

Information Sciences Institute

TRUST in Integrated Circuits Program

Industry Day Test Article Generation

Robert Parker
Deputy Director, USC/ISI

- **TAG Charter**
- **ASIC Development**
- **FPGA Development**
- **Distribution and Support**

- **TAG Charter**
 - ASIC Development
 - FPGA Development
 - Distribution and Support

Definition of Terms

Clean Design Process: For the purposes of this program, the “clean design process” defines the reference design flow that will be used to create the baseline chips for the program. This flow may incorporate libraries, IP, CAD tools, PDKs, etc. that will be used to accurately and faithfully capture the functionality of the devices and to produce clean devices for the program.

Clean Devices: Devices produced by the Clean Design Process are defined to faithfully represent the intended functionality, are to be trusted and are guaranteed to be free of malicious circuitry.

Compromised Design Process: Stages of the Clean Design Flow will be intentionally modified such that they produce compromised devices.

Compromised Devices: These devices are designed and fabricated from a “clean design” specification but will be purposefully altered at one or more stages of design or fabrication resulting in devices with unintended functionality.

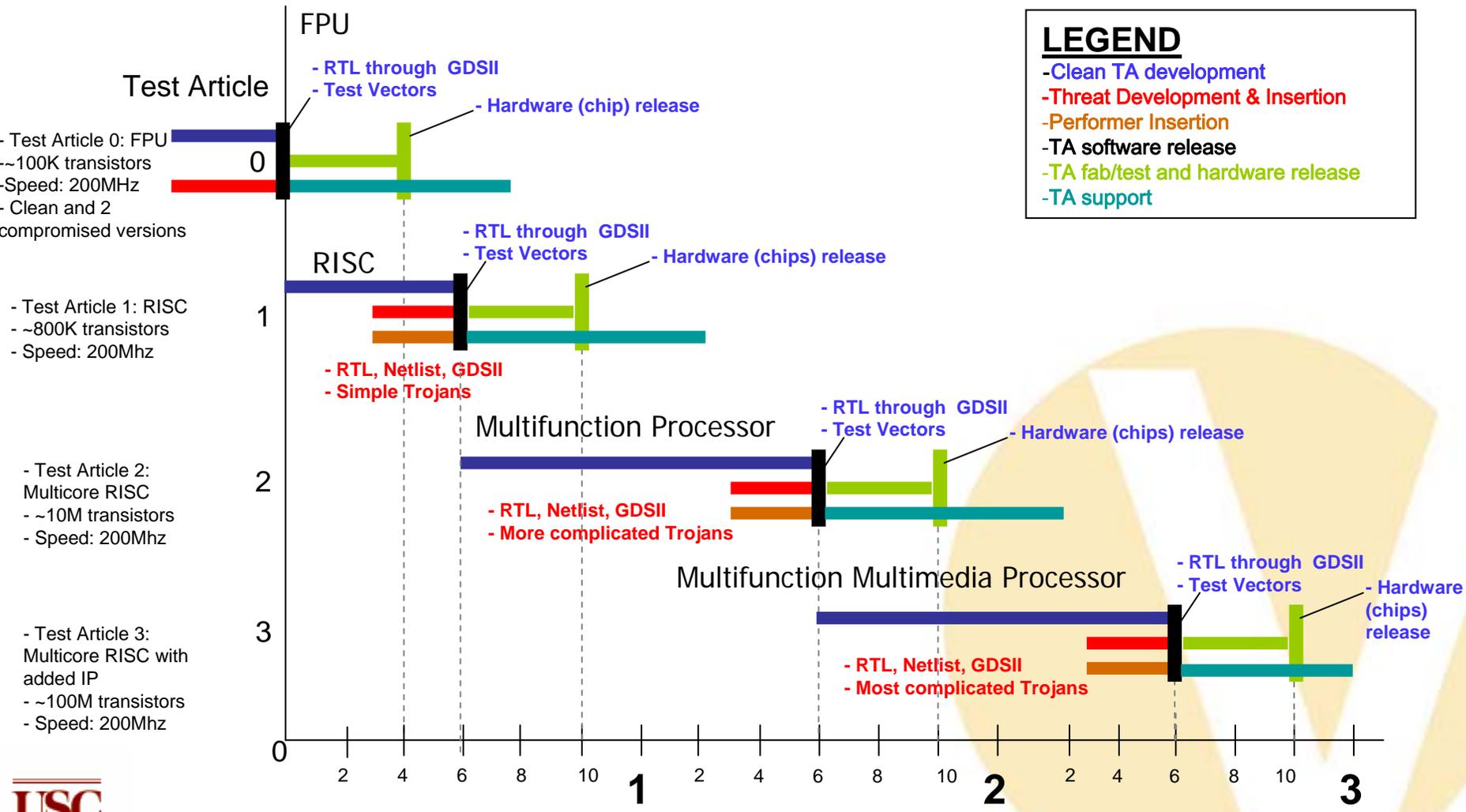
Trusted Electronics Program Design and Coordination Team Tasks

