



The Configurable System-On-Chip Company

E5 Configurable System-on-Chip



Goals of This Section

- Learn the Triscend terminology
- Learn the architecture of the Triscend E5 Configurable System-on-Chip device
- See how the E5 features are integrated within the Triscend FastChip software
- Know applications where the Triscend E5 is a good fit
- Learn about competing technologies

Glossary of Terms

- **CSoC** - Configurable System-on-Chip
- **CSL** - Configurable System Logic
- **CSI** - Configurable System Interconnect
- **MCU** - Microcontroller Unit
- **MIU** - Memory Interface Unit
- **PIO** - Programmable Input/Output
- **DMA** - Direct Memory Access
- **SFR** - Special Function Register
- **JTAG** - Joint Test Action Group, serial programming/debugging interface

Triscend E5 Family

8032 “Turbo” Microcontroller

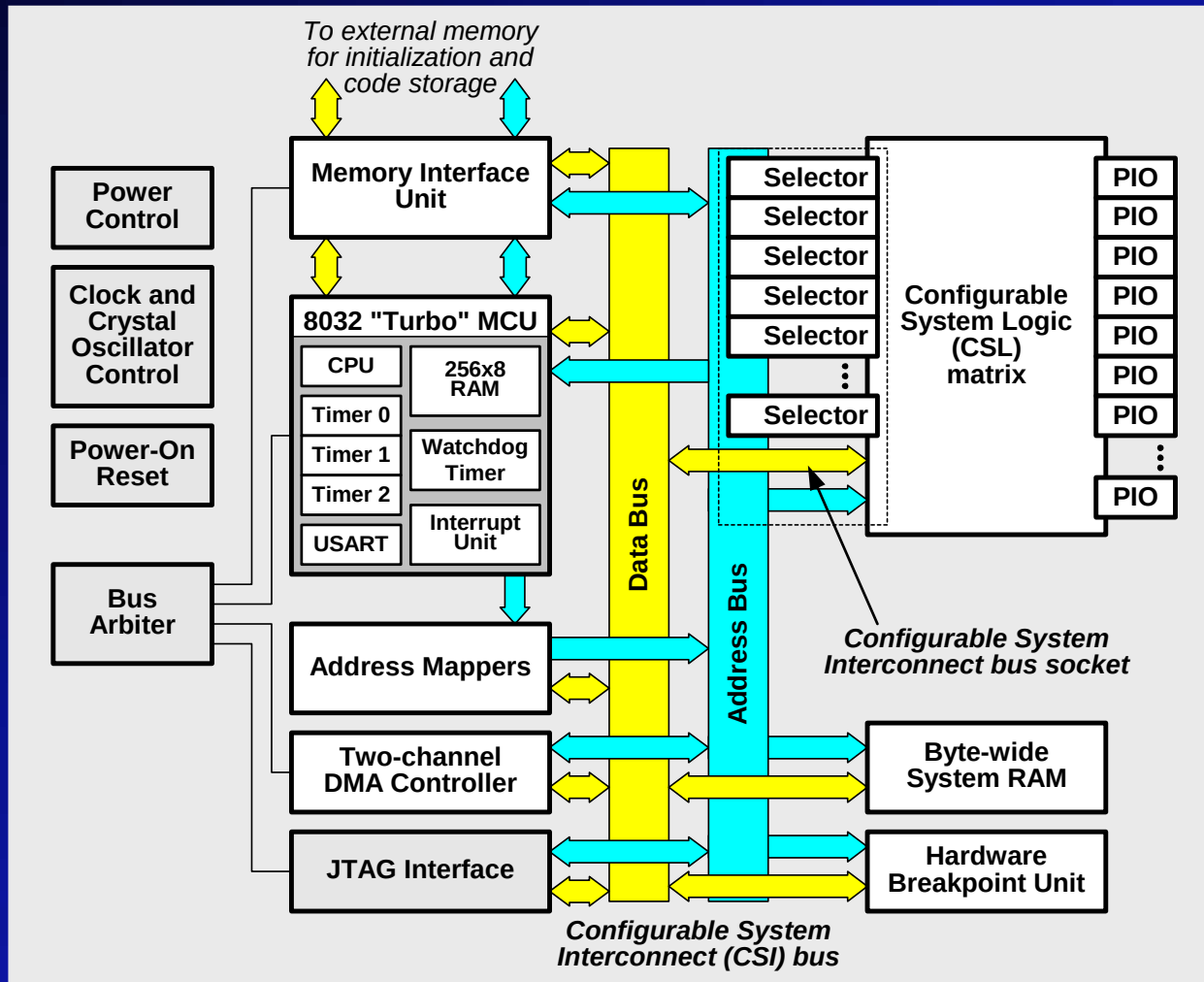
- Three 16-bit timer/counters
- USART
- 26-bit watchdog timer
- Interrupt controller

Dedicated Resources

- High-speed bus, address decoding
- 2-channel DMA controller
- Glue-less memory interface
- Hardware breakpoint unit
- JTAG interface

Device	System RAM	CSL Cells	Address Selectors	PIO Pins (Max)
TE505	16Kx8	512	32	124
TE512	16Kx8	1,152	72	188
TE520	40Kx8	2,048	128	252
TE532	64Kx8	3,200	200	316

Triscend E5 Block Diagram



What is an 8032?

- 8051: ROM-based, 8-bit MCU
- 8031: ROM-less, 8-bit MCU, requires external PROM
- 8052: superset of 8051
 - Additional timer, Timer 2
 - 128 bytes of additional RAM (indirect addressing)
- 8032: ROM-less 8052

Who Makes 8051/52 Compatibles?

- **Intel** (8031, 8032, 8051, 8052, 80151, 80251)
- **Atmel**
- **Philips** (XA)
- **Infineon** (Siemens), C501
- **Dallas Semiconductor** (80C320)
- **ISSI**
- **OKI Data**
- ... and many more

Dallas 80C320

- Superset of 8032
- Only 4 clocks per instruction cycle
 - 8051 requires 12 clocks per instruction
 - Up to 3x faster than original 8051 at the same clock frequency
 - Lower power, one third the clock frequency for a given performance
- Additional peripherals
 - Second UART
 - Watchdog Timer
- Dual data pointer

Comparing 8052-class MCUs

Feature	Average 8052	Dallas 80C320	Philips XA-GA	Triscend E5
8051/8052 binary compatible	Yes	Yes	No	Yes
Max. Frequency	24 MHz	33 MHz	30 MHz	40 MHz
Instruction cycle (clocks)	12	4	3	4
16-bit Timer/Counters	3	3	3	3+
Watchdog Timer	N/A	Yes	Yes	Yes
UARTs	1	2	2	1+
Interrupts	7	13	38	12+
Data Pointers	1	2	-	2
Wait-state support	N/A	N/A	Yes	Yes
PIO pins	32	32	32	60 to 316
On-chip internal RAM	256	256	512	256
On-chip MOVX RAM	N/A	N/A	N/A	16K to 64K
Glue-less memory interface	N/A	N/A	N/A	Yes
DMA channels	N/A	N/A	N/A	2
Maximum address space	64K	64K	1M/16M	64K/16M
On-chip debug hardware	N/A	N/A	Yes	Yes

Triscend E5 Advantages

Over 8052/8032

- **Modern, glue-less interface to external memory**
 - CE-, WE-, OE- signaling
 - Non-multiplexed address/data bus
 - No external latch required
- **Wait-state support**
 - Flexible bus timing to support slower devices
- **DMA Controller**
 - Faster data movement
 - Reduced processor loading
- **On-chip debugger**
 - Real-time in-system debugging
 - Integrated with 3rd party source-code debuggers through JTAG port
- **On-chip MOVX RAM**
 - Local code and data storage
 - Higher system performance
 - Lower power/EMI

Triscend E5 Advantages

Over 8052/8032 (continued)

■ Configurable logic plus on-chip bus

- Custom peripheral set
- Reduced parts count
- Lower power/EMI

■ Programmable address decoder

- Flexible address assignment
- Increased performance

■ Plentiful, modern I/O

- Up to 315 PIO pins
- No shared functionality
- True bi-directional I/O
- Low-power option

■ Extended address support

- Code banking without additional hardware
- Support up to 16M

■ Reduced instruction cycle

- Higher performance
- Reduced power consumption



Target Customers

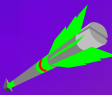


Direct Hits

- Using 8051 or other 8-bit MCU
- Needs additional processing horsepower
- Uses MCU plus additional logic (FPGA or CPLD)
- Wants reduced board space

Big Misses

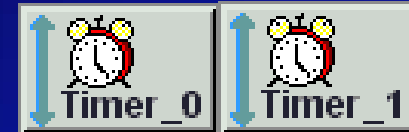
- Only needs a basic 8051 or widely-available derivative
- Has no external logic and has no logic design experience 
- Needs on-chip Flash or analog



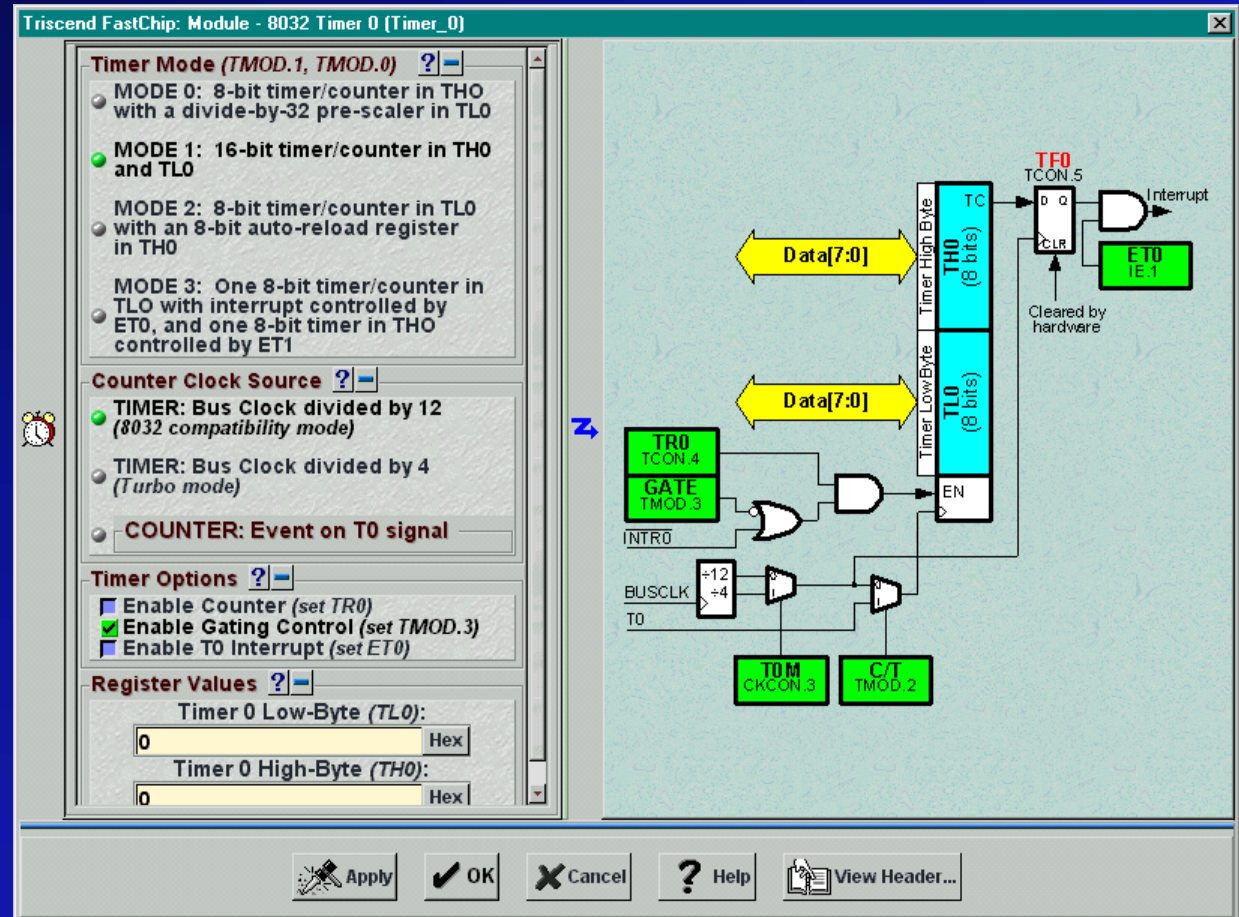
8032 “Turbo” MCU Peripherals

- Three 16-bit Timer/Counters
- UART (additional UARTs as “soft” modules)
- Interrupt Controller
 - 12 interrupt sources (expandable)
 - 3 priority levels
- 26-bit Watchdog Timer
- Flexible Clocking
 - From an external source
 - Internal crystal oscillator amplifier
 - Internal ring oscillator
- Power-on Reset and Power Management

Timers 0/1



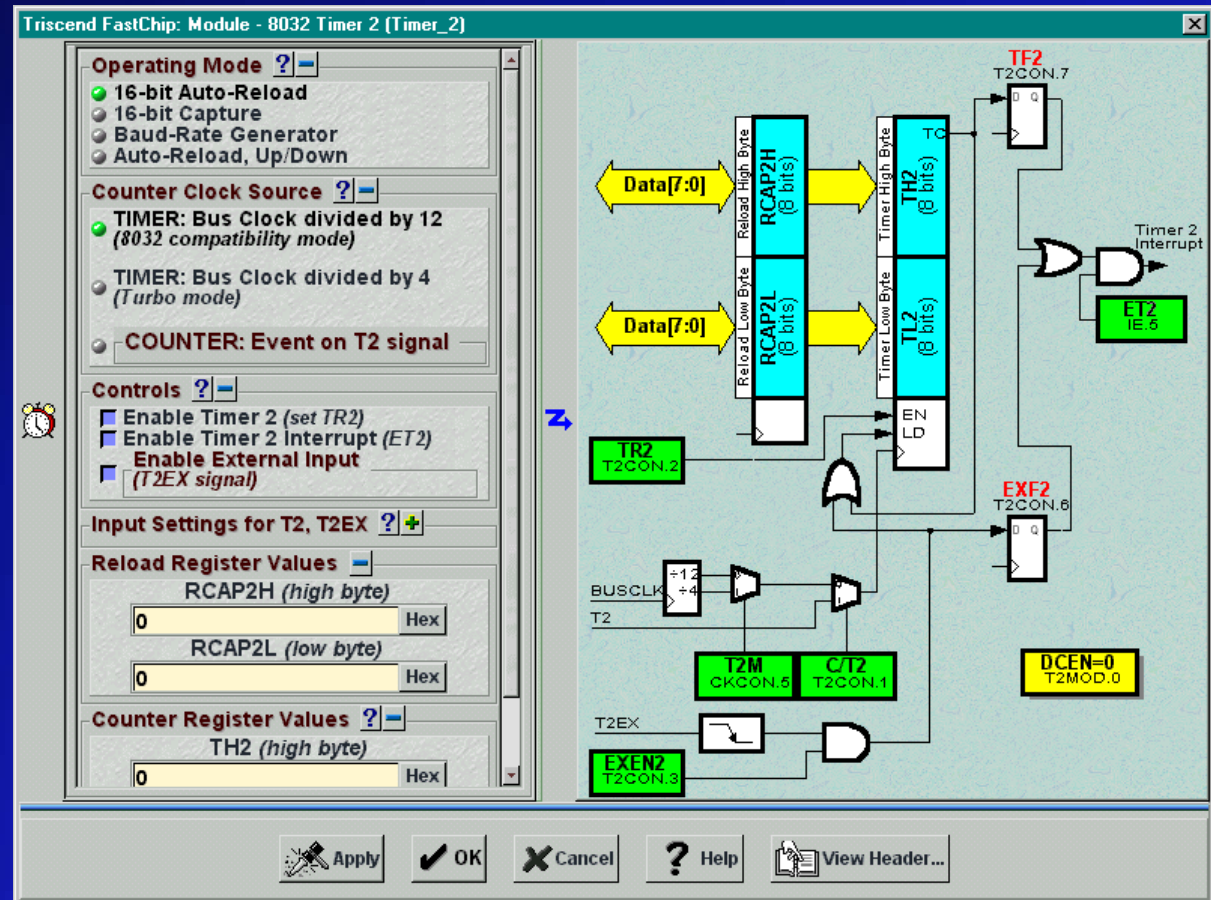
- Four operating modes
 - 13-bit
 - 16-bit
 - 8-bit auto-reload
 - Two 8-bit counters
- Three clock sources
 - 8032 mode
 - Turbo mode
 - External signal via T0 or T1
- Timer 1 also used as baud-rate generator
- Processor interrupt



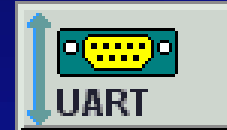
Timer 2



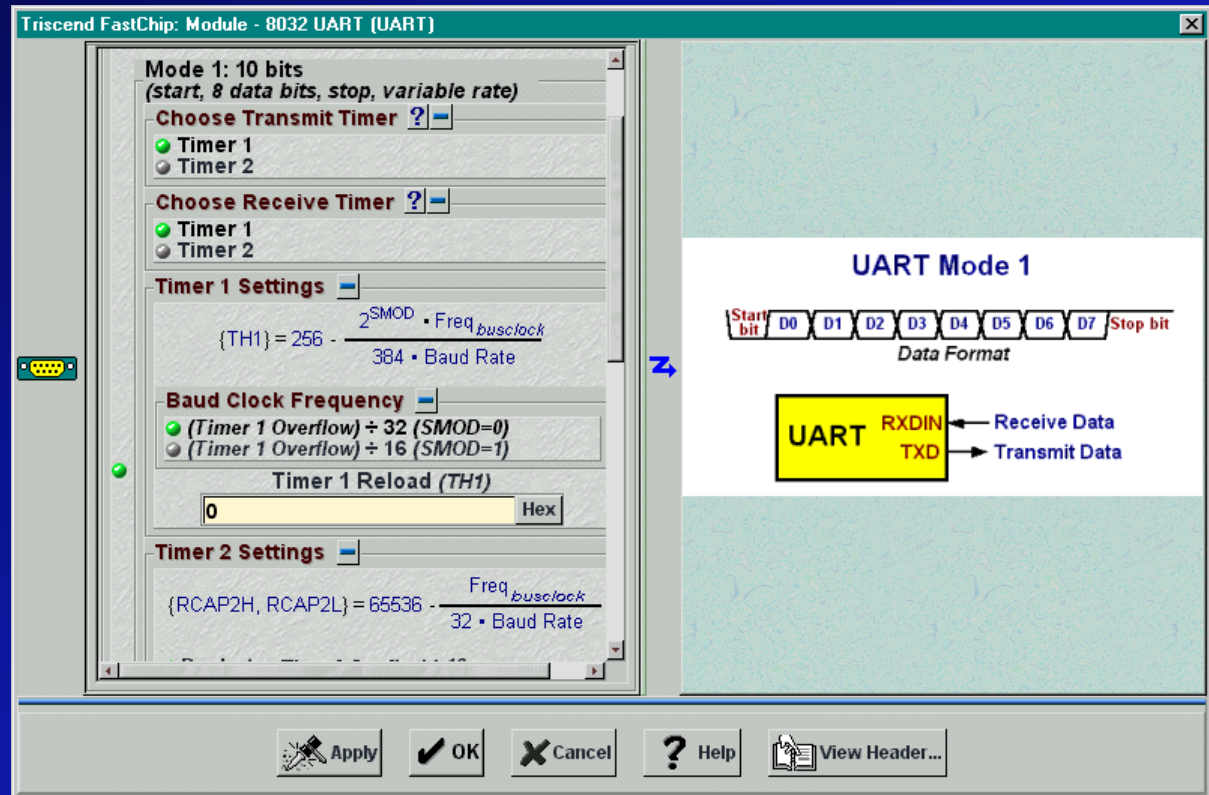
- 16-bit auto-reload or capture counter
- Baud-rate generator for UART
- Three clock sources
 - 8032 mode
 - Turbo mode
 - External signal via T2
- Additional input via T2EX



