

Design Entry

- Purpose of design entry is to enter a schematic for use by a set of Electronic Design Automation tools
 - Simulation, testing, fabrication
 - Documentation
- A circuit *schematic* shows the interconnection of structural elements that make up a circuit
 - Doesn't capture everything about design (e.g., doesn't show contents of ROMs)
 - EDA tools need an electronic (usually ASCII) version of that schematic, called a *netlist*
- Schematic capture
 - Direct entry of the circuit schematic
 - More "bookkeeping" than "automation"

Graphic Editor

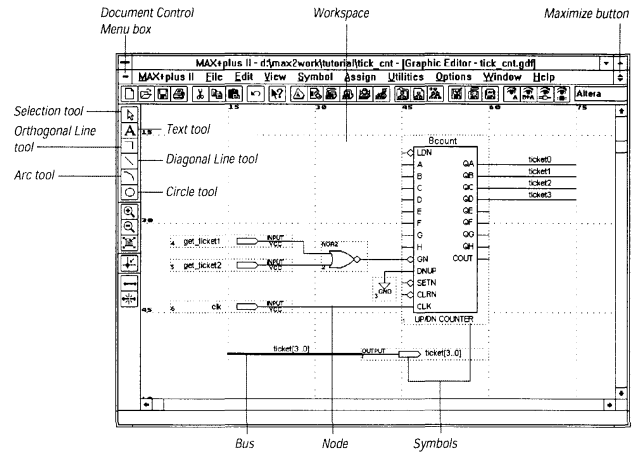


Figure from Altera technical literature

- Enter cells from various predefined component libraries, or user-defined cells
- Connect cells using nets, buses, or name
- "Smart" selection tool — automatically becomes proper tool for task at hand

Schematic Entry

- Circuit schematics are drawn on *schematic sheets*, which come in standard sizes (8.5x11, 11x17, etc.)
 - Each sheet includes a labeled border, and a block listing the circuit name, designer's name, date, etc.
 - There are standards for most of the commonly-used symbols
- Terms used in circuit schematics:

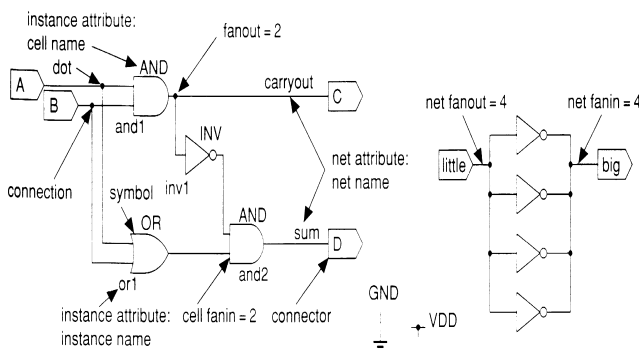


Figure from *Application-Specific Integrated Circuits*, Smith, Addison-Wesley, 1997
Spring 2002, Lecture 05

Cell Library

- *Components* (sometimes called *modules*) in an ASIC schematic are chosen from a library of cells
 - ASIC vendors provide a schematic library of primitive gates for schematic entry
- Problem — no standard exists
 - Individual vendors might use different names to refer to a 2-input xor gate
 - May be some variation on TTL 7400-series names:
 - 2-input NAND = 7400
 - 2-input AND = 7408
 - 2-input OR = 7432
 - 2-input XOR = 7486
 - May be more descriptive:
 - nand2, xor3, ...
 - Behavior may vary
 - Which input does 2-input multiplexor select when select input S = 0?

Names

- Each cell, whether a primitive cell or a subschematic, has a name
 - Each use of a cell in a schematic is a different *instance* of that cell, and is given a unique *instance name*
- Each cell is represented by a picture, or icon, called a *symbol*
 - Primitive cells (e.g., AND gates) have standardized non-rectangular symbols
 - Subschematics are represented by special custom icons

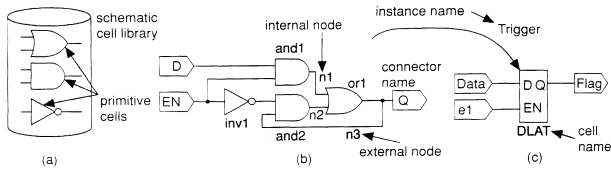


Figure from *Application-Specific Integrated Circuits*, Smith, Addison-Wesley, 1997

Nets

- Cell instances have *terminals*, also known as pins, connectors, or signals, that are the inputs and outputs of the cell
 - A *local* (internal) net is internal to a cell
 - An *external* net connects to the inputs and / or outputs of the cell
- Cell instances are connected by *wire segments*, commonly called *nets*
 - A *local* (internal) net is internal to a cell
 - An *external* net connects to the inputs and / or outputs of the cell
- Nets may sometimes be collected together into *buses* for convenience
 - May be represented by a thicker line on the schematic, with some indication of number of nets involved
 - Individual nets can still be accessed when necessary

