

# Plunify 사 InTime software

**A. 목적 : (Timing and Area optimization) FPGA 설계 최적화 및 FPGA 타이밍 문제 해결**

**B. 구분 : (FPGA 설계시 timing and Area 최적화)**

**InTime Software** (InTime) 머신러닝에 의한 FPGA 설계를 최적화 할 수 있는 독립적인 소프트웨어 is a standalone tool that optimizes FPGA designs with machine learning.

**InTime Service** FPGA 설계를 최적화 할 수 있는 turnkey 방식의 서비스. 전혀 IT 에 문제를 걱정 하지 않아도 되는 분들을 위한 최적화 서비스

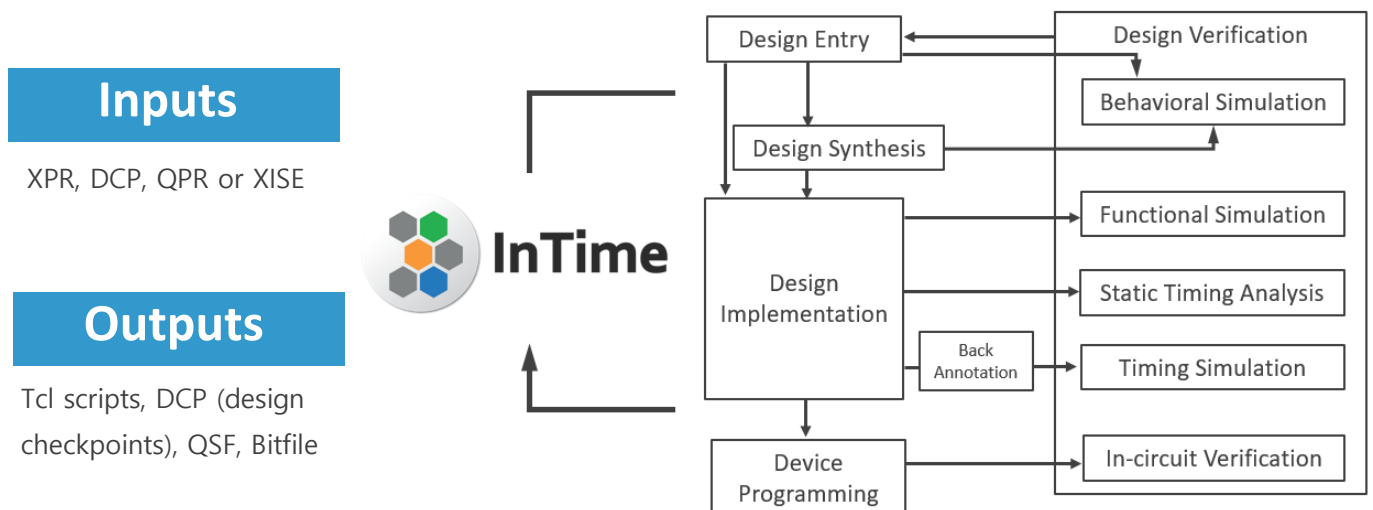
**InTime Cloud** InTime 의 설계를 최적화 할 수 있도록 속도를 가속화 할 수 있는 환경을 구축 하여 추가 하드웨어 투자 없이 cloud sever 을 이용하여 설계가 가능 함.

## C. Supported platform and O/S System

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

## D. 특성및 기능

Vivado, Quartus 등과 같은 FPGA 도구는 FPGA 디자인 성능에 많은 영향을 끼치는 설정값 및 매개변수를 포함하고 있습니다. InTime 은 machine learning 을 이용하여 최적의 조합을 설정 하고 배치 전략을 세워 디자인 성능 목표를 달성 할 수 있도록 최상의 기회를 제공 하고 있습니다.



## InTime Tool 이 꼭 필요한 경우

### A FPGA 설계 최적화 및 생산성 향상

InTime은 FPGA 설계시 사용하는 Vivado, Quartus 등 설계에 사용되는 소프트웨어의 최적의 조합을 가진 설정값 과 배치 전략에 대한 답을 찾을 수 있도록 할 수 있습니다.

설계 최적화는 WNS(worst negative slack ) 또는 TNS( Total negative slack) 및 FMax을 향상을 통해 디자인의 성능을 20% 이상 향상 시킬 수 있습니다.

InTime의 machine learnig 은 데이터 분석에 기반하여 접근을 하고 있습니다. 사용자들은 InTime 을 기초로 내부적으로 machine learning 을 설정 하여 사용 할 수 있습니다. 이로 인해 inTime 사용자들은 좋은 결과를 제공 받을 수 있게 될 것 입니다.

