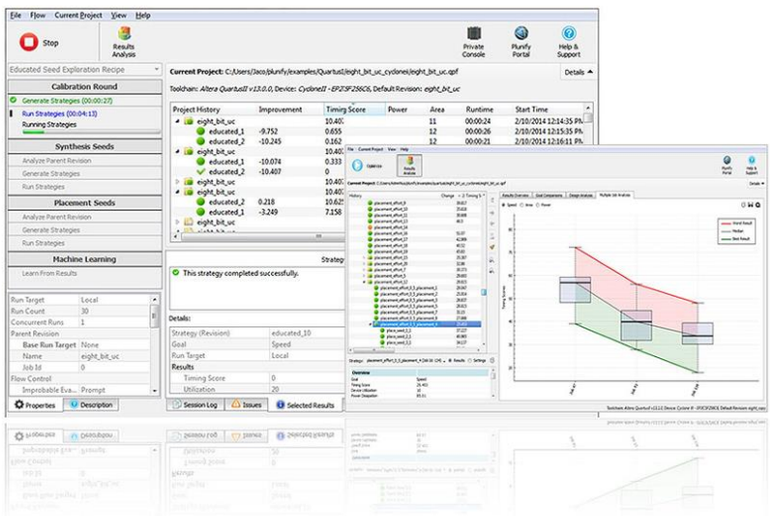
InTime
Datasheet

Overview

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

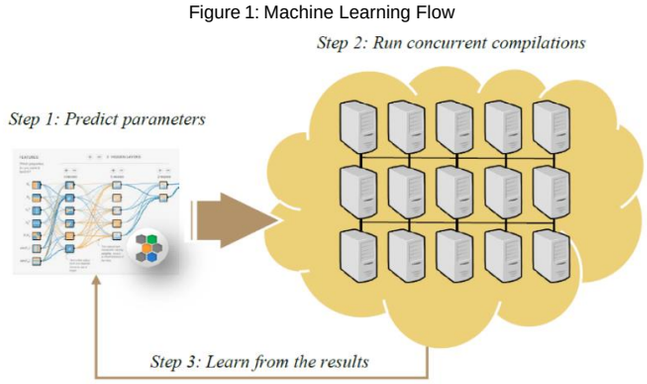
Driven by machine learning and analytics, InTime combines human experience with built-in intelligence to analyze and predict optimal design parameters. InTime learns from every result and improves its predictions with more data.



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.

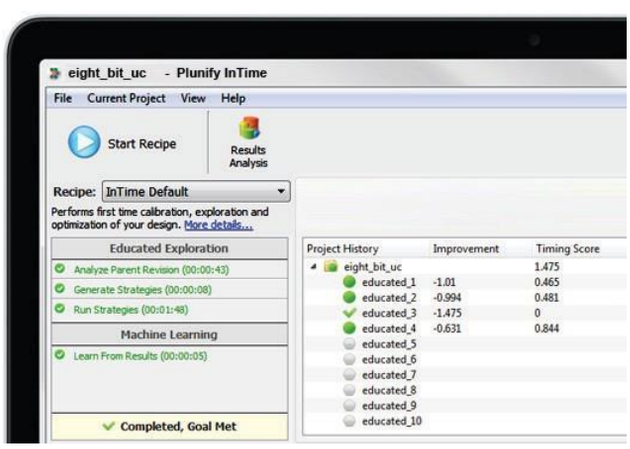
Figure 1: Machine Learning Flow



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.

Figure 2: InTime Graphical User Interface



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.

Advanced Results Analysis

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.