- **Create, Coordinate, and maintain web-accessible design documents, technology files, design files, etc.**
- **In conjunction with the Red Team, define the effects and triggers list**
- **Insert, collect, and integrate malicious circuit descriptions into a design flow**
- **Design test vehicles and insert circuits**
- **Schedule, layout, aggregate, fabricate, and package VLSI test articles**
- **Fabrication process monitoring**
- **Verify functionality of test articles**
- **Distribute articles**
- **Provide contractor community support**

- TAG Charter
- **ASIC Development**
- FPGA Development
- Distribution and Support

- **Program will design ASICS using a Clean Design Process**
 - Clean Devices, RTL, netlists, GDSII, and test vectors will be supplied to the contractor community
- **For each clean ASIC design, one or more Compromised Devices will be designed, fabricated and distributed to the community**
 - RTL, netlists, and GDSII of compromised devices may or may not be made available
- **Contractors must detect and locate the causes of altered functionality**

ASIC Schedule



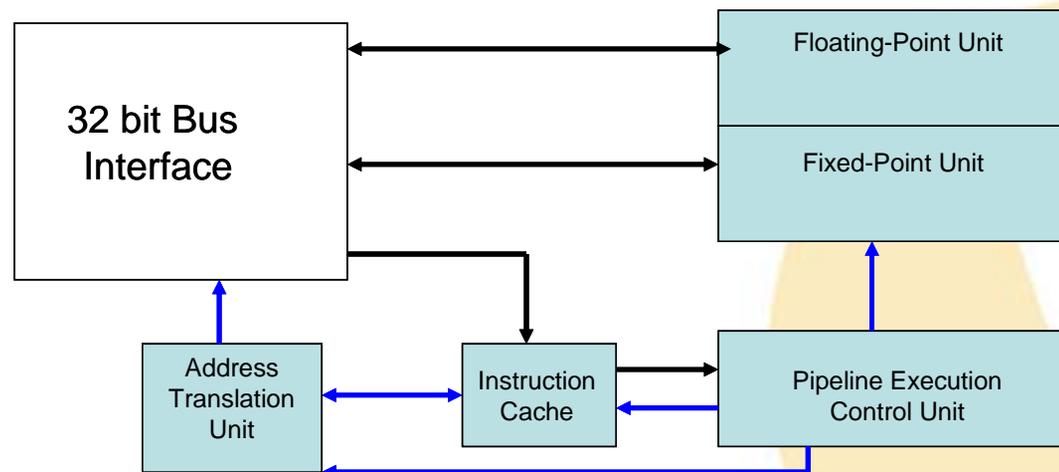
• Test Article 0 – Floating-Point Unit

- 32-bit single-precision pipelined floating-point unit (FPU) using IEEE-754 format
 - *Operations supported: absolute value, negate, add, subtract, multiply, divide, integer-to-floating-point conversion, floating-point-to-integer conversion*
 - *Exceptions: inexact, invalid, divide-by-zero, underflow, overflow*
 - *No support for denormalized numbers*
 - *When result is denormalized number, generate minimum normalized number and raise underflow flag*

