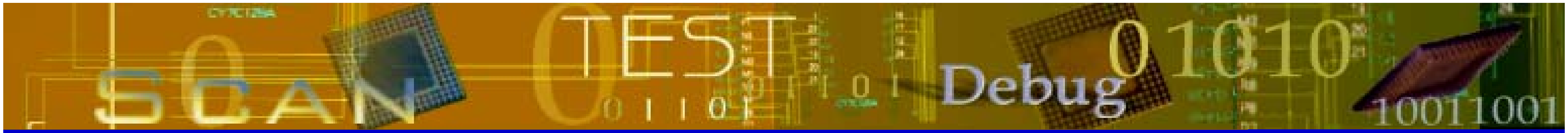
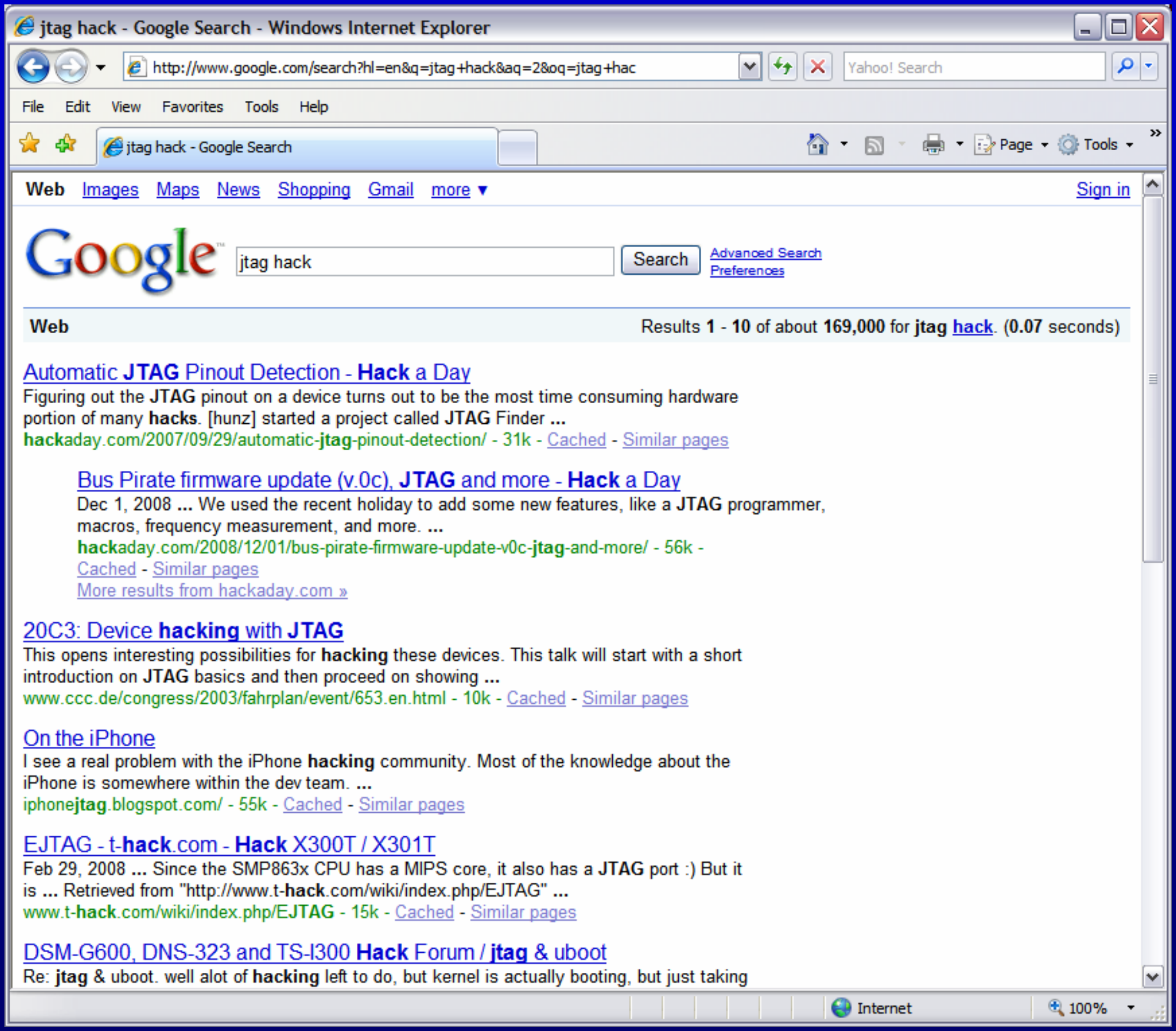


Holistic FPGA Configuration

CJ Clark, CEO
Intellitech Corp
cclarkATintellitechdotcom

JTAG Hack – 169,000 results



Web Results **1** - **10** of about **169,000** for **jtag hack**. (**0.07** seconds)

Automatic **JTAG** Pinout Detection - **Hack** a Day
Figuring out the **JTAG** pinout on a device turns out to be the most time consuming hardware portion of many **hacks**. [hunz] started a project called **JTAG** Finder **...**
hackaday.com/2007/09/29/automatic-**jtag**-pinout-detection/ - 31k - Cached - Similar pages

Bus Pirate firmware update (v.0c), **JTAG** and more - **Hack** a Day
Dec 1, 2008 **...** We used the recent holiday to add some new features, like a **JTAG** programmer, macros, frequency measurement, and more. **...**
hackaday.com/2008/12/01/bus-pirate-firmware-update-v0c-**jtag**-and-more/ - 56k - Cached - Similar pages
More results from hackaday.com »

20C3: Device **hacking** with **JTAG**
This opens interesting possibilities for **hacking** these devices. This talk will start with a short introduction on **JTAG** basics and then proceed on showing **...**
www.ccc.de/congress/2003/fahrplan/event/653.en.html - 10k - Cached - Similar pages

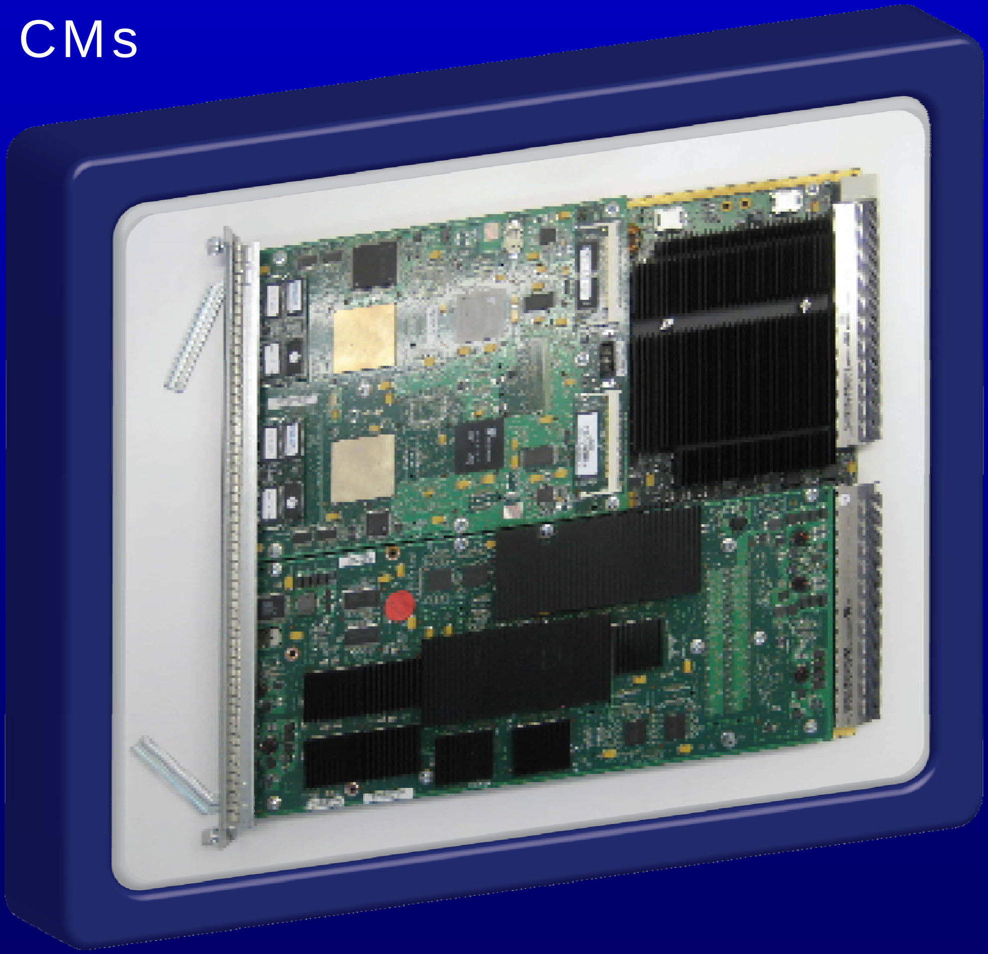
On the iPhone
I see a real problem with the iPhone **hacking** community. Most of the knowledge about the iPhone is somewhere within the dev team. **...**
iphone**jtag**.blogspot.com/ - 55k - Cached - Similar pages

EJTAG - t-**hack**.com - **Hack** X300T / X301T
Feb 29, 2008 **...** Since the SMP863x CPU has a MIPS core, it also has a **JTAG** port :) But it is **...** Retrieved from "http://www.t-**hack**.com/wiki/index.php/EJTAG" **...**
www.t-**hack**.com/wiki/index.php/EJTAG - 15k - Cached - Similar pages

DSM-G600, DNS-323 and TS-I300 **Hack** Forum / **jtag** & uboot
Re: **jtag** & uboot. well alot of **hacking** left to do, but kernel is actually booting, but just taking

Today's PCB

Multi-PCB assemblies
Large heat sinks, BGAs, Gigabit signals
1500 Ball devices, high I/O connectors
Manufactured by multiple CMs
at multiple locations



Intellitech

Traditional System Test

Multiple boards and chips in a system

- Highly integrated, running application software
- Custom ASICs, uP, Memory, FPGA, PHYs,

Low observability and diagnosis on fail

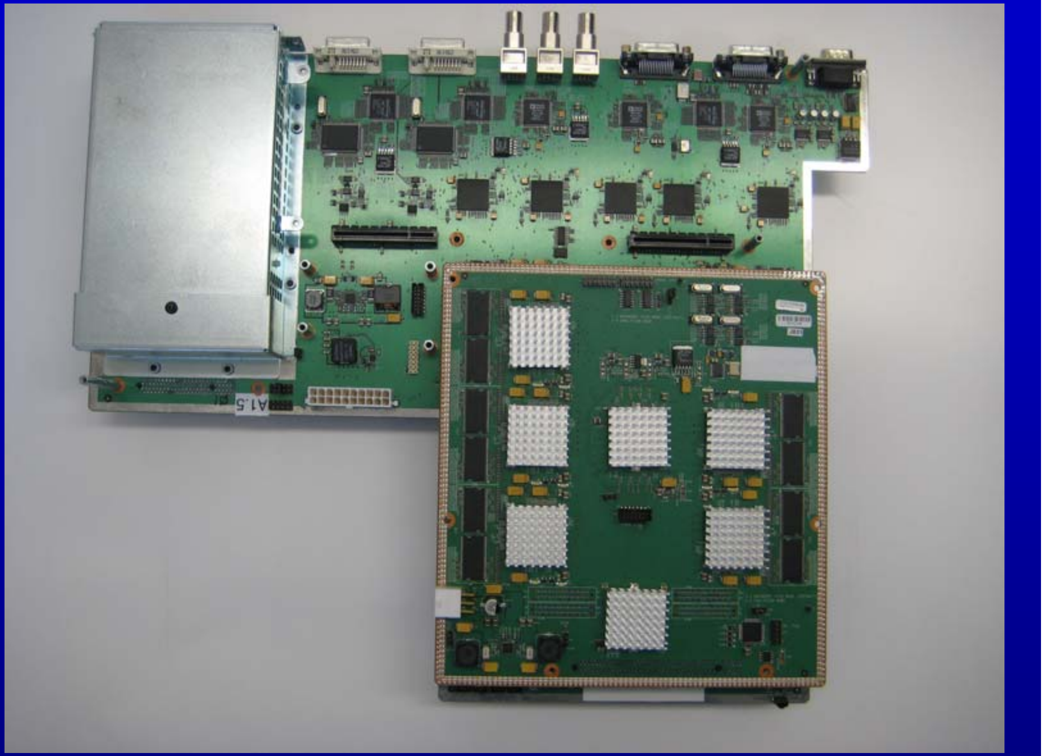
Hard to root cause failure

Did the ASIC cause the problem? PCB? Noise?

- If so what was wrong with it?

Need more than general area

- Which vendor is responsible?



