Triscend E5 Customizable Microcontroller Platform

March 2003 (Version 1.07) **Product Description**

- **Industry's first complete Customizable Microcontroller platform**
  - High-performance, industry-standard 8051/52-compatible microcontroller (10 MIPS at 40 MHz)
  - Up to 64Kbytes of on-chip, dedicated system RAM (XDATA RAM)
  - Up to 2,048 Configurable System Logic (CSL) cells (roughly 40,000 gates)
  - High-performance dedicated internal bus
  - Advanced system debug capability
  - Stand-alone operation from a single external memory (code + configuration)
  - Advanced four-layer metal, 0.35µ CMOS process technology, 3.3 volt with 5 volt-tolerant I/O
- **Enhanced, high-performance, 8051-based microcontroller**
  - Binary- and instruction-set compatible with other 8051-and 8052-based devices
  - 4 cycles per instruction byte provides up to 10 MIPS performance at 40 MHz
  - Configurable, extendable architecture supports user-designed or library-provided peripherals
  - Two-channel DMA controller supporting single-clock transfers
  - Programmable wait-state capability
  - Dual 16-bit data pointers
  - Three programmable 16-bit timer/counters
  - Programmable, full duplex asynchronous serial communications port
  - 256-byte scratchpad RAM
  - Protected programmable watchdog timer
  - Programmable power-down modes, including individual PIO options
  - Separate 64K address spaces for code and data
  - 12 interrupt sources with three priority levels
  - Embedded debugging capabilities

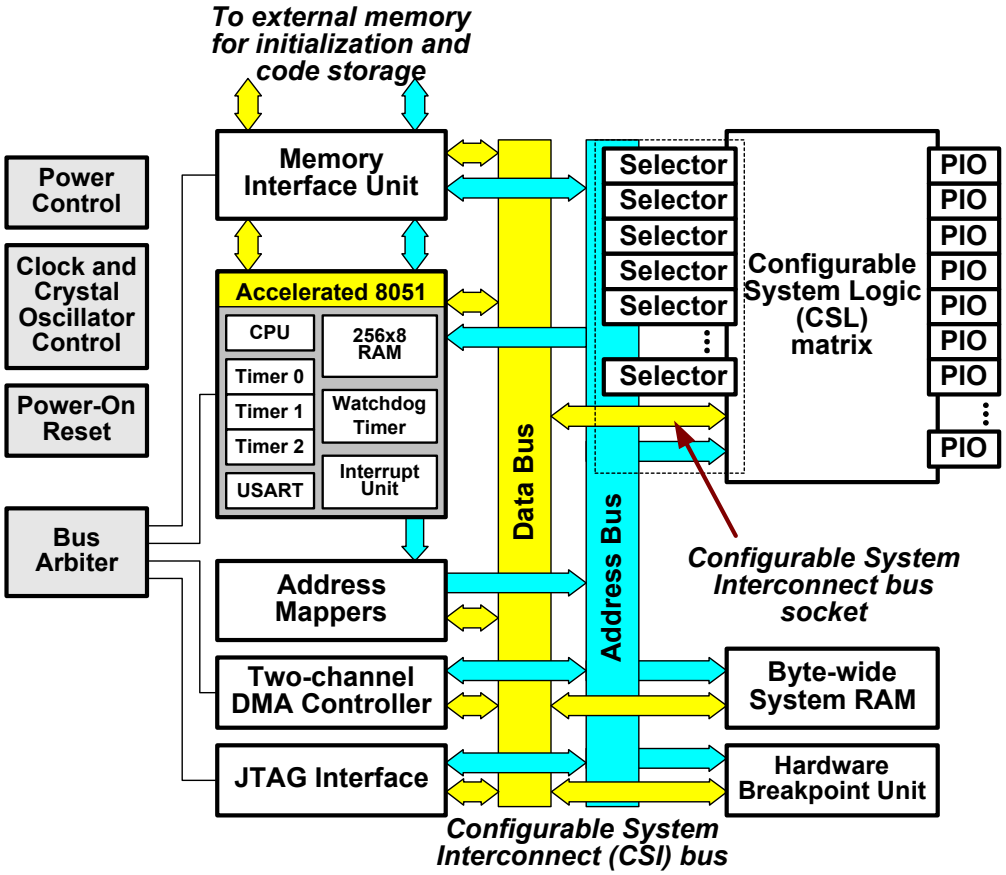


Figure 1. Triscend E5 Customizable MicrocontrollerCustomizable Microcontroller block diagram.

Subject to change. Visit `www.triscend.com` for the latest revision of this document.
Comments, questions, or suggestions on this document? Please send them to mailto:info@triscend.com.

Table 1. Triscend E5 Customizable Microcontroller Family

Device	Embedded Processor Core	Dedicated Resources	System RAM	Configurable System Logic (CSL) Cells	CSI Bus Address Decoders	PIO* Pins (Max)
TE502	Accelerated 8051	High-speed internal bus Memory interface unit 2-channel DMA controller Power management Power-on reset Hardware breakpoint unit JTAG port	8Kx8	256	16	92
TE505	(3) 16-bit counters USART Watchdog timer Interrupt controller		16Kx8	512	32	124
TE512			32Kx8	1,152	72	188
TE520			40Kx8	2,048	128	252

* Maximum PIO on each base device, actual PIO count depends on package style and initialization mode. See Table 45.

- **Embedded Configurable System Logic (CSL) matrix**
  - Fast, flexible CSL logic cells support combinatorial logic, arithmetic, memory, sequential, and bus functions
  - Up to 2,048 CSL cells per device
  - Easy, synchronous access to and from the system bus
  - Programmable intercommunications network between system bus, CSL cells, and programmable I/O (PIO) pins
  - Contention-free bi-directional bussing
- **High-performance, dedicated Configurable System Interconnect (CSI) system bus**
  - 8-bit read and write data, 32-bit address
  - Up to 40 Mbytes/sec transfer rates
  - Simple, synchronous interface to CSL peripherals, seamless connection to the microcontroller
  - Multi-master bus with round-robin arbitration
  - Expandable to off-chip function through memory interface unit (MIU)
  - Flexible on-chip address decoders provide easy access to CSL functions
  - Programmable wait-state support
  - Open standard
  - Forward compatible with future Triscend customizable microcontroller devices
- **Enhanced programmable input/output (PIO) ports**
  - Up to 315 user-programmable I/O per device
  - Inputs, outputs, or bi-directional ports for the microcontroller, dedicated peripherals, or programmable logic peripherals
  - Selectable output drive from 4 mA to 12 mA
  - BusMinder™ circuit provides pull-up, pull-down, or weak-follower capability
  - Optional input hysteresis
  - Optional power-down operation, individually selectable on every pin
  - Input, output and output enable flip-flops for optimal set-up and clock-to-output performance
  - 5 volt tolerant inputs while operating at 3.3 volts
- **Memory interface unit (MIU) for flexible, glue-less interface to external memory**
  - Direct connect interface to an external 256Kx8 memory for initialization and code storage
  - Expandable from 18 up to 32 address lines
  - Variable-speed read/write timing simplifies interface design
  - Access external peripherals by sharing MIU data and address pins
- **Two-channel advanced DMA controller**
  - Proxy bus masters for CSL “soft modules”
  - Up to 40 Mbytes/s transfer rate (1 byte/cycle)
  - Auto-initialization of channels
  - Multiple addressing modes
  - Software-initiated DMA requests
  - Optional interrupt at end of a transfer
  - Block data transfers
  - CRC checking
  - DMA channel request and acknowledge signals distributed to the CSL matrix
- **Programmable power-down modes**
  - Selectively disable function during power-down
  - Typically consumes less than 50 µA in full power-down mode
- **On-chip oscillator, crystal oscillator amplifier, and clock distribution circuitry**
- **Four-pin IEEE 1149.1 JTAG interface port for download and debugging**
  - Supports SAMPLE/PRELOAD, EXTEST, INTEST, BYPASS, and IDCODE instructions

- 8051 reset and Customizable Microcontroller reset
- Full access to the CSI system bus and all addressable locations

- **Multiple in-system programming modes**
  - Unlimited, in-system programmability
  - Byte-wide using standard FLASH, EPROM, or SRAM memories
  - Serially using serial sequential-access PROM memories (SPROMs)
  - Via JTAG using internal system RAM to store program code
  - 'Stealth'-mode operation from internal RAM during battery-backed operation
- **Dedicated in-system debugging, hardware breakpoint unit**
  - Two breakpoint units monitor system address, data, control, and processor instruction type
  - Breakpoint indicator and control from Configurable System Logic (CSL)

Overview

The Triscend E5 Customizable Microcontroller integrates, on a single device, a performance-enhanced Accelerated 8051 embedded microcontroller, a large block of SRAM, a high-speed dedicated system bus, and configurable logic, intimately connected to the processor and system bus. The E5 family is a highly integrated, fully static single-chip system optimized for embedded systems applications. Each member of the E5 family contains an identical microcontroller and set of dedicated resources, as shown in Figure 1. However, the size of the dedicated system RAM, the number of programmable I/O (PIO) pins, and the amount of configurable system logic grows with the larger members of the family, as shown in Table 1.

The embedded high-performance 8051-based "Turbo" microcontroller is instruction-compatible with other industry-standard 8051/52-based devices, leveraging the vast software library for the 8051 architecture. While the instruction cycle for the original 8051 microcontroller is 12 clock cycles, the Accelerated 8051 microcontroller provides better performance because each instruction cycle is only four clock cycles. The result is improved performance at the same clock frequency.

The Accelerated 8051 core offers other advantages over the original 8051. The Accelerated 8051 MCU provides a protected watchdog timer plus an additional data pointer, easing data transfer routines.

The embedded SRAM-based Configurable System Logic (CSL) matrix provides "derivative on demand" system customization. The high-performance configurable logic architecture consists of a highly interconnected matrix of CSL cells. Resources within the matrix provide easy, seamless access to and from the internal system bus.

Each CSL cell performs various potential functions, including combinatorial and sequential logic. The

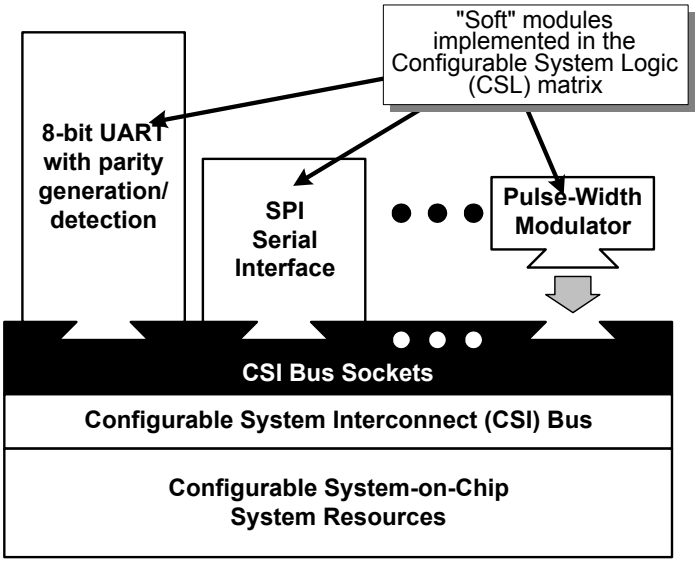


Figure 2. "Soft" modules are built from Configurable System Logic (CSL) resources. Modules connect to the Customizable Microcontroller system via a Configurable System Interconnect (CSI) bus socket, providing easy “drag-and-drop” customization.

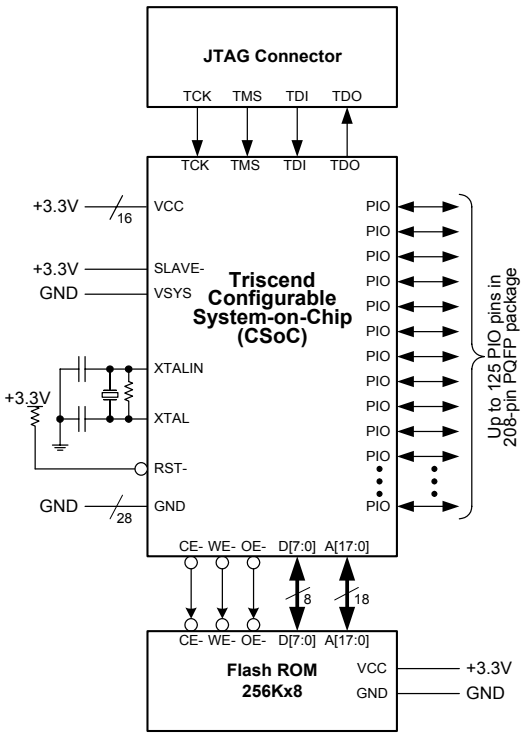


Figure 3. A complete Triscend E520 Customizable Microcontroller design.

combinatorial portion performs Boolean logic operations, arithmetic functions, and memory. The sequential element performs independently or in tandem with the combinatorial function.

The abundant programmable input/output blocks (PIOs) provide the interface between external functions and the internal system bus or configurable system logic. Each PIO offers advanced I/O options including selectable output drive current, optional input hysteresis, and programmable low-power functionality during power-down mode.

A high-performance internal system bus—called the Configurable System Interconnect (CSI) bus—interconnects the microcontroller, its peripherals, and the CSL matrix. The bus provides eight bits of read data, eight bits of write data, and a 32-bit address. Address mapping logic translates the 8051's 16-bit address to the 32-bit address used by the internal system bus.

Multiple masters arbitrate for bus access. Potential bus masters include the Accelerated 8051 microcontroller, the JTAG interface, the read and write channels of each DMA channel, and the memory interface unit (MIU) in some modes of operation. Functions implemented in the CSL matrix can use a DMA channel as a "proxy" master, reusing the control logic already contained in the DMA channels to become a master on the CSI bus.

A memory interface unit (MIU) connects the Triscend E5 Customizable Microcontroller to external memory. The MIU typically connects to an external FLASH-memory device that holds the E5's initialization program plus the user's code. The MIU interface is reusable for connections to other external components. The external read, write control, and chip-select signals are programmable providing flexible set-up, strobe, and hold timing.

The two-channel DMA controller provides high-bandwidth communication between functions, up to 40 Mbytes per second. Its easy-to-use handshake simplifies interface and control logic. Functions from within the CSL matrix can request DMA transfers, the DMA controller providing "proxy" bus mastering capabilities.

A large block of fast, byte-wide SRAM provides internal storage for temporary data storage or for code. Though typically used for data, code can be executed from internal RAM, offering faster access plus security in battery-backed applications.

The majority of the system, including the microcontroller, operates from a single bus-clock signal. Optional sources for the bus clock include driving it directly from an off-chip signal, connecting an external crystal or ceramic oscillator between the dedicated crystal-oscillator amplifier pins, or using the internal ring oscillator. Six other global buffers provide high-fanout signals to CSL functions. The

bus clock and the global buffers can optionally be stopped upon a breakpoint event and shut off during power-down mode.

Power management control provides selectable power-down options over internal functions. Furthermore, each PIO provides pin-by-pin power-down settings.

The E5 Customizable Microcontroller, like other advanced processors, is built from leading-edge static CMOS technology. The E5 device is infinitely in-system programmable. A power-on reset circuit guarantees proper start-up operation after power is asserted. There are various initialization (bootstrapping) modes to support different application requirements. The E5 can load itself automatically after power-on from an external, byte-wide boot memory. Optionally, the E5's configuration data is stored in a serial sequential-access PROM. In serial mode, the user's code is copied to and executed from the internal SRAM. Serial mode frees a number of device pins so that they can be used as user-defined PIO pins.

In security-conscious applications, the user's program is stored in internal RAM and battery-backed using external circuitry. If the E5 Customizable Microcontroller is in 'stealth' mode, it boots from internal RAM when VCC is re-applied after battery back up. Stealth mode optionally disables the JTAG interface port and disables external fetches via the MIU.

An internal initialization boot ROM controls the start of initialization during power-on after the RST-pin is released. The primary purpose of the initialization boot ROM is to find the user's initialization data and code stored in the secondary boot code, usually held in PROM.

Initialization programs can also be downloaded directly to internal SRAM through the JTAG port. Likewise, initialization programs can be written to external flash via JTAG through the MIU interface.

Besides downloading initialization programs, the JTAG port offers nearly full access to the microcontroller, peripherals, and CSL functions to aid in debugging. The JTAG interface can become a bus master on the internal CSI bus. During system debugging, the JTAG port also sets up the internal hardware breakpoint unit.

The hardware breakpoint unit contains two functions that monitor the 8-bit read or write data bus, the 32-bit internal address bus, control signals and the type of processor instruction (code or data access). Upon a predefined set of conditions, the breakpoint unit halts execution of the application program. Via JTAG control, the user can single-step instruction execution of the processor.

Together, the Accelerated 8051 microcontroller, its dedicated peripherals, the on-chip RAM, the internal CSI system bus, and the CSL matrix and PIOs form a powerful, integrated configurable system.

Accelerated 8051 Microcontroller

The Triscend E5 8051-based Customizable Microcontroller is fully instruction set compatible with other industry-standard 8051/8052 microcontrollers. It includes the resources of the standard 8051 including three 16-bit timer/counters; a full-duplex serial port and twelve interrupt sources with three priority levels.

The E5 features a performance-enhanced 8-bit CPU with a redesigned core processor, reducing unnecessary clock and memory cycles. The instruction cycle of a standard 8051 is twelve clock cycles while the Triscend E5 reduces this to four clock cycles for the majority of instructions, thereby improving performance by an average of 1.5 to 3 times.

This naturally speeds up the execution of the instructions. Consequently, the E5 offers more processing power compared to the original 8051, even using the same frequency crystal. For a given throughput, the E5 can be operated from a lower-frequency clock than the original 8051, reducing power consumption.

The E5 also provides dual Data Pointers (DPTRs) to boost block data memory transfers.

Programmable I/O Ports

The original 8051 offers up to four 8-bit ports, a total up to 32 lines. In the E5, the 8051 processor core is embedded with other functions. The processor optionally connects to as many PIO pins as required by the application.

UART

The Triscend E5's UART is a superset of the UART in the original 8051 family, though offers timing compatibility. The UART provides enhanced features such as automatic address recognition and frame error detection.

Timers

The E5's 8051-based microcontroller has three 16-bit timers that are functionally similar to the timers of the original 8051 family. When used as timers, they optionally operate at either 4 clocks or 12 clocks per count, thus providing a mode that emulates the timing of the original 8051.

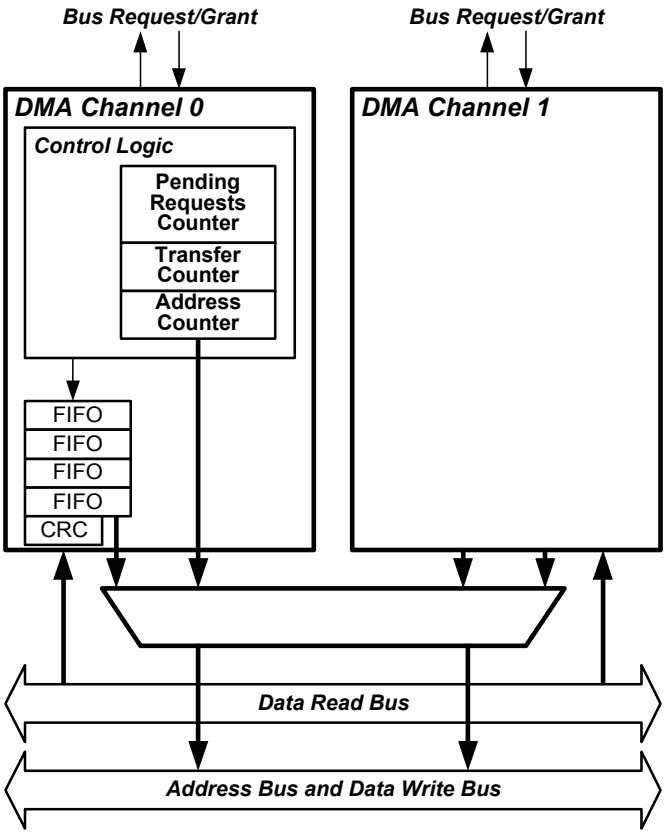


Figure 4. Block diagram of the embedded two-channel DMA controller.

The E5 also features a protected watchdog timer. This timer is used as a system monitor or to time a very long period.

Interrupts

The Interrupt structure in the Triscend Customizable Microcontroller is slightly different from that of the original 8051. Due to the presence of additional features and peripherals, the number of interrupt sources and vectors is increased.

Data Pointers

The original 8051 had only one 16-bit Data Pointer (DPL, DPH). In the E5, there is an additional 16-bit Data Pointer (DPL1, DPH1). This new Data Pointer inhabits two previously unused SFR locations in the original 8051.

Power Management

Like the original 80C31, the E5 provides Idle and Power-Down modes of operation. Idle mode stops the MCU while the timers, serial port and interrupt block continues to operate. Power-Down mode optionally stops all the clocks and completely halts chip operation, the lowest power-consumption state.

Power-On reset

The Triscend Customizable MIcrocontroller has an on-chip Power-On Reset facility. This eliminates the external capacitor-resistor network required in original 8051 designs.

DMA Controller

The Triscend E5's direct memory access (DMA) controller contains two independent channels. Each channel is autonomous from the 8051 microcontroller, freeing the processor from mundane, performance-stealing, data transfer operations. Each channel is designed to transfer a byte of data on each clock cycle, maximizing the data transfer bandwidth. Several different data transfer modes are available including those between memory and a device, in any direction. A device associates itself with a particular channel through a DMA control register (DMACTRL), which contains a request

and acknowledge signal pair. The two DMA channels can also be paired to perform memory-to-memory operations.

The main features of the DMA controller unit are:

- Two independent channels for device or memory transfers
- Transfer rates up to 40 Mbytes/sec
- Auto-initialization of channels
- Programmable transfer parameters
- Multiple addressing modes
- Interrupt capabilities
- Memory-to-memory transfers capabilities
- Block transfers
- Software-initiated DMA requests
- Four-byte FIFO
- Asynchronous request/acknowledge handshake
- CRC on DMA data stream

Functional Description

The DMA unit is composed of two channels as shown in Figure 4. A set of parameters defines the operation for a specific DMA channel, including the memory starting address and starting transfer count, the direction of the transfer, and a variety of other transfer characteristics. Each channel has its own register set. The control logic block contains the address counter, the current transfer count, the pending requests counter and channel control logic. The data FIFO serves as a temporary buffer between the I/O device and memory. The CSI bus arbiter treats each channel as independent bus masters.

DMA Initialization and Termination

The DMA channels are initialized through their programmable registers. Once the transfer parameters have been programmed, the DMA channel must first be enabled and the transfer initialized. Setting the EN bit in the respective DMA_CTRLx_0 register enables the DMA channel. Once enabled, the DMA channel accepts requests. At that point, the DMA accept requests but cannot service them. A transfer is initialized by setting the INIT bit. Upon initialization, the first DMA request loads the start address and start count into their respective counters. The INIT bit is reset by hardware and the DMA controller services the request. Software can detect that a transfer has been initiated when the INIT bit is cleared. The next transfer parameters can be updated and the INIT bit set.

Requests are processed on a cycle basis. Requests lasting more than one clock cycle are interpreted as multiple requests. If the DMA cannot process all the incoming requests, a Pending Request Counter keeps track of such requests. A total of 64K pending requests are possible. If the EN is cleared in the middle of a block transfer, any subsequent requests are ignored by the DMA and by the pending request counter. The counters can be read through software until it is reset. The counters are initialized either by the CLR bit or by enabling the DMA channel.

In case a DMA channel is not responding properly, it is possible to reset it through the CLR bit in the control register. Once the CLR bit is set, the DMA channel is in a reset state. It exits this state only after the CLR bit is cleared. After a power-on reset or other system-wide reset, CLR is set.

A block transfer normally completes when the transfer counter reaches zero. At that point, depending on the mode, the DMA transfer either stops or continues. The DMA continues with the next block transfer if the INIT bit is set, or if continuous mode is enabled. Resetting the DMA channel also terminates a block transfer. The latter method should be used very carefully.

Aborting a DMA transfer

Software can cleanly abort an operation in the middle of a DMA transfer. Once the user detects that the current DMA transfer can be aborted, the DMA channel can be disabled cleanly by clearing the EN bit. The next step is to reset the DMA channel logic, accomplished by setting the CLR bit. Only one clock cycle is necessary to reset the DMA channel logic. Once the CLR is cleared, the DMA channel is again ready to use. Except for a few important control bits (refer to the reset values of each field of the DMA registers), the rest of the configuration registers will have kept their previously programmed values. By writing a few command bits to the control register, the previous transfer could be repeated.

Transfer handshaking

The DMA controller always does the read transfer first and stores the data into its temporary FIFO. Then it performs the write transfer.

Transfer Modes

This section describes the basic DMA transfer types and features. Some of these transfer modes can be combined to form more complex and powerful operations.

Single Transfer Mode

In this mode, the DMA initiates a single byte transfer for each request. If the requests are asserted during every cycle, then the DMA controller attempts to service the requests as fast as it can. The DMA services requests until the transfer count reaches zero. At that point, the transfer is completed. This is the default mode.

Block Transfer Mode

In this mode, a single request initiates a transfer of an entire block of data. Upon receiving the request, the DMA controller starts transferring data until the transfer count reaches zero. If a request is received at any time during the block transfer, the request is recorded in the Pending Requests Counter. The new request is serviced at the end of the current block transfer.

Software Request

When this mode is active, the DMA controller responds to a DMA request initiated from software. Setting the SFTREQ bit in the DMA channel control register enables this mode. Software can then request a DMA transfer. Software requests that cannot currently be served are recorded in the pending request counter. A software request is cleared by hardware on the cycle following the set operation. If the software request is set while the DMA channel is disabled, then the request is ignored.

Single Initialization

Setting the INIT bit initializes a transfer. Upon receiving the first request, the INIT bit is reset by hardware and the single, software or block transfer continues until the transfer count reaches zero. At that point, the DMA controller waits for a new initialization command.

Continuous Auto-Initialization

Setting the CONT initiates a transfer similar to the INIT bit, but upon completion of the current transfer, the DMA controller automatically reinitializes as if the INIT bit were set again. Automatic refresh of some external display is one potential application of this mode.

Memory-to-Memory Transfer

By pairing the channels together, the DMA controller supports memory-to-memory transfers. The channel that reads the data from memory, is the master, and the other channel, which writes the data back into memory, is the slave. Setting the PAIR bit in the control register of ***both*** channels enables this mode. Transfers are initiated using the master's control register. However, the slave channel must be enabled and its transfer parameters set correctly.

Linked transfers

Linked transfers are possible using a single-initialization, software block transfer. The parameters of the first block transfers are programmed into the appropriate control registers and the transfer is initiated through the INIT bit and the SFTREQ bit. Once the INIT bit is cleared by hardware—meaning the DMA channel has initiated the transfer—software loads the address and transfer count parameters of the next block of data. After loading the parameters, software sets the INIT and SFTREQ bits. Upon completion of the first block transfer, the DMA channel loads the new parameters and initiates the new transfer. Software repeats the previous steps until it reaches the end of the linked list.

Bus Address Generation

Each DMA channel generates a memory address for every request. The first address of a block transfer is the starting address, held in starting address control register. This address value is then loaded into the current address counter upon the first request of a block transfer. Once the address is broadcast to memory, it is updated for the next request. The addressing option is configured through two of the DMA channel's control register bits, as shown in Table 2.

For debug purposes, the current address of a block transfer as well as the current count are visible to software.

Data FIFO

The data FIFO serves as a temporary buffer between the requesting I/O device and memory. Because of the CSI bus structure and the multi-master handshaking, four locations are required.

CRC Feature

A cyclic redundancy check (CRC) can be performed on a single DMA stream. The CRC logic monitors the Data Read bus as it enters the DMA's FIFOs. The CRC logic is shared between the two DMA channels and is enabled for either one of the channels by setting the CRC_EN bit in the corresponding DMA channel control register. A 0-to-1 transition on CRC_EN resets the CRC shift-register to zero. Once enabled, the CRC logic is

activated any time a byte of data is written into a FIFO. Once a transfer is completed, the output of the CRC register can be read by software, and the CRC signature can be compared with the expected value.

The CRC logic uses a CRC-CCITT 16-bit divisor polynomial, as shown in the equation below. The algorithm is capable of detecting any one, two or an even number of bits in error as well as a large number of burst errors.

$$X^{16} + X^{12} + X^5 + 1$$

Interrupts Generation

The DMA controller can generate interrupts upon the following three events.

Transfer terminal count: This event is generated when a block transfer is complete (when the transfer counter reaches its terminal count of 0).

Transfer Initialization: This event is generated upon the first request of block transfer if the INIT bit is set.

Pending Request Overflow: This event is generated when the pending request counter overflows, indicating that the DMA controller cannot keep up with the number of incoming requests.

The status of these events is recorded in the interrupt status register, independent of their corresponding interrupt-enable bits. The status bits are reset by software by writing a one into them. Writing a zero does not affect the state of any status bits. Some of the status bits are also cleared by some hardware action (refer to "***DMA Interrupt Register***").

Configuration Registers

Each channel has a set of 21 bytes of control and status registers. Some registers are used to program a specific DMA channel or to query its status.

DMA Start Address Register (one per channel)

This set of registers defines the start address of the DMA transfer. The register values are loaded into the Address Counter at the beginning of a transfer.

The DMA controller operates on 32-bit physical address values, whereas the microcontroller operates on 16-bit logical address values. The physical address for a memory location is typically assigned automatically by the Triscend FastChip development system and included in a header file for the user.

The values are read/writeable.

Channel 0:

DMA Source Address Channel 0 (A[7:0])

7	6	5	4	3	2	1	0
A7	A6	A5	A4	A3	A2	A1	A0

Mnemonic: DMASADR0_0 Address: FF20h

DMA Source Address Channel 0 (A[15:8])

7	6	5	4	3	2	1	0
A15	A14	A13	A12	A11	A10	A9	A8

Mnemonic: DMASADR0_1 Address: FF21h

DMA Source Address Channel 0 (A[23:16])

7	6	5	4	3	2	1	0
A23	A22	A21	A20	A19	A18	A17	A16

Mnemonic: DMASADR0_2 Address: FF22h

DMA Source Address Channel 0 (A[31:24])

7	6	5	4	3	2	1	0
A31	A30	A29	A28	A27	A26	A25	A24

Mnemonic: DMASADR0_3 Address: FF23h

Channel 1:

DMA Source Address Channel 1 (A[7:0])

7	6	5	4	3	2	1	0
A7	A6	A5	A4	A3	A2	A1	A0

Mnemonic: DMASADR1_0 Address: FF34h

DMA Source Address Channel 1 (A[15:8])

7	6	5	4	3	2	1	0
A15	A14	A13	A12	A11	A10	A9	A8

Mnemonic: DMASADR1_1 Address: FF35h

DMA Source Address Channel 1 (A[23:16])

7	6	5	4	3	2	1	0
A23	A22	A21	A20	A19	A18	A17	A16

Mnemonic: DMASADR1_2 Address: FF36h

DMA Source Address Channel 1 (A[31:24])

7	6	5	4	3	2	1	0
A31	A30	A29	A28	A27	A26	A25	A24

Mnemonic: DMASADR1_3 Address: FF37h

DMA Transfer Count Register (one per channel)

This field specifies the byte length of the DMA transfer. ***The actual value is the number of bytes transfer minus one.*** This value is loaded into the transfer counter before the transfer starts.

The values are read/writeable.

Channel 0:

DMA Transfer Count Channel 0 (CNT[7:0])

7	6	5	4	3	2	1	0
CNT7	CNT6	CNT5	CNT4	CNT3	CNT2	CNT1	CNT0

Mnemonic: DMASCNT0_0 Address: FF24h

DMA Transfer Count Channel 0 (CNT[15:8])

7	6	5	4	3	2	1	0
CNT15	CNT15	CNT13	CNT12	CNT11	CNT10	CNT9	CNT8

Mnemonic: DMASCNT0_1 Address: FF25h

DMA Transfer Count Channel 0 (CNT[23:16])

7	6	5	4	3	2	1	0
CNT23	CNT22	CNT21	CNT20	CNT19	CNT18	CNT17	CNT16

Mnemonic: DMASCNT0_2 Address: FF26h

Channel 1:

DMA Transfer Count Channel 1 (CNT[7:0])

7	6	5	4	3	2	1	0
CNT7	CNT6	CNT5	CNT4	CNT3	CNT2	CNT1	CNT0

Mnemonic: DMASCNT1_0 Address: FF38h

DMA Transfer Count Channel 1 (CNT[15:8])

7	6	5	4	3	2	1	0
CNT15	CNT15	CNT13	CNT12	CNT11	CNT10	CNT9	CNT8

Mnemonic: DMASCNT1_1 Address: FF39h

DMA Transfer Count Channel 1 (CNT[23:16])

7	6	5	4	3	2	1	0
CNT23	CNT22	CNT21	CNT20	CNT19	CNT18	CNT17	CNT16

Mnemonic: DMASCNT1_2 Address: FF40h

DMA Current Address Register (one per channel)

This register and Current Count register are used to monitor the current state of a particular DMA channel or for debugging. This register contains the output of the transfer address counter. This register is loaded at the beginning of a transfer and is incremented or decremented at every transfer, controlled by the address mode bits ADRM1 and ADRM0.

The values are read-only.

Channel 0:

DMA Current Address Channel 0 (A[7:0])

7	6	5	4	3	2	1	0
A7	A6	A5	A4	A3	A2	A1	A0

Mnemonic: DMACADR0_0 Address: FF2Bh

DMA Current Address Channel 0 (A[15:8])

7	6	5	4	3	2	1	0
A15	A14	A13	A12	A11	A10	A9	A8

Mnemonic: DMACADR0_1 Address: FF2Ch

DMA Current Address Channel 0 (A[23:16])

7	6	5	4	3	2	1	0
A23	A22	A21	A20	A19	A18	A17	A16

Mnemonic: DMACADR0_2 Address: FF2Dh

DMA Current Address Channel 0 (A[31:24])

7	6	5	4	3	2	1	0
A31	A30	A29	A28	A27	A26	A25	A24

Mnemonic: DMACADR0_3 Address: FF2Eh

Channel 1:

DMA Current Address Channel 1 (A[7:0])

7	6	5	4	3	2	1	0
A7	A6	A5	A4	A3	A2	A1	A0

Mnemonic: DMACADR1_0 Address: FF3Fh

DMA Current Address Channel 1 (A[15:8])

7	6	5	4	3	2	1	0
A15	A14	A13	A12	A11	A10	A9	A8

Mnemonic: DMACADR1_1 Address: FF40h

DMA Current Address Channel 1 (A[23:16])

7	6	5	4	3	2	1	0
A23	A22	A21	A20	A19	A18	A17	A16

Mnemonic: DMACADR1_2 Address: FF41h

DMA Current Address Channel 1 (A[31:24])

7	6	5	4	3	2	1	0
A31	A30	A29	A28	A27	A26	A25	A24

Mnemonic: DMACADR1_3 Address: FF42h

DMA Current Count Register (one per channel)

This is the output of the transfer count counter. It indicates how many bytes are left in the current transfer.

The values are read-only.

