Overview

InTime is an expert software that helps engineers achieve the highest possible performance for their FPGA designs.

Driven by machine learning and analytics, InTime combines human experience with built-in intelligence to analyze and predict optimal design parameters. InTime learns from every result and improves its predictions with more data.

ML-Based Approach

InTime uses a machine learning-driven approach to optimize your design. It analyzes design characteristics to suggest optimization parameters ranging from tool options to constraints. With machine learning, the more builds InTime runs, the smarter it becomes.

Ease of use – Just click “Start”

InTime provides one-click flows called “recipes” to optimize your design automatically. Choose from a variety of recipes that target synthesis, placement and congestion problems.

Runs on Enterprise Build Environments

InTime’s “Run Target” setting enables builds to run locally on a computer, distributed across multiple machines in your network, or on a compute cloud. Configure it for workload management software such as Load Sharing Facility (LSF) or Oracle/Sun Grid Engine (SGE). InTime also works on Amazon Web Services.

Wrapper around FPGA tools

InTime is a wrapper around major FPGA tools. It opens project files such as DCP, PRJX, QPF, XISE and XPR directly. Designs created as Tcl scripts can also be read directly. After optimization, users can export results in formats like programming files, Tcl scripts, netlists or project files.
InTime’s analytics lets you compare multiple builds to highlight similarities between builds, pinpoint design bottlenecks and root-cause failures. Visualize and track important data such as the total number of critical paths, high fanout nets, die crossings and relationships between various design goals like Total Negative Slack (TNS), Worst Slack (WS), Area and Power.
**Recipe Lists**

A "Recipe" is an InTime flow used to optimize a design. Recipes are grouped into four categories: 'Learning', 'Last-Mile', 'General' and 'Advanced' and are customized for each FPGA tool.

<table>
<thead>
<tr>
<th>Recipe Name</th>
<th>ISE</th>
<th>Quartus II</th>
<th>Quartus Prime Pro</th>
<th>Libero</th>
<th>Vivado</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Learning</td>
<td></td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Generates initial strategies by correlating it with other designs in the database</td>
</tr>
<tr>
<td>InTime Default</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Performs calibration, exploration and optimization</td>
</tr>
<tr>
<td>Deep Dive</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Performs extensive analysis of design results for predictions</td>
</tr>
<tr>
<td>SSI Exploration</td>
<td></td>
<td></td>
<td></td>
<td>Y</td>
<td>Y</td>
<td>Explores different SSI (multiple die) optimizations in your design</td>
</tr>
<tr>
<td>Clock Exploration</td>
<td>Y</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Varies specific clock constraints in compilation stages to improve performance</td>
</tr>
<tr>
<td>Fanout Exploration</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td></td>
<td></td>
<td>Discovers appropriate fanout limits for high fanout nets in the design</td>
</tr>
<tr>
<td>Synthesis Exploration</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td></td>
<td></td>
<td>Determines the optimal synthesis setting combinations for the design</td>
</tr>
</tbody>
</table>

**Last-Mile**

- **Auto Placement**
  - ISE: Y
  - Quartus II: Y
  - Quartus Prime Pro: Y
  - Libero: Y
  - Vivado: Y
  - Description: Adjusts placement constraints to improve the worst timing paths

- **Seeded / Effort Level Exploration**
  - ISE: Y
  - Quartus II: Y
  - Quartus Prime Pro: Y
  - Libero: Y
  - Vivado: Y
  - Description: Explores different effort levels and seeds on the best effort level results

- **Placement Seed Exploration**
  - ISE: Y
  - Quartus II: Y
  - Quartus Prime Pro: Y
  - Libero: Y
  - Vivado: Y
  - Description: Explores placement seeds

- **Placement / Router Effort Level**
  - ISE: Y
  - Quartus II: Y
  - Quartus Prime Pro: Y
  - Libero: Y
  - Vivado: Y
  - Description: Explores placement and router effort levels

- **Clock Margin Exploration**
  - ISE: Y
  - Quartus II: Y
  - Quartus Prime Pro: Y
  - Libero: Y
  - Vivado: Y
  - Description: Varies clock uncertainties to trigger variations in timing.