Functional Test is hardest for CM to debug
- they don't specialize in your product
- They know ICT and JTAG

Intellitech

DFT Standards Continue to Grow

- IEEE 1149.1 – Test Access Port & Boundary Scan Standard

Layered on top of the 4 pin IC access of 1149.1:

- IEEE 1149.6 - Boundary Scan for AC coupled nets
- IEEE 1149.4 – Boundary Scan for Mixed Signal
- IEEE 1532 - FPGA configuration over 1149.1
- IEEE P1687 - Internal Instrument access w/ 1149.1
- IEEE ????? - A-Toggle Study Group
- IEEE ????? - SERDES BIST Study Group

Related Standards:

IEEE P1149.7 – 2 Wire low-cost 1149.1
IEEE 1500 - SoC & Core test standard
IEEE P1581 - Static Interconnect for memories

Intellitech

DFT Standards Continue to Grow

- Loss of Physical Access for Test Points due to miniaturization - BGA devices with no pins to probe

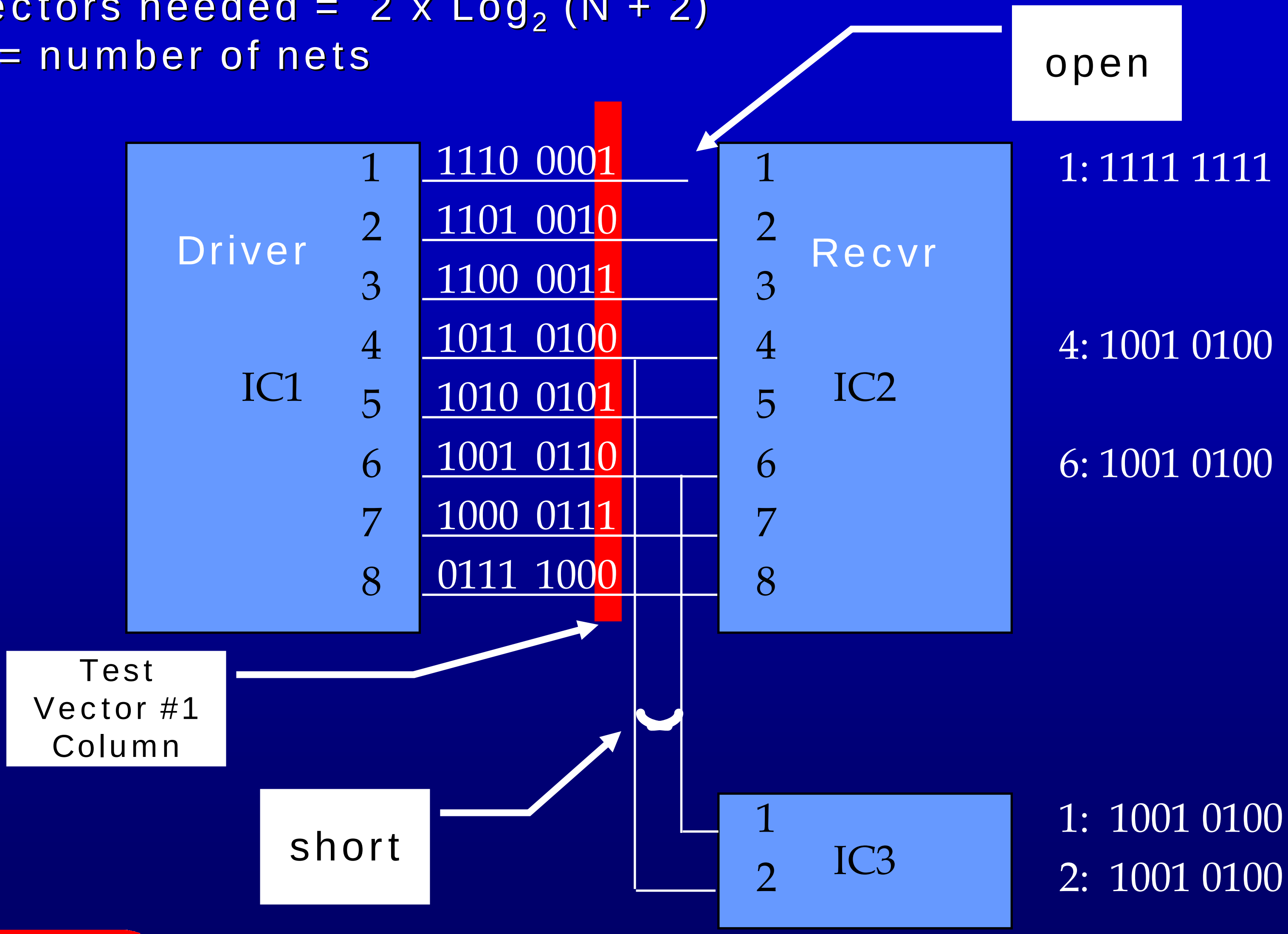
- High speed nets prohibit capacitance/routing for test points
- IC complexity – need ‘black-box’ approach to get pins to toggle on complex ICs

- Fast time to market – FPGAs/CPLDS/FLASH need to be programmed in-situ (on-board)
- Programmable devices need structured method for in-the-field updates

- Reduced high-skilled staff –
- Need to outsource test development to lower costs

Automatic Test Pattern Generation

Vectors needed = 2 x Log_2 (N + 2)
N = number of nets



open

Driver

IC1

1 1110 0001
2 1101 0010
3 1100 0011
4 1011 0100
5 1010 0101
6 1001 0110
7 1000 0111
8 0111 1000

Recvr

IC2

1
2
3
4
5
6
7
8

1: 1111 1111

4: 1001 0100

6: 1001 0100

Test Vector #1 Column

short

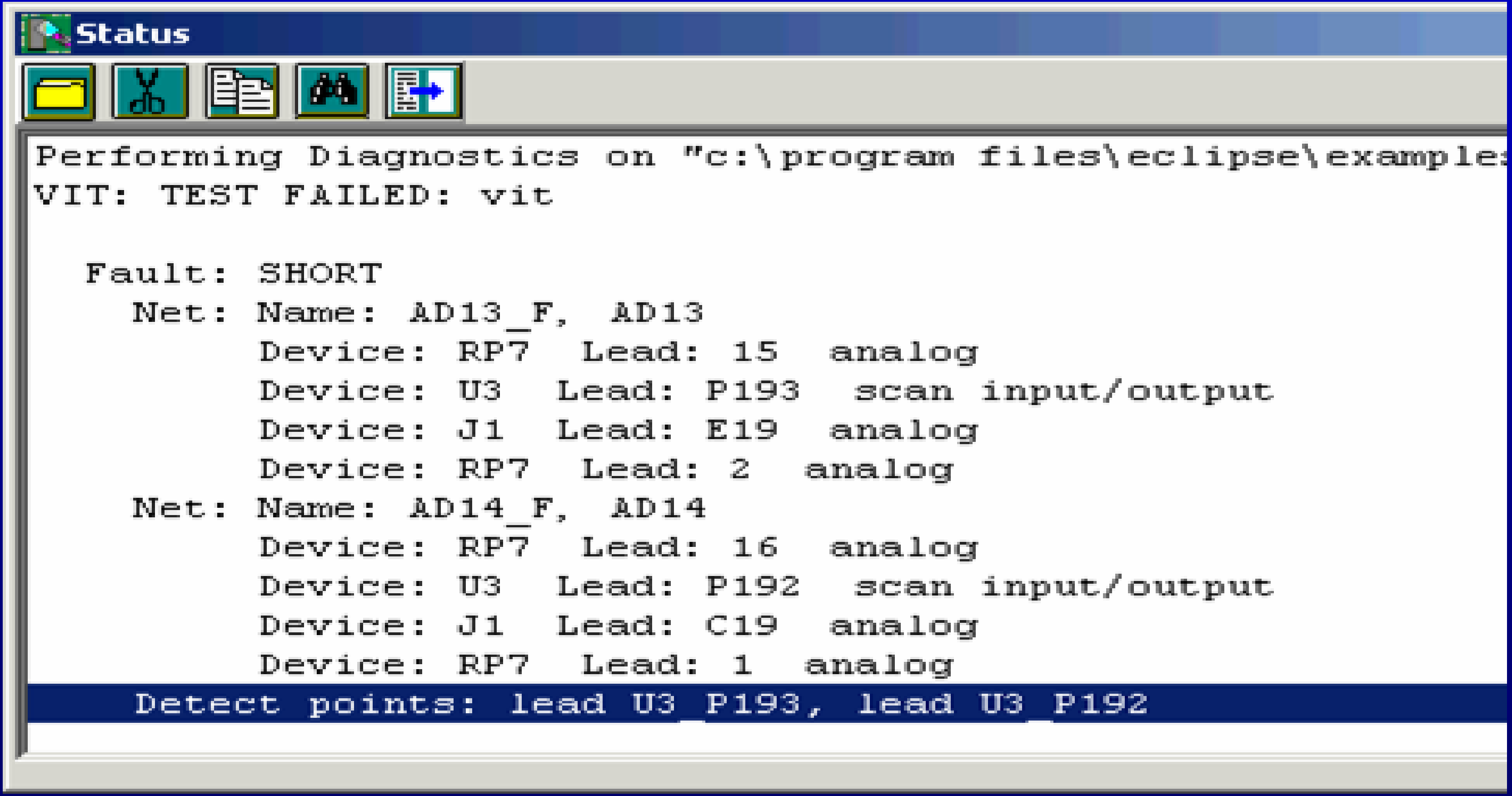
IC3

1
2

1: 1001 0100
2: 1001 0100

ATPG with Diagnostic engines

Instant Pin/Net Diagnostics – no code to write

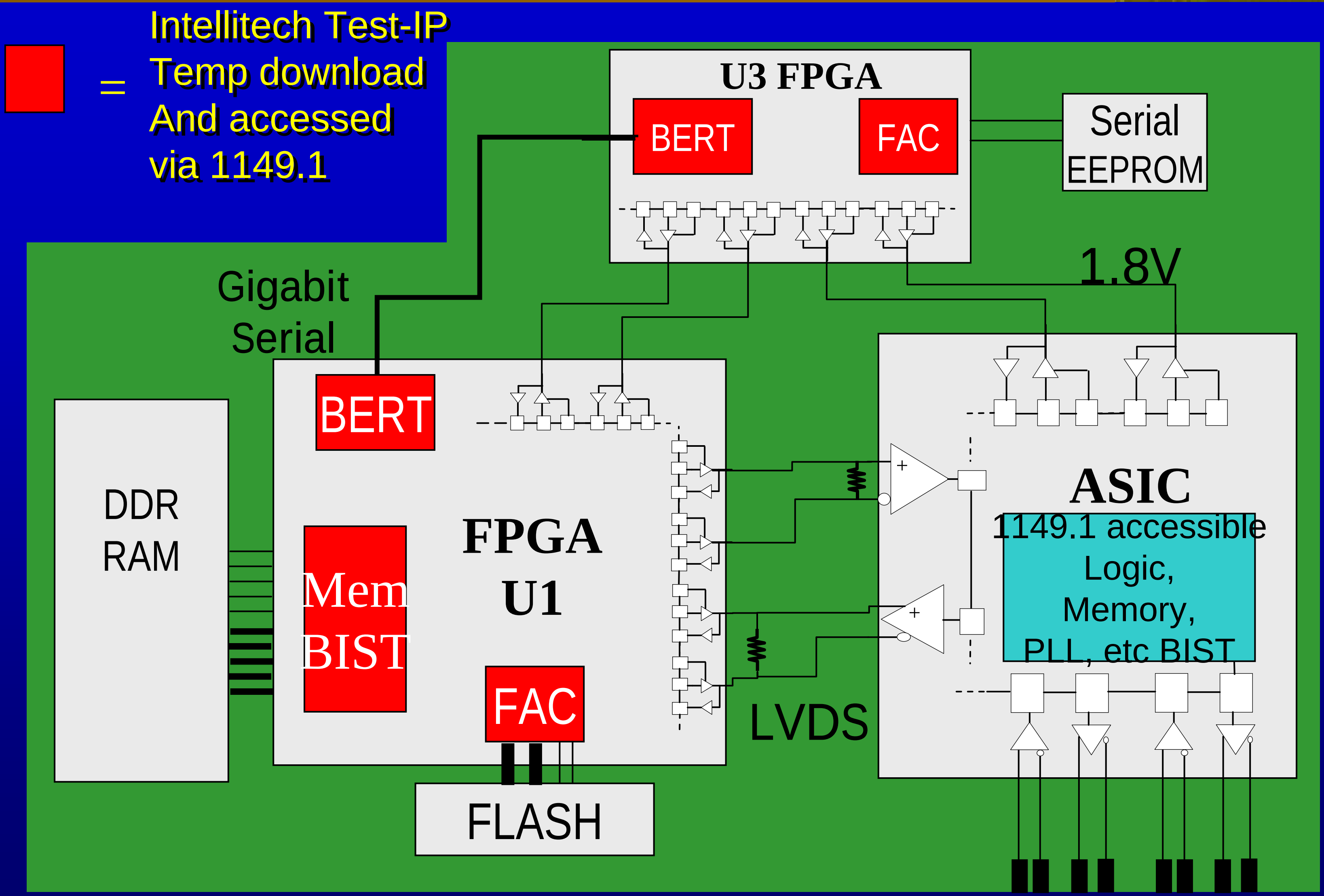


```
Performing Diagnostics on "c:\program files\eclipse\example
VIT: TEST FAILED: vit

  Fault: SHORT
    Net: Name: AD13_F, AD13
         Device: RP7  Lead: 15  analog
         Device: U3  Lead: P193  scan input/output
         Device: J1  Lead: E19  analog
         Device: RP7  Lead: 2  analog
    Net: Name: AD14_F, AD14
         Device: RP7  Lead: 16  analog
         Device: U3  Lead: P192  scan input/output
         Device: J1  Lead: C19  analog
         Device: RP7  Lead: 1  analog
    Detect points: lead U3_P193, lead U3_P192
```