Channel 0:

DMA Current Count Channel 0 (CNT[7:0])

7	6	5	4	3	2	1	0
CNT7	CNT6	CNT5	CNT4	CNT3	CNT2	CNT1	CNT0

Mnemonic: DMACCNT0_0 Address: FF2Fh

DMA Current Count Channel 0 (CNT[15:8])

7	6	5	4	3	2	1	0
CNT15	CNT15	CNT13	CNT12	CNT11	CNT10	CNT9	CNT8

Mnemonic: DMACCNT0_1 Address: FF30h

DMA Current Count Channel 0 (CNT[23:16])

7	6	5	4	3	2	1	0
CNT23	CNT22	CNT21	CNT20	CNT19	CNT18	CNT17	CNT16

Mnemonic: DMACCNT0_2 Address: FF31h

Channel 1:

DMA Current Count Channel 1 (CNT[7:0])

7	6	5	4	3	2	1	0
CNT7	CNT6	CNT5	CNT4	CNT3	CNT2	CNT1	CNT0

Mnemonic: DMACCNT1_0 Address: FF43h

DMA Current Count Channel 1 (CNT[15:8])

7	6	5	4	3	2	1	0
CNT15	CNT15	CNT13	CNT12	CNT11	CNT10	CNT9	CNT8

Mnemonic: DMACCNT1_1 Address: FF44h

DMA Current Count Channel 1 (CNT[23:16])

7	6	5	4	3	2	1	0
CNT23	CNT22	CNT21	CNT20	CNT19	CNT18	CNT17	CNT16

Mnemonic: DMACCNT1_2 Address: FF45h

DMA Control Register (one per channel)

Channel 0:

7	6	5	4	3	2	1	0
W/R-	PAIR	BLOCK	SFTREQ	CONT	INIT	EN	CLR

Mnemonic: DMACTRL0_0 Address: FF27h

7	6	5	4	3	2	1	0
Reserved					CRC_EN	ADRM1	ADRM0

Mnemonic: DMACTRL0_1 Address: FF28h

Channel 1:

7	6	5	4	3	2	1	0
W/R-	PAIR	BLOCK	SFTREQ	CONT	INIT	EN	CLR

Mnemonic: DMACTRL1_0 Address: FF3Bh

7	6	5	4	3	2	1	0
Reserved					CRC_EN	ADRM1	ADRM0

Mnemonic: DMACTRL1_1 Address: FF3Ch

All values are read/writeable.

CLR, when set, disables the DMA channel and clears the transfer counter and pending requests counter. Clearing the bit indicates that the DMA channel is ready to operate. Cleared by reset.

EN, when set, enables the DMA request for the channel. When this bit is set, the pending requests counter is cleared and the DMA channel is ready to accept requests. When this bit is cleared, incoming requests are ignored. Cleared by reset.

INIT, when set, initializes a DMA transfer. An initialization is associated with the transfer of each block transfer. When this bit is set, the starting operation values are loaded into their corresponding counters at the beginning of a transfer. Once the transfer has started, the bit is cleared by hardware. Software can set it again during the current block transfer to prepare the DMA channel for the next block. When cleared, the DMA stops after the current transfer. Cleared by reset.

CONT, when set, indicates continuous initialization mode. When set, the DMA transfer on the current block continues until this bit is cleared. Cleared by reset.

SFTREQ, when set, requests a software-initiated DMA transfer. This bit is cleared by hardware on the following clock cycle.

BLOCK, when set, indicates block request mode. When set, a complete block transfer is performed by the DMA upon receiving a single request from a device. Unaffected by reset.

PAIR, when set in both channels, couples the two DMA channels to perform memory-to-memory transfers. Unaffected by reset.

W/R- indicates the direction of the DMA transfer. When set, the DMA performs a memory-to-I/O transfer (DMA Write). When clear, the DMA performs an I/O-to-memory transfer (DMA Read). Unaffected by reset.

ADRM1 and ADRM0 define the transfer address mode as shown below. Unaffected by reset.

Table 2. DMA Address Mode Settings.

ADRM1	ADRM0	Mode
0	0	Increment address by one after each byte transfer
1	0	Decrement address by one after each byte transfer
x	1	Single address transfer (address remains constant)

CRC_EN enables CRC checking. When set, this bit activates the CRC logic. A 0-to-1 transition on this bit resets the CRC logic to 0.

Cleared by a power-on reset or other device-wide reset.

DMA Pending Requests (one per channel)

This field indicates the number of DMA requests yet to be serviced. Up to 64K requests can be received ahead of their corresponding acknowledge.

Cleared by a power-on reset or other device-wide reset.

The values are read-only.

Channel 0:

DMA Pending Requests Channel 0 (REQ[7:0])

7	6	5	4	3	2	1	0
REQ7	REQ6	REQ5	REQ4	REQ3	REQ2	REQ1	REQ0

Mnemonic: DMAPREQ0_0 Address: FF32h

DMA Pending Requests Channel 0 (REQ[15:8])

7	6	5	4	3	2	1	0
REQ15	REQ14	REQ13	REQ12	REQ11	REQ10	REQ9	REQ8

Mnemonic: DMAPREQ0_1 Address: FF33h

Channel 1:

DMA Pending Requests Channel 1 (REQ[7:0])

7	6	5	4	3	2	1	0
REQ7	REQ6	REQ5	REQ4	REQ3	REQ2	REQ1	REQ0

Mnemonic: DMAPREQ1_0 Address: FF46h

DMA Pending Requests Channel 1 (REQ[15:8])

7	6	5	4	3	2	1	0
REQ15	REQ14	REQ13	REQ12	REQ11	REQ10	REQ9	REQ8

Mnemonic: DMAPREQ1_1 Address: FF47h

DMA Interrupt Enable Register (one per channel)

The DMA interrupt register enables individual interrupt events for each channel.

Channel 0:

7	6	5	4	3	2	1	0
Reserved					OVR_EN	INIT_EN	TC_EN

Mnemonic: DMAEINT0 Address: FF29h

Channel 1:

7	6	5	4	3	2	1	0
Reserved					OVR_EN	INIT_EN	TC_EN

Mnemonic: DMAEINT1 Address: FF3Dh

All values are read/writeable.

TC_EN, when set, enables the DMA interrupt to indicate that the transfer counter reached its terminal count (TC).

INIT_EN, when set, enables the DMA interrupt upon initialization.

OVR_EN, when set, enables the DMA interrupt to indicate that the pending requests counter exceeded 64K.

Cleared by a power-on reset or other device-wide reset.

DMA Status Register (one per channel)

Channel 0:

7	6	5	4	3	2	1	0
Reserved					OVR	INIT	TC

Mnemonic: DMAINT0 Address: FF2Ah

Channel 1:

7	6	5	4	3	2	1	0
Reserved					OVR	INIT	TC

Mnemonic: DMAINT1 Address: FF3Eh

TC, when set, indicates that the transfer counter reached terminal count. This bit is cleared when the INIT bit is set. This bit is set by hardware and is cleared by software by writing a one. Writing a zero has no effect. Used in conjunction with TC_EN to flag an interrupt.

INIT, when set, indicates that initialization has occurred. This bit is set by hardware and cleared by software by writing a one. Writing a zero has no effect. This bit is also cleared when software sets the INIT bit in the corresponding channel's DMA control register. Used in conjunction with INIT_EN to flag an interrupt.

OVR, when set, indicates that the pending requests counter has overflowed. This bit is set by hardware and cleared by software by writing a one. Writing a zero has no effect. This bit is also cleared when the corresponding channel's pending requests counter is cleared. Used in conjunction with OVR_EN to flag an interrupt.

NOTE:

The OVR, INIT, and TC bits in the DMA Status Register are cleared by writing a '1' to their respective bit location. Writing a zero has no effect.

Cleared by a power-on reset or other device-wide reset.

DMA Unit CRC Output Register

Once a transfer is completed, the output of the CRC register can be read by software, and the signature can be compared with the expected value.

DMA CRC Register (CRC[7:0])

7	6	5	4	3	2	1	0
CRC7	CRC6	CRC5	CRC4	CRC3	CRC2	CRC1	CRC0

Mnemonic: DMACRC_0 Address: FF48h

DMA CRC Register (CRC[15:8])

7	6	5	4	3	2	1	0
CRC15	CRC14	CRC13	CRC12	CRC11	CRC10	CRC9	CRC8

Mnemonic: DMACRC_1 Address: FF49h

This register is shared by both DMA channels.

Cleared by a power-on reset or other device-wide reset.

Interfacing CSL Peripherals to the DMA Controller

"Soft" modules implemented in the Configurable System Logic matrix have full access to DMA services. Access is provided via distributed DMA control registers that steer DMA requests from peripherals to the appropriate DMA channel and steer the appropriate DMA channel acknowledge signal back to the peripheral, as shown in Figure 5.

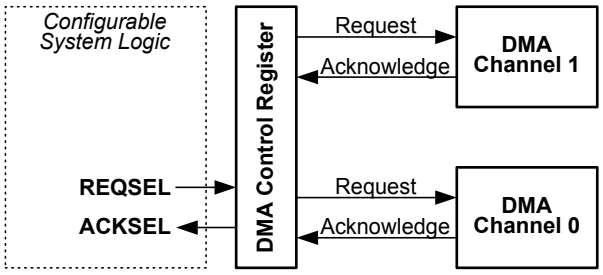


Figure 5. DMA Control Registers steer control signals from CSL "soft" modules to the DMA channels.

DMA control registers share the same programmable address selector functions also used for address decoding and chip selects. See Table 3 for the number of selectors available in each device.

The address for a DMA control register is programmable, similar to any function using a Selector. A symbolic address name for the specific DMA control register is provided during design. The actual address assignment is usually left to the Triscend FastChip development system. All DMA control registers are single-byte registers and must be located within data or SFR memory spaces.

An individual DMA control register controls a unidirectional DMA transfer, *i.e.,* a memory-to-I/O (DMA write) or an I/O-to-memory (DMA read) transaction. However, a DMA control register is associated with a specific DMA channel by changing the SEL bit within the control register. Two DMA control registers are required for DMA read and DMA write operations from the same peripheral.

A DMA control register is enabled for DMA access by setting the ENBL bits. Until enabled, all DMA requests (REQSEL) from a DMA control register are ignored.

In standard use, only one DMA control register should be enabled per channel, per direction at any time, for a maximum of four. These four control registers cover individual DMA read and write operations from both channels 0 and 1. Another DMA control register can be enabled via software after first disabling the active control register.

Distributed DMA Control Register

7	6	5	4	3	2	1	0
-	-	SEL	ENBL	-	-	SEL	ENBL

Write both nibbles with duplicate data.
Read register and OR bits from each nibble.

Mnemonic: User-Defined Address: User Defined

Undefined bit locations are reserved and return 0 when read.

NOTE: *The distributed DMA Control registers only connect to either the high nibble or the low nibble of the CSI data bus. Consequently, application code should write duplicate copies of the high and low nibble. When reading, only the high or the low nibble will contain valid data. The high and low nibbles should be ORed together to determine the actual settings.*

SEL steers control signals to and from the DMA controller. If SEL is cleared, the REQSEL input and ACKSEL output signals are steered to DMA channel 0. If SEL is set, then the signals are steered to DMA channel 1. The SEL control bit must be written to both bits 5 and 1. When read, both bits must be ORed together.

ENBL, when set, allows the CSL "soft" module to access the DMA controller through the REQSEL input and ACKSEL output associated with DMA Control register. When ENBL is cleared, the DMA controller ignores any requests from the CSL module and the CSL module ignores any DMA acknowledges. The ENBL control bit must be written to both bits 4 and 0. When read, both bits must be ORed together.

A DMA control register is reset to 00h by a reset.

There is unrestricted read/write access to this register.

DMA Request Steering

To request a DMA transfer, the CSL "soft" module function asserts the REQSEL input on the DMA control register, as shown in Figure 6. If enabled (ENBL=1), the REQSEL signal is forwarded to appropriate DMA channel request, depending on the SEL value. If SEL=0, then REQSEL requests channel 0, else REQSEL requests channel 1. If disabled (ENBL=0), the request is blocked.

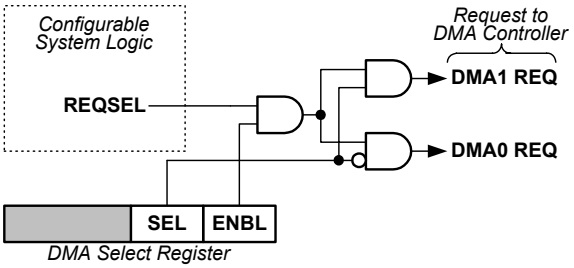


Figure 6. A DMA Control Register steers DMA requests to the appropriate channel.

DMA Acknowledge Steering

Once a "soft" module requests a DMA transfer, control over the transaction shifts to the DMA channel. The DMA channel requests the CSI bus. Once granted, the DMA channel asserts is acknowledge signal, which is steered back to the requesting CSL "soft" module via the ACKSEL output. The control logic is shown in Figure 7.

When ACKSEL is asserted High, the CSL module should respond appropriately. During a DMA read (I/O-to-memory transfer), the CSL module should present data on the Data Read bus when ACKSEL is asserted. During a DMA write (memory-to-I/O transfer), the CSL module should accept the data on the Data Write bus when ACKSEL is asserted.

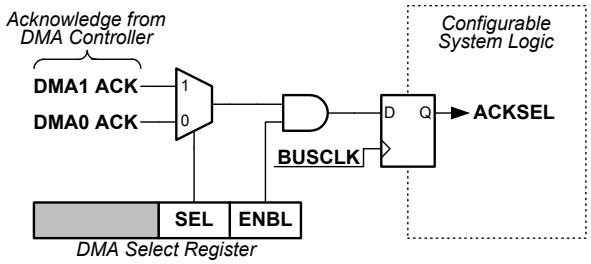


Figure 7. DMA acknowledge signals are steered back to the requesting CSL "soft" module.

The requesting device must be ready to accept the ACKSEL signal in a single cycle, since wait state capabilities are reserved strictly for memory-mapped operations. The DMA controller is designed to maximize bus bandwidth.

Example DMA Write Transaction

Figure 8 shows the waveform for a typical DMA write operation, *i.e.*, a memory-to-I/O transfer. Prior to the first request, one of the DMA channels is configured for a DMA write operation and the proper values loaded into its configuration registers.

Likewise, a DMA control register—part of a CSL "soft" module function—is enabled (ENBL=1) and the channel select bit is set to steer signals to the proper DMA channel (SEL=0 for channel 0, SEL=1 for channel 1).

The remainder of the transaction is as shown Figure 8.

1. The requesting "soft" module asserts its REQSEL DMA request signal. Within the DMA control register, this incoming request is steered to the proper DMA channel.
2. The DMA channel requests the CSI bus from the bus arbiter. This process may require multiple clock cycles.
3. Once the bus arbiter grants the bus to the DMA controller, the DMA presents the transfer address, the write data, and asserts its acknowledge signal. Within the DMA control register, the DMA channel's acknowledge signal is steered back to the requesting CSL "soft" module function. Consequently, the ACKSEL signal is asserted, signaling the "soft" module that data is available. The "soft" module uses the ACKSEL signal to enable a register and capture the value presented on the Data Write bus.

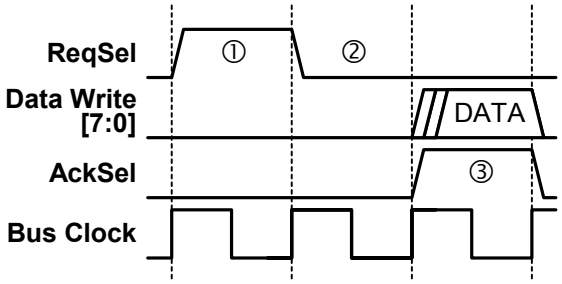


Figure 8. Example DMA Write operation.

Using the DMA Controller as a Proxy Bus Master

"Soft" modules implemented in the CSL matrix are bus slaves, unable to request and control the bus by themselves. However, a CSL module can use the dedicated DMA controller as a "proxy" bus master.

A bus master requires interaction with the bus arbiter, registers and counters to track addresses, and state machines to handle the bus transfer protocol. Such logic would consume CSL logic resources, if direct bus mastering were supported. A more efficient approach is to re-use the dedicated resources already built into the DMA controller.

The DMA controller contains the logic to arbitrate for the bus, track address, and control transfers. A CSL “soft” module uses the DMA controller as a "proxy" master, causing the DMA controller to arbitrate for and control the bus transaction.

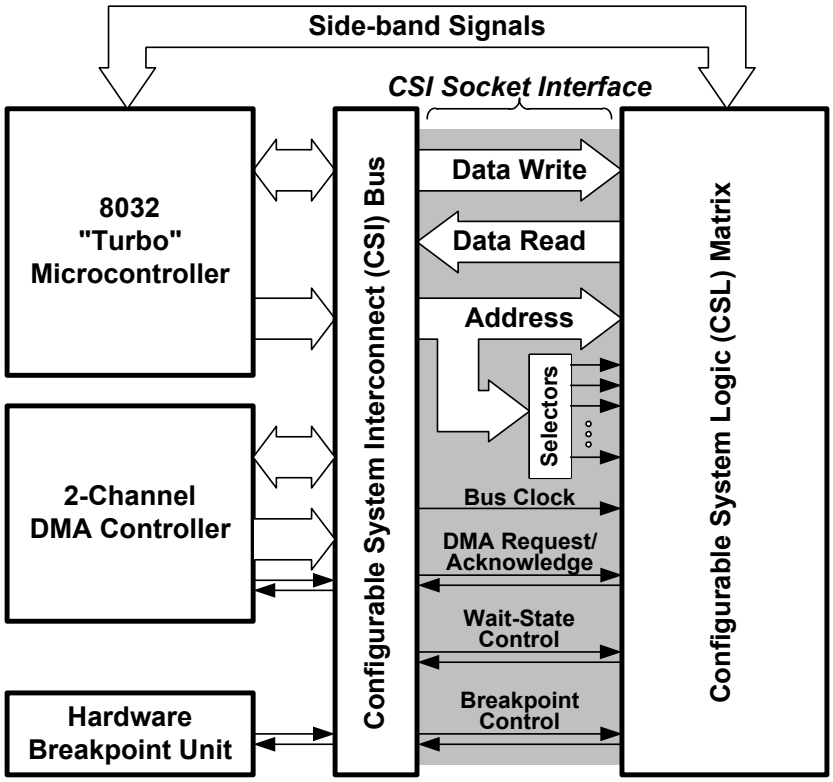


Figure 9. The Configurable System Interconnect (CSI) bus and the socket interface to "soft" modules in the CSL matrix.

Configurable System Interconnect (CSI) Bus

The Configurable System Interconnect (CSI) bus, shown in Figure 9, bridges the microcontroller to its peripherals and functions implemented in the Configurable System Logic (CSL) matrix. The CSI bus provides a processor-independent migration path to future generations of customizable microcontroller devices. User-defined and library-provided "soft" modules can be moved to future Triscend Customizable Microcontroller families with little or no modification

The CSI bus socket provides a simple, synchronous interface to functions implemented in the CSL matrix, as shown in Figure 10. The CSI bus interface socket consists of the following signals.

- An 8-bit write data port
- An 8-bit read data port, including a read enable signal to enable the read data back onto the CSI system bus.
- A 32-bit address port.
- A set of address selector (chip select) functions. The number of selectors varies according to device size as shown in Table 3. The selectors optionally steer DMA request and acknowledge signals to and from the CSL matrix.
- The bus clock.
- Wait-state control and monitor signals.
- Hardware breakpoint control and monitor signals.

Data Read Bus

Data is presented to the 8-bit read bus when the selected CSL logic function or "soft" module asserts its read enable signal. The read data values from all functions are logically OR-ed together. Consequently, all unselected CSL functions drive the read bus with zeroes. Only the selected function presents ones. Read data can be presented during every active bus cycle.

Data Write Bus

The eight data write bits are presented on the CSI socket. Write data may be presented during every active bus cycle.

Address Bus

All 32 address bits are presented via the CSI interface socket. Typically, only a few, if any, of the address signals are used by functions in the CSL matrix. Typically, the actual decoding of the ad-

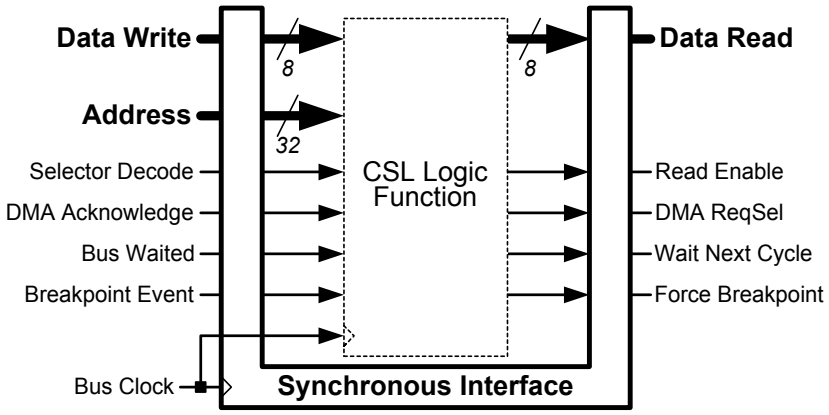


Figure 10. The CSI bus socket represents a simple-to-use, synchronous interface to custom logic functions implemented in the CSL logic matrix.

dress bus is performed using the address selector resources.

Address Selectors

The CSI socket interface practically eliminates the need to use any CSL matrix resources to decode bus transactions. One of the more important elements in the CSI socket interface is the programmable address decoder function, generically called a selector. A selector performs functions much like a chip select unit or an address decoder built in a PAL-type device.

Address Selector Operation

A selector detects a transaction to a specified range of CSI bus addresses, including the particular address space. A selector decodes the full 32-bit address bus. If a transaction is to its target address range, the selector asserts one of its outputs on the same clock edge that the system provides write data and address, synchronized to the system clock. This approach dramatically simplifies the logic used to access CSL “soft” modules.

The number of available selectors depends on the particular device. The number of selectors grows with the increasing size of the CSL matrix. There is one selector for every sixteen CSL cells, as shown in Table 3.

Table 3. Number of selectors by device.

Device	Selectors
TE502	16
TE505	32
TE512	72
TE520	128

Functionally, each selector is similar to diagram shown in Figure 11. Each selector contains two 32-bit registers that define the target address. The MATCH0 register defines which address bits match when the individual bit is Low. The MATCH1 register defines which address bits match when the bit is High. If the same bit location is set in both registers, then the corresponding address bit is a "don't care."

If each address bit matches with values defined in the MATCH0 and MATCH1 register, then the selector further decodes read or write operations. If the address matches, and there is a read transfer to this location, then the selector asserts its RDSEL output. Likewise, if the transfer is a write, then the selector asserts its WRSEL output.

Address Specification

The MATCH0 and MATCH1 register values are automatically defined by the Triscend FastChip development system at design time. These register values are not changed by application software.

The addresses loaded into MATCH0 and MATCH1 are symbolically defined by the user during hardware design. The user specifies

- The ***symbolic name*** for the address range. This is the name used in application software to refer to the target address range.
- The ***size*** of the addressed range, which must be a power of two, ranging from '1' indicating a single byte to '16M' indicating a 16M byte region.
- The ***address space*** or spaces to which the selector should respond. In conjunction with the address mapper, a selector detects transactions to a specific address space. The E5, based on a standard 8051 microcontroller, supports the following address spaces.
  - DATA, which corresponds to the 8051's XDATA memory space
  - SFR (Special Function Register), which is a 128-byte region unique to the 8051 architecture.

The Triscend FastChip development system examines these settings from all of the address selectors defined in the hardware design. It then allocates physical addresses to each selector. The corresponding 8051 logical addresses are defined and specified in a header file, used with the user's application code.

The MATCH0 and MATCH1 values, defined by the Triscend FastChip development system, are loaded into the selectors during the initialization process. FastChip also defines the values initially loaded into the address mapper registers.

Address Selector Modes

An address selector performs one of three potential functions as shown in Table 4.

Table 4. Address Selector Types.

Select Modes	Ports	Function
Selector	RDSEL	Read decode
	WRSEL	Write decode
Chip Select	RDSEL	R/W- control
	SEL	Chip select
DMA Control	REQSEL	DMA request
	ACKSEL	DMA acknowledge

Selector

A selector separately decodes read and write operations to the target address range, as shown in Figure 11. The RDSEL output indicates a read operation, the WRSEL output indicates a write operation.

A selector function is called SELECT in the design library.

Chip Select

A selector in the chip select mode decodes any read or write transaction to the target address range, read or write. Figure 12 shows a functional drawing of a chip select function. The SEL performs like a chip select function. The RDSEL output is asserted only during read operations.

A selector function is called CHIPSEL in the design library.

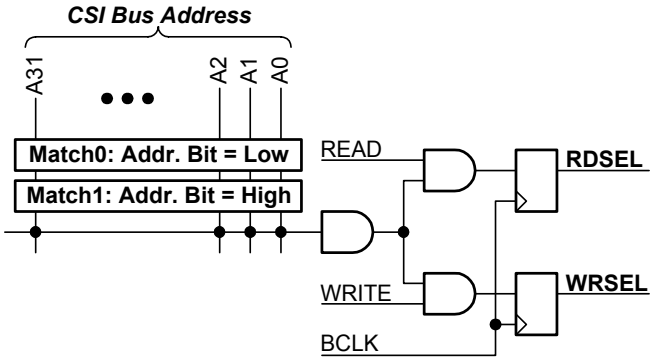


Figure 11. The distributed address selector functions decode read and write transactions to a target address range. The selectors eliminate the need to build decoding logic using CSL resources.

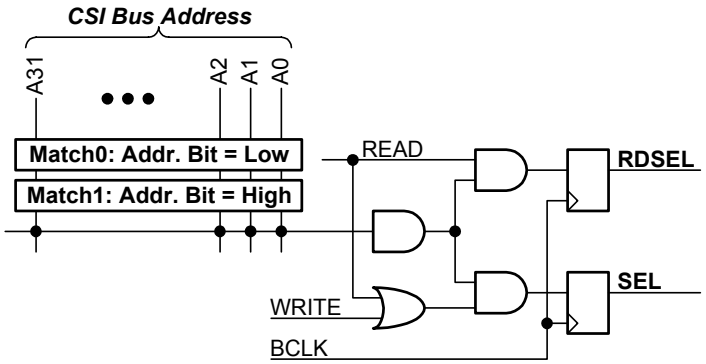


Figure 12. In chip select mode, an address selector decodes any read or write transaction to the target address range.

DMA Control

A selector provides a re-locatable control register for CSL "soft" modules requiring DMA access. The DMA control register enables requests and steers the request and acknowledge signals to the selected DMA channel. DMA control register. See the "*DMA Controller*" section for more information.

External Access through MIU

A "soft" module can access an external device using PIO pins or by re-using the pins on the memory interface unit (MIU). A special selector mode allows a CSL function to access and handle a transaction through the MIU, as shown in Figure 13.

Wait-State Monitor and Control Signals

The CSI socket interface includes signals to monitor and control wait-states on the internal system bus.

WAITED (output from bus socket)

The WAITED signal indicates that a wait-state was asserted during the previous bus cycle. This signal is typically used in control logic for functions such as FIFOs and dual-port memory.

WAITNEXT (input to bus socket)

Some CSL "soft" modules require wait-states if they are too slow to respond in a single bus cycle. If any wait-states are required, an address selector must assert the initial wait-state. However, the WAITNEXT sideband signal itself, when asserted and valid on the rising edge of Bus Clock, always causes a wait-state in the following CSI bus cycle, regardless of the selector.

If a "soft" module requires more than one wait-state, it must assert the WAITNEXT signal before the next rising clock edge on Bus Clock. When asserted, the WAITNEXT signal causes a wait-state on the next bus cycle.

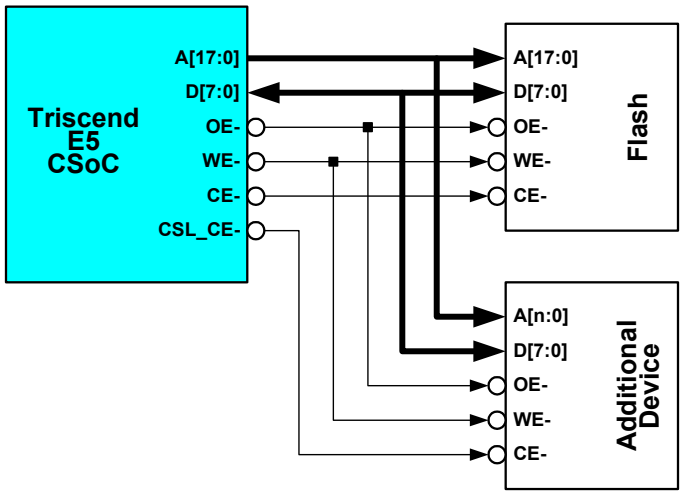


Figure 13. Other external devices share the E5's Memory Interface Unit (MIU) signals. The chip enable for the Flash connects directly to the E5's CE- pin. All other devices use a separate Chip Select function, located in the CSL matrix.

Initial Wait-State Insertion

Some “soft” modules implemented in the CSL matrix may require wait-states, either because the “soft” module handshakes with another asynchronous device or if the “soft” module is too slow to respond in a single bus cycle.

If a “soft” module requires any wait-states, a selector must assert the first wait-state. The selector will only assert a wait-state if the system is accessing the selector's target address space.

Should a “soft” module require additional wait-states beyond the initial wait-state asserted by the selector, then the “soft” module indicates additional wait-states by asserting the WAITNEXT signal on the CSI socket interface.

Breakpoint Event Monitor and Control Signals

The CSI socket interface includes signals to monitor and control hardware breakpoint events. These signals can be used to aid system-level debugging.

BREAK (input to bus socket)

CSL functions can force a hardware breakpoint event by asserting the BREAK signal. The hardware breakpoint unit typically only monitors transactions on the CSI bus. The BREAK signal allows CSL functions to interact with the hardware breakpoint unit.

For example, a CSL function could be monitoring a serial communications stream that rarely interacts with the system bus. Upon detecting a particular pattern, the CSL function could force a breakpoint event, stopping the system. The state of the system or CSL functions could then be monitored through the JTAG port.

EVENT (output from bus socket)

CSL functions can monitor hardware breakpoint events using the EVENT signal. When EVENT is asserted, a hardware breakpoint event has occurred, either caused by the hardware breakpoint event or by another function in the CSL matrix.

CSI Bus Transactions

The following section describes example CSI bus transactions, demonstrating the interaction of a CSL “soft” module and the CSI bus socket.

Data Write Transactions

During a data write transaction, the system sends data to a CSL “soft” module. The system presents both write data and address.

Figure 14 shows a single-cycle write transaction to a CSL logic function. Write data and address are presented on the CSI socket interface. The address is decoded using a Selector. If the transaction is to the Selector's address range, then the Selector asserts its WRSEL signal. The CSL “soft” module uses WRSEL to enable a register and capture the write data.

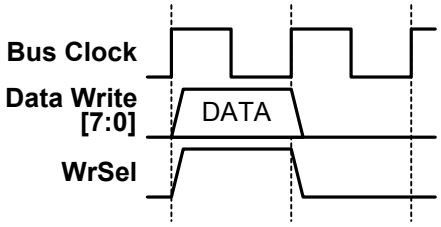


Figure 14. A single-cycle write transaction to a CSL “soft” module.

Figure 15 demonstrates a similar transaction, except that the CSL logic function requires two clock cycles to capture the write data. The CSL logic function's Selector is configured to assert the initial wait-state. When the Selector detects that the system is addressing it, it asserts its WRSEL signal and automatically asserts the initial wait-state.

During the wait-state, the bus master continues presenting write data and address and the Selector continues to assert its WRSEL output. The CSL function will capture the write data during the second bus cycle so it does not assert WAITNEXT. The WAITED signal indicates the wait-state inserted by the Selector during the first bus cycle. The transaction ends after the second bus cycle and the Selector de-asserts its WRSEL output.

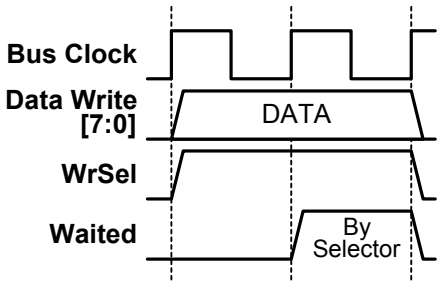


Figure 15. A two-cycle write transaction to a CSL “soft” module.

Data Read Transactions

During a data read transaction, the system presents the read address and the targeted CSL function presents the data.

Figure 16 shows a single-cycle read transaction from a CSL function. The read address is presented on the CSI socket interface. The address is decoded using a Selector. If addressed, then the Selector asserts its RDSEL signal. The CSL function uses RDSEL to enable data onto the Data Read output port on the CSI socket.

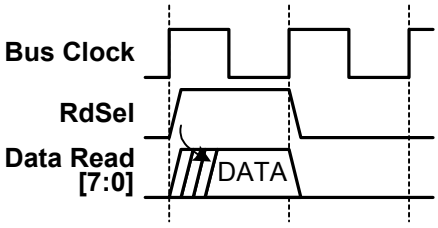


Figure 16. A single-cycle read transaction from a CSL "soft" module.

Figure 17 demonstrates a similar transaction, except that the CSL logic function requires two clock cycles to provide the read data. The CSL logic function's Selector is configured to assert the initial wait-state. When the Selector detects that the system is addressing it, it asserts its RDSEL signal and automatically asserts the initial wait-state.

During the wait-state, the bus master continues presenting address and the Selector continues to assert its RDSEL output. The CSL function will present valid read data during the second bus cycle so it does not assert WAITNEXT. The WAITED signal indicates the wait-state inserted by the Selector during the first bus cycle. The transaction ends after the second bus cycle and the Selector de-asserts its RDSEL output.

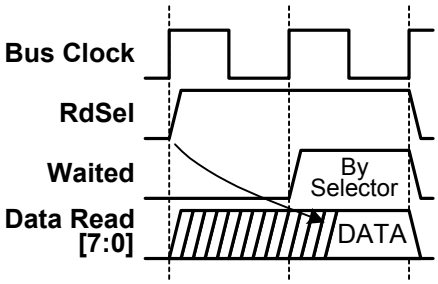


Figure 17. A two-cycle read transaction from a CSL "soft" module.

Using WAITNEXT during a transaction

There are four general rules for asserting a CSI bus wait-state, depending on the CSL function's response time.

1. If the CSL logic function can respond within a single bus cycle, the no wait-states are required.
2. If the CSL logic function can respond by the second bus cycle, then it associated Selector must insert a single wait-state.
3. If three or more bus cycles are required before the CSL logic function can respond, then the Selector is configured to insert the initial wait-state and the CSL function must insert additional wait-states using the WAITNEXT signal.
4. If any wait-states are required on one side of a transaction, either read or write, then the other side of the transaction requires at least one wait-state. The initial wait-state inserted by a Selector occurs on both read and write transactions.

Figure 18 shows an example transaction where the selected CSL function insert wait-states using the WAITNEXT signal. During the first bus cycle, the addressed CSL function's Selector asserts it RDSEL output. In this example, the function always requires at least one wait-state, so the Selector inserts the initial wait-state. The CSL function is not ready to respond in the first bus cycle, so the function asserts WAITNEXT, inserting a wait state during the next bus cycle.

