Triscend®

Configurable System-on-Chip for Communications

Triscend A7 Configurable System-on-Chip Hardware Overview

ARM® POWERED

Triscend®

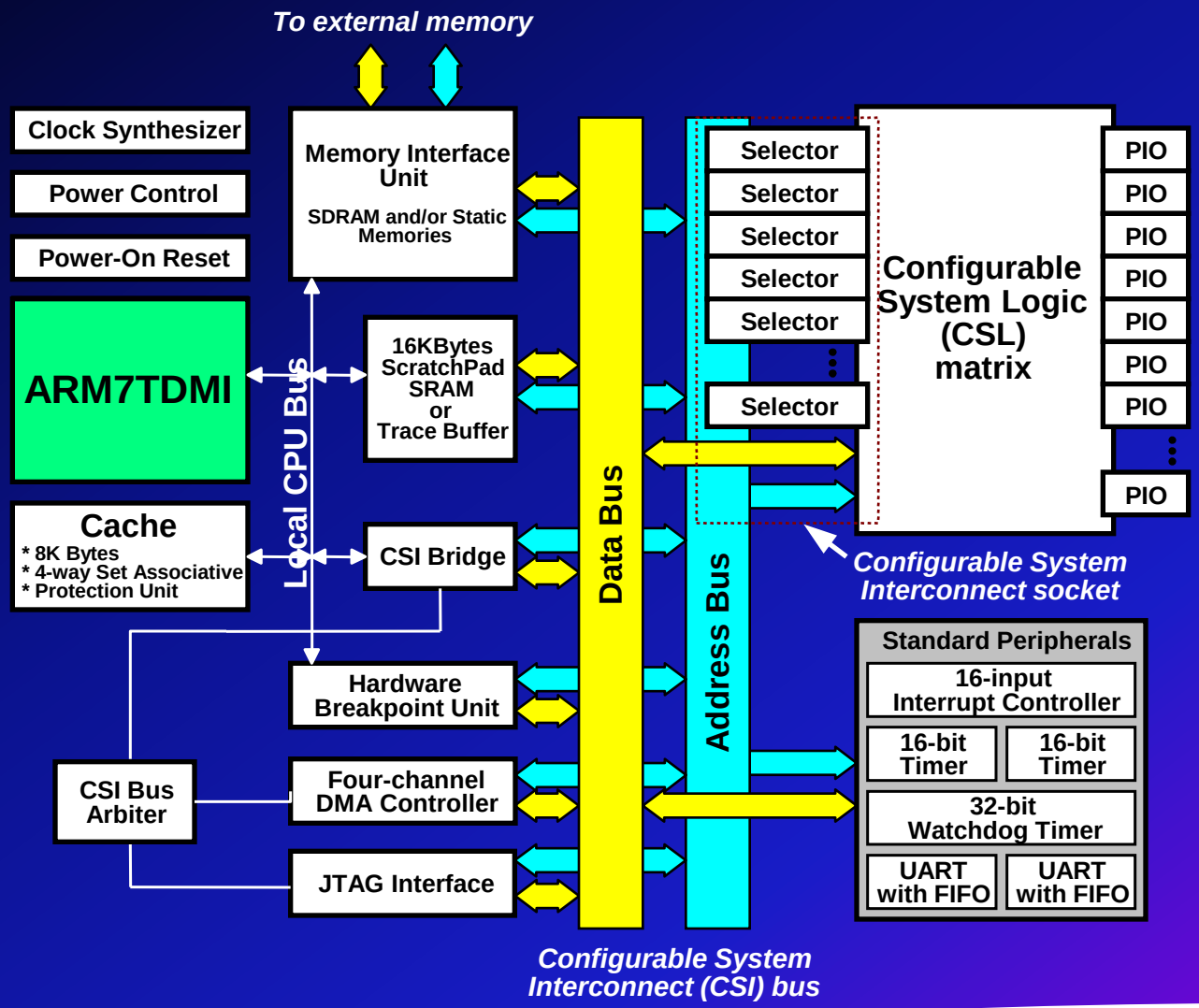
Agenda

- A7 Family Overview
- The ARM7TDMI Processor
- Configurable System Interconnect (CSI)
- Configurable System Logic (CSL)
- Programmable I/O (PIO) and Packaging
- System Memory, Scratchpad, Cache
- Memory Sub-System Interface Unit
- Clock Control, Power-Down

Agenda (cont'd)

- Interrupts
- DMA Controller and Interface
- Debugging CSoC Design
- On-Chip Support Peripherals
- Development Boards
- Summary

The Triscend A7



Triscend®

Triscend A7 Family

Device	Embedded Processor Core	Dedicated Resources	System RAM	Configurable System Logic (CSL) Cells	CSI Address Selectors	PIO Pins (Max)
TA7S05	**ARM7TDMI** 32-bit RISC CPU 8K unified cache Barrel shifter Hardware multiplier Thumb extensions Debug extensions	Memory interface unit 4-channel DMA controller Two 16C550-style UARTs Two 16-bit timers 32-bit watchdog timer 16-input interrupt controller Power management Power-on reset Hardware breakpoint unit JTAG port	4Kx32	512	32	124
TA7S12				1,152	72	188
T**A7S20**				2,048	128	252
TA7S32				3,200	200	316

Triscend®

The ARM Family of Processors

- **ARM7**
  - 3-stage execution pipeline
  - Von Neuman architecture (unified code and data space)
  - 0.9 MIPS/MHz
  - Optimized for low cost, low power
- **ARM9**
  - 5-state executing pipeline
  - Harvard architecture (separate code and data space)
  - 1.1 MIPS/MHz

The ARM Family of Processors (cont'd)

- **StrongARM (SA-110)**
  - Developed by ARM and DEC
  - Purchased by Intel Corp.
  - 5-stage execution pipeline
  - Harvard architecture with two 16Kbyte 32-way caches (instruction and code)
  - Enhanced 32x12 multiplier
  - No Thumb support
  - No debug/breakpoint support
  - Big, fast, more expensive
  - 1.25 MIPS/MHz

The ARM Family of Processors (cont'd)

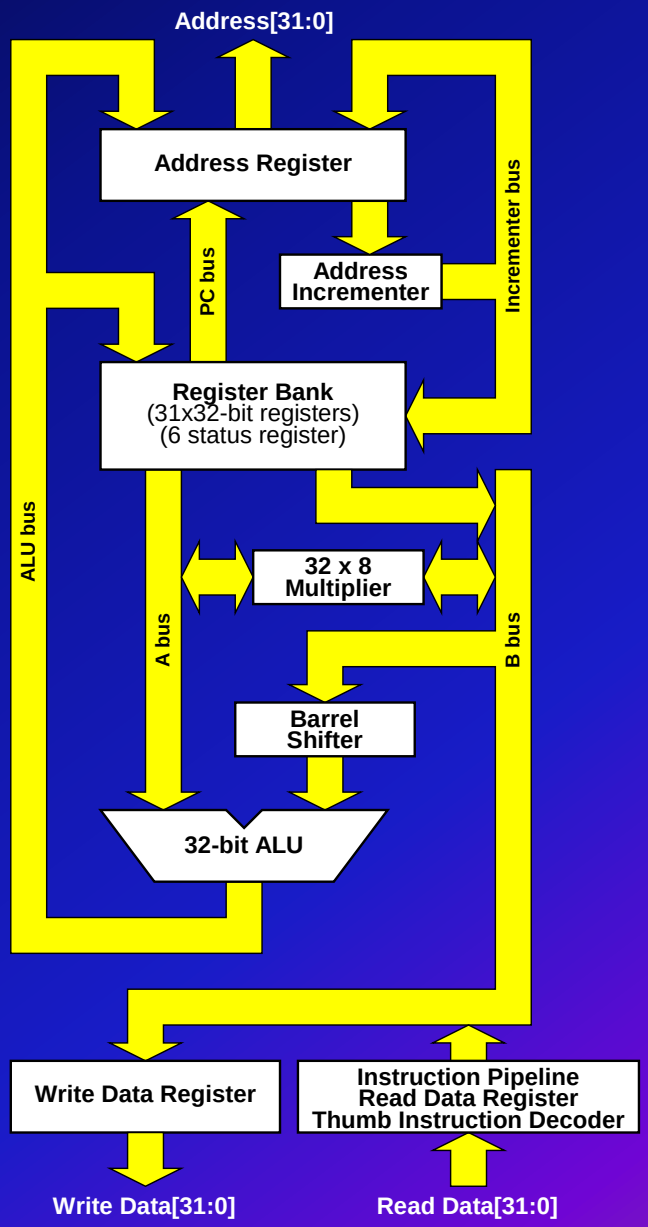
- **ARM10**
  - Vector floating point unit
  - 32Kbyte instruction and data caches
  - Parallel instruction execution
  - Branch prediction
  - 1.25 MIPS/MHz
  - Not your typical embedded processor
- **ARM1, 2, 3, ~~4~~, ~~5~~, 6, 8**
  - Obsolete

ARM7TDMI Alphabet-ese

- **ARM7** = ARM7 RISC architecture
- **T** = Thumb® instruction set support
- **D** = Debugging support via JTAG
- **M** = Multiplier performs single-cycle 32x8 hardware multiply
- **I** = ICEBreaker™ embedded breakpoint unit

ARM7TDMI Block Diagram

- Registers
  - 37 32-bit registers
  - 30 general-purpose
- 32-bit ALU
- 32x8 multiplier
- 32-bit barrel shifter
- Thumb instruction decoder
- 32-bit linear address space



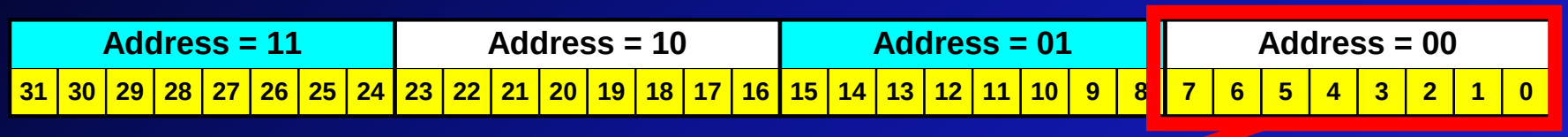
Triscend®

Data Sizes

- **The ARM7TDMI is a 32-bit architecture**
  - **Word** is 32 bits (four bytes)
  - **Half-word** is 16 bits (two bytes)
  - **Byte** is 8 bits
- **Triscend A7 is Little-Endian format**
- **ARM-state instructions are 32 bits**
- **Thumb-state instructions are 16 bits**

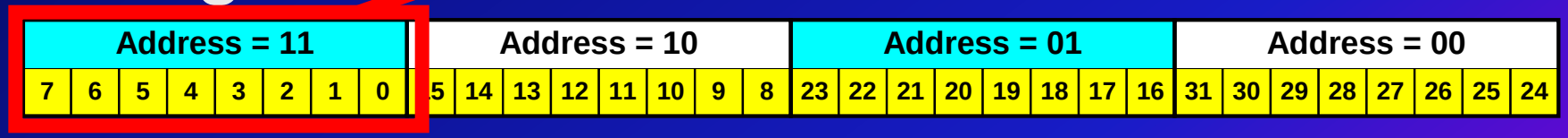
The Endian Wars

- **Triscend A7 is Little Endian Format**



  - Least-significant byte is at lowest byte address
  - Word addressed by address of lowest byte

- **Big Endian Format**



  - Most-significant byte is at lowest byte address
  - Word addressed by address of highest byte

Triscend®

ARM Processor States

- **ARM State**
  - 32-bit Instructions
  - Full richness of the ARM instruction set
- **Thumb State**
  - 16-bit Instructions
  - Limited but very capable Thumb instruction set
- **Switching Between States**
  - Cannot intermix in same code
  - Possible via branches
  - Best balance between performance, code density

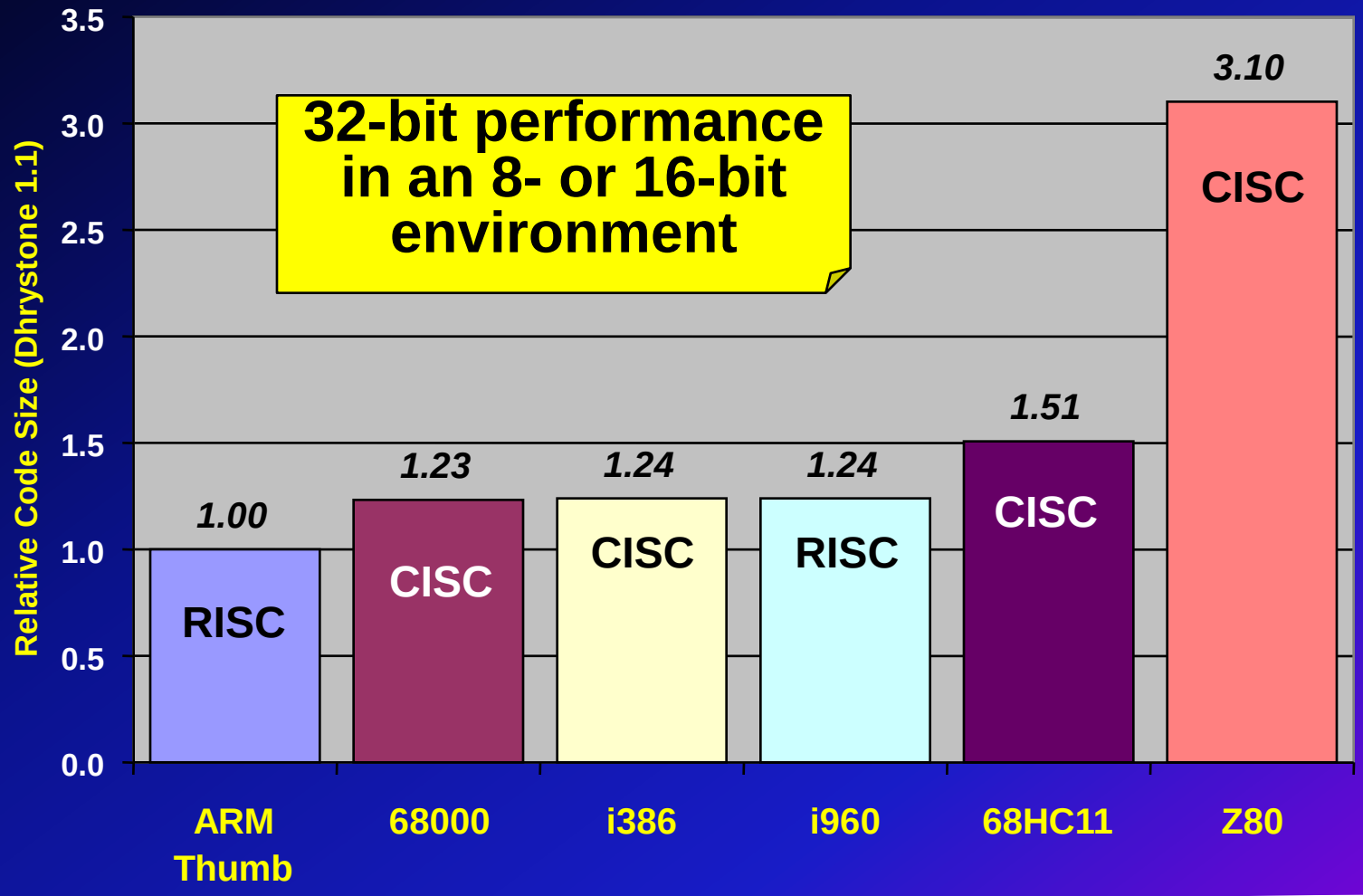
ARM State Instructions

- Conditional execution minimizes short branches
- “Free” barrel shift on second operand
- “CISC”-like instructions (Load and Store Multiple, `LDM`, `STM`)

Why Thumb?

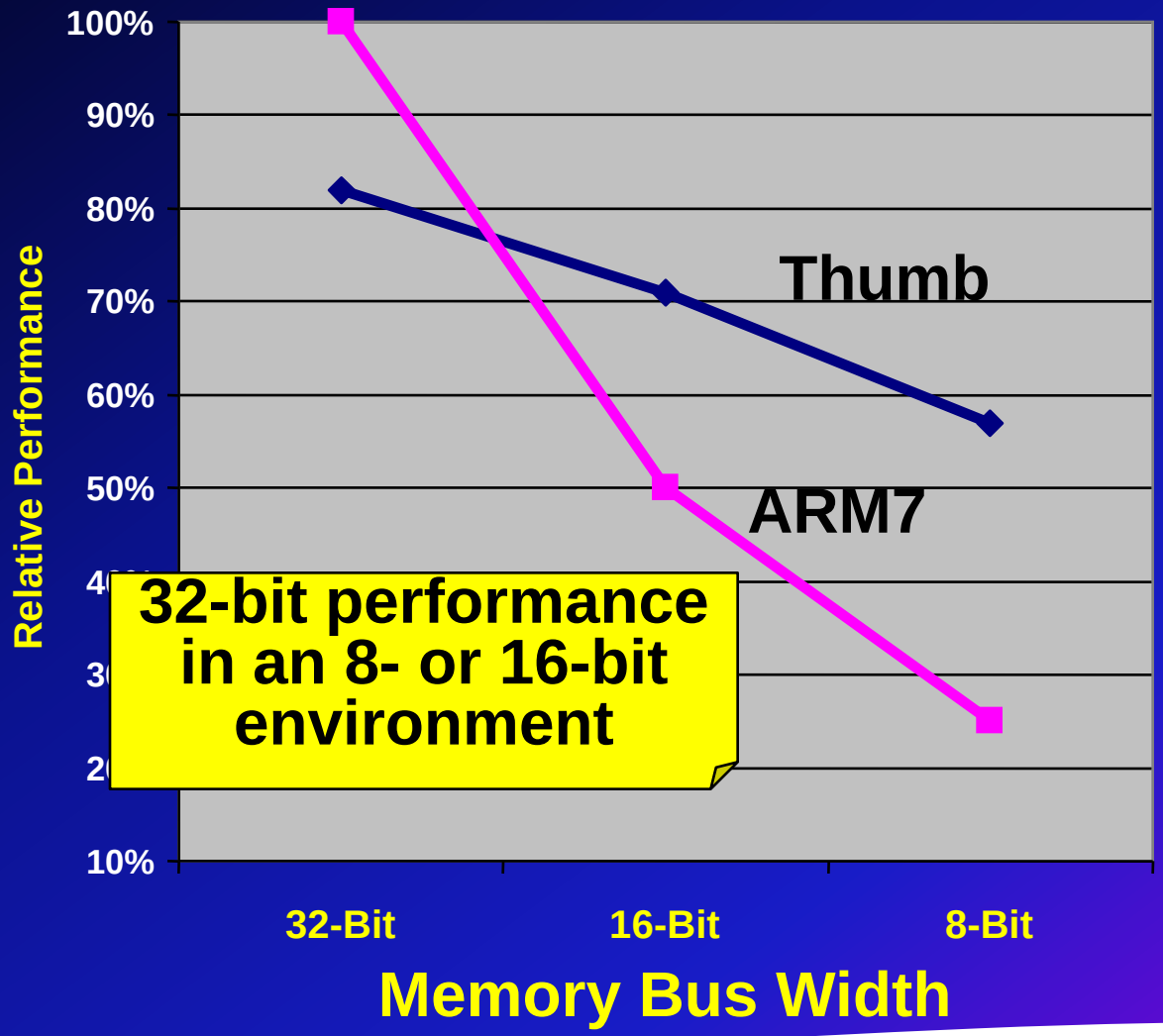
- **RISC inherently uses more ROM space, especially at 32 bits**
- **16-bit Thumb includes 36 most commonly-used instructions**
- **Smaller code size**
  - 32-bit processor in an 8- or 16-bit system
  - Reduced system cost
- **Improved performance in an 8- or 16-bit system**

Smaller Thumb Code Size



Triscend®

Performance in Smaller Systems



Triscend®

32x8 Hardware Multiplier

- Supports Multiply and Multiply-Accumulate
- Useful for DSP
- Early termination based on data, data type
- 32x32 = 32 in two to five cycles
- 32x32 = 64 in three to six cycles
- 32x32+32 = 32 in three to six cycles
- 32x32+64 = 64 in four to seven cycles

Processor Modes

- **User** – unprivileged mode used for most tasks
- **FIQ** – used to service a high-priority fast interrupt
- **IRQ** – used to service all lower-priority interrupts
- **Supervisor** – entered upon reset and when the Software Interrupt instruction is executed
- **Abort** – used to handle memory access violations
- **Undef** – used to handle undefined instructions
- **System** — privileged mode, uses same registers as User mode

ARM7TDMI Register Set

- **16 general-purpose 32-bit registers**
  - Really 30 in total, but some are banked based on context
  - Labeled **R0** through **R15**
  - Register **R15** is Program Counter
  - Register **R14** is Link Register
  - Register **R13** is Stack Pointer
- **Status Register**
  - Current Program Status Register (**CPSR**)
  - Five Saved Program Status Registers (**SPSR**), by context

ARM Mode Registers

System & User	FIQ	Supervisor	Abort	IRQ	Undefined
R0	R0	R0	R0	R0	R0
R1	R1	R1	R1	R1	R1
R2	R2	R2	R2	R2	R2
R3	R3	R3	R3	R3	R3
R4	R4	R4	R4	R4	R4
R5	R5	R5	R5	R5	R5
R6	R6	R6	R6	R6	R6
R7	R7	R7	R7	R7	R7
R8	R8_fiq	R8	R8	R8	R8
R9	R9_fiq	R9	R9	R9	R9
R10	R10_fiq	R10	R10	R10	R10
R11	R11_fiq	R11	R11	R11	R11
R12	R12_fiq	R12	R12	R12	R12
R13 (SP)	R13_fiq	R13_svc	R13_abt	R13_irq	R13_und
R14 (LR)	R14_fiq	R14_svc	R14_abt	R14_irq	R14_und
R15 (PC)	R15 (PC)	R15 (PC)	R15 (PC)	R15 (PC)	R15 (PC)

Thumb state Low registers

Thumb state High registers

ARM State Program Status Registers

CPSR	CPSR	CPSR	CPSR	CPSR	CPSR
	SPSR_fiq	SPSR_svc	SPSR_abt	SPSR_irq	SPSR_und

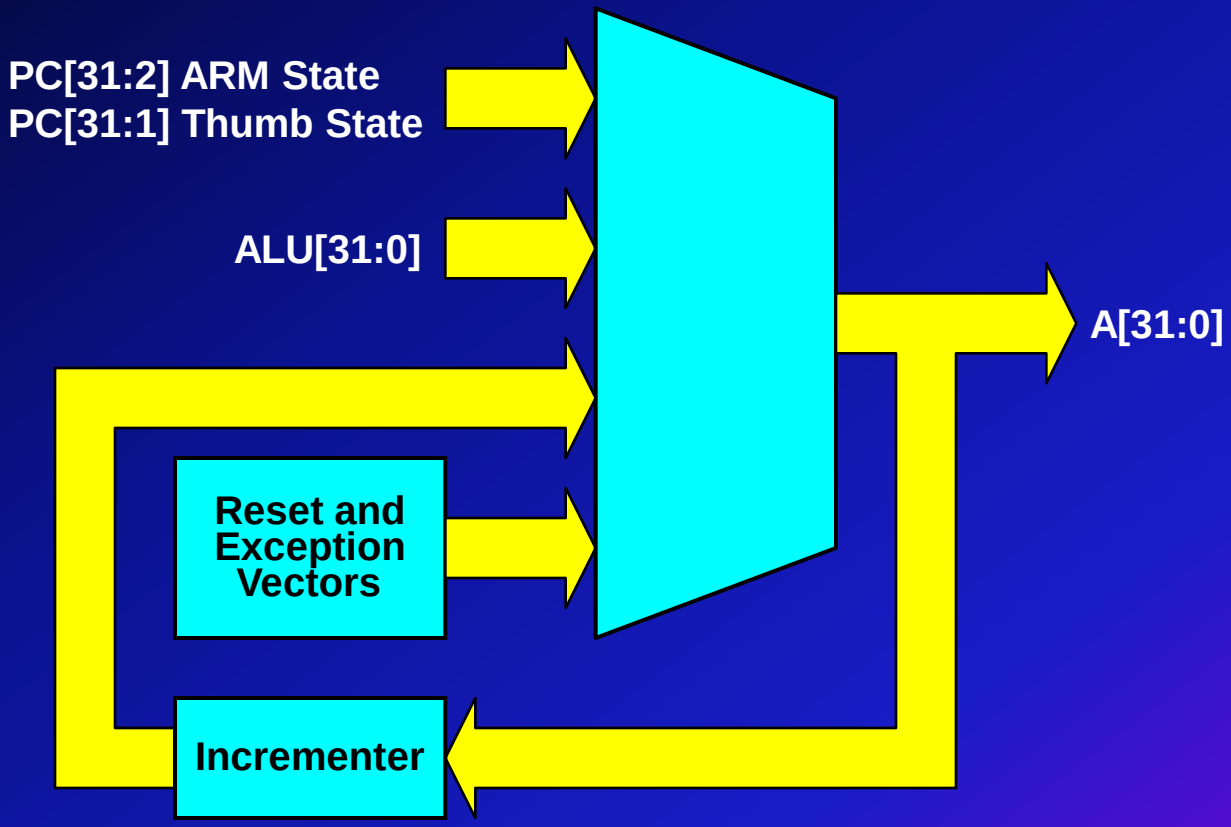
indicates a banked register.

Thumb State Registers

System & User	FIQ	Supervisor	Abort	IRQ	Undefined
R0	R0	R0	R0	R0	R0
R1	R1	R1	R1	R1	R1
R2	R2	R2	R2	R2	R2
R3	R3	R3	R3	R3	R3
R4	R4	R4	R4	R4	R4
R5	R5	R5	R5	R5	R5
R6	R6	R6	R6	R6	R6
R7	R7	R7	R7	R7	R7
SP	SP_fiq	SP_svc	SP_abt	SP_irq	SP_und
LR	LR_fiq	LR_svc	LR_abt	LR_irq	LR_und
PC	PC	PC	PC	PC	PC
	THUMB State Program Status Registers				
CPSR	CPSR	CPSR	CPSR	CPSR	CPSR
	SPSR_fiq	SPSR_svc	SPSR_abt	SPSR_irq	SPSR_und

indicates a banked register.