- **Test Article 1 – RISC processor**

- RISC processor with single-precision FPU, instruction cache, and address translation unit

- *32-bit addresses and data*
- *Single-issue, inorder, 5-stage execution pipeline*



- **Test Articles 2 and 3 (Years 2 and 3)**
 - Successively larger test articles based on the RISC processor design of Test Article 1 will be developed in years 2 and 3
 - Several enhancement options for increasing size will be explored
 - *Multi-core designs*
 - *Multi-media extension units*
 - *Increased on-chip memory*
 - *Addition of IP blocks*

- **Test vectors and expected results will be supplied with each test article**
 - For Test Article 0 (FPU), test vectors will exercise all FPU operations over a wide range of data, including special corner cases, e.g., zero, infinity, NaN, etc
 - For Test Article 1 (RISC processor), test vectors will exercise all operations over a wide range of data and other basic functionality (such as data forwarding in the case of data hazards)
 - In all cases, test vectors will NOT provide exhaustive data coverage for individual operations

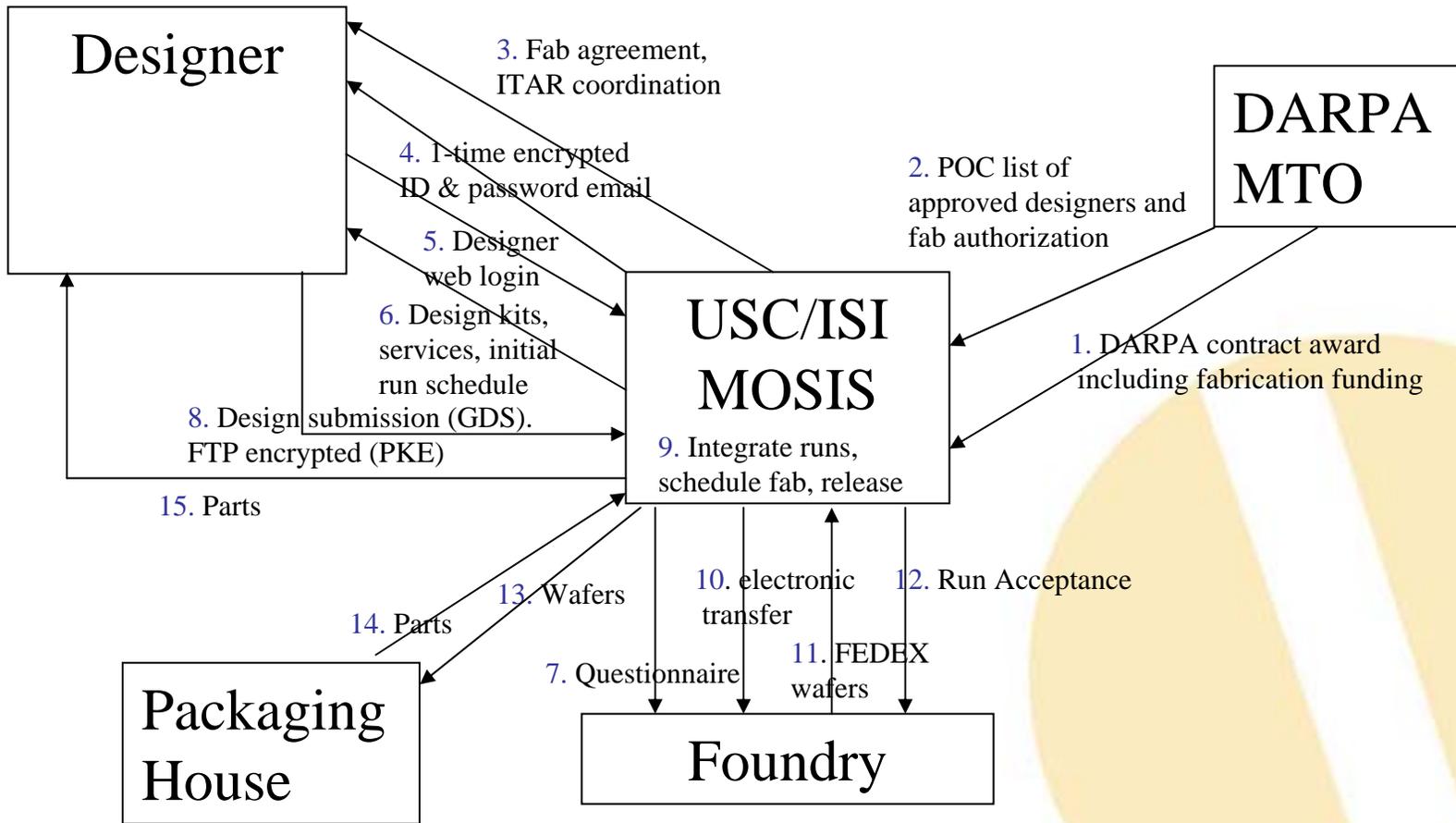
- All files distributed for ASIC test articles will be consistent with the tool flow shown below

