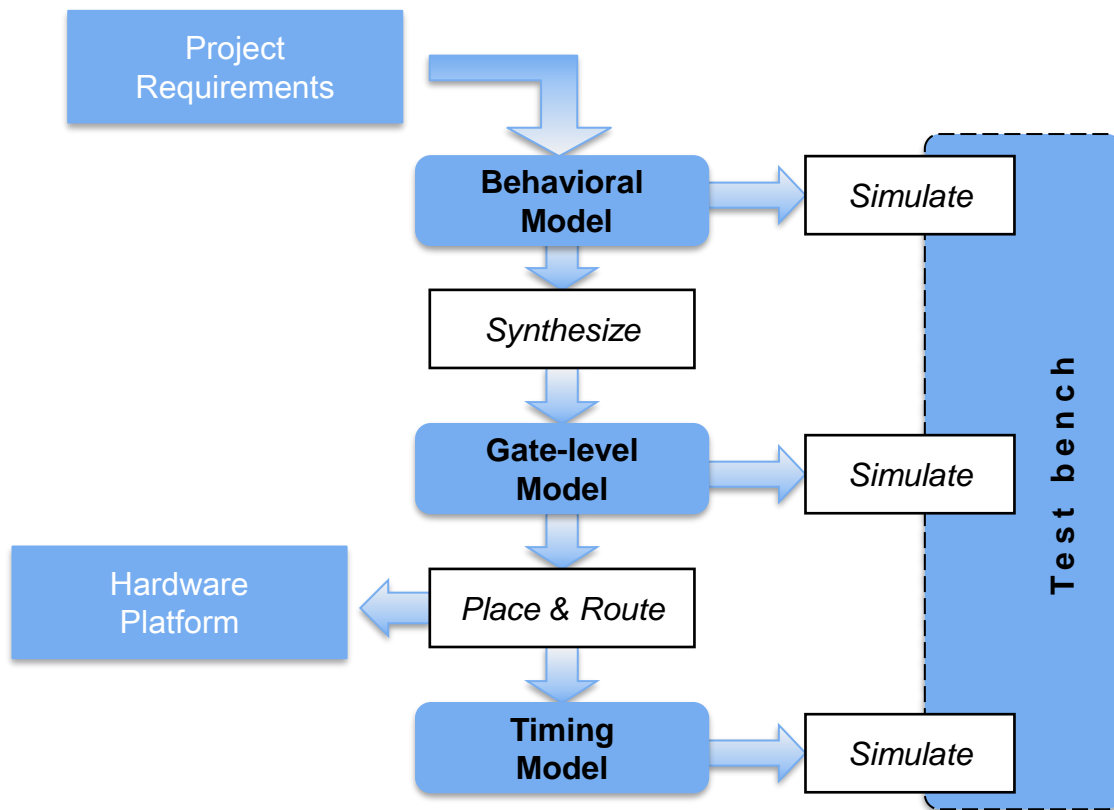


# Modeling Digital Systems

- 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





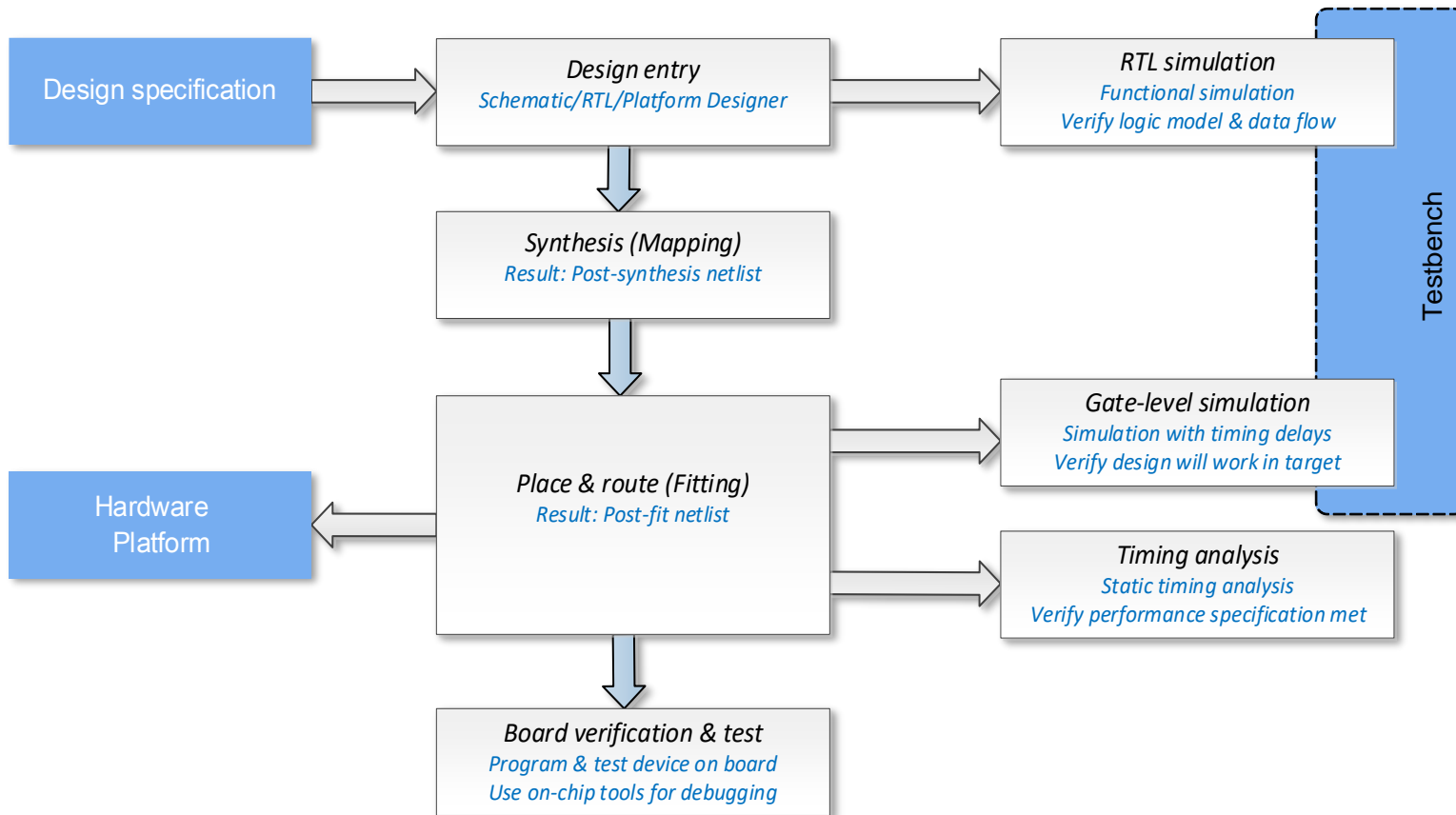
# Design stages

- 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)**



# 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

**Intel® Quartus® Prime**  
Design Software

**Lite Edition (LE)**

**Standard Edition (SE)**

**Pro Edition (PE)**



# Quartus Prime GUI

Digital Logic Design with FPGA

The screenshot shows the Quartus Prime GUI interface with the following components highlighted:

- Project Navigator:** Located on the left side, showing the project hierarchy for 'MAX 10: 10M50DAF484C6GES' and the 'multi\_f' project.
- Task window:** Located below the Project Navigator, showing the compilation tasks such as 'Compile Design', 'Analysis & Synthesis', 'Fitter (Place & Route)', 'Assembler (Generate program)', 'Timing Analysis', and 'FDA Netlist Writer'.
- Tool View window:** The central area displaying the VHDL code for 'multi\_f\_meter\_fsm.vhd'.
- IP Catalog:** Located on the right side, showing the installed IP library with categories like 'Project Directory', 'Library', 'Basic Functions', 'DSP', 'Interface Protocols', 'Memory Interfaces and Controllers', 'Processors and Peripherals', and 'University Program'.
- Messages window:** Located at the bottom, displaying a list of messages from the compilation process, including warnings and successful completion messages.

```
20 signal q : std_logic_vector(23 downto 0);
21 signal fq : std_logic_vector(15 downto 0);
22 signal fdp : std_logic_vector(3 downto 0);
23 signal frange : std_logic_vector(3 downto 0);
24 signal sseg_out : std_logic_vector(41 downto 0);
25 signal rst_f_in : std_logic;
26
27
28 begin
29   f_meter_inst: entity work.multi_f_meter
30     port map(adc_clk_10,rst_f_in,frange,fdp,fq);
31
32   q(23 downto 8) <= fq;
33   q(7 downto 4) <= x"E";
34   q(3 downto 0) <= frange;
35
36   gen: for i in 1 to 6 generate
37     led7seg: entity work.dec7seg
38     port map(q((i*4)-1 downto (i-1)*4),sseg_out((i*7)-1 downto (i-1)*7));
39   end generate;
40
41   hex0 <= '1' & sseg_out(6 downto 0);
42   hex1 <= '1' & sseg_out(13 downto 7);
43   hex2 <= (not fdp(0)) & sseg_out(20 downto 14);
44   hex3 <= (not fdp(1)) & sseg_out(27 downto 21);
45 end;