Hierarchical Design

- Hierarchy* is used to reduce the size and complexity of the schematic
 - The alternative — drawing all symbols on one giant schematic with no hierarchy — is called a *flat schematic*
 - Flat schematics are impractical to work with for even thousands of components
 - Flat netlists, however, are occasionally used when the hierarchy isn't relevant

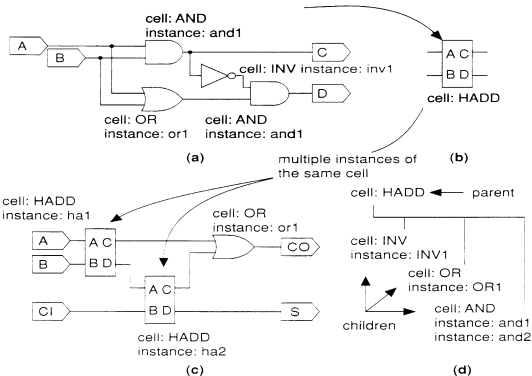


FIGURE 9.3 Schematic example showing hierarchical design. (a) The schematic of a half-adder, the subschematic of cell HADD. (b) A schematic symbol for the half adder. (c) A schematic that uses the half-adder cell. (d) The hierarchy of cell HADD.

Figure from *Application-Specific Integrated Circuits*, Smith, Addison-Wesley, 1997
Spring 2002, Lecture 05

Altera MAX+PLUS II Overview

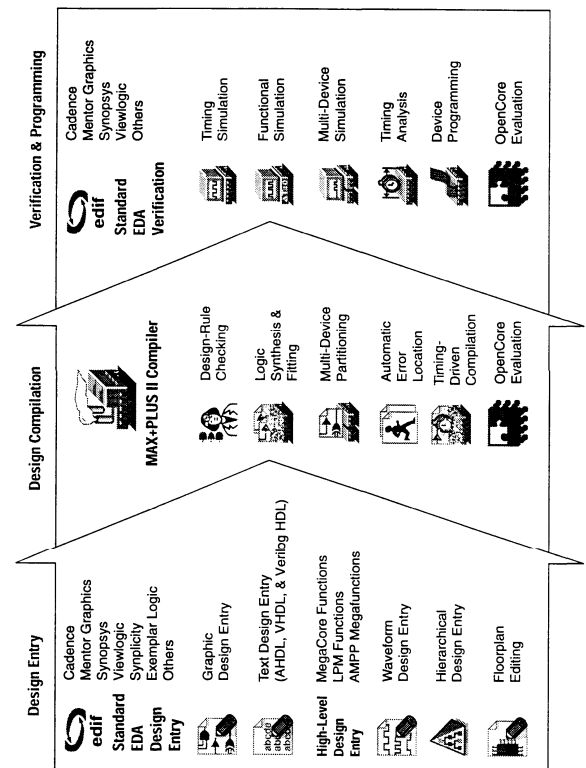


Figure from Altera technical literature

The “Chiptrip” Tutorial Example

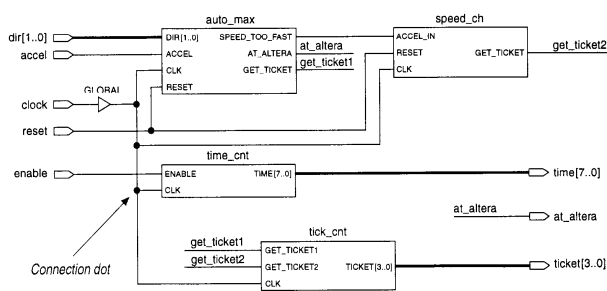


Figure from Altera technical literature

■ Simulates an auto driving around town

- auto_max — AHDL state machine that keeps track of location of auto and acceleration at that point in time, gives ticket if you accelerate on small roads
- speed_ch — waveform state machine that gives ticket if you accelerate for a second time
- tick_cnt — counter that counts tickets
- time_cnt — AHDL counter that keeps track of time taken to reach Altera

Graphic Editor

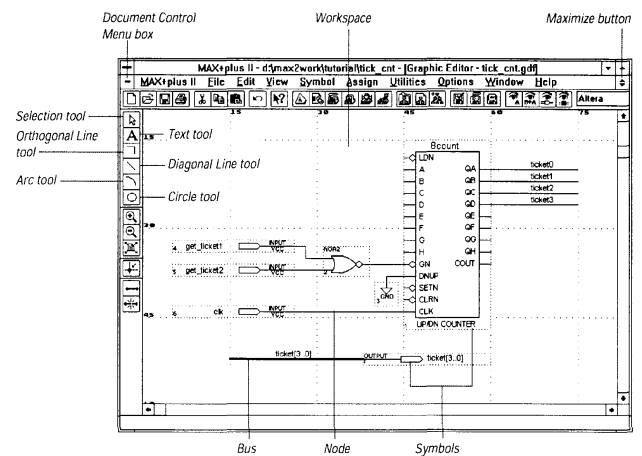


Figure from Altera technical literature

- Enter cells from various predefined component libraries, or user-defined cells
- Connect cells using nets, buses, or name
- “Smart” selection tool — automatically becomes proper tool for task at hand