Intellitech

1149.1/JTAG for At-speed Tests



4 wire test – possible to embed JTAG

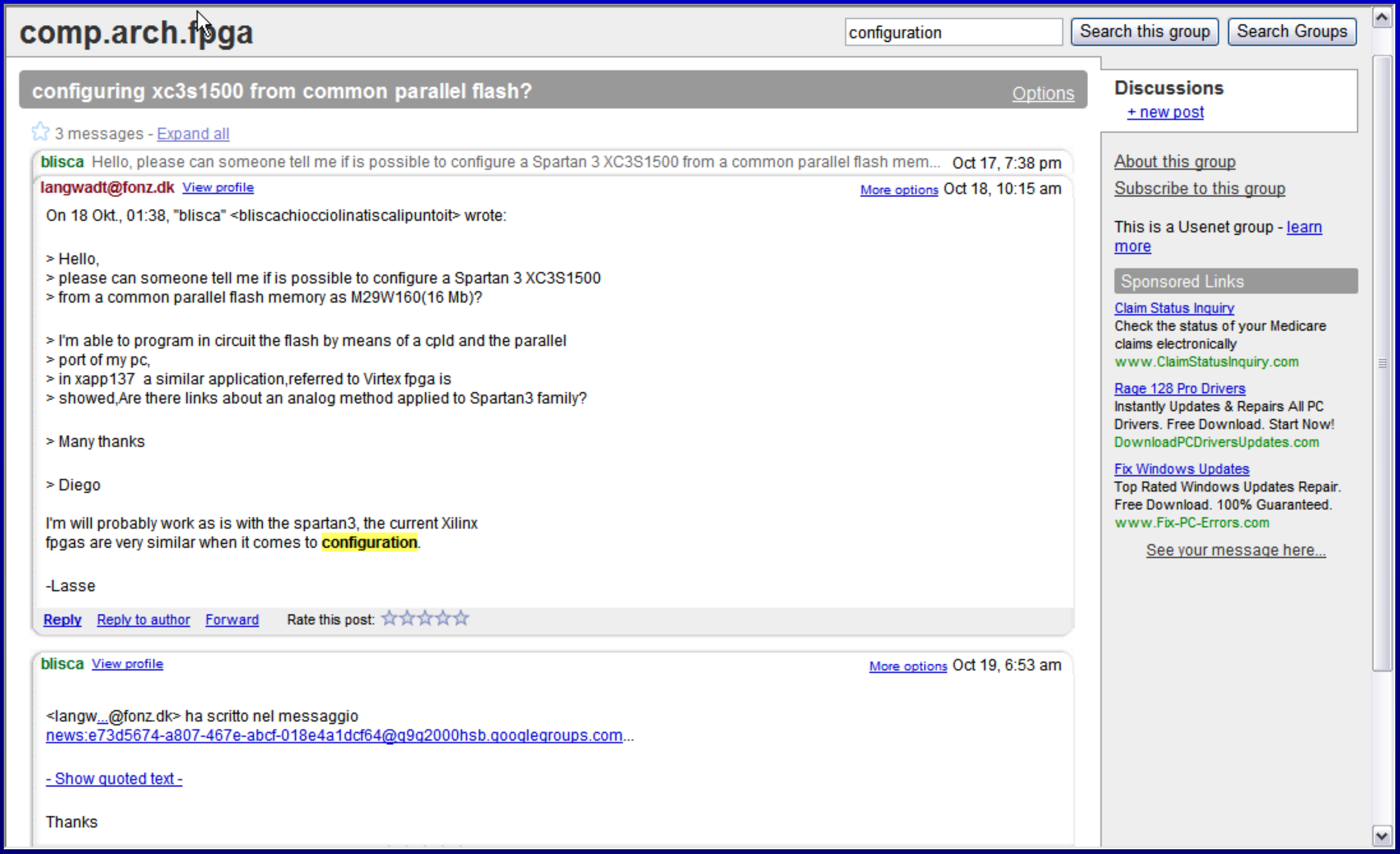
Intellitech

Many FPGA config methods

Serial	8-bit Parallel	16-bit Parallel	32-bit Parallel	JTAG
Master/Slave Serial, Master SPI	Platform Flash, CPLD and Flash, Master BPI-Up, BPI-Down, ECD, uP, SystemBIST	Master BPI-Up, BPI-Down, Master SelectMAP, Platform Flash XL	Slave Select MAP	SystemACE, SystemBIST
	Platform Flash needs JTAG controller in the system to do an update in the field			ad-hoc updates for compact flash of SystemACE
	ad-hoc methods for secure/failsafe updates other than SystemBIST			Use SPI interface for SystemBIST

FPGA & Configuration

Comp.arch.fpga – config not always smooth



comp.arch.fpga

configuring xc3s1500 from common parallel flash?

3 messages - Expand all

blisca Hello, please can someone tell me if is possible to configure a Spartan 3 XC3S1500 from a common parallel flash mem... Oct 17, 7:38 pm

langwadt@fonz.dk View profile — More options Oct 18, 10:15 am

On 18 Okt., 01:38, "blisca" <bliscachiocciolinatiscalipuntoit> wrote:

> Hello,
> please can someone tell me if is possible to configure a Spartan 3 XC3S1500
> from a common parallel flash memory as M29W160(16 Mb)?

> I'm able to program in circuit the flash by means of a cpld and the parallel
> port of my pc,
> in xapp137 a similar application,referred to Virtex fpga is
> showed,Are there links about an analog method applied to Spartan3 family?

> Many thanks

> Diego

I'm will probably work as is with the spartan3, the current Xilinx
fpgas are very similar when it comes to **configuration**.

-Lasse

blisca View profile — More options Oct 19, 6:53 am

<langw...@fonz.dk> ha scritto nel messaggio
news:e73d5674-a807-467e-abcf-018e4a1dcf64@q9g2000hsb.googlegroups.com...

- Show quoted text -

Thanks

Discussions
+ new post

About this group
Subscribe to this group

This is a Usenet group - learn more

Sponsored Links

Claim Status Inquiry
Check the status of your Medicare claims electronically
www.ClaimStatusInquiry.com

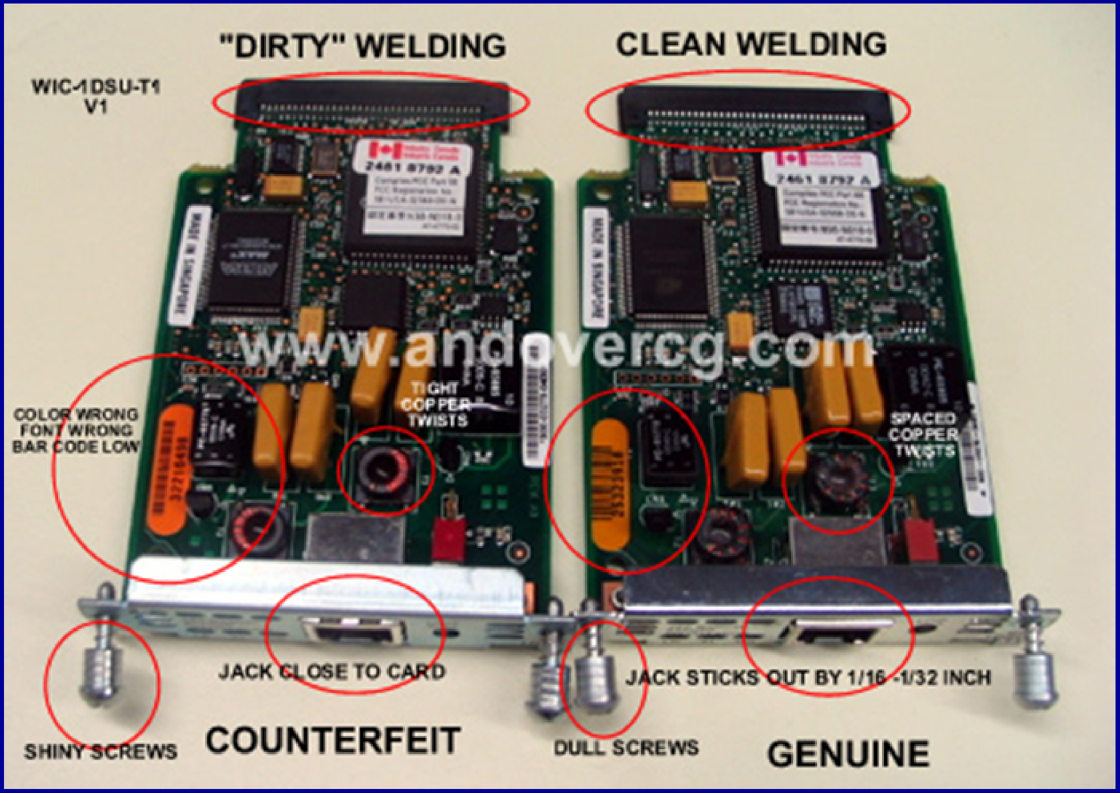
Rage 128 Pro Drivers
Instantly Updates & Repairs All PC Drivers. Free Download. Start Now!
DownloadPCDriversUpdates.com

Fix Windows Updates
Top Rated Windows Updates Repair. Free Download. 100% Guaranteed.
www.Fix-PC-Errors.com

See your message here...

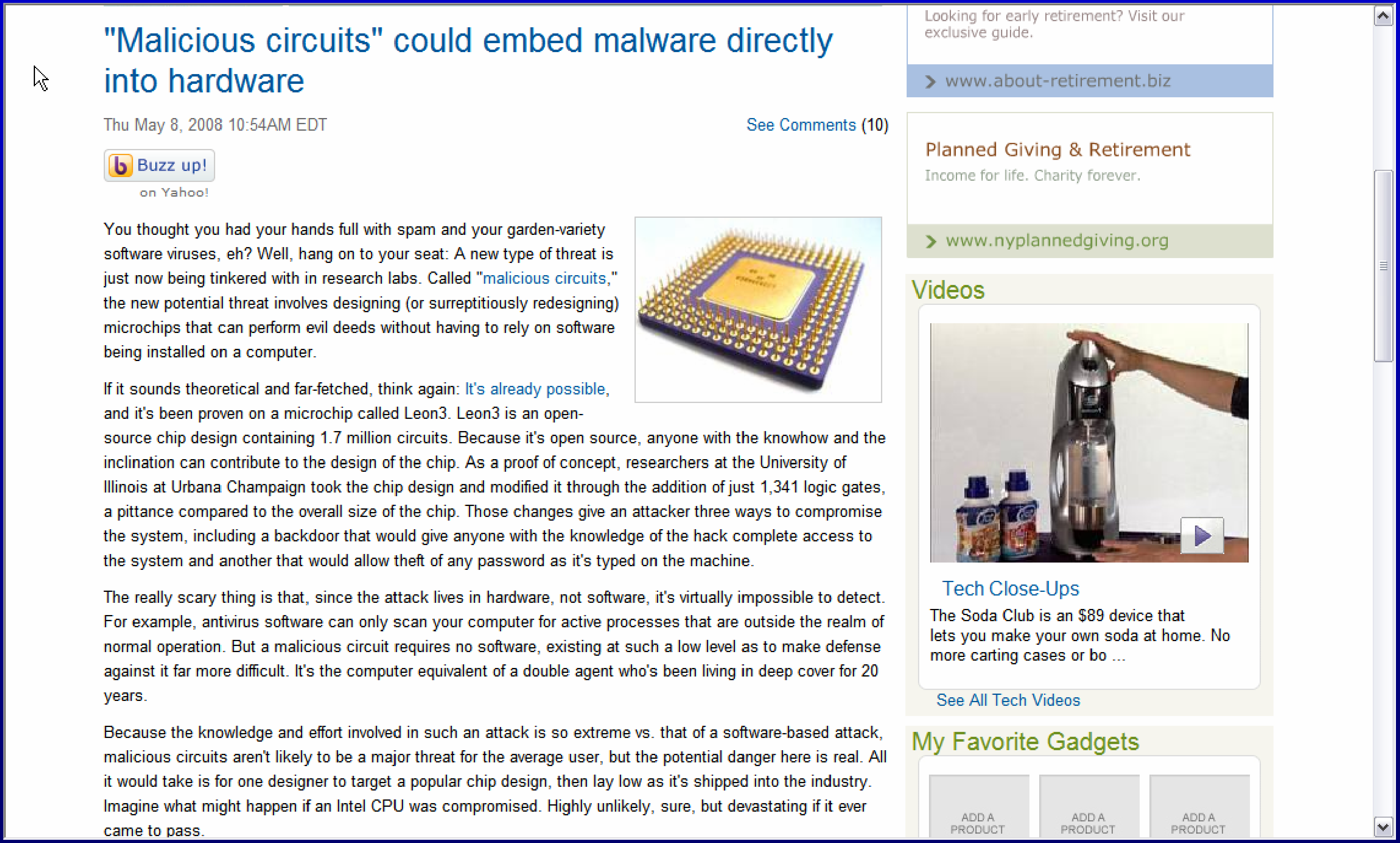
Flash/Proms easy to clone

Counterfeit Products



Source: <http://www.usedcisco.com/press-my-esm_used_cisco_identifying_fake_chisco.aspx>

Future?



"Malicious circuits" could embed malware directly into hardware

Thu May 8, 2008 10:54AM EDT

See Comments (10)

You thought you had your hands full with spam and your garden-variety software viruses, eh? Well, hang on to your seat: A new type of threat is just now being tinkered with in research labs. Called "malicious circuits," the new potential threat involves designing (or surreptitiously redesigning) microchips that can perform evil deeds without having to rely on software being installed on a computer.

If it sounds theoretical and far-fetched, think again: It's already possible, and it's been proven on a microchip called Leon3. Leon3 is an open-source chip design containing 1.7 million circuits. Because it's open source, anyone with the knowhow and the inclination can contribute to the design of the chip. As a proof of concept, researchers at the University of Illinois at Urbana Champaign took the chip design and modified it through the addition of just 1,341 logic gates, a pittance compared to the overall size of the chip. Those changes give an attacker three ways to compromise the system, including a backdoor that would give anyone with the knowledge of the hack complete access to the system and another that would allow theft of any password as it's typed on the machine.