```

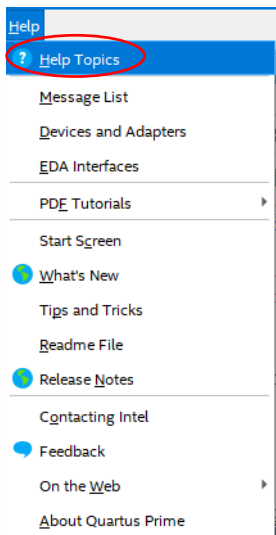
Type	ID	Message
Information	12128	Elaborating entity "dec7seg" for hierarchy "dec7seg:\gen:1:led7seg"
Warning	13024	Output pins are stuck at VCC or GND
Information	286030	Timing-Driven Synthesis is running
Information	16010	Generating hard_block partition "hard_block:auto_generated_inst"
Warning	21074	Design contains 1 input pin(s) that do not drive logic
Information	21057	Implemented 318 device resources after synthesis - the final resource count might be different Quartus Prime Analysis & synthesis was successful. 0 errors, 14 warnings

Messages window

100% 00:00:13



# Quartus Prime Help System



## Quartus Prime Standard Edition Help version 18.1

Search

Content

- Welcome to the Software
  - Welcome to the Quartus® Prime Standard Edition Software
- Managing Projects
  - About Simulating Designs
  - Running Timing Analysis
- Timing Analysis Settings
  - Set Clock Groups Dialog Box (set\_clock\_groups)**
  - Set Clock Latency Dialog Box (set\_clock\_latency)
  - Set Clock Uncertainty Dialog Box (set\_clock\_uncertainty)
  - Set False Path Dialog Box (set\_false\_path)
  - Set Input Delay Dialog Box (set\_input\_delay)
  - Set Output Delay Dialog Box (set\_output\_delay)
  - Set Maximum Delay Dialog Box (set\_max\_delay)
  - Set Minimum Delay Dialog Box (set\_min\_delay)
  - Set Multicycle Path Dialog Box (set\_multicycle\_path)
- Integrating Other EDA Tools
- Simulation Settings
  - About Integrating Other EDA Tools
  - Preparing for EDA Simulation
  - Running EDA Simulators
- Simulation Tools
  - Active-HDL
    - Performing a Simulation of a Verilog HDL Design with the Active-HDL Software
    - Performing a Simulation of a VHDL Design with the Active-HDL Software
  - ModelSim
  - ModelSim-Altera
  - Incisive Enterprise Simulator
  - QuestaSim
  - Riviera-PRO

### Set Clock Groups Dialog Box (set\_clock\_groups)

You access this dialog box by clicking **Set Clock Groups** on the Constraints menu in the TimeQuest Timing Analyzer.

Allows you to specify which clocks in the design are unrelated.

The TimeQuest analyzer analyzes all clocks in a design as related, by default. The **Set Clock Groups** dialog box (and the `set_clock_groups` SDC command) allows you to specify unrelated clocks by creating smaller groups of related clocks.

