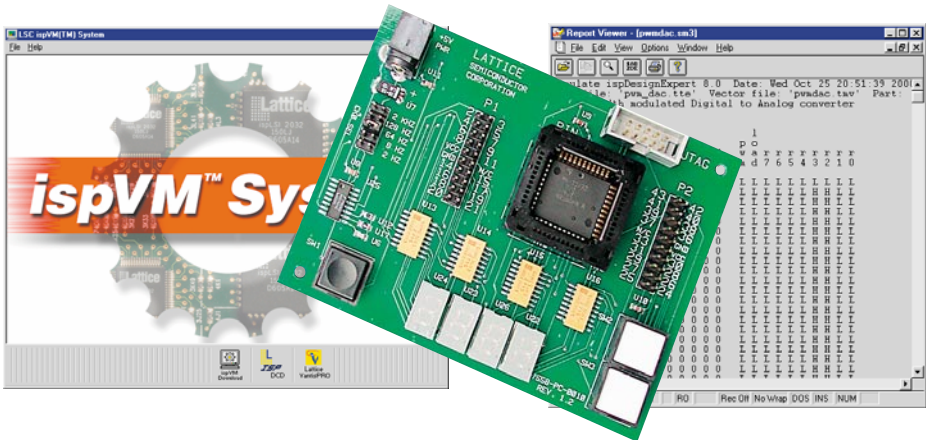

Der eigene Schaltkreis - PLD-Einsteiger-Set

Teil 2

Die PLD-Chips von Lattice erlauben in Zusammenarbeit mit der zugehörigen Entwicklungs- und Compiler-Software das einfache Kreieren eigener, auch komplexer Schaltkreisfunktionen aus Grundgattern und damit individuelle Schaltkreisapplikationen mit minimalem Entwicklungsaufwand. Nach den PLDs stellen wir im zweiten Teil das zugehörige Entwicklungssystem vor.

Das „ispDesignEXPERT"-Entwicklungssystem

Der Kern der mit dem PLD-Einsteiger-Set gelieferten Software ist das „ispDesignEXPERT"-Entwicklungssystem, mit dem der Designentwurf, die Simulation und die Implementierung der eigenen Schaltung erfolgt. Es besteht aus mehreren Komponenten, die eng miteinander arbeiten und nahezu alle von einem so genannten Project Navigator (Abbildung 7) direkt erreichbar sind. Lediglich die eigentliche In-System-Programming-Software „ispVM"arbeitet getrennt.

Im Project Navigator erfolgt das Anlegen neuer Projekte und der gesamte Projektentwicklungsablauf bis hin zur Compilierung zum JEDEC-File für die Programmer-Software. Je nach gewähltem PLD und gewünschter Entwicklungsstrategie erfolgt die Ablaufsteuerung der Entwicklung nach vorgebenen Algorithmen, sodass der Entwickler nur den Prozess-Listen folgen und diese abarbeiten muss.

Der Schaltungsentwurf

Allem voran geht jedoch der eigentliche Schaltungsentwurf. Dieser kann über Schaltungssynthesewerkzeuge wie von Exemplar Logic, Model Technology, Synopsis, Synplicity und Viewlogic erfolgen. Alternativ dazu ist der Entwurf auch per inte-

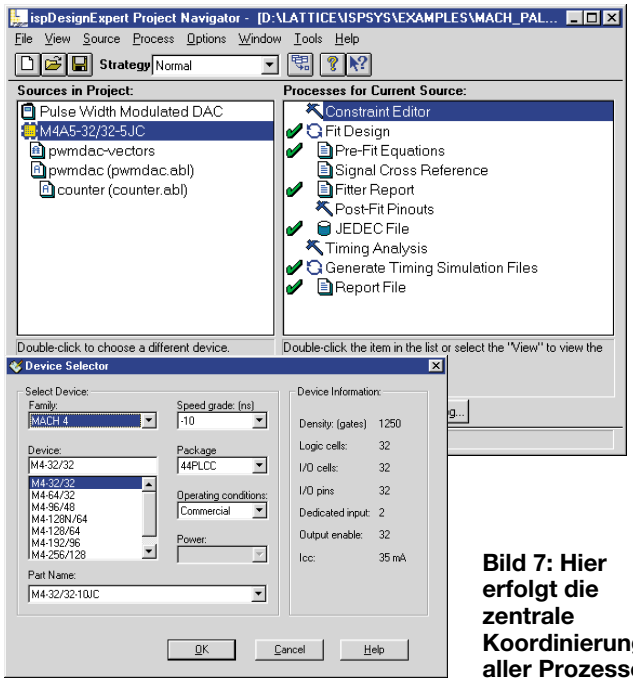

Bild 7: Hier erfolgt die zentrale Koordinierung aller Prozesse.