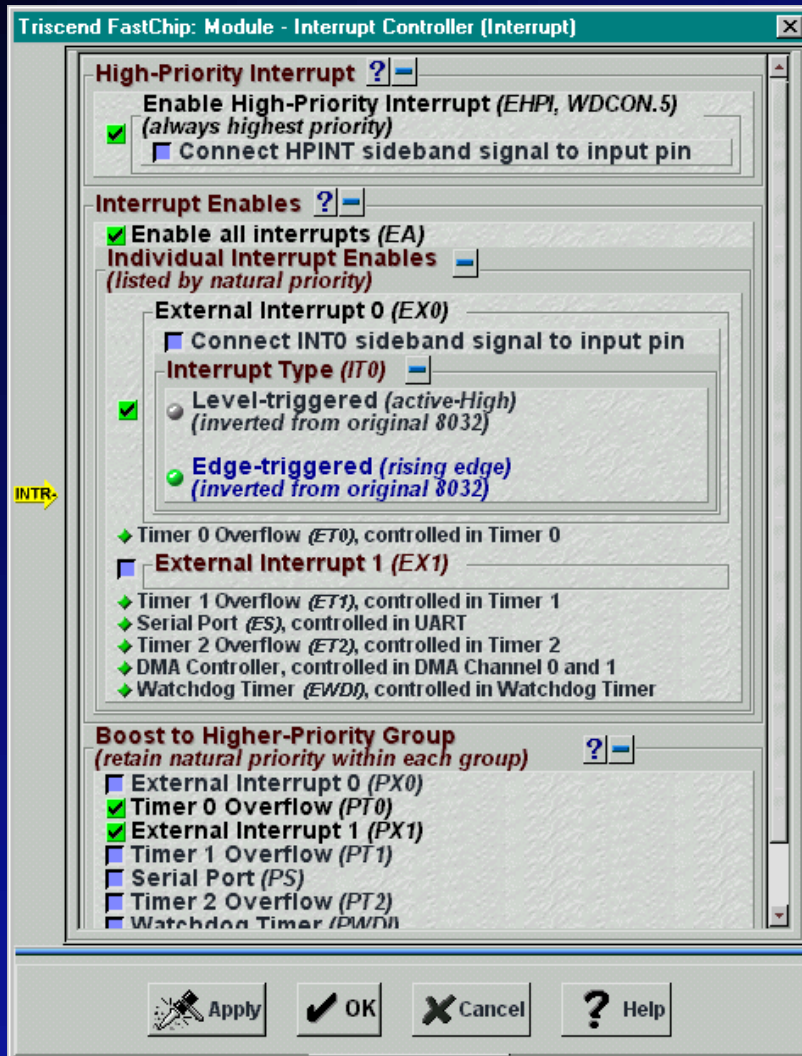
USART



- Four operating modes
 - Synchronous
 - Asynchronous
 - Fixed or variable baud rate
- Timer 1 and Timer 2 used as baud-rate generators
- Independent transmit and receive baud rates
- Synchronous mode (Mode 0) uses an additional pin (RXDOUT)
- Supports 8032-style multiprocessor communication



Interrupt Controller

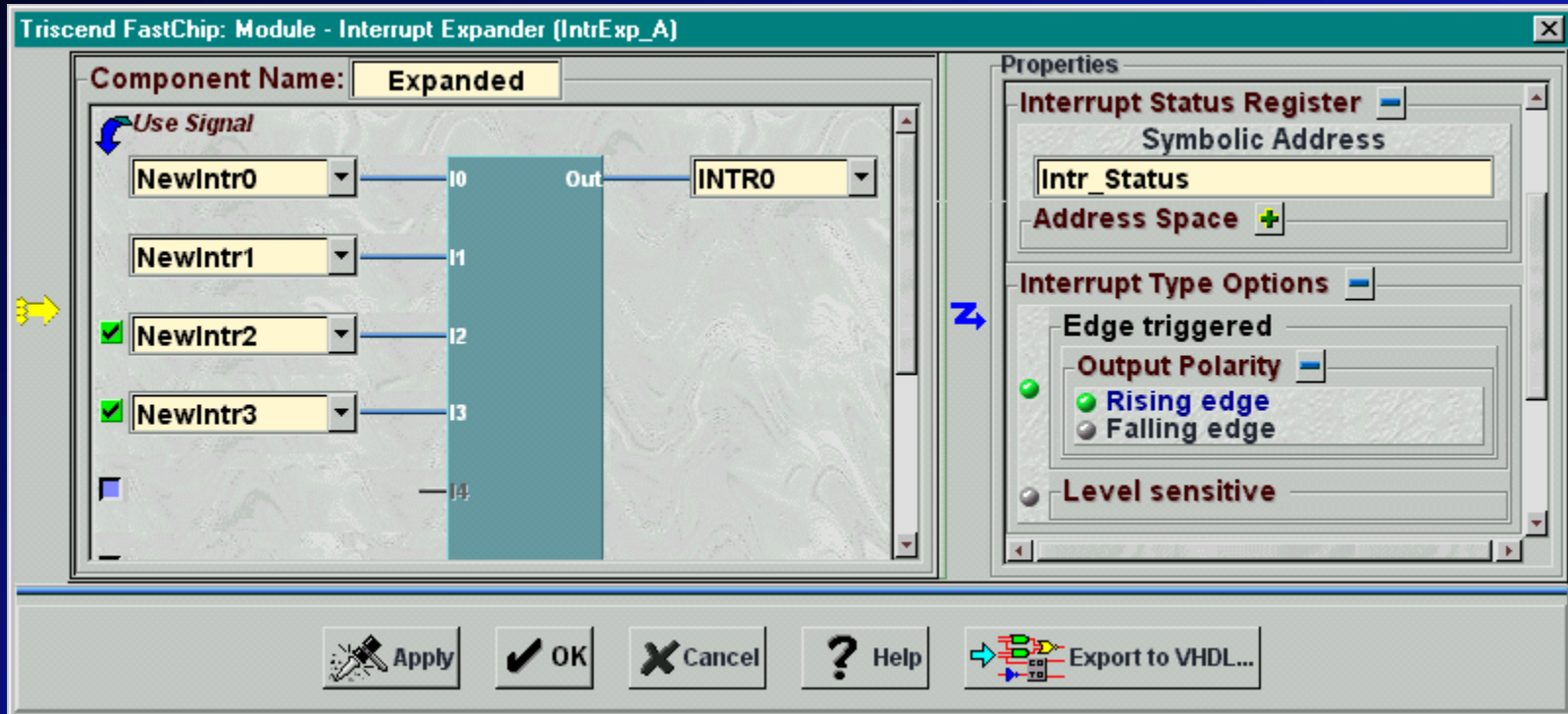


- 12 interrupt sources
- Three priority levels
 - High-priority interrupt
 - Boosted priority
 - Normal priority
- Expandable using “soft” module
- EA (Enable All) enables or disables all interrupts except HPINT
- Each enable has an independent enable control
- Some interrupts can be boosted to a higher priority group
 - Members maintain relative priority within the group

E5 Interrupt Structure

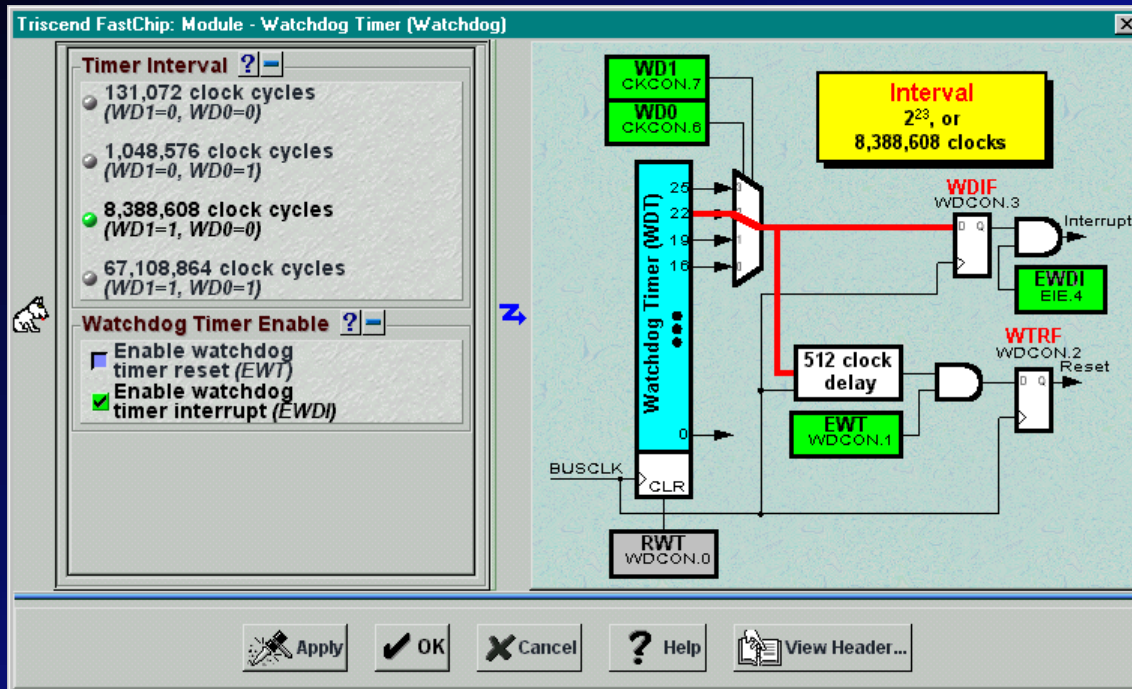
Source	Flag	Priority	Vector	Keil Interrupt #
High-Priority	HPI	1 (highest)	0x33	6
External Interrupt 0	IE0	2	0x03	0
Timer 0 Overflow	TF0	3	0x0B	1
External Interrupt 1	IE1	4	0x13	2
Timer 1 Overflow	TF1	5	0x1B	3
Serial Port	RI TI	6	0x23	4
Timer 2 Overflow	TF2 EXF2	7	0x2B	5
DMA	OVR0 INIT0 TC0 OVR1 INIT1 TC1	8	0x3B	7
Hardware Breakpoint	BP0I BP1I	9	0x43	8
JTAG	JTAGI	10	0x4B	9
Software Breakpoint (A5 instruction)	A5I	11	0x53	10
Watchdog Timer	WDIF	12	0x63	12

Expanding Interrupts



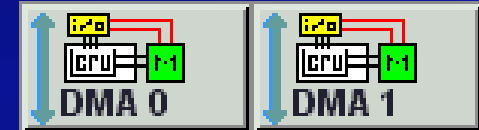
- Interrupt Expander “soft” module allows you to expand the number of interrupts to the 8032
- Does not create a new interrupt vector
- Vectoring handled by interrupt service routing
- Expand HPINT, INTR0, INTR1, T2EX

Watchdog Timer (WDT)

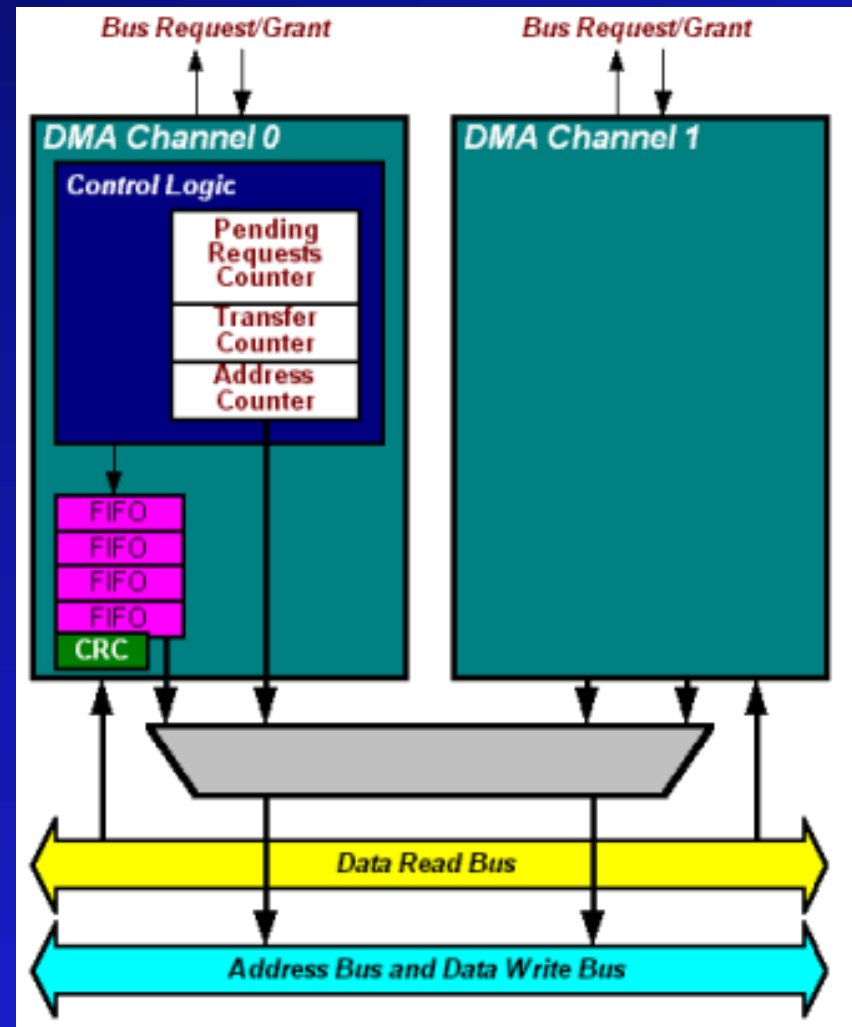


- Can reset errant program
- Useful in noisy or industrial environments
- WDT resets the MCU unless MCU resets the WDT first
- WDT control bits are protected against an errant program
 - Write special code to Timed Access (TA) register
 - Write 0xAA followed by 0x55
 - Opens a two machine cycle window

DMA Controller



- Two independent channels for device/memory transfers
- Transfer rates up to 40 Mbytes/sec
- “Proxy” master for CSL-based functions



Why a DMA?

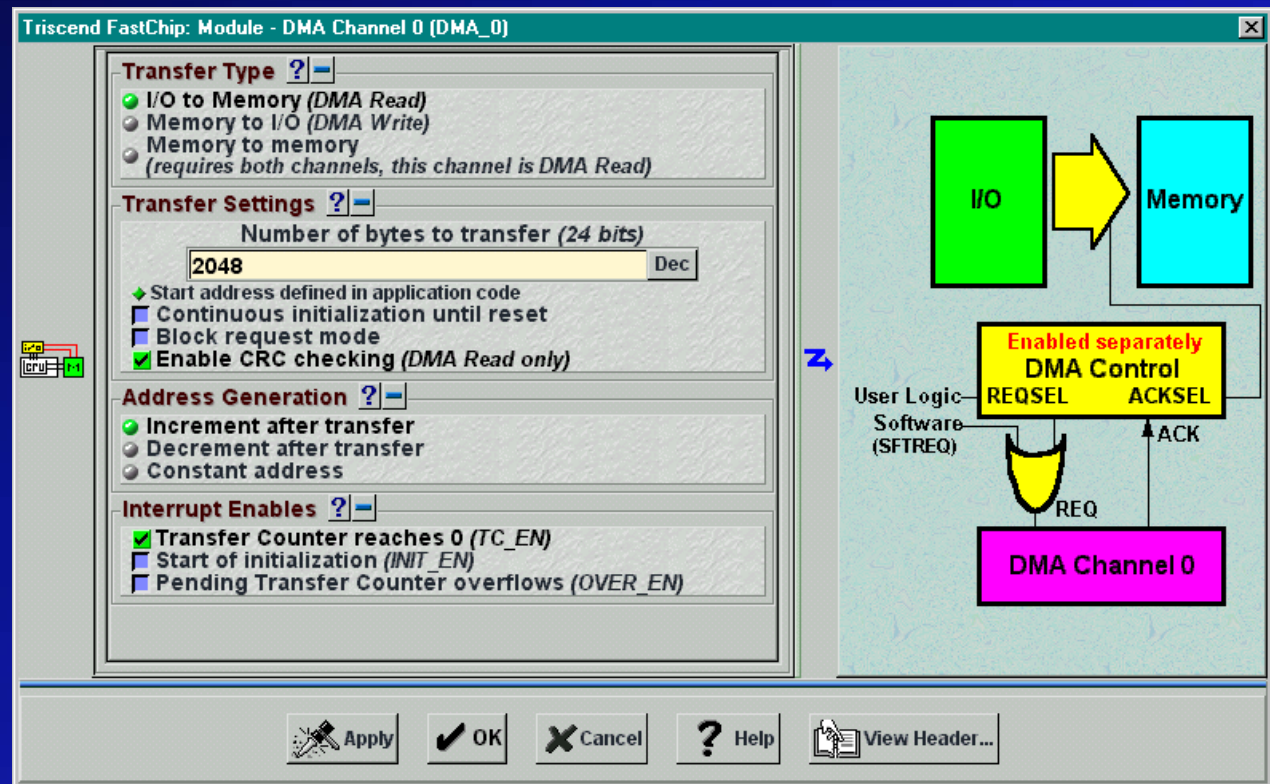
- Faster data movement
 - MOV requires at 1 instruction cycle or 4 clocks
 - MOVX requires at least 2 instruction cycles or 8 clocks plus DPTR overhead
 - DMA moves data in a single clock (typically more than 8 times faster than processor)
- Offloads mundane data movement operations from the processor
 - Processor set up DMA transfer parameters
 - DMA moves data without processor interaction
 - DMA interrupts processor when transfer is complete
 - Processor freed to perform other tasks

DMA Controller Features

- **Various transfer types**
 - Memory to CSL
 - CSL to Memory
 - Memory to Memory (requires both channels)
- **Channel auto-initialization**
- **Multiple address modes**
 - Increment
 - Decrement
 - Constant
- **Block transfers**
- **Tracks pending requests**
- **Interrupt capabilities**
 - Upon initialization
 - At end of transfer
 - If too many pending requests (> 64K)
- **CRC on read operations**
- **Software-initiated DMA requests**

DMA Channels in FastChip

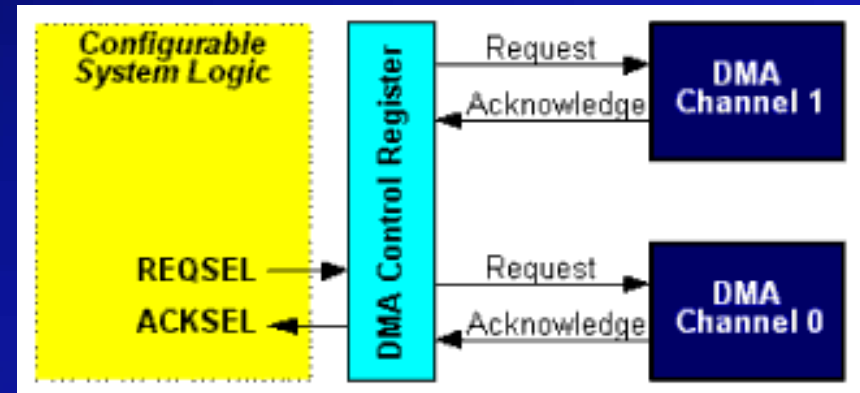
- Transfer Type
- Transfer Count
- Addressing Mode
- Interrupt Conditions