The really scary thing is that, since the attack lives in hardware, not software, it's virtually impossible to detect. For example, antivirus software can only scan your computer for active processes that are outside the realm of normal operation. But a malicious circuit requires no software, existing at such a low level as to make defense against it far more difficult. It's the computer equivalent of a double agent who's been living in deep cover for 20 years.

Because the knowledge and effort involved in such an attack is so extreme vs. that of a software-based attack, malicious circuits aren't likely to be a major threat for the average user, but the potential danger here is real. All it would take is for one designer to target a popular chip design, then lay low as it's shipped into the industry. Imagine what might happen if an Intel CPU was compromised. Highly unlikely, sure, but devastating if it ever came to pass.

Videos

Tech Close-Ups

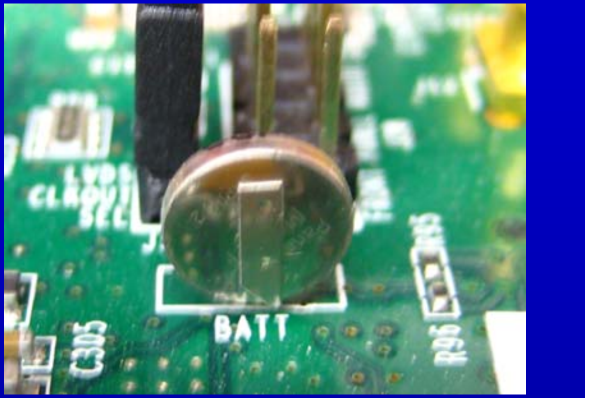
The Soda Club is an $89 device that lets you make your own soda at home. No more carting cases or bo ...

See All Tech Videos

My Favorite Gadgets

AES Security to the rescue?

Xilinx Virtex 4/5
RAM based key – battery backed
 Use JTAG to program key
256 bit key
Accepts bitstreams unencrypted
Keys exposed to CM

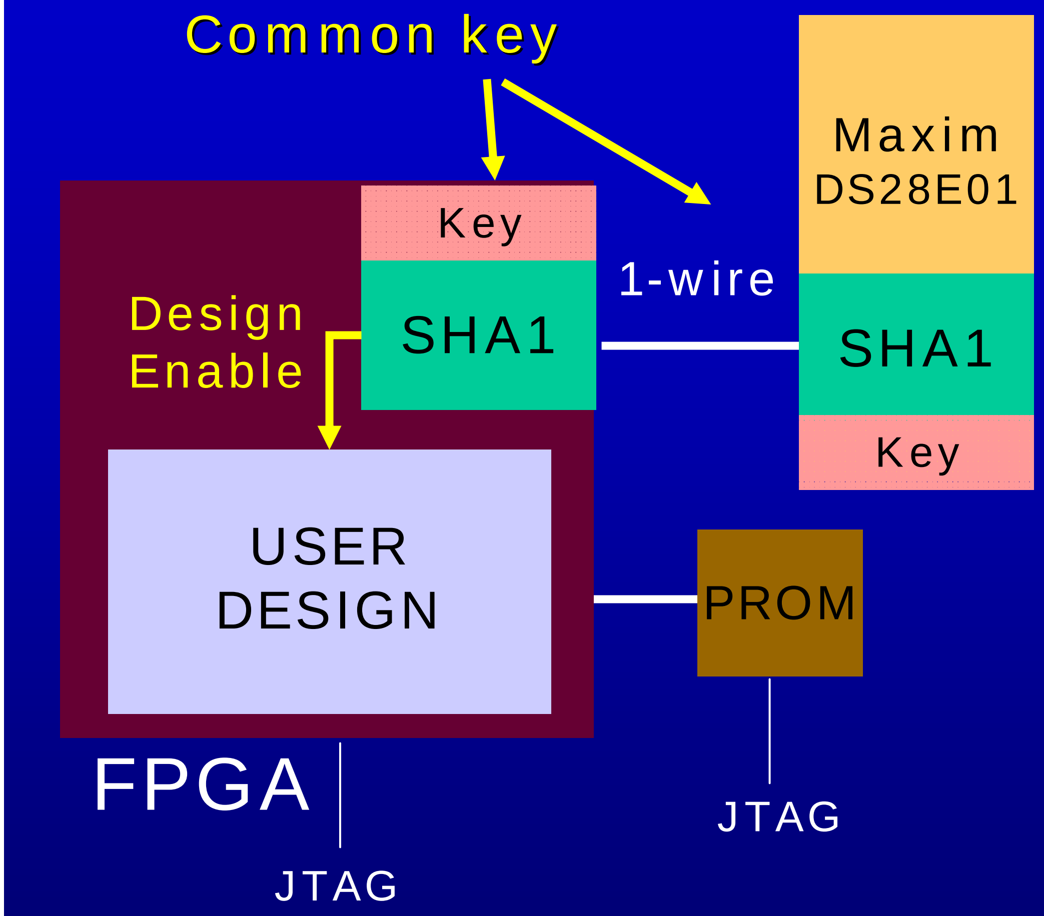


Battery

Altera Stratix III
RAM or ROM
II – ROM based
Need network blaster to program key
256 bit key
Accepts bitstreams unencrypted
Keys exposed to CM

Good for protection of IP
No pre-programming IC
Assumes attacker is not looking to load a trojan bitstream
Not available in Spartans and Cyclones
Battery/Key programmed PER FPGA

Alternate Security



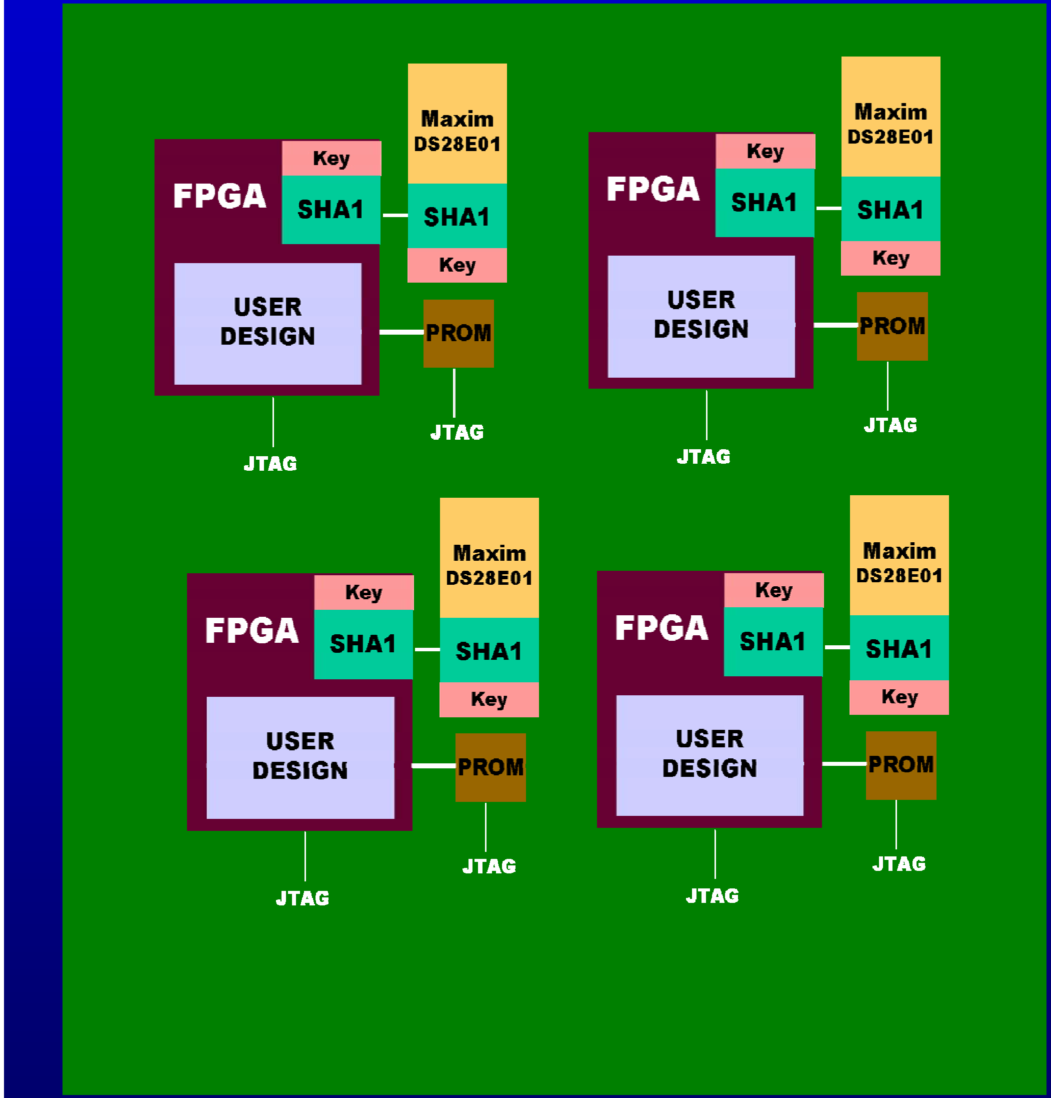
Security initiated by FPGA

Program both FPGA and pre-program Maxim Device with 64 bit SHA1 Key

Some logistics for manufacturing required for OBP over 1-wire
- keys exposed to CM

Trojan in PROM
- PROM/FLASH open to non-authenticated bitstream

Alternate Security



Doesn’t scale
More FPGAs
More OBP

- Longer Manufacturing Times

Higher parts cost

Intellitech

Trojan Bitstreams

Non-authenticated bitstream loaded through JTAG

Need protection:

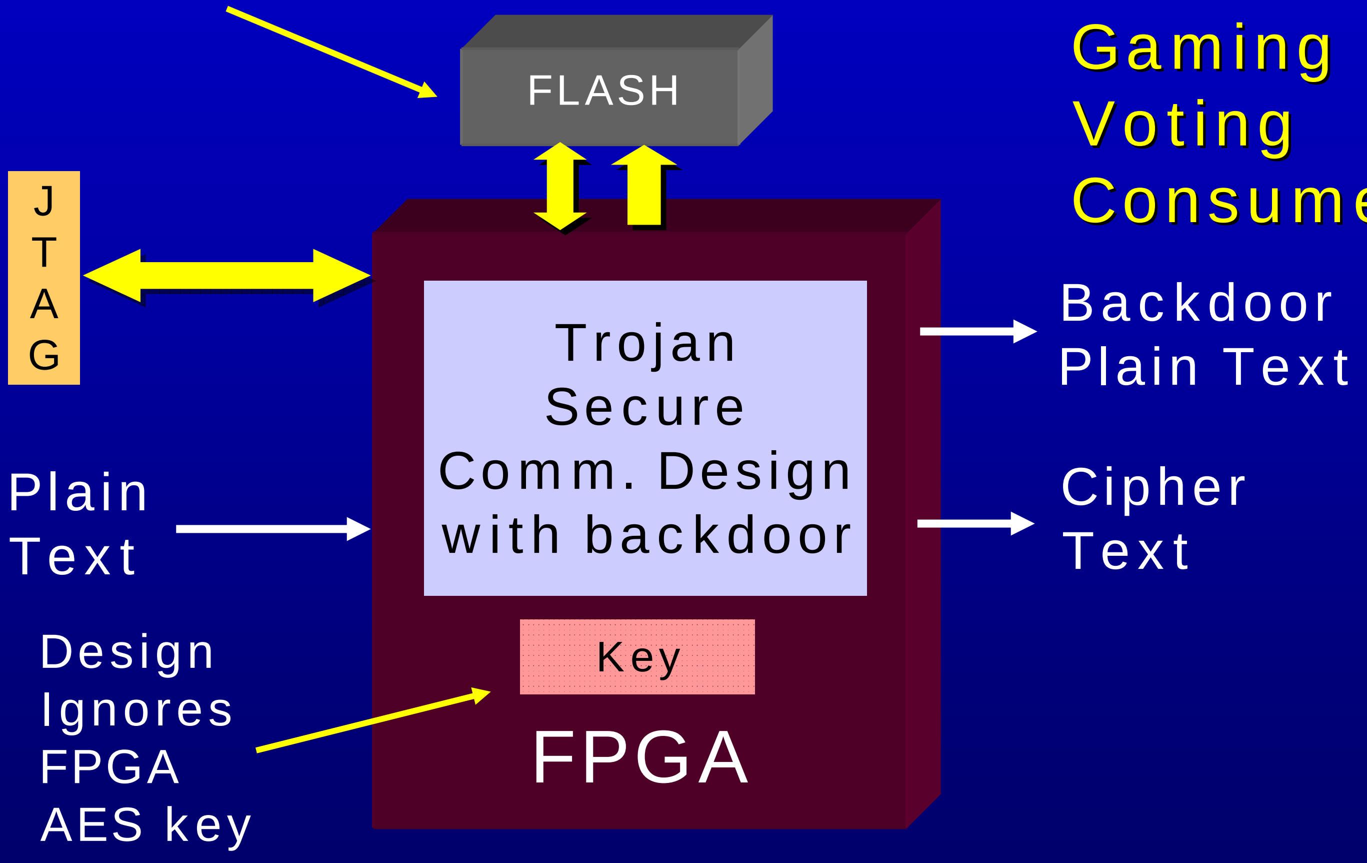
Military

Telecomm

Gaming

Voting

Consumer


FLASH
JTAG
Trojan Secure Comm. Design with backdoor
Key
FPGA
Plain Text
Design Ignores FPGA AES key
Backdoor Plain Text
Cipher Text

