Overview

InTime is an expert software that solves FPGA timing and optimization problems with massive compute power and machine learning. FPGA tools such as Vivado and Quartus, contain multiple advanced settings and parameters that can provide massive improvements on the FPGA design performance. InTime has built-in intelligence to analyze and determine optimal combinations of settings and placement strategies, with the highest chances of meeting design performance targets. InTime learns from every result and constantly improves its database.

By intelligently harnessing compute resources, InTime enables FPGA engineers to solve design problems and get to production silicon significantly faster than before.

Automatic generation of build strategies

InTime’s signature database stores knowledge from over 40,000 CPU hours of benchmarking across multiple tool versions and device families. This enables InTime to tailor tool settings and constraints for specific design conditions.

Ease of use - just click “Start”

Just click “Start Recipe”. InTime learns from your previous build results and improves them based on device, design and tool characteristics. With machine learning, the more builds InTime does, the smarter it gets.

Flexible build environments

In a multi-machine configuration, a designated InTime “control” server allocates machines for every “job” (consisting of one or more builds). InTime communicates with the “worker” machines, encrypting and transferring files/results, enabling builds to be run in parallel, taking advantage of available compute resources.
A "Recipe" is a method used by InTime to optimize a design. Recipes can be grouped into four categories: 'Learning', 'Last-Mile', 'General' and 'Advanced'.

Those in the 'Learning' category are the recipes that Plunify recommends when the user runs designs that are new to InTime. These recipes are intended to perform initial analysis and calibration of design settings and constraints. For example, the InTime Default recipe tweaks the value of all the available compiler settings based on learning done on past results. Meanwhile, recipes in the 'Last-Mile' category perform more specific optimizations when results have been sufficiently improved by the Learning recipes. For example, the Placement Seed Exploration recipe changes the placement seed of compiler settings only. Last but not least, recipes under 'General' and 'Advanced' are more for usability purposes.

InTime allows users to control the execution of InTime recipes using Tcl scripting. For example, you can now make InTime execute the InTime Default recipe followed by the Explorer recipe and finally one or more of the Last-Mile recipes. InTime sets the parent revision for each recipe according to metrics that you can specify, for example, TNS.

### Sample Tcl Script

```tcl
# get worst setup slack right after placement
open_design "calibrate_1_place"
# terminate this strategy if the slack is too high
if [get_slack] < -2 ] { if -kill "calibrate_3"
}
close_design
# move on to the next strategy
```

Figure 3: Sample Tcl Script
Advanced result analysis

InTime’s result analytics enables users to compare multiple builds at a glance to find similarities between passing builds and pinpoint causes of failure. InTime visualizes and tracks data such as total number of critical paths, interconnects, and relationships between various design goals like Total Negative Slack (TNS), Worst Slack (WS), Area and Power.

Figure 4: (clockwise from top left) TNS of different compilations, interconnects chart, machine learning improvements, TNS/WS distribution

System Requirements

<table>
<thead>
<tr>
<th>Run on Private Cloud</th>
<th>Run Standalone / Local</th>
</tr>
</thead>
<tbody>
<tr>
<td>InTime Server (License)</td>
<td>InTime</td>
</tr>
<tr>
<td>RAM: Minimum 2GB</td>
<td>RAM: Minimum required by your FPGA tool</td>
</tr>
<tr>
<td>Disk space: At least 430MB for installation, 40GB for build files</td>
<td>Disk space: At least 200MB for installation</td>
</tr>
</tbody>
</table>

Processor: Intel i3 CPU or similar

OS: 64-bit Windows / Ubuntu 12.04 / RedHat Enterprise 5+ / other compatible Linux distributions

Java: Java Runtime Environment (JRE 1.6 and above)

Networking: Available network ports e.g. 39940, 39941, 39942 (For multiple machines)

Licensing

- Annual license priced according to the maximum number of concurrent builds

Supported FPGA Tools

- Altera Quartus-II 13.0 onwards, Quartus Prime Standard/Pro Edition 15.1 onwards
- Xilinx ISE version 11.1 onwards, Vivado 2014.2 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. Vivado and ISE are registered trademarks of Xilinx, Inc. InTime is a registered trademark of Plunify Pte Ltd