- Start Address
 - Defined in application code
 - Requires a physical 32-bit address, not the 8032's logical 16-bit address
 - Library routine to convert addresses

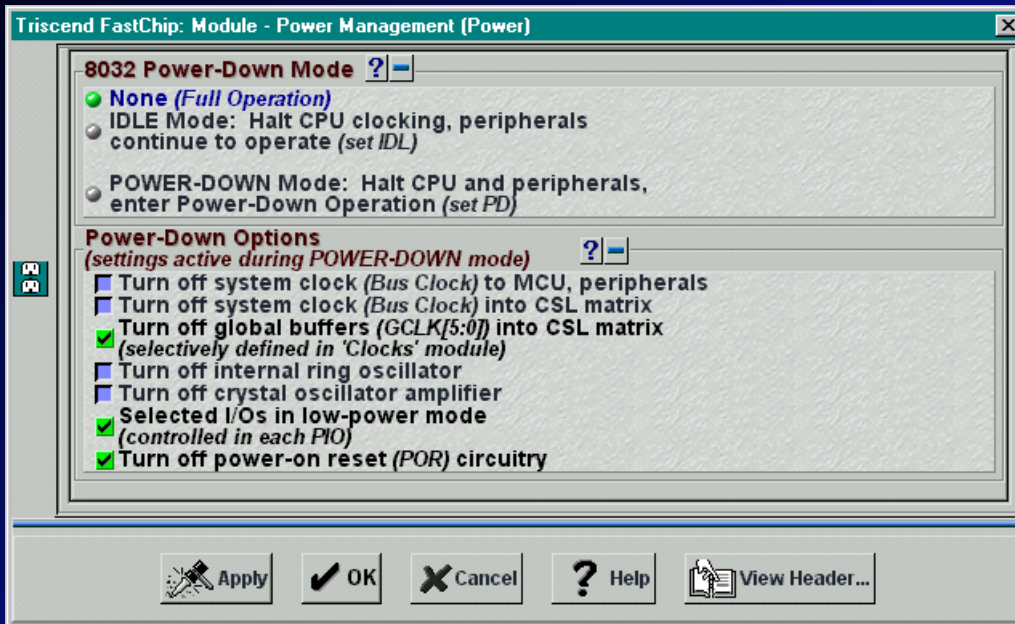
DMA Control Registers

- Distributed DMA request and acknowledge structure to and from the CSL matrix
- DMA Control registers steer request and acknowledge signals to proper DMA channel



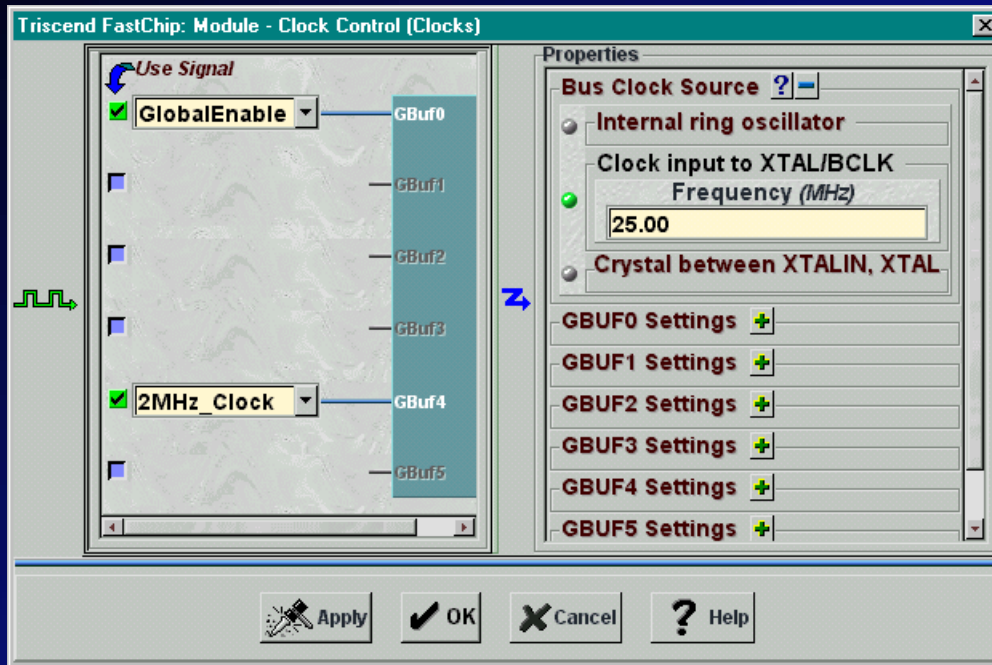
- More later ...

Power Management



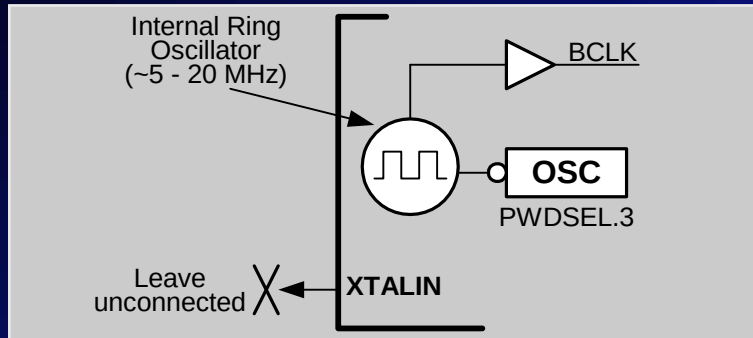
- Three power-down options
 - None
 - Idle: Halt the CPU, remainder clocks normally
 - Power-Down: Halt CPU and peripherals, shut down selected resources
- Power-Down Options
 - Shut down various resources when entering Power-Down mode
 - Turn off clocks
 - Shut off power-on-reset circuitry
 - Careful when turning off the crystal oscillator amplifier
 - Selectively put PIO pins into power-down mode

Clock Management



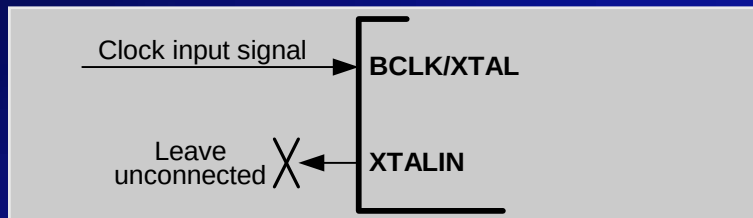
- Single global Bus Clock signal
- Three clock sources for Bus Clock
 - Internal ring oscillator
 - External oscillator connected to XTAL/BCLK pin
 - External crystal connected between XTALIN/XTAL pins
 - Specify the frequency for MIU settings
- Six global buffer signals
 - Drive any high-fanout signal, including clocks or logic
 - Each has a preferred input pin
 - Each can be driven by a PIO or internal logic
 - Each has independent controls for Breakpoint operation

Bus Clock Sources



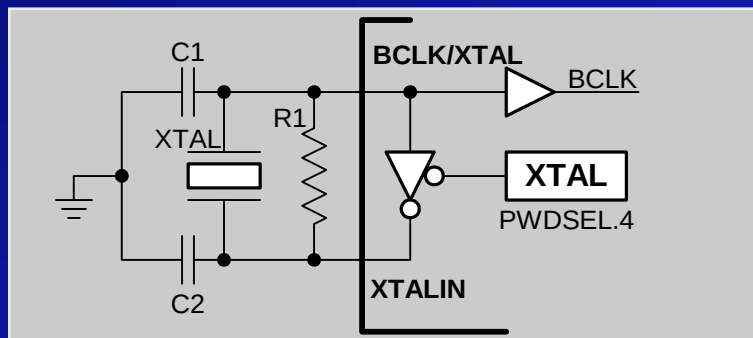
■ Internal Ring Oscillator

- Frequency varies with temperature and voltage
- Frequency varies between about 5 MHz and 20 MHz



■ External Oscillator Input

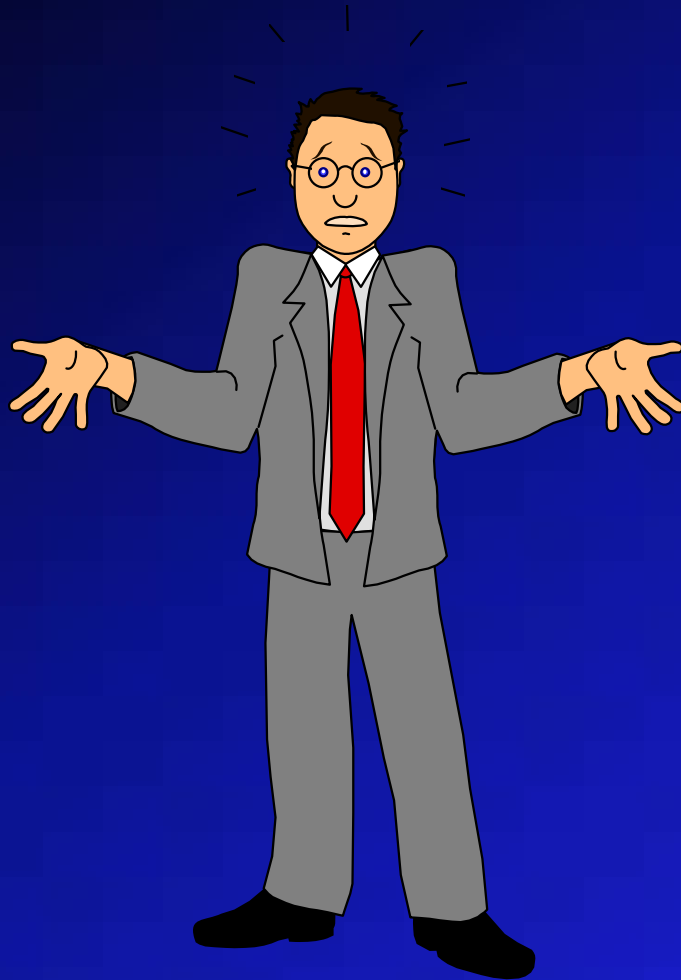
- Clock input connects directly to BCLK/XTAL input pin
- Leave XTALIN unconnected



■ Internal Crystal Oscillator Amplifier

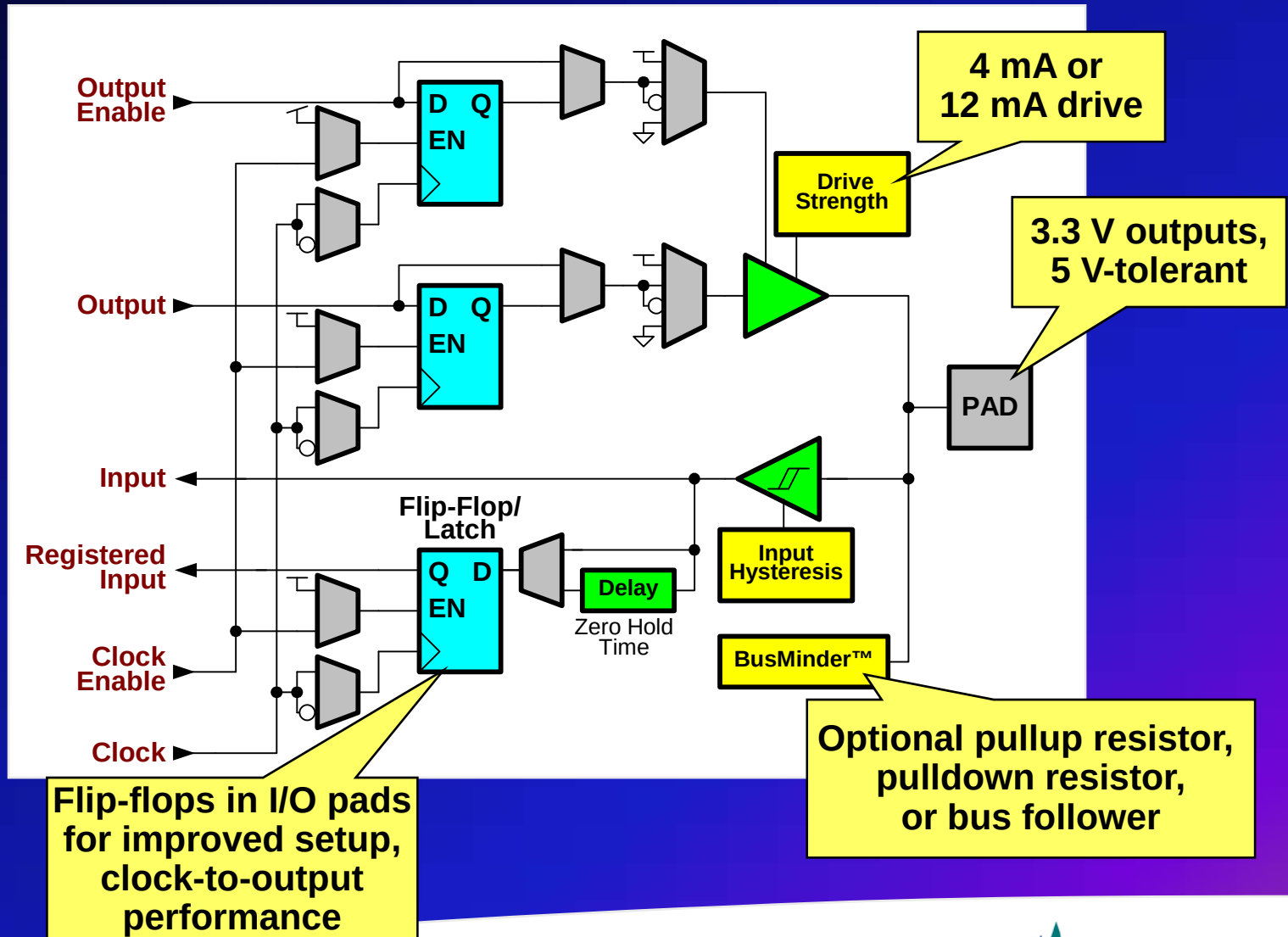
- Connect external crystal or ceramic resonator between XTAL and XTALIN
- Requires external compensation circuitry

Hey! What About PIO Ports?



- Where are P0, P1, P2, and P3?
- The Triscend E5 CSoC does not have any *dedicated* PIO ports
- The E5 offers significantly better capabilities

Programmable I/O (PIO)

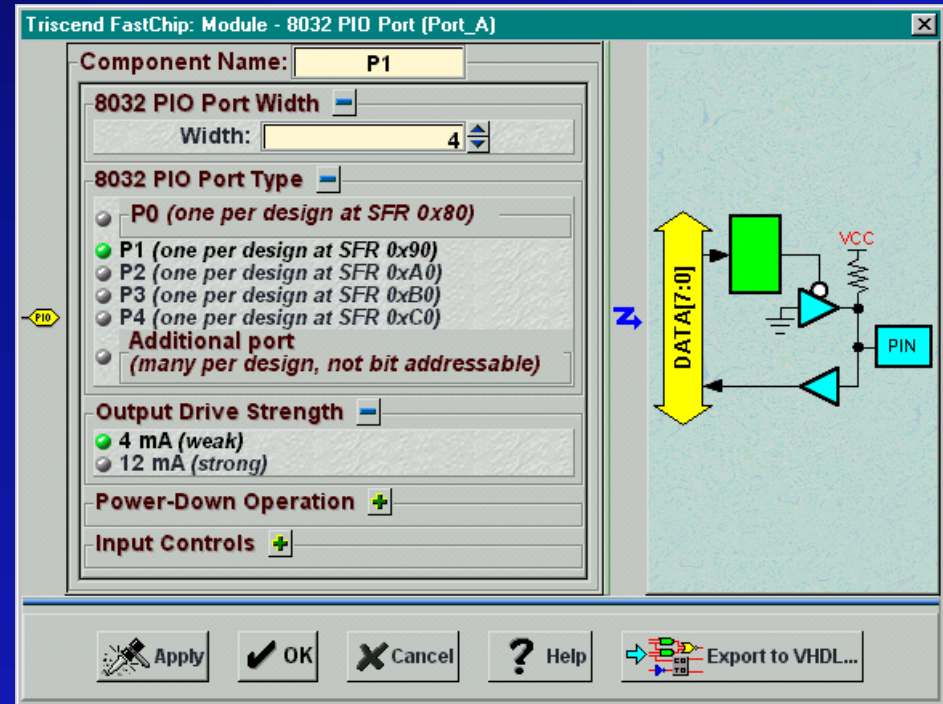


More on PIOs

- PIOs connect to practically anything inside the E5 (processor, bus, peripherals)
- Significantly more capable than original 8032
 - More I/O, up to 316 on largest device
 - No shared I/O pins (INTR0 does not need to share with P3.2)
 - Actively drive a logic High (not possible on 8032 PIOs)
- 3.3 Volt Operation
 - Drive any 3.3 volt device and any 5 volt device with TTL inputs
 - Tolerates 5 volt inputs
 - Lower switching power than 5 volt devices

What if I want 8032-compatible PIO ports?