Program Counter and Address Generation



Triscend®

When an Exception Occurs

- **CPU copies CPSR into SPSR**
- **Sets appropriate CPSR bits**
  - If in Thumb mode, sets ARM state bit
  - Sets mode bits based on exception type
  - Disables interrupts, if appropriate
- **Stores the return address in LR_*mode***
- **Sets PC to vector address**

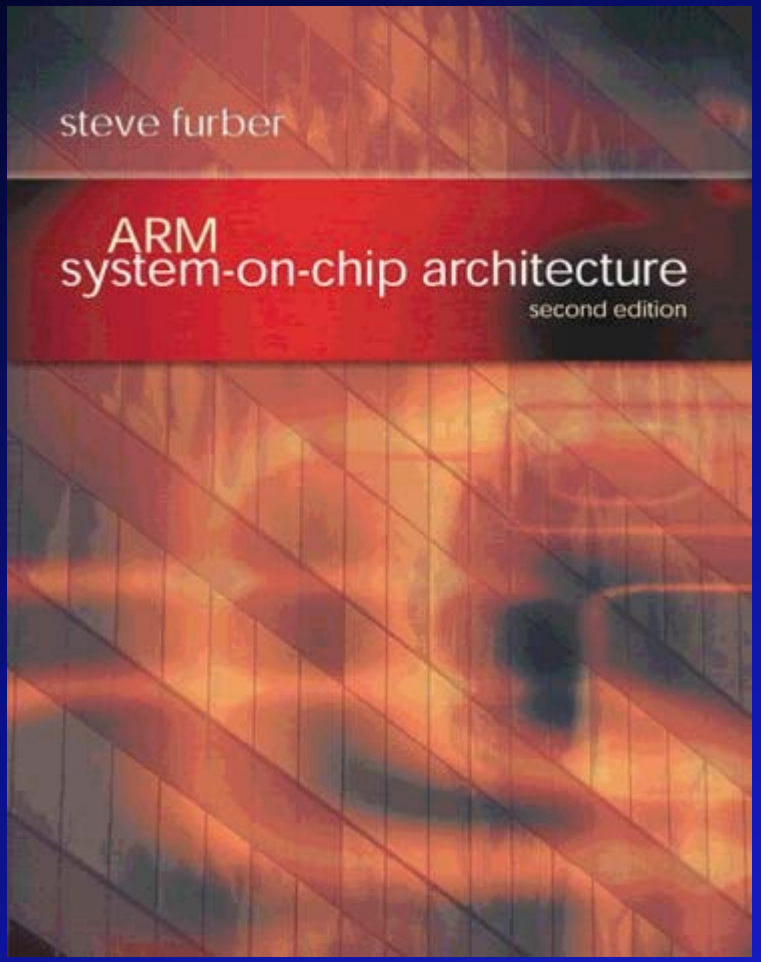
Returning from an Exception

- **Restore CPSR from SPSR_*mode***
  - Automatically restores original mode, state, interrupt settings, and condition codes
- **Restore PC using 'return address' stored in LR_*mode***
  - Execution continues from the place in code where the exception occurred
- **Return must be done from ARM state**
  - There is no Thumb instruction to copy SPSR back into CPSR

Exception Vector Table

Vector Address	Exception
0x1C	FIQ
0x18	IRQ
0x14	Reserved
0x10	Data Abort
0x0C	Prefetch Abort
0x08	Software Interrupt
0x04	Undefined Instruction
0x00	Reset

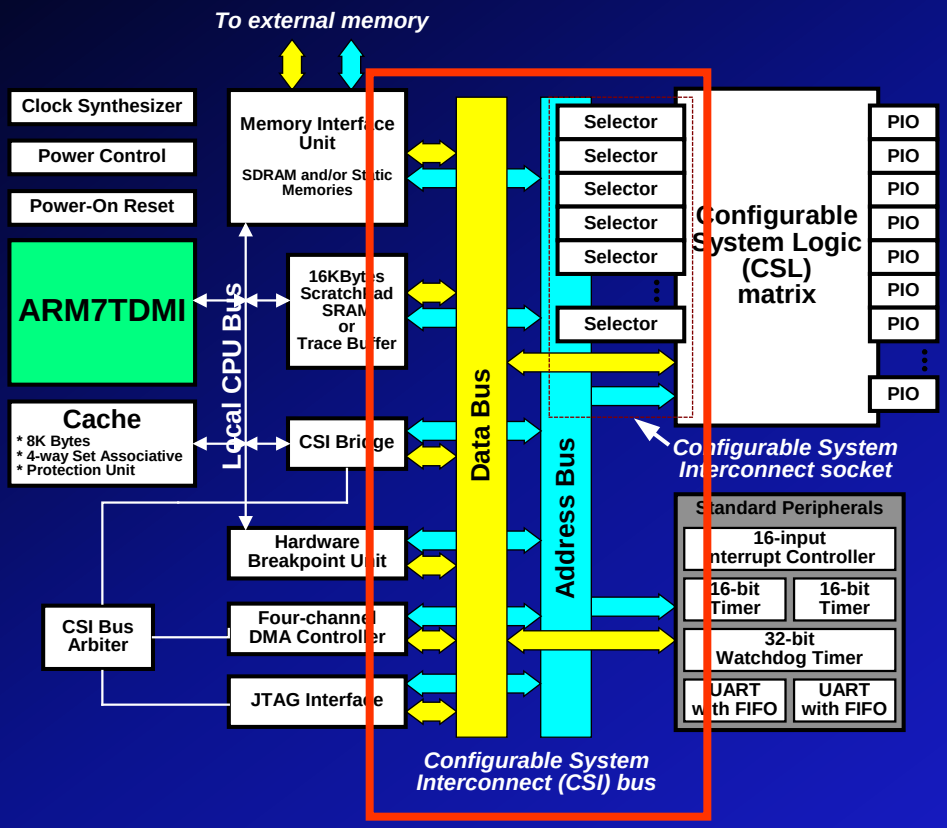
Learning More on ARM



- **ARM System-on-Chip Architecture**
  - Steve Furber
  - Addison Wesley
  - ISBN=0201675196
- **Available at Amazon.com**
  - Link in A7 Data Sheet

Triscend®

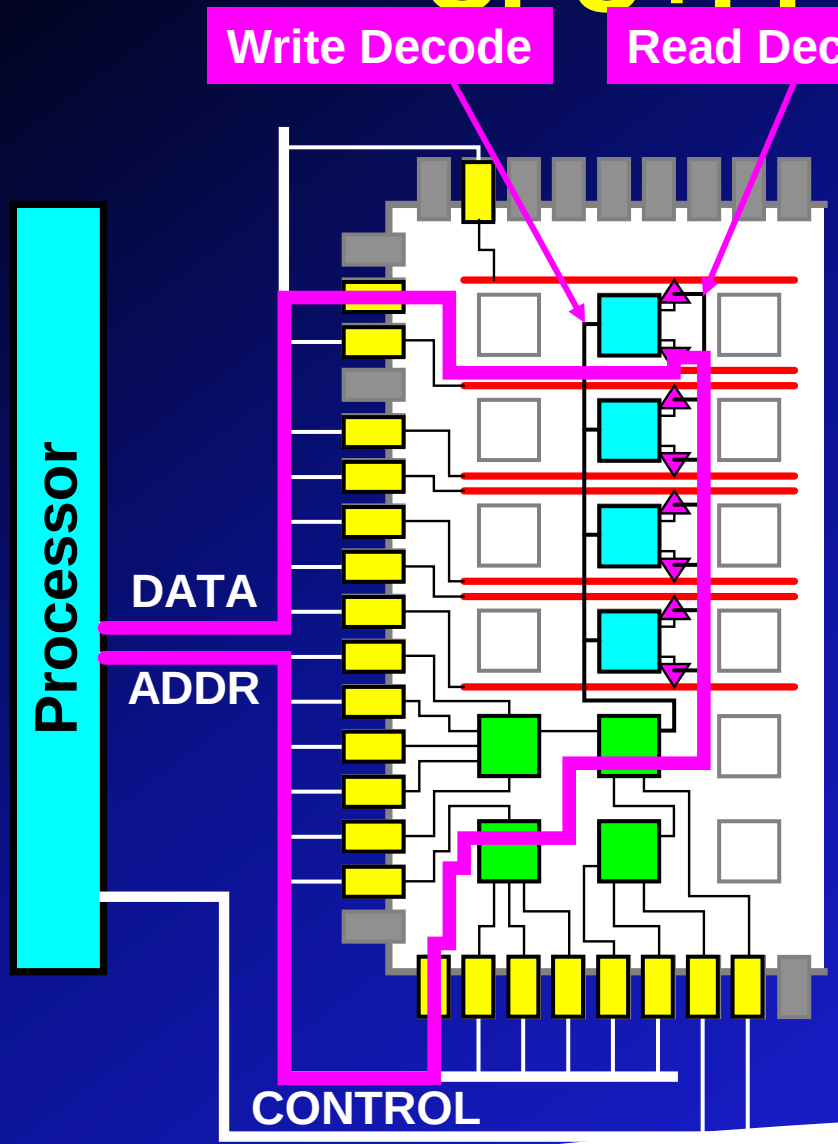
Configurable System Interconnect Bus



- 32-bit Read and Write data buses
- 32-bit Address bus
- Up to 240MB/sec transfer rate
- Separate, independent from CPU local bus
- Flexible address decoders
- Programmable wait-state support
- Compatible with all Triscend CSoC devices

Triscend®

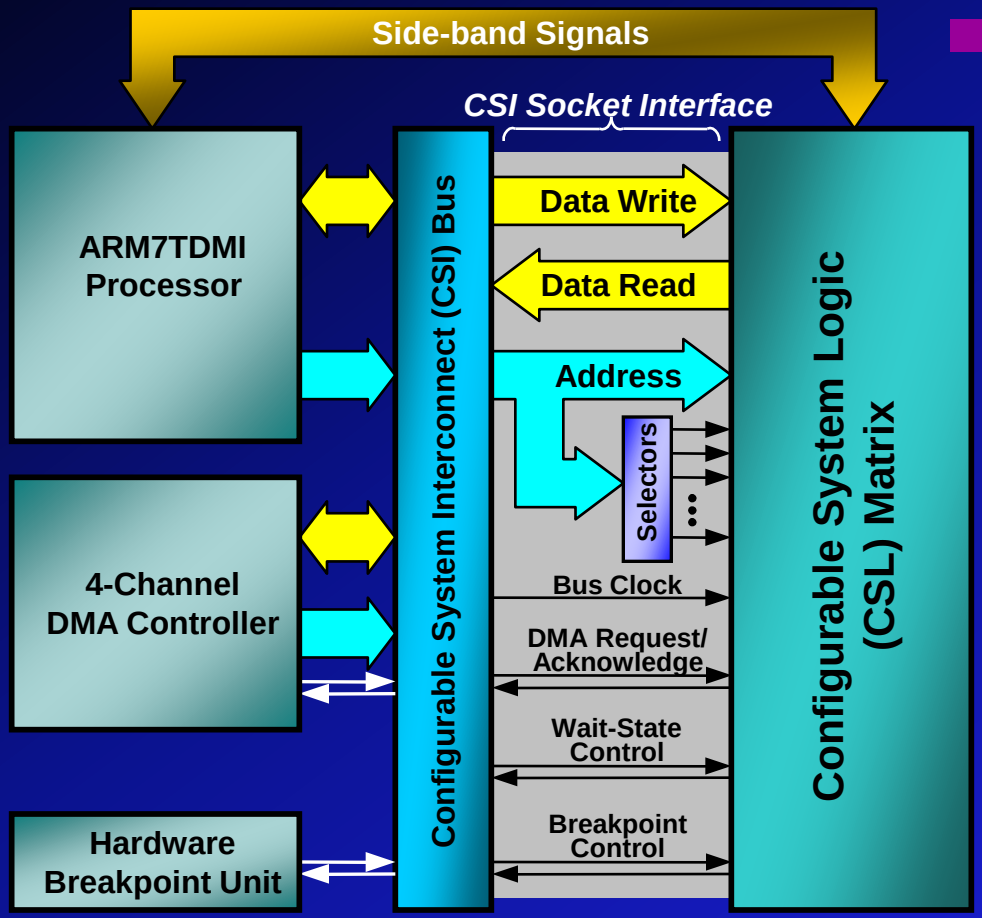
Routing Bus Signals: CPU+FPGA/ASIC



- **I/Os between devices**
  - Many I/Os required, up to 70+ for 32-bit data, address
  - 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

Triscend®

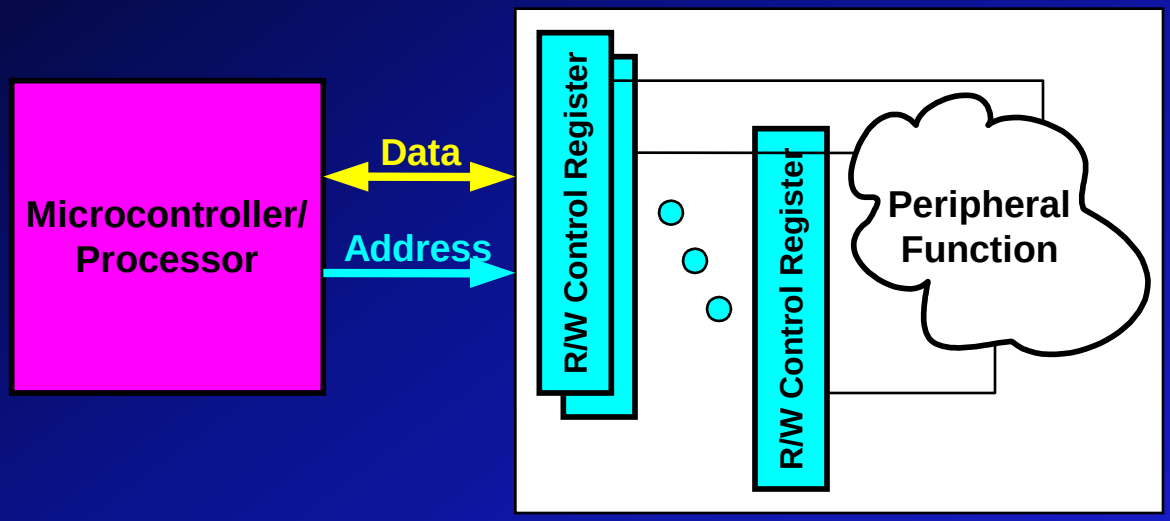
Triscend Solution: CSI Bus Socket
(Configurable System Interconnect)



- **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
  - Patented technology
  - Virtual sockets

Triscend®

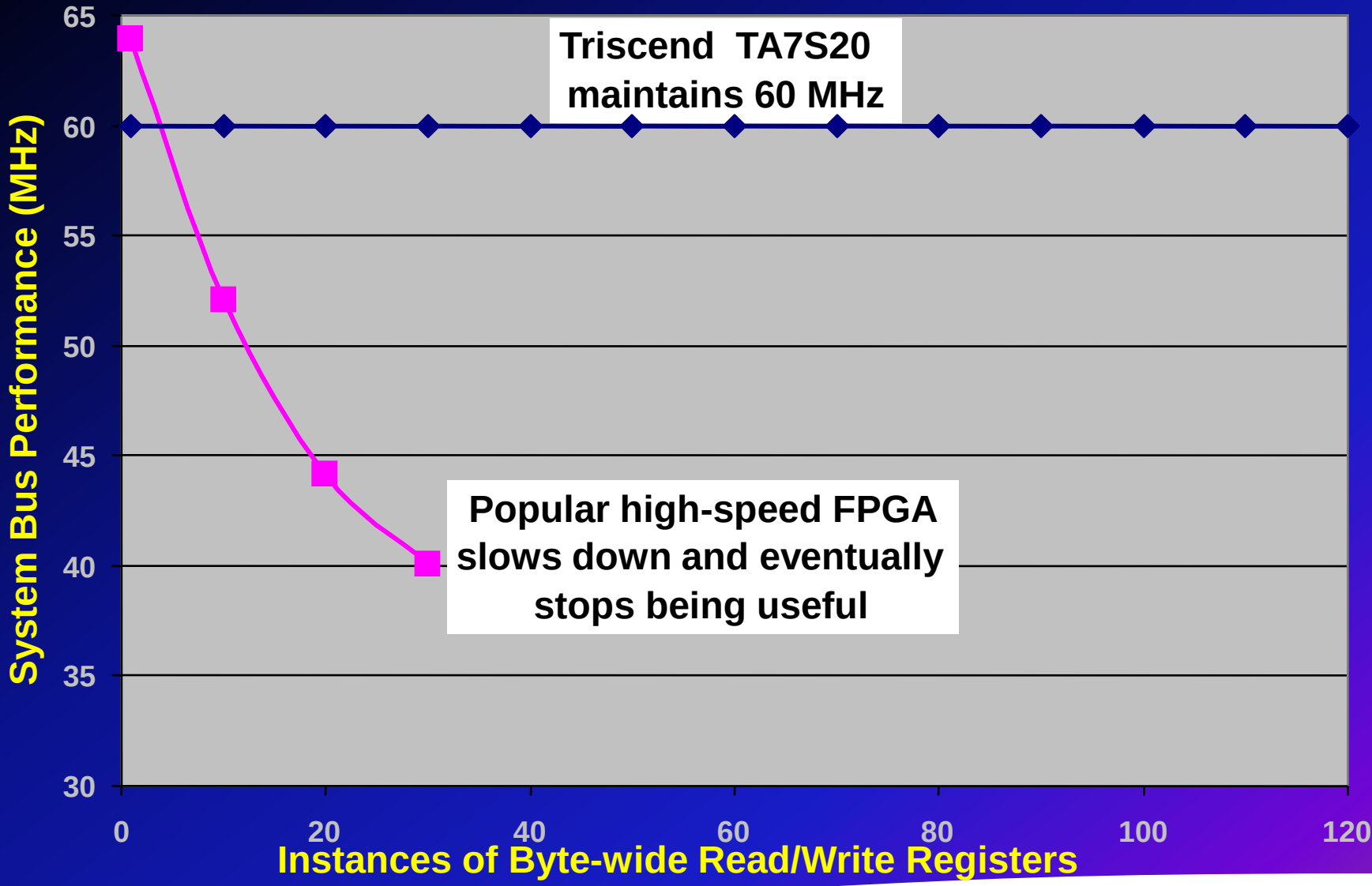
System-Level Performance Benchmark



- Read and write to registers controlling a peripheral function
- Requires bidirectional data bus, address decoding
- What happens to performance when you add more control registers?

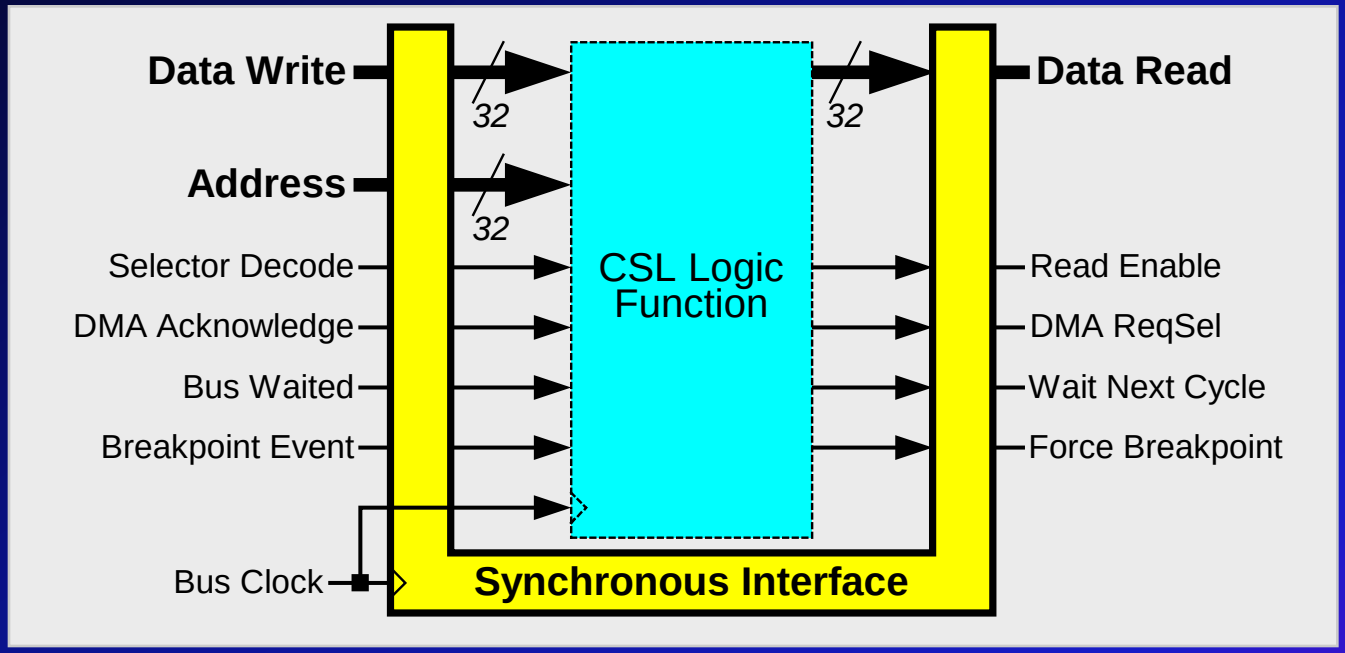
Triscend®

Triscend Wins the Bus Race!



Triscend®

CSI Bus Timing Model



- Simplifies logic design, synchronous interface
- Write data, address, control presented at rising clock edge
- Present read data, respond on next rising clock edge

Triscend®

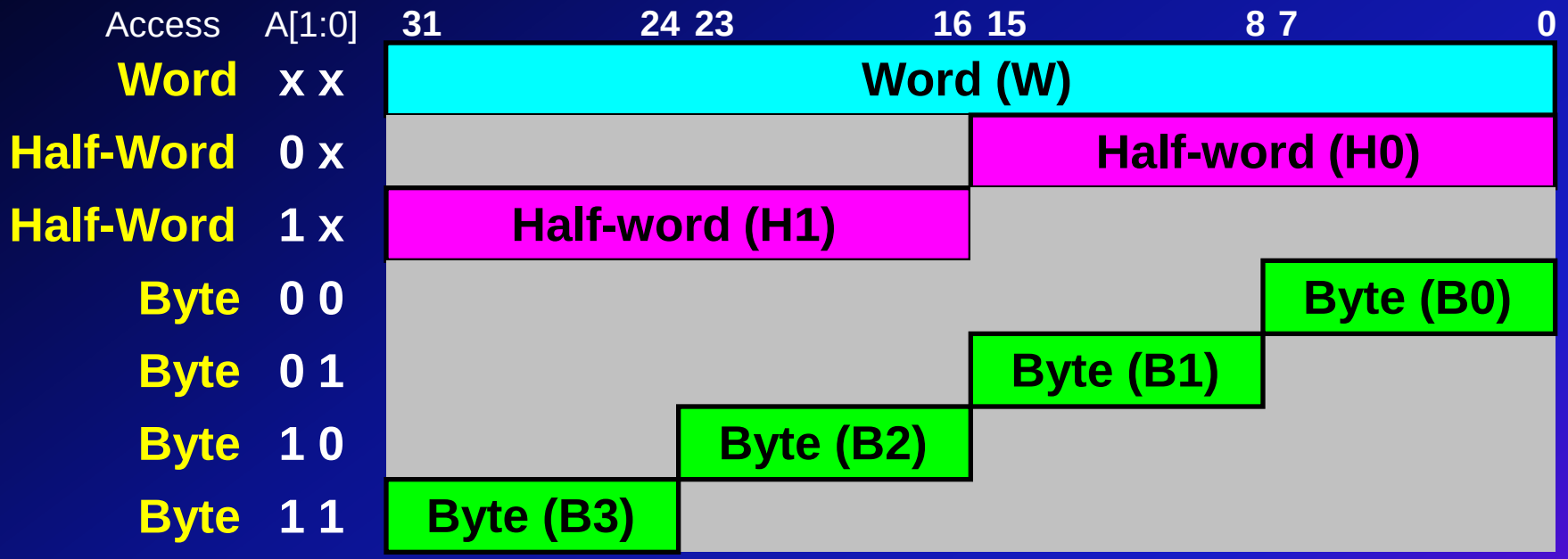
CSI Address Selectors

- **Distributed, integrated address decoders**
- **Up to 200 per device**
- **Multi-purpose**
  - SELECTOR - decode read and write
  - CHIPSEL - decode read and chip select
  - DMACTRL - distributed DMA control
- **Support word, half-word, byte modes**
- **Automatic address assignment in FastChip**