A New Approach

We know:

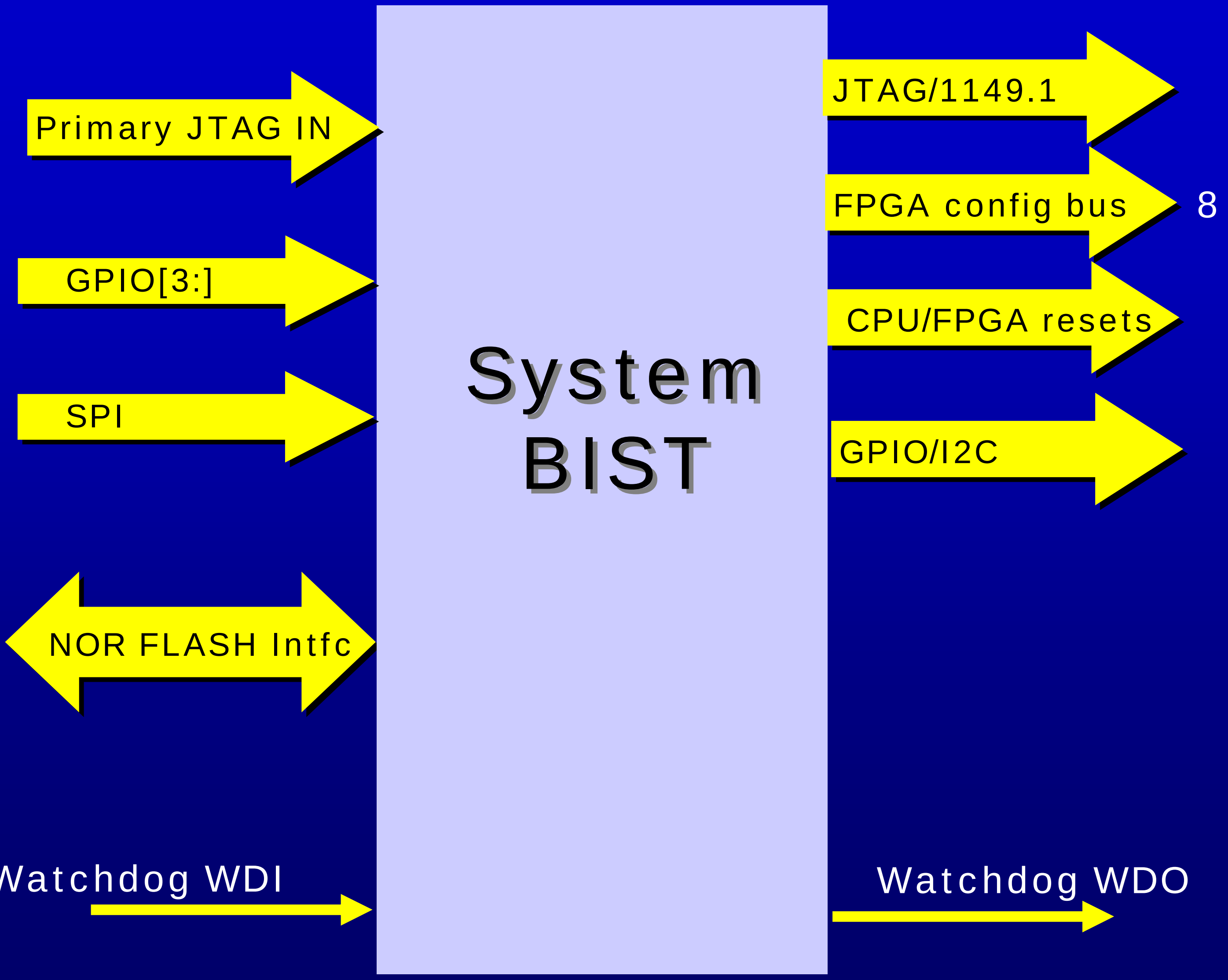
- Tests can be done over 4 wire bus
- Structured test – saves time/re-use
- FPGAs can be programmed multiple ways
- Commodity parts are easily copied/reprogrammed

Goals of New approach:

1) FPGA/Test Data stored tied to configuration device
2) Device manages PCB resets, voltage, FPGA security, watchdog
3) Configure FPGAs based on PCB/FPGAs
4) Updates – FPGAs/EEPROM/CPLDs tied to customer (no open bitstreams)
5) Embedded 1149.1 Structural Tests
6) Downloadable IP to run embedded at-speed tests (RocketIO/Serdes, DDR etc)

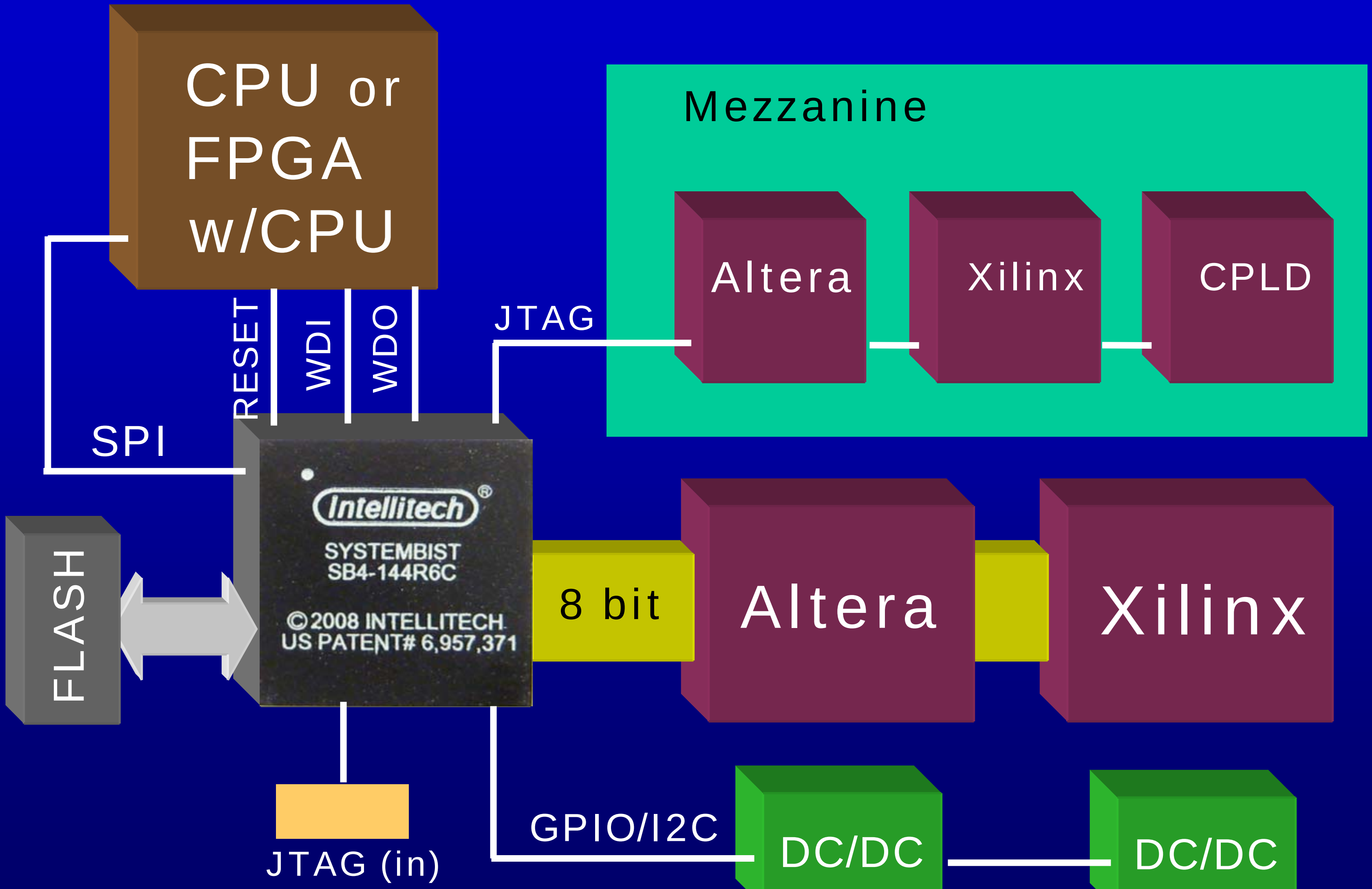
Intellitech

SystemBIST



Intellitech

SystemBIST

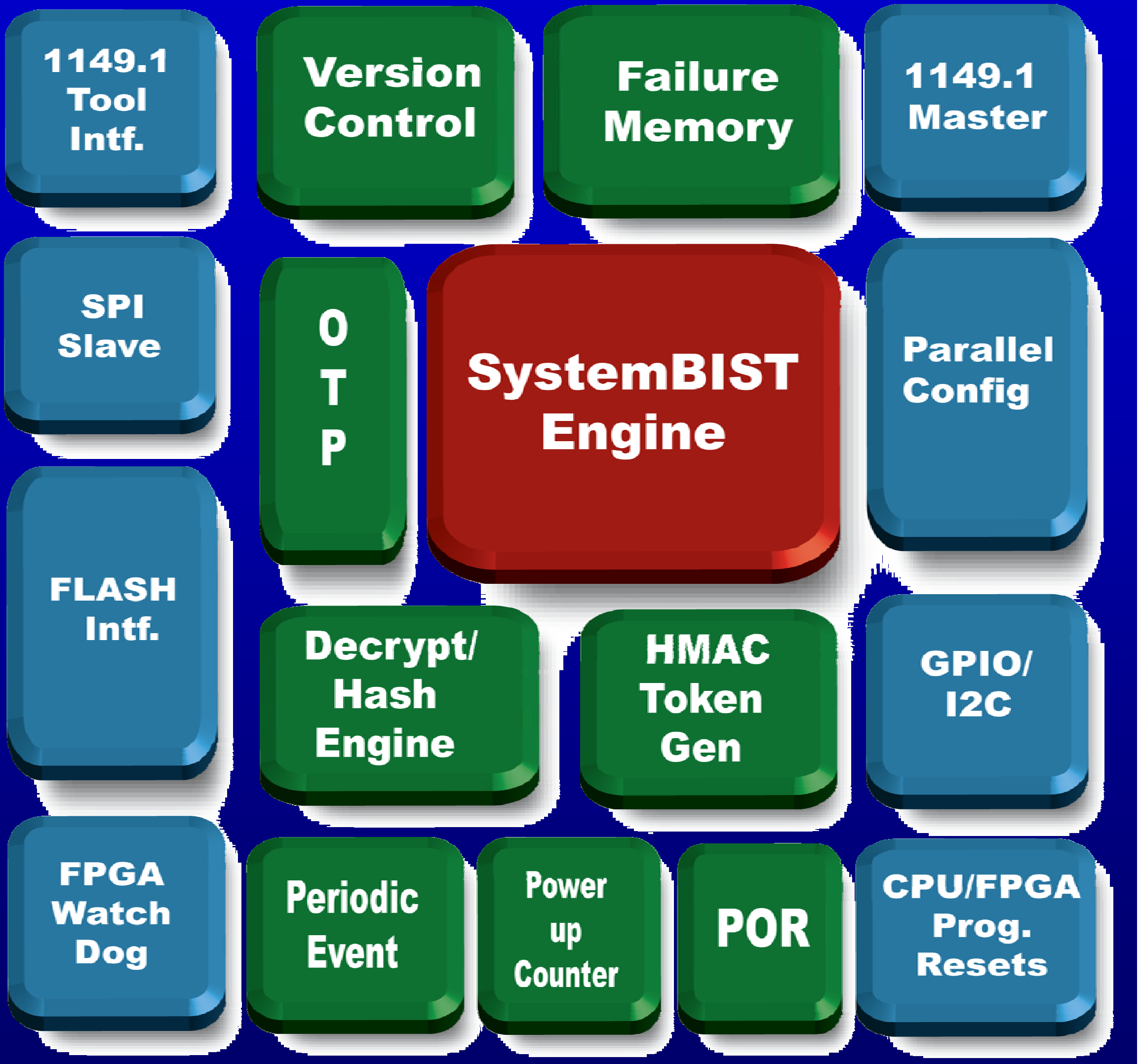

CPU or FPGA w/CPU
Mezzanine
Altera
Xilinx
CPLD
JTAG
RESET
WDI
WDO
SPI
Intellitech®
SYSTEMBIST
SB4-144R6C
©2008 INTELLITECH
US PATENT# 6,957,371
FLASH
8 bit
Altera
Xilinx
GPIO/I2C
DC/DC
DC/DC
JTAG (in)