During the second bus cycle, the system continues providing address and the Selector continues asserting RDSEL. The CSL function determines that it is ready to respond during the next bus cycle and de-asserts WAITNEXT. The WAITED signal shows the wait-state inserted by the Selector during the first bus cycle.

Finally, during the third bus cycle, the CSL logic function is ready to respond. The CSL logic function provides data on the Read Data port on the CSI socket. The WAITED signal shows the wait-state inserted when the CSL function asserted WAITNEXT. The wait-state occurred on the second bus cycle but WAITED reports the wait-state on the following bus cycle.

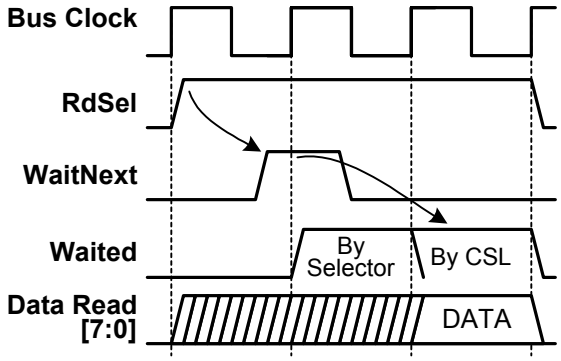


Figure 18. A multi-cycle transaction using WAITNEXT to insert wait-states.

Side-band Signals

All of the signals on the CSI socket interface are designed to be processor independent. "Soft" modules designed using this interface can be reused with other Triscend-compatible Customizable Microcontroller families.

However, some signals are processor specific. The signals are called "side-band" signals. For the Triscend E5 family, these side-band signals include 8051-specific functions.

On a traditional 8051, some of the microcontroller's PIO pins have shared functionality. For example,

the Timer 0 External Input, T0, typically shares a PIO pin on Port 3 (P3.4). On the E5, however, these processor-specific signals can connect to any PIO pin and even to functions implemented entirely within the Configurable System Logic (CSL) matrix.

The side-band signals for the E5 Customizable Microcontroller device family are shown in Table 5. Refer to ***Sideband Signal Timing Characteristics*** for specific timing requirements.

Table 5. E5 Family Side-band Signals.

E5 Function	8051 Signal
Timer 0 external input	T0
Timer 1 external input	T1
External I/O for Timer 2	T2
Timer/Counter 2 capture/reload trigger or an additional external interrupt source if Timer 2 baud-rate generator unused	T2EX
External Interrupt 0	INTR0
External Interrupt 1	INTR1
High-Priority Interrupt	HPINT
Serial port receive data input. Used in modes 1, 2, and 3.	RXDIN
Serial port receiver output. Used in serial port mode 0 for shift clock.	RXDOUT
Serial port transmit data	TXD
Application reset from CSL matrix (RSTC), inverted polarity from the original 8051	RST-
Signal from the 8051 to the CSL matrix (CPURST). Indicates that the 8051 was reset for any reason, including the watchdog timer.	(no 8051 equivalent signal)
Output of the crystal oscillator amplifier, distributed only to CSL functions.	XTAL

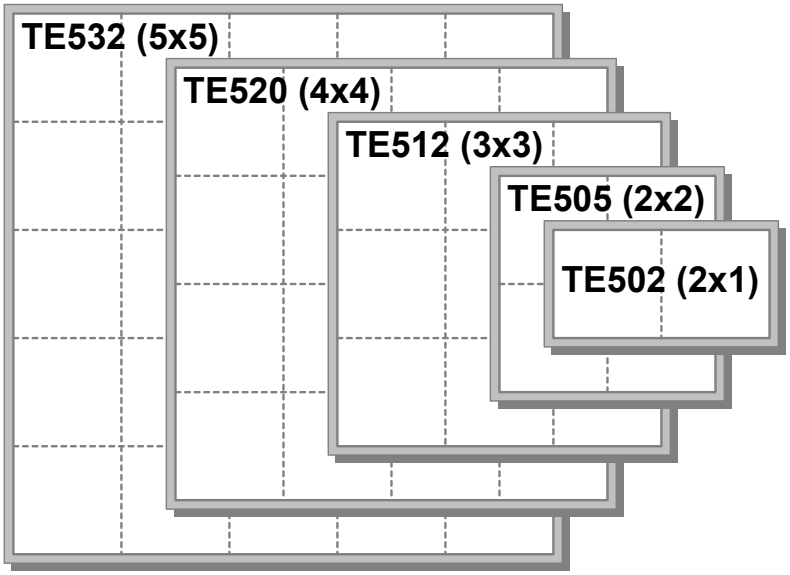


Figure 19. The five member of the E5 Customizable Microcontroller devic family range in density from two CSL banks (256 CSL cells) up to 25 CSL banks (3,200 CSL cells).

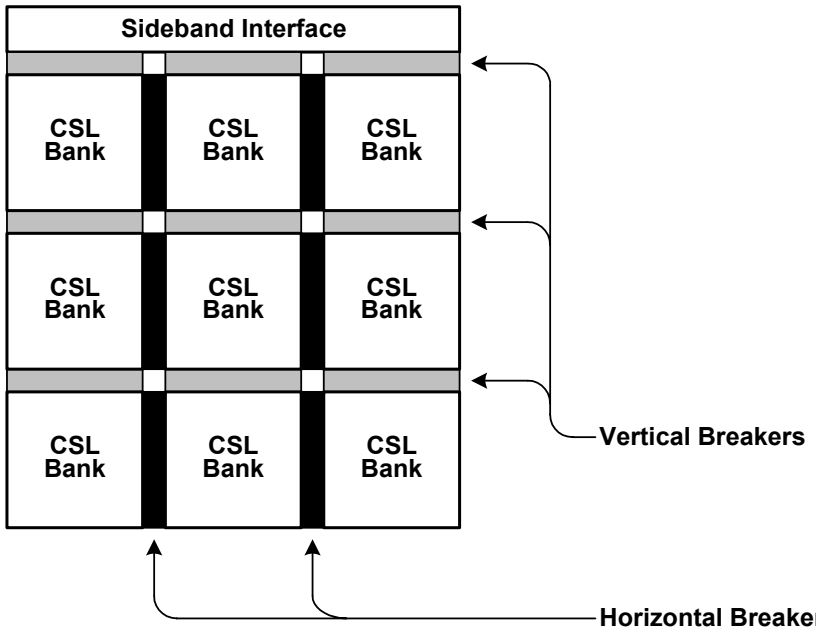


Figure 20. Vertical and horizontal breakers separate the individual CSL banks with Configurable System Interconnect (CSI) bus resources.

Configurable System Logic (CSL)

The Configurable System Logic (CSL) matrix provides flexible, programmable resources to build almost any digital function. Because it is intimately connected to the CSI bus, the CSL matrix is ideal for building any "soft" module functions required by the MCU. The matrix consists of multiple CSL banks. Each bank is a rectangular array of individual CSL cells.

The number of CSL banks varies by part number. The highest-density E5 family device, the TE520, contains 16 CSL banks, arranged in a 4x4 array as shown in Figure 19 and Table 6. The smallest member, the TE502, has just two CSL banks, arranged in a 2x1 array.

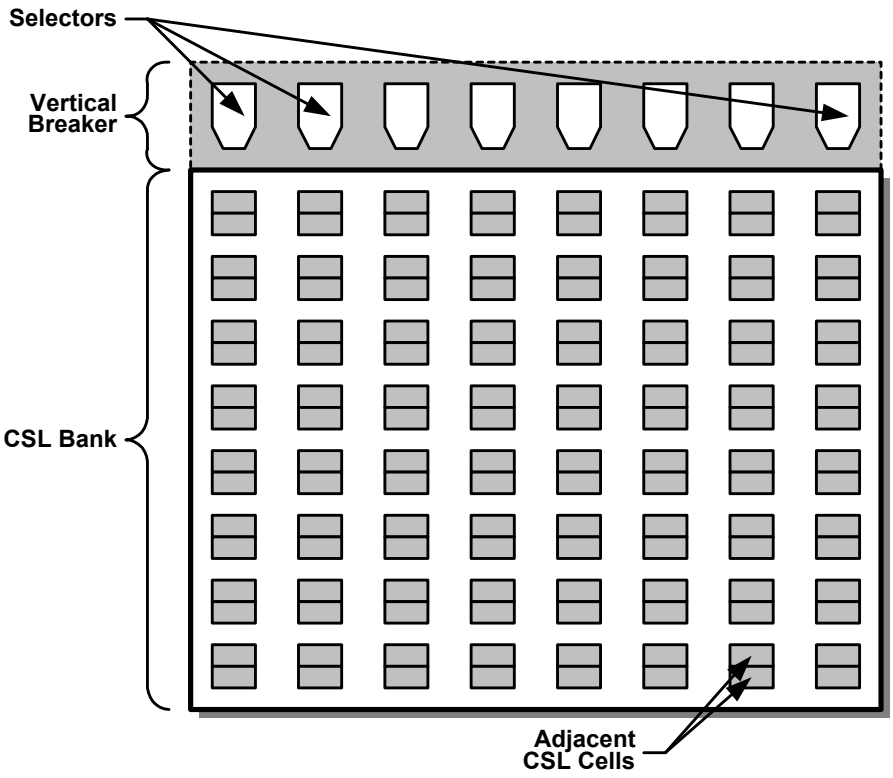


Figure 21. A CSL bank consists of 8 columns by 16 rows of CSL cells, a 128 in total.

Table 6. CSL Banks by Device.

Part Number	*CSL Banks*			Total Cells
	Columns	Rows	Total	
TE502	2	1	2	256
TE505	2	2	4	512
TE512	3	3	9	1,152
TE520	4	4	16	2,048

Vertical and horizontal breakers separate the individual CSL banks on a device, as shown in Figure 20. Vertical breakers appear at the top of every CSL bank. Horizontal breakers appear between adjacent columns of CSL banks. The breakers contain Configurable System Interconnect (CSI) bus resources. The horizontal breakers distribute CSI bus address signals to the CSL banks. The vertical breakers distribute the Selector input and output signals, breakpoint control signals, the global buffer signals, and the wait-state control signals. The CSI read data return path is also located in the vertical breakers.

Signals from one CSL bank can cross into other banks via the breakers, though crossing a breaker adds delay to the signal.

Sideband signals originate and terminate in resources along the top edge of the device.

Bank Resources

Each CSL bank consists of 8 columns by 16 rows of CSL cells, totaling 128 CSL cells per bank. Figure 21 shows the basic layout of the cells within a bank. Pairs of adjacent CSL cells share resources to build more complex cell functions. The address selectors, located in the vertical breaker above the bank, distribute any decoded address. There is one address selector per column of 16 CSL cells.

Programmable interconnect surrounds the CSL cells. These programmable "wires" allow a signal originating from one CSL to communicate with one or more CSL cells, even those in other CSL banks. Likewise, signals to and from the CSI bus are distributed to and from individual CSL cells.

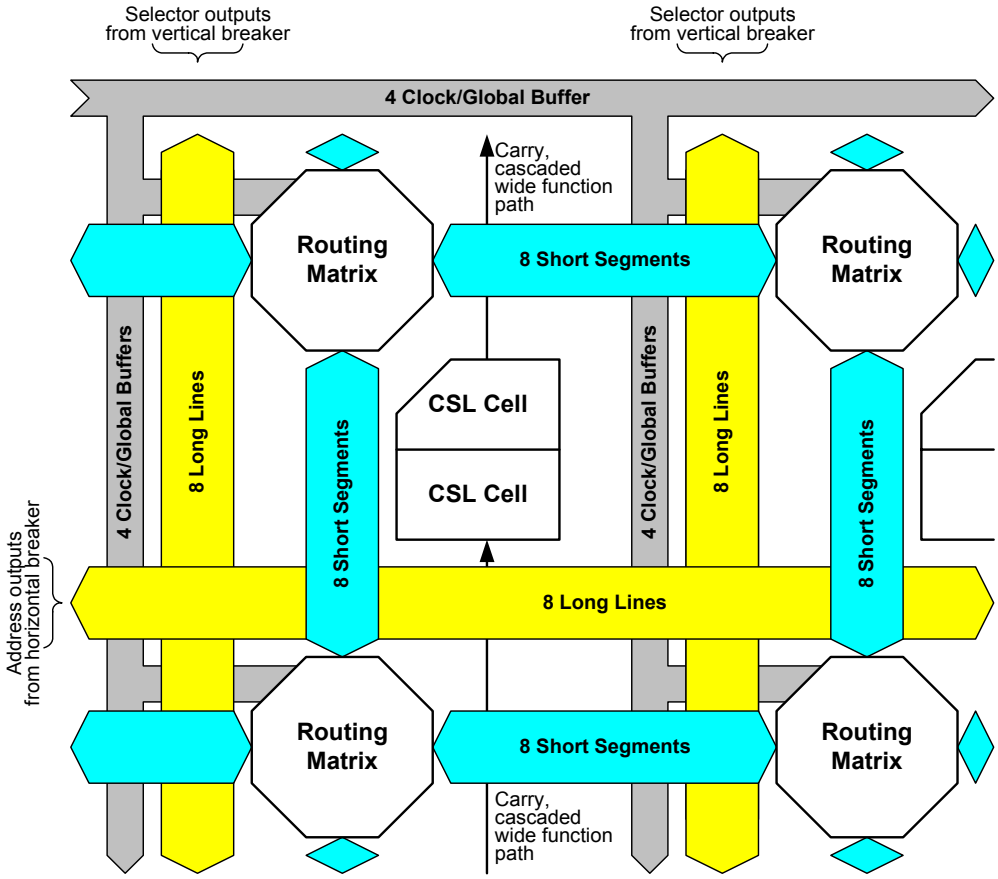


Figure 22. The general-purpose interconnect surrounds a pair of adjacent CSL cells.

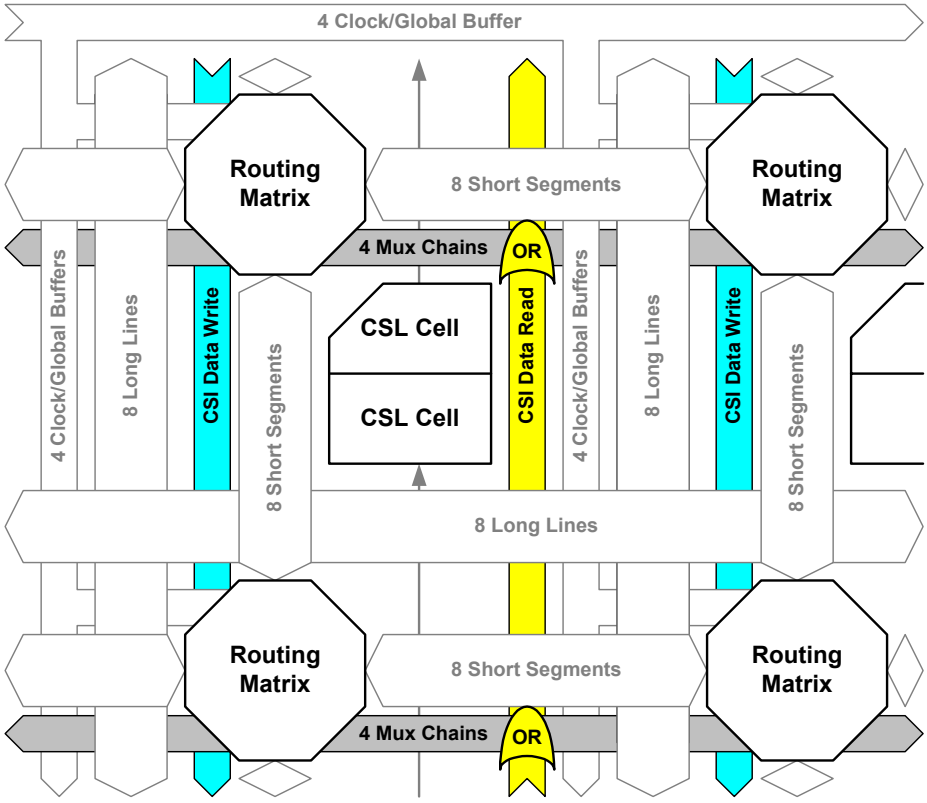


Figure 23. CSI bus write data is available at each routing matrix. Read data returns to the CSI bus.

General-purpose Interconnect

The general-purpose interconnect, shown in Figure 22, distributes signals within a CSL bank. Metal lines of various lengths and purposes connect to individual CSL cells, to the horizontal and vertical breakers, and to the distributed array of routing matrices. Each routing matrix provides connections between the various lines entering or exiting the segment. The various interconnect resources are described below.

- **8 Short Segment** lines in each vertical and horizontal channel.
- **8 Long Lines** in each vertical and horizontal channel. These long lines traverse the width or breadth of the CSL bank. The vertical long lines optionally distribute the outputs from the Selectors located in the vertical breaker. The horizontal long lines optionally distribute address signals from CSI bus.
- **Bus clock and 3 of 6 global buffer signal lines** in every vertical channel. The bus clock signal is distributed globally to all resources on an E5 Customizable Microcontroller device. Within a CSL bank, any three of the six global buffer signals are available.

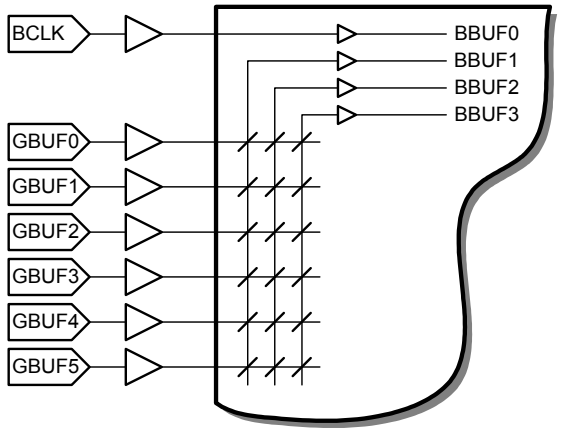


Figure 24. The bus clock signal and any 3 of the 6 global buffer signals are available within a CSL bank.

- **A carry/cascade signal** between adjacent pairs of CSL cells, for faster arithmetic functions and for wide logic functions.

CSI Bus Read and Write Data Distribution

Beyond the general-purpose signals, the programmable interconnect also distributed data signals to and from the CSI bus.

- **CSI Write Data** is accessible at every routing matrix, distributed throughout the CSL bank.
- **4 Multiplexer Chains** for distributing bidirectional data across a CSL bank. The multiplexer chains behave much like a bidirectional, three-state bus but avoids the potential data-contention problems and associated power consumption of a three-state bus because all signals are unidirectional.
- **CSI Read Data** paths gather the values of individual data lines from throughout the device. Ultimately, all the signal return to the CSI bus. The signals from individual bit lines are gathered via an OR-chain.

Signal-Flow Directional Preferences

Though the interconnect was designed to minimize any directional signal-flow preferences, there are few biases inspired by the architecture, as shown in Figure 25.

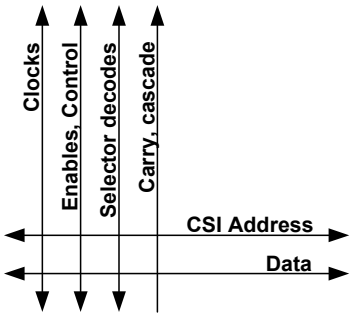


Figure 25. The interconnect architecture inspires a few signal-flow biases.

Clock signals prefer the vertical channels, either to take advantage of the four clock buffers available within a CSL bank or the direct connections between the CSL cell's clock input and the vertical long lines.

Likewise, other control signals and enable signals prefer vertical channels. Addresses decoded using Selectors also prefer vertical channels because vertical long lines distribute these signals from the vertical breakers.

Wide arithmetic functions or wide logic functions benefit from the built-in carry/wide interconnect, which prefers a vertical orientation, from bottom to top. Other orientations are possible, though with decreased performance.

Individual CSI bus address signals are distributed using the horizontal long lines and consequently prefer horizontal channels.

The multiplexer chains, designed to distribute bidirectional data, traverse a CSL bank horizontally. As a result, data flow prefers the horizontal channels.

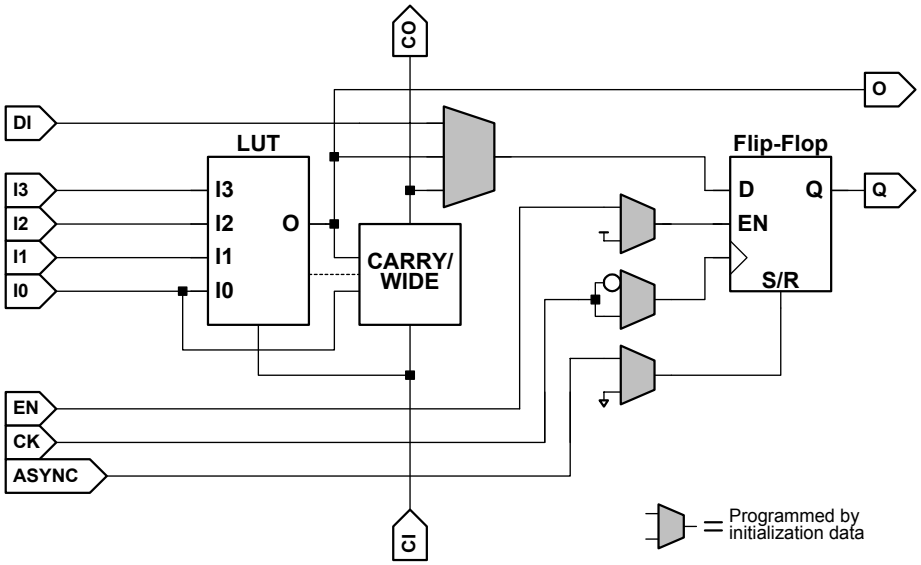


Figure 26. A basic CSL cell of both combinatorial and sequential logic.

CSL Cell Capabilities

A CSL cell, as shown in Figure 26, consists of a flip-flop plus combinatorial functions capable of performing various logic, arithmetic, or memory operations. Both these resources operate independently or in tandem, depending on the specific function implemented. An individual CSL cell is capable of implementing the following types of functions.

- Logic
- Arithmetic
- Memory
- Bus
- Sequential

Most logic functions are implemented using the CSL cell's four-input look-up table (LUT4). Any four-input, single-output function fits within a single four-input LUT, regardless of it logical complexity. A special mode allows two adjacent CSL to implement a five-input LUT (LUT5) and some logic functions of up to nine inputs.

The shaded multiplexers in Figure 26 represent data flow options, defined by the initialization data loaded into the device at power-on or after asserting the RST- pin.

Logic Functions

In logic mode, an individual CSL cell performs a variety of combinatorial functions of the available inputs, as shown in Table 9. A single CSL cell performs any possible combinatorial function of four or less inputs, regardless of complexity. Likewise, two CSL cells working in tandem implement any possible function of five inputs. Two CSL cells also implement some functions of between six to nine inputs, with limitations. A sequence of four- or five-input functions, chained together, create wide gate functions of practically any width. The performance of various logic functions is shown in Table 7

Table 7. Example logic functions and their performance.

Function	CSL Cells	Performance
Four-input XNOR	1	< 5 ns
Compare two 16-bit values for equality (X=Y)	8	< 20 ns

Arithmetic Functions

In arithmetic mode, a CSL cell performs simple arithmetic functions such as add, subtract, or multiply. Various functions, as shown in Table 10, provide common structures for building adders, subtracters, comparators, accumulators, incrementers, decrementers, binary counters, multipliers, and other arithmetic-based operations. Table 8 shows the performance of various arithmetic functions.

Table 8. Example arithmetic functions and their performance.

Function	CSL Cells	Performance
16-bit binary, loadable up counter	16	> 40 MHz
16-bit adder/subtracter (X-Y)	16	< 25 ns
16-bit comparator (X≥Y)	16	< 25 ns

Table 9. Logic functions implemented in a CSL cell.

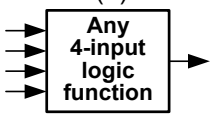
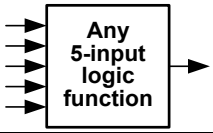
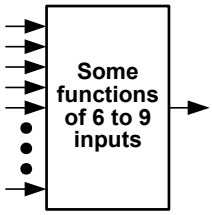
Class	Function	CSL Cells	Application
Logic	f(4) 	1	Any combinatorial logic function with four or less inputs. Includes any mixture of AND, NAND, OR, NOR, XOR, XNOR, and INVERT.
	f(5) 	2	Any combinatorial logic function with five inputs. Includes any mixture of AND, NAND, OR, NOR, XOR, XNOR, and INVERT.
		2	Some combinatorial logic functions of between six to nine inputs. Includes a mixture of AND, NAND, OR, NOR, XOR, XNOR, and INVERT.

Table 10. Arithmetic functions implemented in a CSL cell.

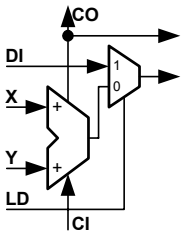
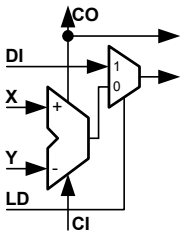
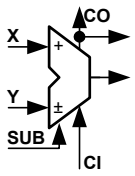
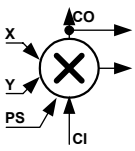
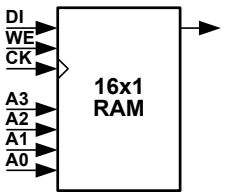
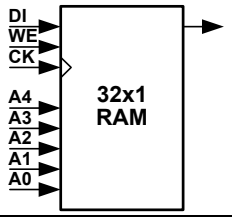
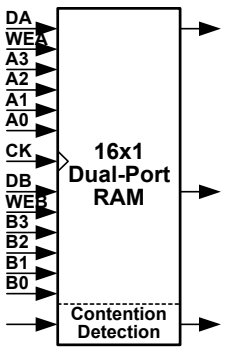
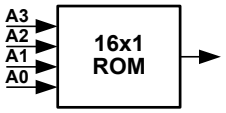
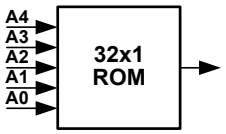
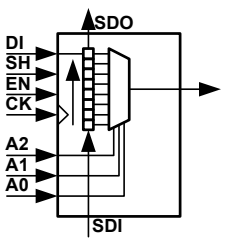
Class	Function	CSL Cells	Application
Arithmetic	ADD 	1	A one-bit full adder where SUM=X+Y. The output of the adder can be multiplexed with another input via the load (LD) input. Useful for building adders, incrementers, accumulators, and binary up counters.
	SUB 	1	A one-bit full subtracter where SUM=X-Y. The output of the subtracter can be multiplexed with another input via the load (LD) input. Useful for building subtracters, decrementers, comparators, and binary down counters.
	ADDSUB 	1	A one-bit full adder/subtracter where SUM=X±Y, depending on the SUB control. The SUB input controls whether the function is X+Y or X-Y. Useful for building adder/subtracters, incrementer/decrementers, and binary up/down counters.
	MULT 	1	A one-bit multiply function. By combining MULT functions with ADD functions, large multipliers can be created.

Table 11. Memory functions implemented in a CSL cell.

Class	Function	CSL Cells	Application
Memory	RAM16x1 	1	A 16-deep by one-bit wide, clocked write, random-access memory (RAM).
	RAM32x1 	2	A 32-deep by one-bit wide, clocked write, random-access memory (RAM).
	RAMDUAL 	2	A 16-deep by one-bit wide, clocked write, dual-port RAM supporting simultaneous read and write operations from both ports. Also includes write contention circuitry to detect simultaneous write operations to the same location with different data.
	ROM16x1 	1	A 16-deep by one-bit wide read-only memory (ROM).
	ROM32x1 	2	A 32-deep by one-bit wide read-only memory (ROM).
	SHIFT8 	1	An 8-bit serial-in/serial-out shift register with selectable output tap and shift/load control.

Memory Functions

In memory mode, a CSL cell performs various memory functions, including single- and dual-ported RAM, read-only memory (ROM), and an 8-bit serial-in/serial-out shift register. The small amount of RAM inside each CSL cell is ideal for building small register files and FIFOs. Should larger quantities of RAM be required, the E5's large system RAM block provides fast and efficient storage.

Single-Port RAM (RAM16X1, RAM32X1)

As a single-port RAM, a CSL cell provides

- Four address inputs for a 16x1 RAM block, five address inputs for a 32x1 block.
- A data input
- An active-High write enable input
- An invertible write clock input

The initial contents of the RAM can be pre-defined and loaded at power-up during initialization. All write operations are synchronized to the clock input, simplifying the timing relationship of the data, address, and write-enable signal. Also, there are no hold time requirements for any of the RAM inputs after the active clock edge.

Read operations are asynchronous and depend only on the address inputs.

Dual-Port RAM (RAMDUAL)

In dual-port RAM mode, two CSL cells provide true dual-port capabilities, supporting simultaneous read and write operations from both ports. As a dual-port RAM, two CSL cells offer

- Two, four-input address input ports
- Two data input ports
- Two active-High write enable input ports
- A single shared, invertible write clock input
- A daisy-chained error monitor that detects simultaneous write operations to the same address with different data

The initial contents of the RAM can be pre-defined and loaded at power-up during initialization. All write operations are synchronized to the clock input, simplifying the timing relationship of the data, address, and write-enable signal. Also, there are no hold time requirements for any of the RAM inputs after the active clock edge.

Read operations are asynchronous and depend only on the address inputs.

ROM (ROM16X1, ROM32X1)

The ROM function is available in two versions; a 16-deep by 1-bit wide ROM (ROM16X1) and a 32-deep by 1-bit ROM (ROM32X1)

A 16-deep ROM provides four address lines to address the 16 memory locations. Likewise, a 32-deep ROM provides five address lines. The value on the address lines directly affects the output value.

A 16x1 ROM consumes a single CSL cell while a 32x1 requires two CSL cells operating in tandem.

The ROM's initial contents are specified in the user's design, loaded during initialization, and cannot be changed during operation.

8-bit Shift Register (SHIFT8)

Another operating mode offered by a CSL cell is an 8-bit, serial-in/serial-out shift register. Serial data arrives on the SDI input. When Low, the shift/load signal, SH, loads data on DI into the shift register location specified by the address lines, A[2:0]. All other shifting halts.

When SH is High, the SDI data is shifted in the first register location. Likewise, all subsequent register values are shifted one position toward the most-significant location (Location 7). The value in location 7 appears on the SDO output. The address lines, A[2:0] select the register location presented on the asynchronous output, O.

The enable signal, EN, disables all shifting and loading operations when Low.

Sequential Functions

Each CSL cell contains a ‘D’-type flip-flop. The flip-flop provides the following controls

- An optional active-High clock enable input
- An invertible, edge-triggered clock input
- An optional asynchronous input to preload the flip-flop to set or clear the flip-flop, defined in the user’s design.

During the initialization process, each flip-flop is loaded with a ‘1’ or ‘0’ as defined in the design. This value is protected against potential spurious writes until the end of the initialization process.

Table 12. Sequential Functionality.

D	EN	CK	ASYNC	Q
During initialization				INITV
X	X	X	1	ASYNCV
X	0	X	0*	Q
D	1*	↑	0*	D

INITV = Initial value, loaded during initialization, defined in user's design.

ASYNCV = Asynchronous preload value, defined in user's design.

0* = Active Low, default value if left unconnected

1* = Active High, default value if left unconnected

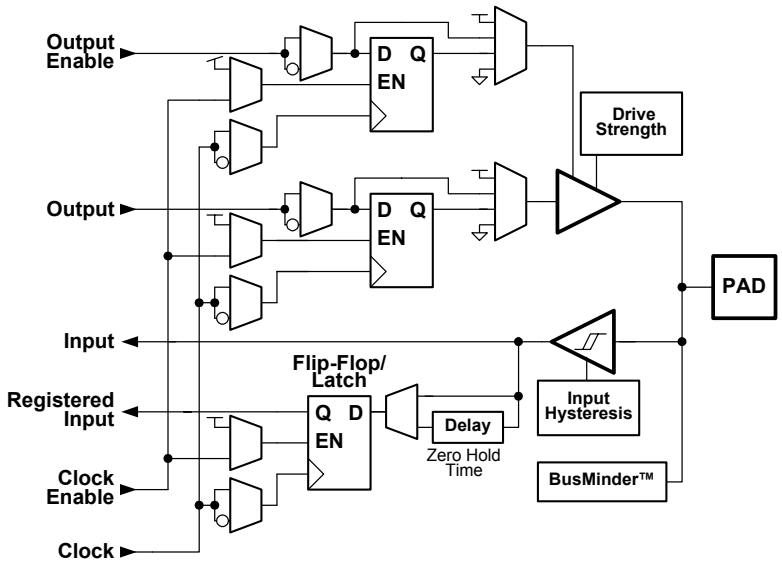


Figure 27. Programmable Input/Output block (PIO).

Programmable Input/Output (PIO) Pins

Programmable input/output blocks (PIOs) interface external package pins to internal functions such as the microcontroller, its peripherals, and the CSL matrix. Each PIO connects to a bonding pad, which may or may not attach to an external package pin. Each PIO can be configured an input, an output, or a bi-directional signal, as shown in Figure 27.

Creating a PIO Port for the Microcontroller

Unlike the original 8051, the Triscend E5 offers nearly unparalleled I/O flexibility. Each of the customizable microcontroller's PIO pins optionally connects to the processor via the CSI bus or works independently in unassociated CSL logic functions.

Connecting a group of PIOs to the processor requires CSI socket resources, including connections to the Data Read and Data Write busses, plus one or two address selectors.

The original 8051 has up to four byte-wide programmable I/O ports. The external memory interface typically consumes two of the 8051's ports (Ports 0 and 2), leaving just two byte-wide ports or 16 individual signals. Furthermore, some port signals have shared functionality. For example, the Timer 1 external input (T1) shares a Port 3 pin (P3.5) on the original 8051. Compared to other microcontrollers, the Triscend E5 offers nearly unparalleled I/O capabilities.

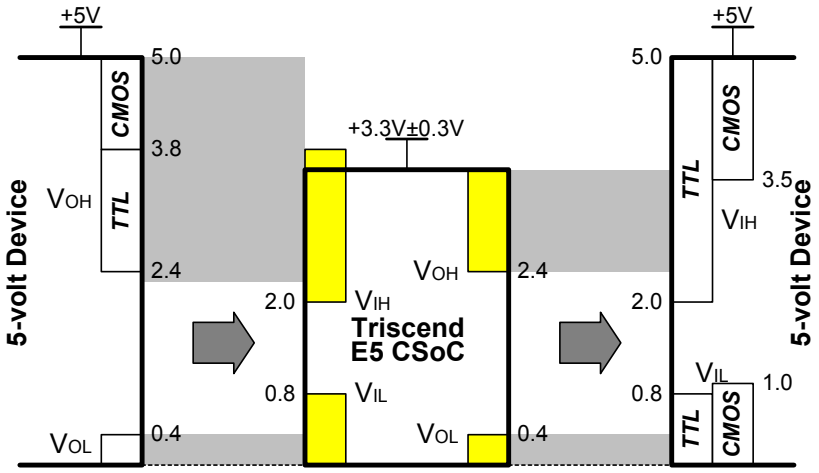


Figure 28. Triscend E5 Customizable Microcontroller devices interface directly with nearly all 5-volt devices, even though the E5 is powered from a 3.3 volt supply.