Asynchronous clocks (`-asynchronous`) are completely unrelated; they have different ideal clock sources. Exclusive clocks (`-exclusive`) are not active at the same time, such as multiplexed clocks. The TimeQuest analyzer treats both types of clocks as if they were the same.

Specifying settings in this dialog box is equivalent to setting the `set_false_path` command for paths from each clock domain in every group to each clock domain in every other group. Specifying a single `-group` option results in the TimeQuest analyzer cutting this group of clocks from all other clocks in the design, including clocks that you might define in the future.

The following sections provide more information about specifying options for this constraint:

#### Group 1 (-group):

Allows you to specify the clocks to which the constraint applies. You can use the Name Finder (...) and the `get_clocks` option to build a [collection Definition](#) of clocks. Clocks specified in this group are cut from the clocks specified in the **Group 2** box.

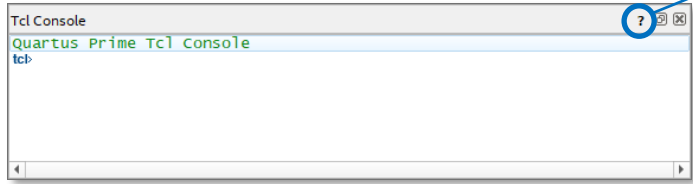
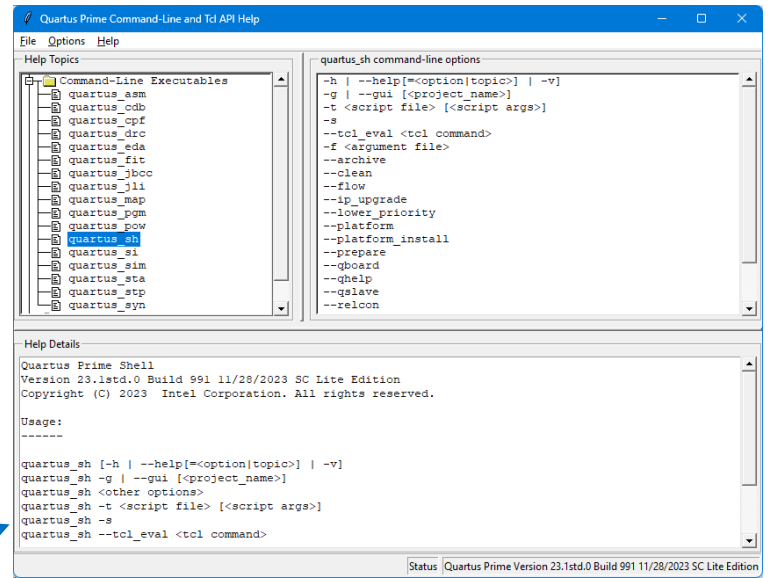
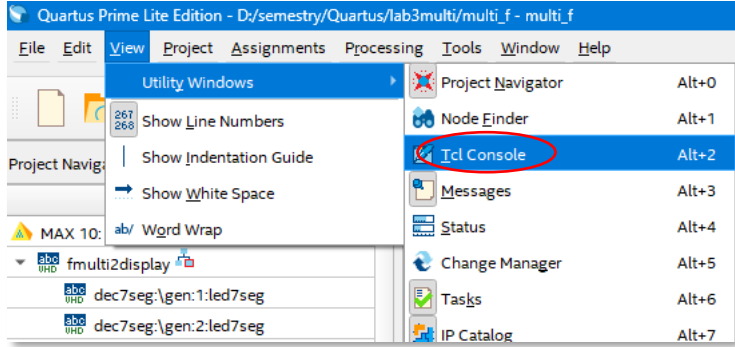
#### Group 2 (-group):

Allows you to specify the clocks to which the constraint applies. You can use the Name Finder (...) and the `get_clocks` option to build a [collection Definition](#) of clocks. Clocks specified in this group are cut from the clocks specified in the **Group 1** box. If you leave this setting blank, then the clocks in the **Group 1** box are cut from every other