- No problem!
- Use the 8032-style PIO port “soft” module
- Create up to five bit-addressable PIO ports
 - P0, including pullup resistor, connects to external memory in most applications
 - P1
 - P2, not available on most 8032s
 - P3
 - P4
- Add as many additional PIO ports as you have available package pins
 - Not bit addressable
- Don't limit yourself to 8032-style PIOs



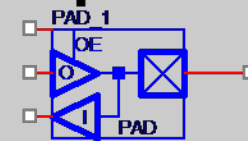
PIO Primitives



Three-State Output

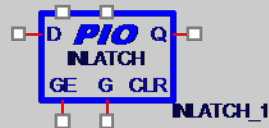


Bi-directional Output

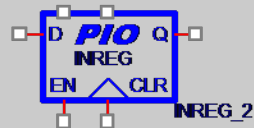


Input Latch

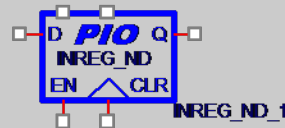
Clear



Input Register



Input Register



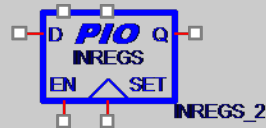
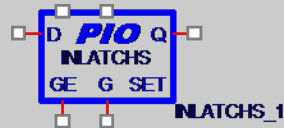
Output Register



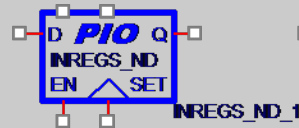
Out Enable Register



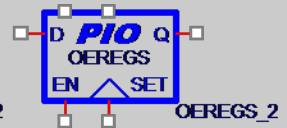
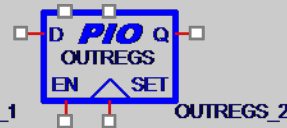
Set



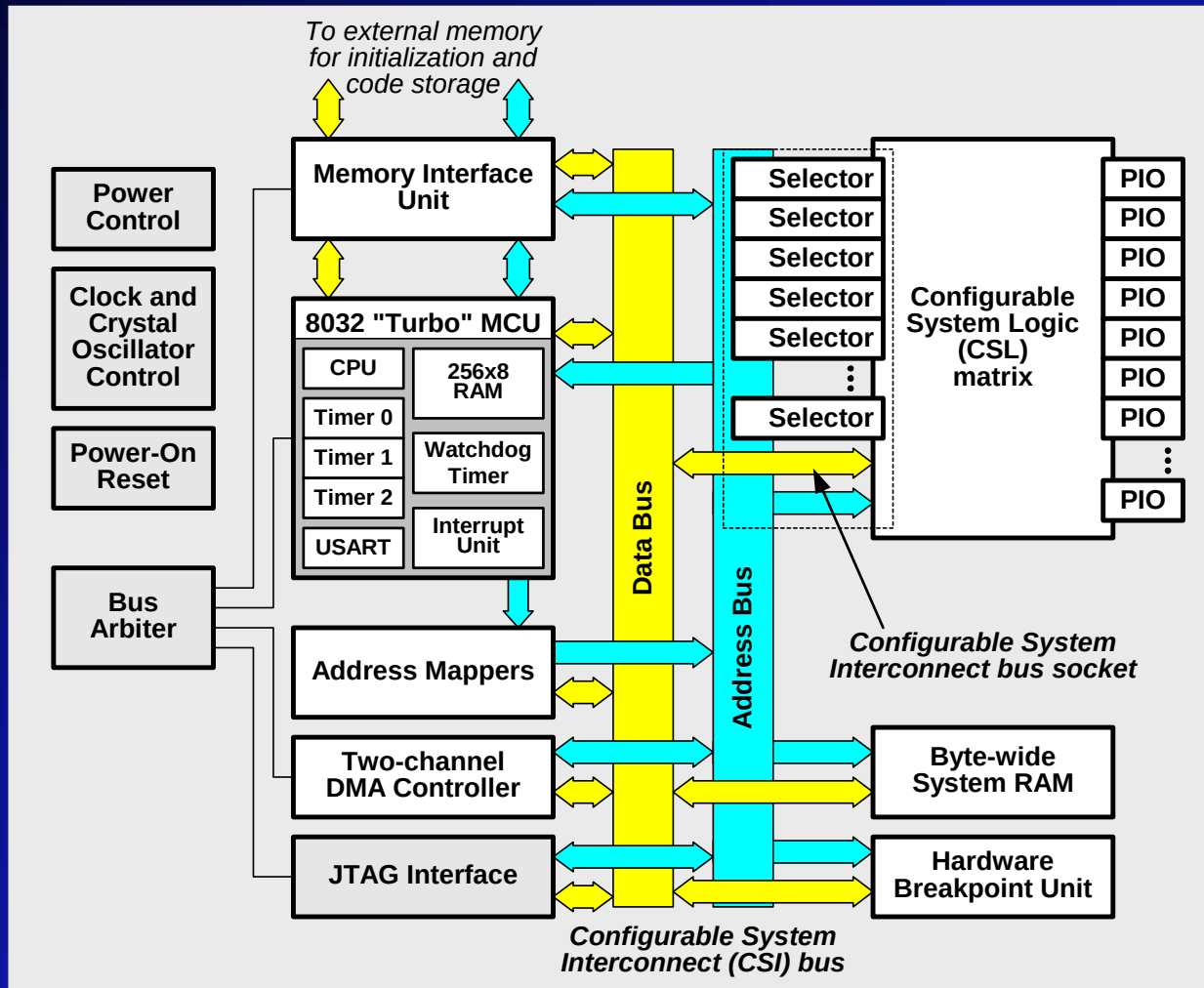
(zero hold)



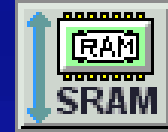
(non-zero hold)



Triscend E5 Block Diagram



Internal System RAM



- Large block of on-chip, byte-wide, single-cycle SRAM

- Most data transfers remain on-chip
- Reduced power consumption/EMI

- Internal variable storage (XDATA)

- Internal code storage

- Faster execution
- Hidden code
- Supports various initialization modes

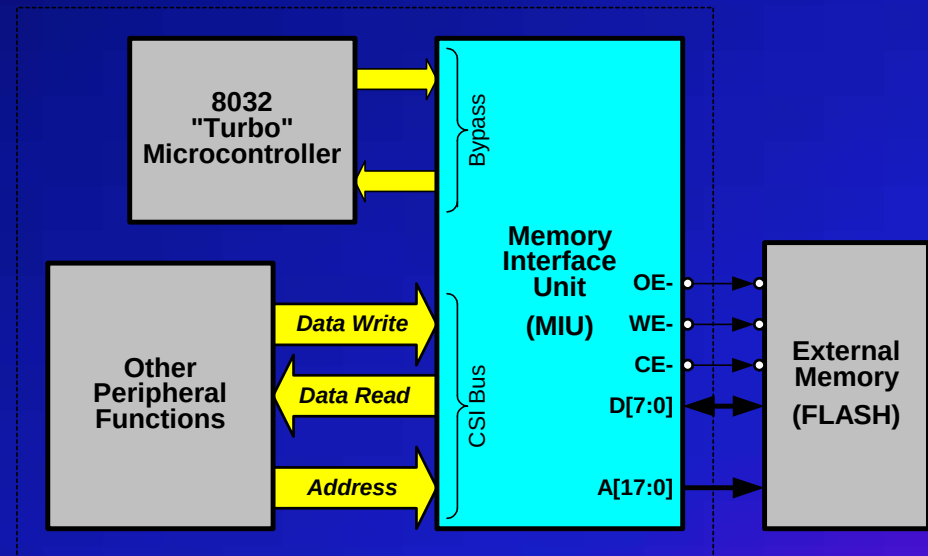
Device	SRAM Size
TE505	16K x 8
TE512	16K x 8
TE520	40K x 8
TE532	64K x 8

- Each Configurable System Logic (CSL) cell optionally provides:
 - 16 bits of single-port RAM, or
 - 8 bits of dual-port RAM with built-in collision detection

Memory Interface Unit

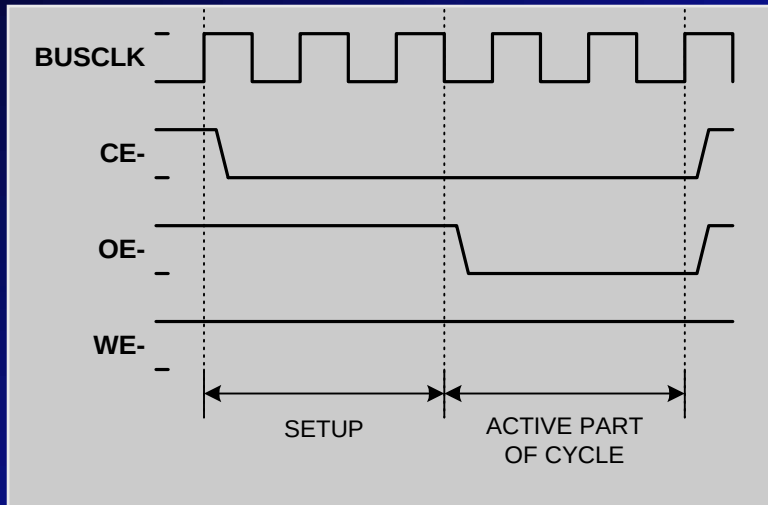


- Interface to 256Kx8 External Memory
 - Flash, SRAM, etc.
 - Holds initialization data and application code
- Supports serial PROMs
 - Unused address, data pins available as user I/O
- Interface expandable to 32 bits



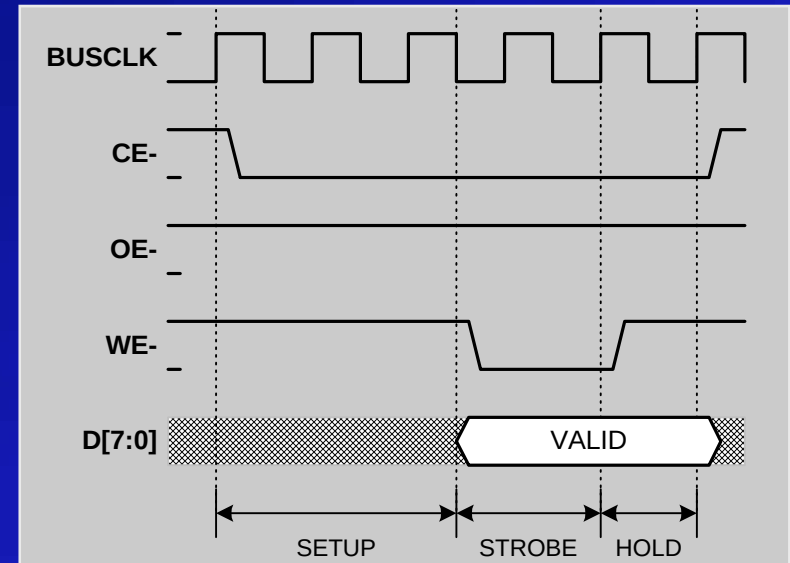
Flexible External Timing

Read Timing



- Setup: 0.5 to 7.5 clocks
- Active: 0.5 to 7.5 clocks

Write Timing



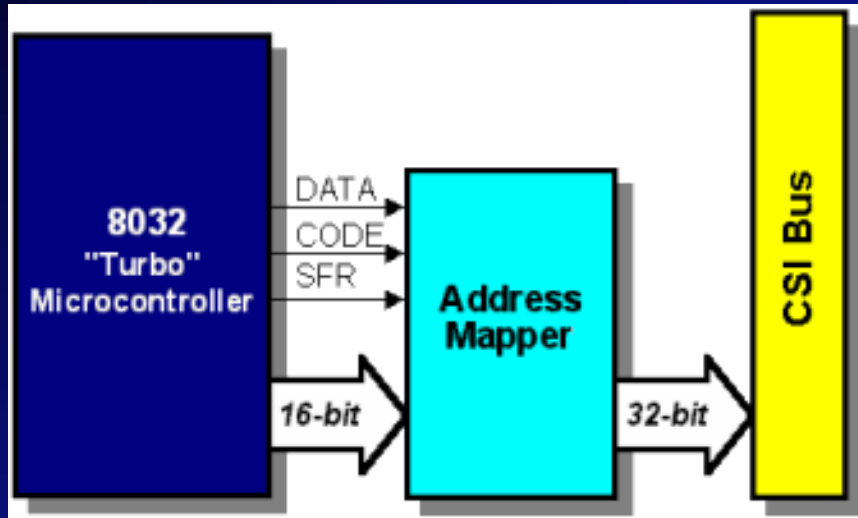
- Setup: 0.5 to 7.5 clocks
- Active: 0.5 to 7.5 clocks
- Hold: 0 to 7 clocks

Setting the MIU Timing

- MIU defaults to slowest settings at power-up
- Initialization program configures correct MIU values
- FastChip sets proper values based on the application
 - Clock source and maximum frequency
 - Selected external memory type and speed grade
 - Maximum number of address lines

Address Mappers

Translates the 16-bit microcontroller address to the 32-bit system address



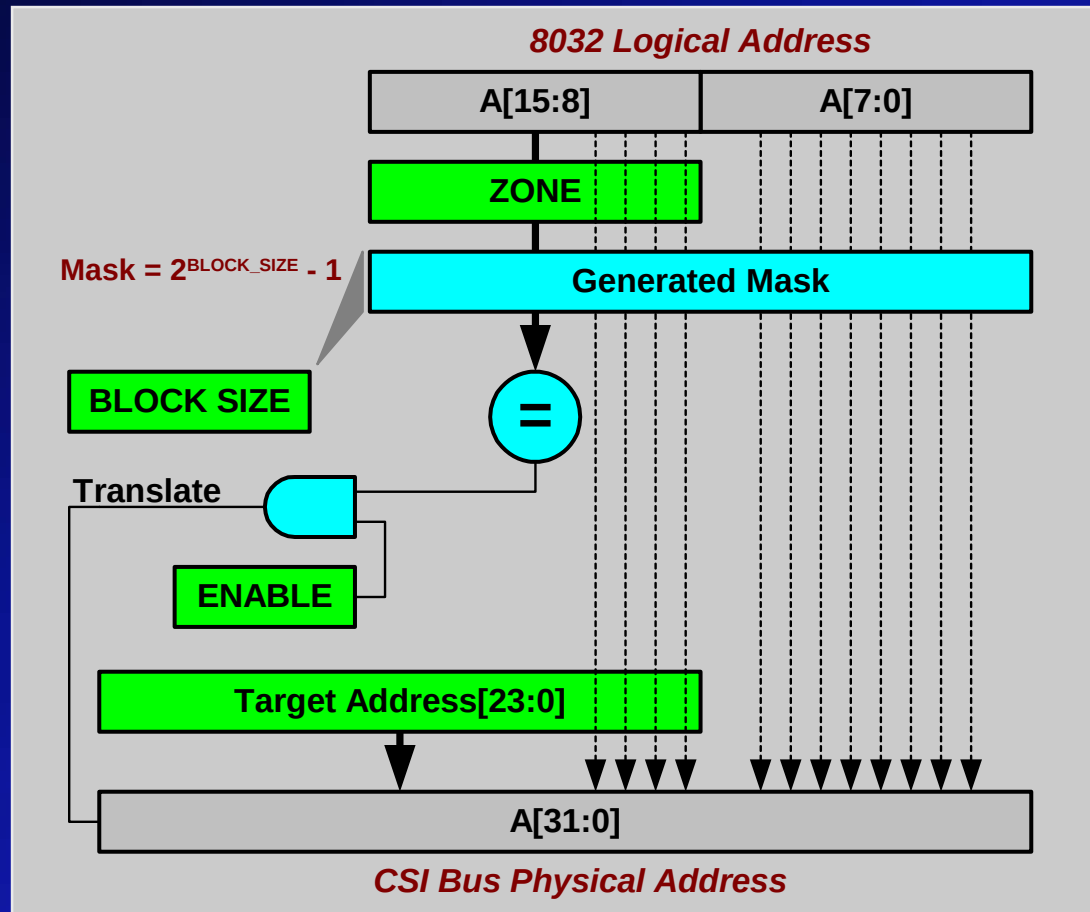
■ Creates a 32-bit physical address based on ...

- The MCU's 16-bit logical address
- The current memory access type
 - XDATA
 - Code
 - SFR

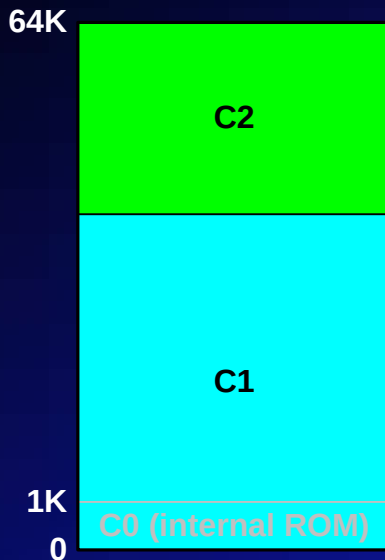
■ Provides glue-less code banking for application programs >64K

**Ignore the Mappers
for most applications**

Mappers: How They Work



Data and Code Mappers



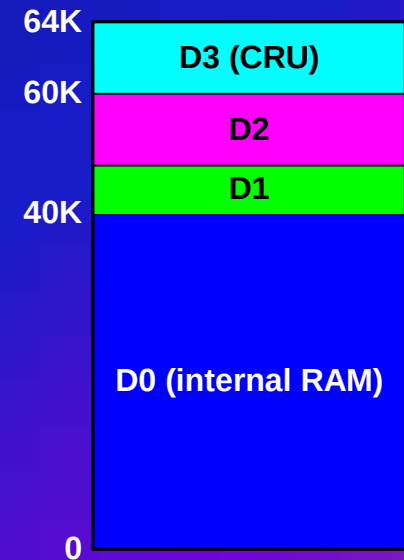
Three Code Mappers

Mapper	Priority	Enable
C2	1	Optional
C1	2	Optional
C0 (ROM)	3	Always Enabled

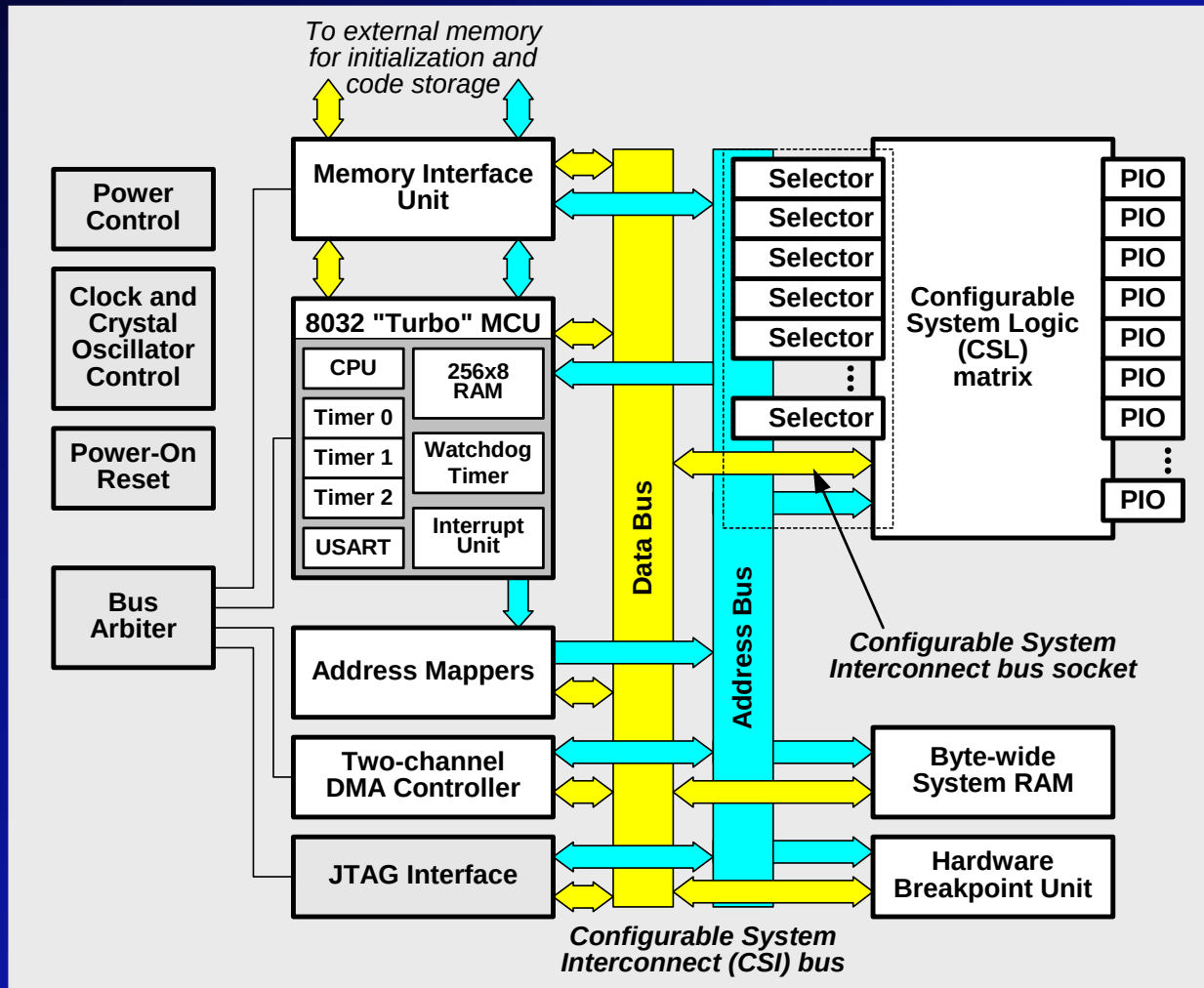
“Disappears”
after initialization

Six Data Mappers

Mapper	Priority	Enable
D3 (CRU)	1	Always Enabled
D5	2	Optional
D4	3	Optional
D2	4	Optional
D1	5	Optional
D0 (RAM)	6	Always Enabled