Device	Selectors
TA7S05	32
TA7S12	72
TA7S20	128
TA7S32	200

Triscend®

Data Types and Byte Lanes

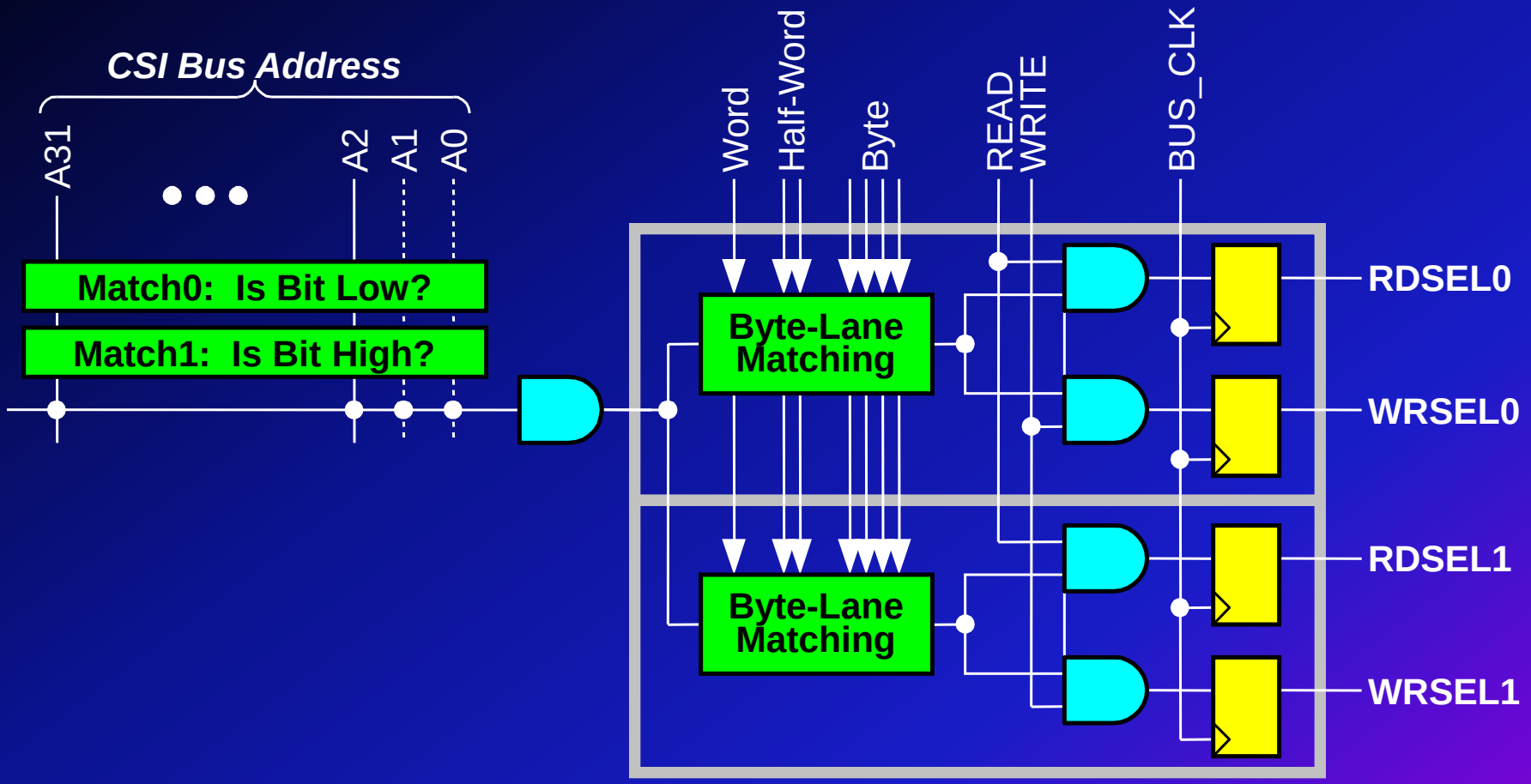


Triscend®

Write Data Duplication

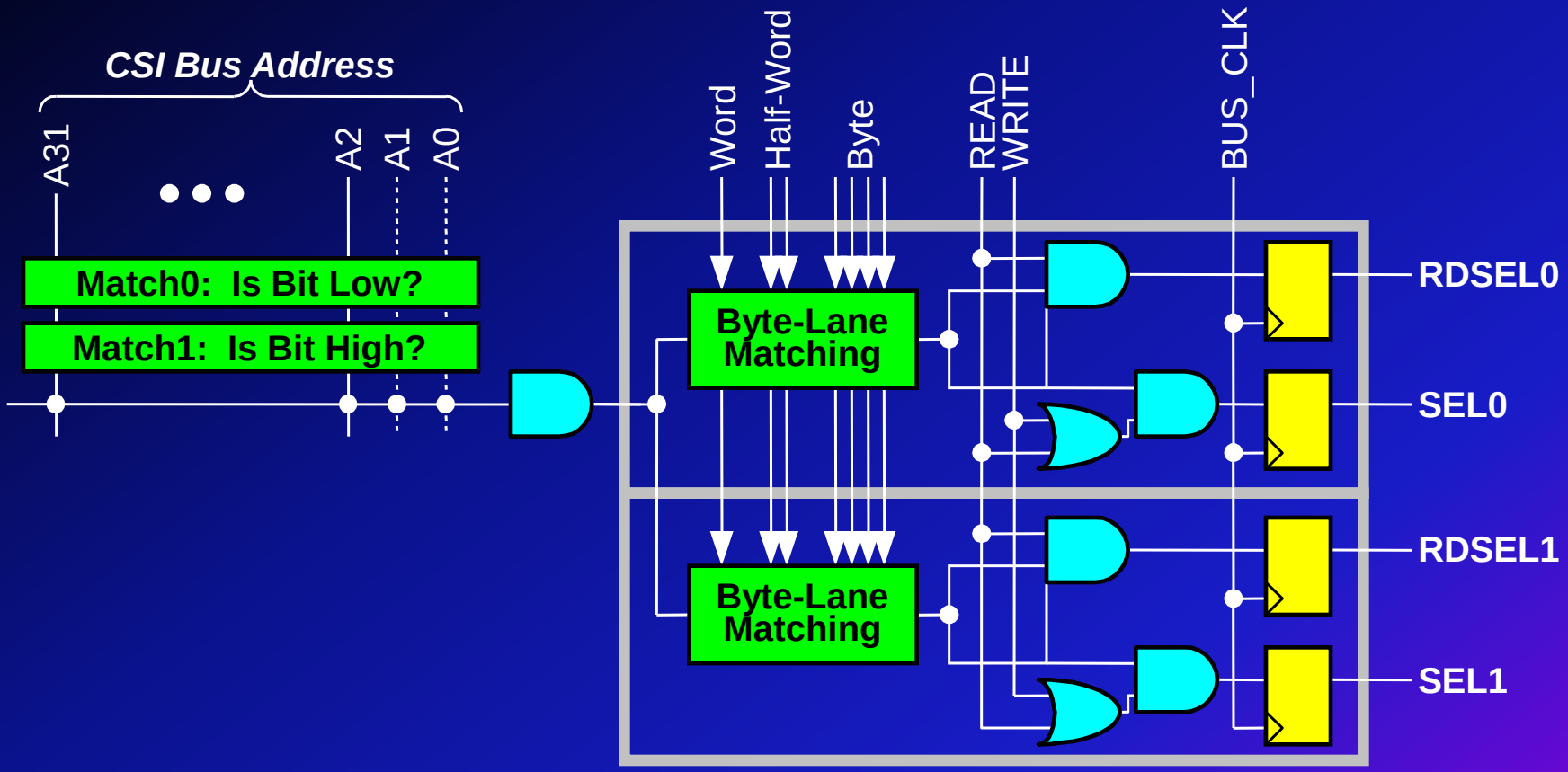
Access	A[1:0]	31 … 24	23 … 16	15 … 8	7 … 0
Word	0 0	D[31:0]			
Half-Word	0 0	D[15:0]		D[15:0]	
Half-Word	1 0	D[31:16]		D[31:16]	
Byte	0 0	D[7:0]	D[7:0]	D[7:0]	D[7:0]
Byte	0 1	D[15:8]	D[15:8]	D[15:8]	D[15:8]
Byte	1 0	D[23:16]	D[23:16]	D[23:16]	D[23:16]
Byte	1 1	D[31:24]	D[31:24]	D[31:24]	D[31:24]

Selector



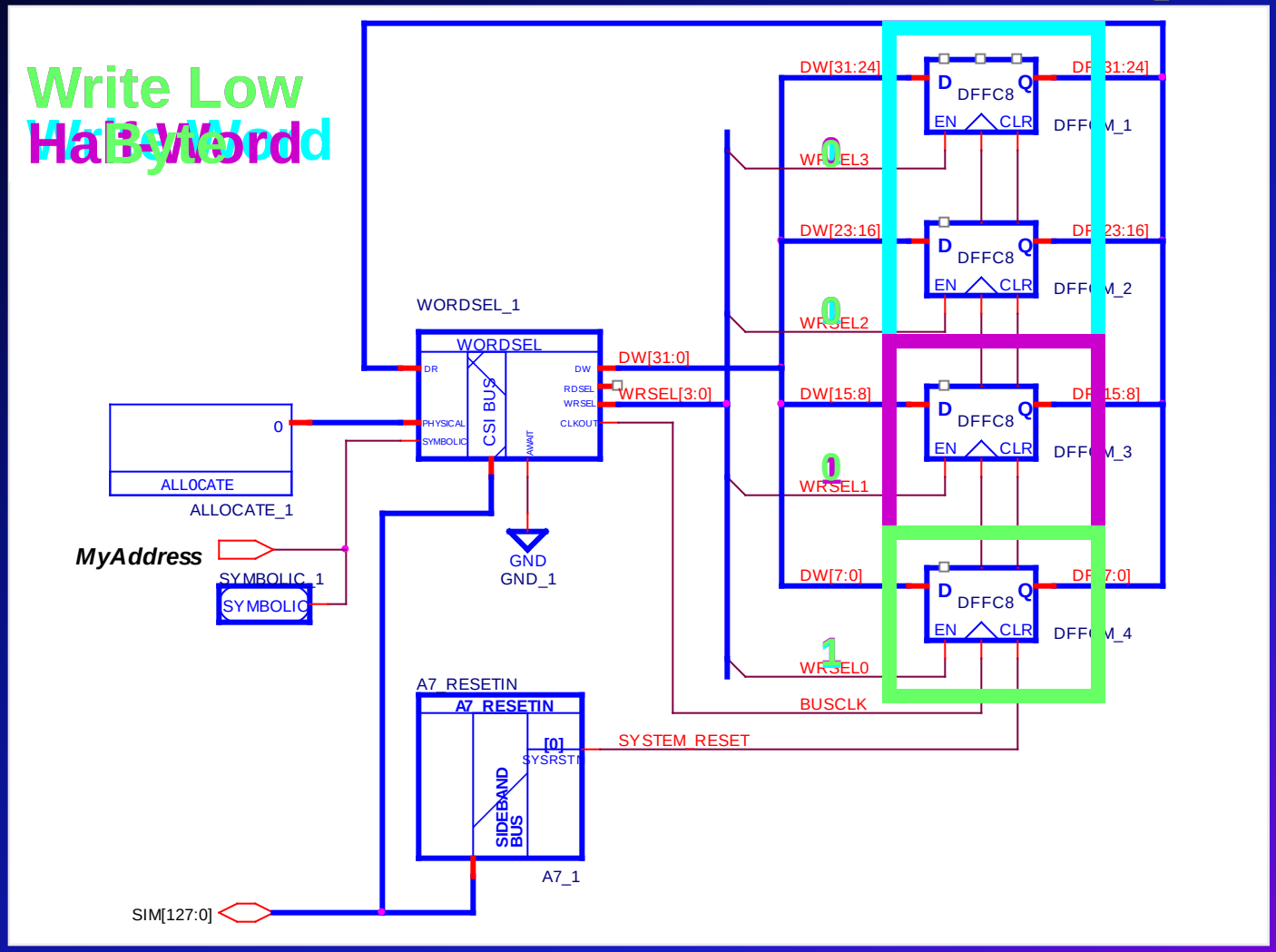
Triscend®

Chip Select Mode



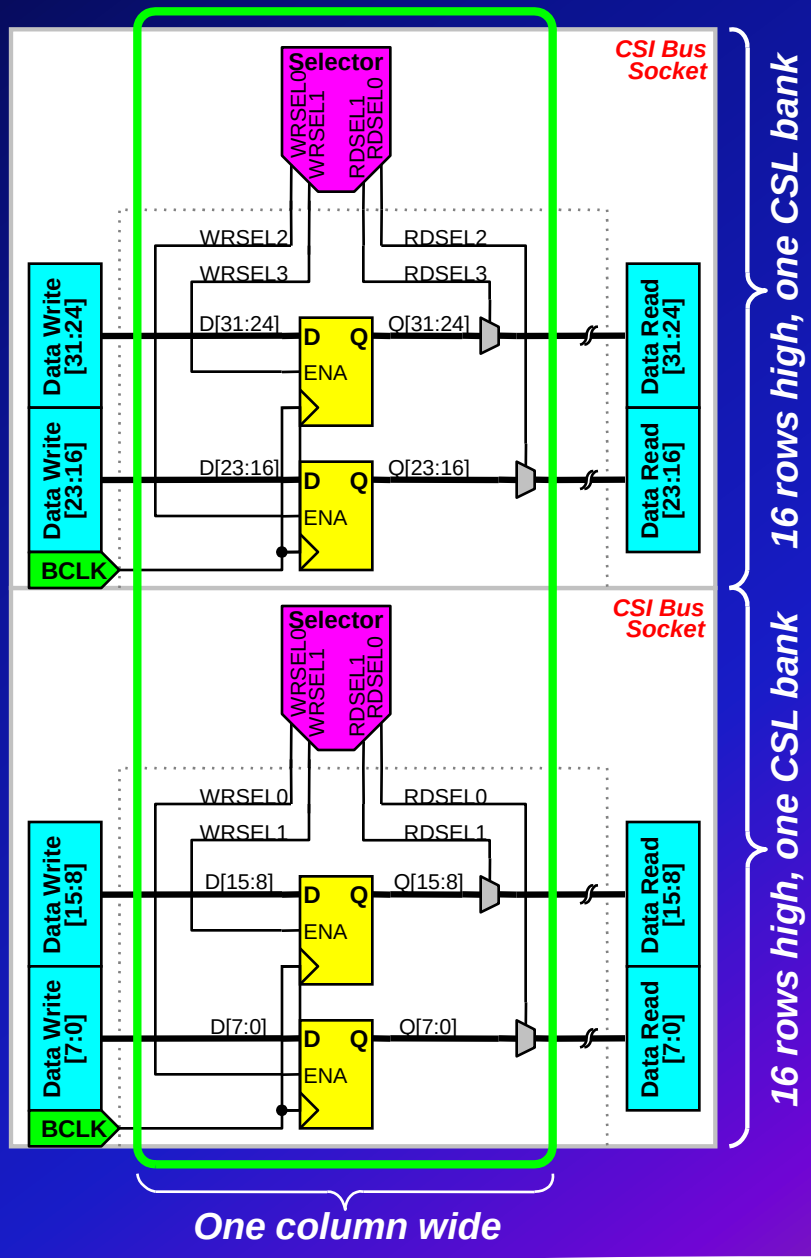
Triscend®

CSI Bus Interface Example



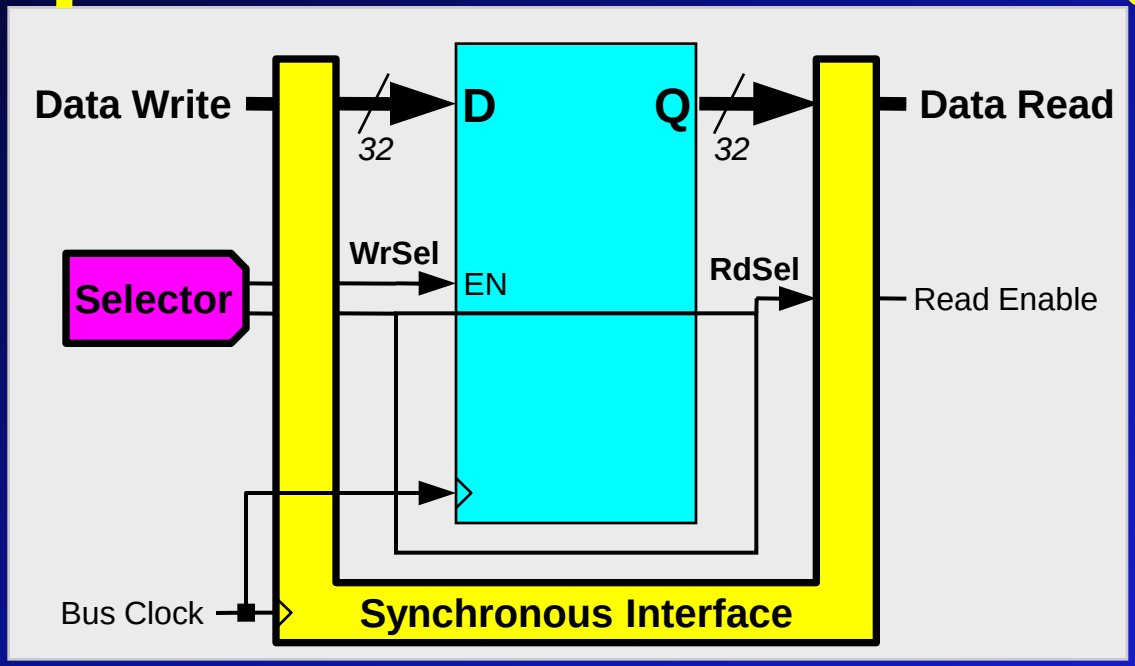
Triscend®

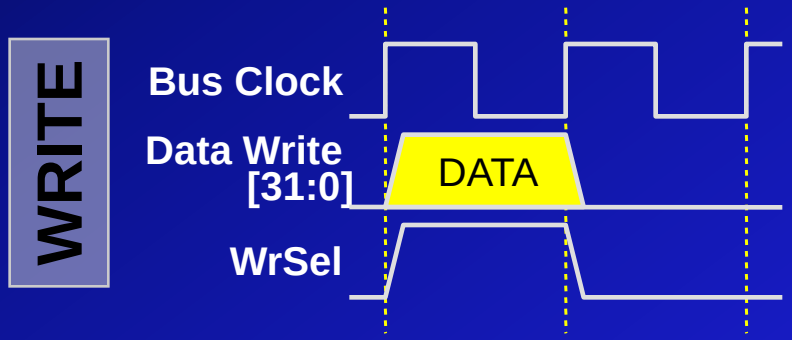
CSI Bus Implementation

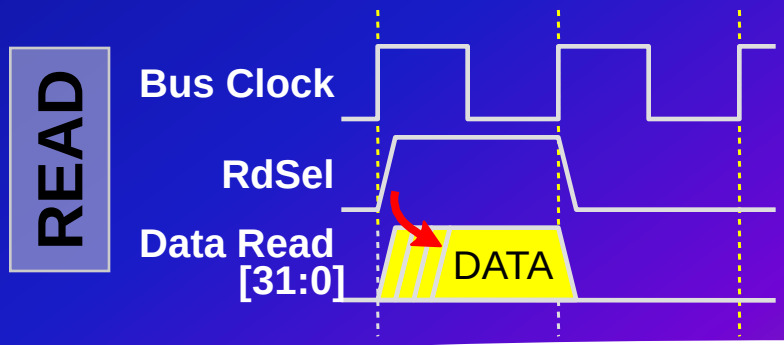


Triscend®

Example: Read/Write Register





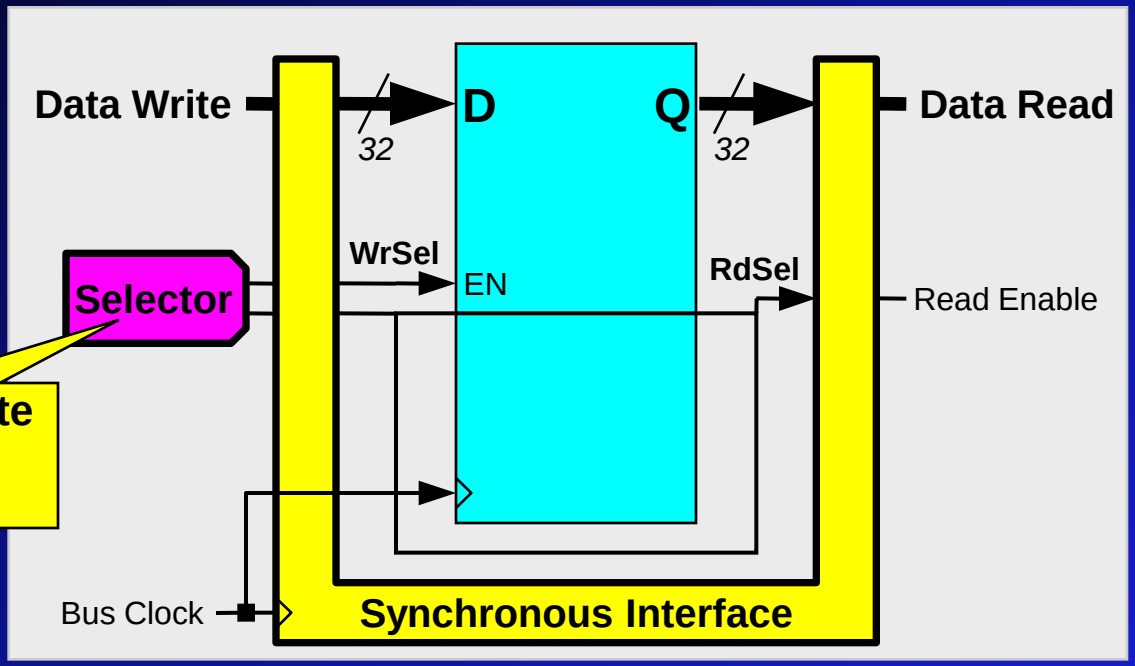


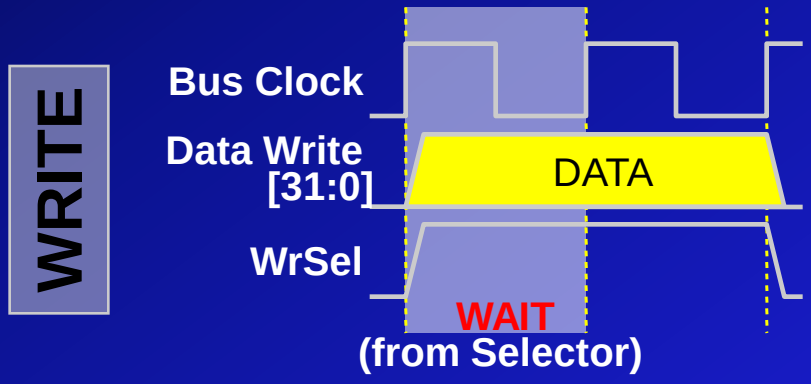
Triscend®

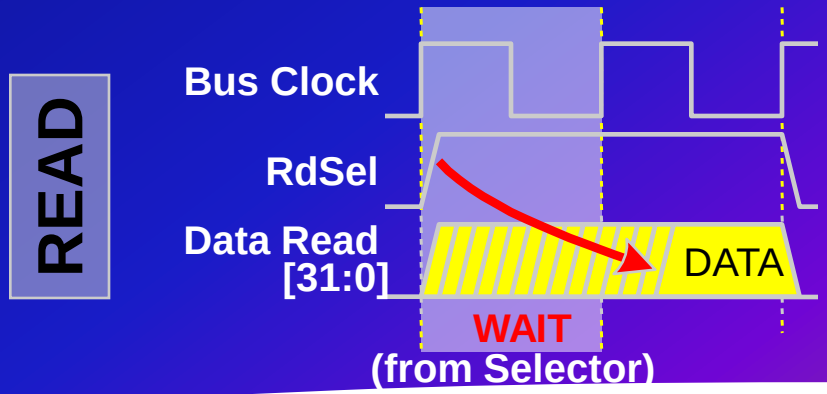
Wait-State Rules

- The CSI Bus supports wait-states, but with a few restrictions
- If a peripheral asserts a wait-state, the first wait-state is asserted by the Selector
- If either read or write transaction requires a wait-state, both must have at least one wait-state
- There are no wait-states allowed on device-side DMA transfers