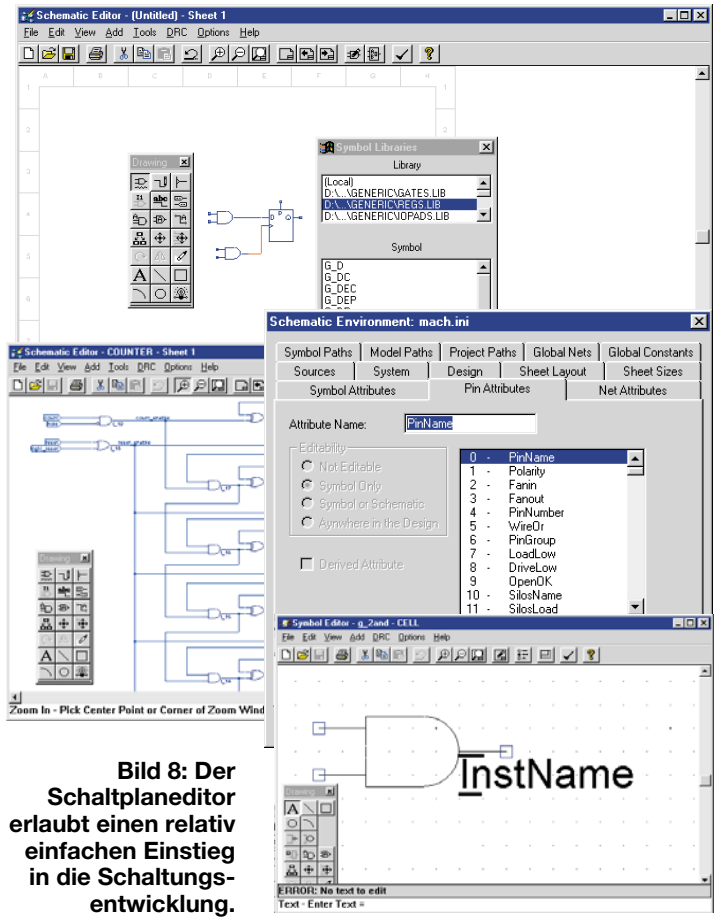

Bild 8: Der Schaltplaneditor erlaubt einen relativ einfachen Einstieg in die Schaltungsentwicklung.

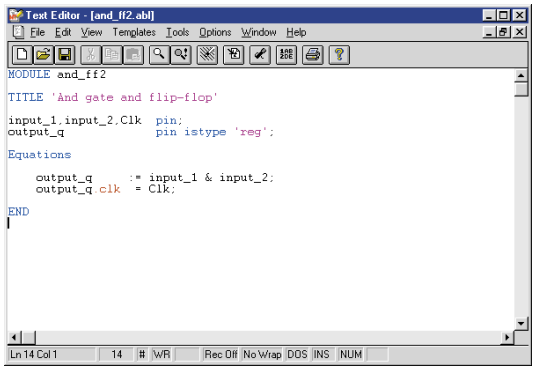

Bild 9: Schaltungseingabe über ABEL-Hochsprache

griertem Schaltplaneditor (Schematic Entry von Viewlogic/Lattice, Abbildung 8) oder per ABEL-Hochsprachen-Eingabe (ABEL Entry, Abbildung 9, Eingabe über Boolesche Gleichungen, Wahrheitstabellen usw.) möglich.

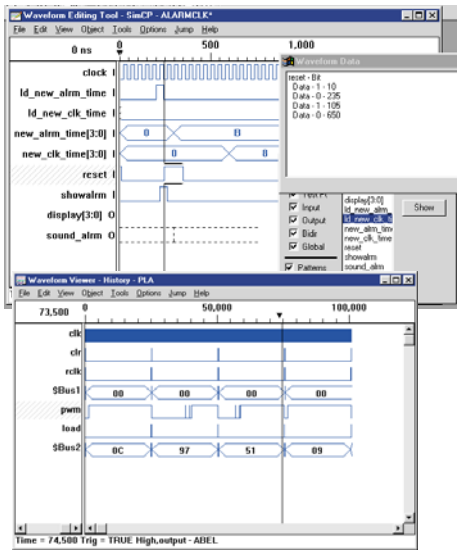

Bild 10: Timing-Diagramme und -Editoren erleichtern den Test der Schaltung.

Der Schaltplaneditor arbeitet mit sehr leistungsfähigen grafischen Werkzeugen, verfügt über eine große Bauteilbibliothek und man kann beliebige eigene Bauteile kreieren. Der Einsteiger wird diese Art des Schaltungsentwurfs sicher bevorzugen, dennoch muss man sich an zahlreiche strenge Entwurfsregeln halten - es geht ähnlich zu wie bei einer professionellen Leiterplattenentwicklung.

Simulations- und Testwerkzeuge

Die jeweils fertige Schaltung wird anschließend unter Echtzeitbedingungen getestet, um das richtige Timing innerhalb des Chips zu kontrollieren bzw. erst zu erarbeiten. Dabei dienen Timing-Diagramme (Abbildung 10) und Performance-Tests (Abbildung 11) zur Beurteilung. Besonders interessant ist hier der Waveform-Editor, der die Einstellung vielfältiger Timingbedingungen und die sofortige Simulation mit visueller Darstellung erlaubt. Alle Simulationsläufe sind über Reports nachvollzieh- und kontrollierbar (siehe Titelbild).