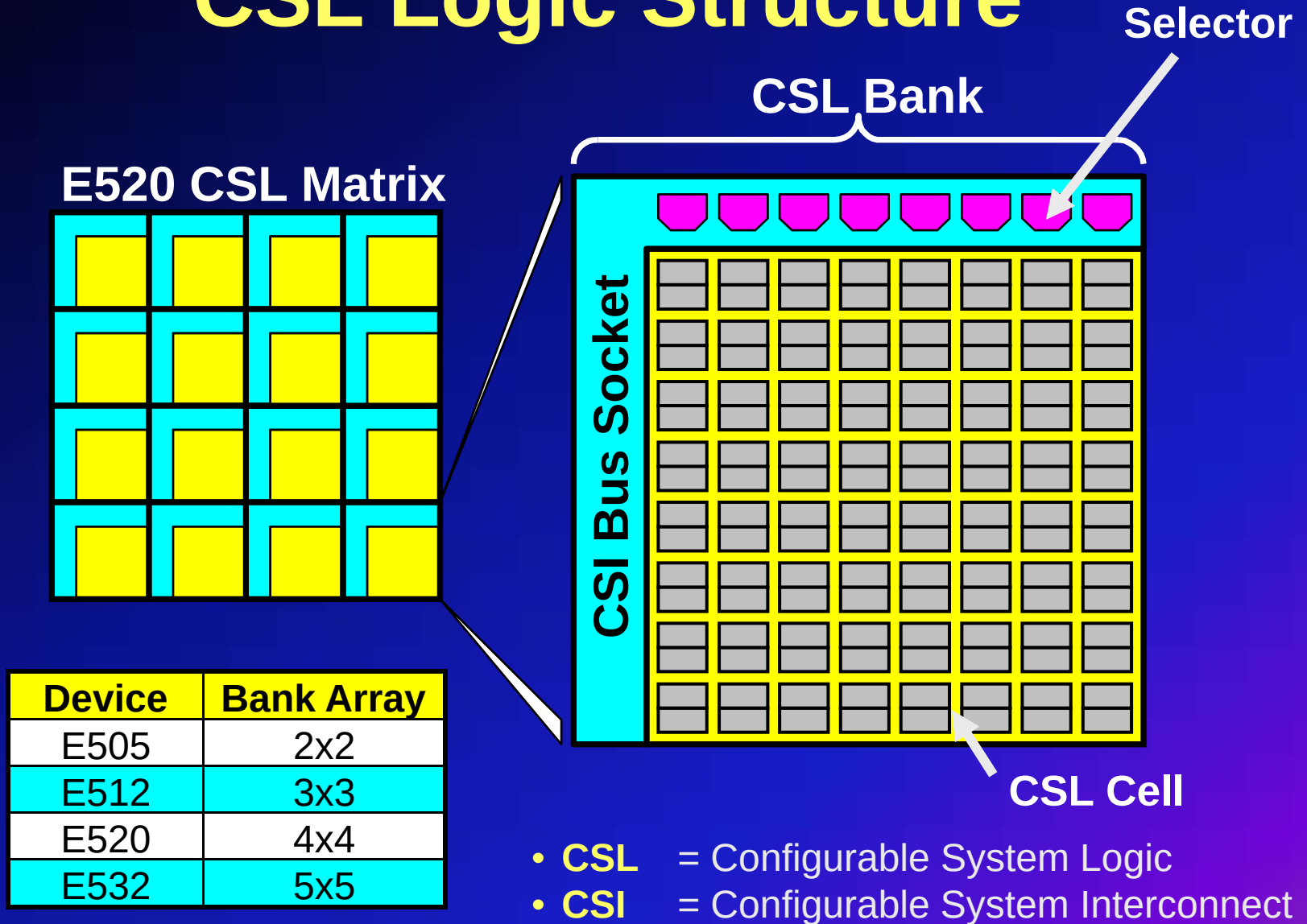
Triscend E5 Block Diagram



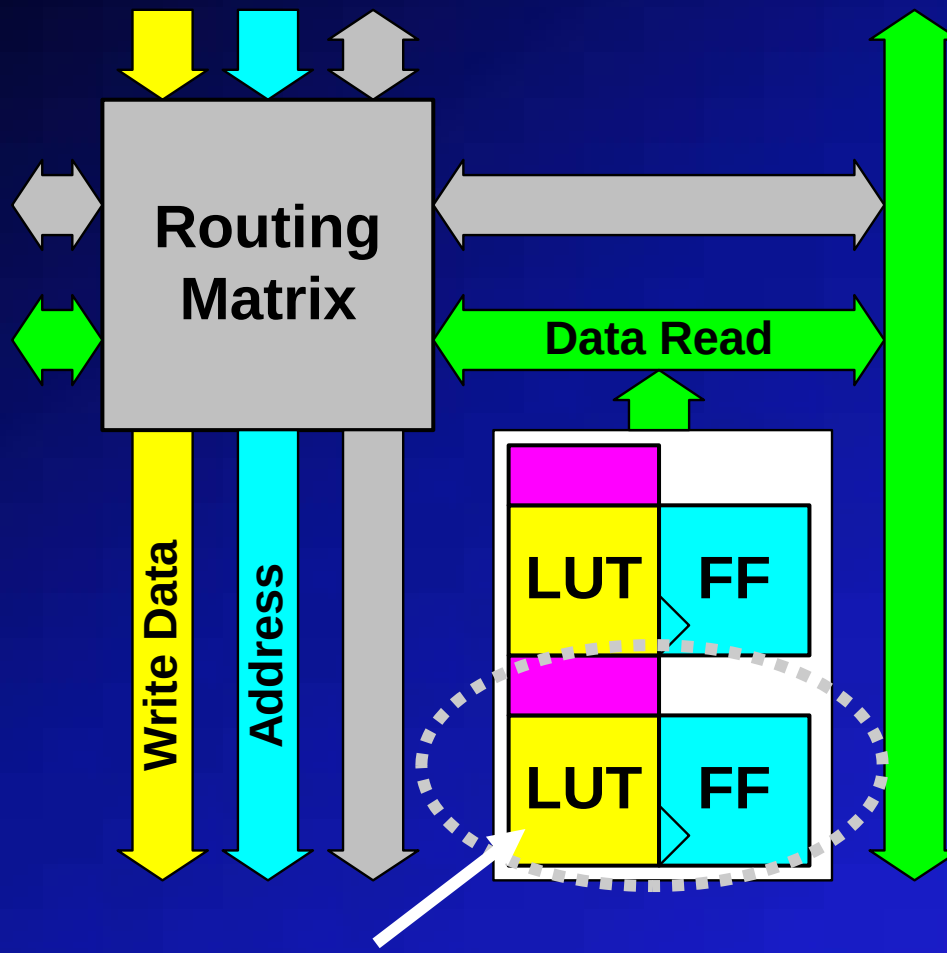
Configurable System Logic (CSL)

- Extends the capabilities of the MCU
- Create custom-tailored peripherals
- SRAM-based programmable logic structure
 - Popular LUT-style logic
 - Register-rich, very flexible
 - Programmable interconnect
- Intimate access to and from the Configurable System Interconnect (CSI) bus

CSL Logic Structure



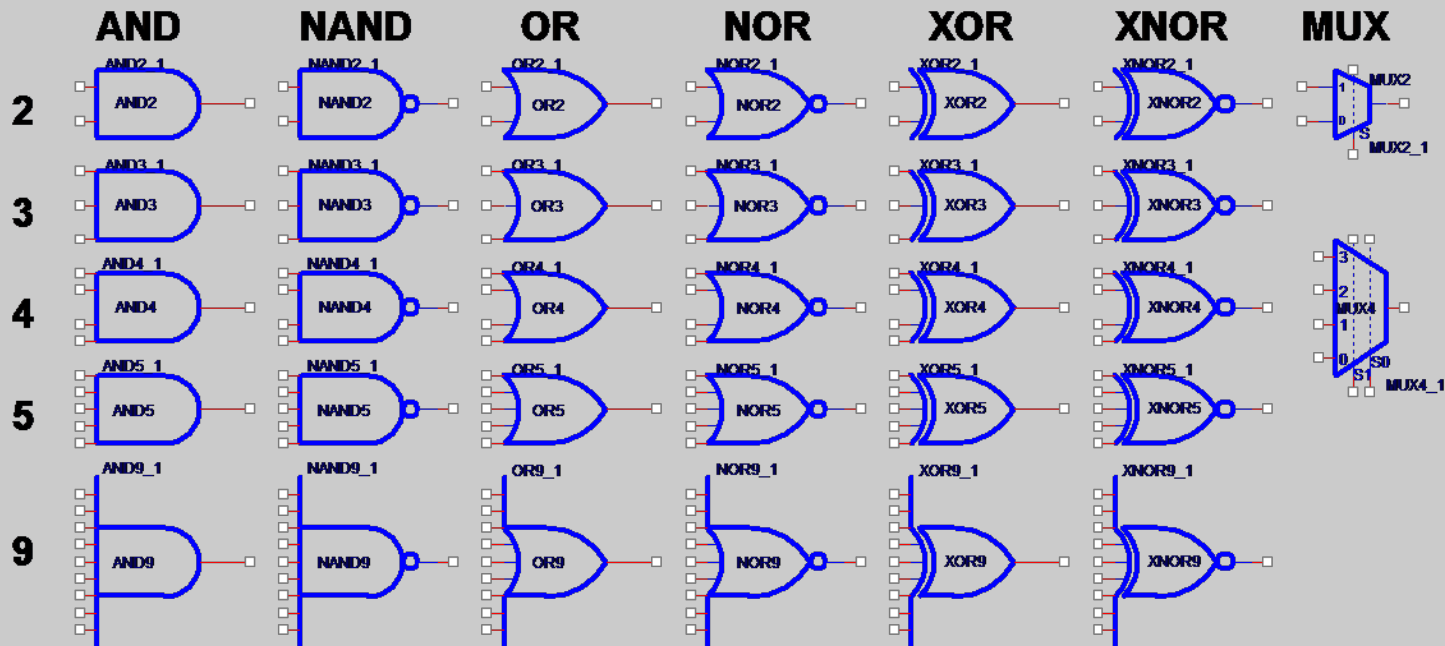
CSL Cell Structure



CSL Cell = LUT+FF

- CSL cell perform various functions
 - Logic
 - Arithmetic
 - Memory
 - Bus
 - Sequential
- Intimate connection to the CSI system bus

CSL Primitives (Logic)



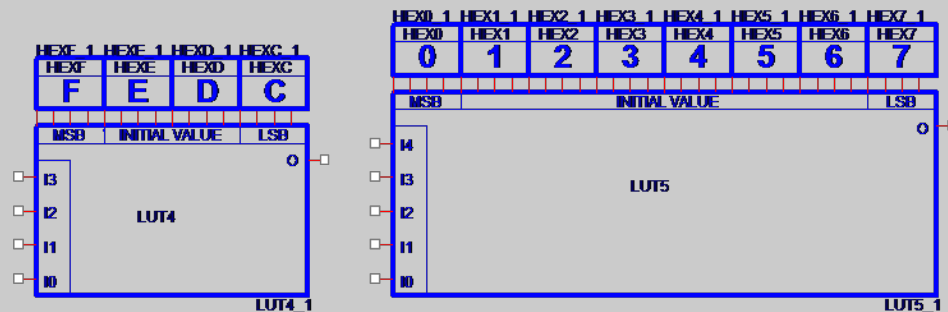
BUFFER



INVERT

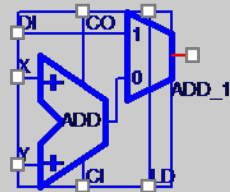


LOOKUP TABLE

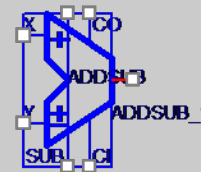


CSL Primitives (Arithmetic)

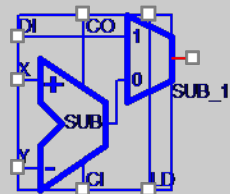
ADD with LOAD



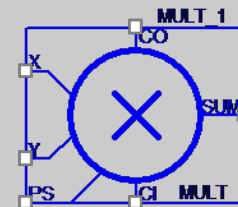
ADD/SUBTRACT



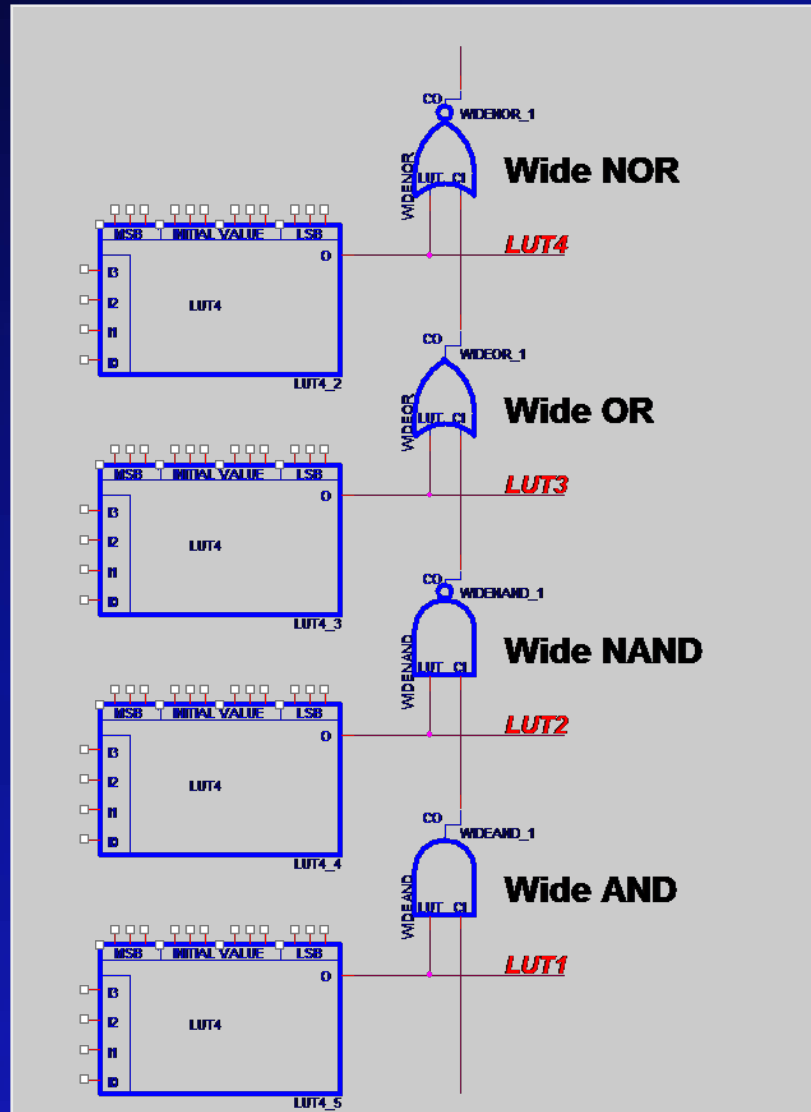
SUBTRACT with LOAD



MULTIPLY

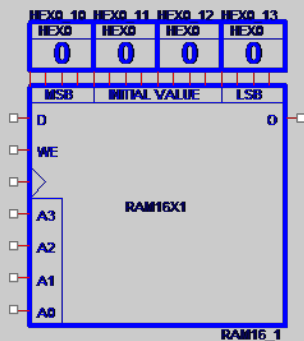


CSL Primitives (Wide)

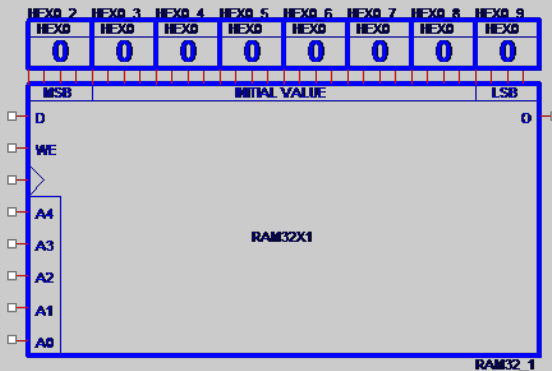


CSL Primitives (Memory)