With One Wait-State



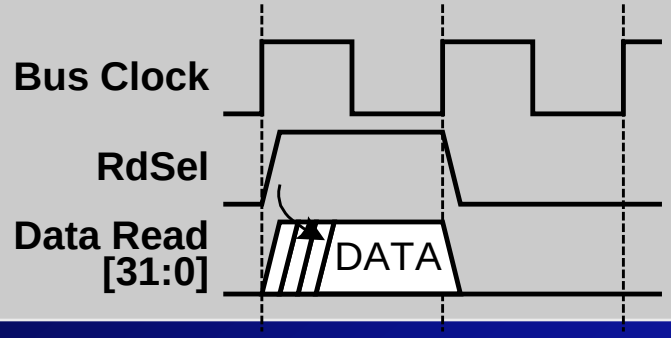




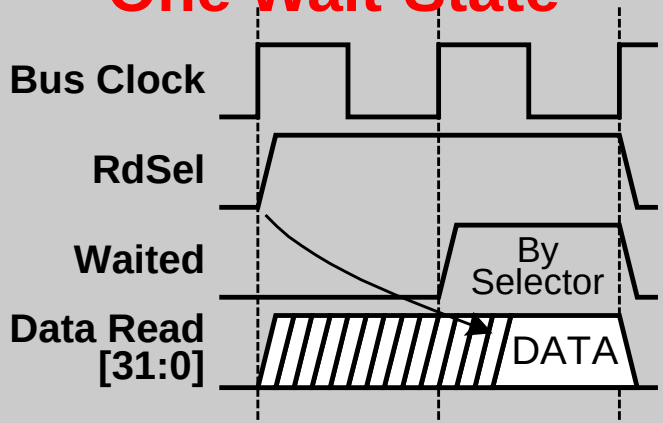
Triscend®

Wait-State Examples

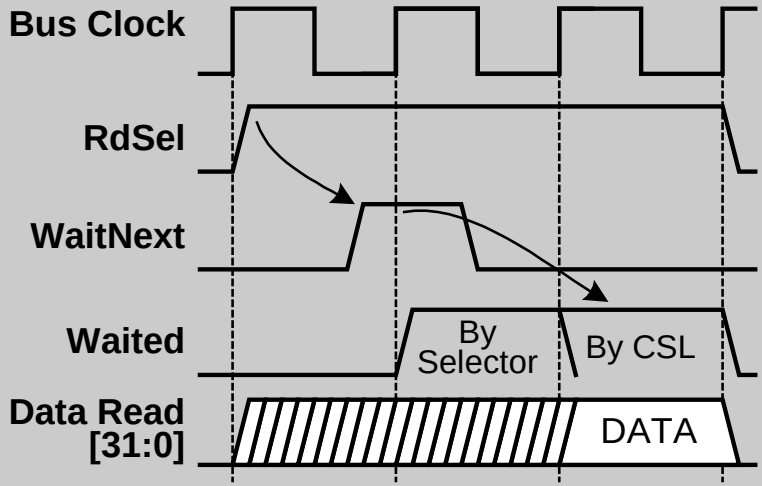
Zero Wait-States



One Wait-State



Multiple Wait-States

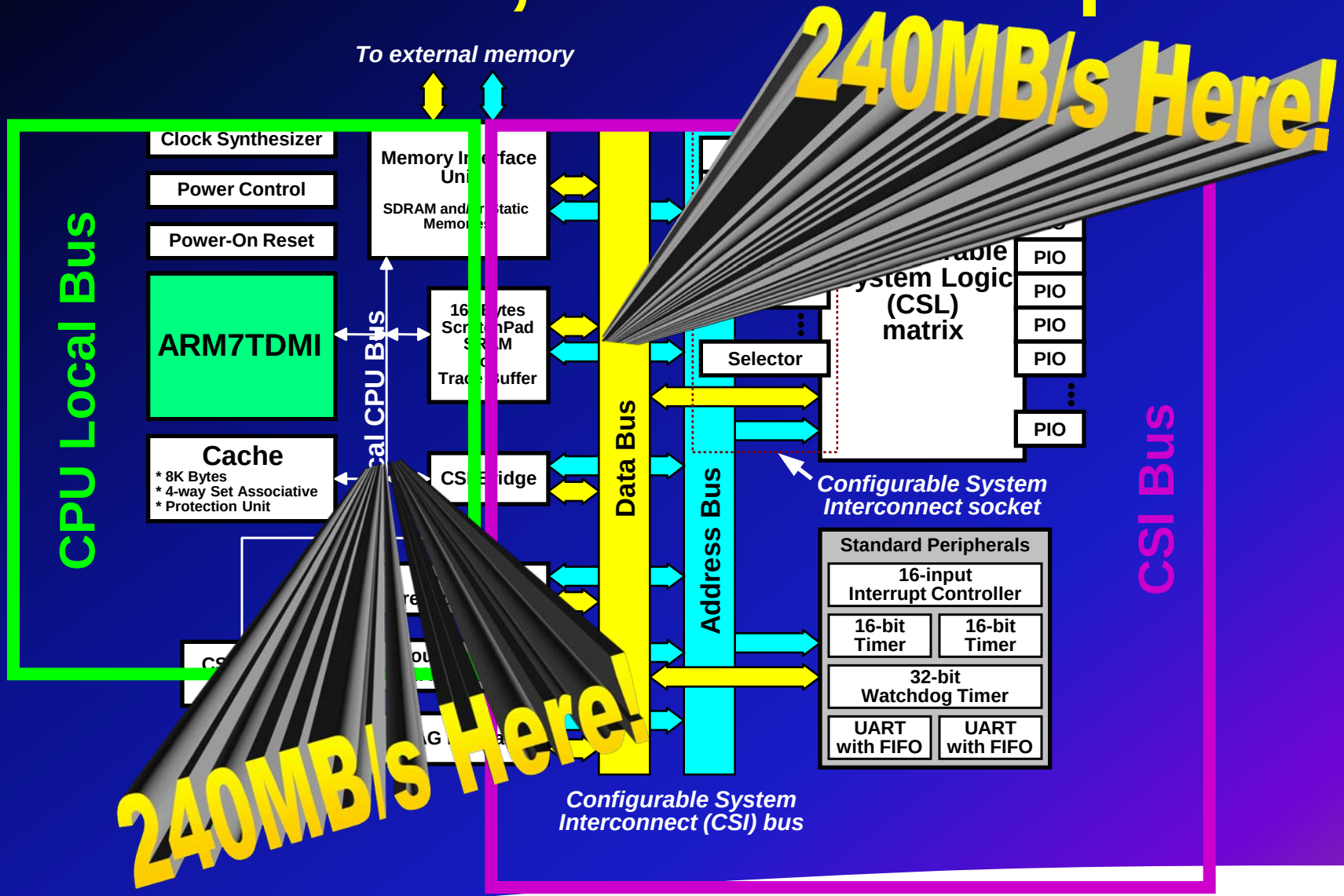


Triscend®

Sideband Signals

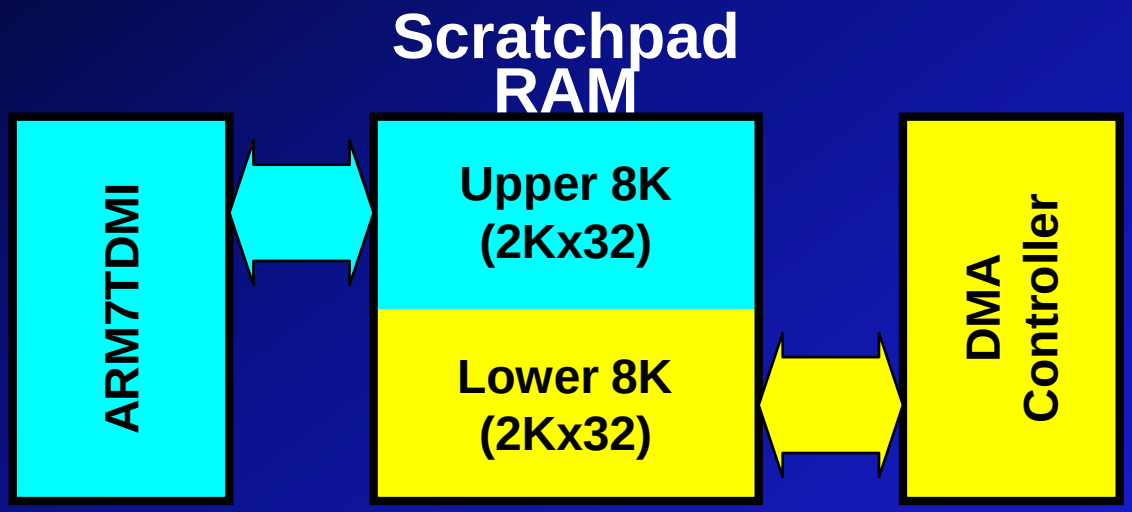
- Asynchronous signals connecting to CPU or dedicated peripherals
- Not part of the CSI bus, most are family-specific
- For the A7, the sideband signals are:
  - Four interrupts (FIQ, IRQ2, IRQ1, IRQ0)
  - Modem interface (DTR, RTS, CTS, DSR, DCD, RI)
  - UART interface (SIN1, SIN2, SOUT2, SOUT2, BDCLK1, BDCLK2)
  - Resets control (AppRst, SysRstN)
  - PLL lock signal (plock)
  - Alternate CSL clock (Aclk)

Two Busses, Increased Speed



Triscend®

Two Busses, No Waiting



CPU Local Bus

- 60 MHz
- 32-bit transfers
- 0 wait-states

CSI Bus

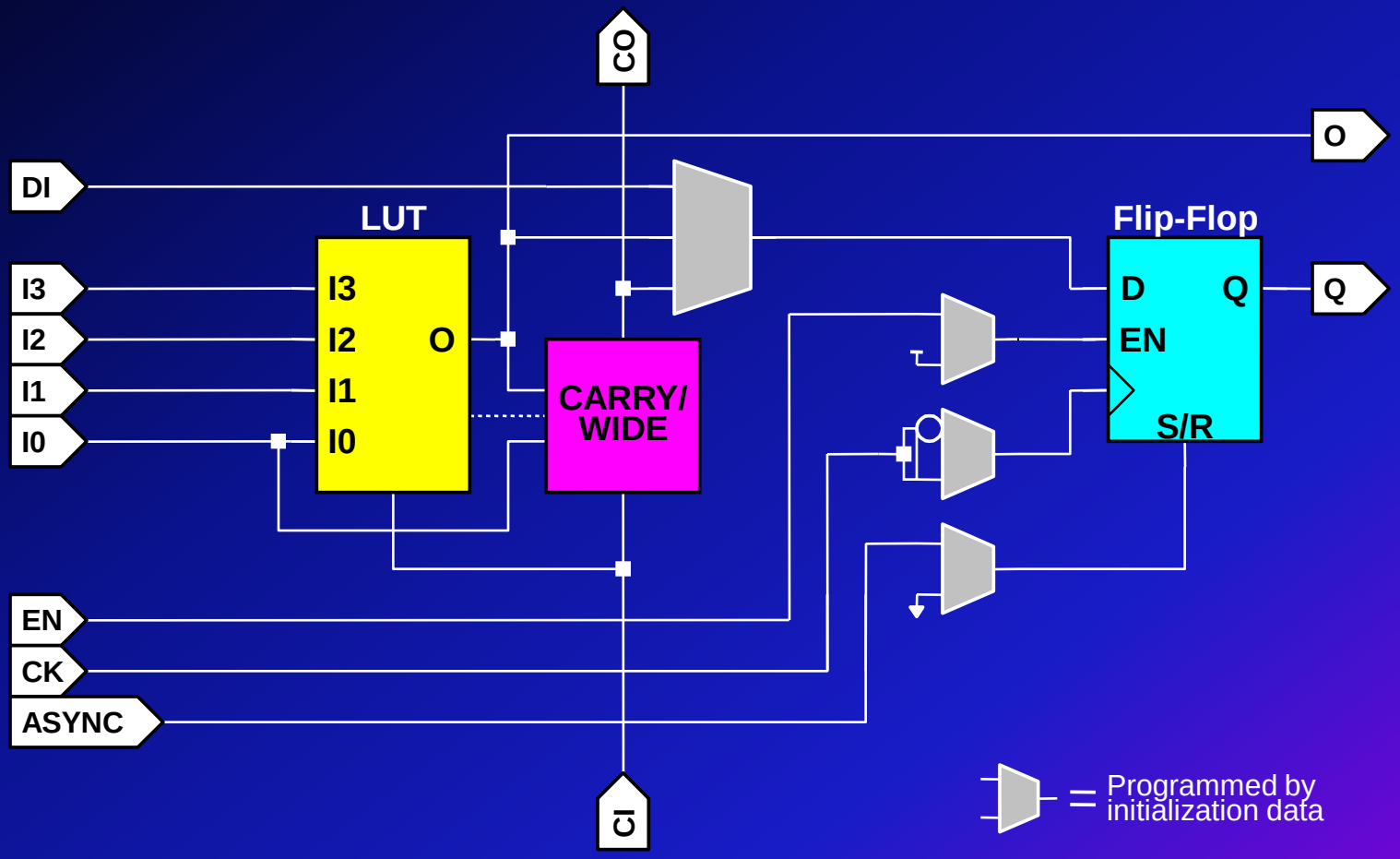
- 60 MHz
- 32-bit transfers
- 0 wait-states

Triscend®

Configurable System Logic (CSL)

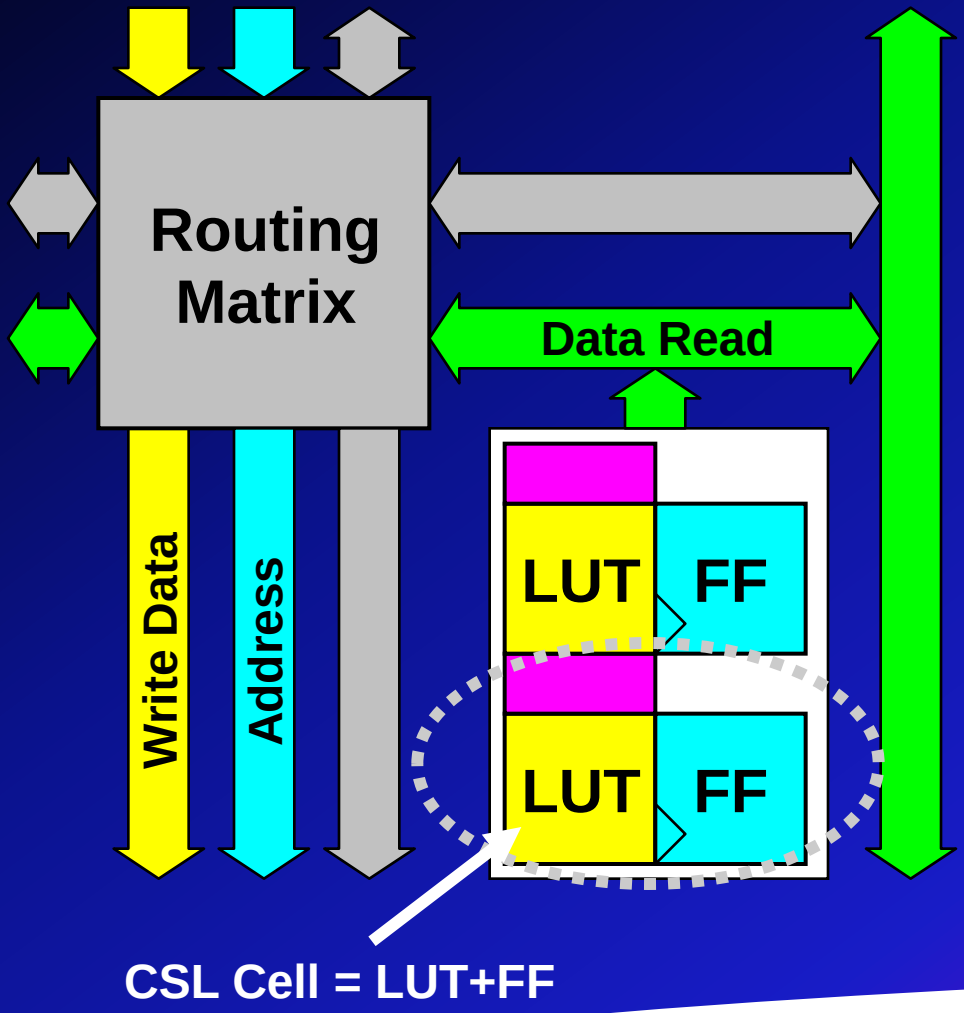
- **Extends the capabilities of the MCU**
- **Create custom-tailored peripherals**
- **SRAM-based programmable logic structure**
  - Popular LUT-style logic
  - Register-rich, very flexible
  - Programmable interconnect
- **Intimate access to and from the Configurable System Interconnect (CSI) bus**

CSL Cell



Triscend®

CSL Cell Structure



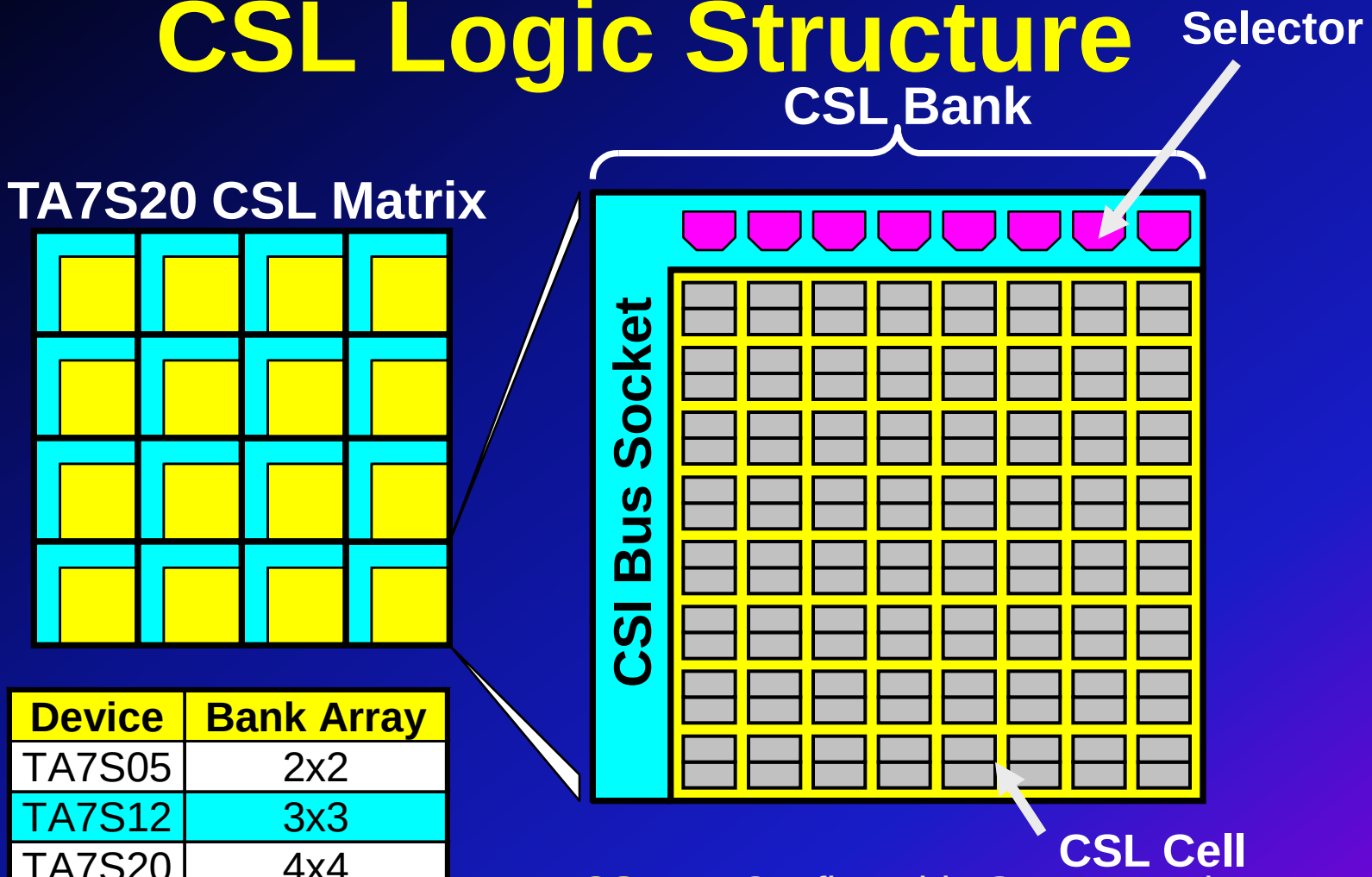
CSL cell perform various functions

- **L**ogic
- **A**rithmetic
- **M**emory
- **B**us
- **S**equential

Intimate connection to the CSI system bus

Triscend®

CSL Logic Structure

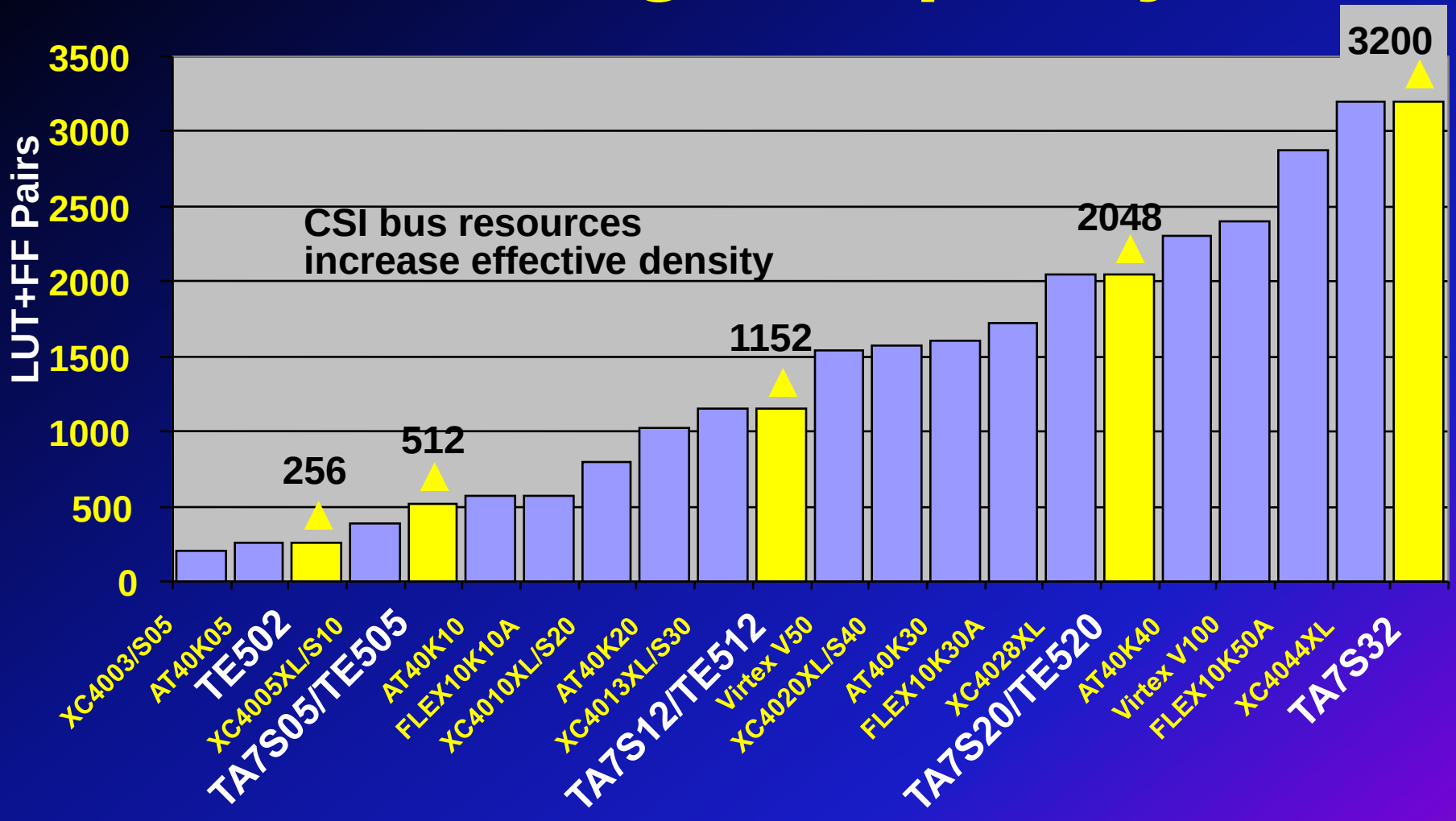


Device	Bank Array
TA7S05	2x2
TA7S12	3x3
TA7S20	4x4
TA7S32	5x5

- **CSL** = Configurable System Logic
- **CSI** = Configurable System Interconnect

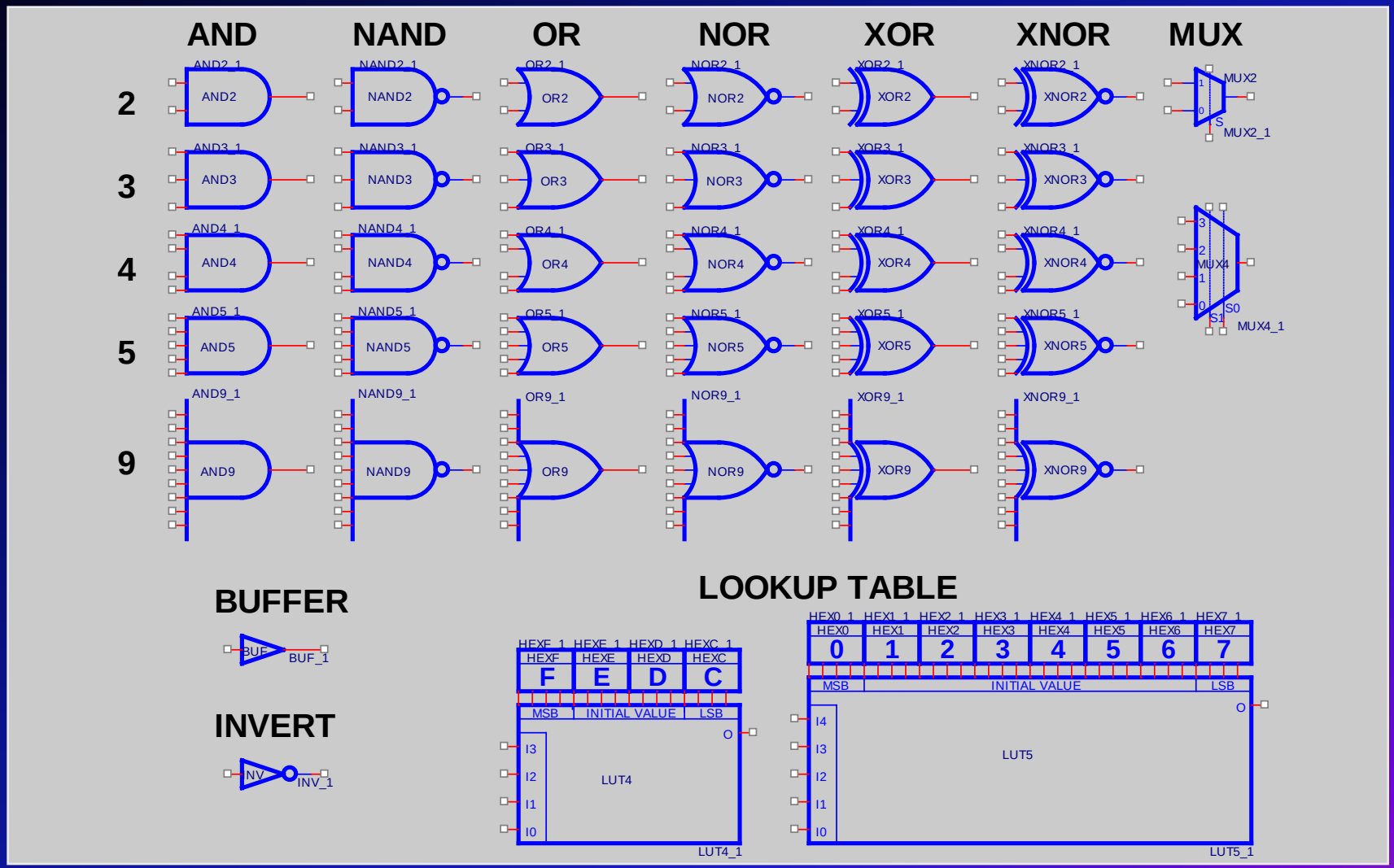
Triscend®

User Logic Capacity



Triscend®

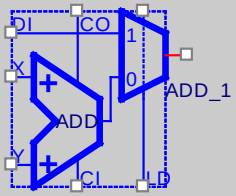
CSL Primitives (Logic)



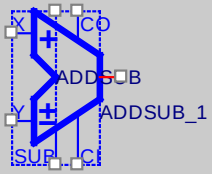
Triscend®

CSL Primitives (Arithmetic)

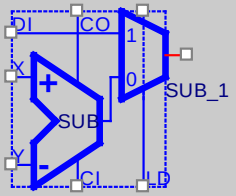
ADD with LOAD



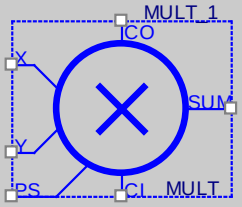
ADD/SUBTRACT



SUBTRACT with LOAD



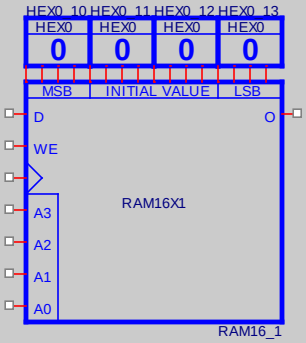
MULTIPLY



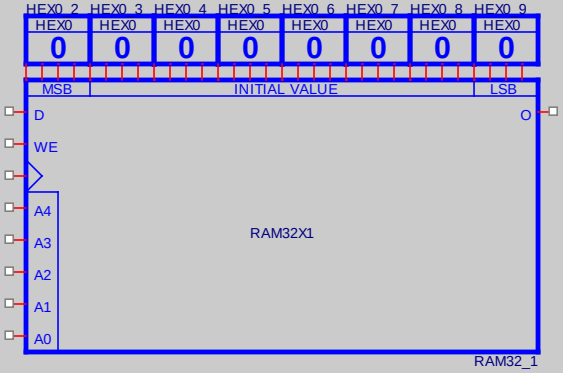
Triscend®

CSL Primitives (Memory)