Design Process Step	Tool
Functional Verification	Cadence Incisive HDL Simulator (includes NC-SIM)
Synthesis	Synopsys Design Compiler
Timing Analysis	Synopsys Primetime
Place & Route	Cadence Encounter
Backend Checks (DRC, LVS, etc)	Mentor Calibre

Detection Techniques Requiring HW Insertion

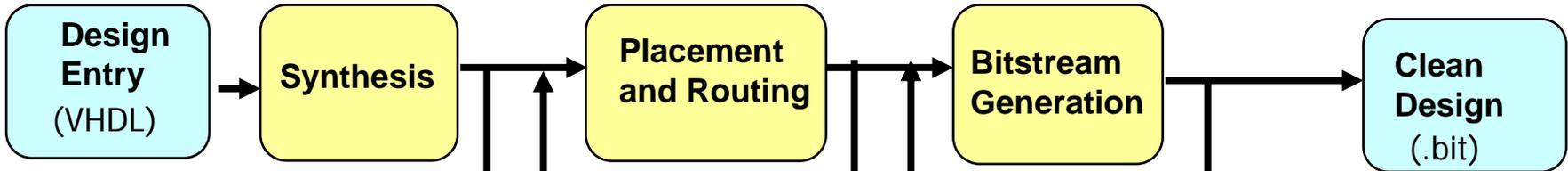
- **Bidders may propose techniques that require insertion of circuitry into Test Articles**
- **Opportunities for inserting such circuitry will be supported during development of Test Articles 1, 2, and 3**
 - Bidders must coordinate with Test Article team early in development stage of each test article
 - *Bidders wishing to insert circuitry into test articles must provide that circuitry to the Test Article team one month prior to RTL design freeze for TA1,2,3. No opportunity will be provided for bidder circuitry for TA0*

Sponsored Foundry Access Management

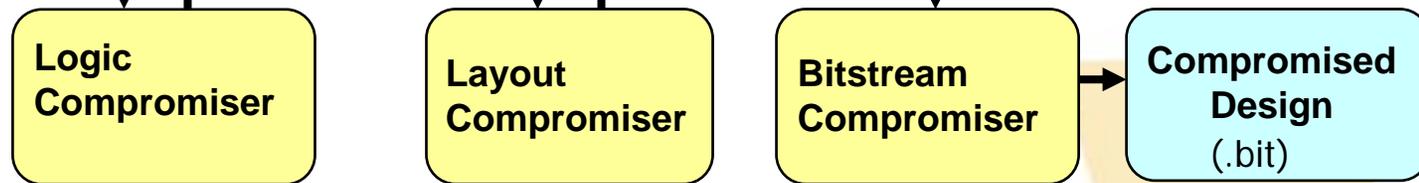


- TAG Charter
- ASIC Development
- **FPGA Development**
- Distribution and Support

Clean Design Process



Compromised Design Process



- Program will use commercial FPGAs (HW is assumed trusted)
- Malicious circuitry can be inserted at any level
- Corruption tools extensible to multiple FPGA families, vendors

