Class 413

Rapidly Developing Embedded Systems Using Configurable Processors

Steven Knapp
(sknapp@triscend.com)

Triscend Corporation
www.triscend.com

(Booth 160)

Triscend
The Configurable Processor Company

EMBEDDED SYSTEMS CONFERENCE SPRING

Agenda

- What is a Configurable Processor?
- Configurable Processors architectures
- Building custom peripherals
- Hardware/software trade-offs, algorithmic acceleration
- Comparing the alternatives
- Configurable Processor technical challenges
  - Communication
  - Software development environment
  - Debugging
- Summary

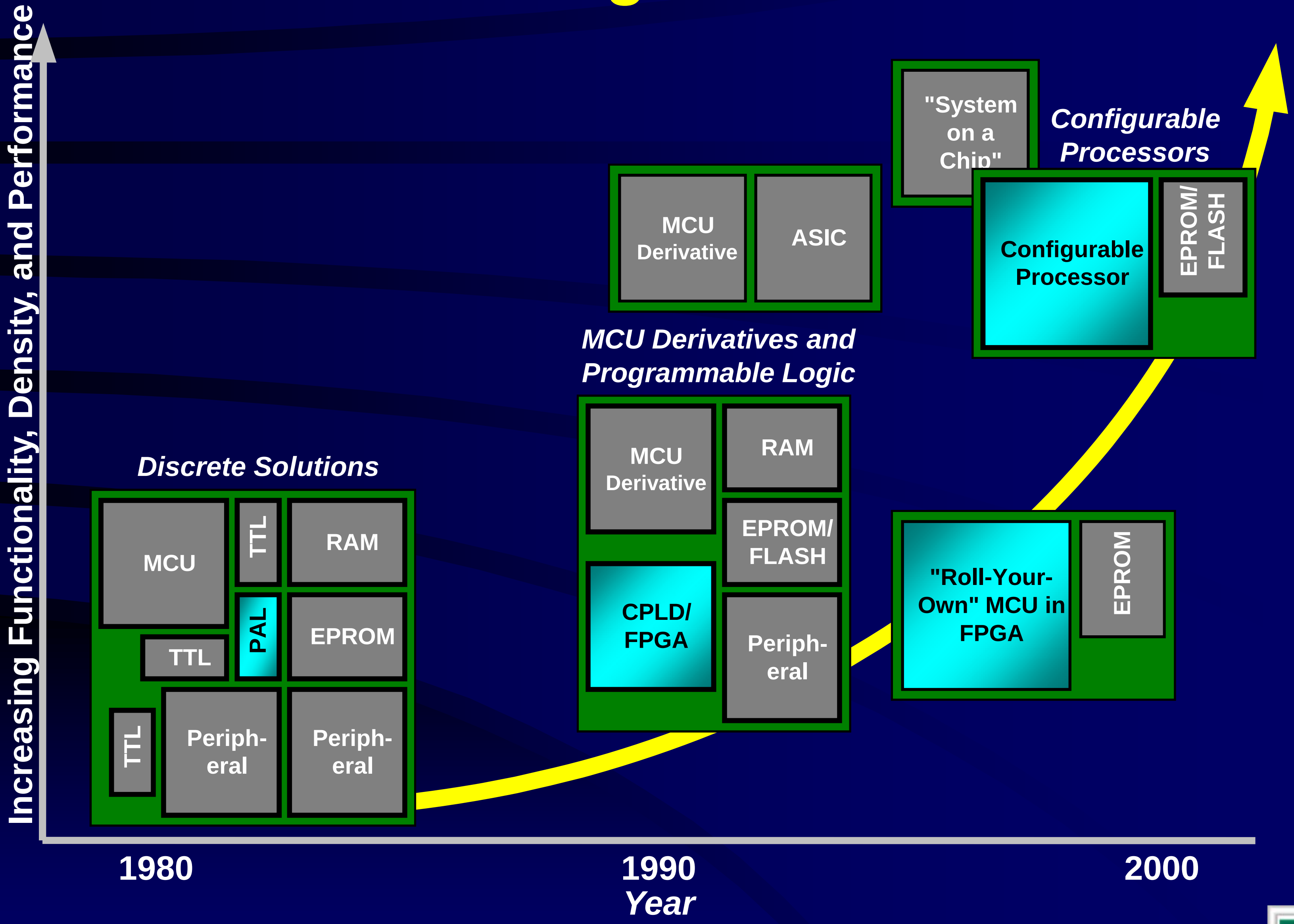
Terminology Used in this Session

- Configurable Processor
  - Embedded processor core
  - Programmable logic to build peripherals
  - Dedicated System Bus
  - On-chip memory
- User-Definable Processor
  - Parameterized or changeable processor hardware description
  - Typically synthesizable from VHDL/Verilog
  - Usually targeted to ASIC or system-on-a-chip
  - Examples: ARC, Tensilica, etc.

Trends Toward the Configurable Processor

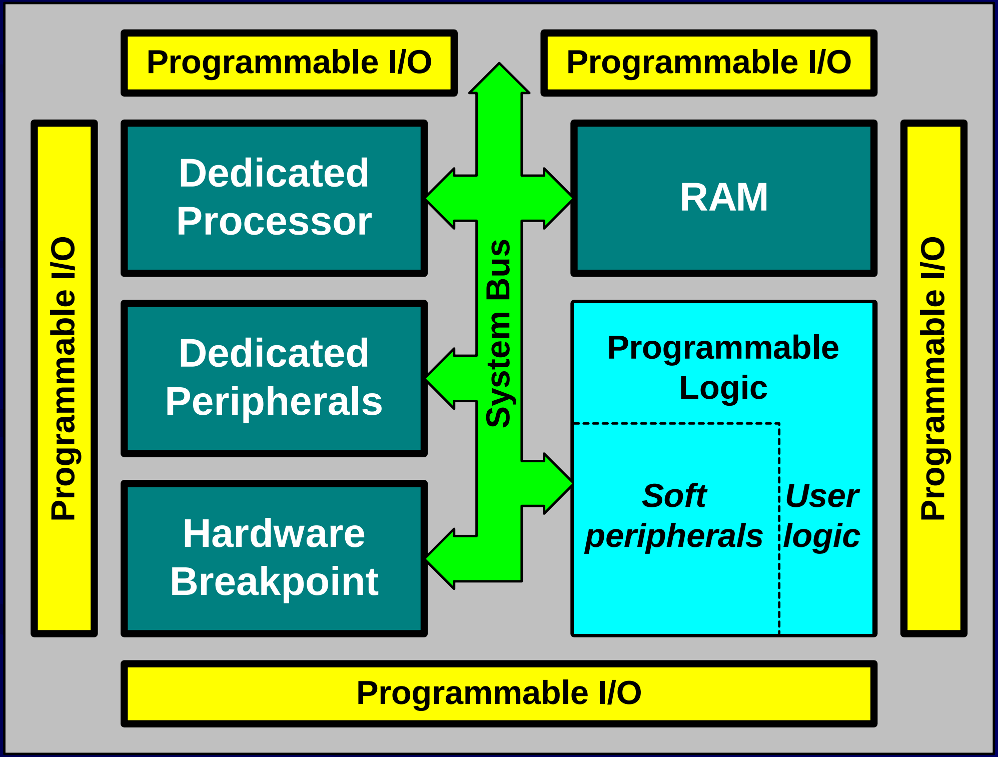
- Decreasing Time-to-Market
  - Fast iterations
  - Fast in-system, real-time debugging
  - Fast component availability
- More Adaptability
  - During design and debug
  - In the field or in the application
- Higher Performance
  - Match the architecture to the problem
  - Fast response to real-time events
  - Parallel operations
- Increased Differentiation
- Ever-improving Process Technology