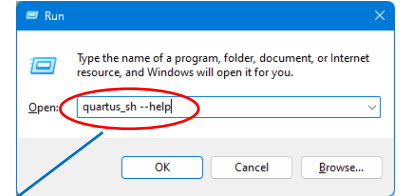


# Tcl Console

- 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

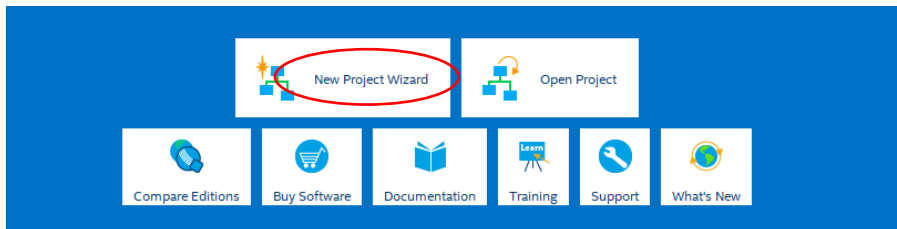


C:\<install dir>\quartus\bin64\quartus\_sh --qhhelp



# 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**

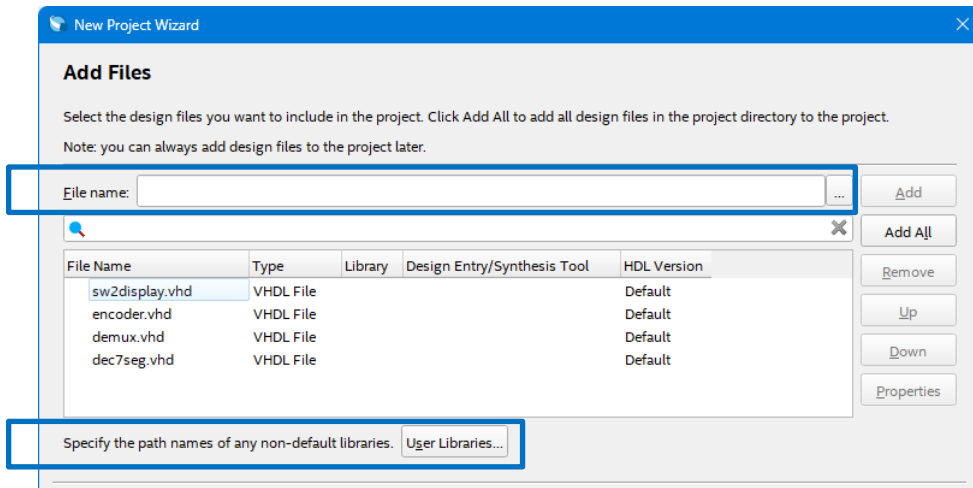


- Can be created using Tcl scripts  
**Tcl:** `roject_new <project_name>`



# 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>
```





# Device & Family Selection

- Choose device family & family category

Device family

Family: MAX 10 (DA/DD/DF/DC/SA/SC/SL)

Device: MAX 10 DA

Target device

Auto device selected by the Fitter

Specific device selected in 'Available devices' list

Other: n/a

Show in 'Available devices' list

Package: FBGA

Pin count: 484

Core speed grade: 8

Name filter:

Show advanced devices

Device and Pin Options...

Available devices:

Name	Core Voltage	LEs	Total I/Os	GPIOs	Memory Bits	Embedded multiplier 9-bit element
10M08DAF484C8G	1.2V	8064	250	250	387072	48
10M08DAF484C8GES	1.2V	8064	250	250	387072	48

Migration Devices... 0 migration devices selected

Buy Software OK Cancel Help

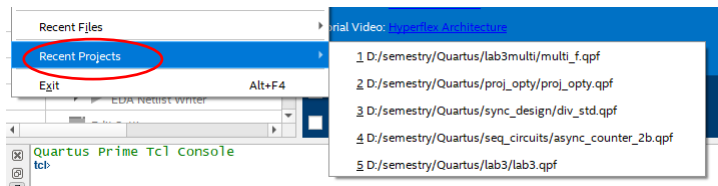
```
Tcl: set_global_assignment -name FAMILY "device family name"
```

```
Tcl: set_global_assignment -name DEVICE <part_number>
```

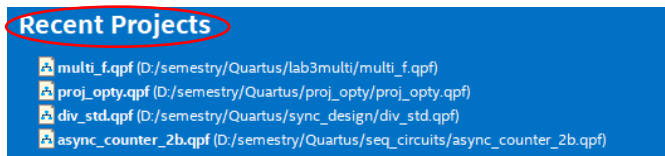


# Opening a Project

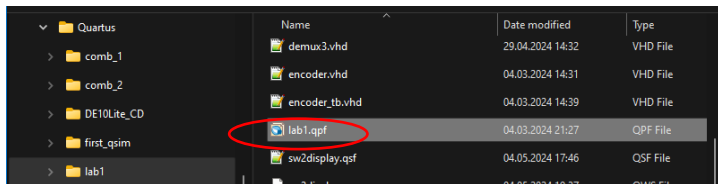
- From File menu



- From Recent Project



- Double-click .qpf file



- Tcl script

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