SystemBIST IC

OTP programmed at time of order

Two 128bit keys programmed
-Algorithm creates Third key

Flash data secured By key and to
-Unique customer identifier
-Unique serial ID



Intellitech

SPI & OTP

Access to OTP Customer Code

- no other entity can have this code

Access to OTP CM Code

Access to OTP Unique Serial Number

OTP holds two 128-bit keys – not accessible through SPI

Access to SystemBIST execute/Run

Access to SystemBIST failure code

- each test/fpga config has failure codes

Access to SystemBIST failuremap

Access to SystemBIST update mechanism

Periodic Engine + SHA1

-Random data generated by FPGA

-SystemBIST Reads via JTAG

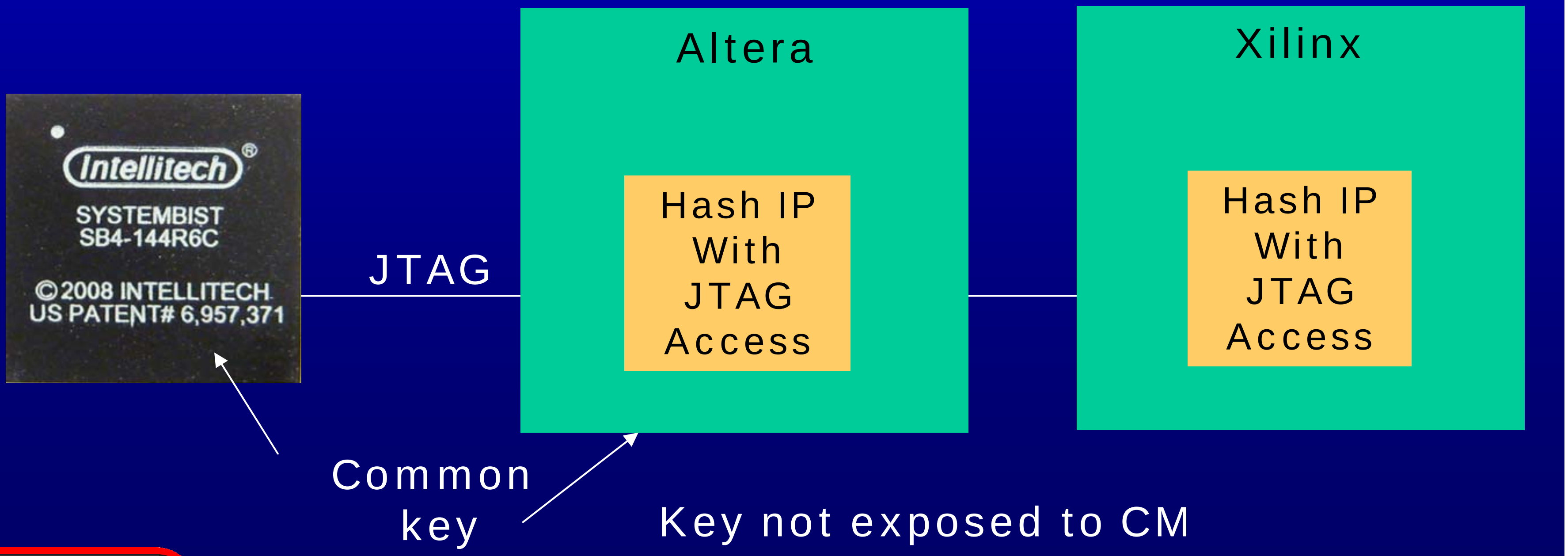
-Generates Hash

-Hash Written via JTAG

- Good matching Hash enables user logic

-2nd 'OK' Hash Read via JTAG

- SystemBIST clears FPGA on bad hash



JTAG

Altera

Hash IP With JTAG Access

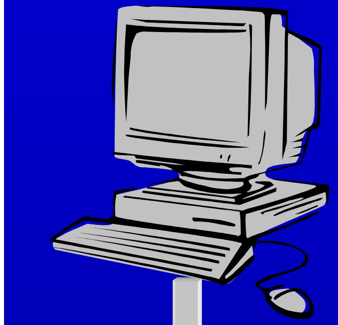
Xilinx

Hash IP With JTAG Access

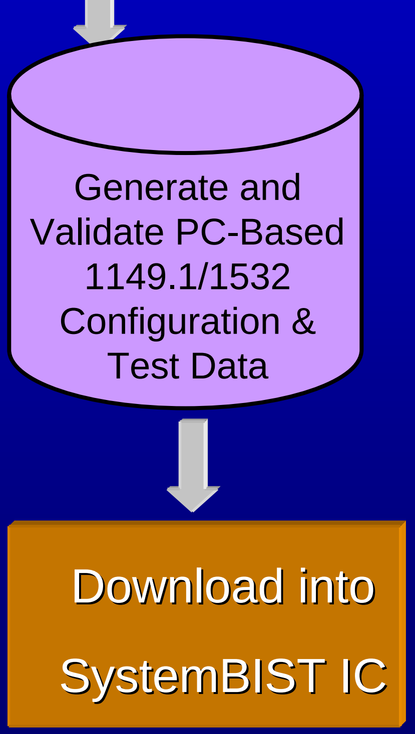
Common key

Key not exposed to CM

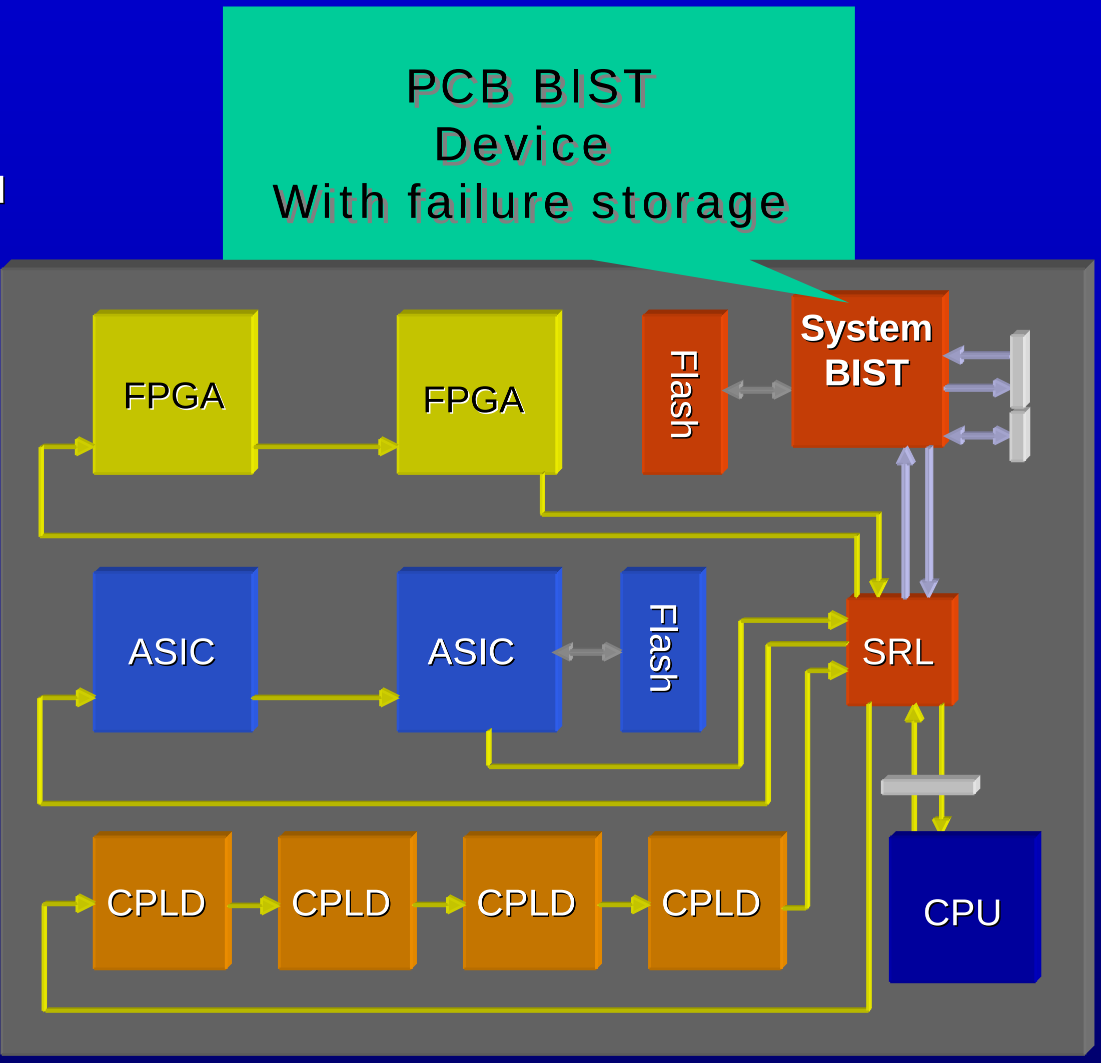
PC defines config/test strategy



PC-Based 1532 Configuration & 1149.1 Test Development and Validation Tools



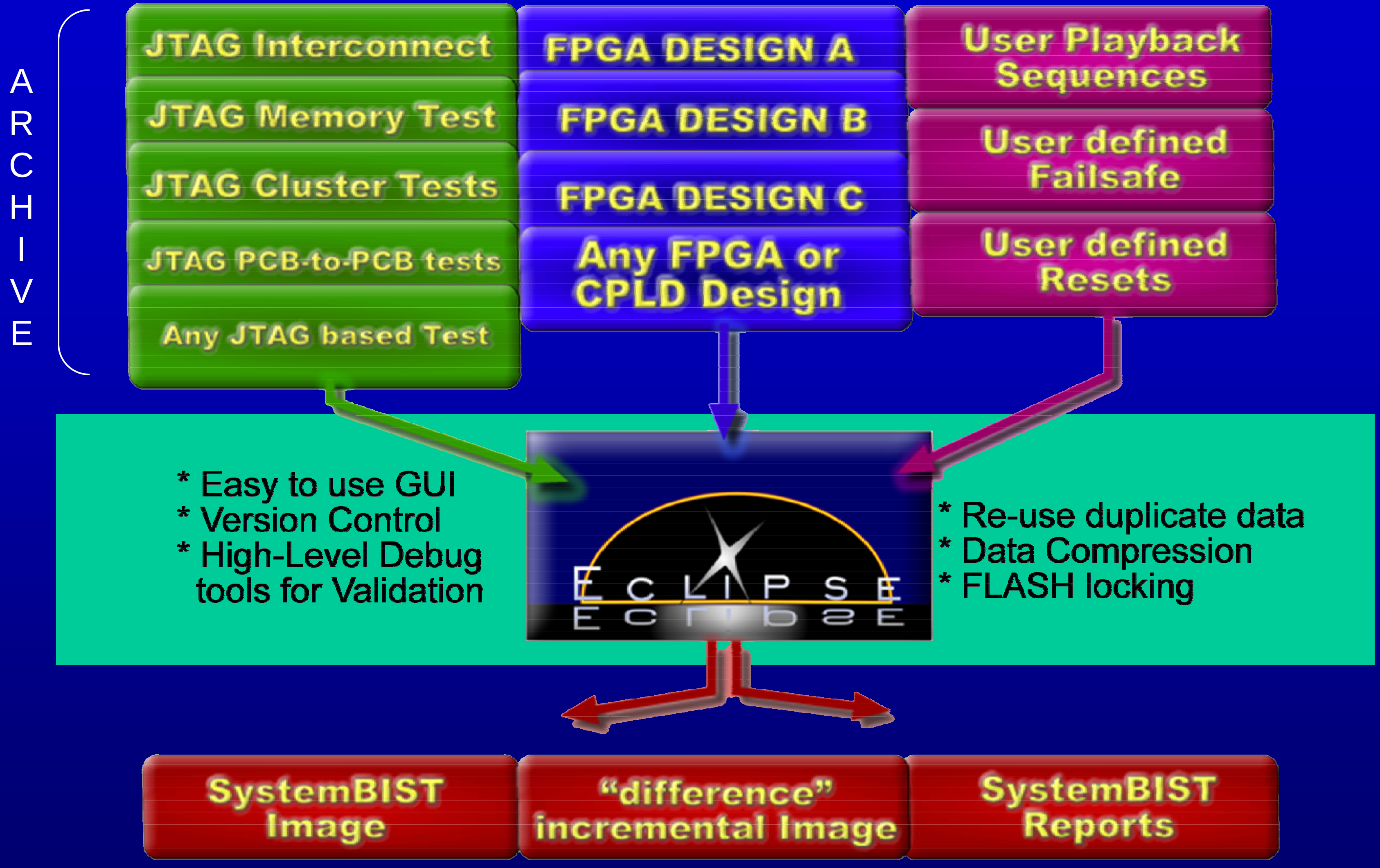
Generate and Validate PC-Based 1149.1/1532 Configuration & Test Data
Download into SystemBIST IC



PCB BIST Device With failure storage

Intellitech

Software generates secure images



A R C H I V E

JTAG Interconnect

JTAG Memory Test

JTAG Cluster Tests

JTAG PCB-to-PCB tests

Any JTAG based Test

FPGA DESIGN A

FPGA DESIGN B

FPGA DESIGN C

Any FPGA or CPLD Design

User Playback Sequences

User defined Failsafe

User defined Resets

- * Easy to use GUI
- * Version Control
- * High-Level Debug tools for Validation

- * Re-use duplicate data
- * Data Compression
- * FLASH locking

SystemBIST Image

“difference” incremental Image

SystemBIST Reports

Images secured to customer key

Eco-system support

WatchDog Suite

- program what you would like to do with watchdog timeout
- JTAG capture of CPU state?