Toward a Configurable Processor



What Is a Configurable Processor?

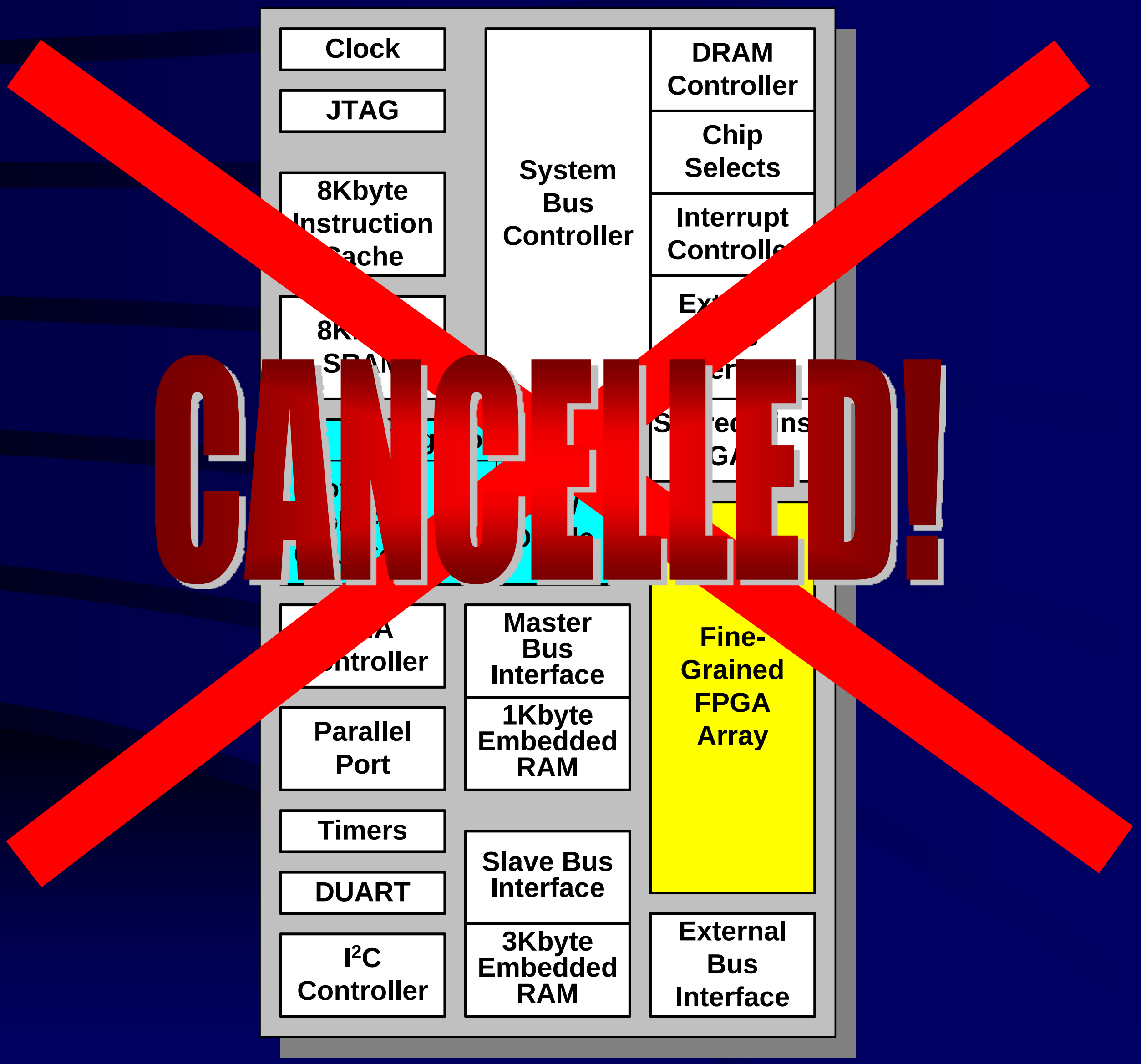
- Industry-standard processor
- Dedicated bus
- Programmable Logic
  - Soft peripherals
  - User-defined functions
  - Hardware acceleration
- On-Chip Memory



Configurable Processors

Vendor/ Family	Processor	Status	Dedicated Resources	Programmable Resources	Embedded Bus Structure
Triscend/ E5	8032 "Turbo"	Sampling	2-channel DMA 8K-64K bytes RAM Hardware debug JTAG	Triscend Coarse-grained, bus oriented	8-bit Data 32-bit Address
Triscend	ARM 7TDMI	In Development		Triscend Coarse-grained, bus oriented	32-bit Data 32-bit Address
Motorola/ CORE+	ColdFire	Cancelled	2-channel DMA 3K bytes RAM DRAM controller Hardware debug	Motorola MPA Fine-grained	Multiple busses Unknown format
National/ NAPA 1000	Compact-RISC	In Development	16K RAM 8x256 RAM Timer JTAG debugger	Concurrent Fine-grained	
Siemens	TriCore	Plans Announced		Gatefield Fine-grained	
Atmel	?	Plans Announced		Atmel AT40K Coarse-grained	
SIDSA/ FIPSOC	8031	Sampling?	Programmable analog	SIDSA Coarse-grained	None, Memory-mapped

Motorola CORE+



CANCELLED!