Waveform Editor

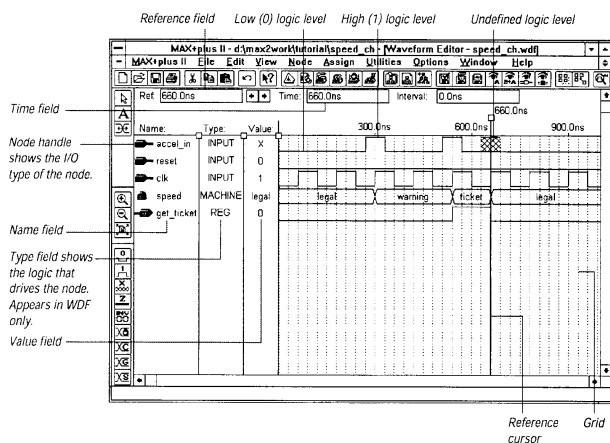
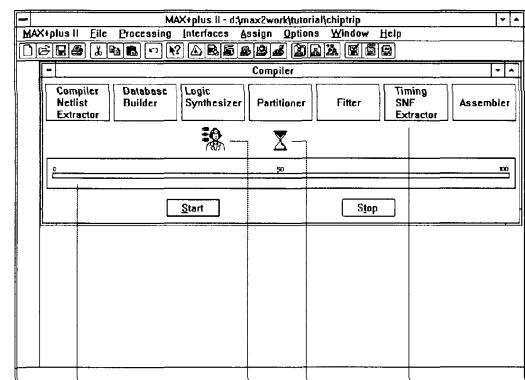


Figure from Altera technical literature

- Can contain logical and state machine inputs; combinational, registered, and state machine outputs; and “buried” nodes to help define desired outputs
- Can specify state names for state machines

■ Can compare desired and actual outputs

Compiler



The progress bar indicates percent completion during processing.

The Design Doctor utility is turned on.

The Timing SNF Extractor module is turned on.

The hourglass flips as the Compiler processes the project.

Figure from Altera technical literature

- Checks for design entry errors, builds a single large flat database
- Logic synthesis to minimize resource usage (see Assign/Global Project Logic Synthesis), partitioner and fitter to match to available devices

Simulation Results

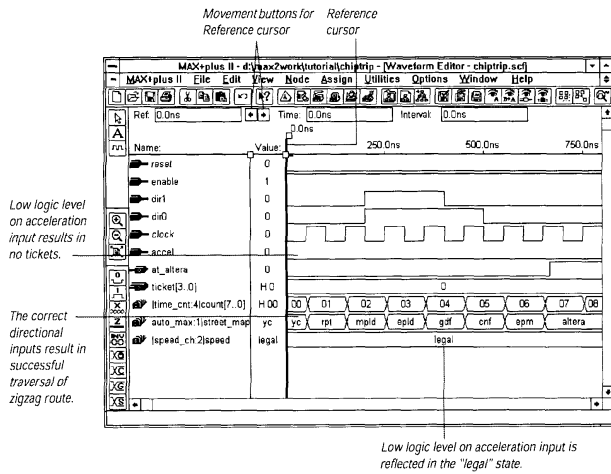


Figure from Altera technical literature

- Use waveform editor to specify simulation inputs
- Simulate, then view results in waveform editor (as shown above)
 - Simulate individual or grouped nodes (particularly good for state machines)

Floorplan Editor

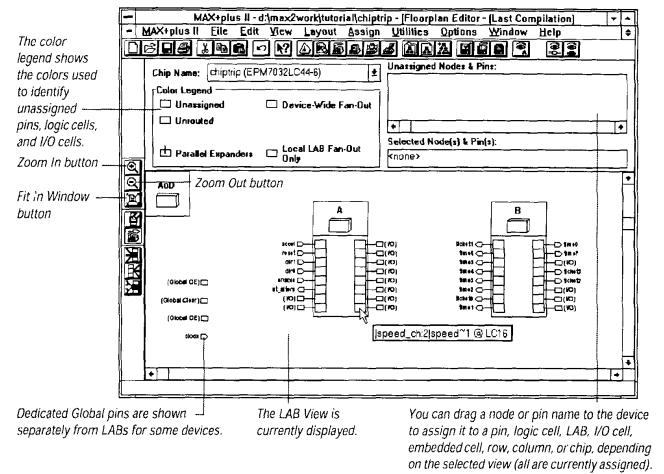


Figure from Altera technical literature

- Device view shows pins, LAB view shows LABs, equations, I/O, and routing
- Can use to edit assignments
- After compilation, get information on most congested area of chip, number of expanders used within each LAB