First, the E5 has a dedicated external memory interface, separate from the programmable I/O ports. Furthermore, there are no shared PIO pins, unless so specified in the users design. Consequently, an E5 design may have as many as 315 PIO pins, depending on the device and package offering. The user determines the number of ports addressable by the processor.

Process and design technologies have advanced significantly since the introduction of the original 8051. Consequently, the Triscend customizable microcontroller family offers advanced I/O capabilities not found in the original 8051. These features include, power-down operation, selectable drive strength, optional input hysteresis, programmable three-state operation, and low-voltage operation.

5-Volt Tolerant I/Os

The PIOs on the Triscend E5 are fully 5-volt tolerant even though the I/O supply voltage is 3.3 volts. This allows 5 volt signals to connect directly to the E5 inputs without damage, as shown in Figure 28. In addition, the 3.3-volt VCC can be applied before or after 5 volt signals are applied to the I/Os. This makes the E5 immune to power supply sequencing problems.

The E5 Customizable Microcontroller device accepts either TTL or CMOS inputs from a 5-volt device. Built-in over-voltage protection prevents any latch up or potential device damage caused by applying a 5-volt signal to a 3.3-volt device.

Likewise, the E5 device outputs drive valid TTL levels for a 5-volt device. The majority of 5-volt devices, including 5-volt CMOS parts, have TTL-compatible inputs.

The only potential voltage-standard mismatch occurs when the E5 drives a pure 5-volt CMOS device with CMOS-only inputs. However, CMOS devices with CMOS-only inputs are typically used for lower power applications. Switching to a 3.3-volt CMOS device cuts power consumption nearly in half and eliminates any potential voltage-standard mismatch.

Storage Elements

There are three storage elements in each PIO:

1. An input flip-flop/latch
2. An output flip-flop
3. An output-enable flip-flop

The functionality of each storage element is defined in Table 13. Note that only the input register can be used as a latch. Each is a 'D'-type element and all share a common clock-enable and clock control input. An individual flip-flop is either permanently enabled or selectively enabled using the common clock-enable input.

Table 13. Flip-Flop/Input Latch Operation (no optional inversions).

Mode	D	ClkEna	Clock	Q
During initialization				INITV Value
Flip-flop	D	1*	↑	D
	X	X	0	Q
Input Latch	X	1*	1	Q
	D	1*	0	D
Both	X	0	X	Q

X = don't care

1* = logic High (default value)

↑= rising clock edge (no clock inversions)

INITV = preloaded initialization value defined in user's design

The polarity of the common clock signal is individual controlled for each flip-flop within a PIO.

A user-defined initial value of a register is loaded during the configuration process.

PIO Input Side

Input signals flow into the device through two possible paths. The input path is a direct logical input while the other is a registered input programmed as either an edge-triggered flip-flop or a level-sensitive latch.

Optional Input Switching Hysteresis

Each PIO input has optional input hysteresis. When enabled, there is about ±150mV of hysteresis, centered around the input switching voltage, based on the I/O supply voltage (VIO).

Registered Input

QA is the registered input. Unlike the other storage elements, the input register can be configured either as an edge-triggered flip-flop or as a level-sensitive latch as shown in Table 13.

Guaranteed Zero Hold Time on Input Register

The data input to the input register can optionally be delayed by several nanoseconds. With the delay enabled, the setup time of the input flip-flop is increased to negate the clock distribution delay. This guarantees that the input register hold time is always zero or negative, never a positive hold time. A positive hold time requirement could lead to unreliable, temperature- or processing-dependent operation.

The delay guarantees a zero hold time with respect to the clock provides by the bus clock buffer.

PIO Output Side

Output signals can be optionally inverted within the PIO, and can pass directly to the pad or be stored in an edge-triggered output flip-flop.

A logical Low on the Output-Enable signal forces the output into a high-impedance state. Consequently, a PIO functions as a three-state output or bi-directional I/O. Conversely, a logical High enables the output buffer. Under configuration control, the Output and Output-Enable signals can be inverted. The polarity of these signals is independently configured for each PIO. In addition, each can be tied High or Low independently.

The outputs on each PIO are full CMOS outputs. The switching threshold is a product of the I/O supply voltage (VIO).

Although the E5's output supply is 3.3 volts, each output is capable of driving a standard 5-volt TTL output levels. Most 5-volt CMOS devices recognize HCT (TTL level) inputs.

An output can be configured as open-drain (open-collector) by tying the output path to ground and driving the output-enabled signal, OE. However, the voltage applied to the pin should never exceed the values defined for VIO.

Selectable Output Drive Current

Each PIO has selectable output drive current, capable of sinking either 4 mA or 12 mA, as shown in Table 14. Reduced drive current results in lower power consumption, lower EMI emissions and lower ground bounce.

Table 14. Output Drive Current Options.

Mode	*Output Current*		Units
	I_{OL}	I_{OH}	
Low drive (default)	-4.0	+2.0	mA
High drive	-12.0	+6.0	mA

Other PIO Options

There are a number of other programmable options available for PIO blocks.

BusMinder™

The BusMinder feature allows each PIO to have an optional

- Pull-up resistor (pulls undriven inputs High)
- Pull-down resistor (pulls undriven inputs Low)
- Weak follower (forces undriven inputs to the last value that appeared on the bus)

Triscend E5 customizable microcontroller devices are fabricated on leading-edge CMOS processes. Like all CMOS devices, input pins should never be left floating. An unused input might float unless tied to High or Low. In addition, an input connected to a bi-directional bus might float if the bus is three-stated (high impedance).

The BusMinder's programmable pull-up/pull-down resistor and weak follower are useful for tying unused or floating pins High or Low to minimize power consumption and reduce noise sensitivity.

The configurable pull-up resistor is a p-channel transistor that pulls to VCC. The configurable pull-down resistor is an n-channel transistor that pulls to Ground. The value of these resistors is 50 kΩ to 100 kΩ. This high value makes them unsuitable as wired-AND pull-up resistors.

The pull-up resistors for PIOs are active during the initialization process. This pulls all as-yet-unprogrammed PIOs High, preventing them from floating. Devices connected to the PIO see a logical One. Other devices driving into the PIO can easily overdrive the weak pull-up resistor.

After initialization, voltage levels of unused pads, bonded or unbonded, must be valid logic levels, to reduce noise sensitivity and avoid excess current. Therefore, by default, unused pads are configured with the internal pull-up resistor active.

The weak follower is used on PIOs that connect to a bi-directional bus. Instead of pulling the floating input High or Low, the weak follower remembers the last value that appeared on the bus before the bus signal was three-stated.

Low-Power Mode

The Triscend E5 device can be placed into power-down mode by setting the PD bit (PCON.1) in the Power Control register. While in power-down mode, each PIO pin can optionally be configured for low-power operation. The PIO bit (PWDSEL.5) bit must be set before entering power-down mode to enable low-power mode.

Output Options

If enabled for low-power operation, a PIO output is disabled during power-down mode. A disabled PIO output is forced into a high-impedance state. The BusMinder function switches to weak follower mode, regardless if configured for a pull-up or pull-down resistor or left floating. The weak follower function keeps the PIO at the voltage level last applied to the pin. This helps to reduce overall power consumption.

Input Options

Likewise, a PIO input can be enabled for low-power operation. A PIO input is optionally forced Low during power-down mode.

JTAG Support

Embedded logic attached to the PIOs contains test structures compatible with IEEE Standard 1149.1 for boundary-scan testing, permitting easy chip and board-level testing.

Default, Unconfigured State

Before configuration, all PIO pins function via the JTAG interface but with the following default conditions:

- The input hysteresis is turned on
- The output drive is lowest output current mode
- The BusMinder is a pull-up resistor

Default, Configured State

All unused but configured PIO blocks are programmed as inputs with the soft BusMinder's pull-up resistor enabled. This prevents unused PIO pins from floating.

Electro-static Discharge (ESD) Protection

Each PIO has built-in ESD protection capable of protecting against a minimum discharge of 2,000 volts, using the human body model.

Memory Interface Unit

The Memory Interface Unit (MIU) provides a flexible, glueless interface between the E5 Customizable Microcontroller and external memory, as shown in Figure 29. The MIU primarily provides access to an external memory device—typically a 256Kx8 Flash PROM—which contains the E5's initialization data and the user's program code. Other types of memory, such as static RAM or EPROM, can substitute for the Flash PROM because the control signals are identical.

The MIU serves two purposes. First, it replaces the original 8051's multiplexed address and data ports that consumed the 8051's port 0 and port 2. Instead, the MIU provides a more flexible demultiplexed address and data bus. Second, it provides a convenient port for accessing external functions from other peripheral functions, including the Configurable System Logic (CSL) matrix.

As shown in Figure 29, the MIU provides a direct connection between the 8051 microcontroller and the external memory. By bypassing the internal CSI bus, the Triscend E5 maintains timing compatibility with the original 8051 implementation.

The external memory timing is variable allowing the E5 to optimize or slow down accesses to fit the particular AC characteristics of different memory devices. The MIU can also interface with an external serial sequential-access PROM memory, similar to those used to configure SRAM-based field-programmable gate arrays.

The address, data, and control pins of the MIU are also controllable from other peripheral functions, including the Configurable System Logic (CSL) matrix, allowing access to external peripherals or memory devices. The MIU requires minimal handshaking logic from within the CSL matrix.

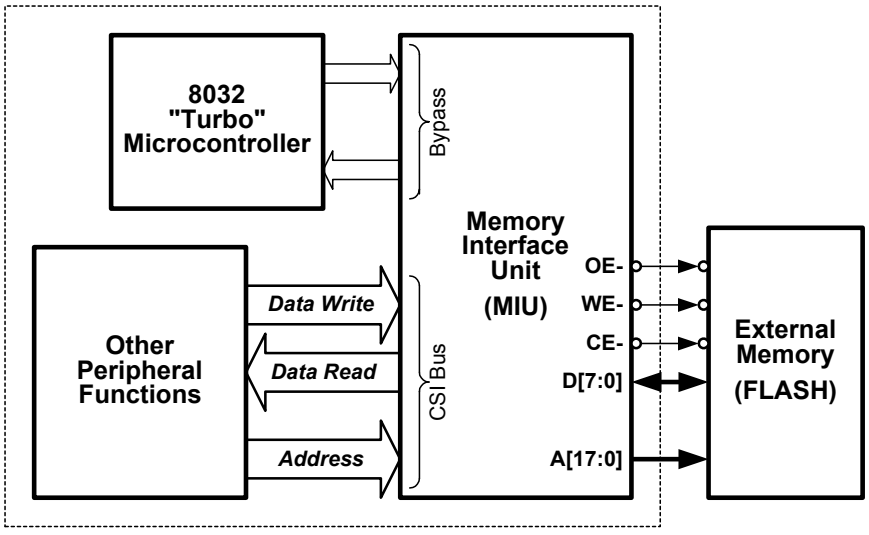


Figure 29. The Memory Interface Unit (MIU) provides a glueless interface between the microcontroller, its peripheral functions, and external memory.

The main features of the memory interface unit are:

- Support for a standard 256Kx8 external memory interface
- Support for expansion up to 4Gx8 of external memory through PIOs
- Programmable read and write timing control
- Bypass mode to support direct access by the 8051 core
- External serial memory support
- Multi-chip expansion capabilities

Functional Description

The MIU provides the interface between the internal CSI bus and external memory. The MIU monitors the internal bus to determine when external memory is accessed. During an external access, the MIU controls external memory and generates the appropriate read, write, and enable strobe signals.

The MIU interface timing is programmable. Three bits define the timing of each portion of external cycles. Read cycles are split into two sections, a setup and an active portion, as shown in Figure 30. In order to support one cycle reads, the minimum setup and width values are respectively 0 and 1. The leading edge of OE- is generated off the falling edge of the clock. The trailing edge of OE- is generated from the rising edge of the clock. In a one cycle read, depending on the clock duty cycle, the setup and width of OE- are roughly half a clock cycle each. Extra clock cycles can be added to either the setup or the strobe width. The minimum setup is ½ clocks and it can be extended to 7½ clocks. Similarly, the pulse width of OE- can take a minimum of ½ clocks up to 7½ clocks. To summarize, a read cycle is programmable to between 25 ns and 375 ns with a granularity of 25 ns, using a 40 MHz system clock.

Write cycles are partitioned into three sections: a setup, a write pulse and a hold portion, as shown in Figure 32. The hold portion is optional. The setup and width are similar to the read case described above. Extra hold cycles can be added after the trailing edge of the WE- signal to guarantee various hold conditions. The fastest write access is performed in one clock cycle, with ½ clock for setup and ½ clock for write-pulse width. Additional clock cycles can be added to any portion of the write cycle through the MIU configuration register. Setup and pulse width can vary from ½ clock to 7 ½ clocks. The hold portion varies from zero to 7 clocks. To summarize, a write cycle is programmable to between 25 ns and 550 ns with a granularity of 25 ns, using a 40MHz bus clock.

The following timing diagrams illustrate the options described earlier.

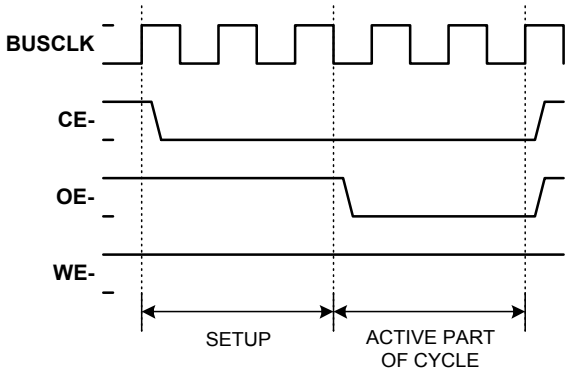


Figure 30. Generalized read cycle using MIU.

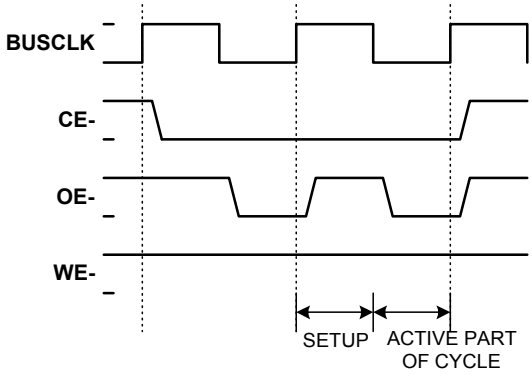


Figure 31. One-cycle read operation using MIU.

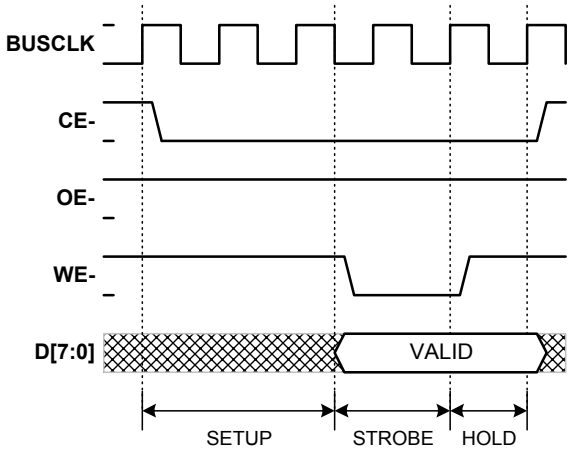


Figure 32. Generalized write cycle using MIU.

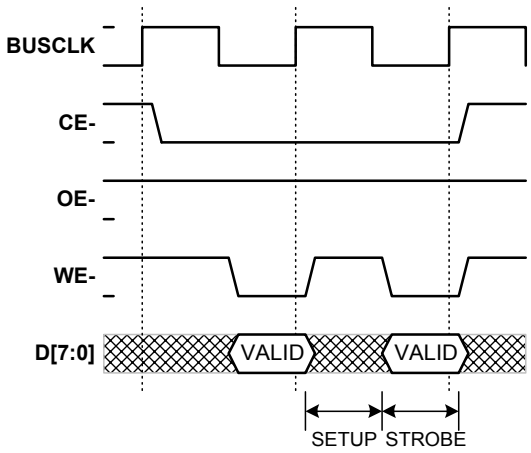


Figure 33. One-cycle write operation using MIU.

Wait-state Generation

Generally, a wait signal is generated for every external memory read access, except for a read from a fast external memory, with access time within one clock cycle.

Memory writes operate differently. Because write data and addresses are held in buffers, the MIU does not need to stall the CSI bus. A wait is generated only if another memory access is requested while the previous write is in progress.

Wait-states are also generated when the 8051 microcontroller directly accesses external memory, bypassing the internal bus. These wait-states occur because every bypass transaction is replicated on the internal bus for debugging purposes. The wait signal is synchronized correctly with the replicated transaction on the bus.

Requests Arbitration

Three potential sources use the MIU as an internal system slave.

1. Any master can access the MIU via the CSI bus.
2. The microcontroller can bypass the CSI bus and directly access the MIU.
3. The programmable address selectors can generate external memory requests.

External Memory Request Mode

"Soft" modules implemented in the Configurable System Logic (CSL) matrix can borrow MIU resources to access external devices. Functions in the CSL matrix can re-use the MIU's address and data signals plus some control via the OE- and WE- pins. The CSL matrix can request a memory cycle by using a special mode of a Selector to generate the chip enable (CE-) to the external device. The WE-, OE-, address and data signals are generated through the MIU. The OE- and WE- signals behave similar to the general case. Their leading edges are generated of the falling edge of the clock, and their trailing edges are generated of the rising edge of the clock. Therefore, one cycle reads or writes are possible. Using the WAITNEXT signal, the MIU knows if it has to extend the external command (OE- or WE-) or if it should end the current cycle and generate a new one.

For write cycles, the setup time is always ½ clock. However, to improve margin at high clock rate, the setup time can be extended by one full clock cycle. The hold time however is only guaranteed by the delay on the address and data lines, requiring external memory with 0ns hold time requirements. If different timing is required by the application, then the OE- and WE- signals can be generated from within the CSL matrix. These signals would then connect to external memory via a PIO pin.

Initialization of the MIU

After a power-on reset or other device-wide reset, the MIU begins operation as a master on the CSI bus. The MIU is then ready to receive commands via the external memory bus. This is useful for slave-mode initialization. If the SLAVE- pin is held High, then the MIU becomes a CSI bus slave during the initialization process.

During a power-on reset or other device-wide reset, the CE- signal is held High via an internal pull-up to prevent spurious writes.

MIU Register Description

The MIU and its different modes are programmed through 4 bytes of registers.

MIU Mode Register

The MIU configuration register defines the current operating mode of the MIU.

7	6	5	4	3	2	1	0
Reserved		DMA_WR	DMA_RD	SER_WR	SER_RD	MASTER	SLAVE

Mnemonic: MIUMODE Address: FE30h

SLAVE, when set, indicates that the MIU is a slave on the CSI bus. For most stand-alone applications, this is the mode used after configuration. Cleared by a Power-On or System Reset event.

MASTER, when set, indicates that the MIU is a master on the internal system bus. This is the initial state of the MIU upon power-up. Set by a Power-On or System Reset event.

SER_RD, when set, indicates that the MIU is configured to read configuration data from an external serial sequential-access PROM. Cleared by a power-on reset or other device-wide reset.

SER_WR is used to program a FLASH- or EEPROM-based external serial sequential-access PROM. Cleared by a power-on reset or other device-wide reset.

DMA_RD is a reserved MIU mode used during manufacturing. Leave this bit cleared. Cleared by a power-on reset or other device-wide reset.

DMA_WR is a reserved MIU mode used during manufacturing. Leave this bit cleared. Cleared by a power-on reset or other device-wide reset.

NOTE:	*The MIU control registers are part of the E5's "hidden" control register set. The BLOCKSIZE bit in the DMAP3_CTL register must be set in order to access the MIU control registers. The MIU registers are normally not modified by user application code.*

MIU Timing Control Registers

The MIU timing control register defines the timing of read and write control strobes issued by the MIU. The write hold time value WHT[2:0] field spans the two control bytes.

7	6	5	4	3	2	1	0
WHT1	WHT0	WPW2	WPW1	WPW0	WSU2	WSU1	WSU0

Mnemonic: MIUCTRL0 Address: FE31h

WSU[2:0] specifies the WE- setup time during a memory write cycle. It specifies the additional number of clock cycles added to the minimum setup time of ½ clock when generating a memory write cycle. The actual setup width is (WSU[2:0] + ½) * (system clock period). These three bits are set to all ones by a power-on reset or other device-wide reset, defaulting to the slowest setting.

Table 15. Write Setup Time.

WSU2	WSU1	WSU0	Bus Clock Cycles
0	0	0	0.5
0	0	1	1.5
0	1	0	2.5
0	1	1	3.5
1	0	0	4.5
1	0	1	5.5
1	1	0	6.5
1	1	1	7.5 (default)

WPW[2:0] specifies the pulse width of WE- during a memory write sequence. The actual pulse width is (WPW[2:0] + ½) * (system clock period). These three bits are set to all ones by a power-on reset or other device-wide reset, defaulting to the slowest setting.

Table 16. Write Pulse-Width Time.

WPW2	WPW1	WPW0	Bus Clock Cycles
0	0	0	0.5
0	0	1	1.5
0	1	0	2.5
0	1	1	3.5
1	0	0	4.5
1	0	1	5.5
1	1	0	6.5
1	1	1	7.5 (default)

WHT[2:0] field specifies the width of the hold portion of a write cycle. The actual hold width is WHT[2:0] * (system clock period). These three bits are set to all ones by a power-on reset or other device-wide reset, defaulting to the slowest setting. ***NOTE:*** The WHT2 bit is part of the MIUCTRL1 byte.

Table 17. Write Hold Time.

WHT2	WHT1	WHT0	Bus Clock Cycles
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7 (default)

7	6	5	4	3	2	1	0
XMSU	RPW2	RPW1	RPW0	RSU2	RSU1	RSU0	WHT2

Mnemonic: MIUCTRL1 Address: FE32h

RSU[2:0] specifies the additional number of clock cycles to add to the minimum setup time of a read cycle. The actual setup width is (RSU[2:0] + ½) * (bus clock period). These three bits are set to all ones by a power-on reset or other device-wide reset, defaulting to the slowest setting.

Table 18. Read Setup Time.

RSU2	RSU1	RSU0	Bus Clock Cycles
0	0	0	0.5
0	0	1	1.5
0	1	0	2.5
0	1	1	3.5
1	0	0	4.5
1	0	1	5.5
1	1	0	6.5
1	1	1	7.5 (default)

RPW[2:0] specifies the pulse width of OE- during a memory read sequence. The actual pulse width is (RPW[2:0] + ½) * (system clock period). These three bits are set to all ones by a power-on reset or other device-wide reset, defaulting to the slowest setting.

Table 19. Read Pulse-Width Time.

RPW2	RPW1	RPW0	Bus Clock Cycles
0	0	0	0.5
0	0	1	1.5
0	1	0	2.5
0	1	1	3.5
1	0	0	4.5
1	0	1	5.5
1	1	0	6.5
1	1	1	7.5 (default)

XMSU enables an additional clock cycle for setup in expansion cycles. This bit is cleared by a power-on reset or other device-wide reset.

MIU Serial Mode Control Register

This register provides control over serial-mode initialization and over device programming for an external, FLASH- or EEPROM-based, sequential-access serial PROM.

7	6	5	4	3	2	1	0
SDIN	CLKDIV	SDOUT_EN-	SDOUT	CLK	CE-	SER_EN-	RESET_OE-

Mnemonic: MIUSCTL Address: FE33 h

RESET_OE- drives the RESET/OE- enable pin of the external serial memory via the MIU's OE- pin. This bit is set by a power-on reset or other device-wide reset.

SER_EN- drives the write-enable input of an external FLASH- or EEPROM-based serial PROM for programming. This bit is set by a power-on reset or other device-wide reset.

CE- drives the chip-enable input of the external serial PROM. This bit is set by a power-on reset or other device-wide reset.

SDOUT is the external serial memory data. Only used when programming an in-system programmable, external, serial PROM. Unaffected by a reset.

SDOUT_EN- is the external serial memory write data output enable. In serial write mode, this bit provides direct control over the three-state enable line of the output data buffers. Cleared by a power-on reset or other device-wide reset.

CLKDIV controls the speed of the serial clock in serial read mode only. When cleared, SCLK = BUSCLK/4. When set, SCLK=BUSCLK/64. This bit is set by default by a power-on reset or other device-wide reset. The default is the slower mode.

SDIN is the serial memory read data. This bit is read-only.

Address Mappers

The address mappers translate the 8051 microcontroller's accesses to data, program or SFR address spaces into addresses for the Triscend CSI bus. Because of its architecture, an 8051 microcontroller supports multiple address spaces, as shown in Figure 34.

The 8051 microcontrollers has two separate, external 64K bytes address spaces: program and data. Program memory holds instructions and arguments used during code fetching while data memory stores variables and other data for applications.

The shaded blocks in Figure 34—external program memory, external data memory and user-selected SFR registers—can be directed onto the Triscend E5's CSI bus. The MCU's indirect and direct internal RAM spaces can not be mapped outside the 8051 microcontroller. The internally implemented SFRs—ACC, B register, power, timers, serial channel control, etc.—are not mapped outside the 8051 MCU to avoid software incompatibilities.

Address mappers translate the microcontroller's 8- or 16-bit logical address to a 32-bit physical address on the CSI bus. Each mapper monitors microcontroller accesses to a particular type of address space such as program, data or external SFR address space.

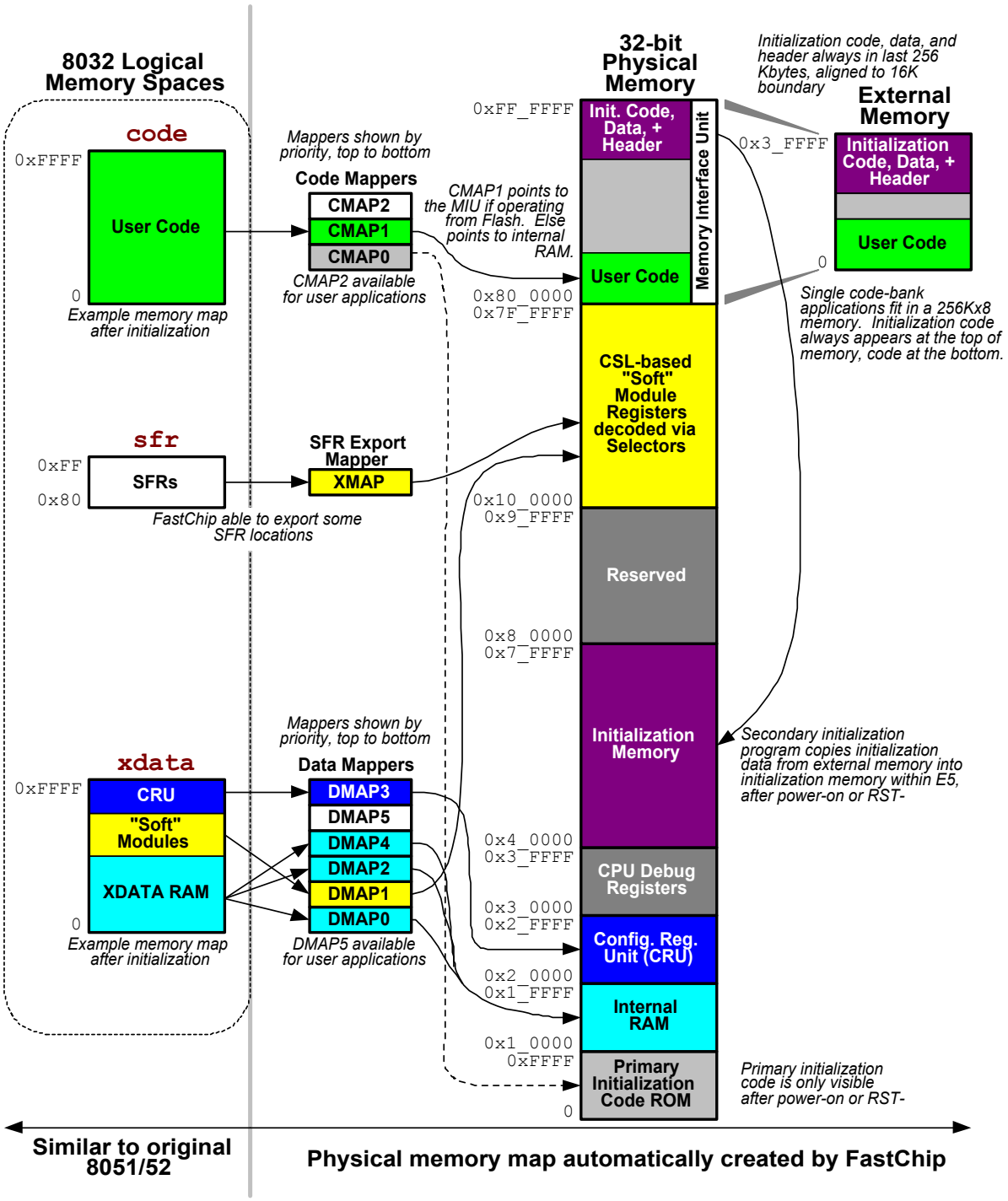


Figure 35. The Triscend FastChip development system uses the internal code and data mappers to automatically create a 16M-byte memory map.

Mappers respond to accesses within a selectable zone of logical addresses. The zones refer to a range of addresses. The zone size is a power of two with a minimum size of 256 bytes and a maximum of 64K bytes. A zone starts on any logical address boundary that is a multiple of its size. When an address match occurs between an 8051 address and an address mapper zone, the MCU's logical address is translated to a physical CSI bus address.

Table 20. Addresses Allocated to System Resources.

CSI physical address space	8051 Address zone (bytes)	Description	Notes
0 – 0xFFFF	1K	Internal ROM	Accessible following a system reset
0x1_0000 – 0x1_FFFF	Depends on device	Internal XDATA RAM	TE502=8K, TE505=16K, TE512=32K, TE520=40K
0x2_0000 – 0x02_FFFF	256 or 4K	CRU – system configuration registers	
0x3_0000 – 0x7_FFFF	N.A.	Reserved	Debugger access only
0x10_0000 – 0x7F_FFFF	User-defined (*)	CSL "soft" modules, external devices, external memories, and exported SFRs.	External devices or memories other than the one connected to CE- output pin should use this address space also.
0x80_0000 – 0xFF_FFFF	<= 64K	External Memory	Connected via the MIU port

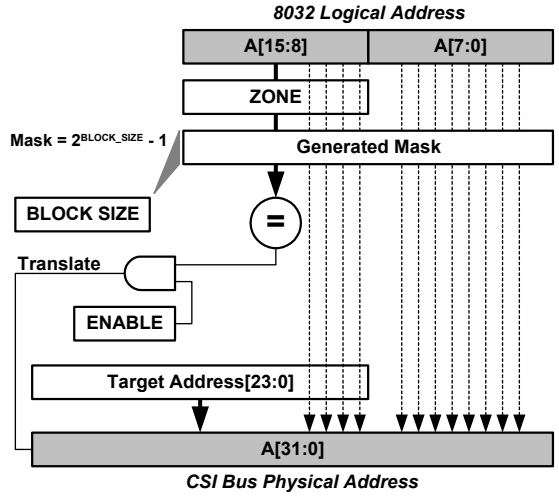


Figure 36. Address mapper conceptual block diagram.

Figure 36 shows a generalized block diagram of how an address mapper translates an 8051 logic address into a physical address displayed on the CSI bus.

When the MCU presents an address, the appropriate mapper tracks the type of operation. A code mapper only responds to code accesses, a data mapper only responds to data accesses, and so on. The block size defines how many lower address bits are ignored when comparing the MCU's logic address and the target address zone. Because all zones must be equal to 256 bytes or larger, the lower eight bits of the generated mask can be ignored during the comparison. The mapper then compares the upper 8 bits of the logical address with the 8 bits of the zone target address and masks off any locations indicated with the BLOCK_SIZE value. If the mapper is enabled and the two addresses compare, then the mapper presents the translated address on the CSI bus. The lower physical address bits consists of the masked bits from the logical address, A[15:0]. Because all zones are 256 bytes or larger, the lower-byte of the logic address, A[7:0], always maps directly to the translated address. The upper bits of the translated address are the unmasked values from the mapper's 24-bit target address value.

If the logical address does not match the mappers target zone, then the next-highest priority mapper examines the logical address. This process repeats among other mappers of the same type—*i.e.* code or data—until a match is found. The lowest-priority mapper is always guaranteed to match.

The dedicated resources on the CSI bus only decode 24 address lines, A[23:0]. The higher order address lines, A[31:24] can carry diagnostics information for debugging purposes.

The mapper values are defined in the configuration registers. In general, each mapper has

- 8 bits of zone address, the value to match against the expected logical value on A[15:8]
- 5 bits for block size, specified as the log base two of zone size in bytes
- 24 bits, which form the upper bits of the CSI physical address values. When a match occurs, this value is placed on the CSI physical address lines A[31:8]
- one enable bit.

The block size results in a mask used during address matching. The block size is specified as the log base two of the desired address zone size, in bytes. The mask is a result of the following calculation.

$$Mask = 2^{Block\_Size} - 1$$

The lower 8 bits of the logical address are not part of the comparison. A block size must be in the range between 256 and 64K. Consequently, the block size value must be between 8 and 16, inclusive.

Various system resources occupy non-contiguous address spaces in the 4G-byte address space available through the CSI bus. The Triscend FastChip development system automatically creates a 16M-byte memory map as shown in Table 20 and Figure 35. The memory map is 16M bytes instead of the full 4G bytes because the upper eight bits of the physical address are assigned to the values shown in Table 23 to aid in debugging.

Code mappers

Code mappers perform address translations on code fetches. There are three code mappers, including a dedicated code mapper for the initialization program stored in on-chip ROM plus two fully programmable code mappers. The code mappers are listed in Table 21.

Table 21. Code Mappers.

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

C0 – ROM code mapper

The C0 mapper has the lowest priority. The internal ROM is always accessible by the microcontroller following a System Reset event. The ROM size is fixed at 1K byte, implemented with 64 copies that are available in the program space. The copies are repeated at every 1K byte boundary in code space.

At the end of the initialization program, the other two code mappers, C1 and C2, can be programmed to override the ROM code mapper.

The C0 mapper has two programmable registers: CMAP0_TAR and CMAP0_ALT, both residing in the CRU. During a C0 address-matching event, the content of the CMAP0_TAR or CMAP0_ALT register is placed on the A[31:24] physical CSI bus address lines whenever an operand or an op-code fetch occurs, respectively. This feature is useful in debugging sessions to discriminate between op-code or operand fetches. C0 is always enabled.