Clock

JTAG

8Kbyte Instruction Cache

8K SRA

System Bus Controller

DRAM Controller

Chip Selects

Interrupt Controlle

Ext

S red ins GA

A ntroller

Master Bus Interface

1Kbyte Embedded RAM

Fine-Grained FPGA Array

Parallel Port

Timers

DUART

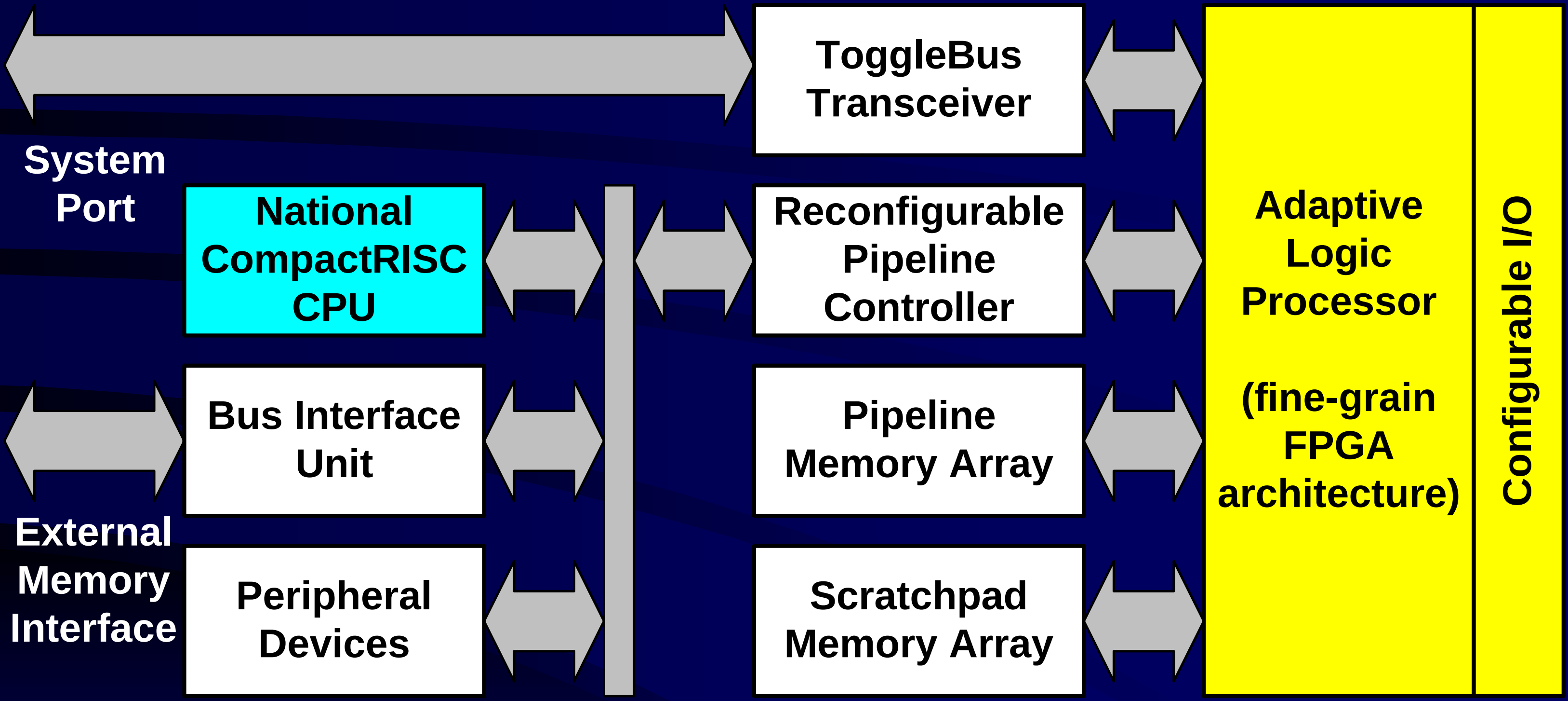
I²C Controller

Slave Bus Interface

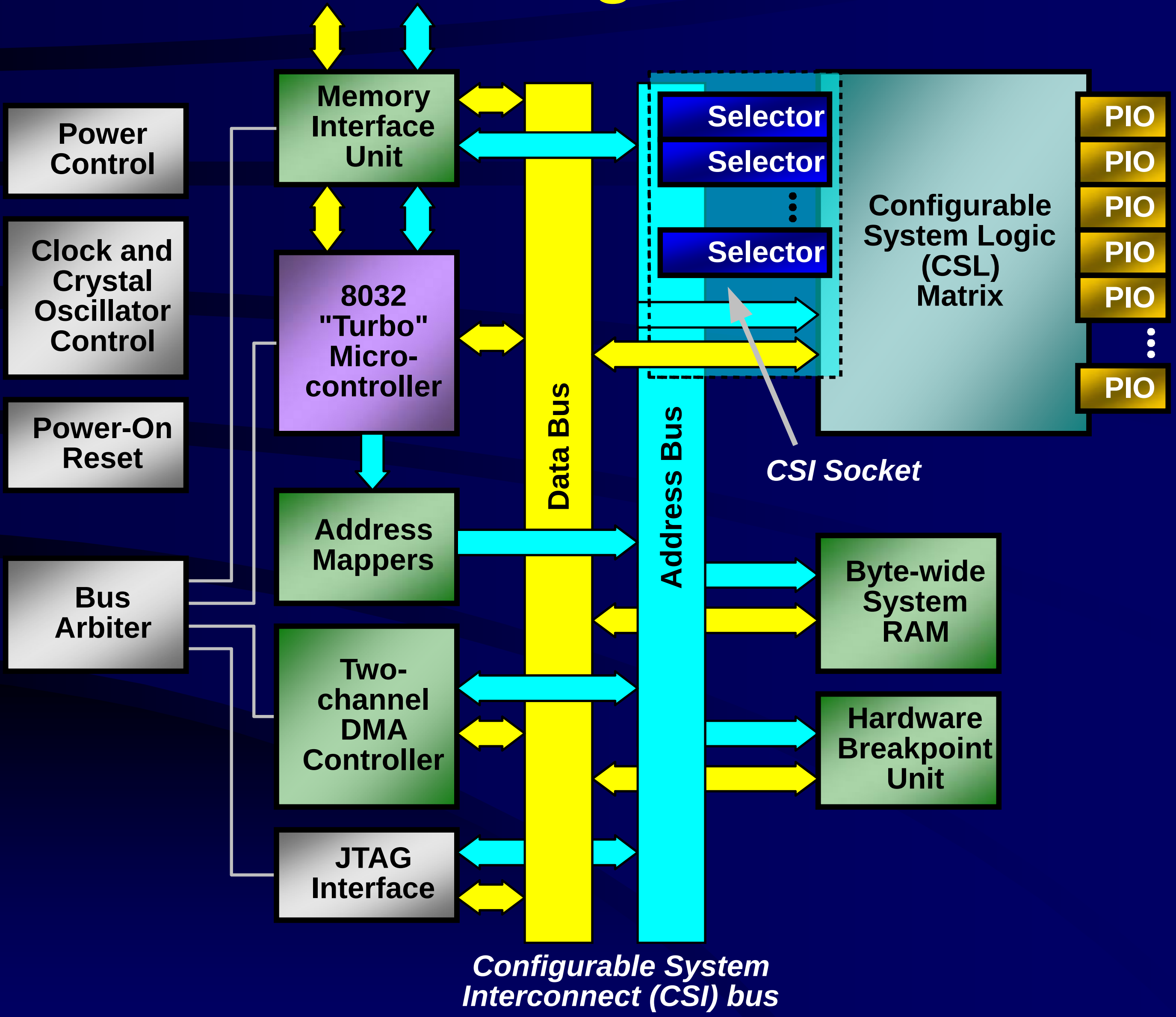
3Kbyte Embedded RAM

External Bus Interface

National NAPA



Triscend E5 Configurable Processor



Configurable Processor Applications

- Custom Peripheral Set
  - Practically any digital function
  - Matched specifically to the application
  - Derivative on demand
- Hardware Acceleration
  - Algorithms in hardware
  - Handling odd-size math
  - Faster real-time response
  - Multiple operations in parallel
  - Bit manipulation

Custom Peripherals (Hardware/Software Trade-Offs)

- **Software Solution** (µs to ms)
  - **Slow peripherals** (serial ports, etc.)
  - **Limited by CPU performance** (Scenix, Teragen)
  - **Easy to modify**
  - **Cheap, re-use existing silicon**
- **Hardware Solution** (ns to µs)
