

Design Entry

- Computer Aided Design (CAD) tools typically support both graphical schematic capture as well as textual design entry (e.g., AHDL, VHDL)
 - Documentation, design, simulation, verification
- A circuit *schematic* shows the interconnection of structural elements that make up a circuit
 - Captures only interconnection; behavior specified separately
 - The electronic (usually ASCII) version of that schematic is called a *netlist*
- Schematic capture
 - Direct entry of the circuit schematic
 - More “bookkeeping” than “automation”

1

Spring 2006, Lecture 06

Graphic Editor

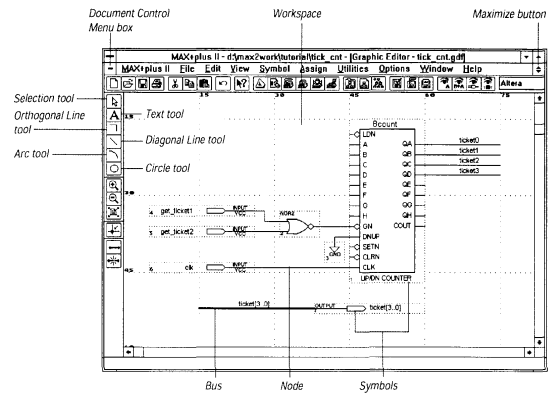


Figure from Altera technical literature

- 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

2

Spring 2006, Lecture 06

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:

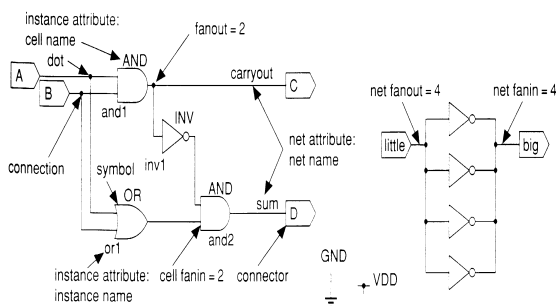


Figure from *Application-Specific Integrated Circuits*, Smith, Addison-Wesley, 1997

3

Spring 2006, Lecture 06

Cell Library

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

4

Spring 2006, Lecture 06

Names & Symbols

- 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
- 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*

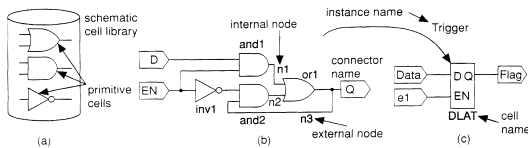


Figure from *Application-Specific Integrated Circuits*, Smith, Addison-Wesley, 1997

5

Spring 2006, Lecture 06

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
- 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

6

Spring 2006, Lecture 06

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

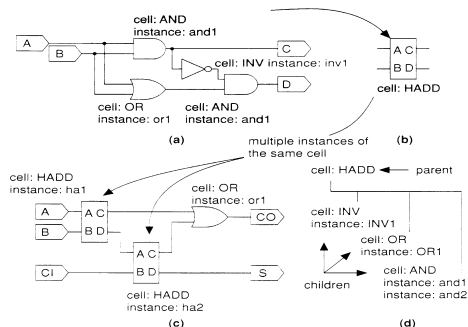


FIGURE 9.3 Schematic example showing hierarchical design. (a) The schematic of a half-adder, the subschematic of cell HADD. (b) A schematic symbol for the half adder. (c) A schematic that uses the half-adder cell. (d) The hierarchy of cell HADD.