# Quartus Prime Project Files

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

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



# 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



# Project Archive & Restore

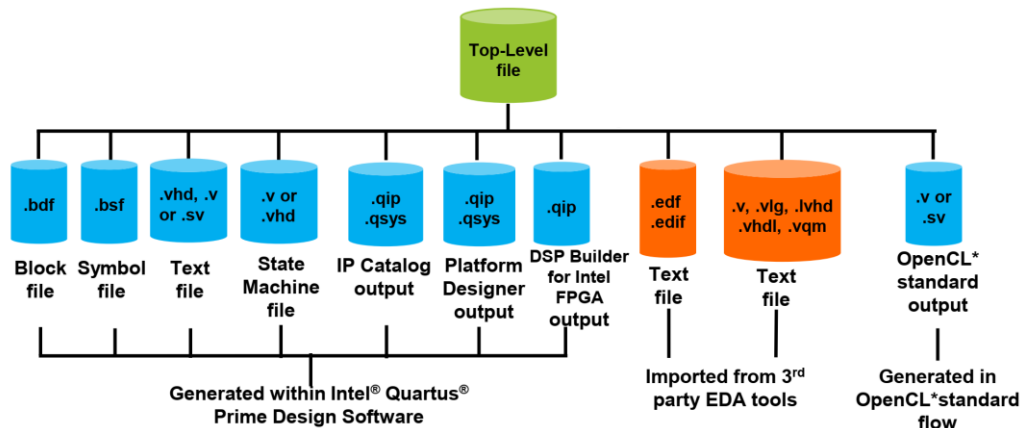
- **Archive** Creates 2 files
    - Compressed Quartus Prime Design Software Archive File (.qar)
      - Includes design files, .qpf file, & .qsf file(s)
      - Option to include databases
      - Creates local .qdf file for archive
    - Archive activity log (.qarlog)
- Tcl:** `project_archive <project_name>`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



# Design entry methods

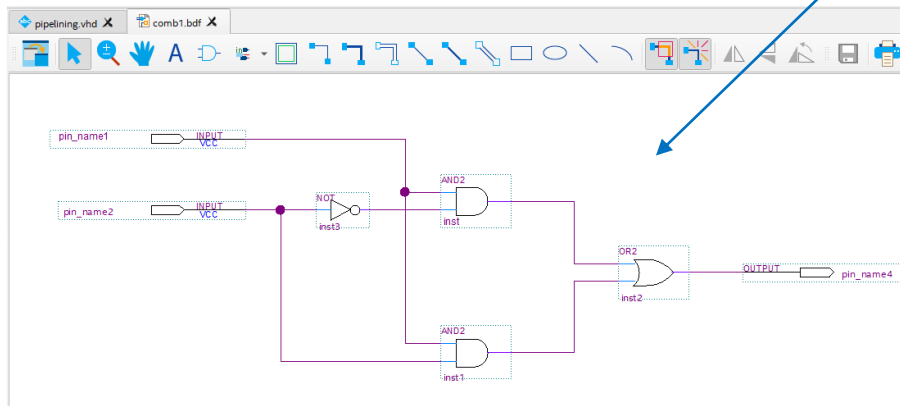
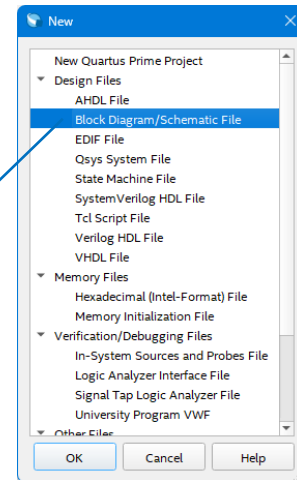
- Quartus Prime design entry
  - Text editor
    - VHDL
    - Verilog or SystemVerilog
  - Schematic editor
    - Block Diagram File
  - System editor
    - Platform Designer
  - State machine editor
    - HDL from state machine file
  - Memory editor
    - HEX
    - MIF
- 3rd-party EDA tools
  - EDIF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





# Schematic Editor

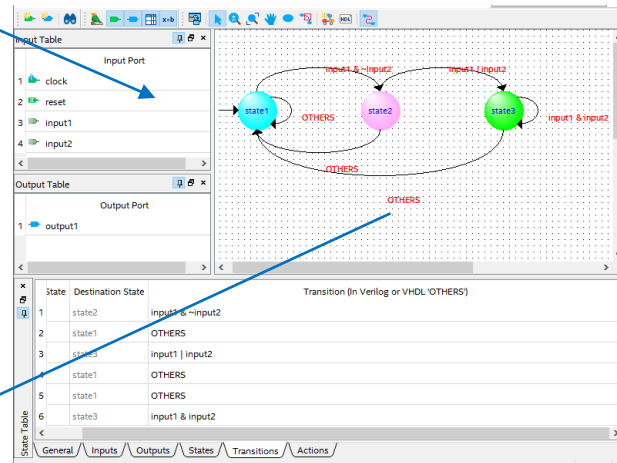
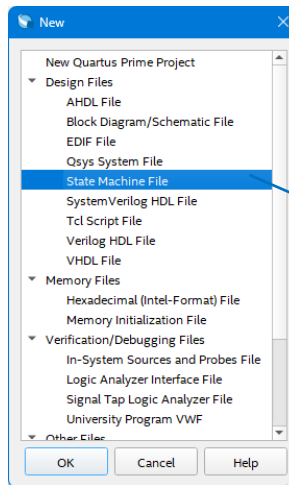
- Full-featured schematic design capability
- Schematic Editor uses
  - Create simple test designs to understand the functionality of an Intel FPGA IP: PLL, LVDS I/O, memory, etc...
  - Create top-level schematic for easy viewing & connection
  - Convert between schematic .bdf, block symbol .bsf, and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





# 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