### B 소요시간의 단축 및 투자 수익률 향상

InTime 은 병렬 처리를 이용 반복 실행에 의해 나쁜 결과는 즉시 버리고 초기 단계에서 최상의 설정값에 초점을 맞추어 동작을 합니다. 각 디자인에 대한 계산 리소스의 구축 시간 및 비용을 대폭 절감합니다

### C RTL code 는 변경 하지 않음

InTime은 합성및 place-and-route 을 위한 최적화된 전략, 배치기술을 향상 하기 위한 최적화 된 전략을 결정 합니다. 그러므로 RTL 코드를 수정하는 일은 없습니다.

### D 완전 자동화 및 확장성이 용이

사용자는 InTime의 학습 및 최적화 프로세스를 시작하기 위해 간단한 클릭 만하면됩니다. 또한 InTime 을 사용하면 Tcl 스크립트를 사용하여 특정 실행 흐름을 사용자 정의 할 수 있습니다.

또한 InTime은 LSF 또는 SGE와 같은 자원 관리 소프트웨어와 잘 작동합니다. 사용자는 회사의 데이터 센터 나 AWS와 같은 클라우드에서 회사 내에서 실행할 수 있습니다.

### E 진보적이고 직관적 인 결과 분석 제공

InTime의 결과 분석을 사용하면 여러 빌드를 한 번에 비교하여 빌드 전달과 실패 원인을 찾아 낼 수 있습니다.

InTime은 전체 경로(critical path) 수, 상호 연결 수(interconnects) , 총 네거티브 슬랙 (TNS), 최악의 슬랙 (WS), 면적 및 전력과 같은 다양한 디자인 목표 간의 관계와 같은 데이터를 시각화하고 추적합니다.

# InTime software 만의 특징적인 기능

- 1 전략은 잘 알려진 전략을 포함하여 합성 및 PNR의 옵션의 조합을 할 수 있습니다. 또한 배치 제약 조건을 포함 할 수 있습니다.

## Examples

### Quartus II

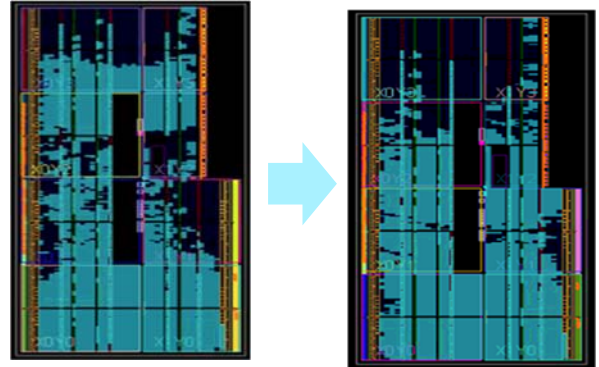
SYNTHESIS	Value
ADV_NETLIST_OPT_ALLOWED	DEFAULT
ADV_NETLIST_OPT_SYNTH_WYSIWYG_REMAP	ON
ALLOW_ANY_RAM_SIZE_FOR_RECOGNITION	ON
ALLOW_ANY_ROM_SIZE_FOR_RECOGNITION	ON
ALLOW_ANY_SHIFT_REGISTER_SIZE_FOR_RECOGNITION	ON
ALLOW_POWER_UP_DONT_CARE	OFF
ALLOW_SHIFT_REGISTER_MERGING_ACROSS_HIERARCHIES	OFF
ALLOW_SYNCH_CTRL_USAGE	ON
AUTO_CARRY_CHAINS	ON

### Vivado

Performance_Explore	Uses multiple algorithms for optimization, placement, and routing to get potentially better results.
Performance_ExplorePostRoutePhysOpt	Similar to Performance_Explore, but enables the physical optimization step (phys_opt_design) with the explore directive after routing.
Performance_RefinePlacement	Increase placer effort in the post-placement optimization phase, and disable timing relaxation in the router.
Performance_MLBlockPlacement	Ignore timing constraints for placing block RAM and DSPs, use wirelength instead.
Performance_MLBlockPlacementFanoutOpt	Ignore timing constraints for placing block RAM and DSPs, use wirelength instead, and perform aggressive replication of high fanout drivers.