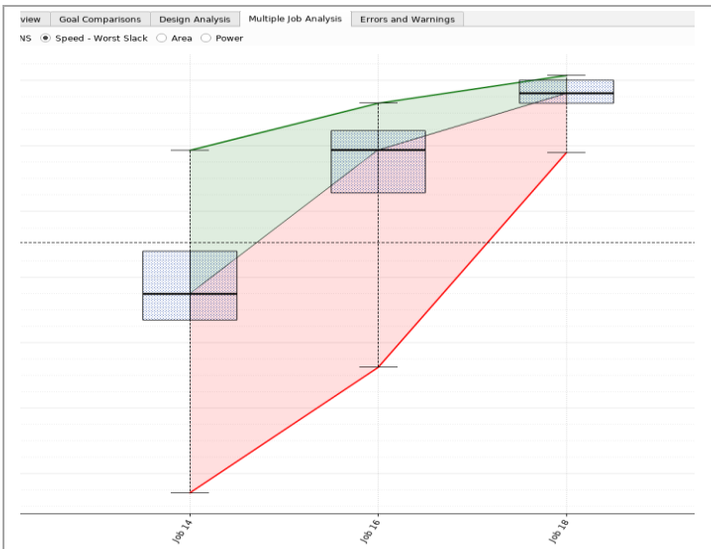


Figure 3: Multiple Jobs Analysis

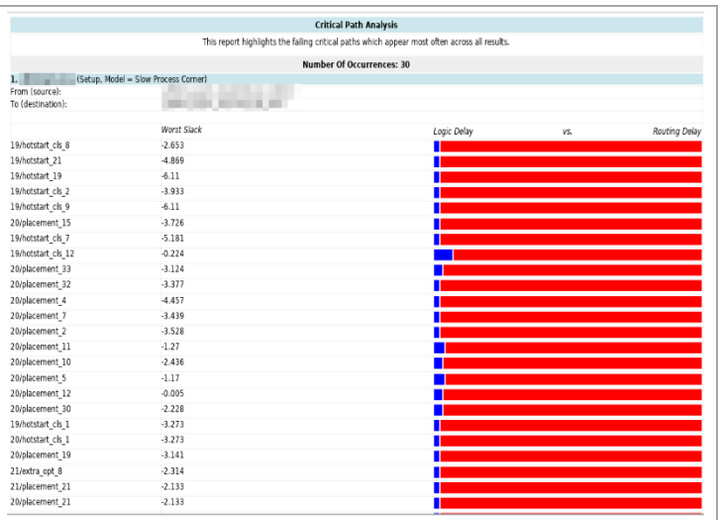
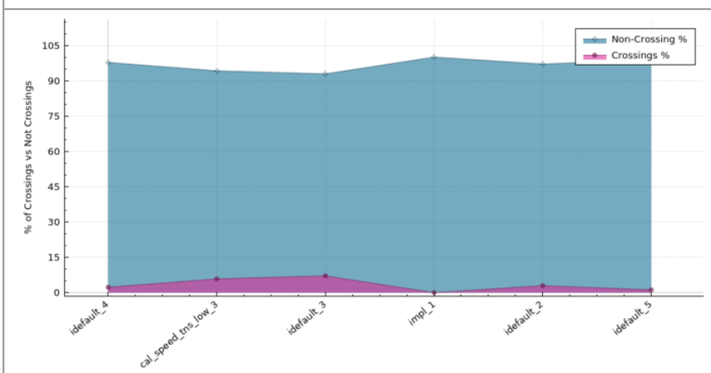


Figure 4: Critical Path Analysis



Number of Crossings Versus Non-crossing Paths

Strategy	Crossings	Non-Crossings	Crossing(%)
idefault_4	183	7835	2.336
cal_speed_tns_low_3	372	6007	6.193
idefault_3	350	4563	7.67
impl_1	0	4	0
idefault_2	295	9705	3.04

Figure 5: Multi-die Crossing Analysis

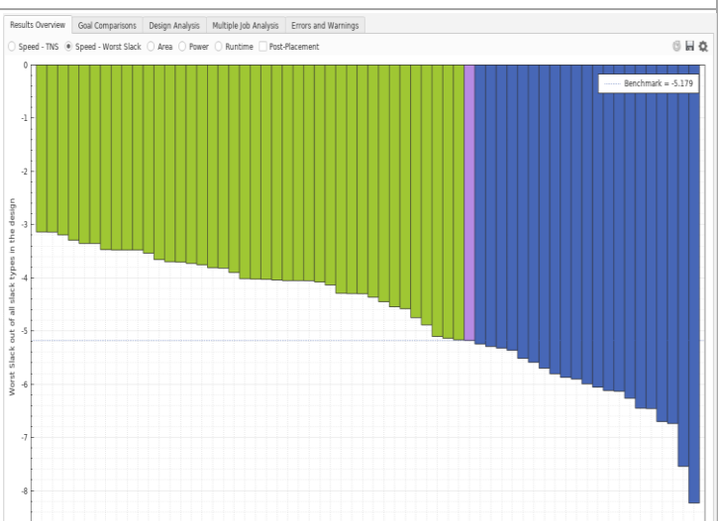


Figure 6: Results Chart (TNS/WNS/Area/Power)