Periodic Suite

- program what to do periodically
- voltage margining, monitor temp, monitor fpga security

Powerup Suite

- control power sequence
- control resets

Eco-system features

64bit serial number

- Available to CPU over SPI
- can use for product uniqueness

IP Protection

- Tools download unique image to FLASH per PCB
- Prevent copies/trojans

Version control

- Prevent wrong version updates

User FLASH area available over SPI

I2C – ‘bit-bang’ to control DC/DC converter

Failure logging

- Failing ‘bits’ all stored in flash
- Enables Standard JTAG type diags
- Archive name stored in failure mem
  - used by offline diagnostics

Power-up logging

- counts the number of pwr-ups

Complex FPGA Config Sequence

- Enables in-the-field FPGA design targeting
    - Ex. Load different DSP algorithms based on environment
- Enables in-the-field updating of system non-volatiles
- Enables JTAG based Self-test

Suite 1:

1 Check_Scan_Cain.script

2 Check U3 DEVICE_ID

3 Branch to 6 if U3 is XC4VLX100

4 Program_DesignA.bit in U3

5 Branch 7

6 Program_DesignB.bit

7 Program_DesignB.bit in U4

Mark this bitstream for failsafe

Suite 2:

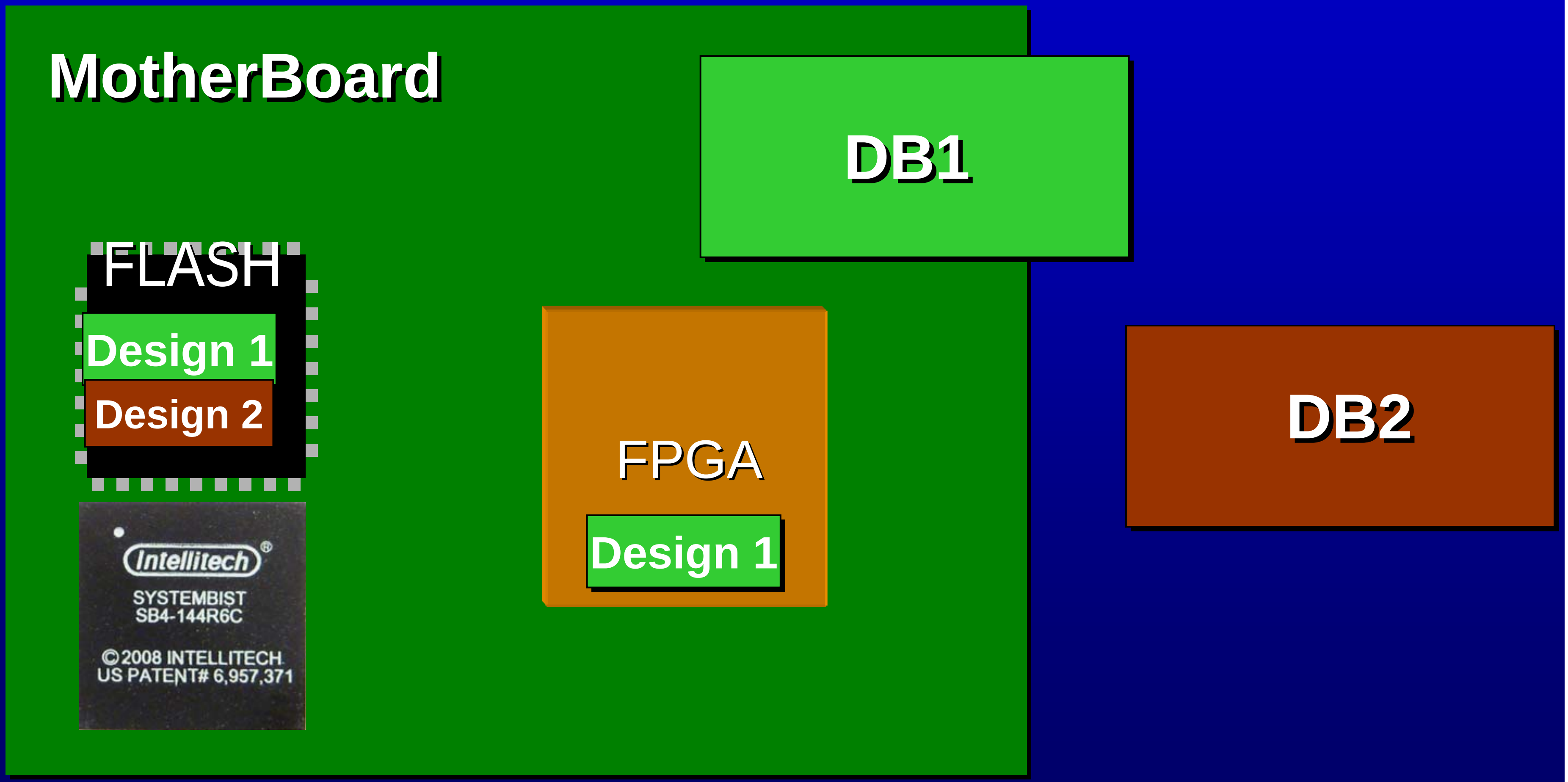
1 Check_Scan_Cain.script
2 Interconnect.svf
3 Test_ASIC.script

4 Update_CPLD_U9.bit
5 Update_CPLD_U10.bit

6 Update_SPI_Prom.script

FPGA programming with Branching

FPGA gets programmed based on which Daughter PCB is plugged in



MotherBoard

FLASH

Design 1

Design 2

FPGA

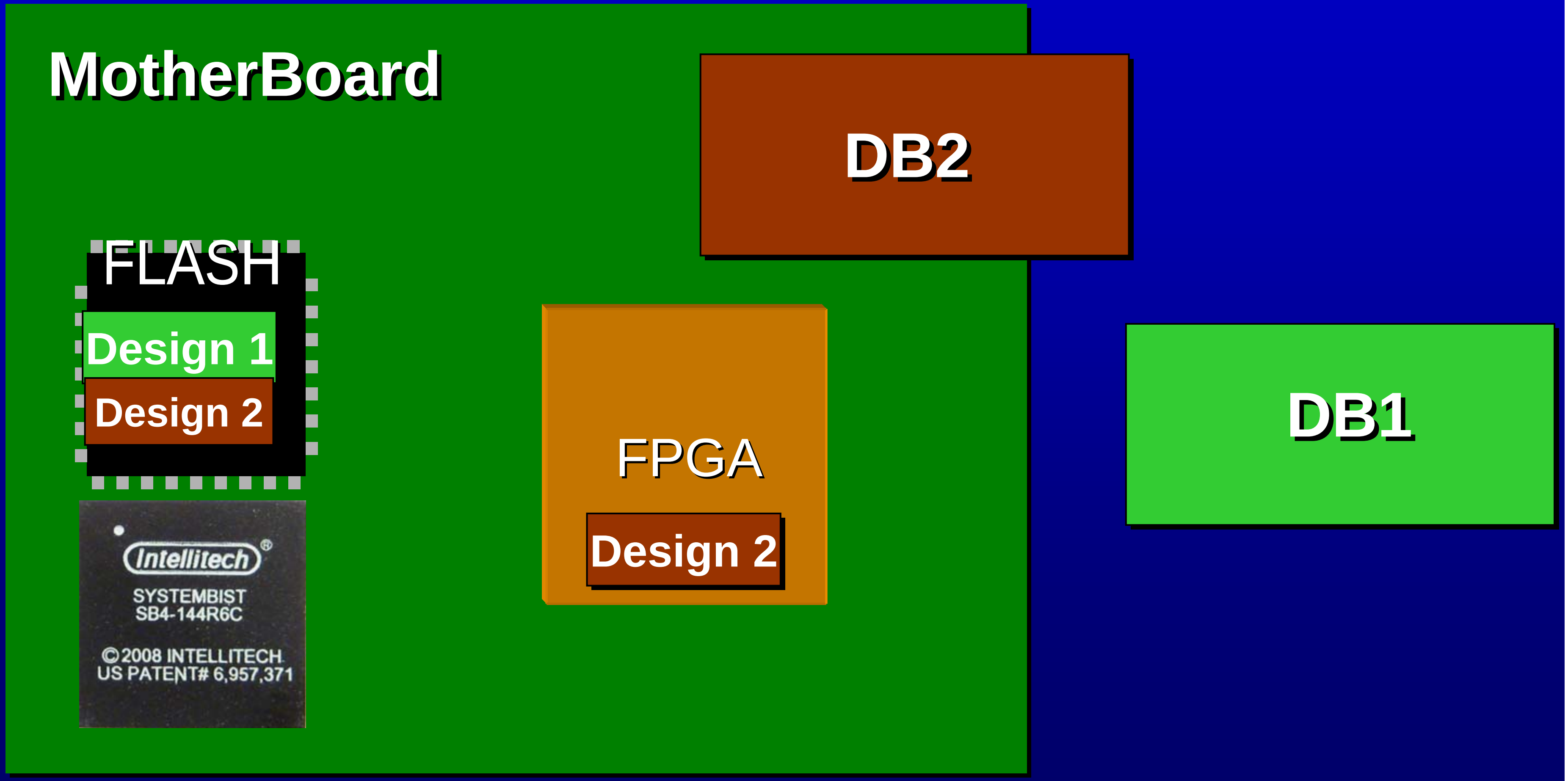
Design 1

DB1

DB2

FPGA programming with Branching

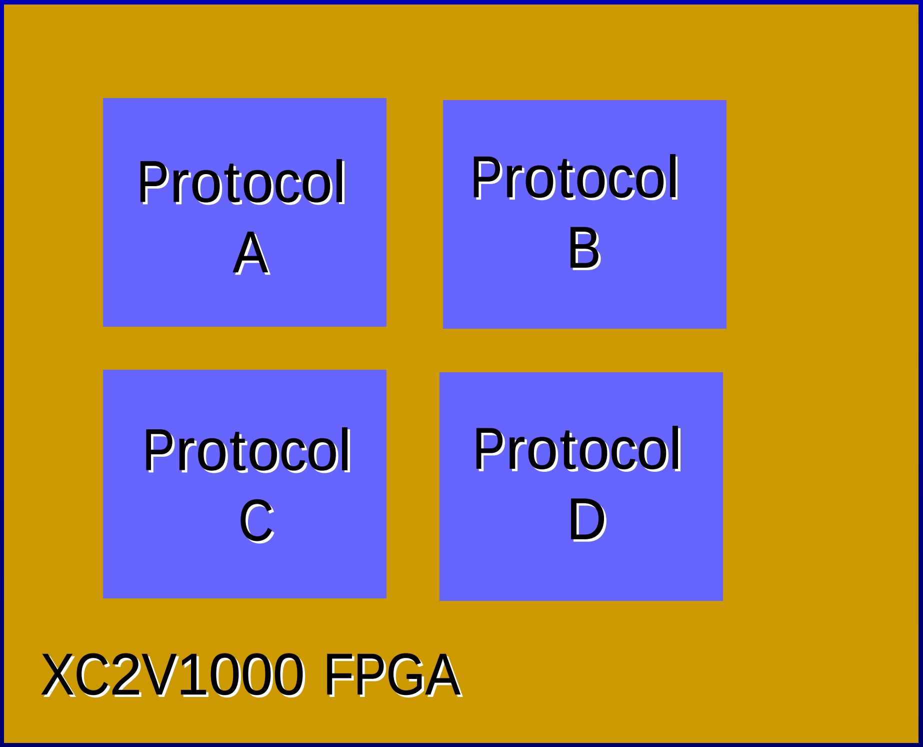
FPGA gets programmed based on which Daughter PCB is plugged in



Smart FPGA Configuration Devices

Consider: FPGA Design needs to support 4 protocols

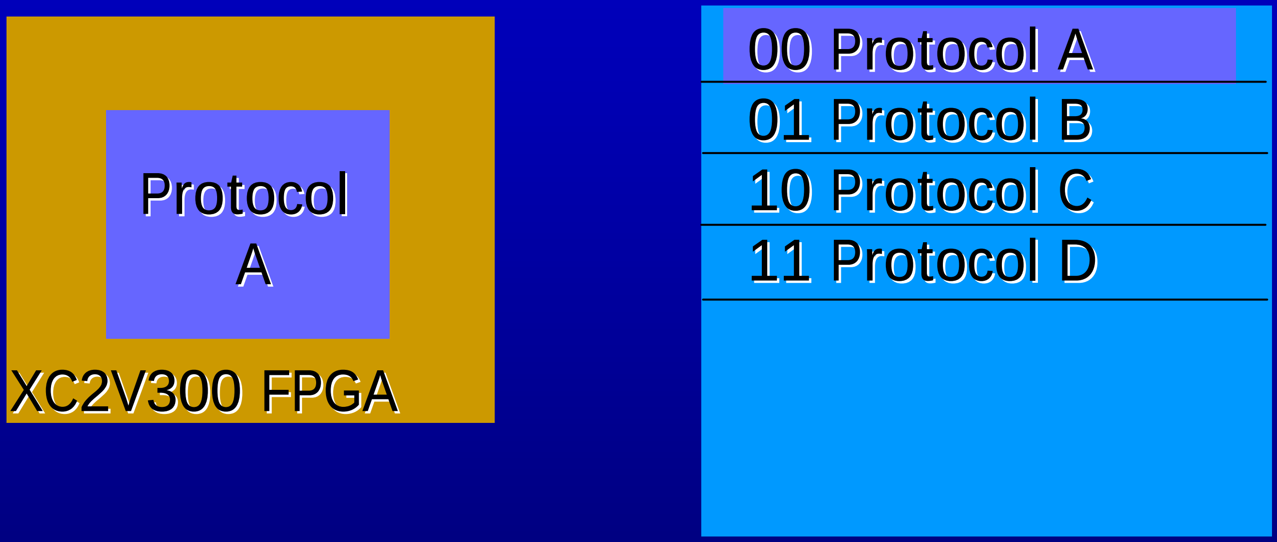
- Each protocol is 250K Gates
- Designs fit in a 1M Gate FPGA
- Customer only needs one protocol at a time