  - **Standard derivative** (no differentiation)
  - **CPU + ASIC/FPGA** (difficult to modify)
  - **Configurable Processor** (easy to modify)
  - **Additional silicon/cost**

Example: SPI Interface

- Find a processor derivative that matches your requirements
  - It has SPI, but does it have everything else you need?
  - Availability? Software support?
- Implement your peripheral in software (ex. Scenix)
- Build your peripheral in an external ASIC or FPGA
- Use a SPI soft peripheral in your configurable processor

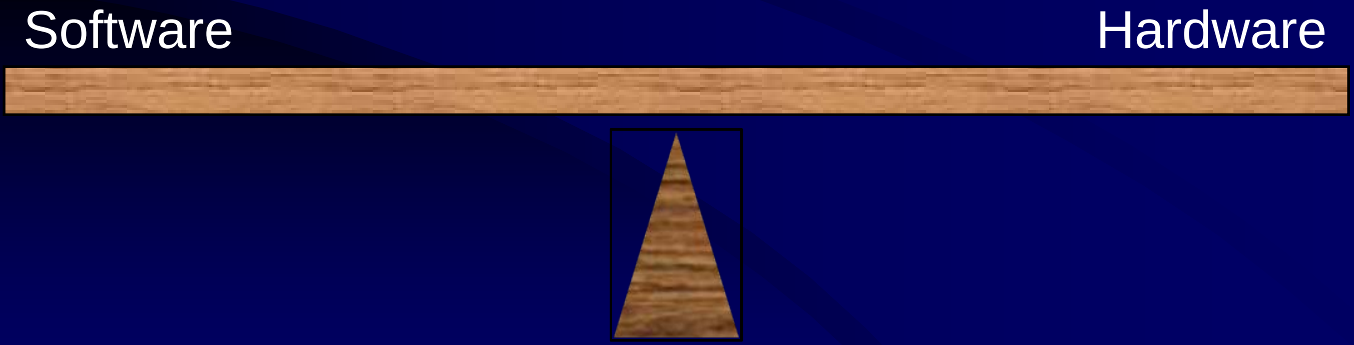
Design Techniques

Peripherals in Software

- ‘C’ language
- Assembly
- Instruction-set simulator
- Function library

Peripherals in Hardware

- Schematic capture
- VHDL/Verilog entry
- Digital logic simulator
- Soft macros available



Hardware Acceleration Example

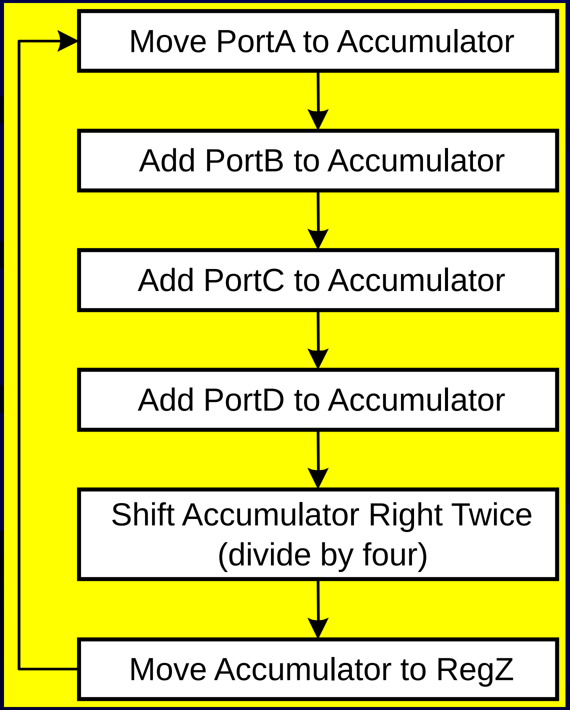
- Calculate the instantaneous average of four 8-bit values

$$Z = \frac{(A+B+C+D)}{4}$$

- Issues
  - Concurrency (I/O, processing requirements)
  - Handling overflow (accumulator width)
  - Performance (processing time)