C0 Operand Target Address (ROM mapper)

7	6	5	4	3	2	1	0
A31	A30	A29	A28	A27	A26	A25	A24

Mnemonic: CMAP0_TAR Address: FF00h

C0 Op-Code Target Address (ROM Mapper)

7	6	5	4	3	2	1	0
A31	A30	A29	A28	A27	A26	A25	A24

Mnemonic: CMAP0_ALT Address: FF01h

C1, C2 – Fully programmable code mappers

Code mapper C2 has the highest priority in case of an address overlap with other code mappers. C1 and C2 can be individually enabled. Following a system reset, mappers C1 and C2 are disabled. Mapper C0 is always enabled. The Triscend initialization program assigns application dependent values to these mappers. The user can modify these assignments by writing different values to the corresponding mapper registers.

C1 Target Address (Low Byte)

7	6	5	4	3	2	1	0
A15	A14	A13	A12	A11	A10	A9	A8

Mnemonic: CMAP1_TAR_0 Address: FF02h

C1 Target Address (Mid Byte)

7	6	5	4	3	2	1	0
A23	A22	A21	A20	A19	A18	A17	A16

Mnemonic: CMAP1_TAR_1 Address: FF03h

C1 Operand Target Address

7	6	5	4	3	2	1	0
A31	A30	A29	A28	A27	A26	A25	A24

Mnemonic: CMAP1_TAR_2 Address: FF04h

C1 Op-Code Target Address

7	6	5	4	3	2	1	0
A31	A30	A29	A28	A27	A26	A25	A24

Mnemonic: CMAP1_ALT Address: FF07h

C1 Zone Source Address

7	6	5	4	3	2	1	0
A15	A14	A13	A12	A11	A10	A9	A8

Mnemonic: CMAP1_SRC Address: FF05h

C1 Control

7	6	5	4	3	2	1	0
-	-	ENBL	SIZE4	SIZE3	SIZE2	SIZE1	SIZE0

Mnemonic: CMAP1_CTL Address: FF06h

C2 Target Address (Low Byte)

7	6	5	4	3	2	1	0
A15	A14	A13	A12	A11	A10	A9	A8

Mnemonic: CMAP2_TAR_0 Address: FF08h

C2 Target Address (Mid Byte)

7	6	5	4	3	2	1	0
A23	A22	A21	A20	A19	A18	A17	A16

Mnemonic: CMAP2_TAR_1 Address: FF09h

C2 Operand Target Address

7	6	5	4	3	2	1	0
A31	A30	A29	A28	A27	A26	A25	A24

Mnemonic: CMAP2_TAR_2 Address: FF0Ah

C2 Op-Code Target Address

7	6	5	4	3	2	1	0
A31	A30	A29	A28	A27	A26	A25	A24

Mnemonic: CMAP2_ALT Address: FF0Dh

C2 Zone Source Address

7	6	5	4	3	2	1	0
A15	A14	A13	A12	A11	A10	A9	A8

Mnemonic: CMAP2_SRC Address: FF0Bh

C2 Control

7	6	5	4	3	2	1	0
-	-	ENBL	SIZE4	SIZE3	SIZE2	SIZE1	SIZE0

Mnemonic: CMAP2_CTL Address: FF0Ch

Data mappers

Data mappers perform address translations on data access instructions. There are six data mappers as shown in Table 22. D0 maps accesses to the internal system SRAM. D1, D2, D4, and D5 are fully programmable and D3 maps the CRU configuration registers.

Table 22. 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

D0 – RAM data mapper

The D0 has the lowest data mapping priority. The internal system RAM is always accessible by the microcontroller in data space following a System Reset. The size and configuration of the internal system RAM is device dependent.

For example, the RAM in the TE520 has a fixed size of 40K bytes. The bottom 32K bytes reside in data locations 0000h – 7FFFh. Four copies of the top 8K bytes appear at data locations 8000h, A000h, C000h and E000h, unless another data mapper is enabled and re-uses these locations.

If enabled, data mappers D1 and D2 override RAM data space. The D3 mapper, used to access the configuration registers (CRU) always overrides the other mappers, as explained below.

The D0 mapper has one programmable CRU register, DMAP0_TAR. The content of DMAP0_TAR is placed on the A[31:24] physical CSI bus address lines when a D0 matching event occurs. D0 is always enabled.

D0 Target Address (RAM mapper)

7	6	5	4	3	2	1	0
A31	A30	A29	A28	A27	A26	A25	A24

Mnemonic: DMAP0_TAR Address: FF0Eh

D1, D2, D4, D5 – Fully programmable data mappers

Data mapper D5 has higher priority in case of an address overlap with D4, D2, D1 or D0. Mappers D5, D4, D2, and D1 can be individually enabled and are disabled after a System Reset event. The Triscend initialization program assigns application dependent values to these mappers. The user can modify these assignments by writing different values to the corresponding mapper registers.

D1 Target Address (Low Byte)

7	6	5	4	3	2	1	0
A15	A14	A13	A12	A11	A10	A9	A8

Mnemonic: DMAP1_TAR_0 Address: FF0Fh

D1 Target Address (Mid Byte)

7	6	5	4	3	2	1	0
A23	A22	A21	A20	A19	A18	A17	A16

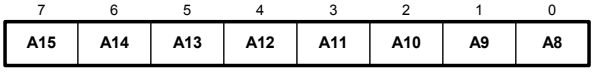
Mnemonic: DMAP1_TAR_1 Address: FF10h

D1 Target Address (High Byte)

7	6	5	4	3	2	1	0
A31	A30	A29	A28	A27	A26	A25	A24

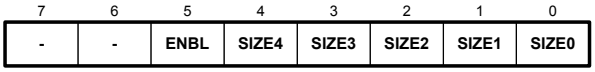
Mnemonic: DMAP1_TAR_2 Address: FF11h

D1 Zone Source Address



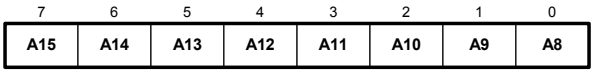
Mnemonic: DMAP1_SRC Address: FF12h

D1 Control



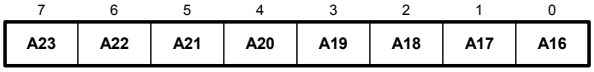
Mnemonic: DMAP1_CTL Address: FF13h

D2 Target Address (Low Byte)



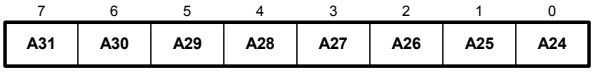
Mnemonic: DMAP2_TAR_0 Address: FF14h

D2 Target Address (Mid Byte)



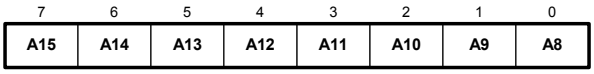
Mnemonic: DMAP2_TAR_1 Address: FF15h

D2 Target Address (High Byte)



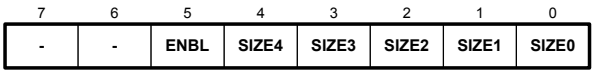
Mnemonic: DMAP2_TAR_2 Address: FF16h

D2 Zone Source Address



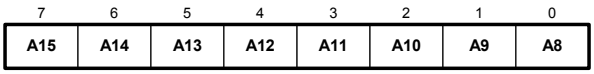
Mnemonic: DMAP2_SRC Address: FF17h

D2 Control



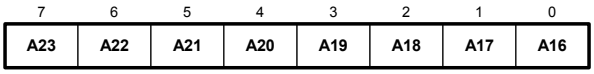
Mnemonic: DMAP2_CTL Address: FF18h

D4 Target Address (Low Byte)



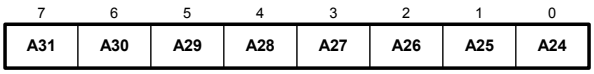
Mnemonic: DMAP4_TAR_0 Address: FF80h

D4 Target Address (Mid Byte)



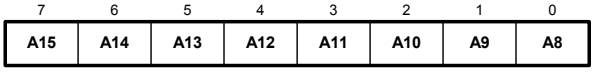
Mnemonic: DMAP4_TAR_1 Address: FF81h

D4 Target Address (High Byte)



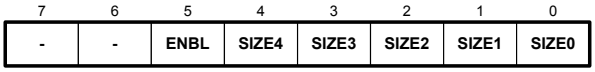
Mnemonic: DMAP4_TAR_2 Address: FF82h

D4 Zone Source Address



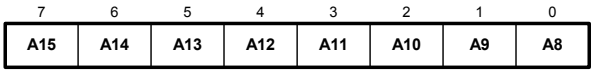
Mnemonic: DMAP4_SRC Address: FF83h

D4 Control



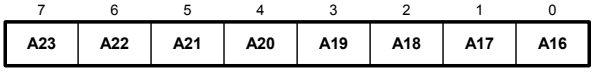
Mnemonic: DMAP4_CTL Address: FF84h

D5 Target Address (Low Byte)



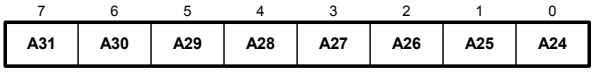
Mnemonic: DMAP5_TAR_0 Address: FF85h

D5 Target Address (Mid Byte)



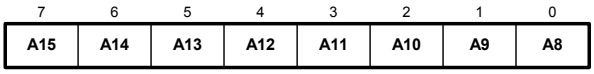
Mnemonic: DMAP5_TAR_1 Address: FF86h

D5 Target Address (High Byte)



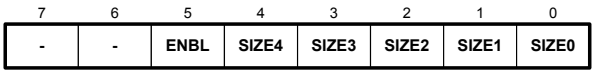
Mnemonic: DMAP5_TAR_2 Address: FF87h

D5 Zone Source Address



Mnemonic: DMAP5_SRC Address: FF88h

D5 Control



Mnemonic: DMAP5_CTL Address: FF89h

D3 – CRU data mapper

Data mapper D3 has the highest data mapping priority. The configuration registers (CRU) are always accessible by the microcontroller. D3 has two selectable sizes of 4K and 256 bytes. Following a system reset, the D3 block size is set to 4K bytes and occupies the upper 4K bytes of the logical data space, between addresses F000h and FFFFh, inclusive. At the end of the initialization program, the D3 data mapper zone is reduced to 256 bytes to conserve address space.

D3 start address must be aligned to 4K address boundaries. All mapper registers are accessible through D3. The D3 mapper is always enabled.

D3 Target Address

7	6	5	4	3	2	1	0
A31	A30	A29	A28	A27	A26	A25	A24

Mnemonic: DMAP3_TAR Address: FF19h

D3 Zone Source Address

7	6	5	4	3	2	1	0
-	-	-	-	A15	A14	A13	A12

Mnemonic: DMAP3_SRC Address: FF1Ah

D3 Control

7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	BLOCK SIZE

Mnemonic: DMAP3_CTL Address: FF1Bh

When set, BLOCKSIZE defines the D3 mapper block size to 4K bytes. The BLOCKSIZE bit is set automatically at power-up. When cleared, the block size is reduced to 256 bytes.

SFR export mapper

A Triscend Customizable Microcontroller user can create custom microcontroller "soft" modules using the on-chip Configurable System Logic (CSL) matrix. These "soft" modules can be mapped into the 8051's external data memory (XDATA) or into SFR space. If a CSL "soft" module is mapped into SFR space, then any control registers implemented in the CSL replace locations within the 8051's SFR memory. The fully programmable SFR export mapper supplies a 32-bit address on the CSI bus whenever an external SFR access is detected. Following a System Reset event, this mapper is disabled. The SFR export mapper has a special programmable register called XMAP_ALT. The content of XMAP_ALT or XMAP_TAR_2 is placed on the A31-A24 CSI bus address lines when a latch or a pin instruction occurs, respectively, during an external SFR access event. This feature can be useful in debugging sessions to discriminate between various SFR instructions.

Overall, there are 128 SFR-addressable locations in an 8051 microcontroller. Table 30 describes the SFRs within the MCU's scratchpad RAM. The Triscend FastChip development software prevents attempts to place CSL "soft" modules on internal SFR addresses already used by the MCU.

The translated address, when accessing an exported SFRs, consists of the XMAP_TAR_2 or XMAP_ALT register, the XMAP_TAR_1 register, and the XMAP_TAR_0 register appended to the A[7:0] address from the 8051 MCU.

"Soft" modules with exported SFR registers can also be accessed via 8051 external-memory references, using the values loaded into the data mappers. Placing "soft" module registers in the 8051's SFR address space allows faster and simpler access using direct reference instructions. This method also enables the application code to use bit-manipulating instructions on these registers.

Application code usually never modifies the SFR export mapper registers. The register values are set during initialization.

SFR Export Target Address (Low Byte)

7	6	5	4	3	2	1	0
A15	A14	A13	A12	A11	A10	A9	A8

Mnemonic: XMAP_TAR_0 Address: FE20h

SFR Export Target Address (Mid Byte)

7	6	5	4	3	2	1	0
A23	A22	A21	A20	A19	A18	A17	A16

Mnemonic: XMAP_TAR_1 Address: FE21h

SFR Export Target Address (Latch Instruction)

7	6	5	4	3	2	1	0
A31	A30	A29	A28	A27	A26	A25	A24

Mnemonic: XMAP_TAR_2 Address: FE22h

SFR Export Target Address (Pin Instruction)

7	6	5	4	3	2	1	0
A31	A30	A29	A28	A27	A26	A25	A24

Mnemonic: XMAP_ALT Address: FE24h

Default Values for Higher-Order Address Mappers

Table 23 shows the default assignment for higher address bits in each of the mappers. These values are loaded into the mapper registers during the initialization process.

The Triscend FastChip development system uses these default values when assigning the 32-bit physical addresses for the CSL address selectors. These values and the mapper registers themselves allow CSL address selectors to selectively respond to different instruction types.

For example, if a Flash memory device is connected to the MIU, some addresses can be accessed as code or data segments. During normal operation, the Flash segment is typically used to store code. However, during a programming session, the Flash memory is regarded as a data segment. This differentiation between code and data using high order address lines distinguishes between code and data accesses to the same physical address.

In another example, assume that an address selector is used to enable off-chip ROM for external

code storage. Code mapper C1 might point the ROM. The selector would need to respond to both operand and op-code fetches. Because CMAP1_TAR_2 = 00101000b and CMAP1_ALT = 000010000b, FastChip will program the upper byte of the selector's MASK register with 0010000, making the selector respond to both operands and op-code fetches.

Table 23. Default Values for Higher-Order Address Mapper Registers.

Transaction Type	Register Name	Default Value A[31:24]
C2 Operand Fetch	CMAP2_TAR_2	30h
C2 Op-Code Fetch	CMAP2_ALT	50h
C1 Operand Fetch	CMAP1_TAR_2	28h
C1 Op-Code Fetch	CMAP1_ALT	48h
C0 Operand Fetch (Init. ROM)	CMAP0_TAR_2	20h
C0 Op-Code Fetch (Init. ROM)	CMAP0_ALT	40h
D3 Data (CRU Registers)	DMAP3_TAR_2	98h
D5 Data	DMAP5_TAR_2	A8h
D4 Data	DMAP4_TAR_2	A0h
D2 Data	DMAP2_TAR_2	90h
D1 Data	DMAP1_TAR_2	88h
D0 Data	DMAP0_TAR_2	80h
SFR Latch	XMAP_ALT	E0h
SFR Pin	XMAP_TAR_2	C0h

The mapper values are also used during debugging allowing the hardware breakpoint unit to distinguish between operand and op-code fetches

The distinction between SFR pin and SFR latch transactions is required only if the designer selects 8051-compatible PIO ports from Triscend "soft" module library. The 8051 microcontroller has instructions that treat 8051 PIO ports differently than their respective SFR. In applications, there is no need for CSL address selectors to distinguish between SFR latch and SFR pin instructions.

System Debugger

On-Chip Debugging Support

A PC acting as an external tester or debugger can interface with the Triscend E5 Customizable Microcontroller via its JTAG port, using a cable connected between the PC parallel port and the target board, as shown in Figure 37.

The JTAG unit can access all addressable system resources by becoming the bus master on the internal CSI bus. Serving as a bus master, the JTAG unit converts serial bitstreams into parallel registers whose contents are placed on the address, data and command buses to emulate CSI bus transactions. A JTAG master also directly accesses internal MCU registers and CSL configuration data with visibility to functions not accessible from application programs.

Furthermore, the JTAG unit can hold the MCU in reset by asserting the J_RESET command or hold the entire system in reset using the FORCE_BRST and FORCE_NOBRST. The JTAG unit also interacts with the MCU using an interrupt handshake mechanism (J_INTR). For additional details, refer to the "***Reset Conditions***" section.

The JTAG unit serves as a bus slave when interacting with the DMA controller. It first becomes master on the CSI bus, programs the DMA unit to transfer data to or from the JTAG unit, and then relinquishes the CSI bus to the DMA controller. After configuring the DMA, the JTAG unit interfaces with the DMA controller as a slave device using the DMA's request and acknowledge signals. There is considerably less overhead in this type of data transactions since the JTAG unit does not need to provide address or control information during DMA operations.

Serving as a bus master, the JTAG unit is able to set breakpoint events within the breakpoint unit, which is also a slave connected on the CSI bus. The breakpoint unit supports two independent breakpoint conditions. The breakpoint unit monitors user-specified combinations of address, data, control or DMA signals as breakpoint events. The user can also count logic conditions occurring inside the CSL as breakpoint events. A breakpoint specification can count up to 64K-1 incidents as the matching condition for a break event.

Once a breakpoint condition occurs, then depending on its current configuration, the MCU freezes at the end of the current instruction or receives a breakpoint interrupt and branches to execute debugger interrupt routines. Following a breakpoint freeze, CSL clocks or global signals can be blocked to aid system debugging. The Triscend FastChip software enables the user to specify which CSL clocks or global signals are affected following a breakpoint freeze.

During a CPU freeze period, the JTAG unit can poll any addressable location residing inside or outside the Customizable Microcontroller and send

all requested information to the host PC for further display and analysis.

At the end of the debugging session, JTAG clears the breakpoint “freeze MCU” condition and the Accelerated 8051 microcontroller resumes code execution from where it left off. Alternatively, the JTAG unit can restart code execution from 0000h by issuing a J_RESET command to the MCU.

At any point in time, the JTAG unit can instruct the MCU to enter a single-step operation and send pertinent display information to the host PC.

The JTAG port can also be used to initialize the Customizable Microcontroller or to update external memory devices connected to the MIU port. Using external Flash memory, the JTAG unit downloads a Flash-programming algorithm to the Customizable Microcontroller’s internal RAM. It then interacts with the internal MCU, allowing the processor to control the actual program / erase / verify algorithms while the JTAG port supplies new data for programming or new series of commands required by the MCU.

Debugging Support System Requirements

Unlike debuggers for traditional 8051 designs, the E5 debugging environment does not require any additional system resources. The E5 environment does not use any additional code space nor does it consume a serial port. All debugging interaction is through the dedicated four-pin JTAG connection.

The recommended debugging header, shown in Figure 38, connects to the Triscend JTAG Download/Debug cable. The header uses through-hole stake-pins on 0.1 inch centers.

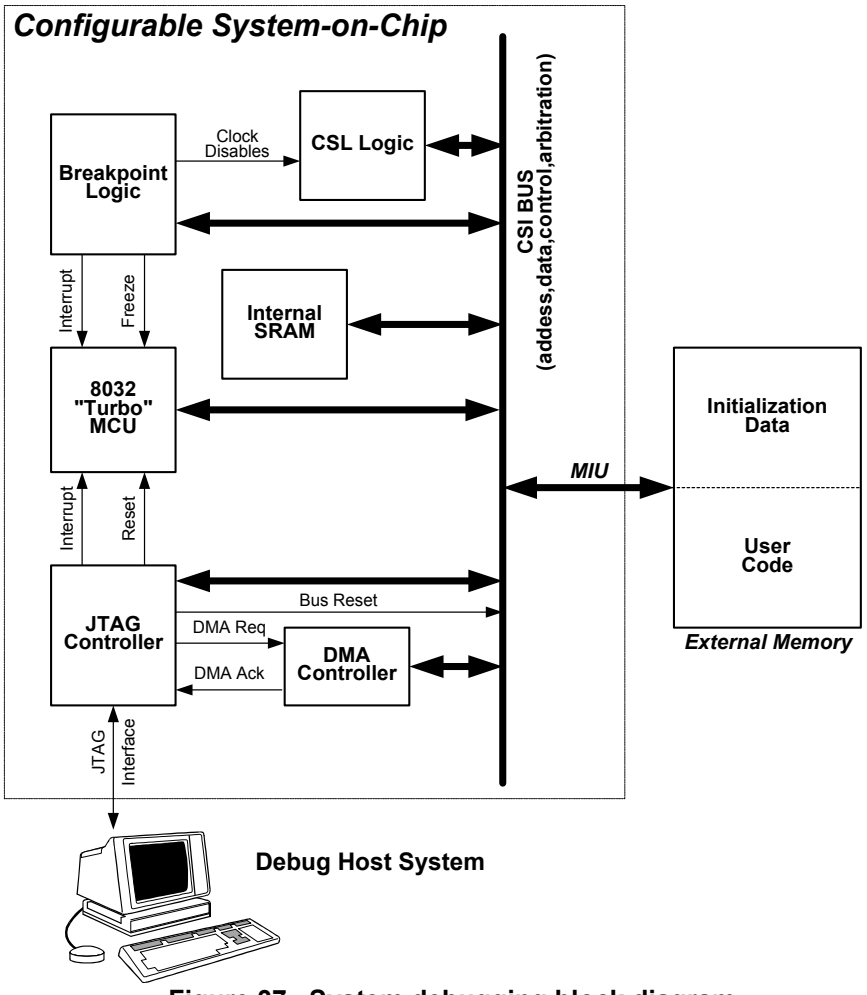


Figure 37. System debugging block diagram.

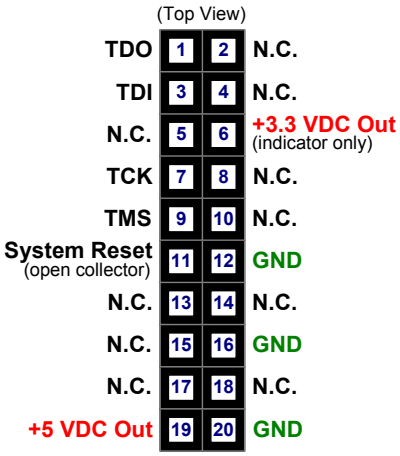


Figure 38. Triscend E5 Download/Debug Cable Header.

Likewise, no special in-circuit emulator (ICE) is required. The E5's on-chip breakpoint unit provides the capabilities available in most ICE systems.

Configuration Register Unit (CRU)

The Configuration Register Unit (CRU) contains the control registers for functions within the Triscend E5 Customizable Microcontroller. For most applications, the CRU registers occupy the upper 256 bytes of external data space.

Table 24 shows the functions controlled by the CRU registers. Unused locations have no function and are not RAM locations.

System Initialization

Like most processors, the Triscend E5 requires initialization data to configure programmable options available within the device before it becomes functional. The Triscend E5 offers four different system initialization options to best fit various target applications, as shown in Table 25. Both active and passive initialization methods are available.

Table 24. Configuration Register Unit (CRU).

FF88h	DMAP5_SRC	DMAP5_CTL						
FF80h	DMAP4_TAR_0 [7:0]	DMAP4_TAR_1 [15:8]	DMAP4_TAR_2 [23:16]	DMAP4_SRC	DMAP4_CTL	DMAP5_TAR_0 [7:0]	DMAP5_TAR_1 [15:8]	DMAP5_TAR_2 [23:16]
FF78h								
FF70h								
FF68h								
FF60h	PROTECT	SECURITY	PWDSEL	PORCTRL				
FF58h								
FF50h								
FF48h	DMACRC_0 [7:0]	DMACRC_1 [15:8]						
FF40h	DMACADR1_1 [15:8]	DMACADR1_2 [23:16]	DMACADR1_3 [31:24]	DMACCNT1_0 [7:0]	DMACCNT1_1 [15:8]	DMACCNT1_2 [23:16]	DMAPREQ1_0 [7:0]	DMAPREQ1_1 [15:8]
FF38h	DMASCNT1_0 [7:0]	DMASCNT1_1 [15:8]	DMASCNT1_2 [23:16]	DMACTRL1_0 [7:0]	DMACTRL1_1 [15:8]	DMAEINT1	DMAINT1	DMACADR1_0 [7:0]
FF30h	DMACCNT0_1 [15:8]	DMACCNT0_2 [23:16]	DMAPREQ0_0 [7:0]	DMAPREQ0_1 [15:8]	DMASADR1_0 [7:0]	DMASADR1_1 [15:8]	DMASADR1_2 [23:16]	DMASADR1_3 [31:24]
FF28h	DMACTRL0_1 [15:8]	DMAEINT0	DMAINT0	DMACADR0_0 [7:0]	DMACADR0_1 [15:8]	DMACADR0_2 [23:16]	DMACADR0_3 [31:24]	DMACCNT0_0 [7:0]
FF20h	DMASADR0_0 [7:0]	DMASADR0_1 [15:8]	DMASADR0_2 [23:16]	DMASADR0_3 [31:24]	DMASCNT0_0 [7:0]	DMASCNT0_1 [15:8]	DMASCNT0_2 [23:16]	DMACTRL0_0 [7:0]
FF18h	DMAP2_CTL	DMAP3_TAR	DMAP3_SRC	DMAP3_CTL				
FF10h	DMAP1_TAR_1 [15:8]	DMAP1_TAR_2 [23:16]	DMAP1_SRC	DMAP1_CTL	DMAP2_TAR_0 [7:0]	DMAP2_TAR_1 [15:8]	DMAP2_TAR_2 [23:16]	DMAP2_SRC
FF08h	CMAP2_TAR_0 [7:0]	CMAP2_TAR_1 [15:8]	CMAP2_TAR_2 [23:16]	CMAP2_SRC	CMAP2_CTL	CMAP2_ALT	DMAP0_TAR	DMAP1_TAR_0 [7:0]
FF00h	CMAP0_TAR	CMAP0_ALT	CMAP1_TAR_0 [7:0]	CMAP1_TAR_1 [15:8]	CMAP1_TAR_2 [23:16]	CMAP1_SRC	CMAP1_CTL	CMAP1_ALT

Unused location, not RAM.

In active initialization modes, the Triscend E5 provides the control signals, directing data transfers and controlling external devices.

In passive modes, an external controller directs data transfers.

Table 25. Initialization Modes.

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

Parallel Mode

Parallel mode initialization, as shown in Figure 39, enables the user to store and execute application programs from a standard, byte-wide external memory. System configuration is also stored in the external memory. System initialization time is much faster relative to serial configuration because user application programs are typically not downloaded to internal system RAM.

The MIU signals are:

D[7:0] – the eight-bit data bus from the external memory.

A[17:0] – the lower 18 bits of the address bus. If additional address bits are required, the upper address lines—A18 and beyond—can be enabled in the user's design.

WE- - write enable signal provided to writeable memories such as Flash, EEPROM, and SRAM

OE- - enables the data output from the external memory during a read operation.

CE- - chip enable to the external memory.

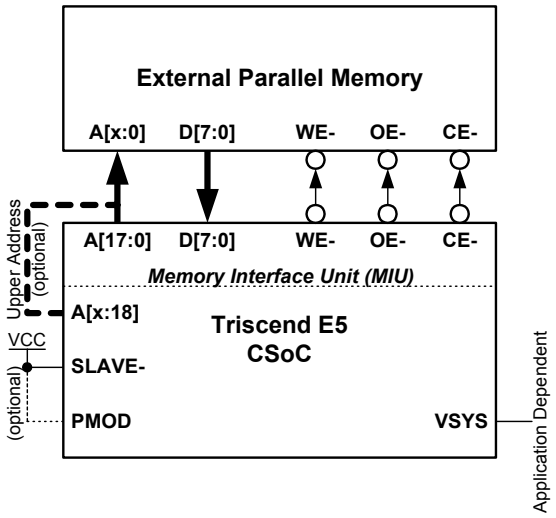


Figure 39. Parallel-mode initialization.

PMOD is optionally pulled up to VCC. If left unconnected, PMOD powers on with a weak pull-up resistor, pulling the pin High during the initialization phase.

Address lines A[17:0] sufficiently address up to 256K bytes of external memory. If a larger memory device is used, additional optional address lines serve as high-order address lines. Even though the 8051 can only directly access 64K bytes of code space, the Triscend E5 provides more code space by using code mappers that dynamically change the 8051's base address. The external parallel memory can also contain data arrays such as look-up tables. Using data mappers, the external memory can also be mapped to the external data memory of the processor.

Once the system initialization is complete, the 8051 microcontroller starts executing application code from location 0000h, which is mapped to the external memory. Parallel mode requires that at least some application code reside in external memory. At the beginning of application program execution, code mapper C1 maps locations 0000H – 7FFFH of program space into the external memory. The user application code must initialize the data mappers and code mappers.

During the parallel initialization phase, all PIO and MIU pins that do not participate in the parallel memory interface, are pulled High by weak pull-up resistors. These pins are properly initialized to their user-defined configurations at the end of initialization.

Serial Initialization

In serial initialization mode, the initialization data and user code are stored in a sequential-access serial PROM, similar to the devices used to program FPGA devices. Sequential-access serial

PROMs are distinct from I^2C- or SPI-format serial PROMs, as they are not addressable and are typically much higher density.

Serial initialization mode frees most of the memory interface unit (MIU) pins so that they can be used as PIO pins in the user's design. Figure 40 shows the connections between the Triscend E5 Customizable Microcontroller and the serial memory.

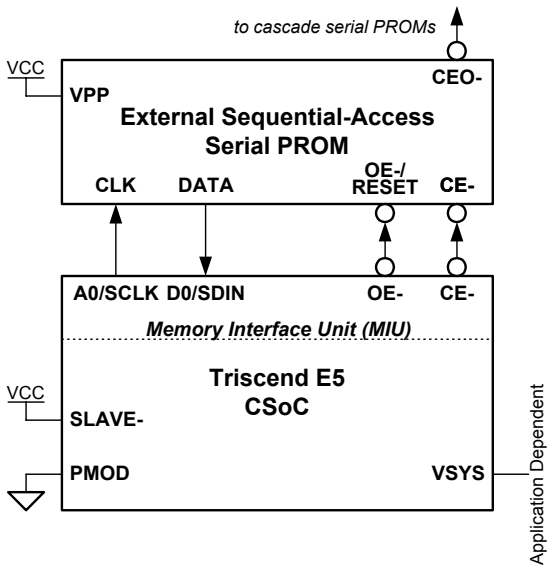


Figure 40. Serial-mode initialization.

The serial memory interface requires only four interface signals.

D0/SDIN - serial data bit from the serial PROM.

A0/SCLK - serial-clock supplied to the serial PROM.

OE-/SRST - resets the serial memory to its starting location when High and enables the serial data output when Low.

CE- - serial chip enable.

The PMOD input pin must tied Low, indicating Serial Mode initialization. Tying the SLAVE- pin High ensures that the system starts up in a single chip mode, configuring itself rather than relying on another intelligent controller to provide initialization data.

Following a power-on reset or a system reset, the Customizable Microcontroller detects Serial Mode and starts downloading initialization bitstream into the CSL. After the system initialization phase, the application program is serially loaded into the internal RAM.

Serial mode requires that application code reside in the internal RAM. Once the user program is resident in the system RAM, the Accelerated 8051 microcontroller starts executing user code from location 0000h, which is mapped to the internal system RAM. At the beginning of user program execution, code mapper C1 maps the entire 64K bytes of program space into the internal system RAM. However, the data mapper values must be handled by application code.

During the serial download phase, all PIO and MIU pins not active during the serial initialization process are pulled High by weak pull-up resistors. After initialization, these pins perform as described in the logic design.

JTAG Initialization

The JTAG mode initializes the Triscend E5 using an external tester, personal computer (PC), or other intelligent host. The JTAG port is also useful for debugging application code or to program external memory devices via the MIU port. When a PC acts as a host, the JTAG unit is controlled by a series of commands entered from the Triscend FastChip development system or third-party hardware debugger.

External memories are not required in JTAG configuration mode. The JTAG link can directly initialize the CSL matrix, download the necessary application code into the internal RAM, assign values to the data and code mappers and direct the 8051 microcontroller to execute code from the internal system RAM.

Since the JTAG unit can serve as a system debugger in this operating mode, any type of external memory device is allowed.

Figure 41 depicts a typical JTAG interface to a Triscend E5. In the figure, the Customizable Microcontroller is already connected to an external parallel memory. The JTAG port is compliant with IEEE standard 1149.1. The four JTAG pins are dedicated only to the JTAG function.

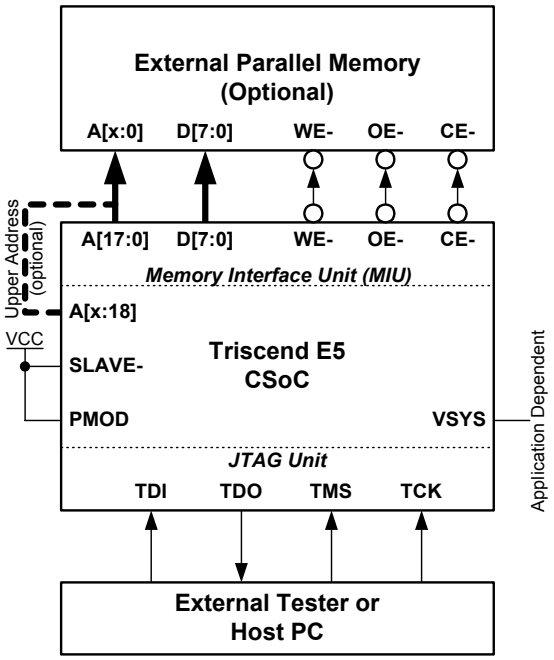


Figure 41. A JTAG interface to the Triscend E5.

TCK – Test clock input. If unused should be tied high.

TMS – Test mode select input. If unused should be tied high.

TDI – Test data input. If unused should be tied high.

TDO – Test data output.

The JTAG unit also serves as a master on the CSI bus and can read and write every addressable entity in the system.

The JTAG port optionally controls the Accelerated 8051 microcontroller by setting breakpoint events that freeze the MCU. Once the processor is frozen, the JTAG unit can single-step the processor, examine the internal registers of the processor, and then resume the processor operation.

With the help of the microcontroller, the JTAG port can also program an external Flash memory. In the Flash programming mode, the JTAG unit downloads program / erase / verify algorithms into the internal RAM. The new Flash programming data can also be stored in the internal RAM. After the code and data mappers are properly defined via the JTAG port, the 8051 microcontroller can execute the program / erase / verify algorithms required to write new data into an external Flash memory device. The processor interacts with the JTAG unit through flags and shared variables or via interrupts.

Slave Mode

In slave mode, the initialization data is downloaded into the Customizable Microcontroller via a byte-wide, memory-mapped interface, as shown in Figure 42. An external controller, microprocessor, or a master Triscend E5 device selects the slave Customizable Microcontroller by asserting the CE- signal. Up to 18 address inputs are required to initialize a single device. The address bus is expandable up to 32 bits if required by the application. The external controller presents address and data, then strobes the WE- signal.

The master controller and the slave Customizable Microcontroller should operate from the same external clock signal.

Once initialized, the master controller removes the slave's internal reset by setting the SRST_DIS bit in the slave's MISC configuration register. The 8051 microcontroller within the slave then begins operation. However, a slave microcontroller is not allowed to perform external code fetches. However, it can execute instructions downloaded by the master controller into the slave's internal system RAM.