- **Extra Opt Exploration**
  - ISE: Y
  - Quartus II: Y
  - Quartus Prime Pro: Y
  - Libero: Y
  - Vivado: Y
  - Description: Explores incremental placement, physical synthesis and routing optimizations

- **Placement Exploration**
  - ISE: Y
  - Quartus II: Y
  - Quartus Prime Pro: Y
  - Libero: Y
  - Vivado: Y
  - Description: Stimulates placement changes to improve performance

- **Map Seed Exploration**
  - ISE: Y
  - Quartus II: Y
  - Quartus Prime Pro: Y
  - Libero: Y
  - Vivado: Y
  - Description: Explores the effect of map seeds on the design goal

- **Region Exploration**
  - ISE: Y
  - Quartus II: Y
  - Quartus Prime Pro: Y
  - Libero: Y
  - Vivado: Y
  - Description: Examines instance placements to find more effective physical locations

**General**

- **Just Compile My Design**
  - ISE: Y
  - Quartus II: Y
  - Quartus Prime Pro: Y
  - Libero: Y
  - Vivado: Y
  - Description: Compiles the active revision in your project

- **Compile with a Guide File**
  - ISE: Y
  - Quartus II: Y
  - Quartus Prime Pro: Y
  - Libero: Y
  - Vivado: Y
  - Description: Uses a compiled netlist as a reference to compile the current design

- **Rerun Strategies**
  - ISE: Y
  - Quartus II: Y
  - Quartus Prime Pro: Y
  - Libero: Y
  - Vivado: Y
  - Description: Reruns all selected strategies

- **Rerun Best Results**
  - ISE: Y
  - Quartus II: Y
  - Quartus Prime Pro: Y
  - Libero: Y
  - Vivado: Y
  - Description: Reruns the best results from the current history

**Advanced**

- **Auto-Pilot**
  - ISE: Y
  - Quartus II: Y
  - Quartus Prime Pro: Y
  - Libero: Y
  - Vivado: Y
  - Description: Customizes and automates sequences of recipes based on criteria like TNS / WNS

- **Custom Flow**
  - ISE: Y
  - Quartus II: Y
  - Quartus Prime Pro: Y
  - Libero: Y
  - Vivado: Y
  - Description: Uses strategies specified by the user to compile

---

**System Requirements**

**InTime & License Server (Private Cloud / LSF/ SGE)**

- RAM: Minimum required by your FPGA tool
- Disk space: At least 430 MB for installation, 40 GB for build files
- Processor: Intel i3 CPU or similar
- OS: 64-bit Windows 7 onwards / Ubuntu 12.04 / RedHat Enterprise 6/7 / other compatible Linux distributions
- Java: Java Runtime Environment (JRE 1.6 and above) or OpenJDK
- Networking: Available network ports e.g. 39940, 39941, 39942 (For multiple machines)

**InTime Software**

- RAM: Minimum 2 GB
- Disk space: At least 500 MB for installation; Temporary disk or scratch space up to 20 GB
- OS: 64-bit Windows 7 onwards / Ubuntu 12.04 / RedHat Enterprise 6/7 / other compatible Linux distributions
- Java: Java Runtime Environment (JRE 1.6 and above) or OpenJDK
- Networking: Available network ports e.g. 39940, 39941, 39942 (For multiple machines)

**Supported FPGA Tools**

- Altera Quartus-II 13.0 onwards, Quartus Prime Standard/Pro Edition 15.1 onwards
- Libero 12.4 onwards
- Xilinx ISE 14.7 onwards, Vivado 2015 onwards

---

**About Plunify**

Plunify helps chip design companies optimize FPGA designs with big data and machine learning. Plunify is based in Singapore and in the United States. InTime is a registered trademark of Plunify Pte Ltd.

Email: tellus@plunify.com
Singapore: 10 Anson Road, #27-15, Singapore 079903
United States: 165 University Ave, Palo Alto, CA 94301, USA