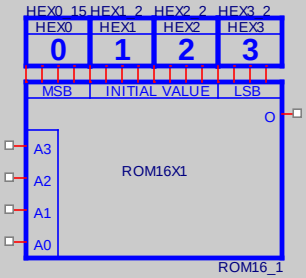
16x1 RAM



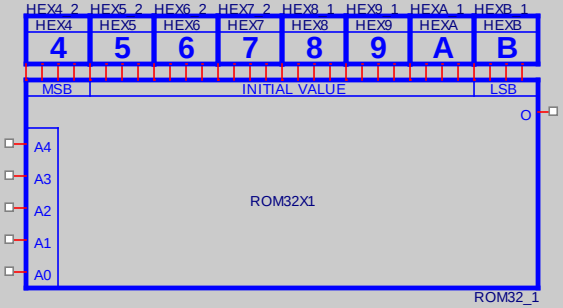
32x1 RAM



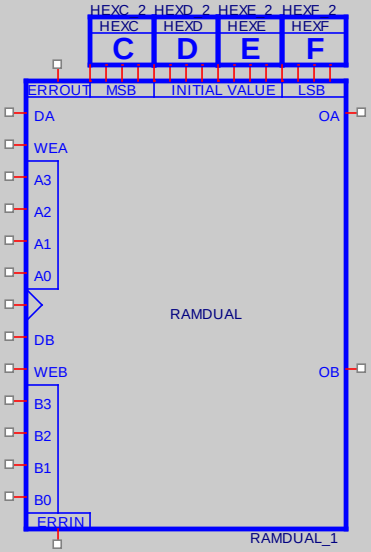
16x1 ROM



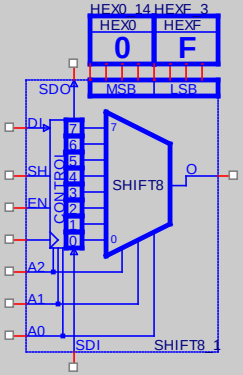
32x1 ROM



16x1 Dual-Port RAM

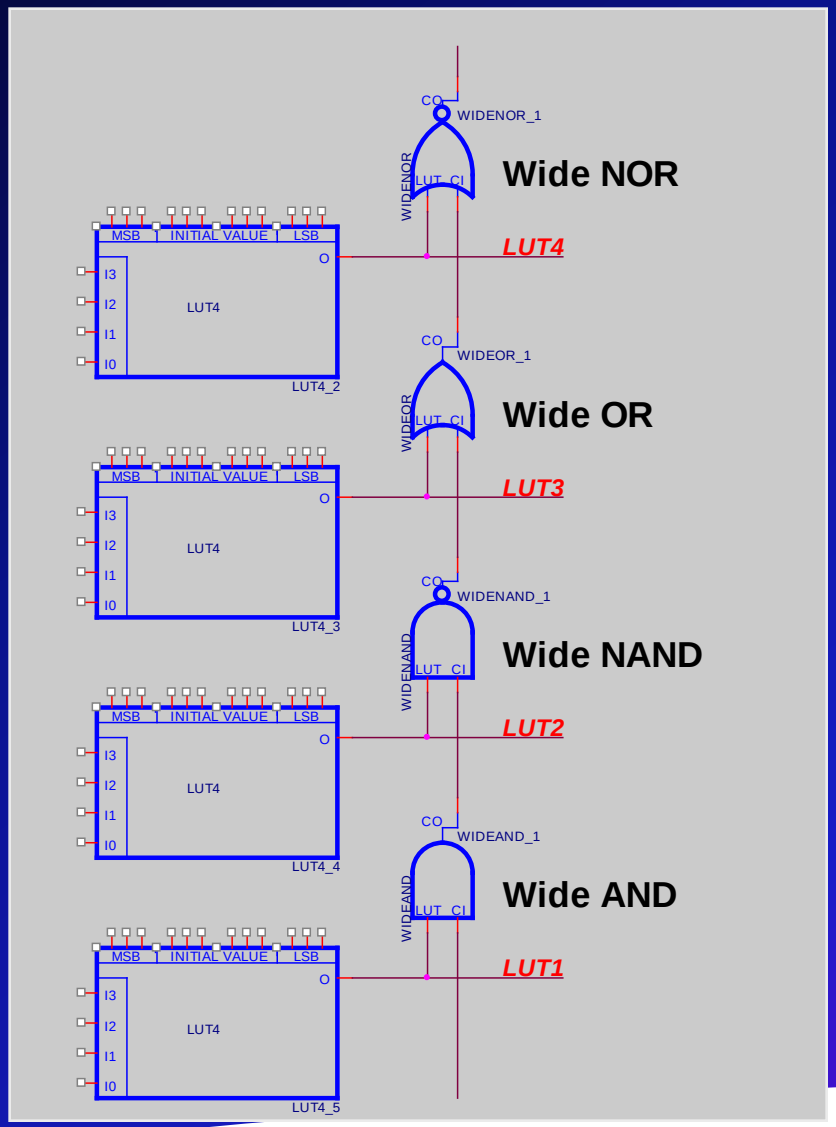


8-bit Serial-In/Serial-Out Shift Register



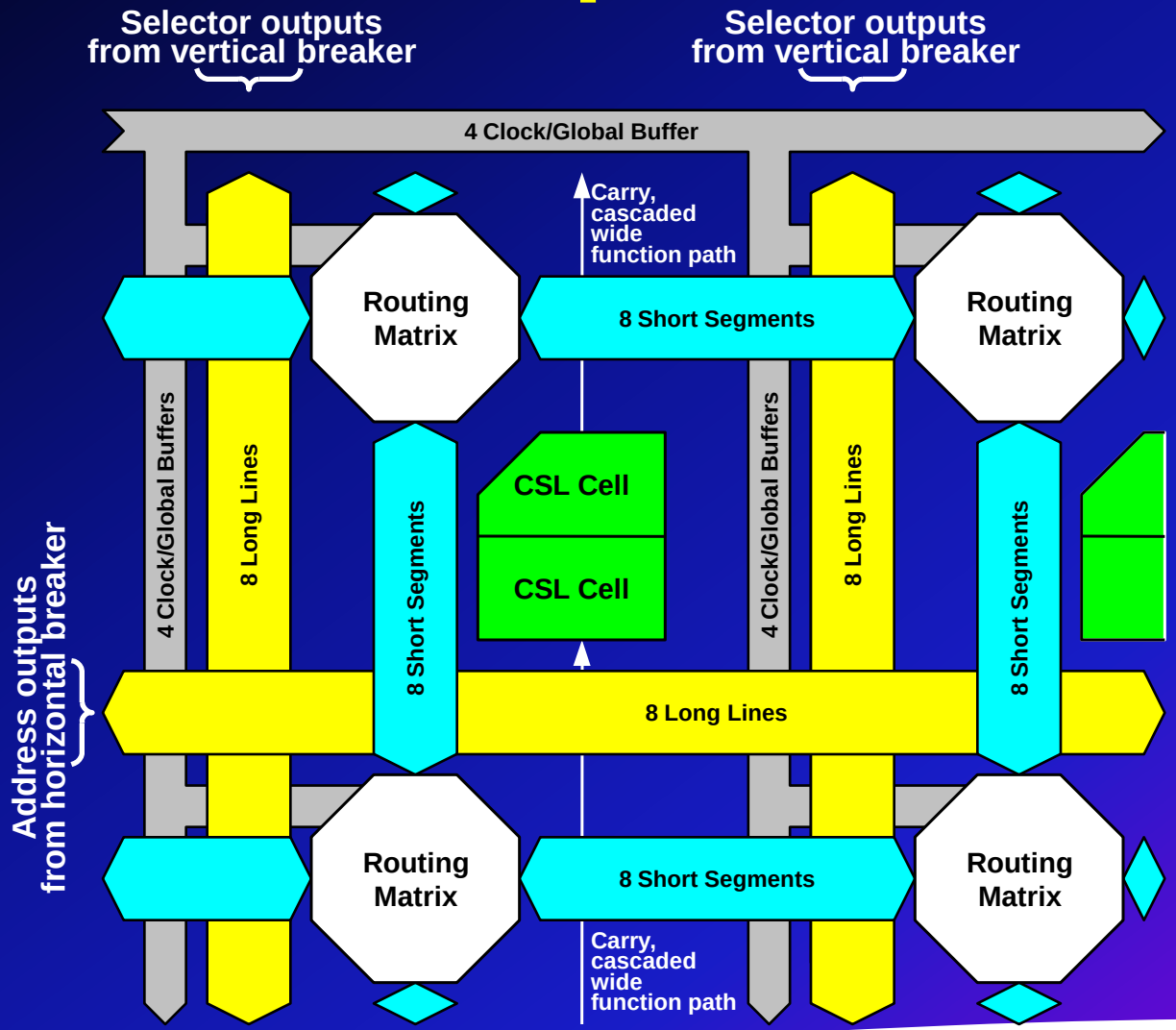
Triscend®

CSL Primitives (Wide)



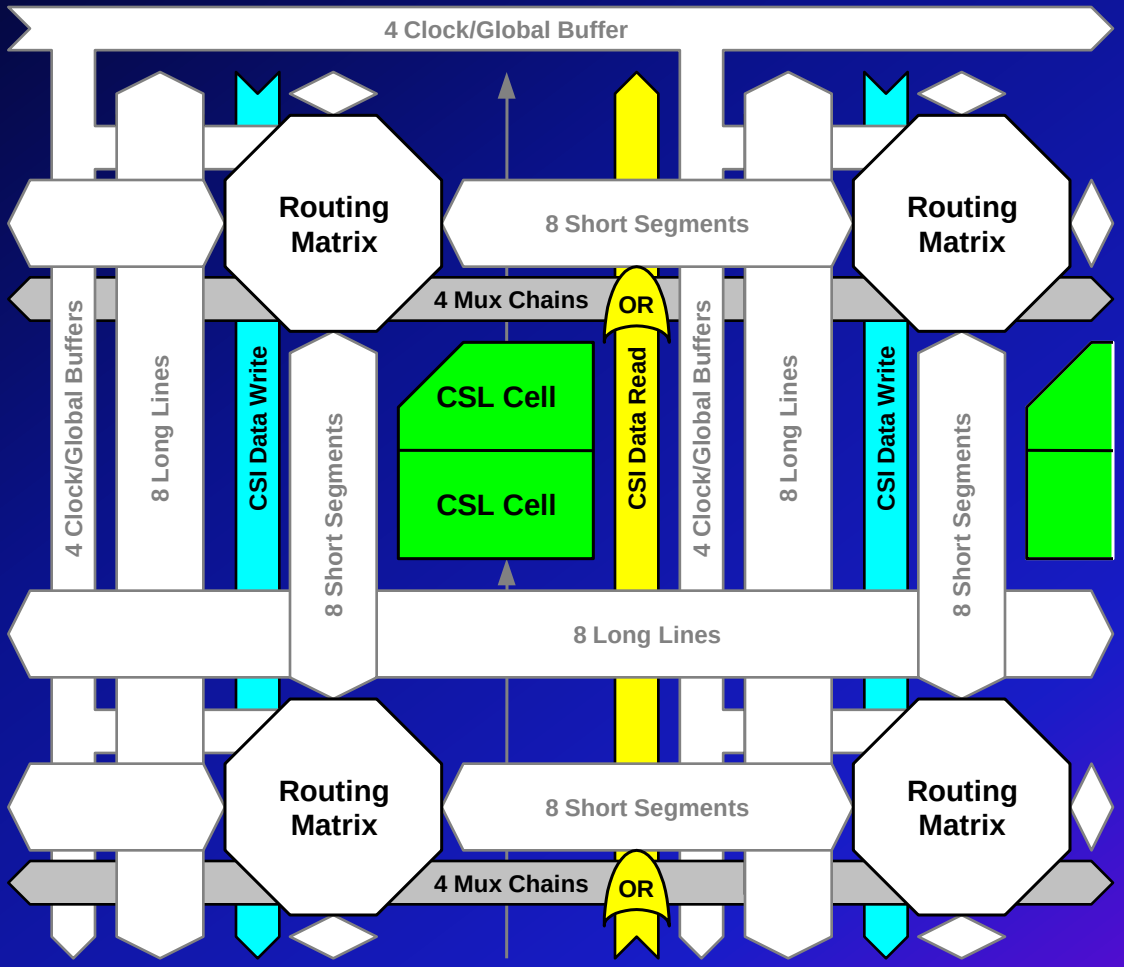
Triscend®

General-Purpose Routing



Triscend®

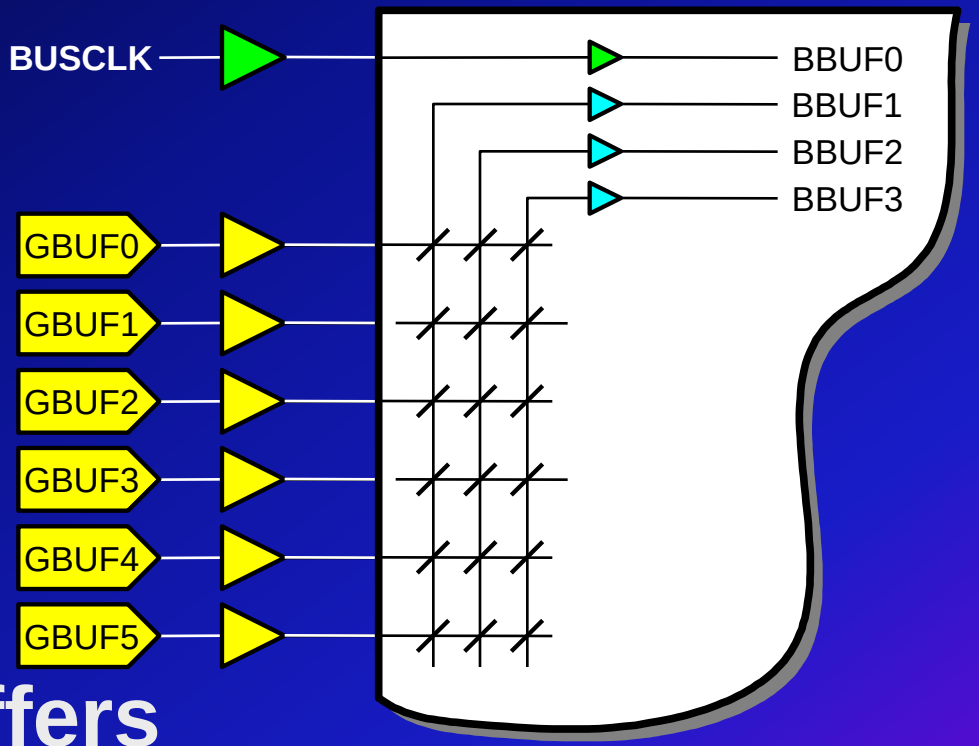
CSI Socket Routing



Triscend®

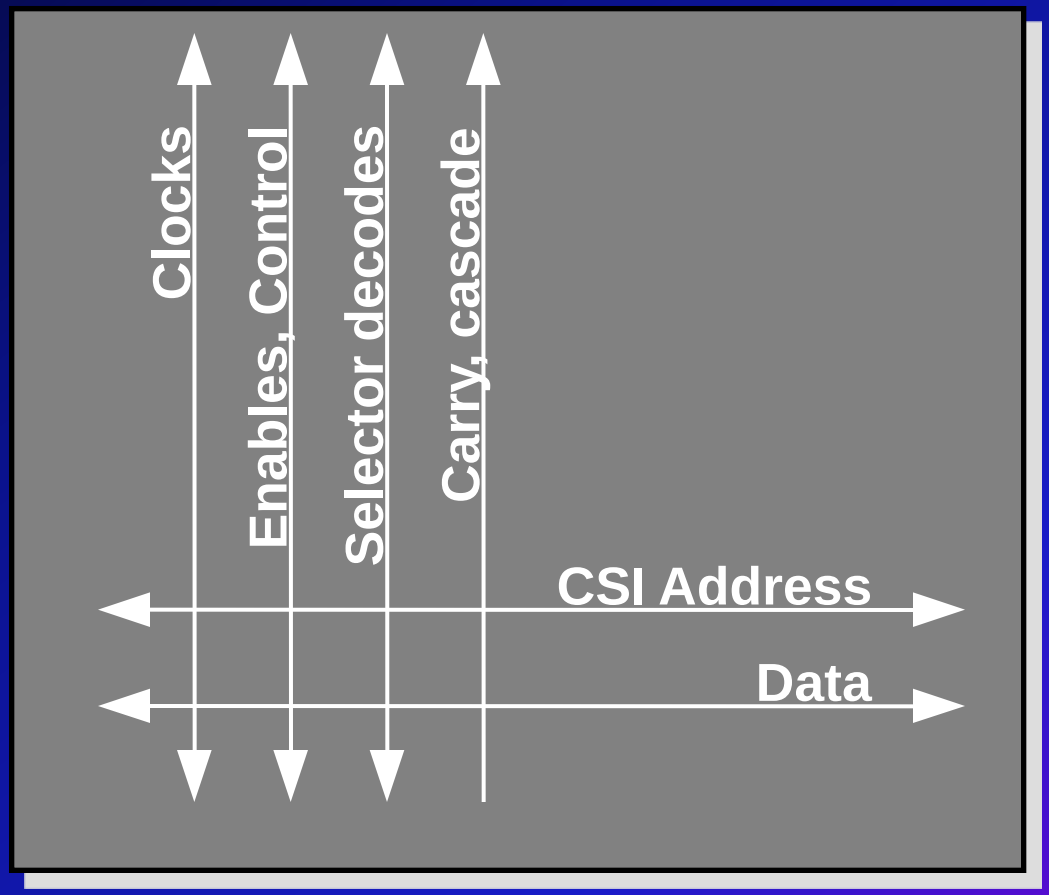
Clock Distribution

- **Bus Clock available everywhere**
  - CPU
  - Peripherals
  - Bus
  - CSL
- **Six Global Buffers**
  - High fanout signals
  - Not just clocks
  - 3 per CSL bank



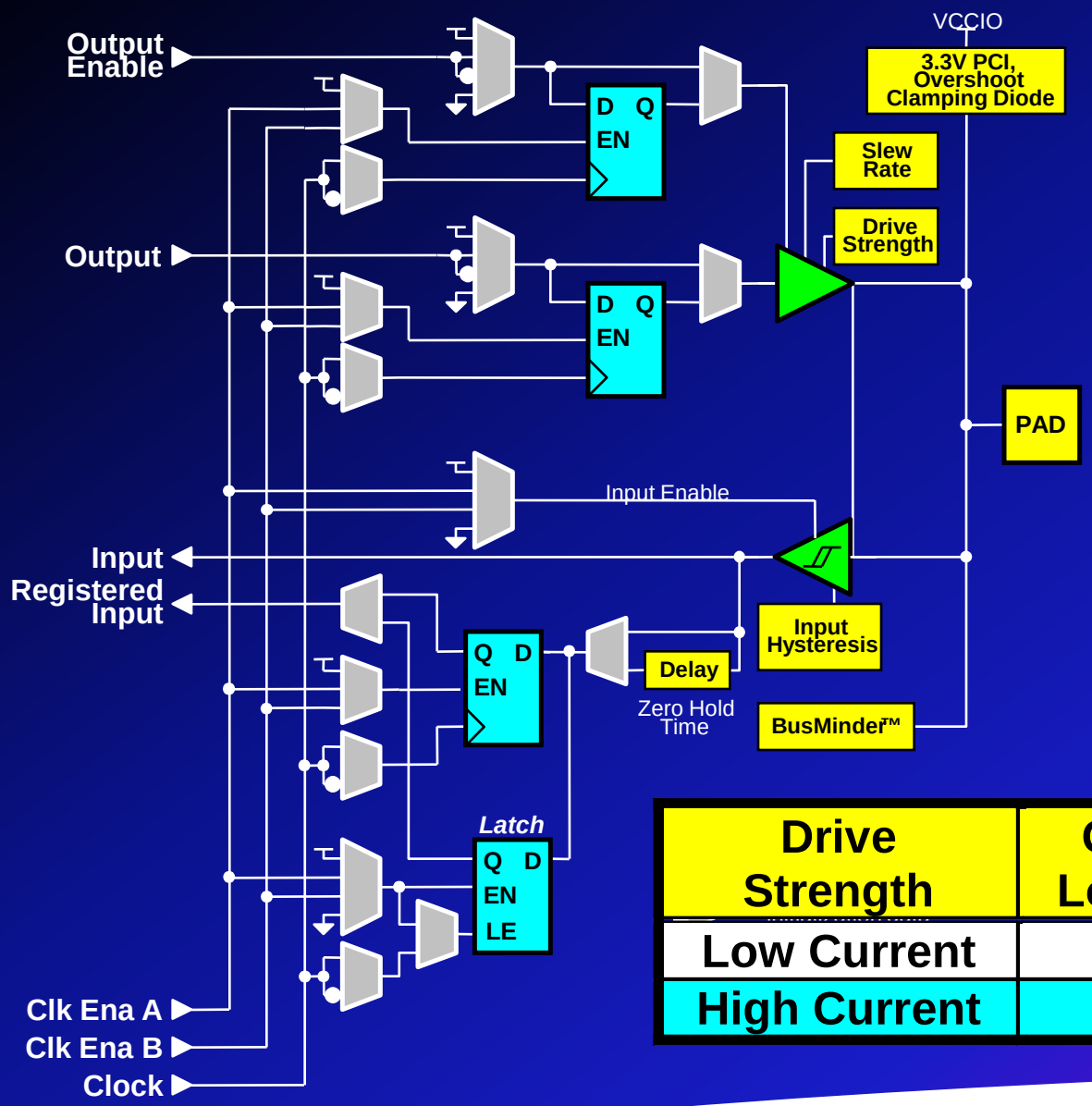
Triscend®

Preferred Signal Flow



Triscend®

PIO Cell



- 2.5 or 3.3 volts support
- Slew rate control
- Weak pull-up or pull-down resistor, or bus follower
- 3.3-volt PCI compliance
- Selectable output drive current

Drive Strength	Output Low (I_{OL})	Output High (I_{OH})
Low Current	4 mA	2 mA
High Current	16 mA	8 mA

Triscend®

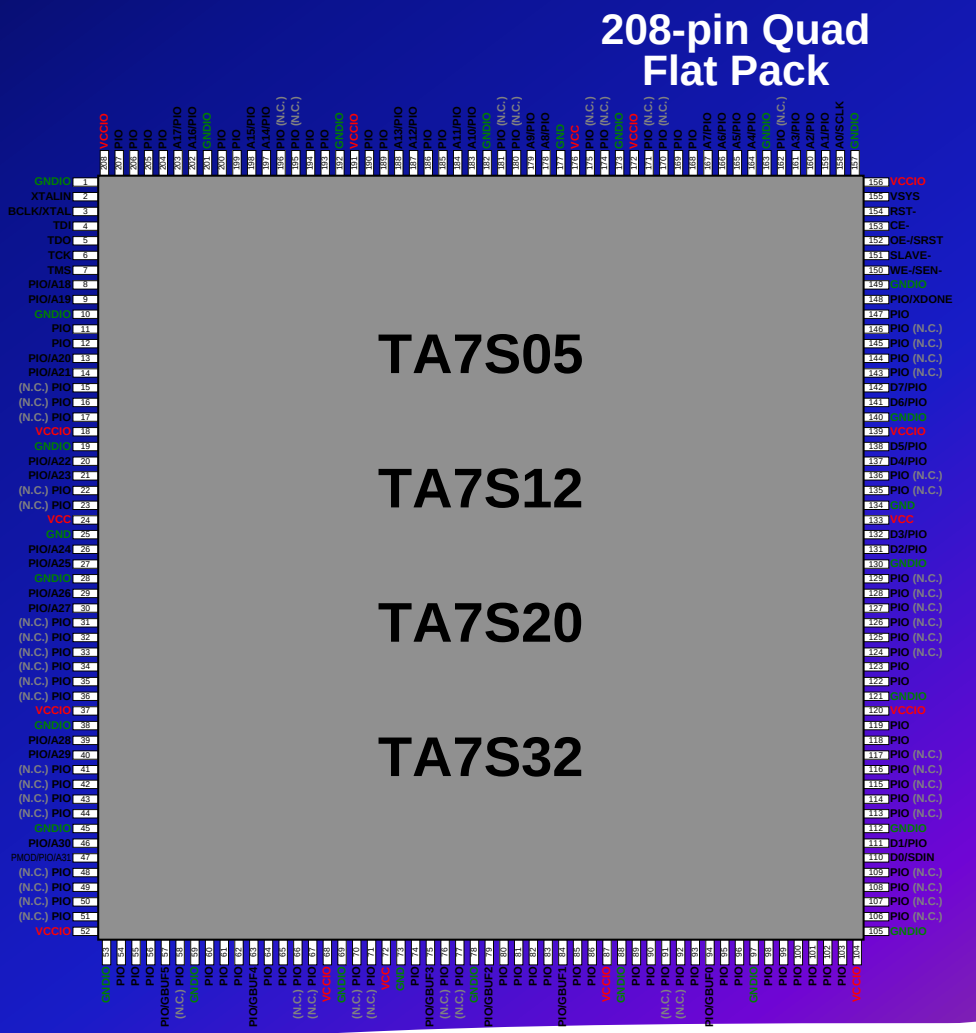
Available PIOs (PQ208)

- Total user PIOs depends on device, package, and operating mode
- Available PIOs does not include memory interface signals

Mode	External Memory Interface	Available PIOs
Maximum	32-bit Data bus 24-bit Address bus (16M)	86
Minimum	8-bit Data bus 21-bit Address bus (2M)	116

A7 Pin-Compatible Footprint

- **Various devices available in same footprint**
- **One board layout supports up to four different E5 devices**
- **E5 and A7 are not pin-compatible** (but power and ground pins on the same pins)