Fitting und Compilieren

Hinter Fitting verbirgt sich der gesamte Prozess der Anpassung des Designentwurfs an den gewünschten PLD. Dies wird zum Teil bereits im Simulationsprozess vollzogen, indem Timings des Schaltungsentwurfs an den PLD angepasst werden. Dazu zählen aber auch Prozesse beim anschließenden Compilieren der Schaltung, die eine optimale Anpassung des Schaltungsentwurfs an die Hardware des PLDs realisieren. So wird hier auch der Pinout-Entwurf nebst Kontrolle und Simulation vorgenommen. Der Entwickler kann im Rahmen der PLD-Hardwarestruktur individuelle Pinouts nach eigenen Wünschen und Platinenlayout-Anforderungen entwickeln, dies aber auch weitgehend dem Compiler überlassen (Platzieren und Routen, Abbildung 12).

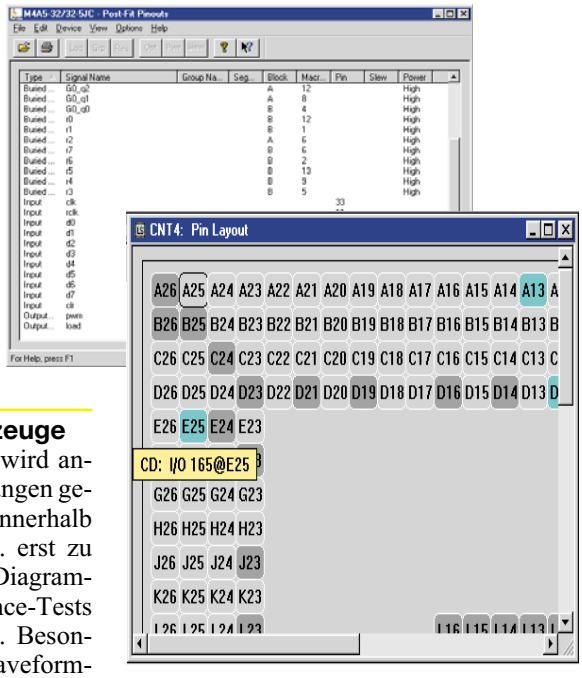

Bild 12: Das Routen und die Pinbelegung erfolgen halbautomatisch.

Am Ende des Compilerlaufs (Abbildung 13) steht ein Standard-JEDEC-File zur Verfügung, das in den PLD geladen werden kann. Auch dieses File ist bei Bedarf nochmals einer Simulation unterziehbar. Auch hier sind alle Prozesse durch Reports nachvollziehbar.

Das Programmieren des Chips

Die Kommunikation zwischen PC und PLD erfolgt über das Tool „ispVM". Hierüber sind sowohl einzelne PLDs als auch mehrere seriell in einer „Daisy Chain"-Anordnung oder (über den Turbo-Mode) mehr als 100 PLDs parallel programmierbar.

Die Software kontrolliert die ordnungsgemäße Verbindung zwischen PC und PLD (dieser kann sich auch in der Anwendungsschaltung befinden) und programmiert nach Auswahl des gewünschten JEDEC-Files den PLD. Auch hier unterstützen mehrere komplexe Kontroll- und File-Erstellungswerkzeuge den Prozess und machen den Bediener auf Unstimmigkeiten zwischen Chip und Programm aufmerksam.

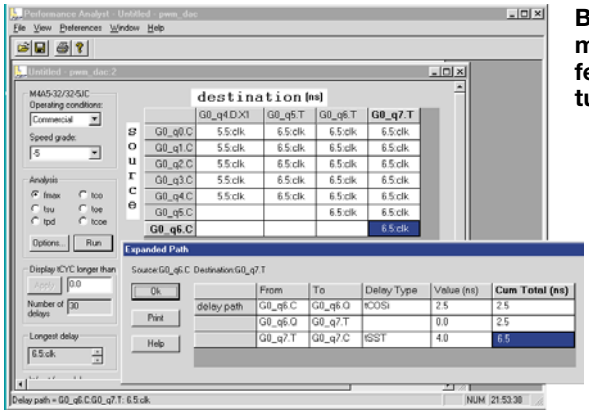

Bild 11: Performance-Test der fertigen Schaltung

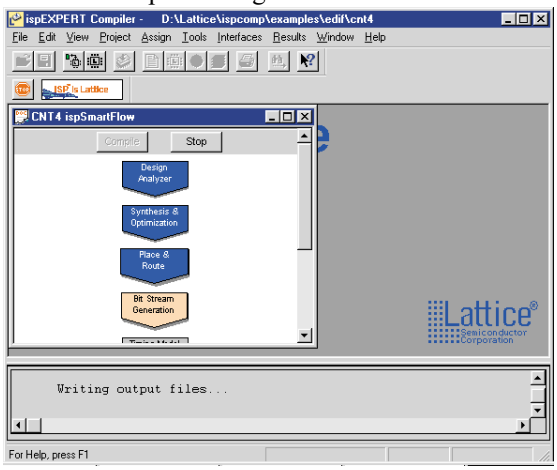

Bild 13: Der Compiler bei der Arbeit