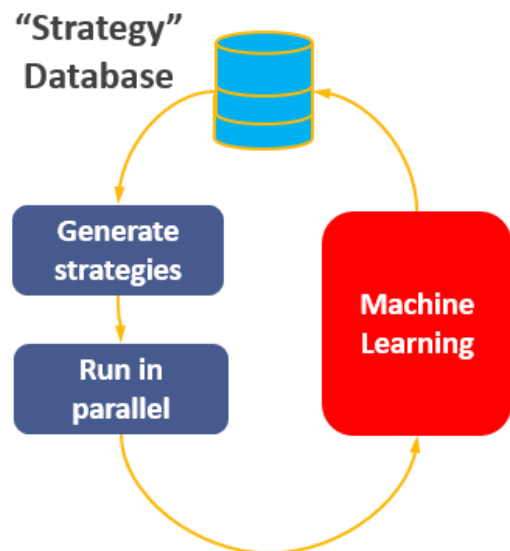
### ISE

SYNTHESIZE - XST	
AUTOMATIC BRAM PACKING	FALSE
BRAM UTILIZATION RATIO	15
CROSS CLOCK ANALYSIS	FALSE
DECODER EXTRACTION	TRUE
EQUIVALENT REGISTER REMOVAL	FALSE
FSM ENCODING ALGORITHM	COMPACT
FSM STYLE	BRAM



## How InTime works with strategies

1. Generate strategies based on database.  
데이터베이스를 기반으로 전략 생성
2. Implement all strategies.  
모든 전략을 시행
3. Use machine learning to analyze results.  
결과 분석을 위한 machine learning 사용
4. Update database with new knowledge.  
새로운 결과를 가지고 데이터베이스를 갱신



## 2 Recipes

방법은 최적화 수행 방법을 결정하는 자동화 스크립트입니다. '학습', '마지막 순간', '일반' 및 '고급'의 네 가지 범주로 그룹화 할 수 있습니다.

학습' 카테고리의 호스는 사용자가 InTime 을 처음 사용하는 디자인을 실행할 때 Plunify 가 권장하는 방법입니다. 이러한 방법은 설계 설정 및 제약 조건에 대한 초기 분석 및 교정을 수행하기 위한 것입니다. 예를 들어, InTime Default recipe 는 과거 결과에 대한 학습을 기반으로 사용 가능한 모든 컴파일러 설정의 값을 조정합니다

한편 'Last-Mile' 카테고리의 방안은 학습 방안으로 결과가 충분히 개선되었을 때보다 구체적인 최적화를 수행합니다. 예를 들어, 배치 시드 탐색 방안은 컴파일러 설정의 배치 시드만 변경합니다.

마지막으로 중요한 것은 '일반' 및 '고급'의 방안이 유용성을 위한 것입니다.

	Recipes	Quartus	Quartus-Pro	Vivado	ISE	Descriptions
Learning	InTime Default	Y	Y	Y	Y	Performs first time calibration, exploration and optimization of your design
	InTime Default Extra	Y	Y	Y		Performs actions in Intime Default recipe, and run additional optimizations
	Hot Start	Y <sup>(1)</sup>		Y <sup>(2)</sup>		Generates initial strategies for your design by correlating it with other designs in the InTime database
	Deep Dive	Y <sup>(1)</sup>		Y	Y	Performs deeper analysis of result of your design
	Explorer			Y		Explores different optimization in your design
Last-Mile	Auto Placement	Y	Y			Performs self-guided placement adjustments to improve the worst timing failure paths in your design.
	Effort Level Exploration	Y	Y			Explores different effort level
	Placement Seed Exploration	Y	Y		Y	Explores the effect of placement seeds on the design goal
	Seeded Effort Level Exploration	Y	Y			Performs exploration of seeds on best effort level results
	Router Effort Level Exploration	Y	Y			Explores different routing effort level
	Extra Opt Exploration			Y		Explores optimization that focus on placement, physical optimization and routing
	Placement Exploration			Y		Explores the effect of placement adjustment on the design
	Map Seed Exploration				Y	Explores the effect of map seeds on the design goal
General	Just Compile My Design	Y	Y	Y	Y	Compiles the active revision in your project
	Rerun Strategies	Y	Y	Y	Y	Rerun all marked strategies
Advanced	Custom Flow	Y	Y	Y	Y	Uses strategies specified by the user to compile

<sup>(1)</sup> Not Supported for Arria10 in QuartusII    <sup>(2)</sup> Coming Soon

### 3 Parallel compilation (병렬 컴파일)

InTime 은 다양한 환경에서 사용 가능한 리소스를 사용할 수 있습니다. "대상 실행"설정을 사용하면 빌드를 회사의 네트워크에있는 여러 컴퓨터에 분산 된 컴퓨터에서 로컬로 실행할 수 있습니다. 사용자는 LSF (Load Sharing Facility) 또는 Oracle / Sun Grid Engine (SGE)과 같은 작업 부하 관리 소프트웨어와 함께 작동하도록 구성 할 수 있습니다.