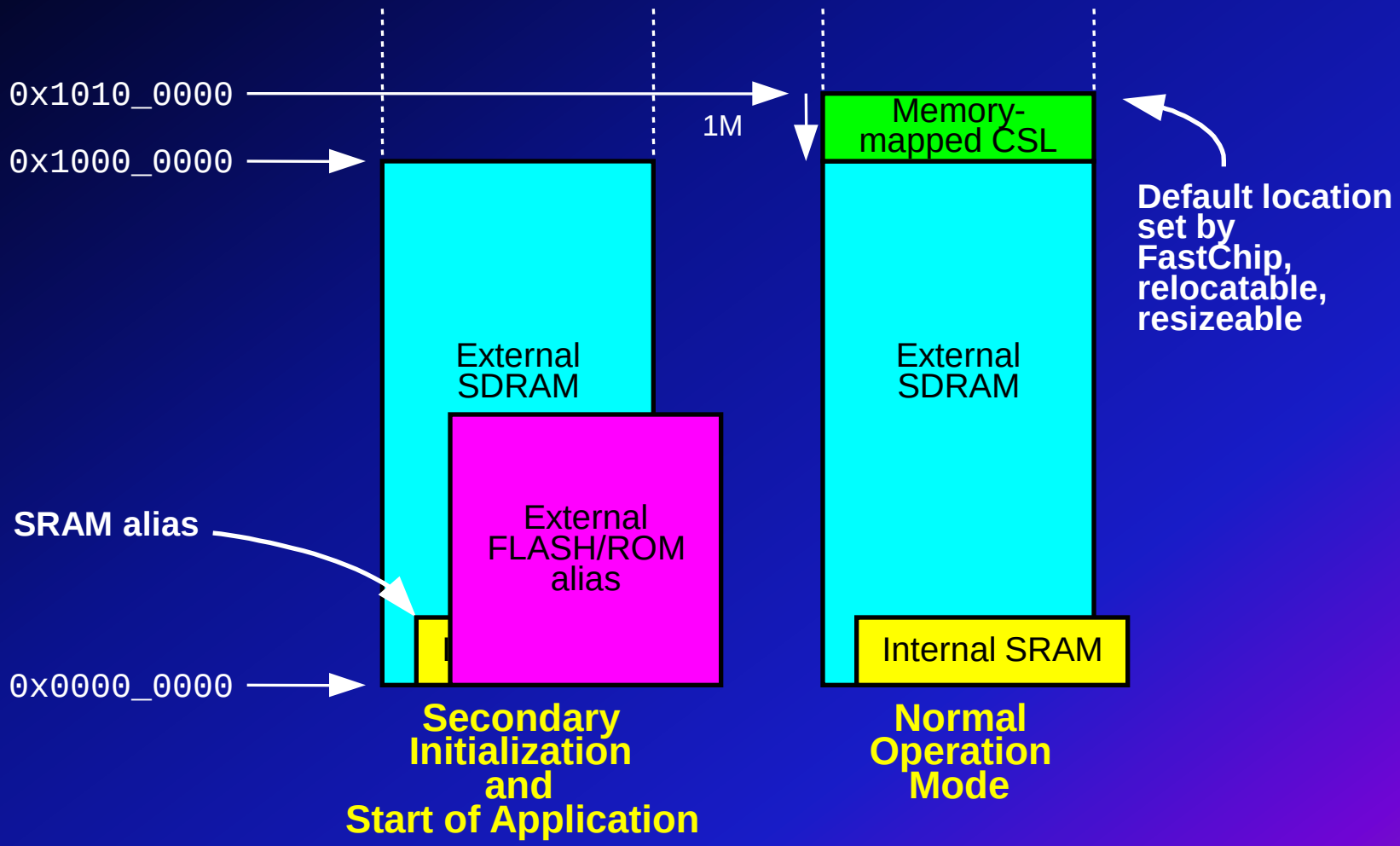
Triscend®

Voltage Compatibility

VCC	VCCIO	Compatible Standards
2.5 V	3.3 V	TTL, LVTTL 3.3V PCI 3.3V CMOS
	2.5 V	2.5V CMOS Tolerates 3.3V inputs

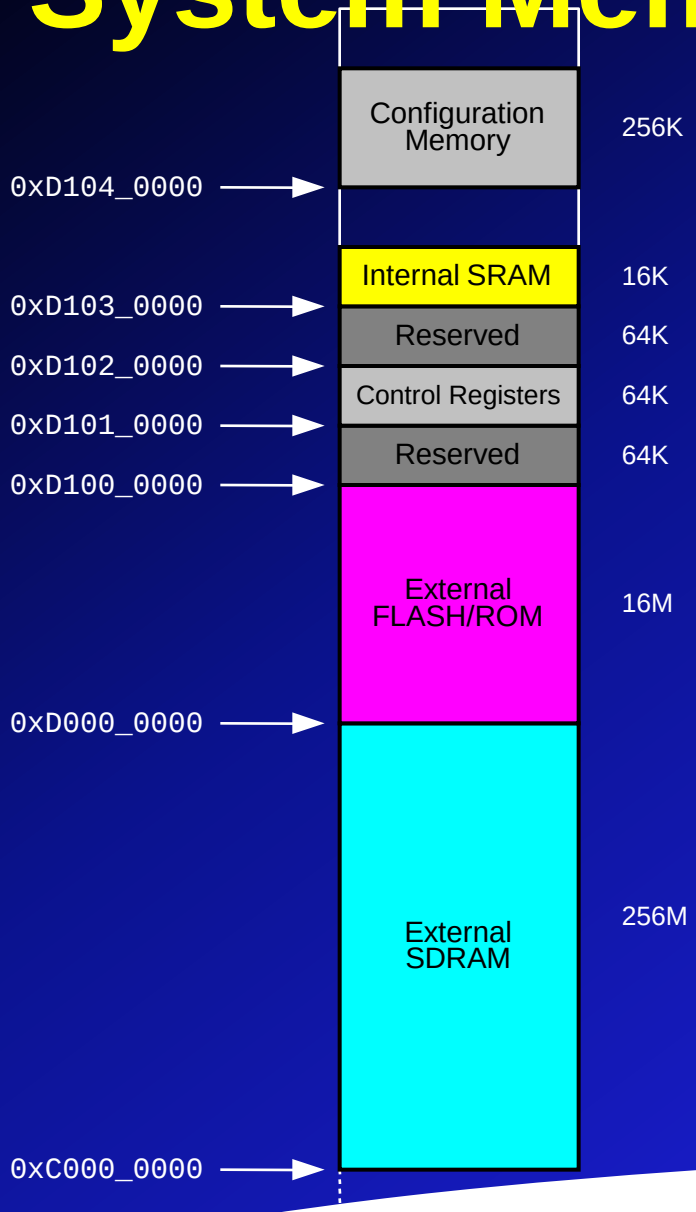
- VCC for core is always 2.5 volts
- VCCIO for PIO pins is either 2.5 or 3.3 volts
- Choose the right VCCIO for the desired logic standard
- The A7 is not 5-volt tolerant

System Memory Map (Low)



Triscend®

System Memory Map (High)



Resource	Allocated Space	Base Address
Control Registers (CRU)	64K bytes	Fixed
Configuration Memory	256K bytes	Fixed
Memory-mapped CSL Functions	1M bytes	Relocatable, resizable in FastChip
Scratch Pad	16K bytes	Re-locatable with alias at 0
External FLASH	16M bytes	Fixed with alias at 0
External Dynamic Memory	256M bytes	Fixed with alias at 0

Triscend®

Remap Register

- **Don’t want it down low? Kick it upstairs!**
- **Controls which type of memory is aliased at 0x0**
- **If more than type is aliased, then overlaid with the following priority**
  - Internal boot ROM
  - External Flash or static memory
  - Internal SRAM
  - External SDRAM

Scratchpad RAM

- 16K bytes (4K x 32)
- Guaranteed single cycle, no wait-states
- Store data or critical code
- Split into two 8K halves for fast, simultaneous access from CPU, DMA
- Basic protection against spurious CSI access (four 4K regions)
- Optionally use upper 8K half for embedded trace

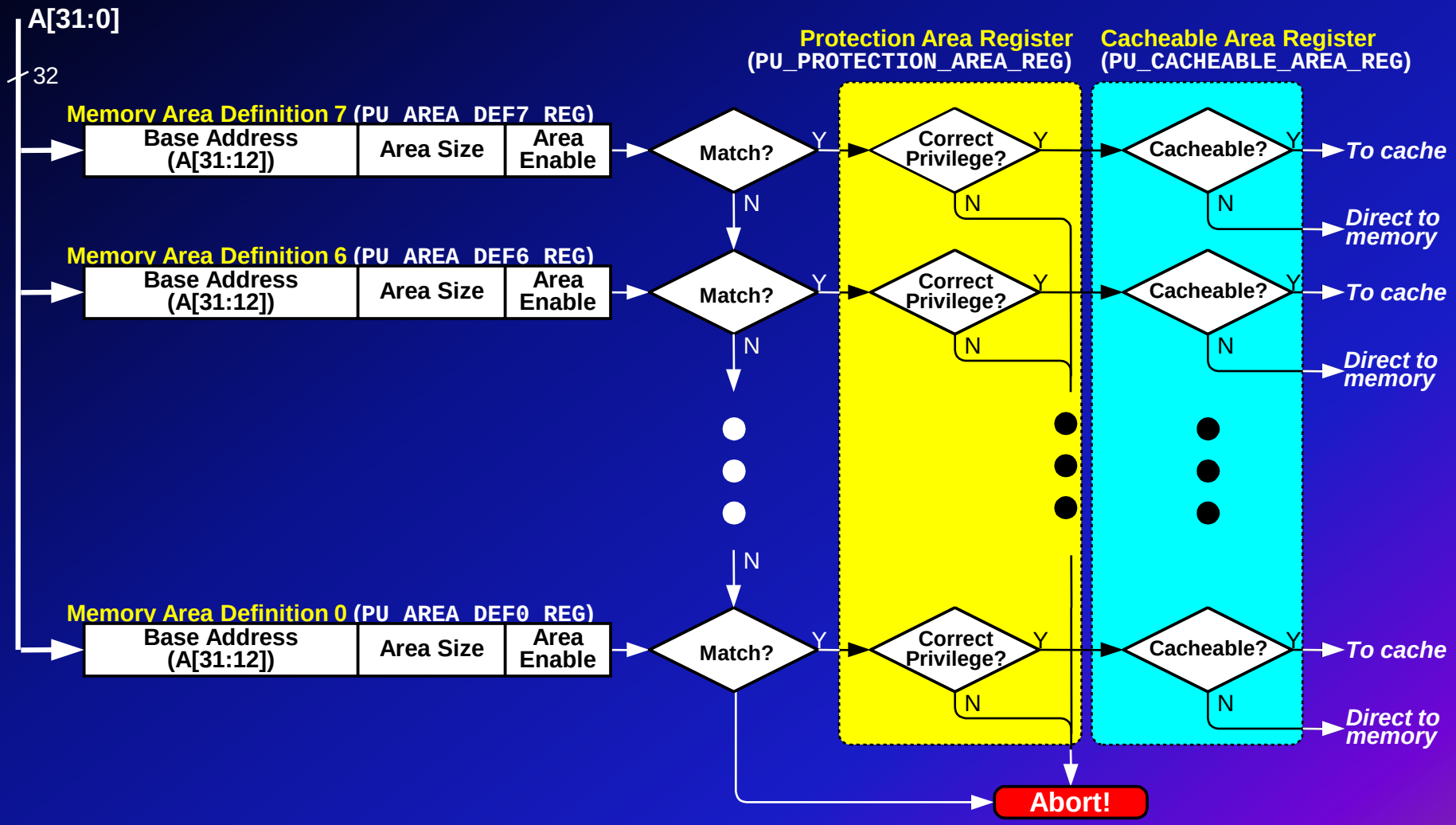
Cache

- **8K-bytes**
  - 512 cache lines
  - Each cache line is four words
- **Write-through cache**
- **Memory protection**
- **Not reclaimable as SRAM**
- **Most applications should use the cache for performance reasons**
- **Memory-mapped CSI functions cannot be cached**

Memory Protection Unit

- **Protects memory locations against illegal access**
- **Settings**
  - Base address for the memory region
  - Size of the memory region (4KB–4GB)
  - Is the region cacheable or not?
  - Access permissions
    - Does User mode have access or not?
    - Does System mode have access or not?
- **Protection violation causes an abort**

Protection Unit

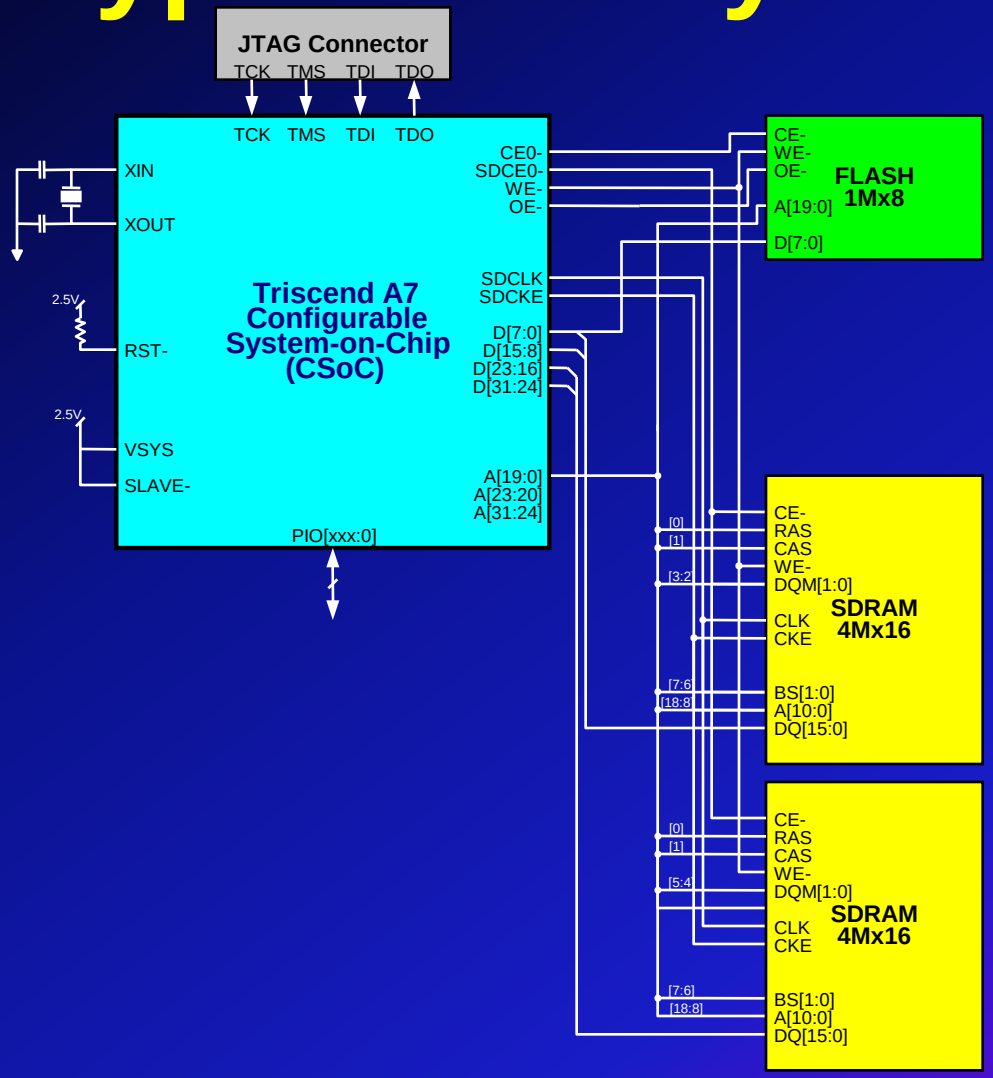


Triscend®

Memory Sub-System Interface Unit (MSSIU)

- **Simultaneous support for independent memory subsystems**
  - Static memory – Flash, SRAM, ROM, etc.
  - SDRAM, including 100-pin SIMMs
  - Glue-less integration
- **Expandable external data bus**
  - 8-, 16-, or 32-bits
  - Automatic support for byte, half-word, or word accesses by CPU

Typical A7 System

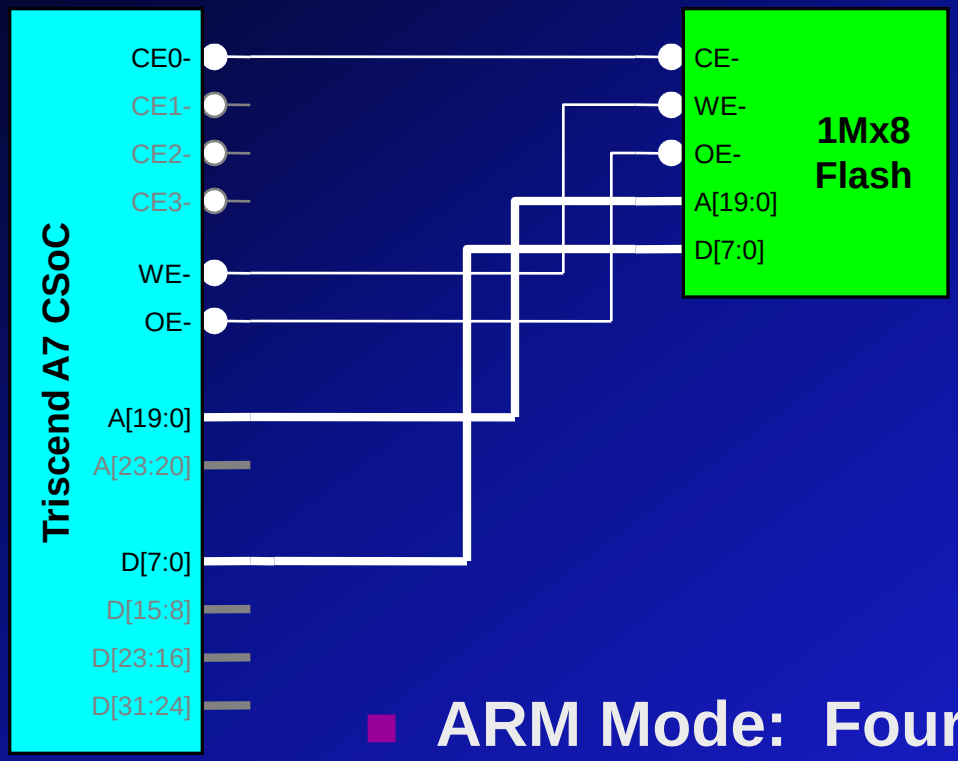


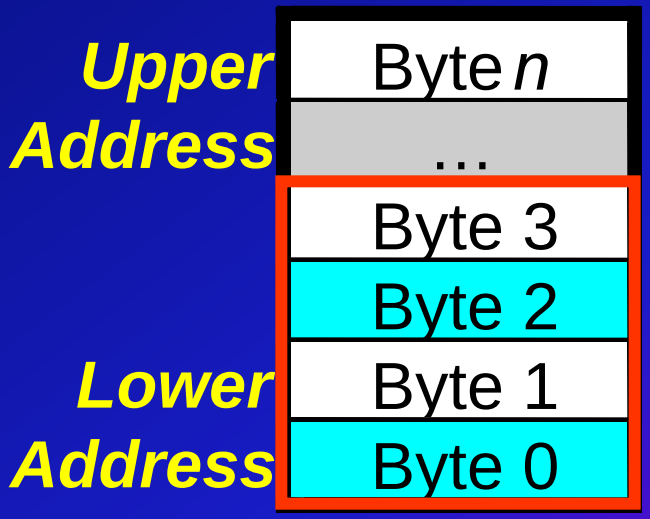
Triscend®

Static Memory Interface

- **Flash, SRAM, ROM, etc.**
- **Optimized for direct execution**
- **16MB total address space available**
  - Requires 24 address lines
  - Mapped between address 0xD000_0000 and 0xD0FF_FFFF.
- **Can be used simultaneously with SDRAM interface**

8-bit Mode

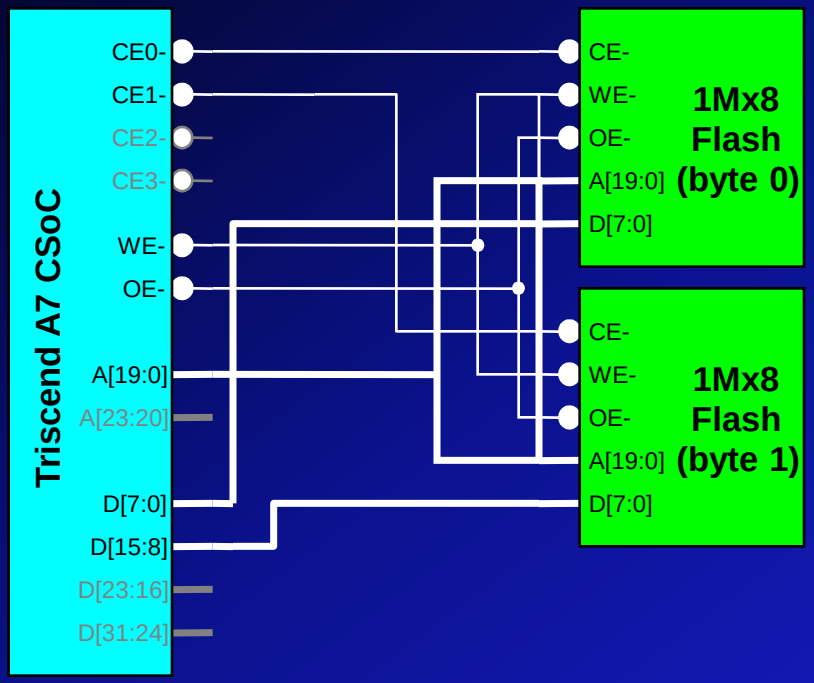


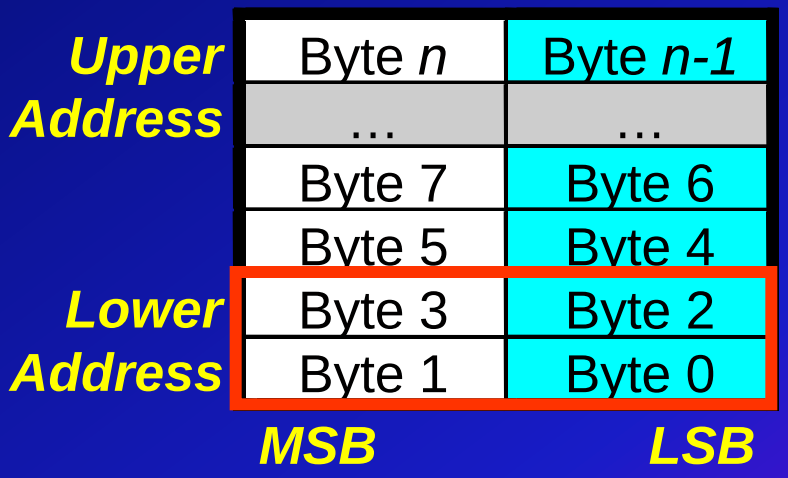


- ARM Mode: Four single-byte fetches
- Thumb Mode: Two single-byte fetches

Triscend®

16-bit Mode

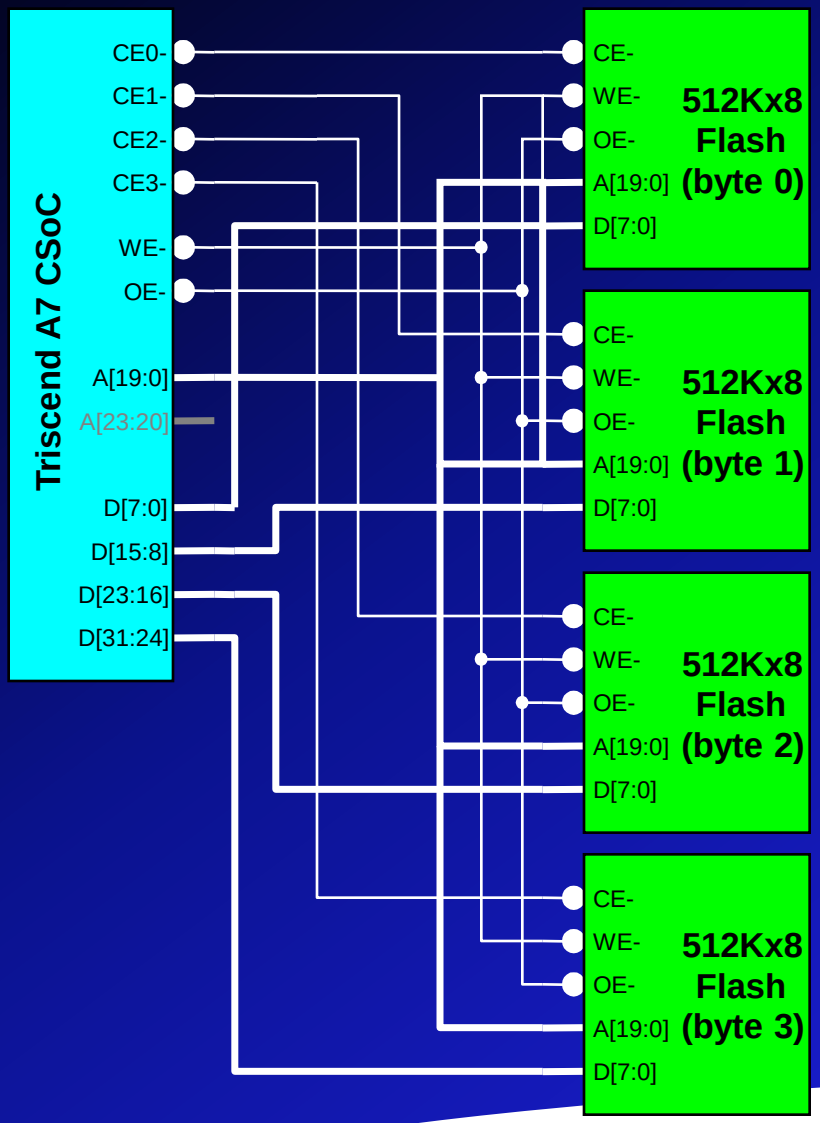




- ARM Mode: Two half-word fetches
- Thumb Mode: Single half-word fetch

Triscend®

32-bit Mode

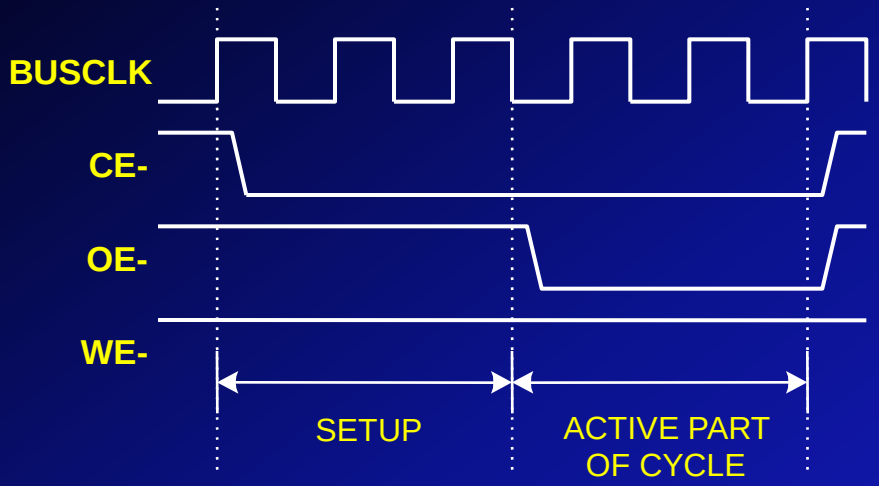


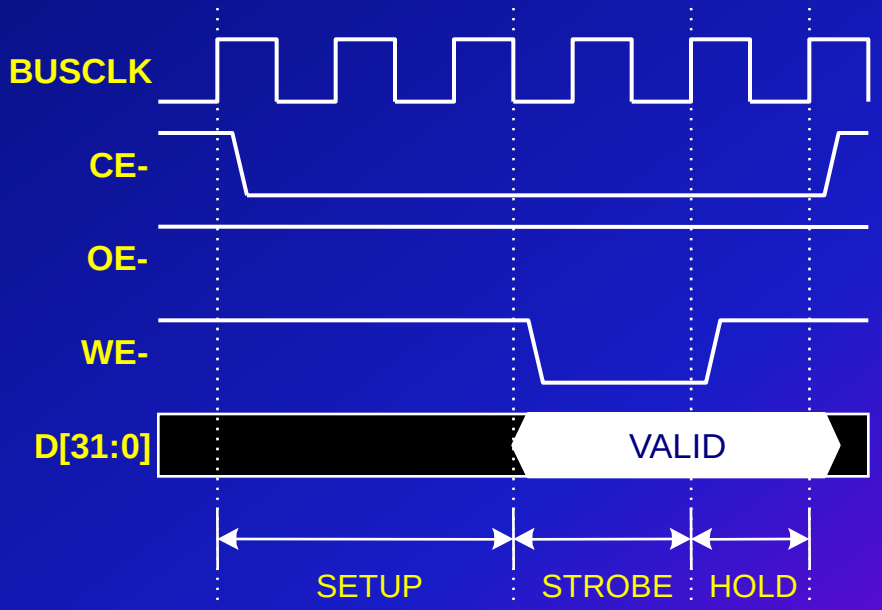
Upper Address	Byte *n*	Byte *n-1*	Byte *n-2*	Byte *n-3*
	…	…	…	…
	Byte 15	Byte 14	Byte 13	Byte 12
	Byte 11	Byte 10	Byte 9	Byte 8
Lower Address	Byte 7	Byte 6	Byte 5	Byte 4
	Byte 3	Byte 2	Byte 1	Byte 0
	MSB			*LSB*