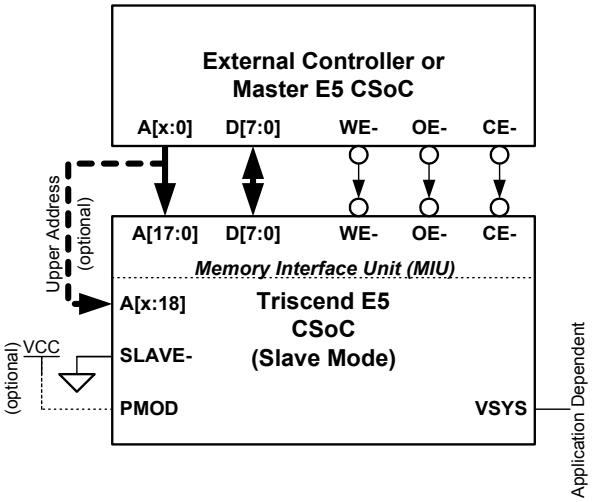


Figure 42. A slave mode interface to a Triscend E5 Customizable Microcontroller.

'Stealth' Mode

The Customizable Microcontroller has a security register allowing users to block external accesses through the MIU port or to prevent JTAG from executing CSI transactions. This mode is most useful when the Customizable Microcontroller's internal contents are battery-backed when power is removed. In this mode, the initialization data is first downloaded into the Customizable Microcontroller and the application code stored and executed from internal system RAM. Then the security bit is set, making the Customizable Microcontroller's operation invisible to external probes.

When power is re-applied after the Customizable Microcontroller was in secure mode and its con-

tents battery backed, then the Customizable Microcontroller begins executing from internal SRAM.

Blocking MIU accesses forces the Accelerated 8051 microcontroller to execute code only from the internal RAM. The executed program is prevented from fetching external code or data through the MIU port. Hackers cannot track the program flow since the MIU port is disabled.

Blocking JTAG transactions on the CSI bus is essential to prevent an external tester from reading configuration and code stored inside the Customizable Microcontroller. When JTAG security is set, boundary scan operations are still functional but bus transactions through the JTAG unit are not functional. However, the security status bit can be read through the JTAG port.

System Reset events have no effect on the Customizable Microcontroller once it has been secured. However, the application reset sideband signal, RSTC, remains fully functional in a secured Customizable Microcontroller. Following a RSTC pulse, the CPU restarts its operation from program location 0000H regardless of the security mode. All MCU Reset events remain functional.

Once set, these security bits can only be cleared by turning off the power. If the MIU security bit is set, the VSYS system voltage monitor level is ignored and the Customizable Microcontroller will not access an external memory via the MIU port.

Secure Mode Timed Access

Mnemonic: PROTECT Address: FF60h

The Secure Mode Timed Access (PROTECT) register controls the access to the E5's secure mode bits. A separate timed access register (TA) controls the protected bits within the Accelerated 8051 microcontroller. To access protected bits, the user must first write AAh to the PROTECT register. This must be immediately followed by a write of 55h to PROTECT. This opens a window for three machine cycles, during which time software can write to these protected bits.

The PROTECT returns FFh when read, but this register is rarely read.

There is unrestricted write access to this register, a read is not required.

Secure Mode Control

7	6	5	4	3	2	1	0
-	-	-	-	-	-	MIU	JTAG

Mnemonic: SECURITY Address: FF61h

SECURITY.7-2 are reserved bits and return a 0 when read.

MIU, when set, prevents any access through the MIU port, including any microcontroller code fetch to external memory. Once set, this bit can only be cleared by turning off the power.

JTAG, when set, prevents any access through the JTAG port. Once set, this bit can only be cleared by turning off the power.

A write cycle to the security register must be preceded by two data-specific, back-to-back write operations to the Protect register. A write operation to the Protect register with a data pattern of AAh must be followed by a write operation to the Protect register with a data value of 55h. Any write cycle directed at a different CSI location or containing a different data pattern would reset the write protect mechanism. After both write cycles are detected in the right order and match the above-specified data patterns, a write to the Security register is enabled for a single write cycle.

The JTAG secure and MIU secure bits are active high. Once set they can only be cleared by turning off the power. The 8051 microcontroller can always read the security register to determine or verify that the part is secured.

Size of Initialization Data

The size of the initialization data file depends on which E5 family device is used in the application and the initialization method used.

Initialization Data Size

Each device has a predefined number of rows and columns of initialization information as shown in Table 26. The total data size is the product of the number of rows and columns.

Table 26. Initialization Data Size.

Device	Columns	Rows	Total Bytes
TE502	102	120	12,240
TE505	102	208	21,216
TE512	144	296	42,624
TE520	186	384	71,424

Parallel Mode Secondary Initialization Program

The secondary initialization program loads an E5 device with the initialization data. The initialization program consists of 8051 instructions. The total size of the secondary initialization data image for parallel mode operation consists of

- 967 bytes of 8051 instruction code to load the initialization data into the E5 device
- 19 bytes of initialization data header and footer information
- the initialization data file from Table 26

- 94 bytes of 8051 instruction code to set up the data and code mappers after loading the initialization data.

Because the secondary initialization program always resides within the top 256 Kbytes of external memory, aligned to a 16K boundary, the absolute size of the data is larger than actual data file. Table 27 shows the total size of the secondary initialization data, including the initialization program, and the memory requirements once the initialization data is aligned to a 16 Kbyte boundary.

Table 27. Parallel Mode Memory Requirements.

Device	Secondary Init. Data Size	Total Memory Requirements
TE502	13,320	16K
TE505	22,296	32K
TE512	43,704	48K
TE520	72,504	80K

Time to Initialize an E5 Customizable Microcontroller Device

Parallel Initialization

The time to initialize an E5 device depends on the device used and the frequency of the bus clock. After power-on or after RST- is asserted, the E5 device begins initialization clocked from the internal ring oscillator. The frequency of the internal ring oscillator is minimum of 5 MHz and a maximum of 20 MHz. During the initialization process, the bus clock source can be switched over to an external clock source or to the crystal oscillator amplifier, both of which operate up to 25 or 40 MHz, depending on the speed grade.

Table 28 shows the initialization time using various frequency bus clock inputs. If bus clock is clocked from the crystal oscillator amplifier, allow an additional 5 ms for crystal settling time. If using the special 32 kHz mode for the crystal oscillator amplifier, allow 200 ms settling time.

Table 28. Estimated Parallel Initialization Times at Various Bus Clock Frequencies.

Device	*Bus Clock Frequency*			
	1 MHz	5 MHz	25 MHz	40 MHz
TE502	94 ms	19 ms	4.5 ms	3.1 ms
TE505	112 ms	23 ms	5.2 ms	3.6 ms
TE512	181 ms	37 ms	8.0 ms	5.3 ms
TE520	265 ms	54 ms	11.4 ms	7.4 ms

VSYS Control

A power-on reset or other device-wide reset causes the Triscend E5 to start the initialization process as shown in Table 39. The VSYS input pin directs the system initialization state machine how to behave should the initialization process fail to find valid external initialization data.

If the VSYS pin is held High through the initialization phase and no valid configuration pattern is found, then the Customizable Microcontroller automatically powers down to conserve power. Another Power-On Reset or System Reset event is required to restart the initialization process.

However, if VSYS is sampled Low during the initialization process, the Customizable Microcontroller device attempts to re-start the initialization process following a failed initialization attempt. The initialization mode, whether serial or parallel, is determined from the PMOD pin. If VSYS is tied to GND, the Customizable Microcontroller will attempt reconfiguring itself indefinitely until it finds valid initialization data. This option might be useful when the system operates in noisy environments.

NOTE:

VSYS must connect to a valid logic level. Do not leave VSYS unconnected.

In applications using a system supervisory chip, a "VCC good" output from the supervisory device can connect to VSYS, ensuring that initialization only takes place when VCC is within the proper operating voltage range.

If the MIU security bit is set, the VSYS level is ignored and the Customizable Microcontroller will not access an external memory via the MIU port.

Clocking and Global Signal Distribution

The Triscend E5 Customizable Microcontroller-provides powerful, flexible clocking resources. A system-wide bus clock (BCLK) provides clocking to most of the E5 device. In addition, six global buffers supply additional clocks or high-fanout signals for the CSL matrix.

System clock select (BCLK)

The system clock, called BCLK, supplies the clock to the Accelerated 8051 microcontroller, its dedicated resources, and to the CSI bus. Additionally, BCLK is distributed globally to the CSL matrix and to the PIO pins.

There are three potential user-defined sources for BCLK.

1. An internal ring oscillator operates at frequencies ranging between 5 MHz to 20 MHz, as shown in Figure 43. The ring oscillator frequency is temperature, voltage, and process dependent. The internal ring oscillator can be

shut off during power-down mode by setting the OSC bit (PWDSEL.3).

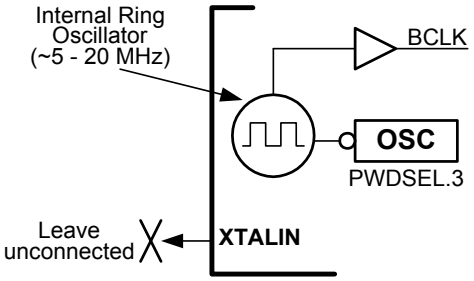


Figure 43. Internal ring oscillator.

2. A crystal oscillator amplifier—connected internally between the BCLK/XTAL and XTALOUT pins—supports 32 kHz operation and frequencies from 2 MHz up to 40 MHz, as shown in Figure 44. A crystal and the external circuitry listed in Table 29 generate the desired frequency. The crystal oscillator amplifier also supports ceramic resonators. The crystal oscillator output can be turned off during power-down mode by setting the XTAL bit (PWDSEL.4). Frequencies above 24 MHz require the crystal to operate in the third overtone. The crystal operates in its fundamental mode at lower frequencies. The internal feedback resistor should also be enabled using the FastChip development system.

Table 29. Crystal Oscillator External Component Values.

Crystal Frequency Range	Suggested External Resistor (R1)	External Capacitor (C1 and C2)
32 kHz	10 MΩ	10 pF
2 MHz—24 MHz (fundamental)	1 MΩ	10 pF
24 MHz—40 MHz (third overtone)	4.7 kΩ	10 pF

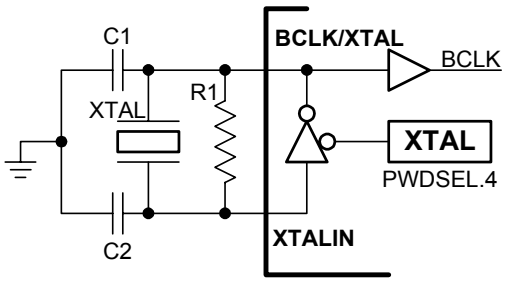


Figure 44. Crystal oscillator input.

3. A user-provided logic signal supplies the operating clock frequency when applied to the XTAL input pin. The XTALOUT pin should be left unconnected, as shown in Figure 45.

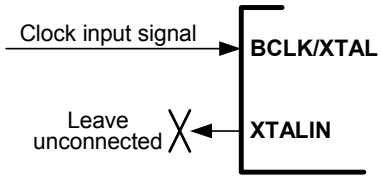


Figure 45. Clock input from external system.

The device always powers up using the internal oscillator, which serves as the system clock during the system initialization phase. The system switches over to the user selected clock source sometime before the MCU starts executing the application program. In most applications, the user-defined clock source is the crystal-oscillator clock (XTAL).

Two copies of the selected clock source propagate in the Customizable Microcontroller. The BCLK signal is shared by all dedicated resources connected to the CSI bus, including the MCU, DMA, mappers, programmable system decoders, etc. The CSL_BCLK propagates into the CSL matrix and optionally connects to programmable CSL "soft" modules and PIO pins.

Usually, the system clock source is selected at design time and unchanged by software. However, application code may need to modify the selection if the crystal oscillator is turned off during power-down mode. See "***Power-On Reset Control***" for more information.

System Clock Select Register

7	6	5	4	3	2	1	0
-	-	-	-	-	-	SCPU_ ENA	BCLK SEL

Mnemonic: MISC Address: FE81h

MISC.7-2 are reserved locations that return 0 when read.

SCPU_ENA, when set, releases the CPU reset caused when the SLAVE- input is Low. Used only in Slave mode applications.

BCLKSEL, when set, selects the crystal oscillator or an incoming, external clock source on the XTAL input pin. When cleared, then the internal ring oscillator provides the system clock. When the SLAVE- pin is Low, the BCLKSEL bit is forced to 1.

Six Global Buffers

In addition to system clock, there are six global signals available from within the CSL matrix. The six signals, GBUF[5:0], are buffered and provide a high-speed, low skew distribution path for additional clocks or high-fanout signals.

Clock and Global Signal Stopping

The system clock and the outputs from the six global buffers can be optionally stopped during power-down and debugging sessions.

Bus Clock Stopping and Single-Stepping

When stopped by a breakpoint event, the MCU completes the current instructions before stopping the bus clock, BCLK.

Global Buffer Stop Value

The six global buffers have a selectable value when stopped. During design, each global buffer is optionally configured to one of the following functions.

1. Actively drive the global buffer, regardless of a stop event.
2. Latch the last state of the global buffer and drive the resulting value.
3. Force the buffer output Low.

Accelerated 8051 Microcontroller Architecture

The Accelerated 8051 microcontroller, embedded in the Triscend E5, is based on the standard 8051 device. The E5 is built around an 8-bit ALU that uses internal registers for temporary storage and to control peripheral devices. It executes the standard 8051 instruction set. A brief description of the internal blocks follows.

ALU

The ALU is the heart of the 8051 microcontroller and performs arithmetic and logical functions. It also makes decisions for jump instructions, and calculates jump addresses. The ALU is not directly user accessible, but the instruction decoder reads the op-code, decodes it, and sequences the data through the ALU and its associated registers to generate the required result. The ALU primarily uses the ACC, which is a Special Function Register (SFR) on the chip. Another SFR, the B Register, is also used in Multiply and Divide instructions. The ALU generates several status signals that are stored in the Program Status Word register (PSW).

Accumulator

The Accumulator (ACC) is the primary register used in arithmetic, logical and data transfer operations in the configurable system-on-chip. Since the Accumulator is directly accessible by the microcontroller, most of the high-speed instructions use the ACC as one argument.

B Register

This 8-bit register is used as the second argument in the MUL and DIV instructions. For all other instructions, it is simply a general-purpose register.

Program Status Word

This is an 8-bit SFR that stores the status bits of the ALU. It holds the Carry flag, the Auxiliary Carry flag, General purpose flags, the Register Bank Select, the Overflow flag, and the Parity flag.

Data Pointers

MOVX instructions use the Data Pointers to transfer data to and from Data locations. These pointers hold the address of the memory location to which data is transferred. Because data can be moved to and from external memory, the Customizable Microcontrollerprovides two separate Data Pointers. The user can switch between either of the two Data Pointers with minimum software overhead, greatly increasing system throughput.

Scratch-pad RAM

The 8051 has a 256-byte scratchpad RAM within the microcontroller. Scratchpad RAM is useful for temporary storage during program execution. Certain sections of this RAM are bit addressable, and directly addressed by the 8051's powerful Boolean instructions.

Stack Pointer

The 8051 has an 8-bit Stack Pointer that points to the top of the Stack. The stack resides in the scratchpad RAM in the processor. Consequently, the size of the stack is limited by the size of this RAM.

Timers/Counters

The Customizable Microcontrollerhas three 16-bit Timer/Counters. Each timer inhabits two SFR locations that software can read or write. Other SFR locations, associated with the timers, control their mode and operation.

Serial Port

The 8051 microcontroller provides one serial I/O port that operates in both synchronous and asynchronous modes. The serial port inhabits several SFR locations.

Memory Organization

The 8051, a classic Harvard architecture device, separates the memory into two sections, the program memory and the data memory. The program memory stores the instruction op-codes, while the data memory stores data values or accesses memory-mapped devices.

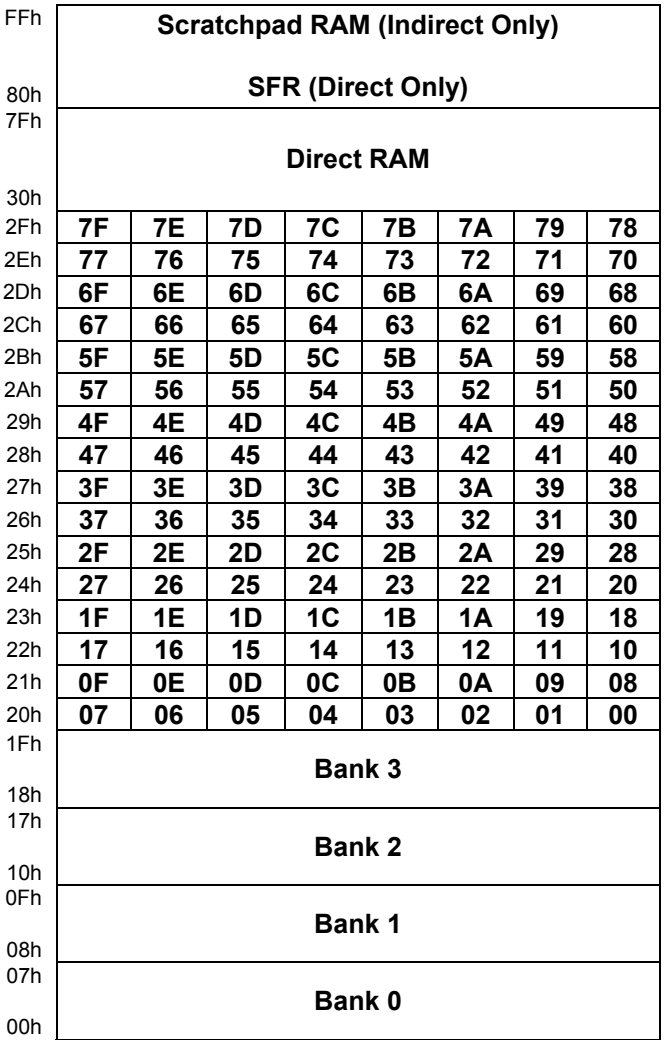


Figure 46. Scratchpad Register Addressing.

Program Memory

The 8051 supports up to 64K bytes of program memory. This memory space is either stored externally—typically in FLASH memory—or internally using the E5's internal system RAM. All instructions are fetched for execution from this memory area. The MOVC instruction also accesses this memory region.

Data Memory

The 8051 supports up to 64K bytes of data memory. This memory region is accessed by the MOVX instructions. In addition, the 8051 microcontroller has 256 bytes of scratchpad RAM, plus can access an additional 8K bytes to 64K bytes of internal system RAM—the memory size depends on the specific device. Scratchpad RAM locations are accessed either by direct addressing or by indirect addressing. The microcontroller's Special Function Registers (SFRs) can only be accessed by direct addressing.

Register Map

The 8051 has separate program and data memory areas. The on-chip 256-byte scratchpad RAM is in addition to the on-chip system RAM and any external memory. There are also several SFRs accessed by software. The SFRs are only accessible through direct addressing, while the on-chip scratchpad RAM is accessible through either direct or indirect addressing.

Since the scratchpad RAM, shown in Figure 46, is only 256 bytes deep, it can be used only when data contents are small. Larger amounts of data should be stored either in the internal system RAM or in external data memory. However, on-chip system RAM has faster access times than external RAM. There are several other special-purpose areas within the scratchpad RAM. These are described as follows.

Working Registers

There are four sets of working registers, each consisting of eight 8-bit registers. These are named Banks 0, 1, 2, and 3. Individual registers within these banks are directly accessed by separate instructions. These individual registers are named as R0, R1, R2, R3, R4, R5, R6 and R7. However, the 8051 operates with only one particular bank at a time. One of the four banks is selected by setting the RS1-RS0 bits in the PSW. The R0 and R1 registers store the address for indirect accessing.

Bit Addressable Locations

The scratchpad RAM area from location 20h to 2Fh is both byte- and bit-addressable. A bit within this region is individually addressed using the appropriate instruction. In addition, some of the SFRs are also bit addressable. The instruction decoder is able to distinguish a bit access from a byte access by the type of the instruction itself. An SFR whose address ends in a 0 or 8 is bit addressable.

Stack

The scratchpad RAM can be used for the stack. This area is selected using the Stack Pointer (SP), which stores the address of the top of the stack. Whenever a jump, call, or interrupt is invoked, the return address is placed on the stack. There is no restriction as to where the stack can begin in the RAM. By default however, the Stack Pointer contains 07h at reset, though the user can change this to any desired value. The SP points to the last used location. Therefore, the SP is incremented and then address pushed onto the stack. Conversely, while popping from the stack, the contents are first read, and then the SP is decremented.

Special Function Registers

The 8051 uses Special Function Registers (SFRs) to control and monitor peripherals and their modes.

The SFRs reside in the locations 80-FFh and are accessed by direct addressing only. Some of the SFRs are bit addressable. This allows a program to modify a particular bit without changing the others. The bit-addressable SFRs are those with addresses that end in 0 or 8. The Accelerated 8051 microcontroller contains all the SFRs present in the original 8051. However some previously unused SFRs locations are added. The list of the SFRs is shown in Table 30 with eight locations per row. Empty locations indicate that these are no SFR registers at these locations. When a bit or register is not implemented, it returns a High when read.

A brief description of the SFRs now follows.

Stack Pointer

7	6	5	4	3	2	1	0
SP.7	SP.6	SP.5	SP.4	SP.3	SP.2	SP.1	SP.0

Mnemonic: SP Address: 81h

Table 30. Special Function Register (SFR) Locations.

F8h	**EIP**							
F0h	**B**							
E8h	**EIE**							
E0h	**ACC**							
D8h	**WDCON**							
D0h	**PSW**							
C8h	**T2CON**	**T2MOD**	**RCAP2L**	**RCAP2H**	**TL2**	**TH2**		
C0h								**TA**
B8h	**IP**	**SADEN**						
B0h	**P3***							
A8h	**IE**	**SADDR**						
A0h	**P2***							
98h	**SCON**	**SBUF**						
90h	**P1***							
88h	**TCON**	**TMOD**	**TL0**	**TL1**	**TH0**	**TH1**	**CKCON**	
80h	**P0***	**SP**	**DPL**	**DPH**	**DPL1**	**DPH1**	**DPS**	**PCON**

indicates a bit-addressable location.

indicates an unused SFR register location, available for use by a “soft” module

Px* — the PIO ports P0 through P3 are implemented in CSL logic. Their SFR locations are exported. There are special precautions when exporting the P2 port (see ***Register Indirect Addressing*** section).

The Stack Pointer points to the current top of the stack, located in the scratchpad RAM.

The SP SFR is set to 07h on any reset.

There is unrestricted read/write access to this SFR.

Data Pointer (Low Byte)

7	6	5	4	3	2	1	0
DPL.7	DPL.6	DPL.5	DPL.4	DPL.3	DPL.2	DPL.1	DPL.0

Mnemonic: DPL Address: 82h

This is the low byte of the standard 8051 16-bit data pointer. The DPL is reset to 00h by a reset.

There is unrestricted read/write access to this SFR.

Data Pointer (High Byte)

7	6	5	4	3	2	1	0
DPH.7	DPH.6	DPH.5	DPH.4	DPH.3	DPH.2	DPH.1	DPH.0

Mnemonic: DPH Address: 83h

This is the high byte of the standard 8051 16-bit data pointer.

The DPH SFR is reset to 00h by a reset.

There is unrestricted read/write access to this SFR.

Second Data Pointer (Low Byte)

7	6	5	4	3	2	1	0
DPL1.7	DPL1.6	DPL1.5	DPL1.4	DPL1.3	DPL1.2	DPL1.1	DPL1.0

Mnemonic: DPL1 Address: 84h

This is the low byte of the additional 16-bit data pointer added to the Triscend configurable system-on-chip. The user can switch between DPL/DPH and DPL1/DPH1 simply by setting the DPS bit (DPS.0) in the Data Pointer Select register. The instructions using DPTR will then access DPL1 and DPH1 in place of DPL and DPH. If DPL1/DPH1 are not used, then they can be used an conventional register locations.

The DPL1 SFR is reset to 00h by a reset.

There is unrestricted read/write access to this SFR.

Second Data Pointer (High Byte)

7	6	5	4	3	2	1	0
DPH1.7	DPH1.6	DPH1.5	DPH1.4	DPH1.3	DPH1.2	DPH1.1	DPH1.0

Mnemonic: DPH1 Address: 85h

This is the high byte of the additional 16-bit data pointer added to the Triscend configurable system-on-chip. The user can switch between DPL/DPH and DPL1/DPH1 simply by setting DPS = 1. The instructions using DPTR will then access DPL1 and DPH1 in place of DPL and DPH. If DPL1/DPH1 are not used as a data pointer, then they can be used as conventional register locations.

The DPH1 SFR is reset to 00h by a reset.

There is unrestricted read/write access to this SFR.

Data Pointer Select

7	6	5	4	3	2	1	0
RESERVED (Reads = 0)							DPS.0

Mnemonic: DPS Address: 86h

DPS.0 selects either the DPL/DPH pair or the DPL1/DPH1 pair as the active Data Pointer. When set to 1, DPL1/DPH1 is selected, else DPL/DPH is selected.

DPS.1-7 are reserved but will read 0.

The DPS SFR is reset to 00h by a reset and selects the standard DPTR by default.

There is unrestricted read/write access to this SFR.

Power Control

7	6	5	4	3	2	1	0
SMOD	SMOD0	-	-	GF1	GF0	PD	IDL

Mnemonic: PCON Address: 87h

SMOD, when set, doubles the serial baud rate when the serial port is in Modes 1, 2, and 3.

SMOD0 is the Framing Error Detection Enable. When SMOD0 is set to 1, then SCON.7 indicates a Frame Error and acts as the FE flag. When SMOD0 is 0, then SCON.7 behaves as a standard 8051 function.

GF1-0 are two general-purpose user flag bits.

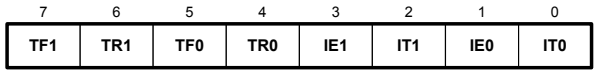
Setting the PD bit causes the Customizable Microcontrollerto go into the power-down mode. In this mode, all the clocks are stopped and program execution is halted. While in power-down mode, additional low-power options are enabled, controlled by the PWDSEL. See "*Clocking and Global Signal Distribution*" for more details.

Setting the IDL bit causes the Accelerated 8051 microcontroller to go into the IDLE mode. In this mode, the clock to the microcontroller is stopped, halting program execution. However, the clock to the serial, timer and interrupt blocks is not stopped, and these blocks continue operating unhindered.

The PCON SFR is reset to 00110000b by a reset.

There is unrestricted read/write access to this SFR.

Timer Control



Mnemonic: TCON Address: 88h

TF1 is the Timer 1 overflow flag. This bit is set when Timer 1 overflows and is cleared automatically when the program executes a Timer 1 interrupt service routine. Software can also set or clear this bit.

TR1 is the Timer 1 run control bit. This bit is set or cleared by software to turn Timer 1 on or off.

TF0 is the Timer 0 overflow flag. This bit is set when Timer 0 overflows and is cleared automatically when the program executes a timer 0 interrupt service routine. Software can also set or clear this bit.

TR0 is the Timer 0 run control bit. This bit is set or cleared by software to turn Timer 0 on or off.

IE1 is the Interrupt 1 edge-detect flag. It is set by hardware when an edge/level is detected on the INTR1 sideband signal. This bit is cleared by hardware when the service routine is vectored to only if the interrupt was edge triggered. Otherwise, it tracks the value that appears on the INIT1 sideband signal.

NOTE: *The polarity of the external interrupt signals, INTR0 and INTR1, are opposite that of the original 8051. In the original 8051, these signals are falling-edge triggered or active-Low level. In the E5 family, these signals are rising-edge triggered or active-High level.*

IT1 is the Interrupt 1 type control bit. It is set by software to specify a rising-edge trigger, cleared to specify an active-High level-triggered external input.

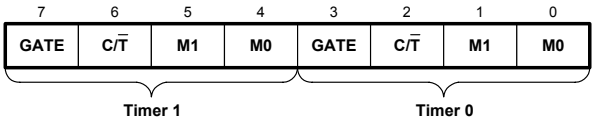
IE0 is the Interrupt 0 edge-detect flag. It is set by hardware when an edge/level is detected on the INTR0 sideband signal. This bit is cleared by hardware when the service routine is vectored to only if the interrupt was edge triggered. Otherwise, it tracks the value that appears the INIT0 sideband signal.

IT0 is the Interrupt 0 type control bit. It is set by software to specify a rising-edge trigger, cleared to specify an active-High level-triggered external input.

The TCON SFR is reset to 00h by a reset.

There is unrestricted read/write access to this SFR.

Timer Mode Control



Mnemonic: TMOD Address: 89h

GATE is the Gating control bit. When this bit is set, Timer/counter x is enabled only while INTRx pin is high and TRx control bit is set. When GATE is cleared, the TRx bit (TR0 for Timer 0, TR1 for Timer 1) enables the timer when set.

C/T is the Timer or Counter Select bit. When cleared, the timer increments on internal clocks. When set, the timer counts high-to-low edges of the Tx (T0 for Timer 0, T1 for Timer 1) sideband signal.

Table 31. Timer 0 and Timer 1 Mode Select Bits.

M1	M0	Mode
0	0	Mode 0: 8-bits with 5-bit pre-scaler.
0	1	Mode 1: 16-bits, no pre-scaler.
1	0	Mode 2: 8-bits with auto-reload from THx
1	1	Mode 3: (Timer 0) TL0 is an 8-bit timer/counter controlled by the Timer 0 control bits. TH0 is a 8-bit timer only controlled by Timer 1 control bits. (Timer 1) Timer/counter is stopped.

M1 and M0 are the Mode Select bits. They control the operation of the Timer/Counter as defined in Table 31.

The TMOD is reset to 00h by a reset.

There is unrestricted read/write access to this SFR.

Timer 0 LSB

Mnemonic: TL0 Address: 8Ah

TL0.7-0 is the least-significant byte of Timer 0.

The TL0 SFR is set to 00h on any reset.

There is unrestricted read/write access to this SFR.

Timer 1 LSB

Mnemonic: TL1 Address: 8Bh

TL1.7-0 is the least-significant byte of Timer 1.

The TL1 SFR is set to 00h on any reset.

There is unrestricted read/write access to this SFR.

Timer 0 MSB

Mnemonic: TH0 Address: 8Ch

TH0.7-0 is the most-significant byte of Timer 0.

The TH0 SFR is set to 00h on any reset.

There is unrestricted read/write access to this SFR.

Timer 1 MSB

Mnemonic: TH1 Address: 8Dh

TH1.7-0 is the most-significant byte of Timer 1.

The TH1 SFR is set to 00h on any reset.

There is unrestricted read/write access to this SFR.

TH1 potentially also controls the baud for the 8051' serial port, when the serial port operates in variable baud rate mode, *i.e.*, Mode 1 or Mode 3.

The equation for controlling the baud rate is shown below, where SMOD is PCON.7, F is the bus clock frequency, and Baud is the desired rate.

$$\{TH1\} = ROUND\left(256 - \frac{2^{SMOD} \bullet F_{busclock}}{384 \bullet Baud}\right)$$

The resulting value may be a non-integer. Round the value up or down, whichever most closely matches the desired baud rate. In some instances, selecting a different bus clock frequency will result in a value closer to the desired baud rate.

Clock Control

7	6	5	4	3	2	1	0
WD1	WD0	T2M	T1M	T0M	-	-	-

Mnemonic: CKCON Address: 8Eh

WD1-0 are the Watchdog timer mode select bits. These bits determine the time-out period for the watchdog timer. For all four of the time-out options, the reset time-out is 512 clocks after the interrupt time-out period, as shown in Table 32.

Table 32. Watchdog Timer Modes.

WD1	WD0	Interrupt time-out	Reset time-out
0	0	2^{17}	2^{17} + 512
0	1	2^{20}	2^{20} + 512
1	0	2^{23}	2^{23} + 512
1	1	2^{26}	2^{26} + 512

T2M is the Timer 2 clock select bit. When T2M is set to 1, Timer 2 uses a divide-by-4 clock. When set to 0, Timer 2 uses an 8051-compatible divide-by-12 clock.

T1M is the Timer 1 clock select bit. When T1M is set to 1, Timer 1 uses a divide-by-4 clock. When set to 0, Timer 1 uses an 8051-compatible divide-by-12 clock.

T0M is the Timer 0 clock select bit. When T0M is set to 1, Timer 0 uses a divide-by-4 clock. When set to 0, Timer 0 uses an 8051-compatible divide-by-12 clock.

CKCON.2-0 are reserved.

The CKCON SFR is set to 00h on any reset.

There is unrestricted read/write access to this SFR.

Serial Port Control

7	6	5	4	3	2	1	0
SM0/FE	SM1	SM2	REN	TB8	RB8	TI	RI

Mnemonic: SCON Address: 98h

SM0/FE performs as either Serial port, Mode 0 bit or as the Framing Error Flag. The SMOD0 bit in PCON SFR determines whether this bit acts as SM0 or as FE. The operation of SM0 is described in Table 33.

When used as FE, this bit is set to indicate an invalid stop bit. Software must clear the FE condition.

SM1 is the Serial port Mode bit 1.

SM2 is the Serial port Mode bit 2, which controls multiple MCU communication. Setting this bit enables the multiprocessor communication feature in Mode 2 and 3. In Mode 2 or 3, if SM2 is set, then RI is not activated if the received 9th data bit (RB8) is 0. In Mode 1, if SM2 = 1, then RI is only activated if a valid stop bit is received. In Mode 0, the SM2 bit controls the serial port clock. If cleared, then the serial port runs at 1/12 the system clock frequency, providing compatibility with the standard 8051. When set, the serial clock is 1/4 the system clock frequency, resulting in faster synchronous serial communication.

Table 33. Serial Mode Control Values.

SM0	SM1	Mode	Description	Length	Baud rate
0	0	0	Synch.	8	BUSCLK ÷4/÷12
0	1	1	Asynch.	10	variable
1	0	2	Asynch.	11	BUSCLK ÷64/÷32
1	1	3	Asynch.	11	variable

REN is the Receiver enable bit. Setting this bit to 1 enables serial reception, else reception is disabled.

TB8 is the 9th bit to be transmitted in Modes 2 and 3. This bit is set and cleared by software as desired.

RB8 is the received 9th data bit in Modes 2 and 3. In mode 1, if SM2 = 0, RB8 is the received stop bit value. In Mode 0, it has no function.

TI is the Transmit interrupt flag. This flag is set by hardware at the end of the 8th bit time in mode 0,

or at the beginning of the stop bit in all other modes during serial transmission. This bit must be cleared by software.

RI is the Receive interrupt flag. This flag is set by hardware at the end of the 8th bit time in mode 0, or halfway through the stop bits time in the other modes during serial reception. However, the restriction on SM2 applies on this bit. This bit can be cleared only by software

The SCON SFR is set to 00h by a reset.

There is unrestricted read/write access to this SFR.