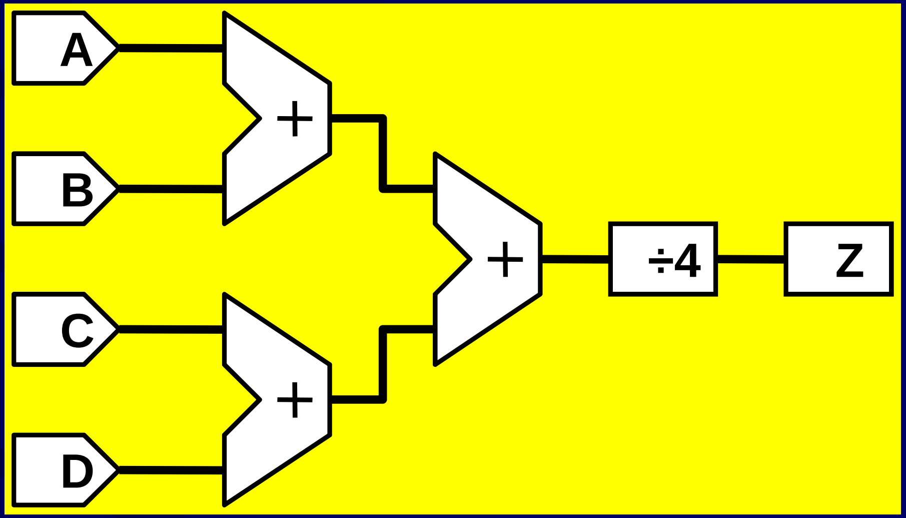
Two Solutions

- **Processor Solution**



More instances require additional time

- **Logic Solution**



More instances require additional logic

Comparing the Alternatives

Solution	Device Cost	Development Time/Cost	Issues	When to Use It
Processor Derivative	$1 - $15	Quick/ Low	Availability, software support, differentiation	Lowest cost, if your application fits
System-on-a-Chip	$5 - $50 + development cost	Long/ High	Acquiring cores, verification, NRE, vendor selection	Volume, complexity, performance justify it.
Fast Processor	$5 - $50	Moderate/ Low	Creating ‘soft’ peripherals	If it fits and it’s fast enough, use it!
CPU + ASIC/FPGA	$10 - $100	Moderate/ Moderate	Multi-chip solution, inter-chip communication, debugging support, multiple CAE tools	For applications where a configurable processor does not yet exist.
Configurable Processor	$8 - $80	Moderate/ Moderate	New technology	Fast time to market, complete embedded system

Configurable Processor Technical Challenges

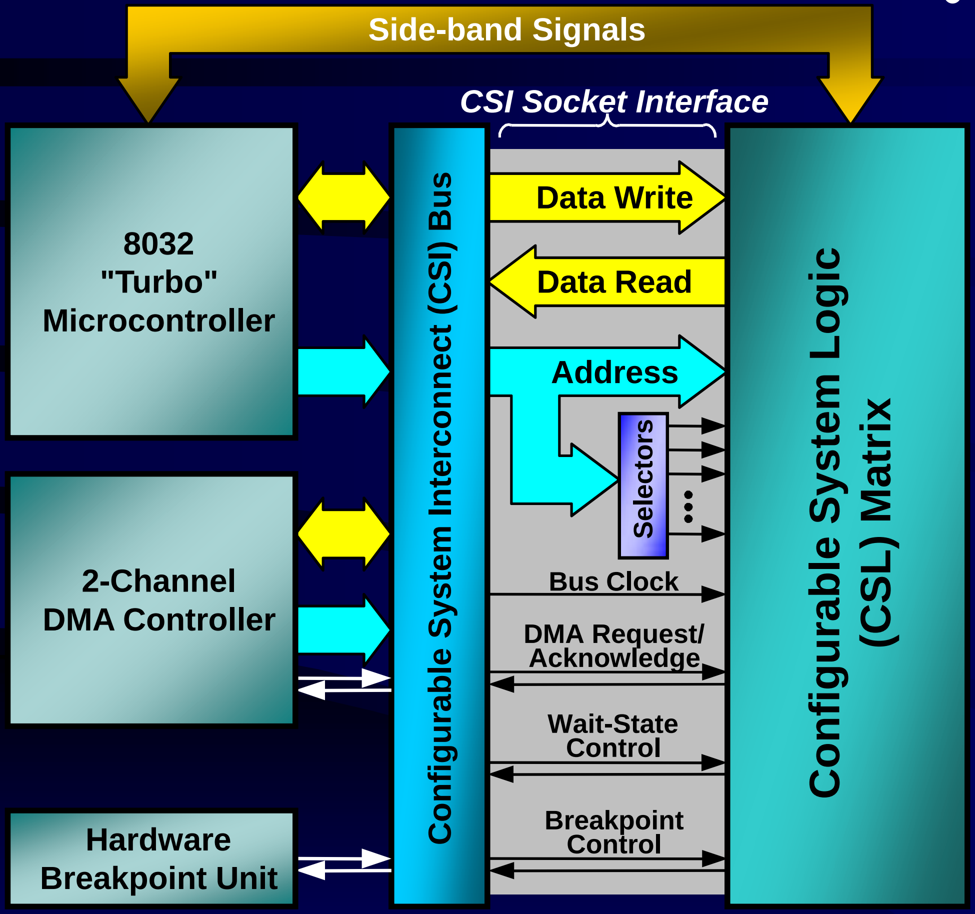
- Communication between the processor and programmable logic functions
- Maintaining a standard development flow
- Debugging a system with both processor and programmable logic

Communication between the Processor and Programmable Logic

- Distributing the data and address bus to the programmable logic
- Decoding/controlling bus transactions
- Multi-master bus support
- Register intimacy
- Debugging support