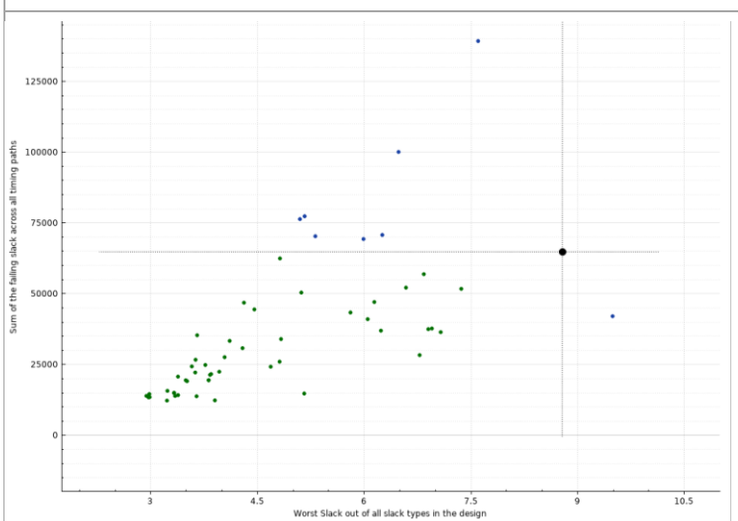


Figure 7: Goal Comparison Chart

Warnings

Total Occurrences	Occurrences Rate	Step	Message ID	Examples	M
98	80.00%	opt_design	Runs 36-537	WARNING: File F:/40/hotstart_impl_1_pre_opt_design_action.tcl is not part of fileset utils_1, but has specified as a Tcl hook script for run(s) hotstart_impl_1. This file will not be handled as part of the project for archive and other project based functionality.	cal_
106	73.33%	place_design	Vivado 12-180	WARNING: No cells matched 'get_cells -filter { REF_NAME =~ LUT* } -of_object {{regs/addr_reg/C --> regs/mem_regfile_reg_0_31_0_0/SP/ADR1}}'.	
10	66.67%	route_design	Route 35-39	CRITICAL WARNING: The design did not meet timing requirements. Please run report_timing_summary for detailed reports.	
12	80.00%	reporting	Timing 38-282	CRITICAL WARNING: The design failed to meet the timing requirements. Please see the timing summary report for details on the timing violations.	
2	13.33%	route_design	Route 35-328	WARNING: Router estimated timing not met.	
58	26.67%	synth_design	Synth 8-3332	WARNING: Sequential element (prgmcntr/pc_reg_rep) is unused and will be removed from module eight_bit_uc.	
4	26.67%	opt_design	Constraints 18-5210	WARNING: No constraints selected for write.	cal_
2	13.33%	opt_design	Vivado_Tcl	WARNING: -remap and -aggressive_remap	cal_

Figure 8: Errors and Warnings Analysis

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.

	Recipe Name	ISE	Quartus II	Quartus Prime Pro	Libero	Vivado	Description
Learning	Hot Start		Y	Y	Y	Y	Generates initial strategies by correlating it with other designs in the database
	InTime Default	Y	Y	Y	Y	Y	Performs calibration, exploration and optimization
	Deep Dive	Y	Y[(1)]	Y		Y	Performs extensive analysis of design results for predictions
	SSI Exploration					Y	Explores different SSI (m*ultiple die)* optimizations in your design
	Clock Exploration				Y		Varies specific clock constraints in compilation stages to improve performance
	Fanout Exploration		Y	Y			Discovers appropriate fanout limits for high fanout nets in the design
	Synthesis Exploration						Determines the optimal synthesis setting combinations for the design
Last-Mile	Auto Placement		Y	Y			Adjusts placement constraints to improve the worst timing paths
	Seeded / Effort Level Exploration		Y	Y			Explores different effort levels and seeds on the best effort level results
	Placement Seed Exploration	Y	Y	Y	Y		Explores placement seeds
	Placement / Router Effort Level		Y	Y			Explores placement and router effort levels
	Clock Margin Exploration					Y	Varies clock uncertainties to trigger variations in timing.
	Extra Opt Exploration					Y	Explores incremental placement, physical synthesis and routing optimizations
	Placement Exploration				Y	Y	Stimulates placement changes to improve performance
	Map Seed Exploration	Y					Explores the effect of map seeds on the design goal
	Region Exploration				Y		Examines instance placements to find more effective physical locations
General	Just Compile My Design	Y	Y	Y	Y	Y	Compiles the active revision in your project
	Compile with a Guide File					Y	Uses a compiled netlist as a reference to compile the current design
	Rerun Strategies	Y	Y	Y	Y	Y	Reruns all selected strategies
	Rerun Best Results	Y	Y	Y	Y	Y	Reruns the best results from the current history
Advanced	Auto-Pilot		Y	Y	Y	Y	Customize and automate sequences of recipes based on criteria like TNS / WNS
	Custom Flow	Y	Y	Y	Y	Y	Uses strategies specified by the user to compile

[1) Not Supported for Arria10 devices in Quartus II/Prime Standard Edition

System Requirements

InTime & License Server (Private Cloud / LSF/ SGE)	InTime Software
RAM: Minimum 2GB Disk space: At least 430MB for installation, 40GB for build files	RAM: Minimum required by your FPGA tool Disk space: At least 500MB for installation; Temporary disk or scratch space up to 20GB
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)	

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