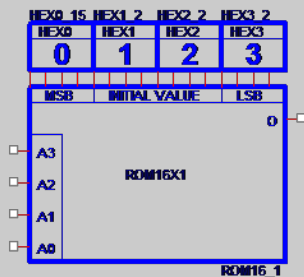
16x1 RAM



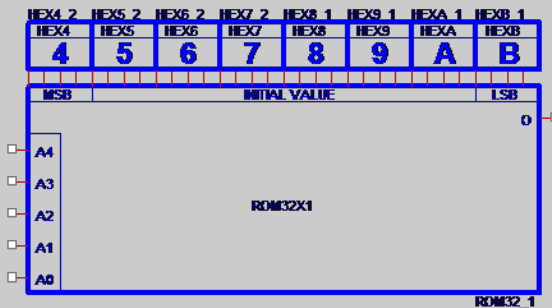
32x1 RAM



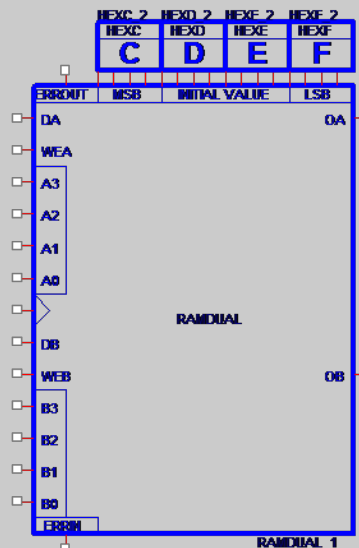
16x1 ROM



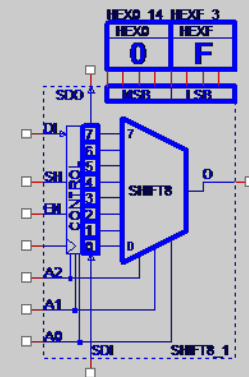
32x1 ROM



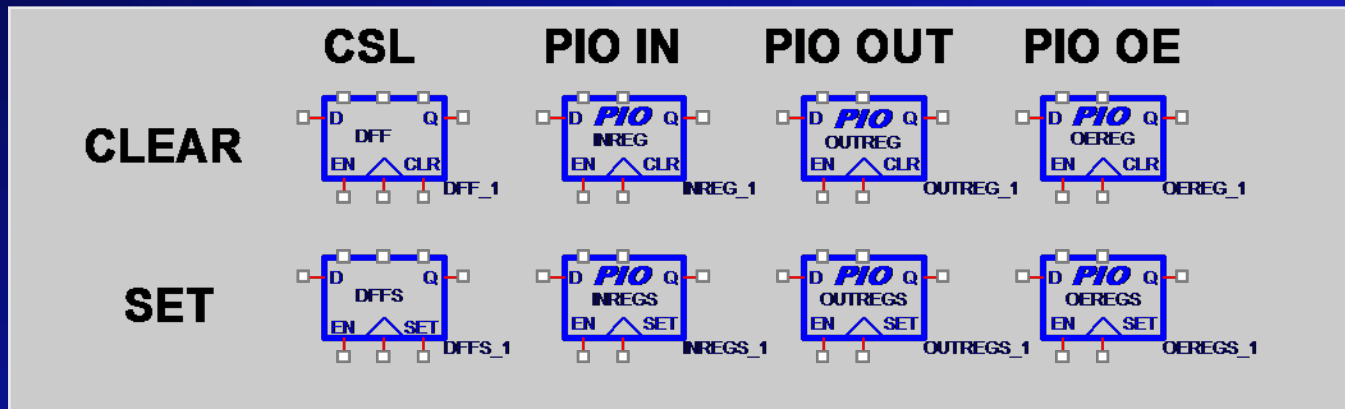
16x1 Dual-Port RAM



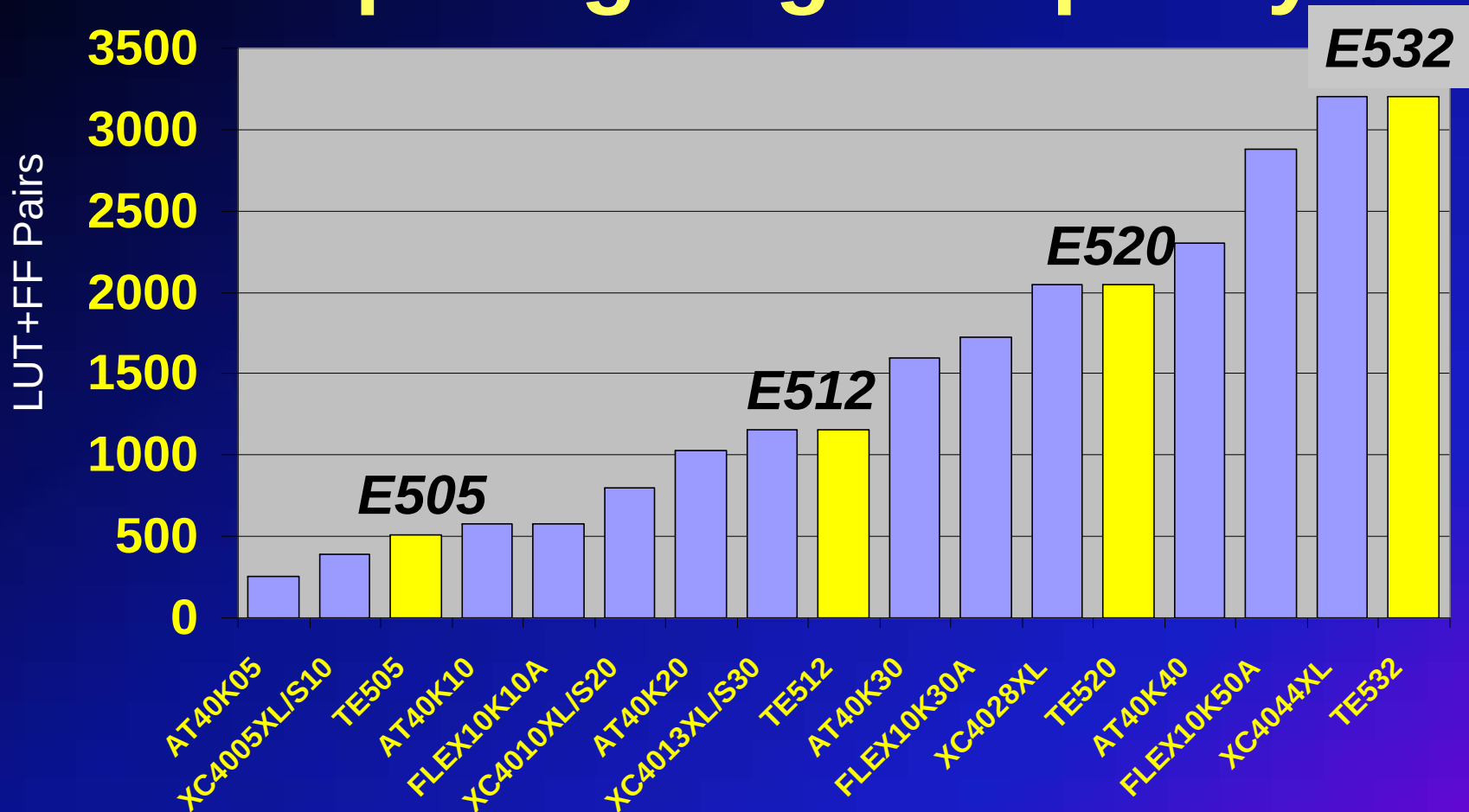
8-bit Serial-In/Serial-Out Shift Register



CSL Primitives (Sequential)



Comparing Logic Capacity

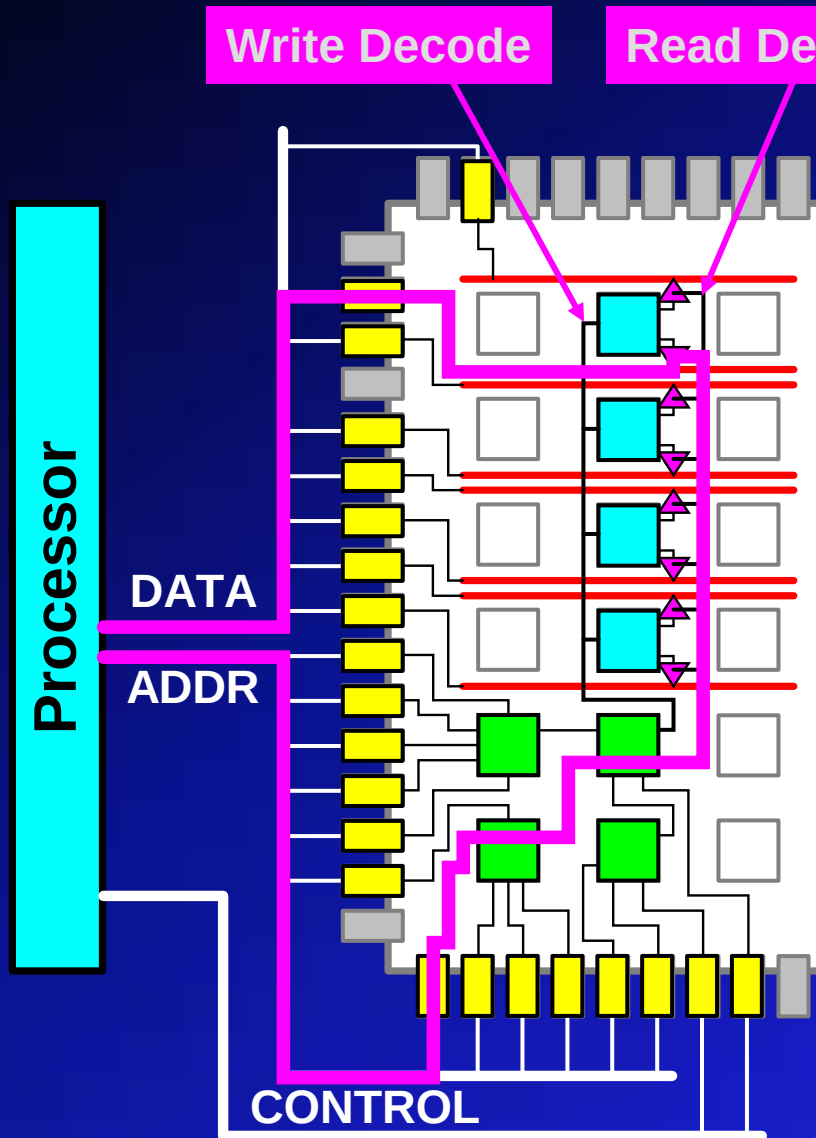


- Triscend CSL cell = 1 LUT4+FF pair
- Xilinx CLB = 2 LUT4+FF pair
- Altera FLEX LE = 1 LUT4+FF pair
- Atmel logic cell = 1 LUT4+FF pair

Communication between the Processor and Programmable Logic

- Connecting the data and address bus
- Decoding/controlling bus transactions
- Register intimacy
- Debugging

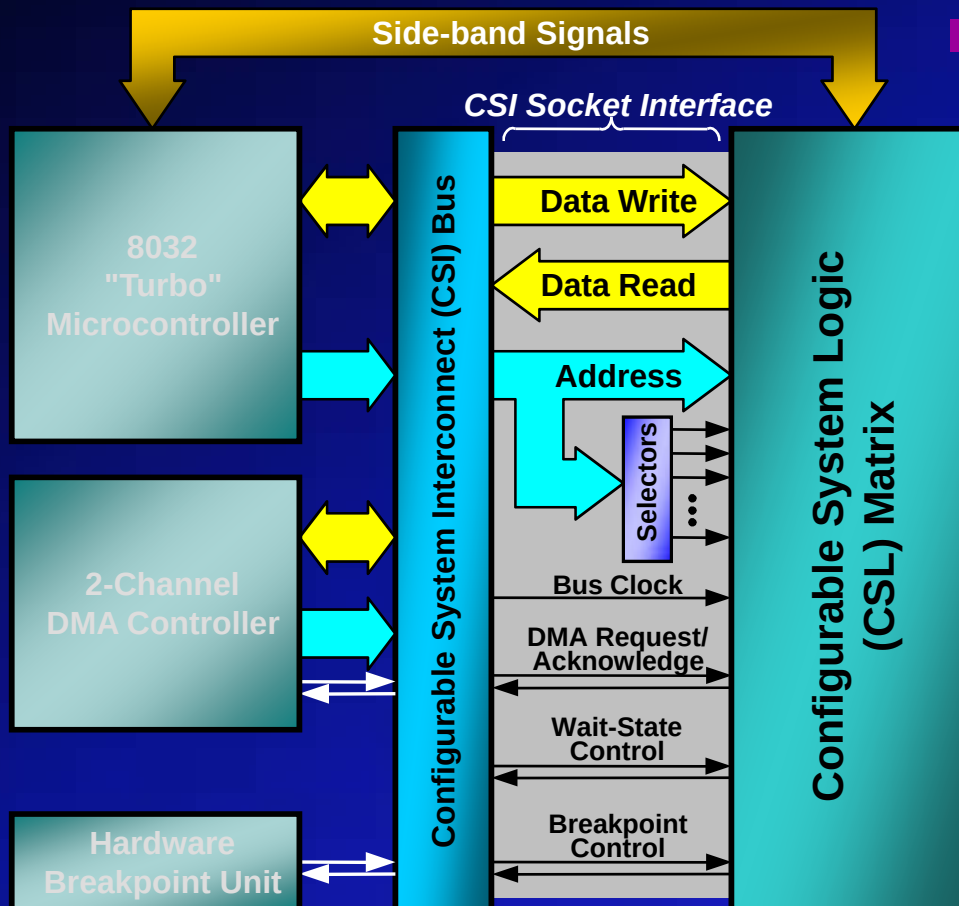
Routing Bus Signals: FPGA Example



- I/Os between devices
 - Many required, even for basic 8-bit interface)
 - Adds delay to critical path
 - Extra power consumption and EMI in two-chip solution
- Distributing address/data on-chip
 - Uses programmable interconnect
 - Adds delay to critical path
 - Variable delays in some architectures
 - Some devices provide bidirectional bussing

Another Approach: CSI Bus Socket

(Configurable System Interconnect)

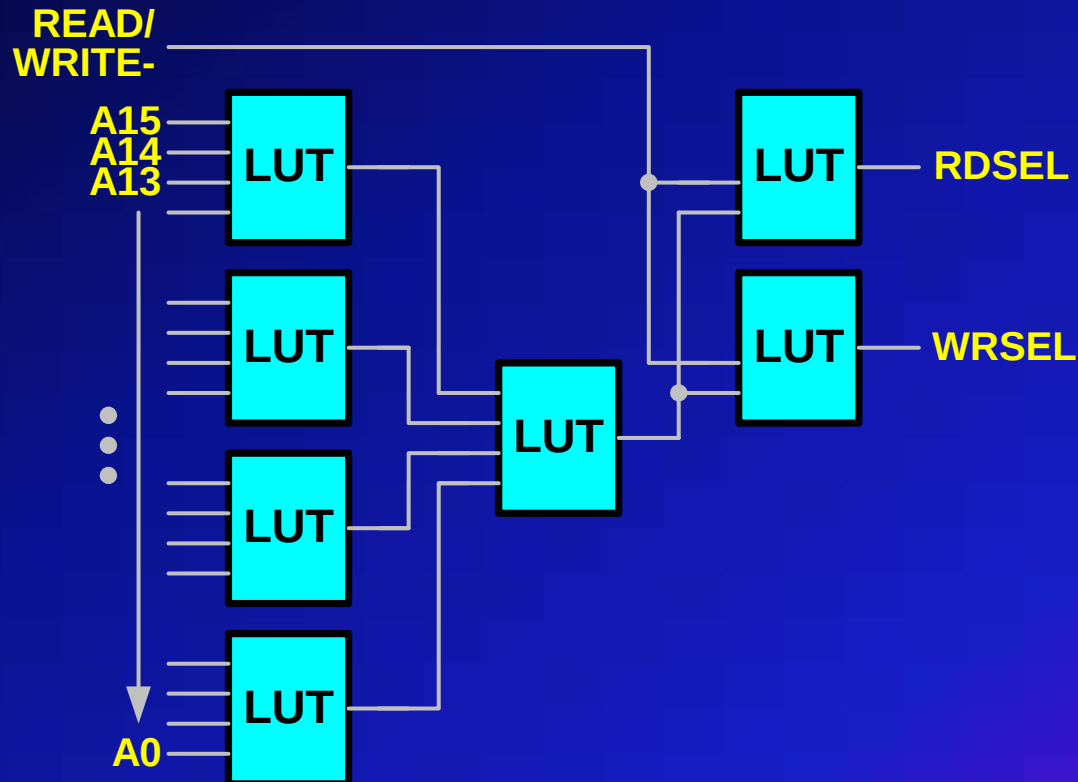


■ Distributes address and data to CSL matrix

- No additional I/O required
- Dedicated address decoding
- Predictable, synchronous timing
- Forward compatible with future configurable processors
- Wait-state control
- Contention-free bussing

Decoding Bus Transactions

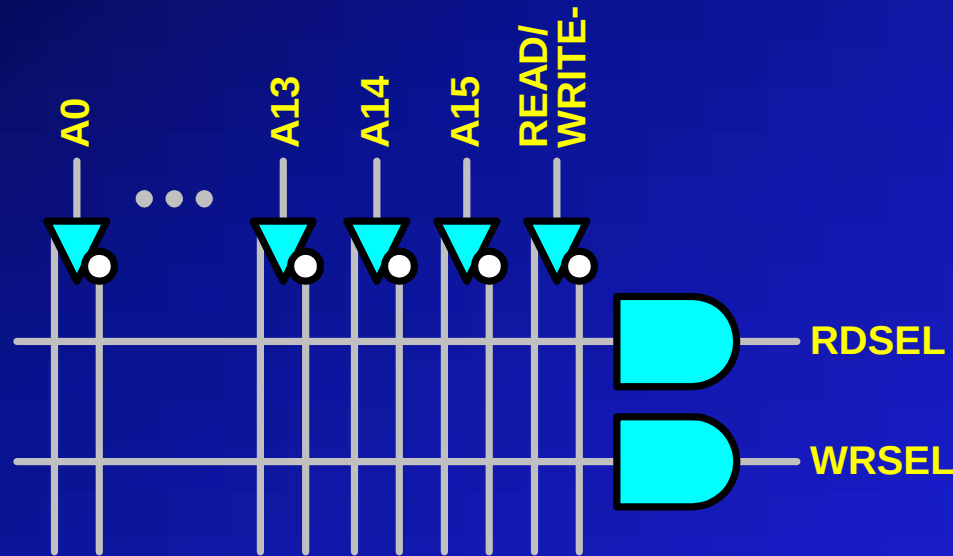
FPGA Style



- Decode delay is fan-out and routing dependent

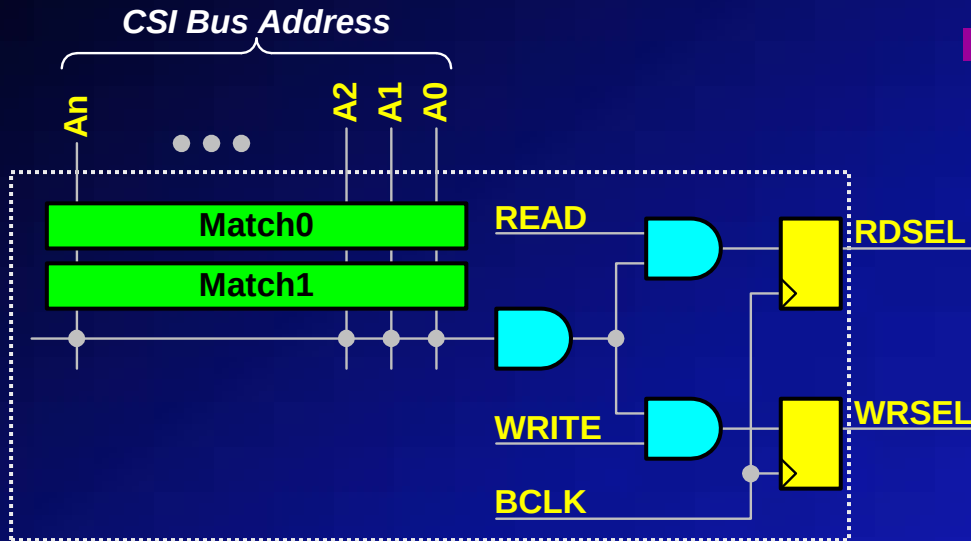
Decoding Bus Transactions

CPLD Style Decoding is Ideal



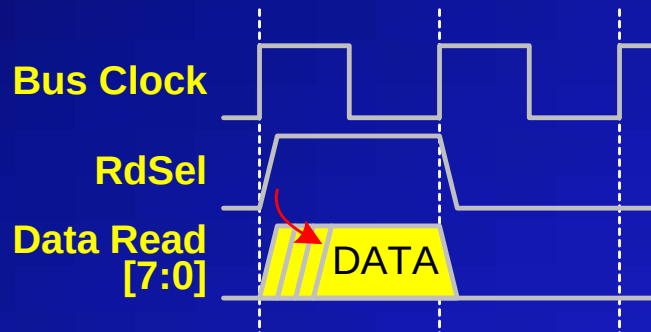
- Decode delay is constant

Selector



Fast address decoding

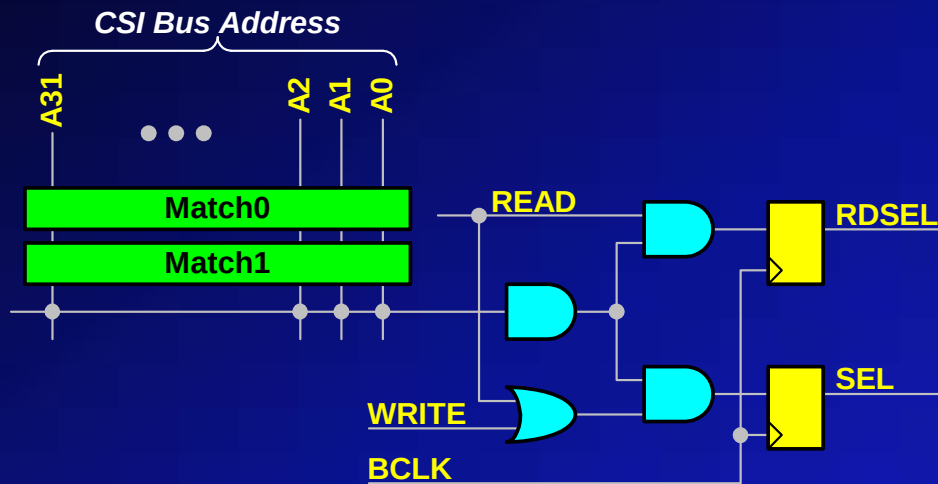
- Any address range
- Access type
 - Code
 - Data
 - Special Function Register (SFR)