One Solution: CSI Bus Socket

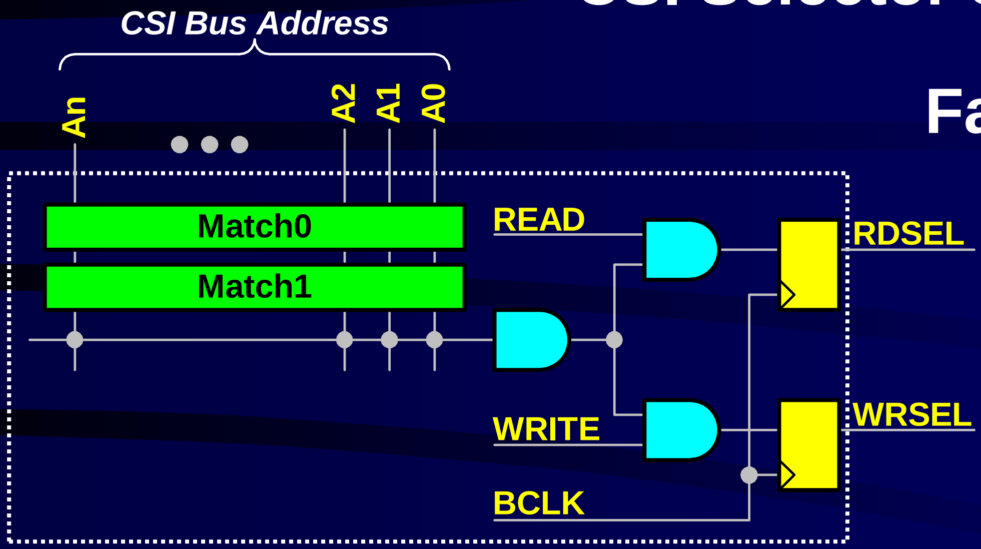
(Configurable System Interconnect)

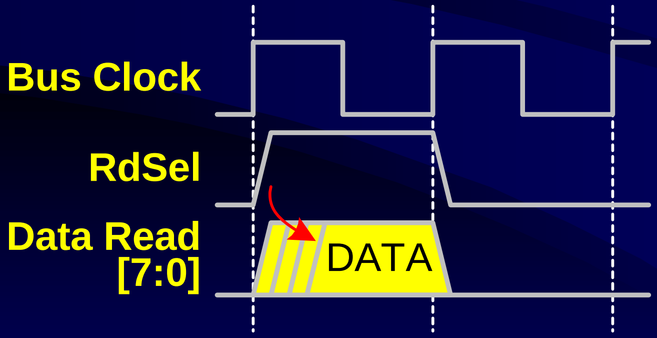


- Distributes address and data to CSL matrix
  - – Full access to data and address bus
  - – Dedicated address decoding
  - – Predictable, synchronous timing
  - – Forward compatible with future configurable processors
  - – Wait-state and breakpoint control
  - – Contention-free bussing

Decoding Bus Transactions

CSI Selector Style





Decode delay is constant
(less than 5 ns after clock)

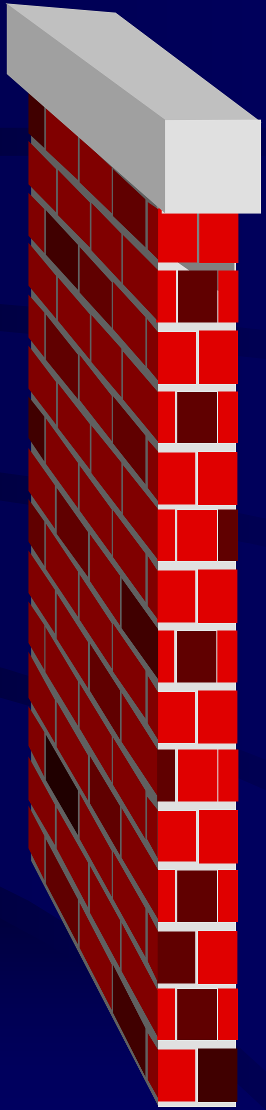
Fast address decoding

- Any address range
- Access type
  - Code
  - Data
  - Exported Special Function Registers (SFR)
- Three modes
  - Selector
  - Chip Select
  - DMA Control Register
- Up to 200 selectors in a single device

Connecting the Two Development Worlds

Hardware Development

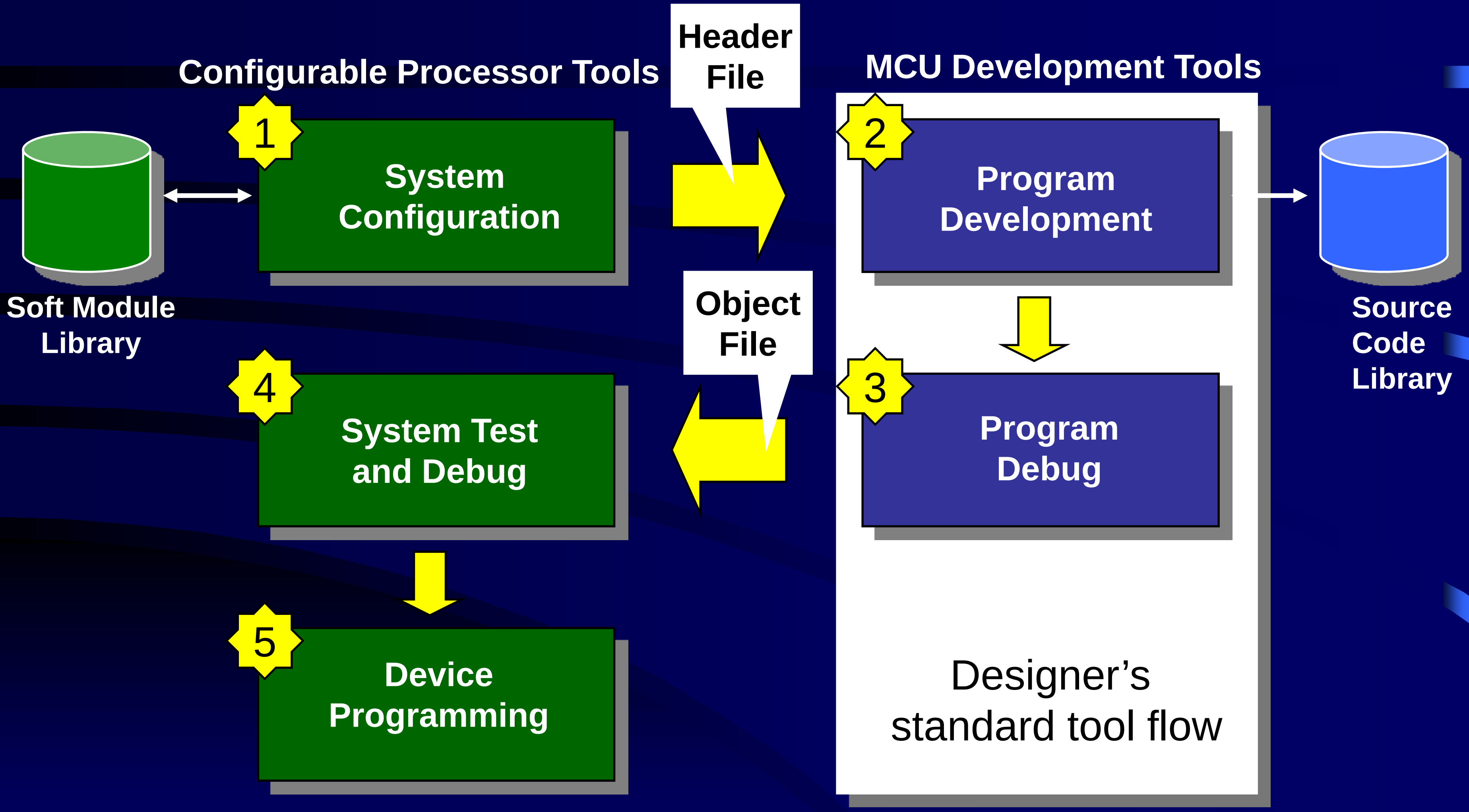
- Design and “soft” module libraries
- Passing register addresses to compiler/assembler
- Vendor place and route software
- Device programming support
- System-wide in-system debugging support