Smart Configurator makes decisions

1 Protocol Design will fit in a 300K Gate FPGA

- SystemBIST loads designs on demand



Protocol A

XC2V300 FPGA

00 Protocol A

01 Protocol B

10 Protocol C

11 Protocol D

SystemBIST Memory

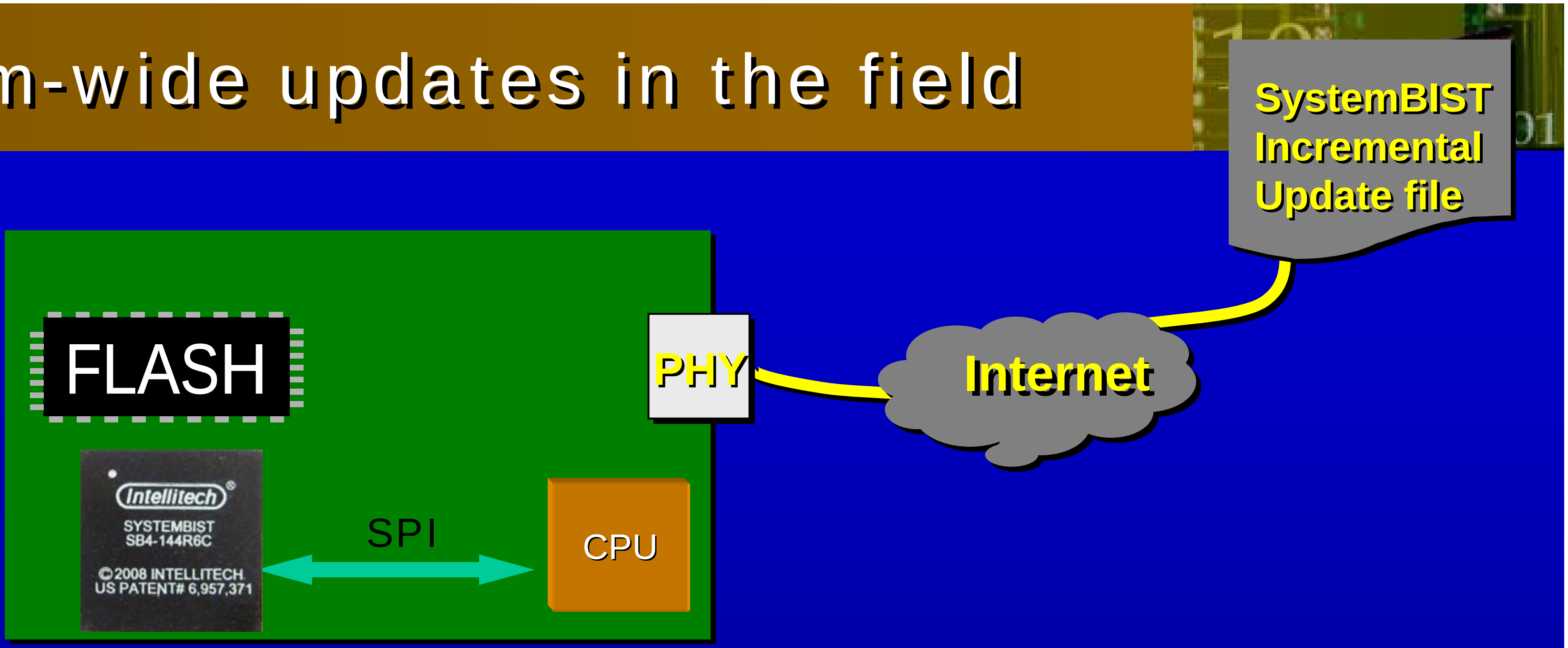
Sequences – files stored in archive

All data used to create an image for deployment
In the field is stored in an Eclipse "Archive"

Each image and update image has an associated
Archive.

Archives are used by diagnostic managers to take
In-the-field failure information and perform
Diagnostics on the failure using the original
Source (bitstreams, JTAG test files)

System-wide updates in the field



-Software generates protected image

- tied to internal customer key/code

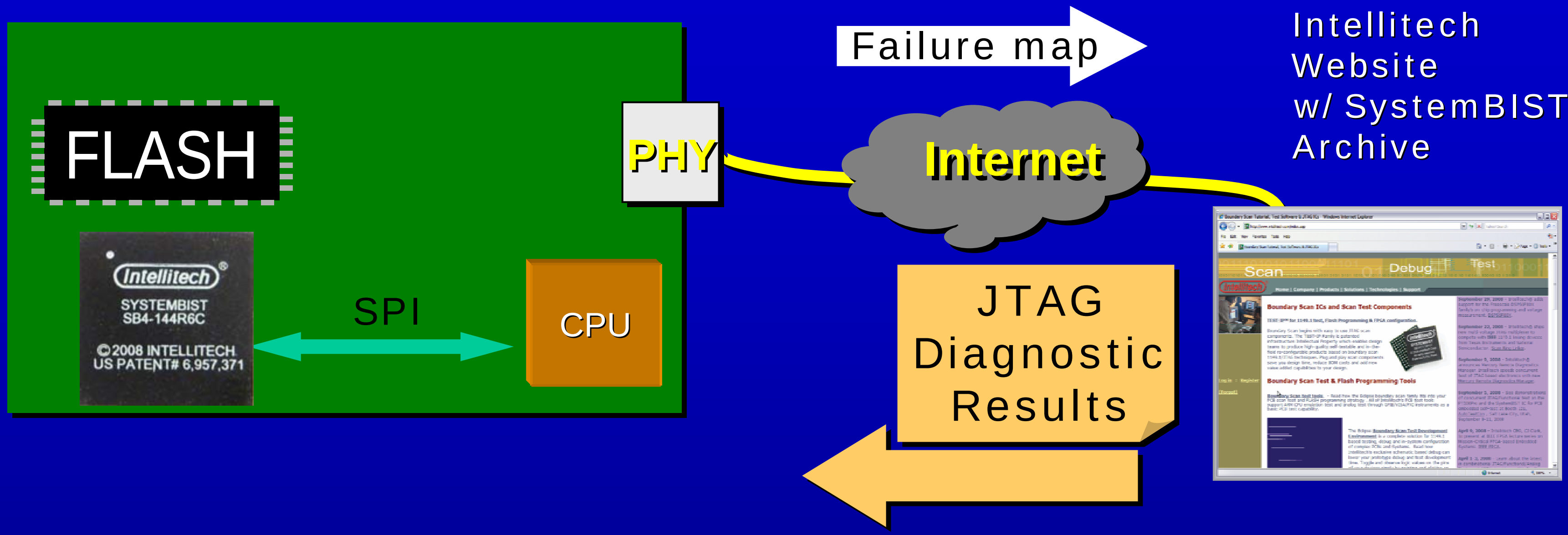
-CPU – accepts file over internet medium

-CPU Source code only needs to write file to SystemBIST over SPI interface.

-Where to put the file, when to erase, Correct version, correct archive/product name

All managed by SystemBIST

Diagnostics in the field



- •SystemBIST archives uploaded to Intellitech website during development process
- •SystemBIST records all power-up events and all FPGA config failures, all test failures Bit by bit.
- •Failure memory uploaded to Intellitech's website SystemBIST Remote Diagnostics Module

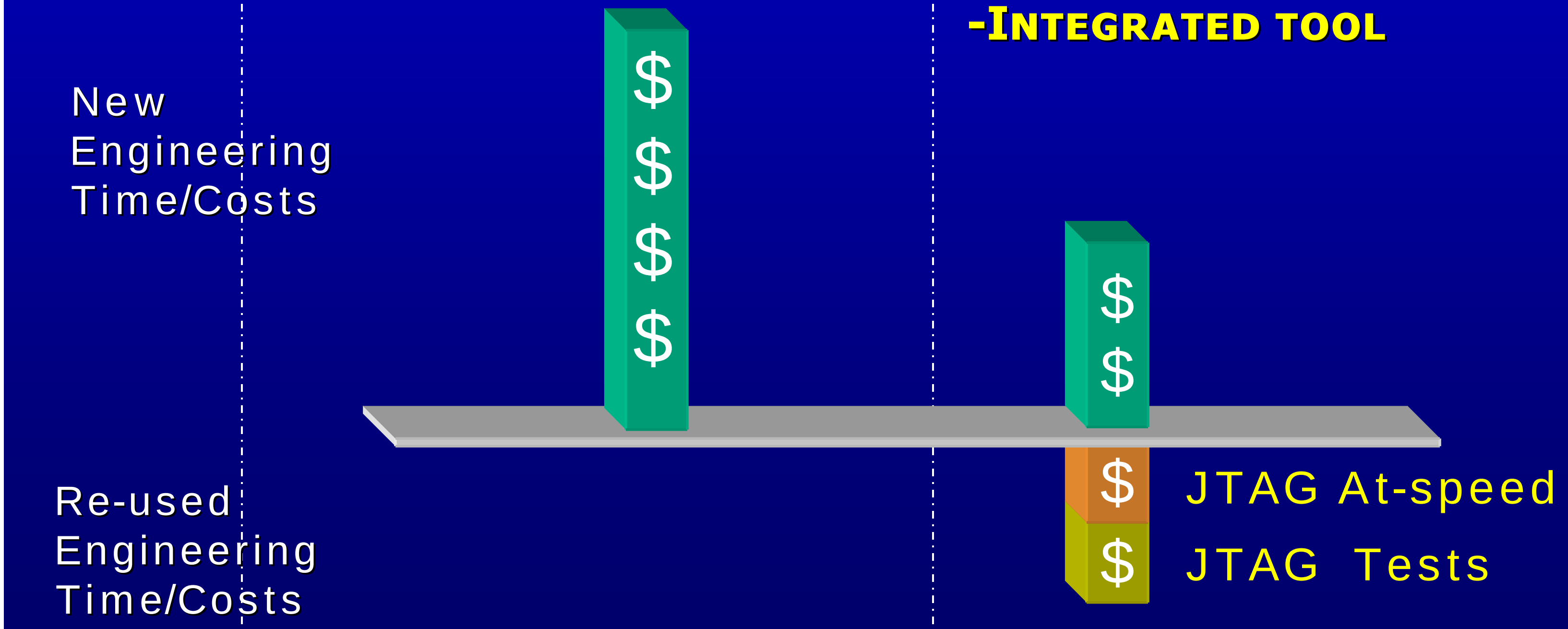
Re-using manufacturing tests

Traditional CPU Firmware/software

- Develop tests
- FPGA config/update
- Security
- Watchdog/Resets
- Full custom/no outsource
- piecemeal tool support

SystemBIST

- Reuse manufacturing test
- Pre-built config/update
- Pre-built Security
- Physically unclonable
- Pre-built Watchdog/Resets
- Standardized – 3rd party help!
- Integrated tool



Conclusion

Active Device monitors FPGA authentication

1) Manages updates and other eco-system functions – reset/watch-dog
2) IEEE standards enable structured embedded PCB test with diags
3) Flexible FPGA configuration
5) Security, trojan protection
6) Test & FPGA Config de-coupled from system resources
   - outsource-able
   - re-usable

Further Reading

Using the Design Security Feature in Stratix II and Stratix II GX Devices, Altera Corporation, July 2008. http://www.altera.com/literature/an/an341.pdf

Trusted Design in FPGAs, Steve Trimberger, Xilinx, Design Automation Conference, 2007

http://videos.dac.com/44th/papers/1_2.pdf

Authentication of FPGA Bitstreams:

Why and How, Saar Drimer, ARC 2007

http://www.springerlink.com/content/t71pqn4g7565w806/

A Code-less BIST Processor for Embedded Test and in-system configuration of Boards and Systems, CJ Clark, Intellitech Corp, Mike Ricchetti, ATI Research, ITC 2004,

http://www.intellitech.com/pdf/itc04sb.pdf

Design Security in Stratix III FPGAs, Altera Corporation

http://www.altera.com/products/devices/stratix-fpgas/stratix-iii/overview/architecture/st3-design-security.html

Secure Update Mechanism for Remote Update of

FPGA-Based System, Benoît Badrignans1,2, Reouven Elbaz3 and Lionel Torres. SEIS 2008, http://ieeexplore.ieee.org/Xplore/login.jsp?url=/iel5/4569831/4577669/04577703.pdf?temp=x

Further Reading

Physical Unclonable Functions for Device Authentication and Secret Key Generation
G. Edward Suh, Srinivas Devadas
http://videos.dac.com/44th/papers/1_3.pdf

Xilinx® FPGA IFF Copy Protection with 1-Wire SHA-1 Secure Memories, Maxim,
http://www.maxim-ic.com/appnotes.cfm/an_pk/3826

An FPGA Design Security Solution Using a Secure Memory Device, Altera,
http://www.altera.com/literature/wp/wp-01033.pdf

Altera Configuration Handbook
http://www.altera.com/literature/lit-config.jsp
Xilinx Virtex-5 FPGA User Guide
http://www.xilinx.com/support/documentation/user_guides/ug190.pdf

Holistic FPGA Configuration

CJ Clark is the president and CEO of Intellitech Corp.
He was the elected chairperson of the IEEE 1149.1 JTAG working group from 1996 to 2002. He has been active in other IEEE working groups and has presented at International Test Conference, TECS (Testing Embedded Cores-Based Systems) Workshop, the Board Test Workshop, Ottawa Test Workshop and VLSI Test Symposium.

CJ serves on the University of New Hampshire College of Engineering and Physical Science (CEPS) Advisory Board. He also serves on the UNH Department of Electrical Engineering Advisory Board. He is co-inventor on three US patent related to scan-based test, two Canadian, one Taiwanese patent with others pending world-wide. His first job in test was in 1978 with Plantronics/Wilcom.