### 4 Flexible build environments (유연한 빌드 환경)

다중 기계 구성에서 지정된 InTime "제어"서버는 모든 "작업"(하나 이상의 빌드로 구성됨)에 대해 기계를 할당합니다. InTime 은 "작업자"컴퓨터와 통신하여 파일 / 결과를 암호화 및 전송하므로 사용 가능한 컴퓨팅 리소스를 활용하여 빌드를 병렬로 실행할 수 있습니다.

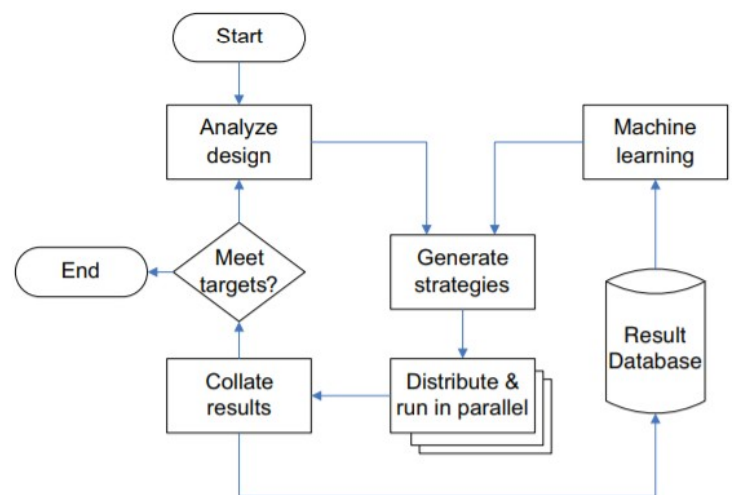
## InTime TOOL 구성 요소

### InTime's Machine Learning Approach

.InTime 은 최적화 과정을 안내하고 유도하기 위해 기계 학습, 검색 공간 제거 및 제한된 무작위 탐색의 조합을 사용합니다.

오른쪽 그림과 같이 InTime 흐름은 이전 디자인의 특성 및 기존 결과를 분석하여 시작합니다. InTime 은 다음 기능과 관련된 수학적 모델을 유도합니다.

- design characteristics like target device family and logic structure
- compiler settings like synthesis and place-and-route options
- design performance metrics like timing, area and power estimates.



이 모델은 설계 특성에 따라 FPGA 도구 설정의 통계적 관련성을 결정하는 데 사용됩니다. 그런 속성은 70 가지 이상의 컴파일러 설정 중 타이밍에 가장 큰 영향을주는 질문에 대답하는 데 적용됩니다. 어떤 컴파일러 설정 값이 특정 장치 제품군에 가장 큰 영향을 미칩니 까?

InTime 은 장치 및 틀체인 관련 초기 기계 학습 예상치를 사용하여 전략 생성 엔진을 안내합니다. 각 전략이 기본 FPGA 틀 체인에 적용된 후에는 전략이 선택된 실행 타겟에서 분산되고 평가됩니다.

InTime 프레임 워크는 사용자로부터 IT 인프라를 추상화하고 InTime 의 실행 데이터베이스에 통합되기 전에 생성 된 결과를 자동으로 분석하고 대조합니다. 반복 접근법은 흐름을 구성 가능한 횟수만큼 반복 할 수있게합니다. 각각의 새로운 라운드는 이전에 저장된 결과를 사용하여 학습 생성 데이터를 생성하여 전략 생성 엔진이 교육 데이터의 양이 증가함에 따라보다 지능적인 전략을 유도하도록 안내합니다.

InTime 은 통계를 사용하여 설계, 공구 설정 및 합성 / 구현 결과 간의 상관 관계를 분석합니다. Timing Score 의 중앙값이 빌드에서 빌드까지 향상되는 한, Timing Closure 가 달성 될 때까지 개선 할 수있는 좋은 기회가 있습니다. 이러한 상관 관계는 저장되어 잘 작동하는 것이 무엇인지에 대한 정보가 축적되어 후속 설계에보다 신속하게 적용됩니다.

**Timing improvements (TNS) with optimal group of settings**



**Other Information of InTime .**

**1 Licensing**

Annual license priced according to the maximum number of concurrent builds.  
 최대 동시 빌드 수에 따라 가격이 책정 된 연간 라이선스

**2 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



Company name : Plunify Pte Ltd

Address : 82, Lorong 23 Geylang, Atrix Building, #05-14, 388409

Home page : <https://www.plunify.com/>