```

37 always @(posedge clock)
38 begin
39   if (clock) begin
40     fstate <= reg_fstate;
41   end
42 end
43
44 always @(fstate or reset or input1 or input2)
45 begin
46   if (reset) begin
47     reg_fstate <= state1;
48   end
49   else begin
50     case (fstate)
51       state1: begin
52         if ((input1 & ~(input2)))
53           reg_fstate <= state2;
54         else
55           reg_fstate <= state1;
56       end
57       state2: begin
58         if ((input1 | input2))
59           reg_fstate <= state3;
60         else
61           reg_fstate <= state1;
62       end
63       state3: begin
64         if ((input1 & input2))
65           reg_fstate <= state3;
66         else
67           reg_fstate <= state1;

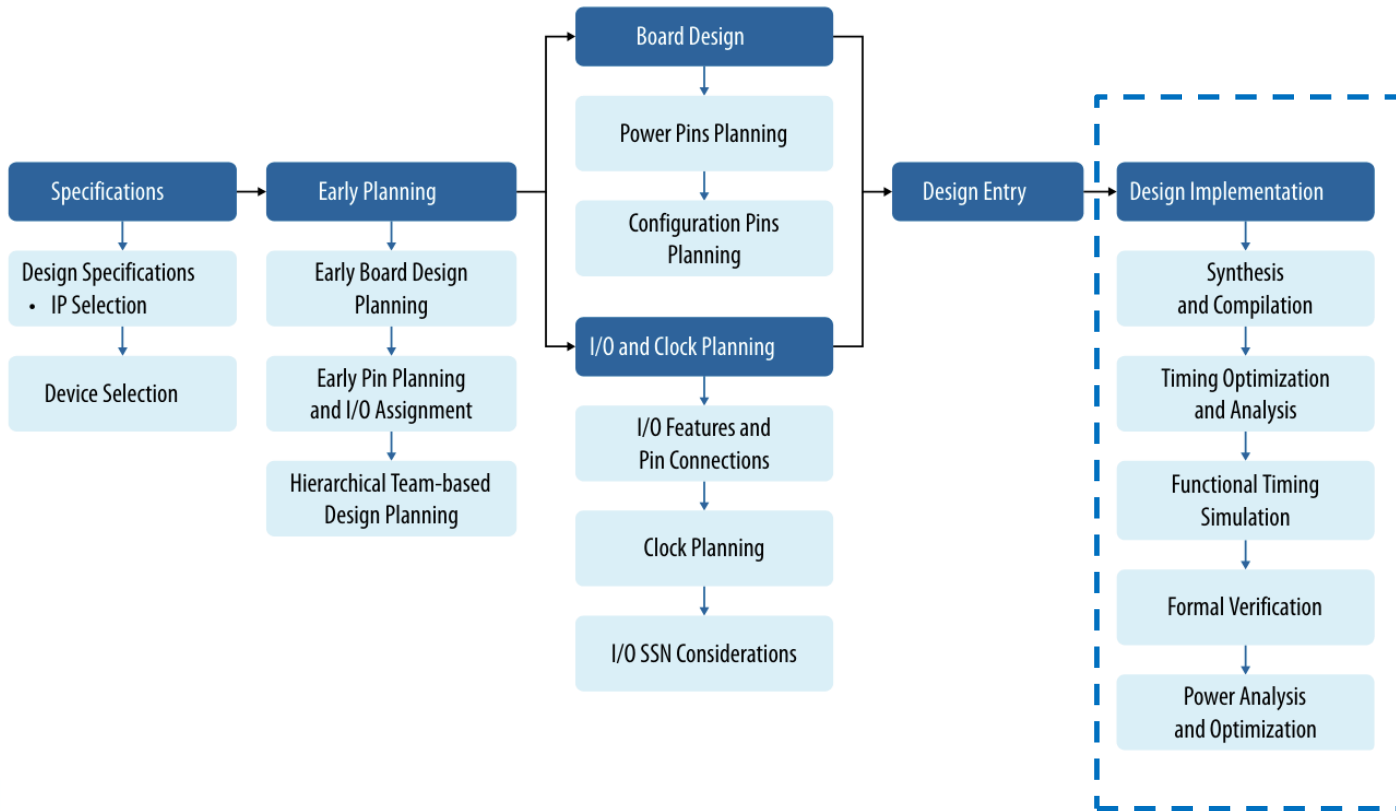
```

from .smf  
to HDL



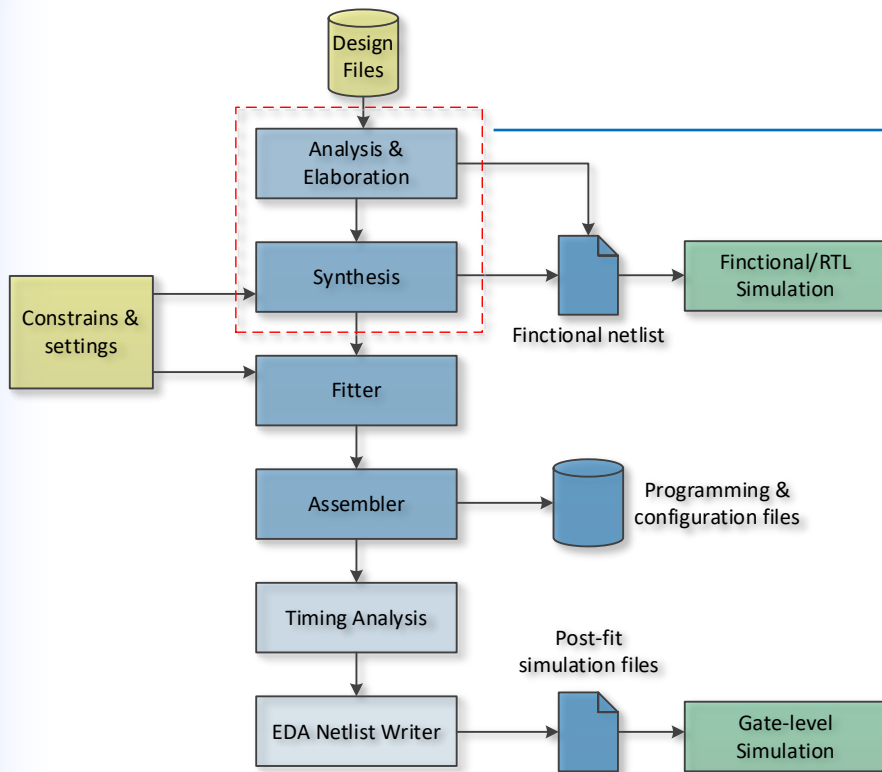


# Intel MAX10 Design Flow

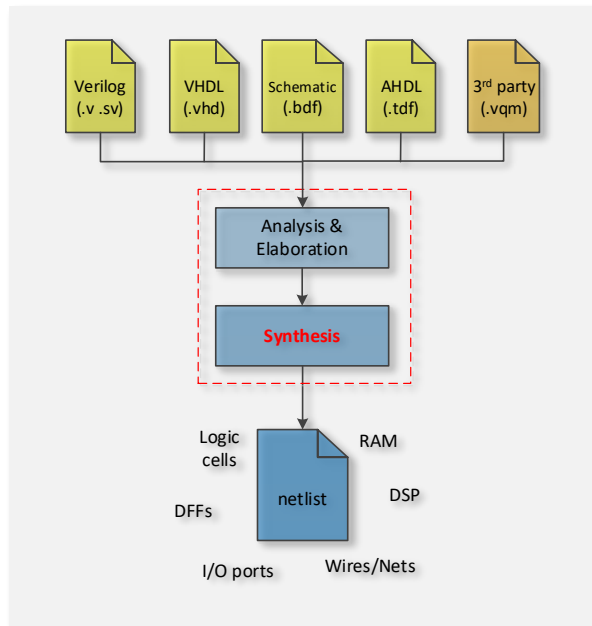




# Design compilation



## Synthesis

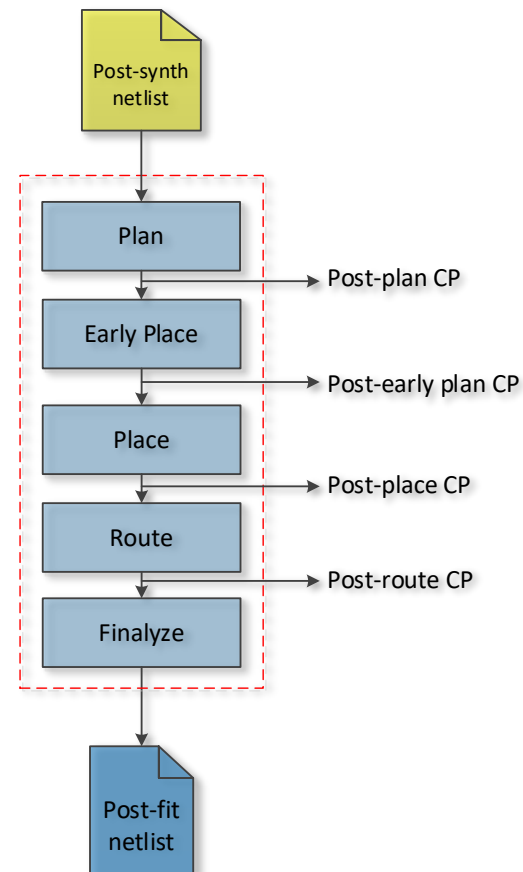


- Translates HDL source files into an atom netlist
- Generates an advanced hierarchical database



# 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





# 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**

The screenshot displays two windows from the Quartus Prime software. The left window, titled 'Table of Contents', shows a hierarchical tree of report sections. The right window, titled 'Flow Summary', displays a table of compilation statistics.

Flow Summary	
<<Filter>>	
Flow Status	Successful - Thu May 9 11:11:04 2024
Quartus Prime Version	23.1std.0 Build 991 11/28/2023 5C Lite Edition
Revision Name	reg_duplication
Top-level Entity Name	pipe7
Family	MAX 10
Device	10M08DAF484C8G
Timing Models	Final
Total logic elements	26 / 8,064 (< 1 %)
Total registers	58
Total pins	42 / 250 (17 %)
Total virtual pins	0
Total memory bits	0 / 387,072 (0 %)
Embedded Multiplier 9-bit elements	0 / 48 (0 %)
Total PLLs	0 / 2 (0 %)
UFM blocks	0 / 1 (0 %)
ADC blocks	0 / 1 (0 %)



# 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

The screenshot displays the 'Compilation Report GUI' with two main panels. The left panel, titled 'Table of Contents', shows a hierarchical tree of report sections. The right panel, titled 'Flow Summary', displays a table of key metrics.