Serial Data Buffer

Mnemonic: SBUF Address: 99h

SBUF.7-0 is the serial data, read from or written to this location. It actually consists of two separate 8-bit registers. One is the receive register and the other is the transmit buffer.

Any read access gathers data from the receive data buffer, while write transfers are to the transmit data buffer.

The SBUF SFR is set to 00h by a reset.

There is unrestricted read/write access to this SFR.

Interrupt Enable

7	6	5	4	3	2	1	0
EA	-	ET2	ES	ET1	EX1	ET0	EX0

Mnemonic: IE Address: A8h

EA, when set, is the global enable bit and enables all interrupts. When cleared, this bit disables all interrupts, except for HPINT.

IE.6 is an un-implemented bit. It returns a High when read.

ET2, when set, enables the Timer 2 interrupt.

ES, when set, enables the Serial interrupt.

ET1, when set, enables the Timer 1 interrupt.

EX1, when set, enables External interrupt 1.

ET0, when set, enables the Timer 0 interrupt.

EX0, when set, enables External interrupt 0.

The IE SFR is set to 01000000b by a reset.

There is unrestricted read/write access to this SFR.

Slave Address

Mnemonic: SADDR Address: A9h

SADDR is used only during multiple MCU operations involving the serial port.

SADDR is loaded with the given or broadcast address for the serial port if the E5 is used as a slave processor during multiprocessor communications.

The SADDR SFR is set to 00h by a reset.

There is unrestricted read/write access to this SFR.

Interrupt Priority

7	6	5	4	3	2	1	0
-	-	PT2	PS	PT1	PX1	PT0	PX0

Mnemonic: IP Address: B8h

The IP.7-6 bits are un-implemented and return a High when read.

PT2, when set, defines the Timer 2 interrupt as a higher priority interrupt.

PS, when set, defines the Serial port interrupt as a higher priority interrupt.

PT1, when set, defines the Timer 1 interrupt as a higher-priority interrupt.

PX1, when set, defines the External Interrupt 1 as a higher-priority interrupt.

PT0, when set, defines the Timer 0 interrupt as a higher-priority interrupt.

PX0, when set, defines External Interrupt 0 as a higher-priority interrupt.

The IP SFR is set to 11000000b by a reset.

There is unrestricted read/write access to this SFR.

Slave Address Mask Enable

Mnemonic: SADEN Address: B9h

The SADEN register enables the Automatic Address Recognition feature of the Serial port. When a bit in the SADEN is set to 1, the same bit location in SADDR is compared with the incoming serial port data. When SADEN is 0, then the bit becomes a don't-care in the comparison. When all the bits of SADEN are 0, an interrupt occurs for any incoming address.

The SADEN SFR is set to 00h by a reset.

There is unrestricted read/write access to this SFR.

Timed Access

Mnemonic: TA Address: C7h

The Timed Access (TA) register controls the access to protected Watchdog controller bits in the 8051 microcontroller. A separate timed access register (PROTECT) supports the secure initialization mode. To access protected bits, application software must first write AAh to the TA SFR. This must be immediately followed by a write of 55h to

TA. This opens a window for three machine cycles, during which time software can write to these protected bits.

The TA returns FFh when read.

There is unrestricted write access to this SFR, a read is not required.

Example:

```
MOV TA, 0AAH  ; Write 'AA' followed by '55' to open
              ; window
MOV TA, 055H  ; Window now open for three machine
              ; cycles
SETB RWT      ; Reset watchdog timer
```

Timer 2 Control

7	6	5	4	3	2	1	0
TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/$\overline{T2}$	CP/$\overline{RL2}$

Mnemonic: T2CON Address: C8h

TF2 is the Timer 2 overflow flag. This bit is set when Timer 2 overflows. It is also set when the count equals the capture register value while in down count mode. It can be set only if RCLK and TCLK are both 0. It is only cleared by software. Software can also set or clear this bit.

EXF2 is the Timer 2 External Flag. A falling-edge transition on the T2EX sideband signal or a Timer 2 underflow/overflow condition sets this flag based on the CP/RL2, EXEN2 and DCEN bits. If set by a falling-edge transition, this flag must be cleared by software. Setting this bit in software or a detected falling-edge transition on T2EX pin forces a timer interrupt, if enabled.

RCLK is the Receive clock flag. This bit determines the serial port timebase when receiving data in serial modes 1 or 3. If this bit is 0, then Timer 1 overflow is used for baud rate generation, else Timer 2 overflow is used. Setting this bit forces Timer 2 into baud-rate generator mode.

TCLK is the Transmit clock flag. This bit determines the serial port timebase when transmitting data in serial modes 1 and 3. If it is set to 0, the Timer 1 overflow is used to generate the baud rate clock, else Timer 2 overflow is used. Setting this bit forces Timer 2 into baud-rate generator mode.

EXEN2 is the Timer 2 External Enable bit. This bit enables the capture/reload function on the T2EX sideband signal if Timer 2 is not generating baud clocks for the serial port. If this bit is 0, then the T2EX sideband signal is ignored, else a falling-edge transition detected on the T2EX sideband signal results in capture or reload.

TR2 is the Timer 2 Run Control bit. This bit enables/disables the operation of Timer 2. Halting Timer 2 by clearing this bit preserves the current count value in TH2, TL2.

C/T2 is the Counter/Timer select bit. This bit determines whether Timer 2 functions as a timer (counting clock cycles) or as a counter. Independent of this bit, the timer runs at 2 clocks per tick when used in baud rate generator mode. If it is set to 0, then Timer 2 operates as a timer at a speed depending on T2M bit (CKCON.5). Otherwise, it counts falling edges on the T2 sideband signal.

CP/RL2 is the Capture/Reload Select bit. This bit determines whether the capture or reload function is used for Timer 2. If either RCLK or TCLK is set, this bit will not function and the timer will function in an auto-reload mode following each overflow. If the bit is 0, then auto-reload occurs whenever Timer 2 overflows or a falling edge is detected on T2EX if EXEN2 = 1. If this bit is 1, Then Timer 2 captures happen whenever a falling edge is detected on T2EX if EXEN2 = 1.

The T2CON is reset to 00h by a reset.

There is unrestricted read/write access to this SFR.

Timer 2 Mode Control

7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	DCEN

Mnemonic: T2MOD Address: C9h

DCEN is the Down Count Enable bit. This bit, in conjunction with the T2EX sideband signal controls the direction that Timer 2 counts in 16-bit auto-reload mode.

When DCEN is set, Timer 2 counts up or down, controlled by the T2EX sideband signal. When High, T2EX causes Timer 2 to increment. When Low, Timer 2 decrements. When DCEN=0, Timer 2 only counts up.

The T2MOD is reset to FEh by a reset.

There is unrestricted read/write access to this SFR.

Timer 2 Capture LSB

Mnemonic: RCAP2L Address: CAh

RCAP2L is the least-significant byte of the Timer 2 capture register. This register is used to capture the TL2 value when a Timer 2 is configured in capture mode. RCAP2L is also used as the least-significant byte of a 16-bit reload value when Timer 2 is configured in auto-reload mode.

The RCAP2L SFR is set to 00h on any reset.

There is unrestricted read/write access to this SFR.

Registers RCAP2H and RCAP2L potentially also control the baud for the 8051's serial port, when

the serial port operates in variable baud rate mode, *i.e.*, Mode 1 or Mode 3.

The equation for controlling the baud rate is shown below, where {RCAP2H,RCAP2L} represent a 16-bit binary value, F is the bus clock frequency, and Baud is the desired rate.

$$\{RCAP2H, RCAP2L\} = ROUND\left(65536 - \frac{F_{busclock}}{32 \bullet Baud}\right)$$

The resulting value may be a non-integer. Round the value up or down, whichever most closely matches the desired baud rate. In some instances, selecting a different bus clock frequency will result in a value closer to the desired baud rate.

Timer 2 Capture MSB

Mnemonic: RCAP2H Address: CBh

RCAP2H is the least-significant byte of the Timer 2 capture register. This register is used to capture the TH2 value when a Timer 2 is configured in capture mode. RCAP2H is also used as the most-significant byte of a 16-bit reload value when Timer 2 is configured in auto-reload mode.

The RCAP2H SFR is set to 00h on any reset.

There is unrestricted read/write access to this SFR.

Registers RCAP2H and RCAP2L potentially also control the baud for the 8051's serial port, when the serial port operates in variable baud rate mode, *i.e.*, Mode 1 or Mode 3. See description for Timer 2 Capture LSB.

Timer 2 LSB

Mnemonic: TL2 Address: CCh

TL2 is the least-significant byte of Timer 2.

The TL2 SFR is set to 00h on any reset.

There is unrestricted read/write access to this SFR.

Timer 2 MSB

Mnemonic: TH2 Address: CDh

TH2 is the most-significant byte of Timer 2.

The TH2 SFR is set to 00h on any reset.

There is unrestricted read/write access to this SFR.

Program Status Word

7	6	5	4	3	2	1	0
CY	AC	F0	RS1	RS0	OV	F1	P

Mnemonic: PSW Address: D0h

CY is the Carry flag. This bit is set by arithmetic operations that generate a carry in the ALU. It is also used as the accumulator for the bit operations.

AC is the Auxiliary carry bit. This bit is Set when the previous operation resulted in a carry during an addition operation or resulted in a borrow during a subtraction from the high order nibble.

F0 is User flag 0, a general-purpose flag that can be set or cleared by the user by software.

RS.1-0 select the active Register bank as shown in Table 34.

Table 34. Register Bank Select.

RS1	RS0	Register bank	Address
0	0	0	00-07h
0	1	1	08-0Fh
1	0	2	10-17h
1	1	3	18-1Fh

OV is the Overflow flag. Set by arithmetic operations. Indicates that two most significant accumulator bits are different. Used primarily with signed or twos-compliment data to indicate that the arithmetic operation overflowed the accuracy of the accumulator.

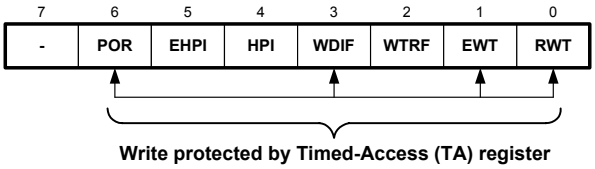
F1 is User Flag 1, a general-purpose flag that can be set or cleared by the user by software

P is the Parity flag. This bit is set or cleared by hardware to indicate odd/even number of 1's in the accumulator.

The PSW SFR is set to 00h by a reset.

There is unrestricted read/write access to this SFR.

Watchdog Control



Mnemonic: WDCON Address: D8h

WDCON.7 is un-implemented and returns a High when read.

POR is the Power-On Reset flag. Hardware sets this flag on a power-up condition. This flag can be read or written by software. A write by software is the only way to clear this bit once it is set.

EHPI is the Enable High-Priority Interrupt bit. When set to 1, this bit enables the High-Priority Interrupt. If cleared, then the high-priority sideband signal (HPINT) is ignored.

HPI is the High-Priority Interrupt flag. This flag is set by hardware whenever the high-priority side-band signal (HPINT) is asserted. Software must clear this bit. Writing a 1 to this bit forces an interrupt if the EHPI bit is also set.

NOTE:	*Some of the bits in the Watchdog Control register (WDCON) are protected. They cannot be accidentally overwritten by an errant program. Before attempting to change the RWT, EWT, WDIF, or POR bits, you must write the proper sequence to the Timed Access (TA) register.*

WDIF is the Watchdog Timer Interrupt Flag. If the watchdog interrupt is enabled, hardware sets this bit to indicate that a watchdog interrupt occurred. If the interrupt is not enabled, then this bit indicates that the time-out period has elapsed.

WTRF is the Watchdog Timer Reset Flag. Hardware sets this bit whenever the watchdog timer causes a reset. Software must clear this bit. A Power-On or System Reset event also clears the bit. This bit helps software to determine the cause of a reset. If EWT = 0, the watchdog timer has no affect on this bit.

EWT is the Enable Watchdog Timer reset. Setting this bit enables the Watchdog Timer Reset function.

RWT is the Reset Watchdog Timer bit. This bit forces the watchdog timer into a known state. Setting this bit via software resets the watchdog timer, usually before a watchdog time-out occurs. If the RWT bit is not set before a watchdog time-out happens, then two possible subsequent events occur. If a watchdog time-out occurs while the EWDI (EIE.4) interrupt enable bit is set, then a watchdog interrupt results. If a watchdog time-out occurs, then 512 clock cycles later, the Watchdog Timer resets the MCU if EWT is set. Setting RWT during this 512 clock period resets the Watchdog Timer, avoiding the MCU reset. This bit is self-clearing.

The WDCON SFR is set to a 1x0x0xx0b on an external reset, an 'x' indicating a bit unaffected by reset. WTRF is set to a 1 on a Watchdog timer reset, but to a 0 on power on/down resets. WTRF is not altered by an external reset. POR is set to 1 by a power-on reset. EWT is set to 0 on a Power-on reset and unaffected by other resets.

NOTE:	*When the MCU is reset, most Watchdog Timer control bits are unaffected. If the Watchdog Timer is enabled before the reset, it remains enabled afterward. However, the Watchdog Timer time-out period is controlled by the CKCON register, which is reset along with the MCU. Consequently, a reset changes the Watchdog Timer time-out value to its minimum value.*

All the bits in this SFR have unrestricted read access. POR, EWT, WDIF and RWT require timed access write. The remaining bits have unrestricted write accesses.

Accumulator

Mnemonic: ACC Address: E0h

ACC.7-0, the A or ACC register, is the standard 8051 accumulator

The ACC is reset to 00h on any reset

This SFR has unrestricted read/write access.

Extended Interrupt Enable

7	6	5	4	3	2	1	0
-	-	-	EWDI	-	-	-	-

Mnemonic: EIE Address: E8h

EIE.7-5 are reserved bits and return a High when read.

EWDI enables the Watchdog timer interrupt.

EIE.3-0 are reserved bits and return a High when read.

The EIE SFR is set to 11101111b on any reset.

This SFR has unrestricted read/write access.

B Register

Mnemonic: B Address: F0h

The B register is used during multiply and divide operations. For other instructions, it can be treated as another scratch pad register.

The B register is reset to 00h on any reset

This SFR has unrestricted read/write access.

Extended Interrupt Priority

7	6	5	4	3	2	1	0
-	-	-	PWDI	-	-	-	-

Mnemonic: EIP Address: F8h

EIP.7-5 are reserved bits and return a High when read.

PWDI, when set, defines the Watchdog timer as a higher-priority interrupt.

EIP.3-0 are reserved bits and return a High when read.

The EIP SFR is set to 11101111b on any reset.

This SFR has unrestricted read/write access.

Instruction Set

The Triscend Customizable Microcontroller- executes all the instructions of the standard 8051 family. The operation of these instructions, their effect on the flag bits and the status bits is exactly the same. However, timing of these instructions is different. There are two reasons for the timing differences.

First, each Accelerated 8051 microcontroller machine cycle takes 4 clock periods, while in the original 8051 requires 12 clock periods. Second, the "Turbo" microcontroller only performs one fetch per machine cycle, *i.e.* 4 clocks per fetch. The standard 8051 can require two fetches per machine cycle, or 6 clocks per fetch.

The Accelerated 8051 microcontroller's advantage is that there is only one fetch per machine cycle. In most cases, the number of machine cycles is equal to the number of operands required by the instruction. Jumps and calls require an additional cycle to calculate the new address. Overall, the Accelerated 8051 microcontroller reduces the number of dummy fetches and wasted cycles, thereby improving efficiency compared to the original 8051.

Addressing Modes

The 8051 microcontroller uses eight different addressing modes. These modes are as follows, with detailed descriptions below.

1. Register addressing
2. Direct addressing
3. Register Indirect addressing
4. Immediate addressing
5. Base register plus Index register Indirect addressing
6. Relative addressing
7. Absolute addressing
8. Long addressing

Register Addressing

Register addressing uses the eight working registers (R0-R7) of the current register bank located in the scratchpad RAM. The last three bits in the instruction op-code indicate the selected register. The current register bank is one of four register banks residing in the scratchpad RAM. The two bits RS1-RS0 in the Program Status Word (PSW) select the working bank. Consequently, the same instruction can access different registers simply by switching between banks. ACC, B, the current DPTR determined by DPS.0, and CY can also be addressed as registers.

Example:

```
ADDC A, R2    ; Add Accumulator to register R2 with carry.
DEC R7        ; Decrement contents of register R7
CLR A         ; Clear the Accumulator
SETB C        ; Set the Carry flag (Boolean Processor
              ; Accumulator)
```

Direct Addressing

Direct addressing is the only method to refer to the Special Function Registers (SFRs). This mode accesses the entire lower 128 bytes of the Scratchpad RAM.

Direct addressing also applies to bit operations. The Scratchpad RAM area from 20h to 2Fh is bit-addressable. The bits in this area have their own unique bit addresses from 00h to 7Fh. These bits may be addressed by directly specifying the bit address. Several SFRs are also bit addressable. The bits in these SFRs are addressed by adding the bit position to the SFR address.

Example:

```
MOV 20h, 21h  ; Move contents of RAM location 20h to
              ; RAM location 21h
MOV P1, P3    ; Move data at Port 1 Pins to Port 3 latch
SETB 7Fh      ; Set bit 7Fh in Scratchpad RAM. This is
              ; actually MSB of the RAM location 2Fh.
MOV TR1, C    ; Move carry flag contents to TR1 flag in
              ; TCON SFR
```

Register Indirect Addressing

Register Indirect addressing mode uses the R0 and R1 registers to store the address of the selected Scratchpad RAM location. By changing the contents of R0 or R1, the same instruction will address different RAM locations. However, this addressing mode is limited to the 128 bytes of Scratchpad RAM only (the upper 128 bytes). Special Function Registers (SFRs) cannot be addressed by this method.

Example:

```
MOV R0, #20h  ; Load register R0 with 20h
MOV A, @R0    ; Move contents of RAM location 20h to
              ; the Accumulator
```

Stack operations are also examples of Register Indirect addressing. In this case, the SFR Stack Pointer (SP) stores the address rather than the working registers R0 and R1. Instructions like PUSH and POP use this mode of addressing.

Example:

```
PUSH TCON     ; Save the contents of SFR TCON onto the
              ; stack
POP DPL       ; Loads the SFR DPL with the data at the
              ; top of the stack
```

Data Memory may also be accessed using Register Indirect Addressing mode. The MOVX instruc-

tions are used for this purpose. The registers R0, R1, DPTR or DPTR1 can be used as pointers for this purpose. When using R0 and R1 as pointers, the lower 8-bits of the address are the contents of R0 or R1, while the upper 8-bits of the address are supplied by the P2 SFR. The P2 SFR must therefore be set to the correct value before using this instruction. When using DPTR or DPTR1 as the pointer, then the entire 16-bit address is taken from the selected Data Pointer.

Example:

```
MOVX @R0, A    ; Moves the Accumulator contents to the
               ; external Data Memory locations pointed
               ; to by R0 in the page pointed to by the P2
               ; SFR.

MOVX A, @DPTR  ; Move data from the Data Memory pointed
               : to by the selected Data Pointer, into the
               : Accumulator.
```

Immediate Addressing

Immediate addressing is used when part of the opcode instruction is a constant. This method is most commonly used to initialize registers and SFRs or to perform mask operations.

Example:

```
MOV A, #40h     ; Mov 40h to Accumulator

XRL SCON, #FFh    ; Xor SCON SFR with FFh, thereby
                  ; complementing its contents
```

Base Register plus Index Register Indirect Addressing

This addressing mode provides access to a byte from the 8051's program memory—separate from data memory—via an indirect move. The address of the move location is the sum of the base register (PC or DPTR/DPTR1) and an index register (ACC). The result is always written to the ACC, overwriting the index value that previously resided there.

Example:

```
MOV DPTR, #1234h      ; Data Pointer is loaded with
                  constant
                  ; 1234h

MOV A, #23h         ; Accumulator is loaded with index
                    ; value 23h

MOVC A, @A+DPTR   ; Accumulator is loaded with byte from
                  ; the Program Memory at location
                  ; 1234h + 23h = 1257h
```

Relative Addressing

Relative Addressing is used to specify the destination address for jumps within 128 bytes of the current program counter, common is short looping functions.

The destination address is given in the form of a two's complement address offset that is added to the PC. The relative address is one byte long, and so the offset will vary from -128 to +127.

Example:

```
SJMP 20h      ; Jump to a location 20h bytes after the current
              ; instruction. If the SJMP instruction is at
              ; 1010h, then the jump will be to location
              ; 1010h + 2h + 20h = 1032h

JZ FEh        ; If the Accumulator is zero, then jump to the
              ; same instruction ; If the JZ instruction is at
              ; 2359h, then the jump will be to location
              ; 2359h + 2h + FEh = 2359h + 2h - 2h = 2359h
```

Absolute Addressing

Absolute addressing is used to specify the destination address in ACALL and AJMP instructions. The operand is an 11-bit address within the current 2K block of program memory.

In this mode of addressing, the 16-bit address is generated by taking the five highest order bits of the next instruction (PC + 2) and concatenating the lowest order 11 bit field contained in the current instruction. The resulting addressed location lies within the 2K page of the Program Memory, relative to the first byte of the instruction following the current instruction.

Example:

```
ACALL 0200h ; If the current instruction is at location
            ; 7FFFh, then a call will be made to the
            ; location 8200h
```

Long Addressing

Long addressing specifies the full 16-bit address anywhere within the 64K program memory for the LJMP and LCALL instructions.

Example:

```
LJMP 8000h  ; Jump to location 8000h in program memory.
```

Interrupts

The Triscend E5 has a three priority level interrupt structure with 12 interrupt sources. Each of the interrupt sources has an individual flag, interrupt vector and enable bit. The standard 8051 microcontroller interrupts have individual priority bits. Furthermore, the interrupts—with the exception of the High-Priority Interrupt—can be globally enabled or disabled.

Interrupt Sources

The external interrupt sideband signals, INTR0 and INTR1, are either edge triggered or level triggered, depending on bits IT0 and IT1 in the TCON register. The flag bits IE0 and IE1 in the TCON register generate the interrupt and indicate which interrupt has occurred. When an external interrupt input generates an interrupt, the appropriated flag bit is cleared upon entering the Interrupt Service routine,

but only if the interrupt type is edge triggered. In level activated mode, the external requesting source controls the interrupt flag bit rather than the on-chip hardware. For level-triggered interrupts, the IE0 and IE1 flags are not cleared by hardware and must be cleared by the software.

The TF0 and TF1 flags generate the Timer 0 and 1 Interrupts. These flags are set when Timer 0 and Timer 1 overflow, respectively. Hardware automatically clears the TF0 and TF1 flags when the timer interrupt is serviced.

The Timer 2 interrupt is generated by a logical OR of the TF2 and the EXF2 flags. These flags are set by overflow/underflow or capture/reload events in the Timer 2 operation. The hardware does not clear these flags when a Timer 2 interrupt is executed. Software has to resolve the cause of the interrupt between TF2 and EXF2 and clear the appropriate flag.

The Watchdog timer can be used as a system monitor or a simple timer. In either case, the Watchdog timer interrupt flag WDIF (WDCON.3) is set when the time-out count is reached. If the interrupt is enabled by the enable bit EIE.4, then an interrupt occurs.

The serial port also generates interrupts on data reception or transmission. However, there is only one interrupt source from the Serial block, the logic OR of the RI and TI bits in the SCON SFR. These bits are not automatically cleared by the hardware and the software must clear these bits.

The High-Priority Interrupt flag, HPI, (WDCON.4) is set when the HPINT sideband signal is asserted. If this interrupt is enabled then an interrupt occurs. The high-priority interrupt has the highest priority over all the interrupts. The user cannot alter its priority but it can be enabled or disabled via software. The EHPI bit enables the high-priority interrupt. This bit is not controlled by the global interrupt enable EA.

The standard 8051 interrupt flags that generate interrupts can be set or reset by writing to the appropriate registers. Consequently, software can generate interrupts. The addition E5 interrupts (DMA, JTAG, software and hardware breakpoint) can only by reset by software. Individual interrupts can be enabled or disabled by setting or clearing a bit in the IE or the EIE SFR. IE also has a global enable/disable bit, EA, which is cleared to disable all the interrupts at once, except for the high-priority interrupt (HPI), which is unaffected by EA.

Priority Level Structure

There are three priority levels for the interrupts, highest, high and low. The high-priority interrupt HPI has the highest priority. No other interrupt can have this priority. Naturally, a higher priority interrupt cannot be interrupted by a lower priority interrupt. However there is a predefined hierarchy among the interrupts themselves. This hierarchy helps the interrupt controller to resolve simultaneous requests having the same priority level.

The default hierarchy is defined as shown Table 35. The interrupts are numbered starting from the highest priority to the lowest. Some of the interrupts have an individual priority bit that, when set, places those interrupts in a higher-priority category.

The default priority remains for interrupts placed in the higher-priority category.

Table 35. Priority structure of interrupts.

Source	Flag	Priority Level	Priority Bit
High-Priority	HPI	1 (highest)	
External Interrupt 0	IE0	2	✓
Timer 0 Overflow	TF0	3	✓
External Interrupt 1	IE1	4	✓
Timer 1 Overflow	TF1	5	✓
Serial Port	RI TI	6	✓
Timer 2 Overflow	TF2 EXF2	7	✓
DMA	OVR0 INIT0 TC0 OVR1 INIT1 TC1	8	
Hardware Breakpoint	BP0I BP1I	9	
JTAG	JTAGI	10	
Software Breakpoint (A5 instruction)	EA5	11	
Watchdog Timer	WDIF	12 (lowest)	✓

The interrupt flags are sampled in C2 of every machine cycle. In the same machine cycle, the sam-

pled interrupts are polled and their priority is resolved. If certain conditions are met, then the hardware executes an internally generated LCALL instruction that vectors the process to the appropriate interrupt vector address. The conditions for generating the LCALL are …

1. An interrupt of equal or higher priority is not currently being serviced.
2. The current polling cycle is the last machine cycle of the instruction currently being executed.
3. The current instruction does not involve a write to IP, IE, EIP or EIE registers and is not a RETI.

An LCALL is only generated if all of the conditions are met. The polling cycle is repeated every machine cycle, with the interrupts sampled in C2 of the same machine cycle. If an interrupt flag is active in one cycle but not responded to, and is not active when the above conditions are met, the denied interrupt will not be serviced. This means that active interrupts are not remembered. Every polling cycle is new.

Table 36. Interrupt Sources and Reset Method.

Interrupt Source	Flag	Cleared by Hardware	Cleared by Software
HPINT	HPI		✓
INTR0	IE0	Edge-triggered	Level-triggered
Timer 0	TF0	✓	
INTR1	IE1	Edge-triggered	Level-triggered
Timer 1	TF1	✓	
Serial Port	SI		✓
	TI		✓
Timer 2	TF2 EXF2		✓
DMA Controller	OVR0		✓
	INIT0		✓
	TC0		✓
	OVR1		✓
	INIT1		✓
	TC1		✓
Watchdog	WDIF		✓

The processor responds to a valid interrupt by executing an LCALL instruction to the appropriate service routine. Hardware may or may not clear the flag that caused the interrupt, as shown in Table 36. For Timer interrupts, the TF0 or TF1 flags are cleared by hardware whenever the processor vectors to the appropriate timer service routine. With External interrupts, INTR0 and INTR1, the flags are cleared only if they are edge triggered. For Serial interrupts, the flags are not cleared by hardware. The ISR must poll to determine if the mine if the transmitter or receiver generated the interrupt and clear the appropriate flags. With the Timer 2 interrupt, the flag is not cleared by hardware. The Power-Fail Interrupt flag (PFI) and Watchdog timer interrupt flag (WDIF) must be cleared by software. The hardware LCALL behaves exactly like the software LCALL instruction. This instruction saves the Program Counter (PC) contents onto the Stack, but does not save the Program Status Word (PSW).

The Program Counter (PC) is reloaded with the vector address of the interrupt that caused the LCALL. The vector addresses for the different sources are as shown in Table 37. Note that the vector addresses are not evenly spaced in memory.

Execution continues from the vectored address until an RETI instruction is executed. On executing the RETI instruction, the processor pops the Stack and loads the PC with the contents at the top of the stack. The user's software must restore the stack to its values prior to the hardware LCALL when execution returns to the interrupted program. All interrupt service routines end with the Return From Interrupt (RETI) instruction. Note that a RET instruction performs like a RETI instruction, but it does not inform the Interrupt Controller when the interrupt service routine is complete, leaving the controller to think that the service routine is in progress.

Table 37. Vector locations for interrupt sources.

Source	Vector Address	Keil Intr. #
High-Priority Interrupt	0x0033	6
External Interrupt 0	0x0003	0
Timer 0 Overflow	0x000B	1
External Interrupt 1	0x0013	2
Timer 1 Overflow	0x001B	3
Serial Port	0x0023	4
Timer 2 Interrupt	0x002B	5
DMA	0x003B	7
Hardware Breakpoint	0x0043	8
JTAG	0x004B	9
Software Breakpoint	0x0053	10
Watchdog Timer	0x0063	12

External Interrupts

There are three external interrupt sources in this processor. The High-Priority Interrupt is supplied via the HPINT sideband signal. This interrupt has priority over all other interrupts and typically connects to an external power-fail signal. The other external interrupts are INTR0 and INTR1, also

available as sideband signals. These interrupts can be programmed to be edge triggered or level activated, by setting bits IT0 and IT1 in TCON SFR.

NOTE: *The polarity of the external interrupt signals, INTR0 and INTR1, are opposite that of the original 8051. In the original 8051, these signals are falling-edge triggered or active-Low level. In the E5 family, these signals are rising-edge triggered or active-High level.*

In the edge-triggered mode, the INTR0 and INTR1 sideband signals are sampled at C4 in every machine cycle. If the sample is Low in one cycle and High in the next, then a low-to-high transition is detected and the interrupts request flag IE0 or IE1 is set in the timer control register, TCON. The flag bit then requests the interrupt. Because external interrupts are sampled every machine cycle, they must be held High or Low for at least one complete machine cycle. Asserting the signal for five bus clock cycles guarantees sufficient duration. The interrupt service routine automatically clears the appropriate IE0 or IE1 flag.

In level activated mode, the requesting source must hold the pin High until the interrupt is serviced. The IE0 or IE1 flag is not cleared by the hardware on entering the service routine.

NOTE: *The external interrupt signals, including edge-triggered interrupts, must be asserted for a minimum of one 8051 machine cycle. Asserting the signal for five bus clock cycles guarantees sufficient duration.*

If the interrupt continues to be held High—even after the service routine is completed—then the processor may acknowledge another interrupt request from the same source.

Response Time

The response time for each interrupt source depends on several factors like nature of the interrupt, the instruction currently executing, and wait-states on the CSI bus.

External interrupts INTR0 and INTR1 are sampled at C3 of every machine cycle and then their corresponding interrupt flags IE0 and IE1 are set or reset. Similarly, the Serial port flags RI and TI are set in C3. The Timer 0 and 1 overflow flags are set at C3 of the machine cycle in which overflow has occurred. These flag values are polled only in the next machine cycle. If a request is active and all three conditions are met, then the hardware-generated LCALL is executed. This call itself takes four machine cycles to complete.

Consequently, there is a minimum latency of five machine cycles between the time the interrupt flag is set and the interrupt service routine begins executing. A longer response time should be anticipated if any of the following three conditions is met.

1. An interrupt of equal or higher priority is currently being serviced.
2. The current polling cycle is not the last machine cycle of the instruction currently being executed.
3. The current instruction involves a write to IP, IE, EIP or EIE registers and is not a RETI.

If a higher or equal priority interrupt is being serviced, then the interrupt latency time depends on the service routine currently executing. If the polling cycle is not the last machine cycle of the instruction being executed, then an additional delay is introduced. The maximum response time—if no other interrupt is currently being serviced—occurs if the 8051 is performing a write to IE, IP, EIE or EIP and then executes a MUL or DIV instruction. From the time an interrupt source is activated, the longest response time is 12 machine cycles. This includes …

- 1 machine cycle to detect the interrupt
- 2 machine cycles to complete the IE, IP, EIE or EIP access
- 5 machine cycles to complete the MUL or DIV instruction
- 4 machine cycles to complete the hardware LCALL to the interrupt vector location.

Thus in a single-interrupt system, the interrupt response time is always more than 5 machine cycles but not more than 12 machine cycles. The maximum latency of 12 machine cycles is 48 clock cycles. In the original 8051, the maximum latency is 8 machine cycles or 96 clock cycles. Consequently, the E5's interrupt response time is two times faster than an average 8051.

Table 38. Interrupt Response Latency (Single-Interrupt System).

	Minimum	Maximum
Machine Cycles	5	12
Clock Cycles	20	48

Reset Conditions

Several possible conditions reset the Triscend E5 configurable system-on-chip. Different reset sources affect the device differently.

Table 39. Reset Types and Affected Resources.

	RESET CONDITION					
	Power-On Reset	RST- Pin	JTAG Command (FORCE_BRST)	Application Reset (RSTC)	Watchdog Timer Reset	JTAG Command (J_RESET)
Device Resources Affected						
Accelerated 8051 MCU	✓	✓	✓	✓	✓	✓
DMA Controller	✓	✓	✓			
Memory Interface Unit (MIU)	✓	✓	✓			
Configuration Registers (CRU)	✓	✓	✓			
Clock control registers	✓	✓				
JTAG Unit	✓	✓				
Post-Reset Behavior						
(Re-) Initialize Device	✓	✓	✓			
Begin Executing Code from 0000h	✓	✓	✓	✓	✓	✓

For example, a Power-On Reset (POR) condition resets all of the device functions including the Accelerated 8051 microcontroller, peripherals connected to the internal CSI bus, many of the internal system registers (CRU), the clock logic, and the JTAG unit.

By contrast, asserting the RST- pin resets most of the device, but leaves the clock control registers and JTAG unit unaffected.

Finally, a watchdog reset or asserting the application reset sideband signal, RSTC, only affects the embedded 8051 microcontroller. All other system logic is unaffected.

Table 39 summarizes the various reset conditions and the system resources that each affects. The table indicates how the device behaves after the reset. Some reset conditions cause the device to re-initialize and restart code execution from 0000h. Others merely restart code execution.

Power-On Reset (POR)

The on-chip analog circuitry continuously monitors the VCC levels and detects a power on or power fail condition. As long as VCC is below the threshold VRST, the device remains in the reset condition. Once VCC goes above the VRST level, the analog circuit releases the reset.

On-chip logic generates a power-on reset (POR) when power is applied to the device, resetting all device resources that can be reset. POR is released after a clock source is available (after the first two rising clock edges).

The software should clear the POR flag after reading it, otherwise it will not be possible to correctly determine future reset sources. If the power fails, *i.e.* falls below VRST, then the device again goes into the reset state. When the power returns to the proper operating levels, the device again performs a power on reset and sets the POR flag.

RST- Device Pin

The RST- pin, when asserted Low, resets all internal resources except for the clock control registers and the JTAG unit. The assertion of the reset is asynchronous but its de-assertion is automatically synchronized with bus clock signal, BCLK. A glitch filter prevents glitches on the RST- pin lasting less than 2ns. RST- should always be asserted with a signal lasting more than 8 ns in duration.