- ARM Mode: Single word-wide fetch
- Thumb Mode: Single word-wide fetch grabs two instructions

Triscend®

Flexible Static Memory Timing



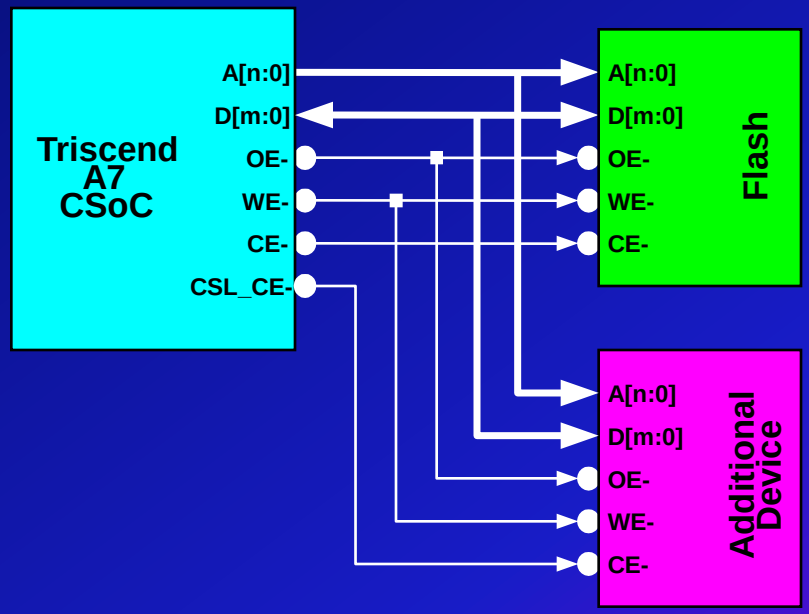


- Setup, Strobe, and Hold times are programmable
- 0.5 to 15.5 clock cycles in one cycle increments

Triscend®

Expansion via MSSIU

- **Connect external devices to MSSIU pins**
  - Typically additional memory-mapped peripherals
  - Requires separate chip select using Selector
  - CSL must control external timing via wait-states
- **Conserves PIO pins**
- **No (very limited) DMA support**
- **Always requires wait-states**



Triscend®

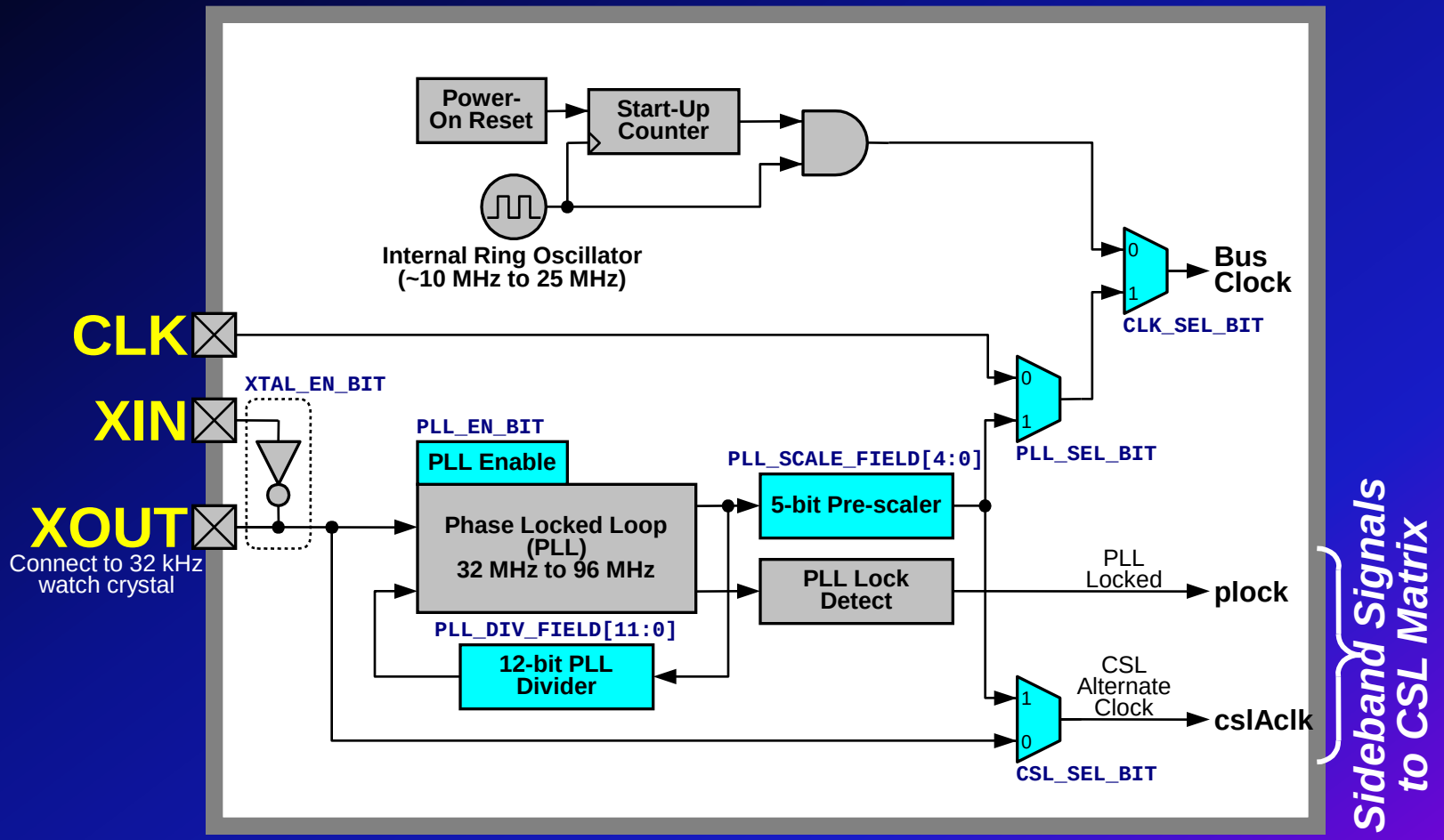
SDRAM Interface

- **Available simultaneously with static memory interface**
- **Not required for low-end systems**
- **Supports SDRAMs**
  - 64Mbit, 128Mbit, and 256Mbit
  - x8 or x16 data configuration
- **Supports 100-pin SIMMs**
  - 4M-byte to 64M-byte
- **8-, 16-, or 32-bit interface**
- **Optimized for cache fills, DMA transfers**
- **Supports automatic and self refresh**

SDRAM Operating Modes

- **Disable** – Power-on state. No access or refresh
- **Active** – Transition to active state from the disable state triggers the SDRAM power-up sequence. The SDRAM is ready for full speed access
- **Stand-by** – Same as active state but the SDRAM enters a power saving mode as soon as SDRAM accesses end
- **Power-down** – Generates a self-refresh sequence. The SDRAM retains its data indefinitely in this state

Clock Control



Triscend®

Power-Down Operation

- ARM Pause Register shuts down just the ARM7TDMI
- Other A7 Power-Down Options

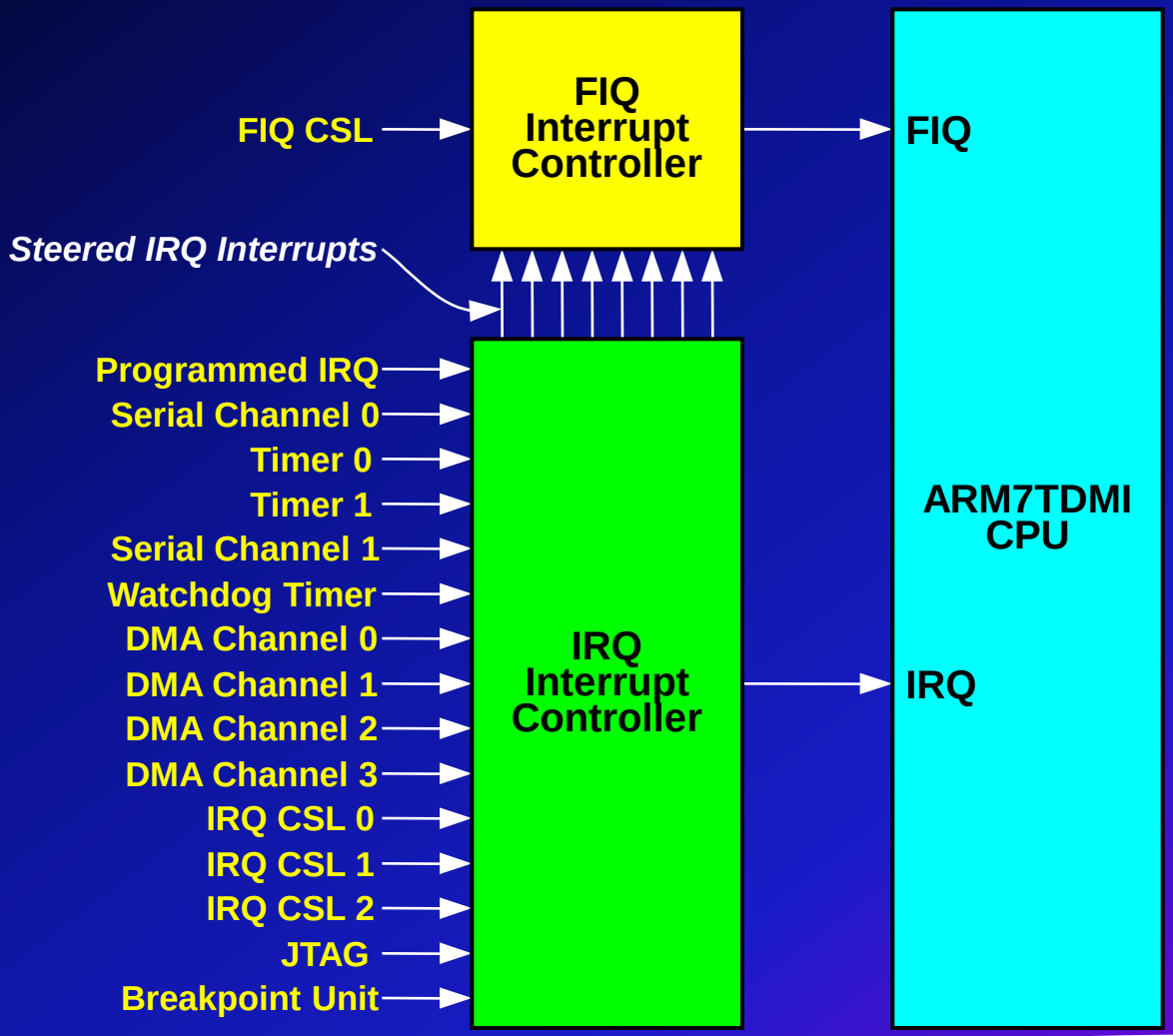
Feature	Software Shut Down	Dynamic Shut Down
Power-On Reset (POR)	✓	
PLL	✓	
Crystal Oscillator	✓	
Internal Ring Oscillator		✓
I/O buffers		✓
System clock		✓
CSL user system clock		✓
CSL clocks (ENCK)	✓	

Estimated lowest power mode is 100 µA

Interrupts

- **ARM7 has two interrupts**
  - FIQ (fast interrupt)
  - IRQ (everything else)
- **A7 Expands IRQ to 15-inputs**
- **IRQ interrupts can be steered to FIQ**
- **IRQ interrupts**
  - Software prioritization
  - Level-sensitive, active-High
- **FIQ Latency**
  - 28 clock cycles (very worst-case), 467 ns @ 60 MHz
  - 4 clock cycles (best case), 67 ns @ 60 MHz

A7 Interrupt Controller



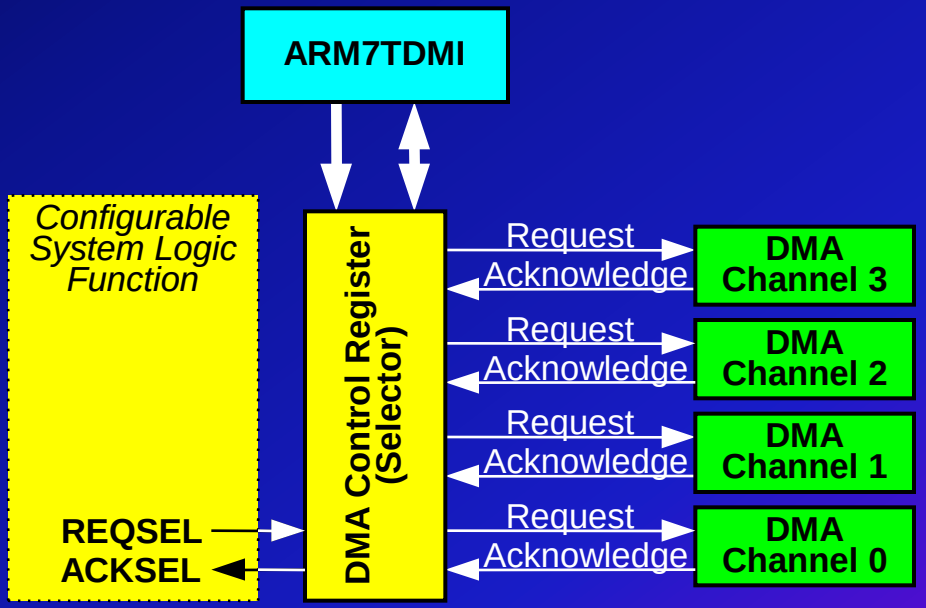
Triscend®

DMA Controller

- **Four-channel high-performance DMA**
  - Fly-by performance
  - Memory-to-memory using only one channel
  - Linked-list DMA
  - Frame-transfer support
- **CSI Bus Master**
  - Operates independently of processor
  - Full 240MB/sec transfer rate on CSI bus
- **Supports 8-, 16-, and 32-bit transfers**

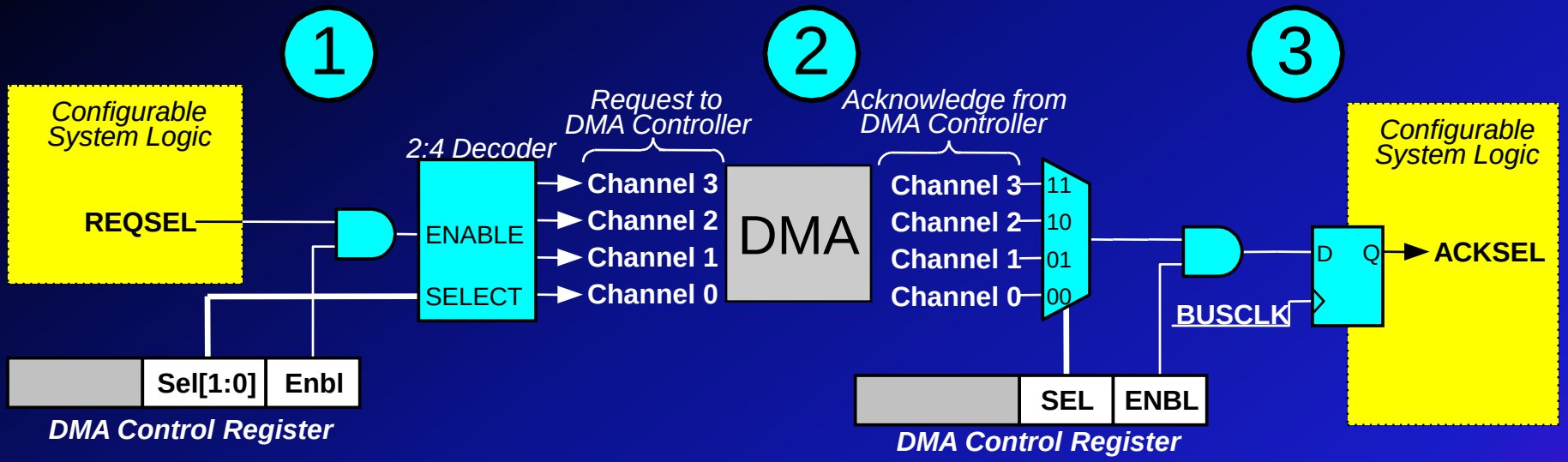
DMA Control Register in CSL

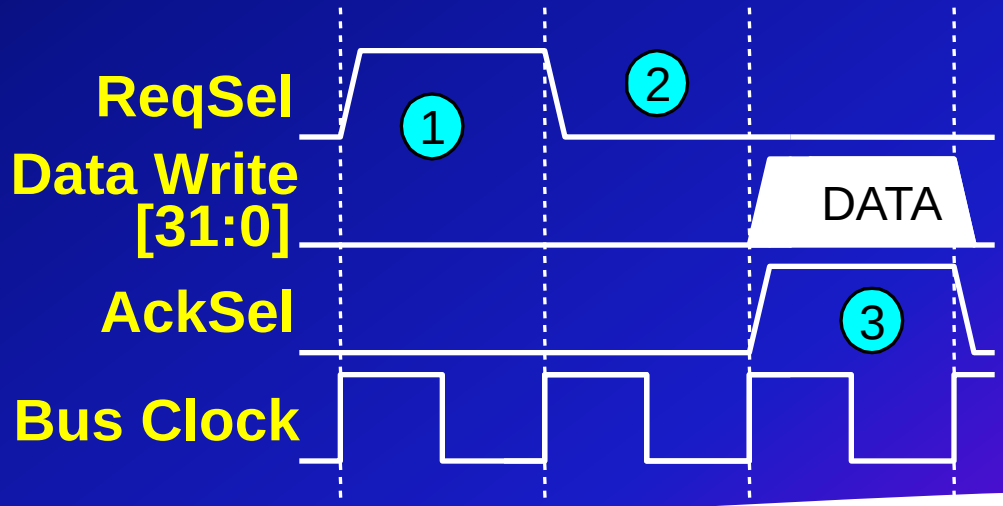
- How does a device in CSL request DMA services?
- DMA Control Register (Selector) steers request and acknowledge signals
- Every Selector can be a DMA Control Register
- Four DMA channels are shared
- CPU provides high-level control



Triscend®

Request/Acknowledge Steering



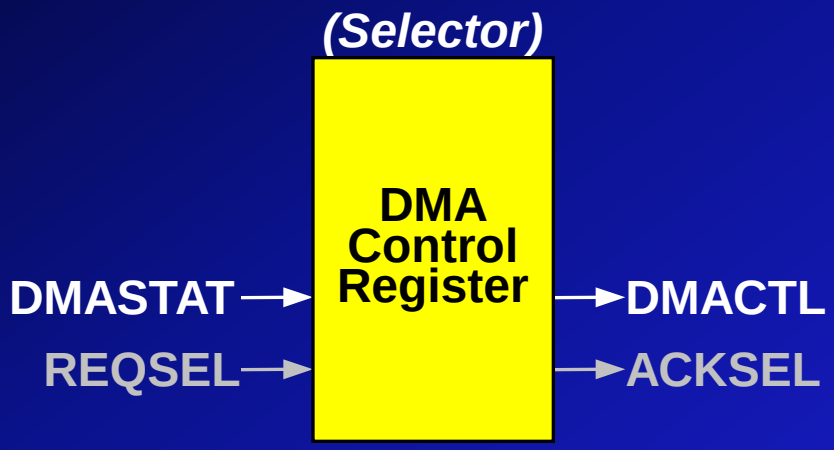


Triscend®

New “Auxiliary” DMA Handshake Signals

- “E5 Compatibility” mode is the default
- Clear the **AUX_DIS_BIT** to enable auxiliary control signals
- Two new auxiliary handshake signals
  - **DMASTAT** – Retransmit, Last request
  - **DMACTL** – Retransmit acknowledge, Last request acknowledge
- Useful when you don’t know the size of the data transfer

Auxiliary DMA Handshake



E5 Mode

REQSEL	DMASTAT	Action
0	0	No Request
1	0	Request
0	1	Retransmit
1	1	Last request

ACKSEL	DMACTL	Action
0	0	No Acknowledge
1	0	Acknowledge
0	1	Retransmit Acknowledge
1	1	Last Acknowledge

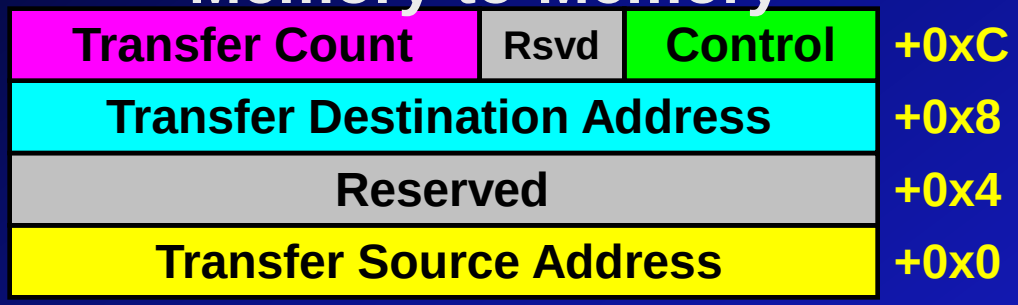
Triscend®

New DMA Descriptor Mode

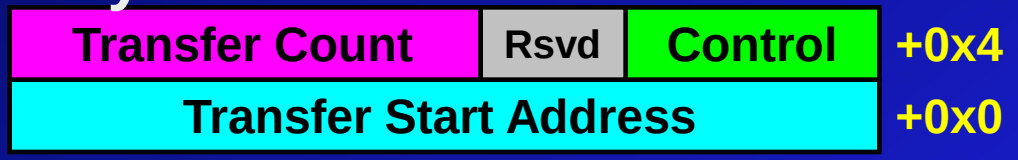
- **“Fire and forget” capability frees CPU**
  - CPU sets up transfer characteristics in memory (descriptor table)
  - CPU initiates DMA transfer
  - DMA reads “descriptor table” and manages entire transfer
- **Advanced DMA transfer capabilities**
  - Automated linked-list transfers
  - Scatter/gather
- **Set `METHOD_BIT` to enable**

What's a Descriptor?