Flow Summary	
<<Filter>>	
Flow Status	Successful - Thu May 9 11:11:04 2024
Quartus Prime Version	23.1std.0 Build 991 11/28/2023 5C Lite Edition
Revision Name	reg_duplication
Top-level Entity Name	pipe7
Family	MAX 10
Device	10M08DAF484C8G
Timing Models	Final
Total logic elements	26 / 8,064 (< 1 %)
Total registers	58
Total pins	42 / 250 (17 %)
Total virtual pins	0
Total memory bits	0 / 387,072 (0 %)
Embedded Multiplier 9-bit elements	0 / 48 (0 %)
Total PLLs	0 / 2 (0 %)
UFM blocks	0 / 1 (0 %)
ADC blocks	0 / 1 (0 %)



# 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

Table of Contents

- Parallel Compilation
- Flow Summary
- Flow Settings
- Flow Non-Default Global Settings
- Flow Elapsed Time
- Flow OS Summary
- Flow Log
- Synthesis**
  - Summary
  - Settings
  - Parallel Compilation
  - Source Files Read
  - IP Cores Summary
  - Partition Summary
  - Source Assignments
  - Parameter Settings by Entity Instance
  - Partition "root\_partition"
    - Resource Utilization by Entity
    - Optimization Results
    - Parameter Settings by Entity Instance
    - Post-Synthesis Netlist Statistics for Partition "root\_partition"
    - Resource Usage Summary for Partition "root\_partition"**
    - RAM Summary for Partition "root\_partition"
- Messages

Resource	Usage
1 Estimate of Logic utilization (ALMs needed)	121
2	
3 <b>Combinational ALUT usage for logic</b>	219
1 -- 7 input functions	0
2 -- 6 input functions	0
3 -- 5 input functions	0
4 -- 4 input functions	120
5 -- <=3 input functions	99
4	
6 Dedicated logic registers	96
7 I/O pins	78
8 Total MLAB memory bits	0
9 Total block memory bits	2048
10	
11 <b>Total DSP Blocks</b>	0
1 -- Total Fixed Point DSP Blocks	0
2 -- Total Floating Point DSP Blocks	0
12	
13 Maximum fan-out node	clk1
14 Maximum fan-out	128
15 Total fan-out	1602

Table of Contents

- Filter
  - Summary
  - Settings
  - Parallel Compilation
  - Partition Summary
  - Netlist Optimizations
  - Plan Stage**
    - Device Options
    - Operating Settings and Conditions
    - Pin-Out File
    - Input Pins
    - Output Pins
    - I/O Bank Usage
    - All Package Pins
    - I/O Assignment Warnings
    - Control Signals
    - Global & Other Fast Signals Summary
    - Global & Other Fast Signals Details