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

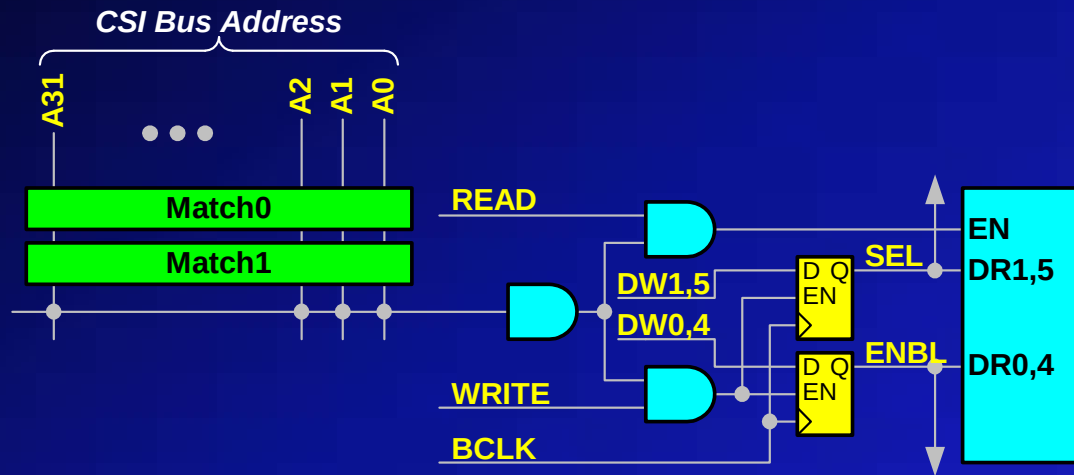
Device	Selectors
TE505	32
TE512	72
TE520	128
TE532	200

Chip Select



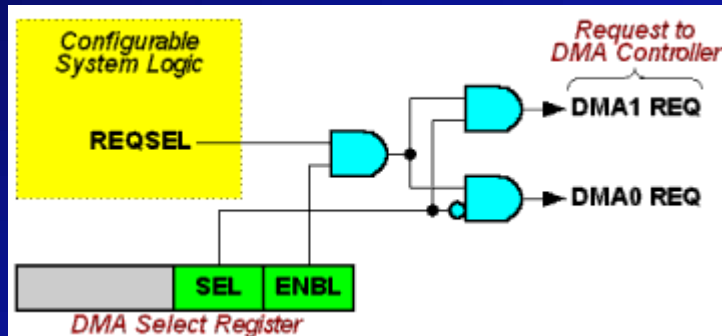
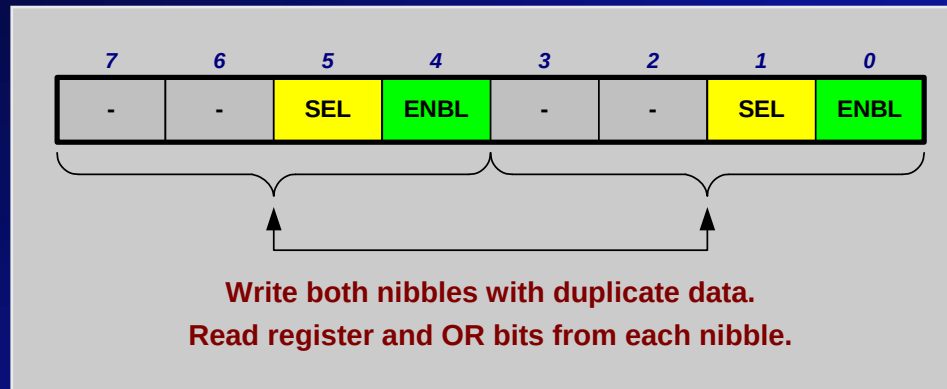
- Same Selector hardware, different mode
- Separately decodes select (read or write) and read (direction control)

DMA Control Register

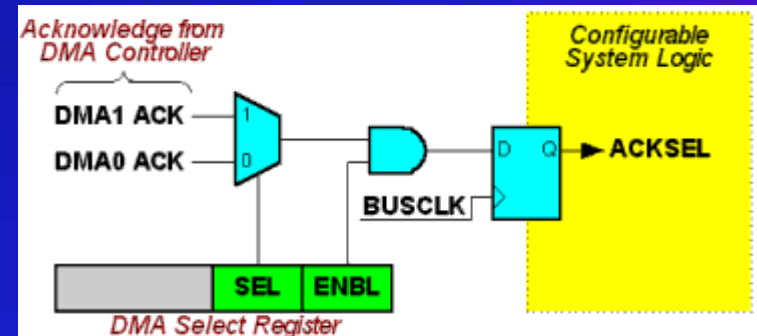


- Same Selector hardware, yet another mode
- Controls two registers that steer DMA request and acknowledge signals to the appropriate DMA Channel
- Enable bit (ENBL) appears on either D0 or D4
- Select bit (SEL) appears on either D1 or D5

Distributed DMA Steering

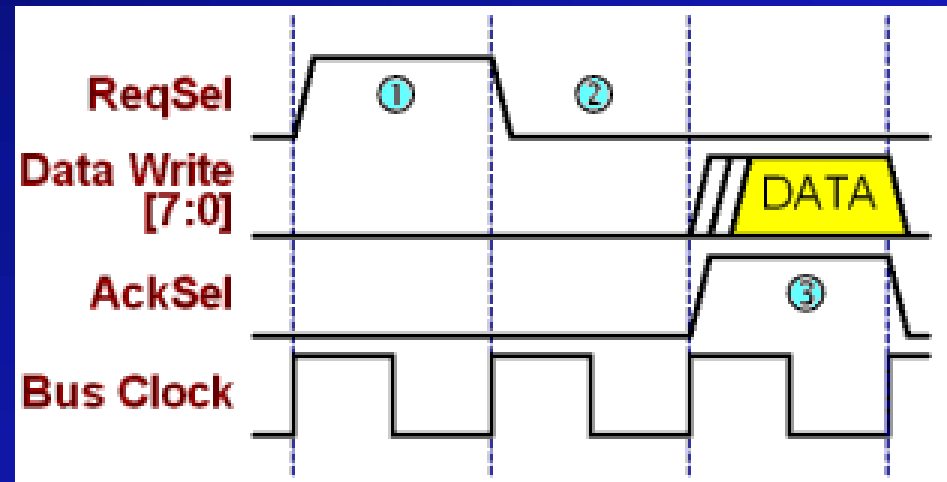


DMA Controller



Example DMA Transaction

- The CSL Peripheral asserts the **REQSEL** input on its associated DMA Control register.
- The appropriate DMA channel recognizes the request.
- The DMA channel then presents the write data on the CSI Socket's **DATA WRITE** bus. The CSL peripheral uses the **ACKSEL** signal from the DMA Control register to latch the data.



CSI Bus Arbiter

- Up to 7 potential bus masters
 - 8032 “Turbo” microcontroller
 - DMA Controller
 - Channel 0 Read
 - Channel 0 Write
 - Channel 1 Read
 - Channel 1 Write
 - JTAG Interface
 - Memory Interface Unit (slave mode only)
- Fixed, round-robin arbitration

More CSI Bus Information Later Today

Initialization

How does a CSoC know what it wants to be when it powers up?

Initialization

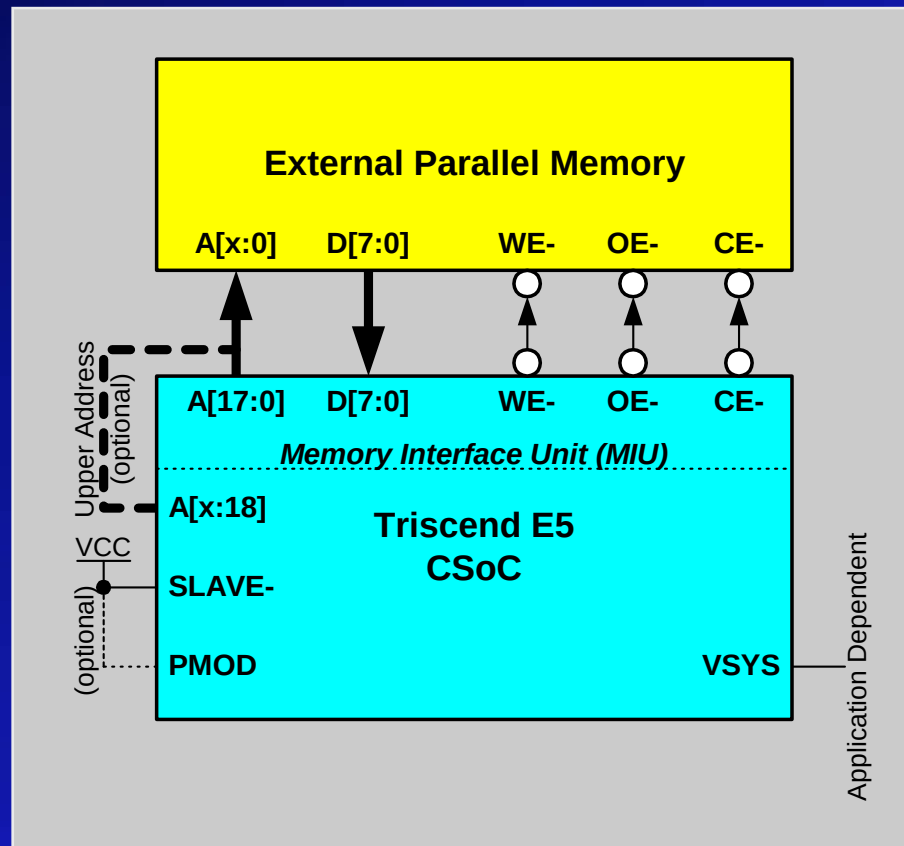
- Initialization is the process of configuring the CSoC to perform the desired function
- Occurs at power-up or whenever the RST- pin is asserted Low
- Loads the CSL matrix, configures the dedicated resources, then executes the application code
- Various modes available

Initialization Modes

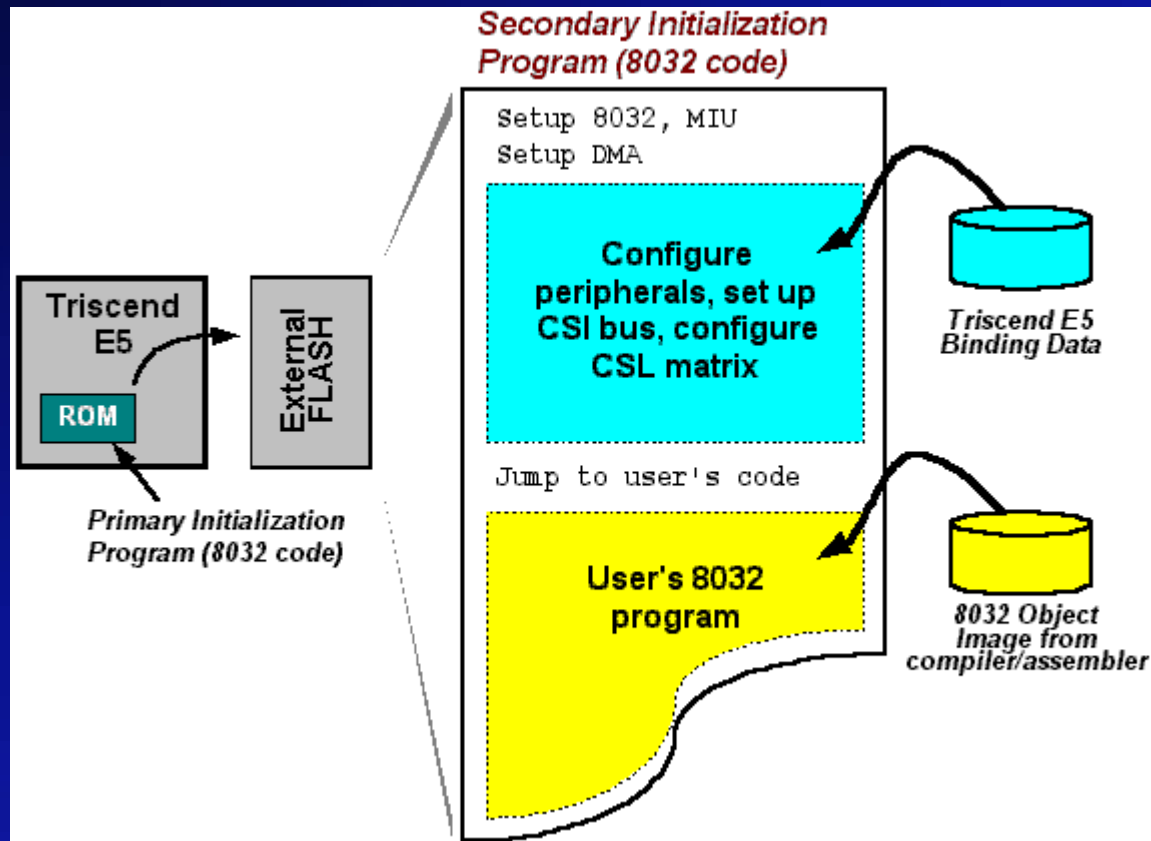
Initialization Mode	Method	Data Source
Parallel	Active	Byte-wide parallel memory (FLASH, EPROM, etc.)
Serial	Active	Sequential-access serial PROM
Stealth	Active	Battery-backed internal SRAM
JTAG	Passive	Downloaded by intelligent host through JTAG port
Slave	Passive	Downloaded by other controller through MIU bus interface.

Parallel Mode

- Load automatically from standard byte-wide external memory

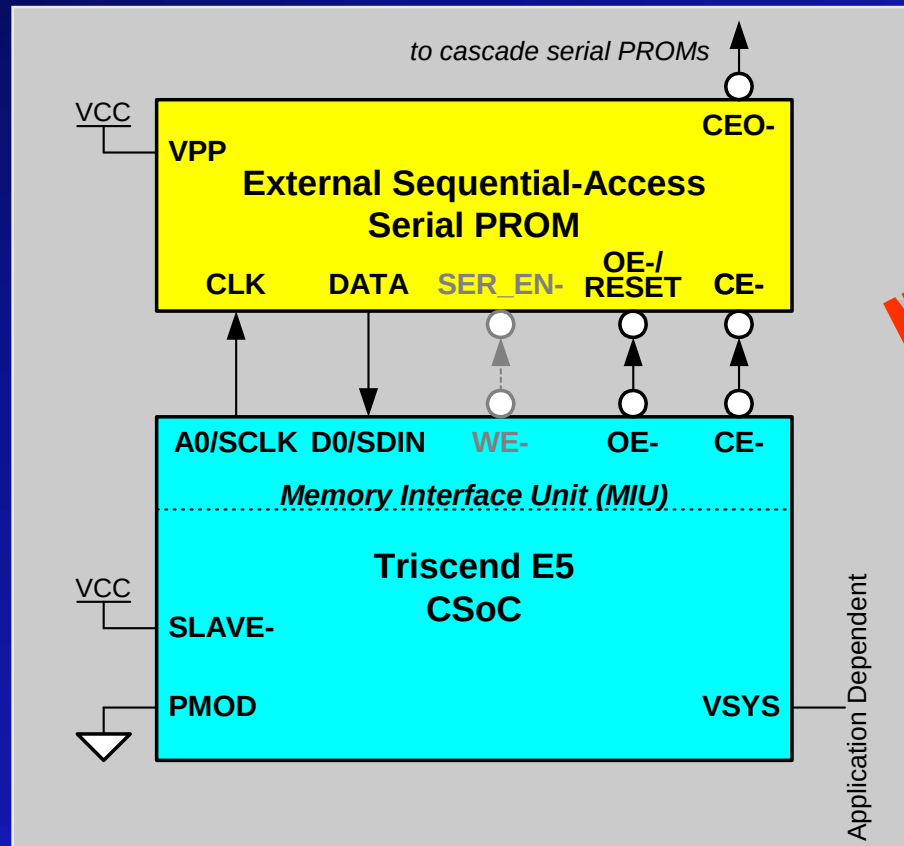


Parallel Initialization Process



Serial Mode

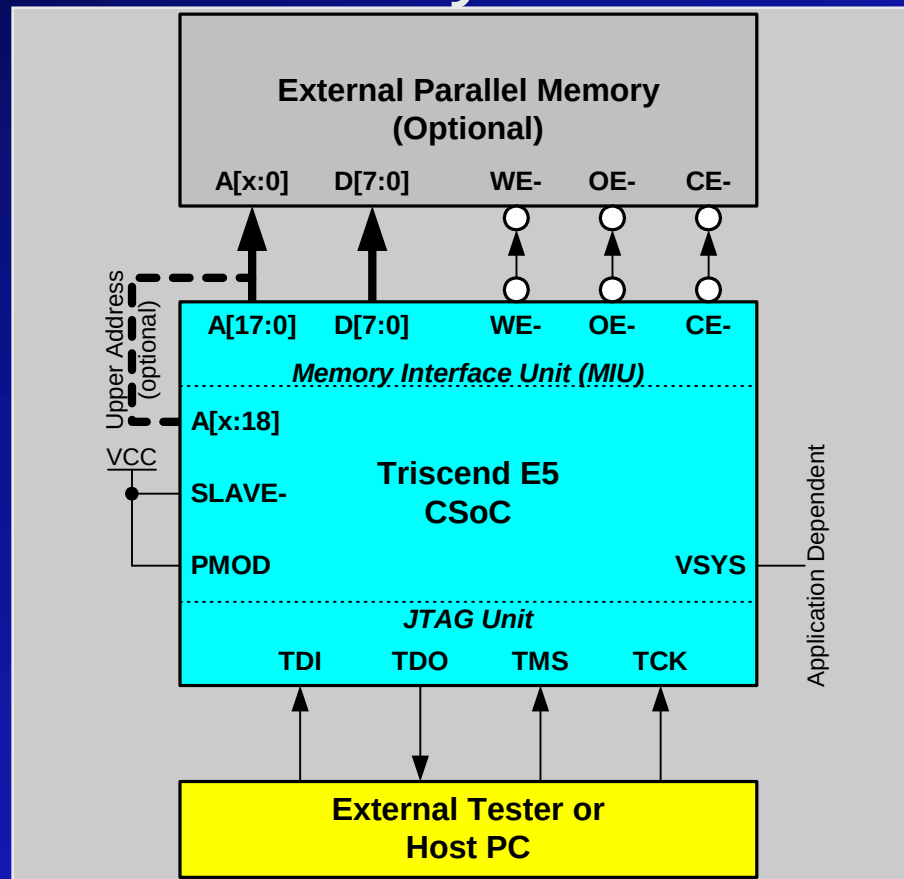
- Serial PROM offers 24 additional PIO
- Code executed from internal SRAM