Memory-to-Memory



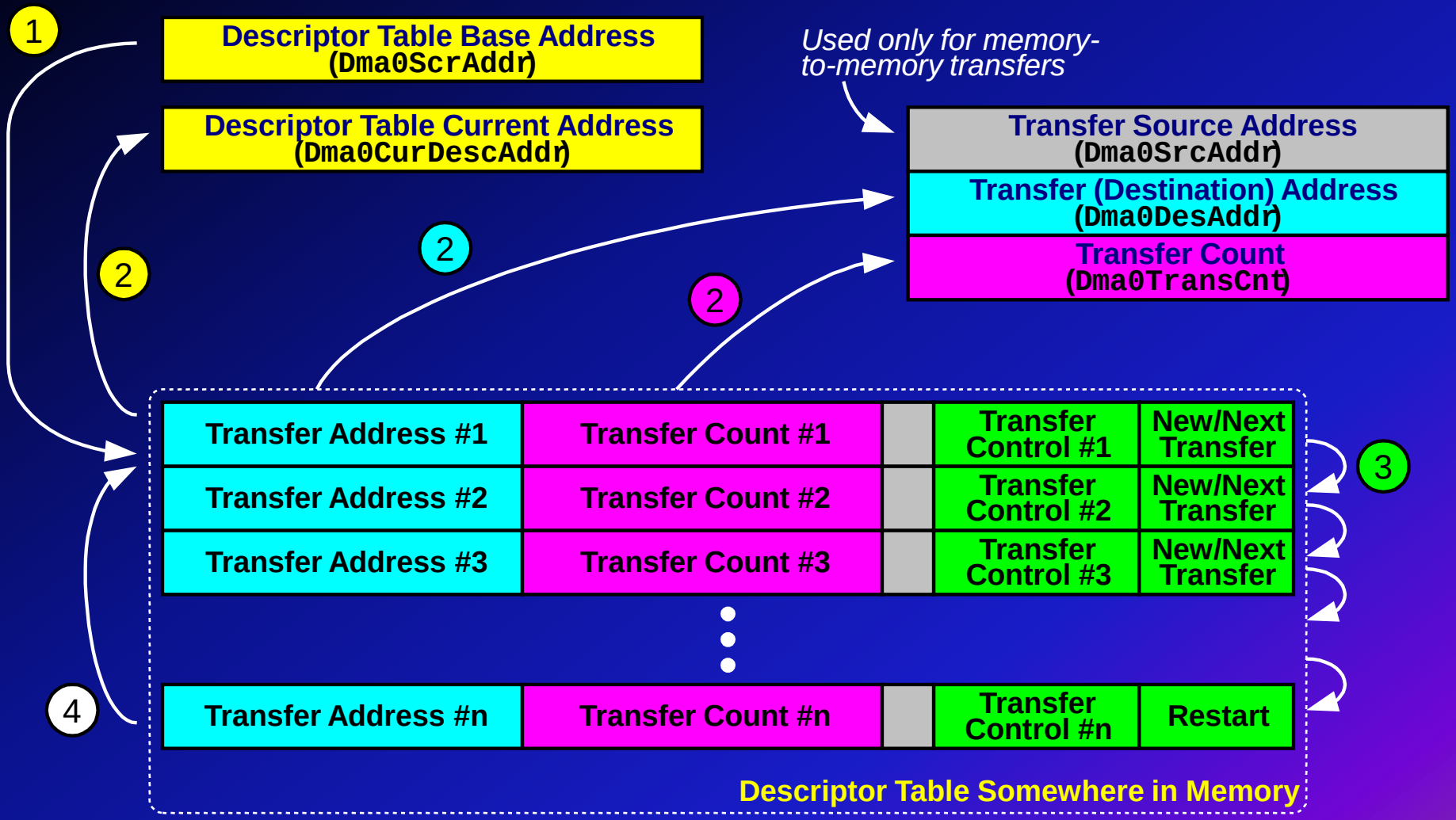
Memory-to-Device or Device-to-Memory



- **Table located somewhere in memory**
- **Contains transfer characteristics**
- **Control/Status**
  - Buffer Full/Empty
  - Continuous operation
  - Descriptor interrupt
  - Next action

Triscend®

Example Descriptor Transfer



Triscend®

Debugging CSoC Designs

- Real-time
- In-system
- Using the same production silicon
- Without an expensive in-circuit emulator
- Using all the other system hardware
- Using all the other system software
- Without sacrificing device resources *(well mostly)*

Unparalleled Debugging Support

- **JTAG Debugging Port (IEEE 1149.1)**
  - ARM7TDMI – Supports ARM debuggers
  - Triscend CSoC – Supports CSI bus and CSL logic debugging
  - JTAG is a CSI bus master
    - Read any memory-mapped location
    - Monitor and CSL LUT or flip-flop output
    - Single-step the system
  - Program external Flash memory
- **ARM EmbeddedICE™**
  - Standard with the ARM7TDMI
  - ARM breakpoint, watchpoint unit

Unparalleled Debugging Support

- **CSoC Hardware Breakpoint Unit**
  - Monitors CPU and CSI busses
  - Two independent or interworking breakpoints
  - Interrupt or freeze CPU
  - Single-step CPU or CSI via the JTAG port
- **Embedded Trace Capability**
  - Trace window controlled by CSoC Hardware Break point unit
  - Requires upper 8K-bytes of scratchpad RAM
  - Invaluable capability when you need it
    - Embedded, cached processor
    - Embedded bus

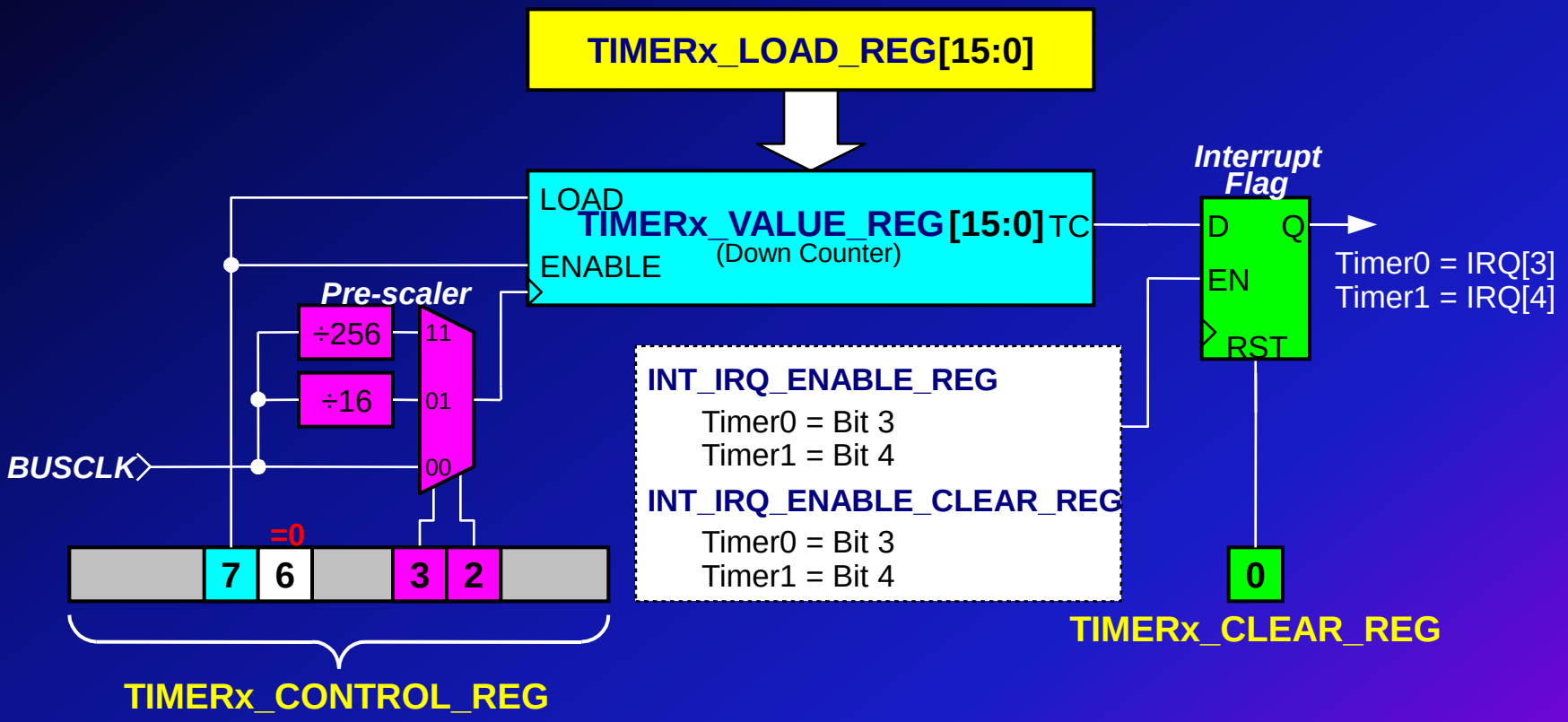
On-Chip Support Peripherals

- Provides basic support for most RTOSes
- Two reloadable 16-bit timers
- 32-bit Watchdog Timer
- Two 16C450/550-style UARTs, with modem interface

Timer 0 and Timer 1

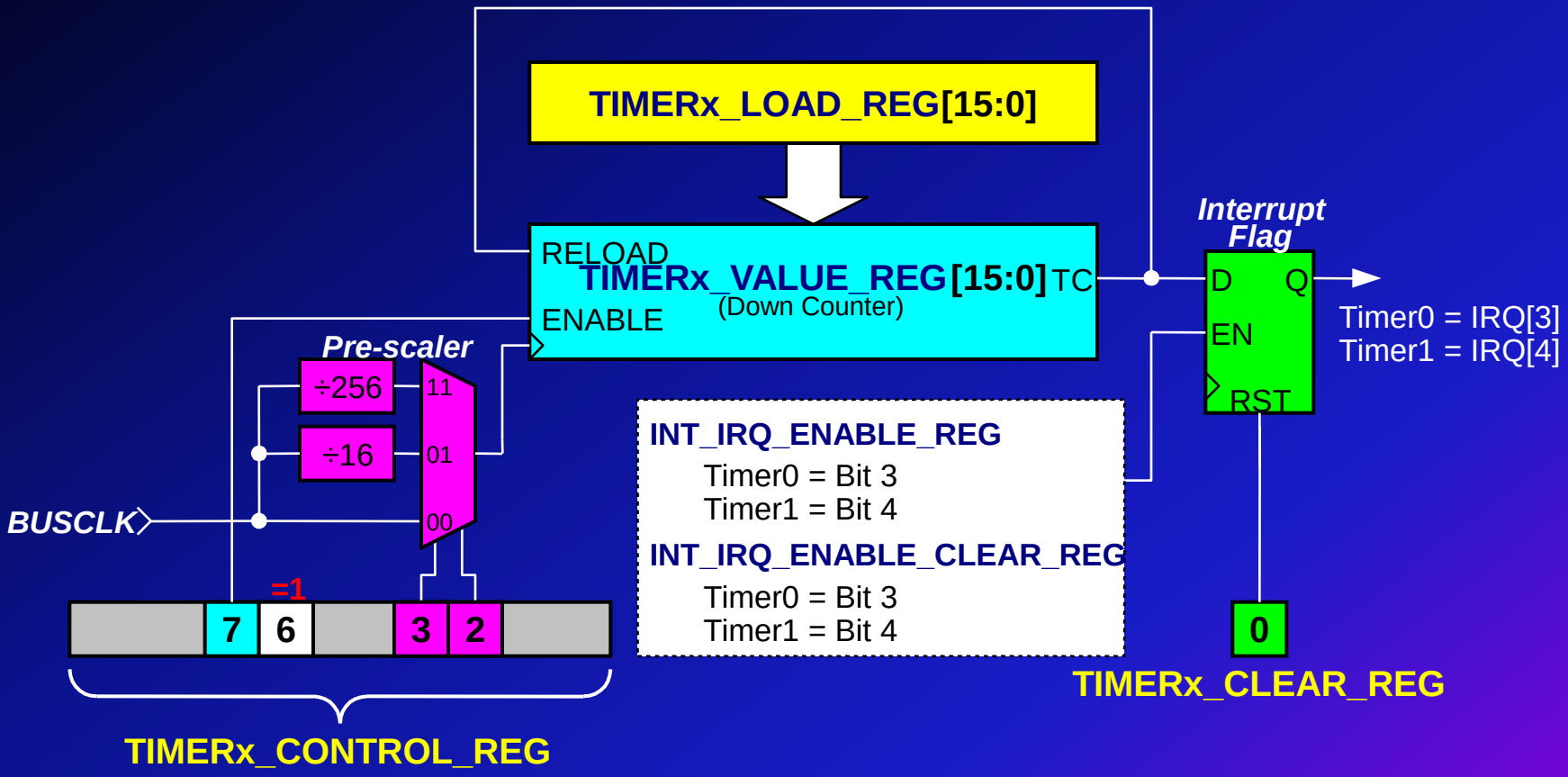
- **16-bit reloadable counters**
- **Two operating modes**
- **Free-running mode**
  - Load preload value and count down
  - Generate interrupt when reaching 0
  - Wrap around to maximum counter value
- **Periodic**
  - Load preload value and count down
  - Generate interrupt when reaching 0
  - Reload counter value

Free-Running Mode



Triscend®

Periodic Mode

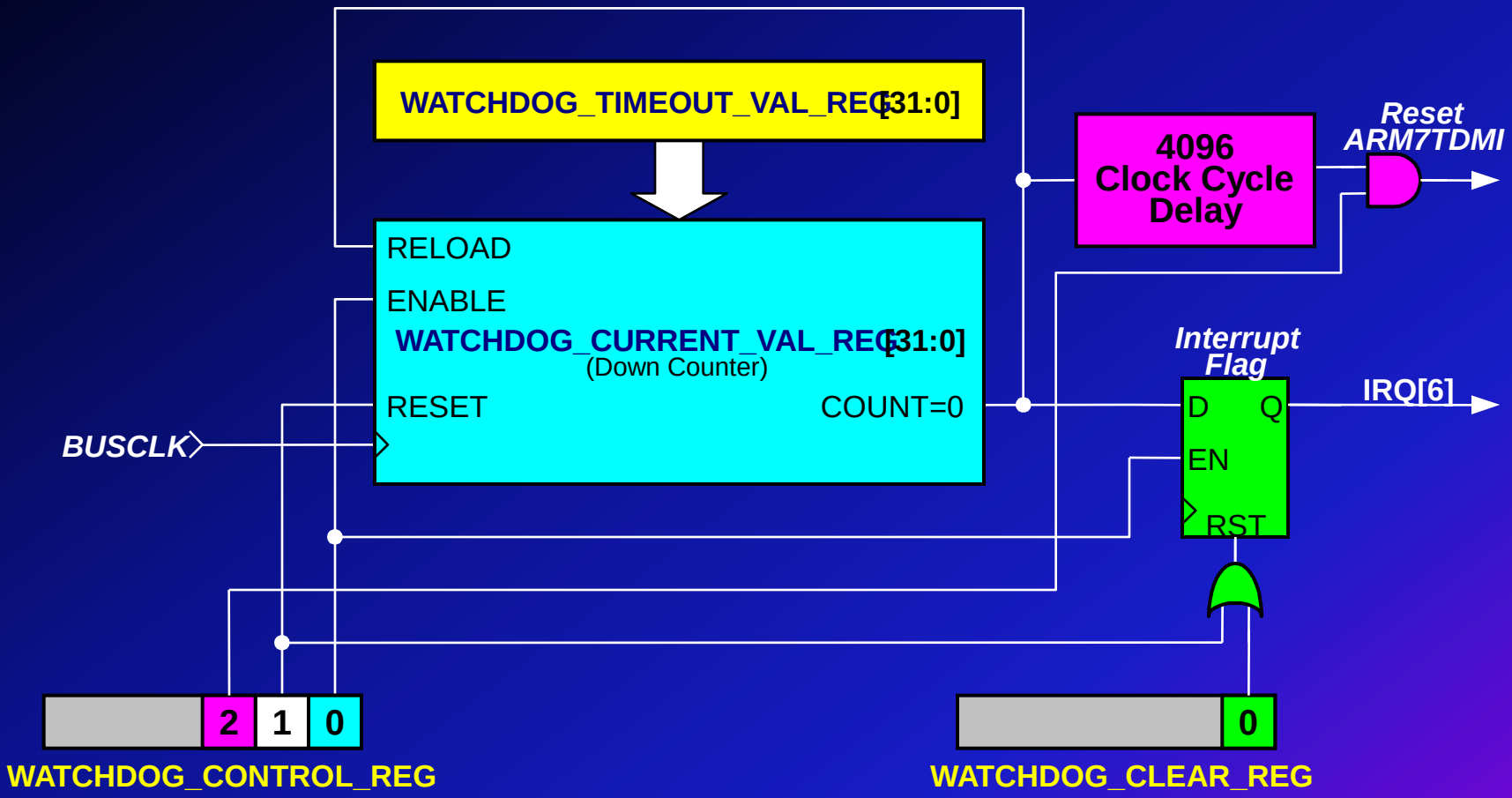


Triscend®

Watchdog Timer

- **32-bit reloadable counter**
- **Typically used to prevent errant operation**
  - Watchdog generates interrupt
  - Processor has 4096 cycles to reset Watchdog
  - If Watchdog not reset, Watchdog resets system
- **Can also be used for interval timer**

Watchdog Timer

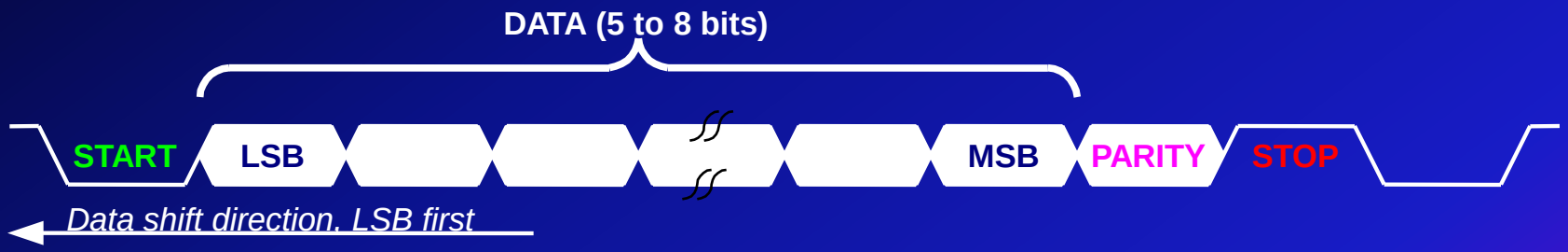


Triscend®

UARTs

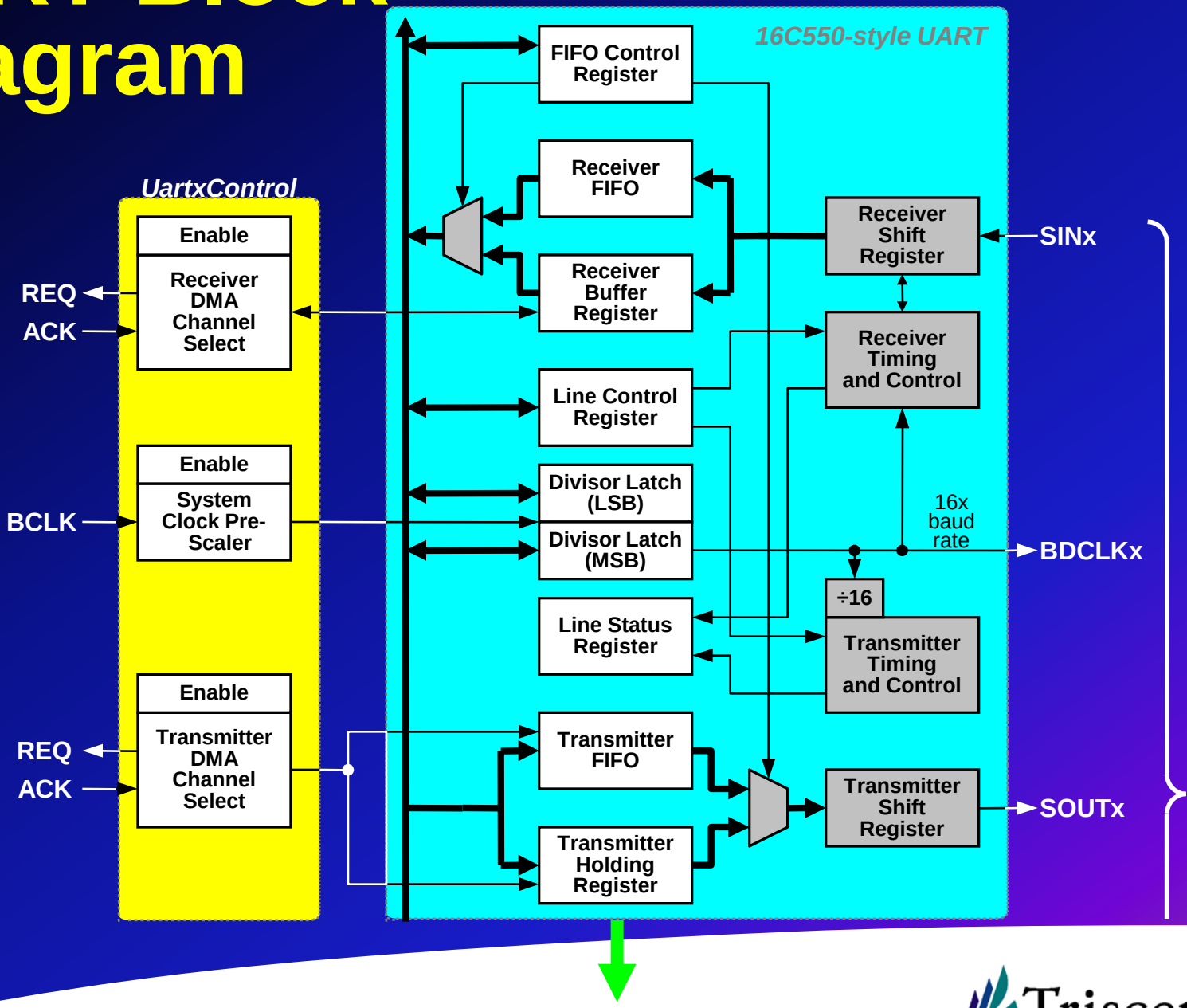
- **16C550-style (FIFO) or 16C450-style (non-FIFO) operating modes**
  - Transmitter is buffered with 16 byte FIFO
  - Receiver is buffered with 16 byte FIFO plus 3 error bits per data byte
- **5-, 6,- 7-, or 8 bit data transmission**
- **Even, odd, or no parity**
- **Internal programmable baud-rate generator**
- **Modem handshake capability**
  - Via side-band signals
  - One only

UART Data Format



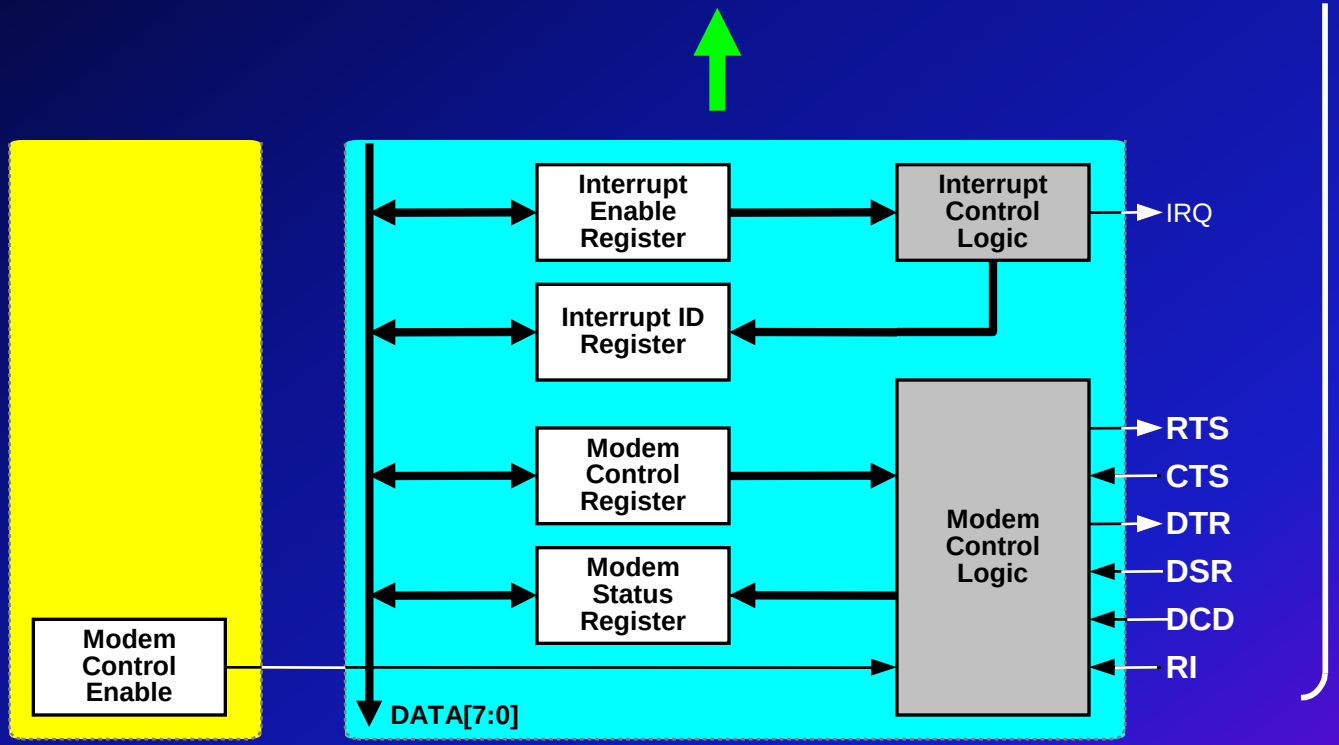
Triscend®

UART Block Diagram



Triscend®

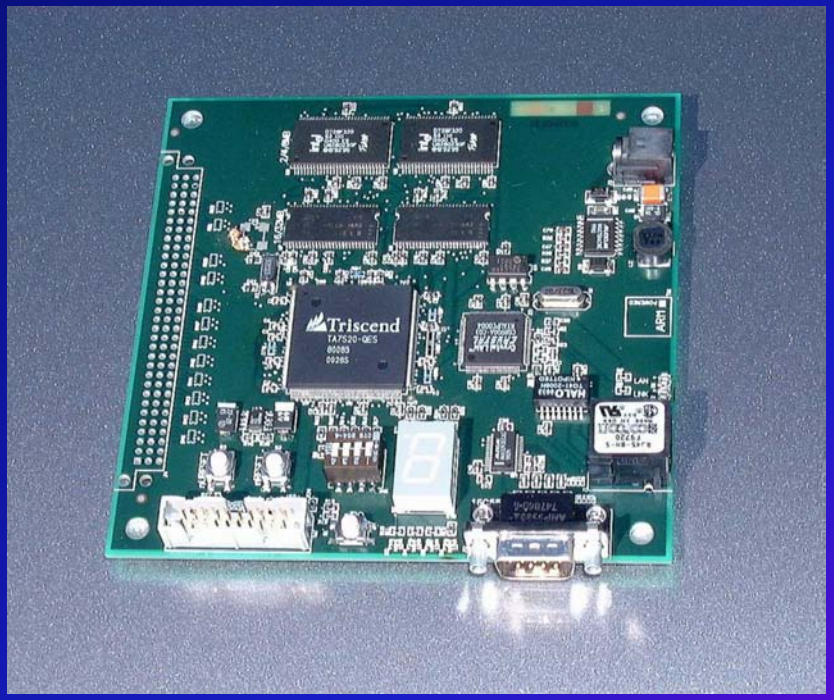
UART Block Diagram



Triscend®

Development Boards

A7 Evaluation Board



A7 Application Development Board

Triscend®