    - Plan Messages
    - Place Stage
      - RAM Summary
      - Place Messages
      - Resource Usage Summary**
      - Resource Utilization by Entity
    - Route Stage
      - Delay Chain Summary
      - Routing Usage Summary
      - Route Messages
      - Estimated Delay Added for Hold Timing

Resource	Usage	%
1 Logic utilization (ALMs needed / total ALMs on device)	116 / 427,200	< 1 %
2 <b>ALMs needed [=A+B+C]</b>	116	
1 <b>[A] ALMs used in final placement [=a+b+c+d]</b>	110 / 427,200	< 1 %
1 [a] ALMs used for LUT logic and registers	16	
2 [b] ALMs used for LUT logic	94	
3 [c] ALMs used for registers	0	
4 [d] ALMs used for memory (up to half of total ALMs)	0	
2 [B] Estimate of ALMs recoverable by dense packing	1 / 427,200	< 1 %
3 <b>[C] Estimate of ALMs unavailable [=a+b+c+d]</b>	7 / 427,200	< 1 %
1 [a] Due to location constrained logic	0	
2 [b] Due to LAB-wide signal conflicts	0	
3 [c] Due to LAB input limits	7	
4 [d] Due to virtual I/Os	0	
3		
4 Difficulty packing design		Low
5		
6 <b>Total LABs: partially or completely used</b>	14 / 42,720	< 1 %
1 -- Logic LABs	14	
2 -- Memory LABs (up to half of total LABs)	0	
7		
8 <b>Combinational ALUT usage for logic</b>	219	
1 -- 7 input functions	0	
2 -- 6 input functions	0	
3 -- 5 input functions	0	
4 -- 4 input functions	120	
5 -- <=3 input functions	99	
9 Combinational ALUT usage for route-throughs	0	



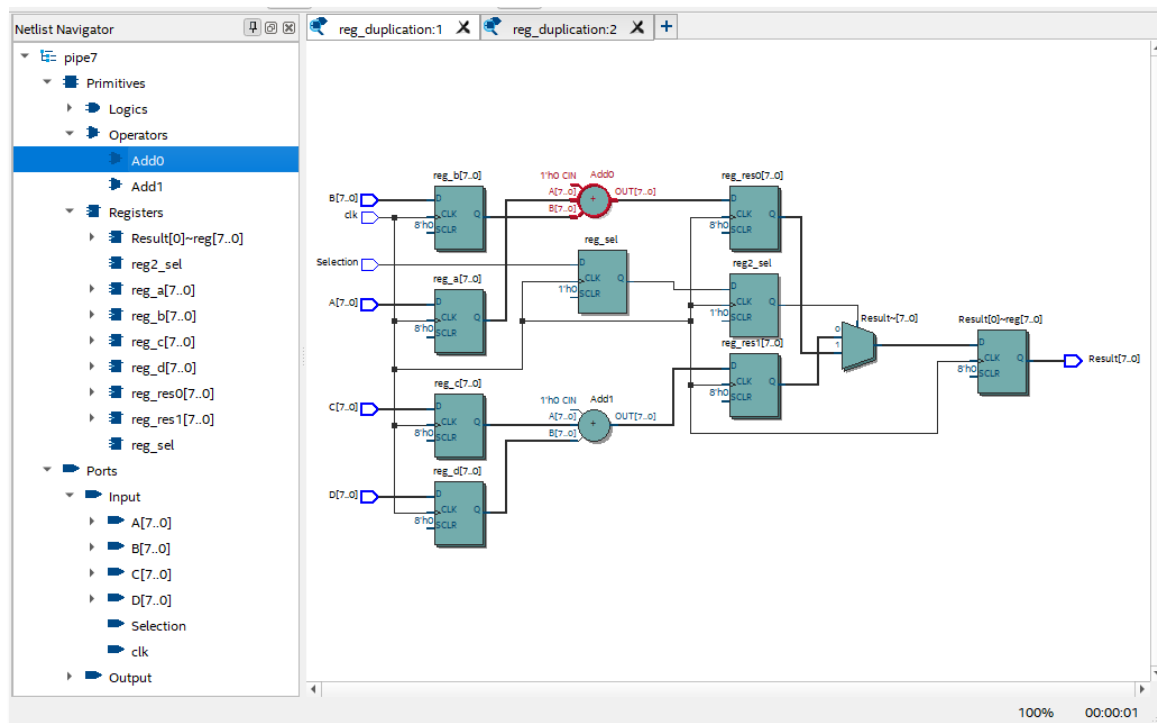
# Netlist Viewers

- **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)



# Netlist Viewers

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



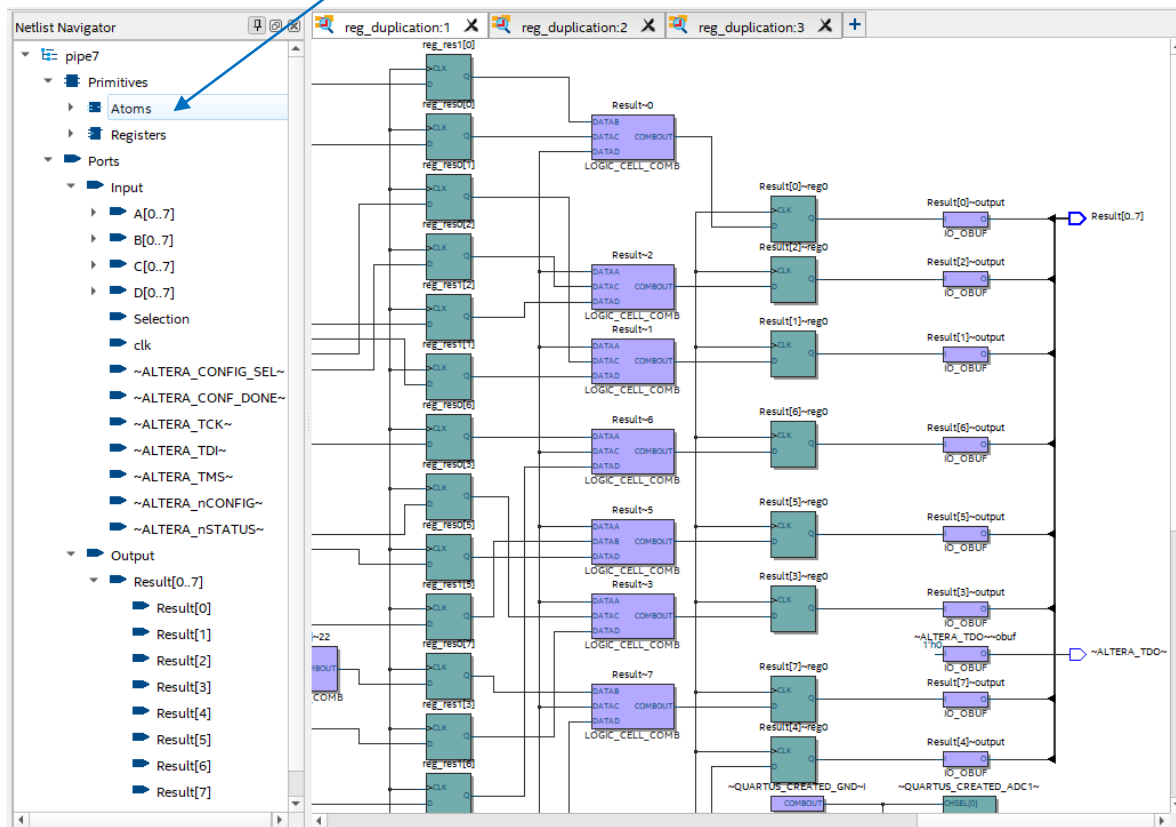




# Netlist Viewers

- **Technology Map Viewers** - represents design using atoms

- I/O pins & cells
- Logic cells (Lcells)
- Memory blocks
- MAC (DSP blocks)





# State Machine Viewer

Use drop-down to select from multiple state machines

- Tools menu → Netlist Viewers

State Flow Diagram

Source State	Destination State	Condition
1 BASE_OFF	CHECK_0	
2 BASE_ON	BASE_OFF	(time_base_pulse)
3 BASE_ON	BASE_ON	(!time_base_pulse)
4 CHECK_0	MEM_WRITE	(!proc_fsm)
5 CHECK_0	DEC_A	(proc_fsm)
6 CLEAR	BASE_ON	
7 DEC_A	CHECK_0	

Highlighting state in state transition table highlights corresponding state in state flow diagram

View transitions and encoding