Software Development

- Compiler/assembler support
- Function libraries
- Instruction-set simulator
- System-wide in-system debugging support

Preserving Existing Tool Flow



Configurable Processor Tools

Header File

MCU Development Tools

1 System Configuration

2 Program Development

Soft Module Library

Object File

Source Code Library

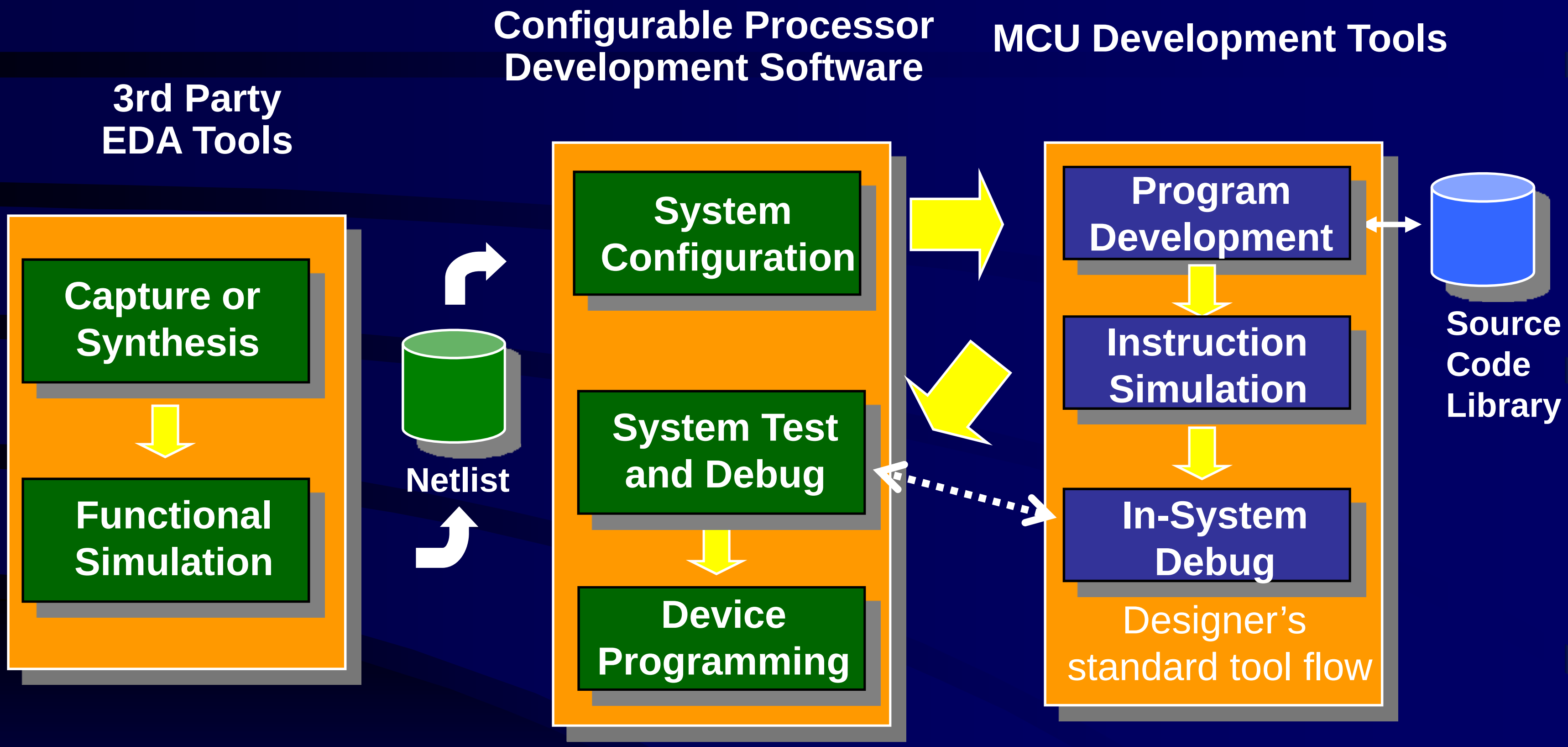
4 System Test and Debug

3 Program Debug

5 Device Programming

Designer’s standard tool flow

System on a Chip Flow



3rd Party EDA Tools

Capture or Synthesis

Functional Simulation

Netlist

Configurable Processor Development Software

System Configuration

System Test and Debug

Device Programming

MCU Development Tools

Program Development

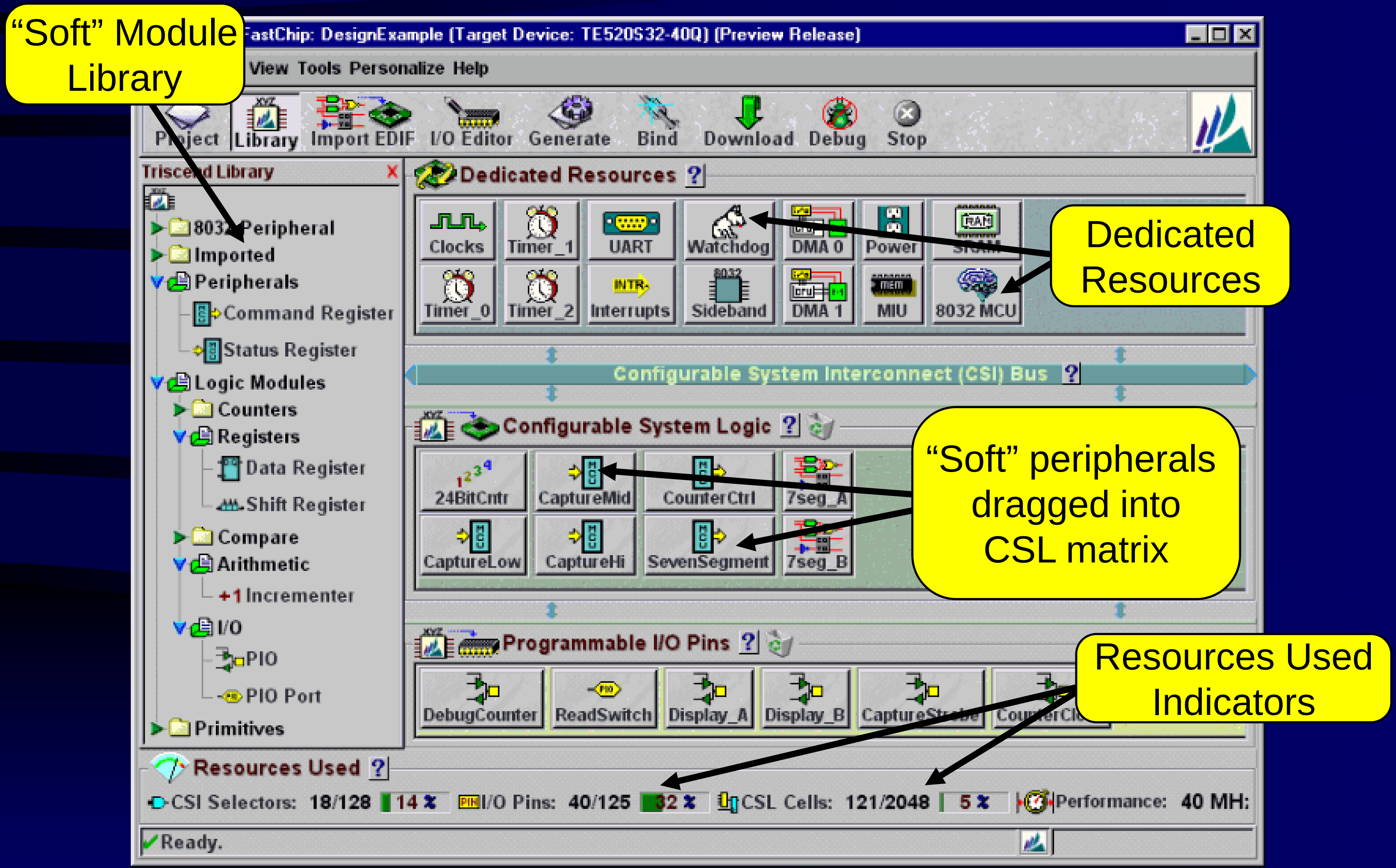
Instruction Simulation

In-System Debug

Designer’s standard tool flow

Source Code Library

Example System: FastChip Software



CSI Selectors: 18/128 14 %
I/O Pins: 40/125 32 %
CSL Cells: 121/2048 5 %
Performance: 40 MH:

"Soft" Module Library

Dedicated Resources

"Soft" peripherals dragged into CSL matrix

Resources Used Indicators

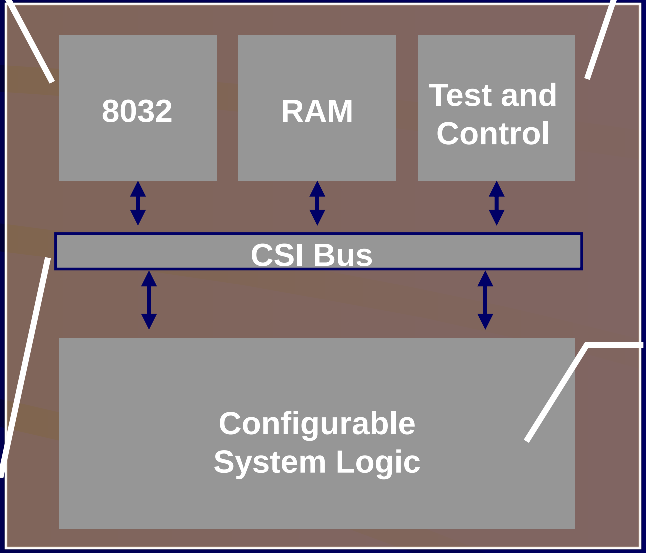
Real-Time, In-System Debugging

- **Difficult in most ASIC or system-on-a-chip designs**
  – Must rely on simulation before completing design
- **Most debuggers only support the processor**
  – Monitor bus activity
  – Monitor processor registers
  – Break on event and single-step
- **Additional debugging desired for programmable logic functions**
  – Monitor the state of logic and flip-flops in “soft” peripherals
  – Monitor or force a breakpoint from programmable logic

Configurable Processor Debugging Capabilities

Access to all address mapped and other key processor resources

Breakpoint unit snoops the internal bus, providing complex runtime control features



Triscend E5

Commands from 3rd party debuggers translated to JTAG instructions

All sequential and combinatorial logic nodes have complete observability

Summary

- Configurable Processors offer benefits in embedded system design:
  - Faster time to market than ASIC/SOC designs
  - Higher performance compared to most processors
  - Higher product differentiation
- Configurable processors are a new class of single-chip programmable devices designed for embedded systems applications
  - Industry-standard processor
  - Dedicated, high-performance internal bus
  - Programmable logic, connected to internal bus
  - On-chip, high-density memory