FPGA Schedule

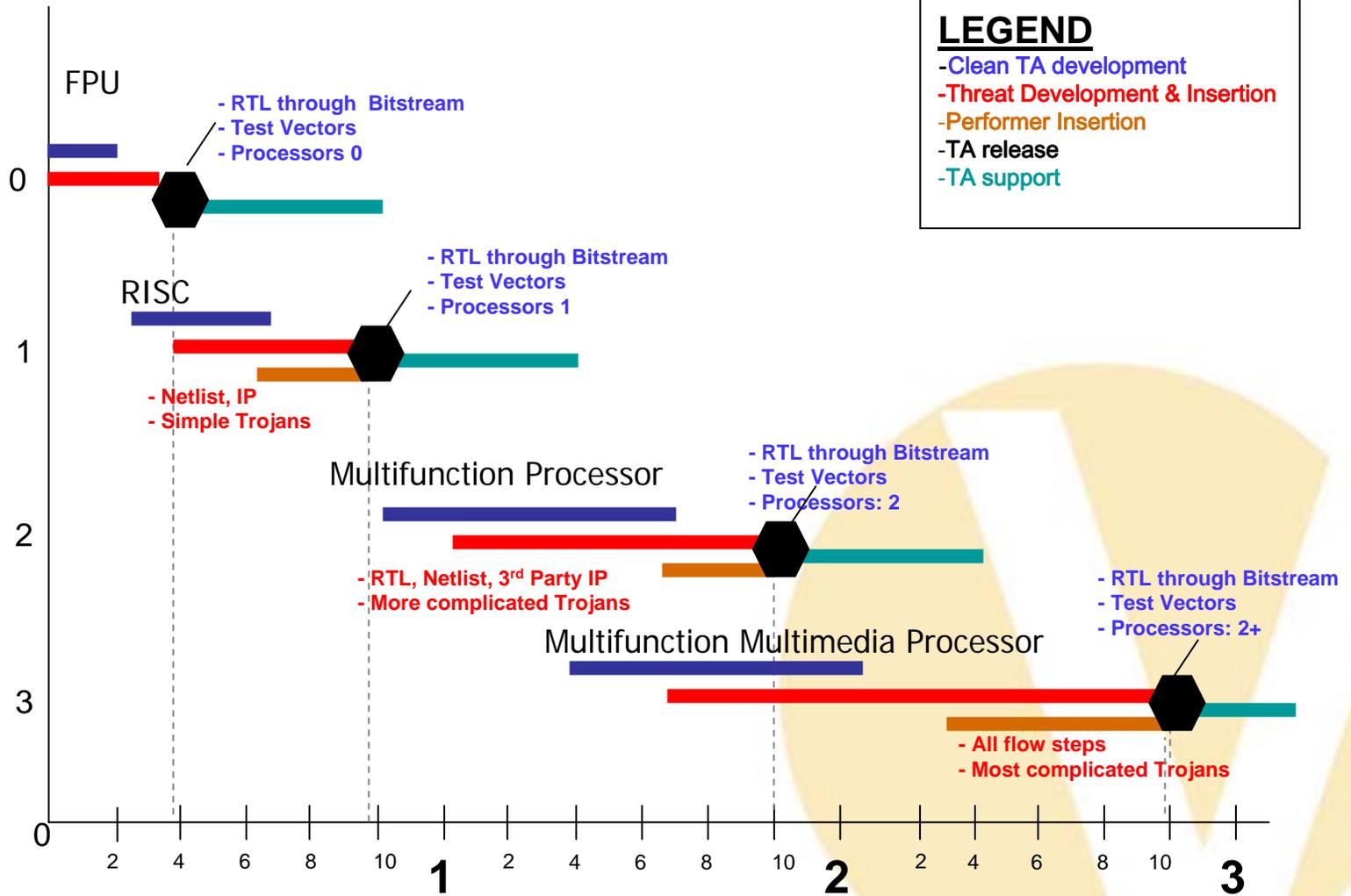
Test Article

- Part: Virtex-4
- Utilization: N/A
- Speed: N/A
- Minimal Functionality

- Part: Virtex-4
- Utilization: 60-70%
- Speed: 150Mhz
- New logic:30%
- Add IP

- Part: Virtex-5
- Utilization: 60-70%
- Speed: 200Mhz
- New logic: 30%
- Add IP

- Part: Virtex-5
- Utilization: 70%
- Speed: 200Mhz
- New logic: 30%
- Add IP



- **Goal: RTL through bitstream, test vectors, and primitive Trojans delivered to contractors early in program**
- **Approach: use existing ISI FPU design from ASIC test article and resynthesize to FPGA technology**
- **Plan, similar to ASIC flow, is to develop clean and compromised versions – separate bitstreams**
- **FPGA TA0 will use same Trojans as ASIC TA0**

FPGA Reference Tool Sets

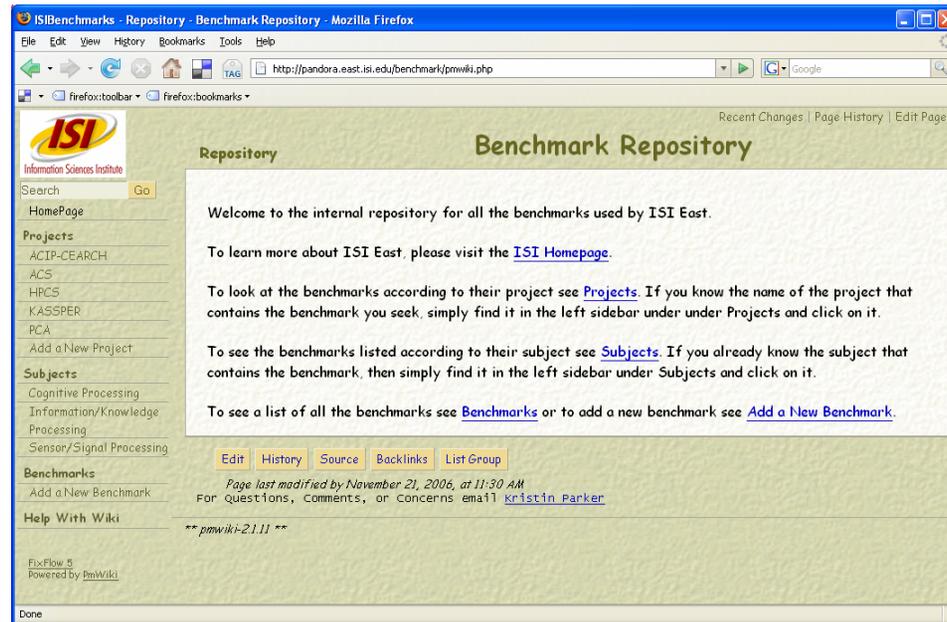
Functional Verification	Mentor Modelsim®
Synthesis	Synopsis SynplifyPro®
Placement, Routing, and Bitstream Generation	Xilinx ISE*

Proposers can use other toolsets

— Reference Tool Set is deferred to in the case of a discrepancy

***Phase II and III will add Altera and Quartus-II tool suite**

- TAG Charter
- ASIC Development
- FPGA Development
- **Distribution and Support**



Website

- General documentation describing the TA, test vector description, and exact software version of tools used
- General FAQs
- Directions on how to access code repository

Support

- Questions related to the TAs posted on program-accessible website
- Questions related to a particular contractor's technique: ta_support@isi.edu

SSH-based access to Version Control repository of:

- **RTL**
- **Cell libraries**
- **Synthesis scripts**
- **Post synthesis netlist and log files**
- **Static timing analysis results**
- **Place and Route scripts**
- **GDSII**
- **Test Vectors and expected results**



SSH-based access to Version Control repository of

- Input files (VHDL / Verilog source, 3rd Party IP)
- Project files, Makefiles, and Constraint files
- Test Vectors
- Netlists (EDIF Synthesis output)
- Bitstream

Proposer defined circuitry

- TA0: No contractor defined circuitry, malicious circuits hand inserted
- TA1,2,3:
 - *TAG will insert any contractor defined circuitry*
 - *Insert malicious circuits via scripts*