No Software Support Today

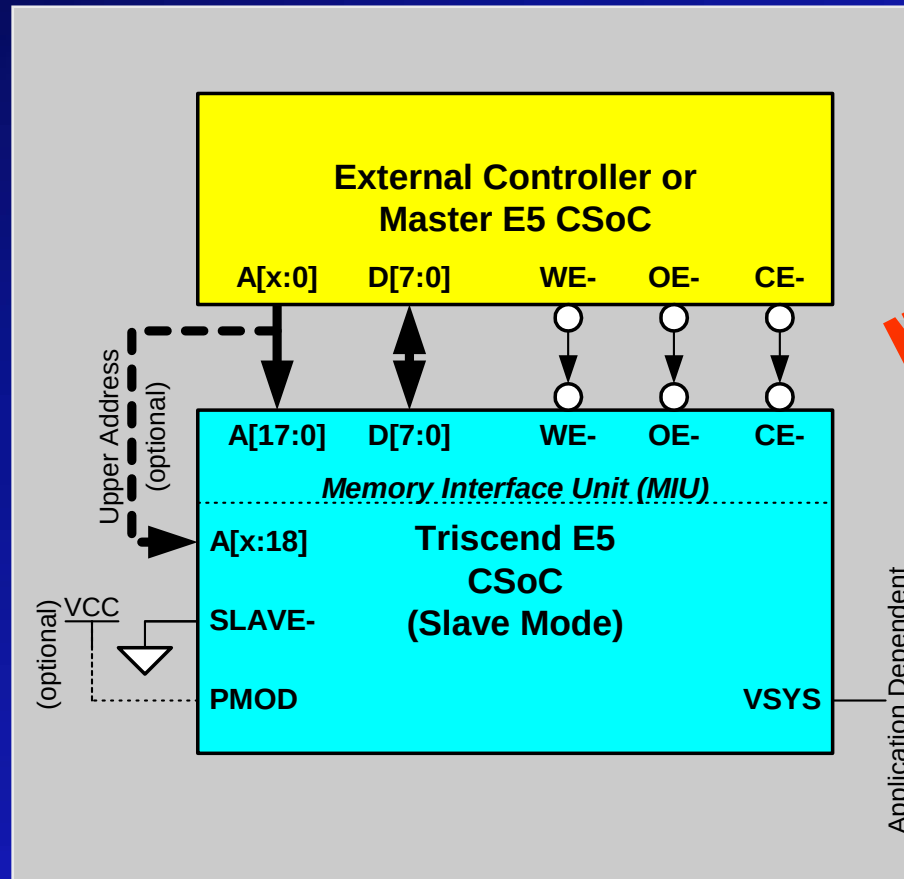
JTAG Mode

- Download directly to device or program an external memory



Slave Mode

- Download programming from external processor



No Software Support Today

CSoC Applications

Applications/Segments

■ Datacom/Telecom	Cellular Base Station LAN Switching	Line cards System Diagnostics
■ Industrial	Motor Controls Radio Identification	Handheld Terminals Process Control Equip.
■ Instrumentation	Data Logging Telecom Test Equip.	Laboratory Instruments Avionics
■ Automotive	Engine Diagnostics Dashboard Controls	Brake Systems Chassis Communications
■ Office/Home	Security Systems Encryption Systems	Computer Peripherals Equipment Diagnostics

Triscend E5 Applications

- MCU “Derivative on Demand”
 - Hardware peripherals for your specific application
 - Single-chip integration (lower power, EMI)
- System-on-a-chip
 - Faster development than ASIC or hard-masked SoC
 - Higher system density, faster than FPGA
- Hardware acceleration
 - Algorithms in silicon

Processor-Style 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)
- Easy to modify
- Cheap, re-use existing silicon

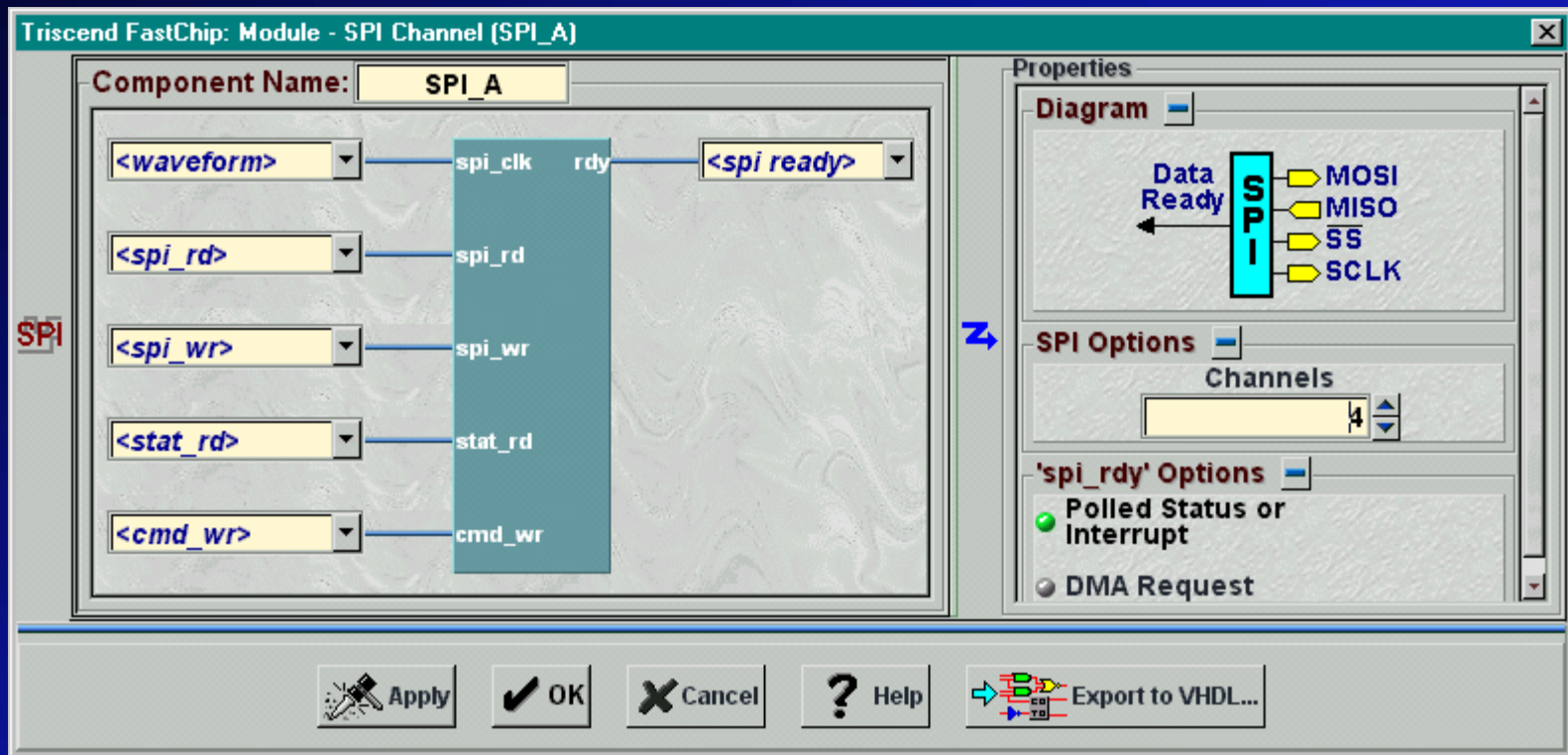
■ **Hardware Solution** (ns to μs)

- 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

FastChip SPI “Soft” Module



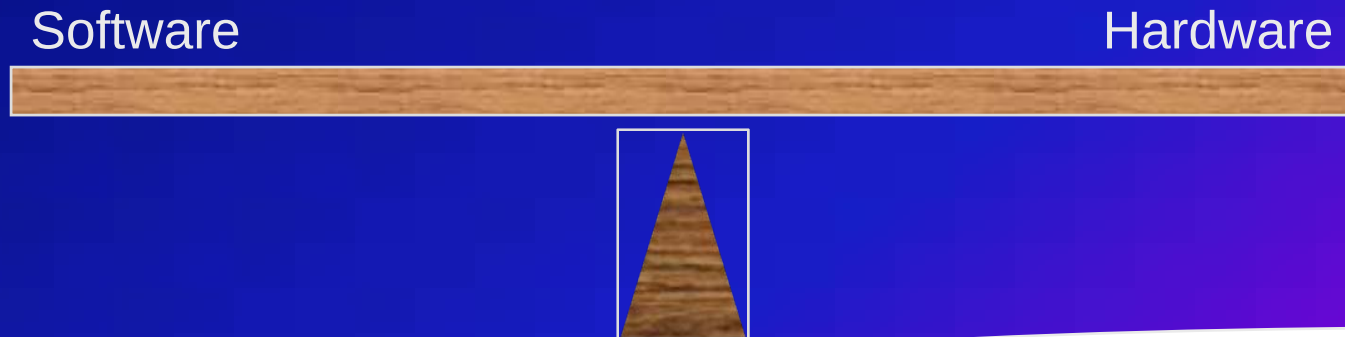
Design Techniques

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

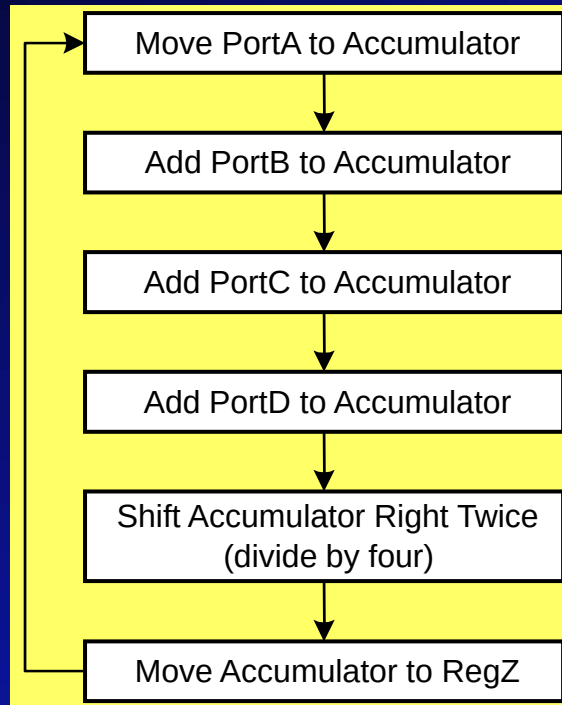
- Calculate the instantaneous average of four 8-bit values

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

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

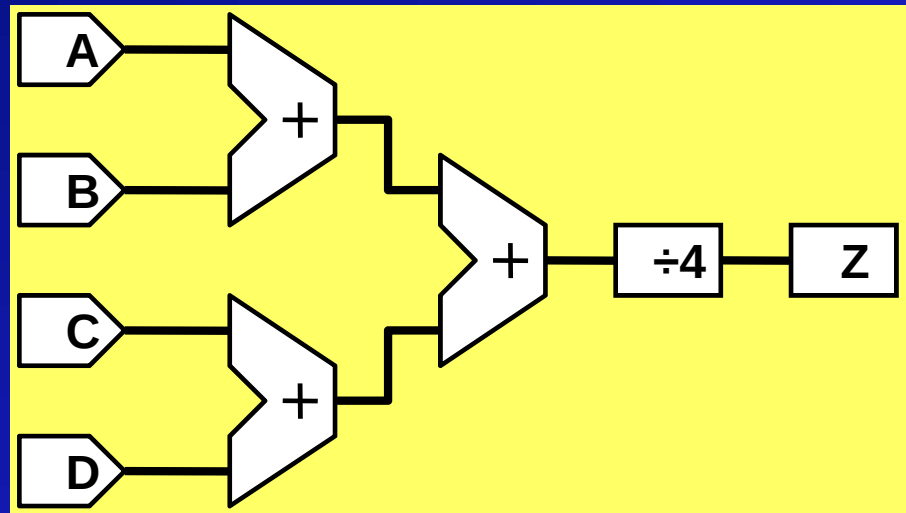
Two Solutions

■ Processor Solution



More instances require
additional time

■ Logic Solution



More instances require
additional logic

Comparing the Alternatives

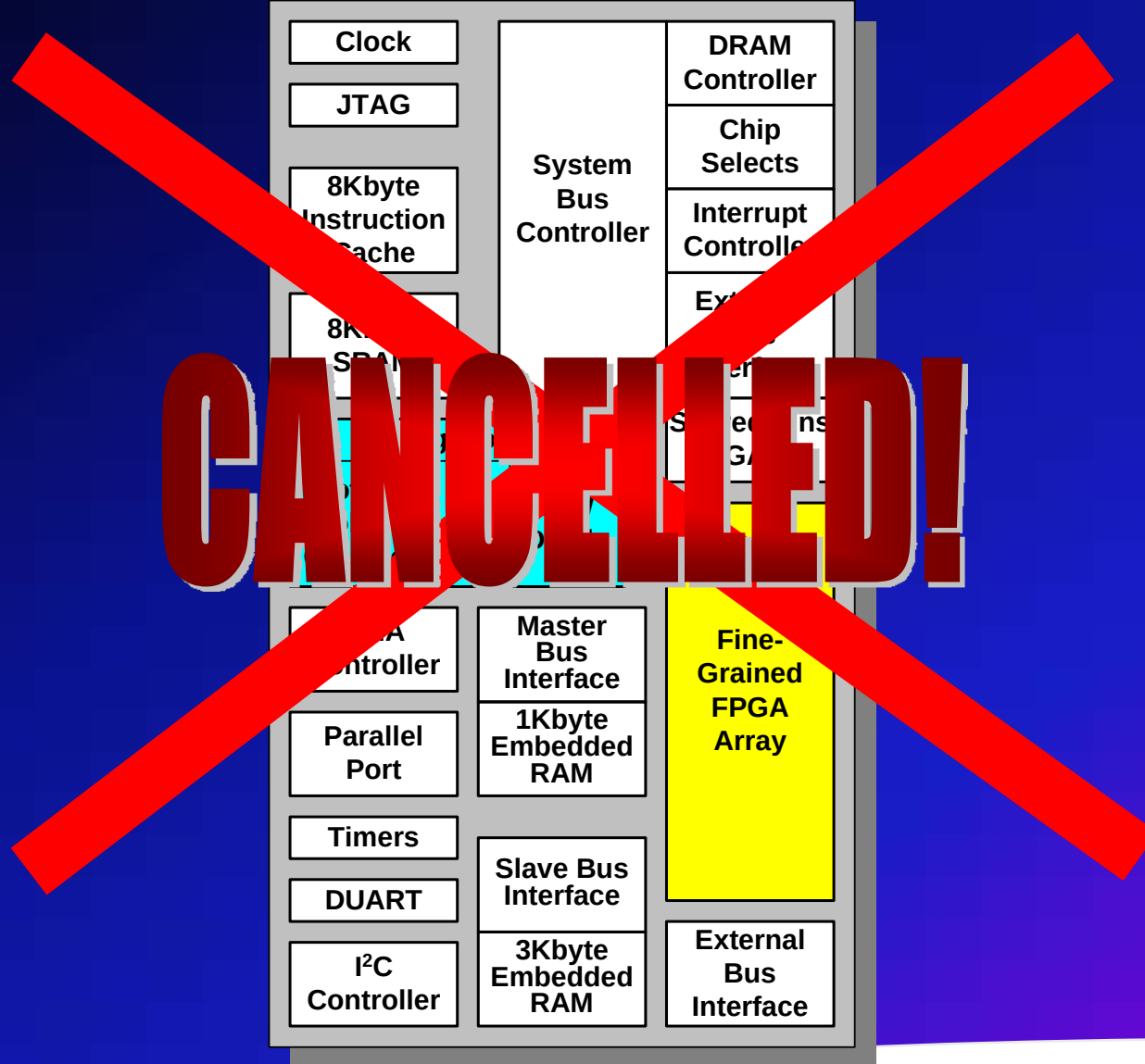
Solution	Device Cost	Development Time/Cost	Issues	When to Use It
Processor Derivative	\$1 - \$15	Quick/ Low	Availability, software support, differentiation	Lowest cost, if your application fits
System-on-a-Chip	\$5 - \$50 + development cost	Long/ High	Acquiring cores, verification, NRE, vendor selection	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 that require a processor not supported by a CSoC
Triscend E5	\$10 - \$80	Quick/ Moderate	New technology	Fast time to market, complete embedded system

Direct Competition

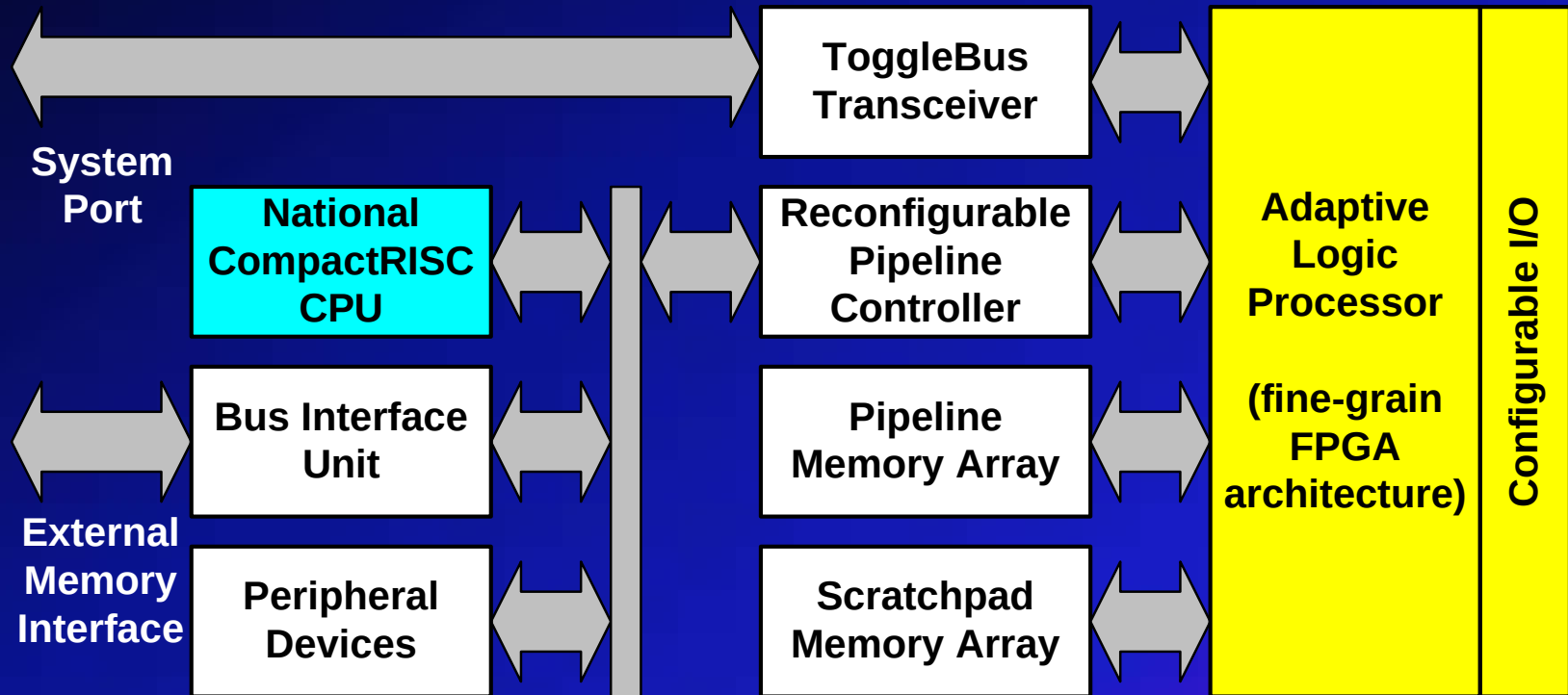
CSoC Devices

Vendor/ Family	Processor	Status	Dedicated Resources	Programmable Resources	Embedded Bus Structure
Triscend/ E5	8032 "Turbo"	Shipping	2-channel DMA 16-64K bytes RAM Hardware debug JTAG debugger	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 (unknown)	16K RAM 8x256 RAM Timer JTAG debugger	Concurrent Fine-grained	
Siemens	TriCore	Plans Announced		Gatefield Fine-grained	
Atmel	AVR	Plans Announced		Atmel AT40K Coarse-grained	
Chameleon Systems	ARC?	Plans Announced			
SIDSA/ FIPSOC	8031	Sampling?	Programmable analog	SIDSA Coarse-grained	None, Memory- mapped

Motorola CORE+



National NAPA



Review of This Section

- Learned the Triscend terminology
- Learned the architecture of the Triscend E5 Configurable System-on-Chip device
- Saw how the E5 features are integrated within the Triscend FastChip software
- Know applications where the Triscend E5 is a good fit
- Learned about competing technologies