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

7

Spring 2006, Lecture 06

Altera MAX+PLUS II Overview

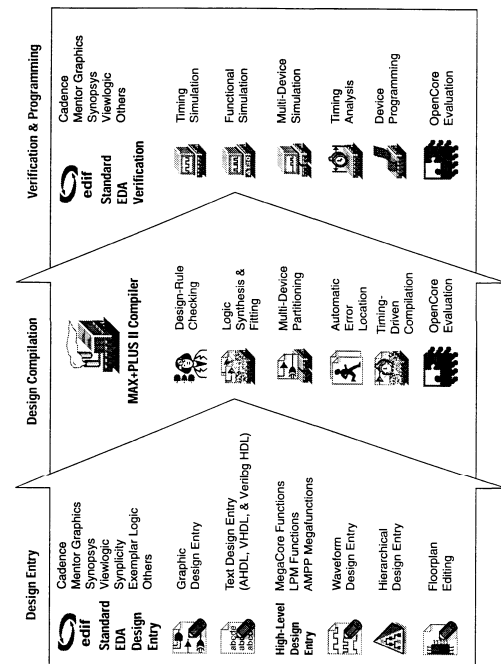


Figure from Altera technical literature

8

Spring 2006, Lecture 06

The “Chiptrip” Tutorial Example

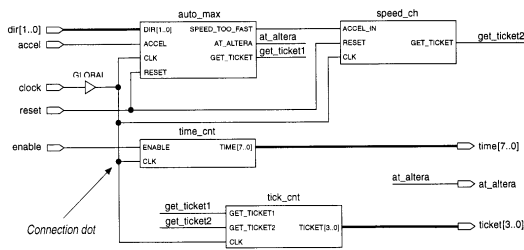


Figure from Altera technical literature

- 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

9

Spring 2006, Lecture 06

Graphic Editor

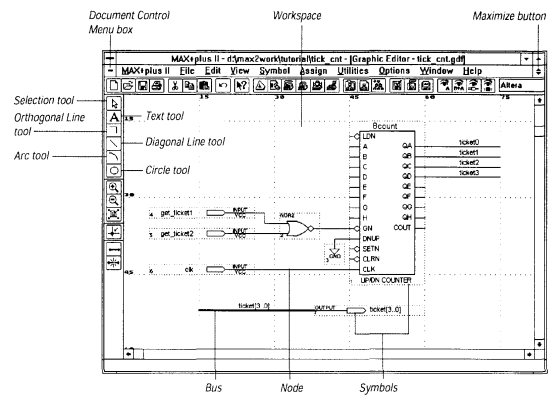


Figure from Altera technical literature

- 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

10

Spring 2006, Lecture 06

Waveform Editor (for Design Entry)

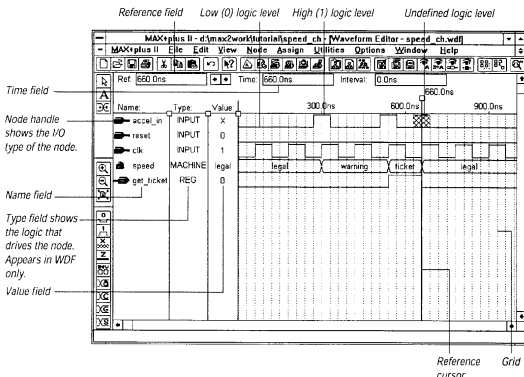


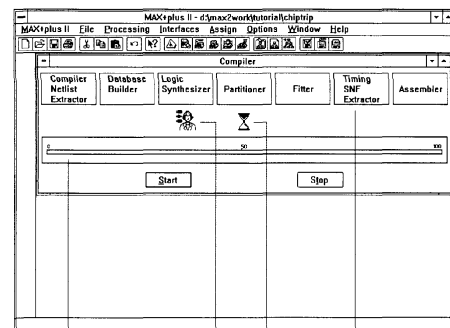
Figure from Altera technical literature

- 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

11

Spring 2006, Lecture 06

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.

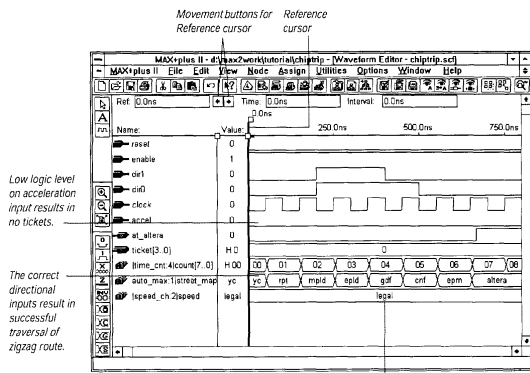
Figure from Altera technical literature

- 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

12

Spring 2006, Lecture 06

Waveform Editor + Simulation

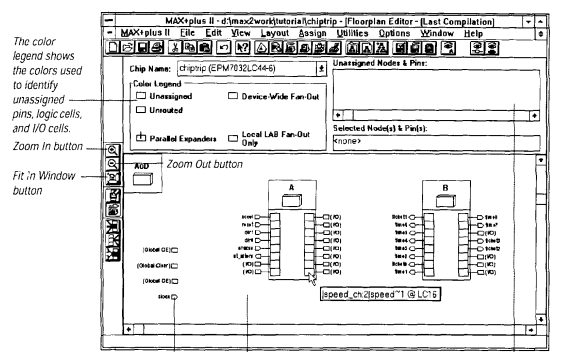


Low logic level on acceleration input is reflected in the "legal" state.

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



Dedicated Global pins are shown separately from LABs for some devices.

The LAB View is currently displayed.

You can drag a node or pin name to the device to assign it to a pin, logic cell, LAB, I/O cell, embedded cell, row, column, or chip, depending on the selected view (all are currently assigned).

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