The RST- pin can be disabled by two conditions. First, the RST- pin is ignored if the Customizable Microcontroller is placed in Secure mode by setting the MIU bit (SECURITY.0) in the Security register. This condition is cleared once power is removed from the device. Second, a FORCE_NOBRST command can be issued via the JTAG unit during debugging. This JTAG-initiated condition is cleared once the command is no longer asserted.

JTAG Reset

For debugging purposes, the JTAG unit can force a System Reset or an MCU Reset event. There are three reset-related commands that can be issued via the JTAG unit.

1. J_RESET resets the MCU.
2. FORCE_BRST resets everything except the watchdog timer, the clock control registers, and the JTAG unit.
3. FORCE_NOBRST disables the RST- pin, preventing the pin from causing a reset event. This function overrides FORCE_BRST.

Application Reset via RSTC Sideband Signal

The reset input to the Accelerated 8051 microcontroller is supplied via an active-High sideband signal called RSTC. When asserted, RSTC resets the MCU but no other peripherals or system-related resources. If enabled before the reset, the Watchdog Timer remains enabled, but there are side-effects. The RSTC sideband signal is usually supplied by functions in the CSL matrix or from a PIO pin.

Watchdog Timer Reset

The Accelerated 8051 microcontroller contains a watchdog timer, typically used to insure proper system operation. The Watchdog timer is a free running timer with programmable time-out intervals. The user's software can clear the watchdog timer at any time causing it to restart the count. When the timer reaches its time-out interval, an interrupt flag is set. If the Watchdog reset is enabled then the watchdog timer generates a reset under the following condition.

1. If the watchdog timer reset is enabled by setting EWT (WDCON.1).
2. When the watchdog timer reaches its time-out count.
3. If the application software does not reset the Watchdog timer by setting the RWT bit (WDCON.0) within 512 clock cycles after time-out.

A watchdog timer reset only resets the MCU. The reset condition is maintained by hardware for two machine cycles.

System Behavior after a Reset Event

After a reset event, the system either re-initializes and re-starts code execution or just re-starts code execution, depending on the type of reset event.

Re-Initialization

Following some types of resets, as shown in Table 39, the Customizable Microcontroller starts or re-starts the device initialization process. The initialization process, described separately in the "*System Initialization*" section, typically lasts tens to hundreds of milliseconds, depending on the Customizable Microcontroller clock source and the operating mode. After the E5 is fully initialized, the user application program begins executing from microcontroller address 0000h.

After MCU Reset

After resetting the MCU, the MCU starts or re-starts executing the user application program from 8051 code location 0000h. There are no other implications of an MCU Reset. An MCU Reset event does not cause the device to re-initialize.

Microcontroller Reset State

Table 40. SFR Reset Conditions.

SFR Name	Reset Value
ACC	00000000b
B	00000000b
CKCON	00000001b
DPH	00000000b
DPH1	00000000b
DPL	00000000b
DPL1	00000000b
DPS	00000000b
EIE	11101111b
EIP	11101111b
IE	01000000b
IP	11000000b
P0	11111111b
P1	11111111b
P2	11111111b
P3	11111111b
PC	0000h
PCON	00110000b
PSW	00000000b
RCAP2H	00000000b
RCAP2L	00000000b
SADDR	00000000b
SADEN	00000000b
SBUF	11111111b
SCON	00000000b
SP	00000111b
T2CON	00000000b
T2MOD	11111110b
TA	11111111b
TCON	00000000b
TH0	00000000b
TH1	00000000b
TH2	00000000b
TL0	00000000b
TL1	00000000b
TL2	00000000b
TMOD	00000000b

Most of the SFRs and registers on the device are cleared by reset. The Program Counter is forced to 0000h and is held there as long as the reset condition is applied. The reset state however does not affect the on-chip RAM. The data in the RAM is preserved during the reset. However, the stack pointer is reset to 07h, and therefore the stack contents are lost. However, the RAM contents will be lost if the Vcc falls below approximately 2V, as this is the minimum voltage level required for the RAM to operate normally. Therefore, after a first-time power on reset, the RAM contents are indeterminate. During a power fail condition, if the power falls below 2V, the RAM contents are lost. After a power on/fail reset, POR = 1, the RAM contents should be assumed lost.

When reset, most SFRs are cleared, as shown in Table 40. Interrupts and Timers are disabled.

The Watchdog timer is disabled by a Power-On Reset (POR). The WDCON SFR bits are set or cleared in reset condition depending on the source of the reset, as shown in Table 41.

Table 41. Watchdog Timer Reset Values.

	External Reset	Watchdog Reset	Power-on Reset
WDCON	1x0x0xx0b	1x0x01x0b	11000000b

An 'x' indicates that the reset does not change the bit. The POR bit WDCON.6 is set only by the power on reset. The PFI bit WDCON.4 is set when the power fail condition occurs. However a power-on reset clears this bit if Vcc> Vpfw following the crystal start-up time. The WTRF bit WDCON.2 is set when the Watchdog timer causes a reset. A power-on reset also clears this bit. The EWT bit WDCON.1 is cleared by power-on resets. This disables the Watchdog timer resets. A watchdog or external reset does not affect the EWT bit.

Power Management

The Triscend E5 Customizable Microcontrollerhas several features that reduce power consumption, including two power saving modes, Power-Down and Idle. A power-on/power-fail reset guarantees proper start-up or restart operation.

Power Management

The Triscend E5 Customizable Microcontrollerhas a built-in power-on reset system. This ensures that if the power levels are below the VRST level, the device is forced into the reset state. When power is turned on—such as during a cold reset—the device is reset automatically and remains so as long as VCC is less that VRST. Similarly, when power falls below VRST, the device is automatically reset. This eliminates the need for any external reset circuitry.

Power Saving

The Triscend E5 has two power saving modes of operation, the Power Down mode and the Idle mode. These two modes are possible because the E5 is implemented in fully static CMOS logic. This permits the device to reduce the clock speed to DC. In the most aggressive power down state, where subsystems are properly shut down or blocked, the overall current can be reduced to a typical value of less than 50 µA.

Idle Mode

The Idle mode is primarily provided for compatibility with the original 8051 microcontroller. The device is placed in Idle mode by setting the IDL bit (PCON.0). The instruction that sets the idle bit is the last instruction executed before the device goes into Idle mode.

In the Idle mode, the MCU clock is disabled. However, the interrupt controllers, timers, watchdog timer, and serial port continue to operate. Idle mode freezes the MCU state including the Program Counter, the Stack Pointer, the Program Status Word, and the Accumulator. The other registers retain their contents.

There are two ways to exit Idle mode. Because the interrupt controller remains active, asserting any enabled interrupt wakes up the processor. This automatically clears the Idle bit, terminates the Idle mode, and executes the appropriate interrupt service routine (ISR). After the ISR, the program continues to execute from the instruction following the instruction that placed device into Idle mode.

An alternate method to exit the Idle mode is by forcing a reset condition. The device is reset by either applying a Low on the external RST- pin, a Power on/fail reset condition, a Watchdog timer reset, or asserting an application reset using the RSTC sideband signal. Some reset conditions cause the Triscend E5 device to re-initialize before re-starting code execution from 0000h. See the "*System Initialization*" section for more information.

After a reset condition, the program counter is reset to 0000h and all the SFRs are set to the reset condition. Because the clock is already running, there is no delay and execution starts immediately. In the Idle mode, the Watchdog timer continues to run, and if enabled, a time-out will cause a watchdog timer interrupt, which will wake up the device. The software must reset the Watchdog timer in order to preempt the reset, which will occur 512 clock periods after the time-out. When the MCU exits from the Idle mode with a reset, the instruction following the one which put the device into Idle mode is not executed.

Power-Down Mode

The device is placed into power-down mode by setting the PCON.1 bit. The instruction that sets this bit is the last instruction executed before the device goes into power-down mode. In power-down mode, all clocks are optionally stopped and internal functions optionally disabled. All activity is completely stopped and the power consumption is reduced to the lowest possible value.

The power down control logic optionally prevents the global clock signals from toggling when the MCU powers down. The PWDSEL defines the system behavior when the system enters the power down mode.

The MCU is powered down following the execution of an instruction that set the PD flag inside its PCON register. The MCU exits power down by a reset pulse or by responding to an external interrupt, as described later.

Prior to entering a power down state, the application program configures PWDSEL with settings that best fits the particular application.

Power Down Select

7	6	5	4	3	2	1	0
-	-	PIO	XTAL	OSC	GBUF	CSL BCLK	BCLK

Mnemonic: PWDSEL Address: FF62h

PIO, when set, signals all of the PIO pins when a power-down event occurs. Within each PIO, user-defined values configure the PIO to optionally block inputs by driving a Low value. In addition, the output driver is optionally three-stated, turning on the weak-follower feature. The individual PIO controls are loaded during device initialization.

XTAL, when set, blocks the external oscillator during power down mode. BCLK is frozen in its High state. Note that the memory interface unit (MIU) must be set up correctly to properly exit power down mode, as described later.

OSC, when set, shuts off the internal oscillator during power down mode.

GBUF, when set, signals the six global buffers when a power-down event occurs. Each global buffer has individual control bits that define its behavior during a power down or breakpoint event.

CSL_BCLK, when set, blocks the bus clock into the configurable system logic (CSL) matrix during power-down mode.

BCLK, when set, blocks the bus clock during power-down mode.

Power down mode is initiated by writing a “1” to the power down bit in the microcontroller’s PCON SFR. Depending on what portions of the device are selected for shut down or blocking, the effect is immediate. However, special attention is required when exiting power down mode after shutting off the crystal oscillator.

Power-On Reset Control

7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	PORCT

Mnemonic: PORCTRL Address: FF63h

PORCT, when set, disables the Power-On Reset (POR) circuitry during power-down mode. Clearing this bit keeps the POR circuitry active.

Power down mode is initiated by writing a “1” to the power down bit in the microcontroller’s PCON SFR.

Shutting Down the Crystal Oscillator

If the external crystal is selected to drive the internal system clock (BCLK), it can be shut down during power-down by setting the XTAL bit. However, if the crystal oscillator is shut down, the device automatically selects the internal ring oscillator as the default clock source upon exiting power down mode. Consequently, if the user program fetches instructions via the MIU when leaving power down—the usual case—potential timing problems could occur if the frequency of the internal ring oscillator is greater than the frequency of the external crystal. Before entering power down mode, the user application code must properly configure the MIU’s external memory timing with values that are more conservative. See Table 42 for more details.

Restarting the Crystal Oscillator after Exiting Power-Down Mode

If the crystal oscillator is disabled during power down, then the internal ring oscillator becomes the clock source for the bus clock, BCLK.

Before switching back to the crystal oscillator, the user program must wait for the crystal oscillator to stabilize. This can be accomplished either by a software timing loop or by using one of the MCU timers. Once the necessary time has elapsed, the user program must switch back to the external crystal and set the MIU read timing parameters to the pre-power-down values.

Table 42 summarizes the sequence for entering and exiting power down mode, when turning off the crystal oscillator during power-down mode. The method used depends on the crystal frequency. With the crystal shut off, the device wakes from power-down mode using the internal ring oscillator. The sequences in Table 42 represent the cases where the crystal is faster and slower than the internal ring oscillator.

The registers accessed in Table 42 are MIUCTRL and MISC. MIUCTRL controls the MIU timing and MISC contains the clock select bit. These registers are configuration register (CRU) bits. The CRU block size must be set to 4K bytes in order to enable access to these registers by clearing the DMAP3_CTL.0 bit.

Table 42. MIU Timing Considerations for Powering Down the Crystal Oscillator.

XTAL Freq.	Power Down Sequence	Exit power down sequence
< 20MHz	Add more cycles to MIU read timing in order to work with a 20MHz clock Set power down mode.	Wait desired time for crystal to stabilize Switch clock source to external crystal Optimize MIU read timing back to XTAL frequency

XTAL Freq.	Power Down Sequence	Exit power down sequence
		value.
> 20MHz	Set power down mode	Wait desired time for crystal to stabilize Optimize MIU read timing back to XTAL frequency value. Switch clock source to external crystal

The MIUCTRL register (external data default location FE33h) is described in the MIU section.

The MISC register is located at the external default address location FE81h. The clock select bit is in bit location 0. When cleared, the default value, the internal ring oscillator becomes the bus clock source. If set, then the external clock oscillator drives the bus clock.

Exiting Power-Down Mode

The Triscend E5 Customizable Microcontrollerexits power down mode either from a reset or by asserting either INTR0 or INTR1 sideband signal configured as a level-sensitive interrupt. An external reset causes the device to exit the Power down state. The Low on RST- pin terminates the Power Down mode, and restarts the clock. In the Power down mode, all clocks are stopped, so the Watchdog timer cannot be used to provide the reset to exit Power down mode. The Power on/fail reset, however, provides reset if power falls below the threshold level of VRST.

Three different events cause the device to exit power down mode.

1. Apply a reset pulse on the RST- pin.
2. Assert the application-reset signal, RSTC.
3. Interrupt the MCU using one of the external interrupt sideband signals, INTR0 or INTR1 and hold the signal for at least two machine cycles.

Reset Pulse on RST-

Applying a reset pulse on the external RST- pin causes the Triscend E5 to exit power-down mode. When asserted, the RST- pulse resets the entire system, including MCU and its PCON SFR. The internal ring oscillator becomes the default system clock source and the device begins the initialization process. The clock source selection for bus clock happens during initialization. At the end of initialization, the CPU begins executing the user's application program. This method of exiting power down mode might be too slow for some applications.

Asserting Application Reset (RSTC)

Another method to exit power down mode is to assert the application reset sideband signal, RSTC, from within the CSL matrix. The MCU restarts executing the application program from 0000h. If the crystal oscillator was blocked during power down, then default clock source is the internal oscillator. Before switching back to the crystal oscillator, the user program must wait for the crystal oscillator to stabilize.

External Interrupt via Sideband Signal

The E5 wakes from the Power Down mode by asserting the INTR0 or INTR1 interrupt sideband signals. The corresponding interrupt must be enabled, the global interrupt enable bit must be set and the INTR0 and INTR1 sideband signals configured in level-sensitive mode. If these conditions are met, then asserting the external pin re-enables the on-chip oscillator. The INTR0 or INTR1 sideband signal should be asserted long enough for the internal oscillator to start and stabilize. The power down mode is finally exited when the external interrupt is released. The device then executes an interrupt service routine (ISR) for the corresponding external interrupt. After the interrupt service routine is completed, the program execution returns to the instruction after the one that put the device into Power Down mode and continues from there.

Upon receiving an interrupt, the PCON SFR power down bit (PDBIT) is asynchronously reset to "0". In turn, it re-enables all the circuits that were selected for power down blocking. Again, if the crystal oscillator was blocked during power down, the interrupt routine must switch back from the internal ring oscillator to the crystal oscillator.

Glossary

CSoC - Configurable System-on-Chip. A configurable device containing a dedicated processor, configurable logic, on-chip RAM, and a dedicated internal bus.

CRU - Configuration Register Unit. Registers that define special functions on the Triscend E5 device.

CSL - Configurable System Logic. The peripheral configurable logic matrix providing user-programmable functions to the microcontroller and its peripherals.

CSI - Configurable System Interconnect. The on-chip bus that bridges the Configurable System Logic (CSL) matrix, the Accelerated 8051 micro-

controller, the dedicated peripherals, and the Memory Interface Unit (MIU).

DMA - Direct Memory Access. A controller that offloads memory and I/O transfers from the microcontroller.

JTAG - Joint Test Action Group. A general name given to the four-wire serial interface described in IEEE 1149.1.

MCU - Microcontroller Unit. The Accelerated 8051 microcontroller embedded within the Triscend E5 configurable system-on-chip.

MIU - Memory Interface Unit. Connects the microcontroller, its peripherals, and the internal system bus to external memory resources.

PIO - Programmable Input/Output.

SFR - Special Function Register. An 8051-based term for registers that control the microcontrollers peripheral functions.

Life Support Policy

Triscend's products are not authorized for use as critical components in life support devices or systems unless a specific written agreement pertaining to such intended use is executed between the manufacturer and the President of Triscend.

1. Life support devices or systems are devices which (a) are intended for surgical implant into the body or (b) support or sustain life and whose failure to perform, when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in significant injury to the user.
2. Critical component in any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device system, or affect its safety or effectiveness.

Revision List

Version 1.06 to 1.07

- Changed CSoC to Customizable Microcontroller
- Removed all references to the TE532
- Removed all references to BGA packages

Version 1.05 to 1.06

- Added text to **JTAG Reset** stating that the watchdog timer is not reset by a FORCE_BRST command.
- Corrected **208-pin PQFP package footprint drawing** to remove N.C. pins for TE502. The TE502 is not offered in this package.
- Added **Appendix A: E5 Instruction Timing**.
- Added note on side-effects of a MCU reset on the Watchdog Timer.

Version 1.04 to 1.05

- Added highlighting to show hyperlinks in document.
- Further clarified that WAITNEXT signal always asserts wait-states on the following bus cycle, independent of other logic. See **WAITNEXT (input to bus socket)**.
- Added power-down and power consumption information. See **Triscend E5 Power/Current Consumption Characteristics**.
- Added additional information on how to connect PMOD/A31, VSYS, SLAVE-, and RST- pins in typical applications. See **Pin Description**.
- Added production pin-to-pin timing information for all devices. See **Pin-to-Pin Guaranteed Timing Specifications**.
- Changed the term “8032 ‘Turbo’ microcontroller” to “Accelerated 8051 microcontroller” throughout the document. Many developers are not familiar with the 8032 device number. The 8032 is a ROM-less version of the 8052, which is a superset of the 8051. Generally changed all references from 8032 to 8051.
- Added a note about how long to assert sideband interrupt signals. See second note in **External Interrupts**.
- Corrected obsolete figure numbers throughout the **Triscend E5 Switching Characteristic Guidelines** section.
- Added Unique Technologies as an authorized Triscend distributor in North America. See

U.S. Distribution.

- Removed duplicated paragraph under Priority Level Structure.
- Corrected values for Bus Clock, Input Set-Up and Hold. The –25 and –40 speed grade values were swapped.
- Clarified text to show that T_{MDRZ} is a hold time and correctly defined it as a minimum specification value. See Memory Interface Unit (MIU) Timing Characteristic Guidelines, Stand-alone Operation (SLAVE- = High).
- Removed maximum specification on T_{MZOE} as it is not required. Data is captured by the bus clock. The OE- pin is also controlled by the same bus clock edge. The hold time is actually negative, relative to OE-, but is specified as zero. See Asynchronous Memory Interface Timing.

Version 1.03 to 1.04

- Added **Triscend E5 Switching Characteristic Guidelines** section.
- Removed 208-pin PQFP as a package option for the TE502. Affects Table 45, 208-pin PQFP Package Footprint, top view drawing, 208-pin PQFP Package Pinout Tables, and 208-pin PQFP Package Pins by **Type**.
- Added information on 484-ball BGA package.
- Added notes 2 through 7 to **Electrical and Timing Characteristics** page.
- Added TTL output current specifications to **Recommended Operating Conditions/DC Characteristics**.
- Added **128-pin and 208-pin PQFP Package Land Pattern Dimensions**.
- Provided additional details on output drive current. See **Table 14**.
- Added additional description to VSYS pin.

Version 1.02 to 1.03

- Corrected the address for the MISC register in text. The correct location is 0xfe81.
- Corrected a typo in the zip code for Triscend's mailing address.
- Updated hyper-links to Triscend sales office and international representatives.
- Updated the E-mail address for Triscend.

Version 1.00 to 1.02

- Updated specification for TE512 device to reflect SRAM size change from 16K bytes to 32K bytes. Change affects **Table 1**, **Table 20**, chart under **Ordering Information**, and the 128-pin and 208-pin package pinout drawings.
- Corrected polarity of input signals during power-down. Correct value is Low. Affects description of PIO bit in the **Power Down Select** register and **Input Options** for the PIO low-power mode.
- Added a register description for the 8051's B register SFR. See **B Register**.
- Added information about pin directionality during initialization to **Pin Description**.
- Added additional information on the VSYS and SLAVE- pins to **Pin Description**.
- Added information on using TH1 to control the baud rate for the serial port. See **Timer 1 MSB**.
- Added information on using RCAP2H and RCAP2L to control the baud rate for the serial port. See **Timer 2 Capture LSB** and **Timer 2 Capture MSB**.
- Added details to 5-volt tolerant PIO support. See **5-Volt Tolerant I/Os**.
- Updated PIO drawing to reflect proper location of inversion multiplexer on output and output-enable signal paths, as shown in **Figure 27**.
- Added additional detail on how the CSI bus resources operate. See **Configurable System Interconnect (CSI) Bus**.
- Added information on how a Selector can address an external peripheral connected to the MIU bus. See **Figure 13** and **External Memory Request Mode**.
- Added debugging support requirements. See **Debugging Support System Requirements**.
- Added phrase about using the internal feedback resistor when using the internal crystal oscillator amplified. See **System clock select (BCLK)**.
- Corrected the direction of the inverting buffer for the crystal oscillator amplified in **Figure 44**.
- The largest ball-grid array packaged changed to a 484-ball BGA package. Affects **Table 44**, **Table 45**, **Ordering Information**, and the previous BGA pinout diagrams and pinout tables were removed from this version.
- Renamed sideband signals INT0, INT1 to INTR0, INTR1 to be consistent with FastChip naming.

Pin Description

Table 43 describes the pins available on an E5 Customizable Microcontroller device. The directionality of each pin is described for parallel, serial, and slave mode initialization. O=output. I=input. I (pull-up)=input pin with soft pull-up during initialization—pin appears to float High.

Table 43. Pins available on an E5 Customizable Microcontroller and their function.

Pin Name	Pin Description	Parallel	Serial	Slave
A0/SCLK	Address bus bit 0 from the MIU in parallel mode. Serial clock output to serial PROM in serial mode. In slave mode, this pin has an internal high-impedance pull-up resistor that is active until initialization is complete.	O	O	I (pull-up)
A17/PIO A16/PIO A15/PIO A14/PIO A13/PIO A12/PIO A11/PIO A10/PIO A9/PIO A8/PIO A7/PIO A6/PIO A5/PIO A4/PIO A3/PIO A2/PIO A1/PIO	Address bus bit in parallel mode. In other modes, this pin may be used as a PIO. Until initialization is complete, this pin has an internal high-impedance pull-up resistor.	O	I (pull-up)	I (pull-up)
BCLK/XTAL	Bus clock input from an external clock source or the output from the internal crystal oscillator amplifier. Connect to one side of the external crystal or ceramic resonator or to an external clock source.	O	O	O
CE-	Active-Low chip-enable signal. An output when the Customizable Microcontroller accesses external memory during initialization in either parallel or serial modes. An input while in slave mode.	O	O	I
D0/SDIN	Data bus bit 0 in parallel mode. Serial data input (SDIN) in serial mode. In slave mode, this pin has an internal high-impedance pull-up resistor that is active until initialization is complete.	I	I	I (pull-up)
D7/PIO D6/PIO D5/PIO D4/PIO D3/PIO D2/PIO D1/PIO	Data bus bit in parallel mode. In other modes, this pin may be used as a PIO. Until initialization is complete, this pin has an internal high-impedance pull-up resistor.	I	I (pull-up)	I (pull-up)
GND	Ground connection for internal logic; connect to ground for I/O functions. All must be connected.	I	I	I

Pin Name	Pin Description	Parallel	Serial	Slave
GNDIO	Ground connection for I/O functions; connect to ground for internal logic. All must be connected.	I	I	I
N.C.	No connect. There is no function on this pin.	N.C.	N.C.	N.C.
OE-/SRST	Active-Low output-enable signal. An output when the Customizable Microcontrolleraccesses external memory during initialization in parallel or serial modes. An input while in slave mode.	O	O	I
PIO	General-purpose input, output, or bi-directional signal pin after initialization is complete. Before initialization is complete, these pins have internal high-impedance pull-up resistors that pull the signal pin to a High logic level.	I (pull-up)	I (pull-up)	I (pull-up)
PIO/A18 PIO/A19 PIO/A20 PIO/A21 PIO/A22 PIO/A23 PIO/A24 PIO/A25 PIO/A26 PIO/A27 PIO/A28 PIO/A29 PIO/A30	Typically, a general-purpose input, output, or bi-directional signal pin after initialization is complete. Before initialization is complete, this pin has an internal high-impedance pull-up resistor that pulls the signal pin to a High logic level. Also used in multi-chip mode to expand the upper address bits for external accesses.	I (pull-up)	I (pull-up)	I (pull-up)
PIO/GBUF5 PIO/GBUF4 PIO/GBUF3 PIO/GBUF2 PIO/GBUF1 PIO/GBUF0	Global buffer input. Typically, a general-purpose input, output, or bi-directional signal pin after initialization is complete. Before initialization is complete, this pin has an internal high-impedance pull-up resistor that pull the signal pin to a High	I (pull-up)	I (pull-up)	I (pull-up)
PIO/XDONE	Typically, a general-purpose input, output, or bi-directional signal pin after initialization is complete. Before initialization is complete, this pin has an internal high-impedance pull-up resistor that pulls the signal pin to a High logic level. If SLAVE-=Low, then used in multi-chip mode to for inter-device transfer handshaking.	I (pull-up)	I (pull-up)	I (pull-up)
PMOD/PIO/ A31	Tied Low during initialization to indicate serial mode. Left floating or pulled High during initialization for parallel mode. Also a potential general-purpose input, output, or bi-directional signal pin after initialization is complete but with the restrictions described above. Before initialization is complete, this pin has an internal high-impedance pull-up resistor that pulls the signal pin to a High logic level. Also used in multi-chip mode to expand the upper address bits for external accesses.	Optionally pull High using a pull-up resistor I (pull-up)	Pull Low using a pull-down resistor I (pull-up)	I (pull-up)
RST-	Device reset input. Active Low. Pull High if unused.	I	I	I
SLAVE-	Slave mode input. Active Low. Tie High for most applications indicating that the Customizable Microcontroller device initializes itself from an external PROM, either parallel or serial. Tie Low if another processor initializes the Customizable Microcontroller device via the MIU interface. This pin must be connected. Do not allow it to float.	Tie High (I)	Tie High (I)	Tie Low (I)
TCK	JTAG Test Clock input. Tie High if unused.	I	I	I
TDI	JTAG Test Data Input. Tie High if unused.	I	I	I
TDO	JTAG Test Data Output.	O	O	O

Pin Name	Pin Description	Parallel	Serial	Slave
TMS	JTAG Test Mode Select input. Tie High if unused.	I	I	I

Pin Name	Pin Description	Parallel	Serial	Slave
VCC	Supply voltage for internal logic functions, separate from I/O. Connect to a +3.3 volt supply. All must be connected and each must be decoupled with a 0.01 to 0.1 µF capacitor to ground.	I	I	I
VCCIO	Supply voltage for I/O functions, separate from internal logic. Connect to a +3.3 volt supply. All must be connected and each must be decoupled with a 0.01 to 0.1 µF capacitor to ground.	I	I	I
VSYS	External 'system voltage-good' indicator input. Indicates when external devices have sufficient operating voltage. In most applications, connect VSYS to VCCIO supply source or to a "Power Good" output from the system power supply. If VSYS is High during initialization and no valid configuration is found, the Customizable Microcontroller device automatically powers down to conserve power. If VSYS is Low, the Customizable Microcontroller device attempts to initialize itself until it finds valid initialization data. This pin must be connected. Do not allow it to float.	Always connect (I)	Always connect (I)	Always connect (I)
WE-/SEN-	Active-Low write-enable signal typically used to program external FLASH-based devices. An output when the E5 accesses external memory during initialization in parallel or serial modes. An input while in slave mode.	O	O	I
XTALIN	The input from the external crystal into the internal crystal oscillator amplifier. Connect to one side of the external crystal or ceramic resonator. Leave unconnected if not using an external crystal or resonator.	I	I	I

Pinout Diagrams and Tables

Available Packages and Package Codes

Each Triscend E5 family member is available in different packages. Each ordering code contains a letter defining the package style, as shown in Table 44.

Table 44. Package Codes.

Package Code	Package Type
L	128-pin PQFP
Q	208-pin PQFP

Footprint-Compatibility

Each Triscend E5 Customizable Microcontroller is designed to be footprint compatibly with other E5 family members available in the same package, providing additional design and manufacturing flexibility. A designer can easily select a larger or more cost-effective E5 device that fits a pre-existing printed circuit board (PCB) design.

Though the Triscend E5 and A7 families are available in similar package options, the E5 and A7 families are ***not*** pin compatible with one another.

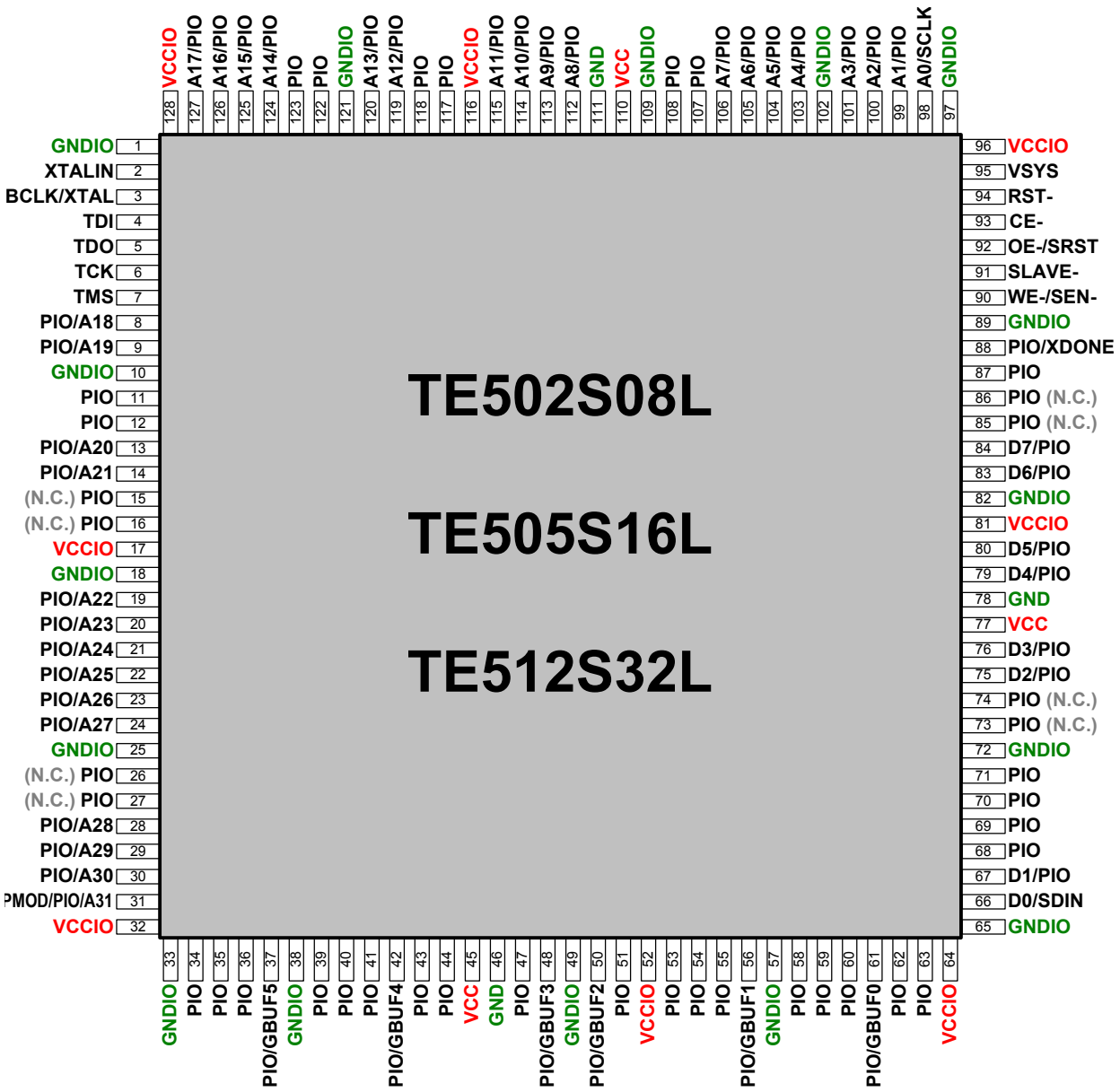
Available PIOs by Package

The number of user-configurable PIO pins depends on the base device type and the package in which it is packaged. Table 45 shows the available PIOs by package style. Two values are indicated for each device type, one for parallel mode and one for serial mode configuration. Both cases assume that the upper address bits, A[31:18] are used as PIOs, which is the default case.

Table 45. Available PIOs by Package

Device	Mode	LQ128	PQ208
E502	Parallel	52	
	Serial	76	
E505	Parallel	60	100
	Serial	84	124
E512	Parallel	60	126
	Serial	84	150
E520	Parallel		126
	Serial		150

128-pin Thin PQFP Footprint, top view



(N.C.) indicates a pin that is not connected on the TE502S08L device in this package.

128-pin Thin PQFP Package Pinout Tables

Shaded cells represent a pin that is not connected on the TE502 device but is a PIO pin for all other devices.

Pin	TE502	TE505 TE512
1	GNDIO	GNDIO
2	XTALIN	XTALIN
3	BCLK/XTAL	BCLK/XTAL
4	TDI	TDI
5	TDO	TDO
6	TCK	TCK
7	TMS	TMS
8	PIO/A18	PIO/A18
9	PIO/A19	PIO/A19
10	GNDIO	GNDIO
11	PIO	PIO
12	PIO	PIO
13	PIO/A20	PIO/A20
14	PIO/A21	PIO/A21
15	N.C.	PIO
16	N.C.	PIO
17	VCCIO	VCCIO
18	GNDIO	GNDIO
19	PIO/A22	PIO/A22
20	PIO/A23	PIO/A23
21	PIO/A24	PIO/A24
22	PIO/A25	PIO/A25
23	PIO/A26	PIO/A26
24	PIO/A27	PIO/A27
25	GNDIO	GNDIO
26	N.C.	PIO
27	N.C.	PIO
28	PIO/A28	PIO/A28
29	PIO/A29	PIO/A29
30	PIO/A30	PIO/A30
31	PMOD/PIO/ A31	PMOD/PIO/ A31
32	VCCIO	VCCIO
33	GNDIO	GNDIO
34	PIO	PIO
35	PIO	PIO
36	PIO	PIO
37	PIO/GBUF5	PIO/GBUF5
38	GNDIO	GNDIO
39	PIO	PIO
40	PIO	PIO
41	PIO	PIO
42	PIO/GBUF4	PIO/GBUF4
43	PIO	PIO
44	PIO	PIO
45	VCC	VCC
46	GND	GND
47	PIO	PIO
48	PIO/GBUF3	PIO/GBUF3
49	GNDIO	GNDIO
50	PIO/GBUF2	PIO/GBUF2
51	PIO	PIO
52	VCCIO	VCCIO
53	PIO	PIO
54	PIO	PIO
55	PIO	PIO
56	PIO/GBUF1	PIO/GBUF1
57	GNDIO	GNDIO
58	PIO	PIO
59	PIO	PIO
60	PIO	PIO
61	PIO/GBUF0	PIO/GBUF0
62	PIO	PIO
63	PIO	PIO
64	VCCIO	VCCIO
65	GNDIO	GNDIO
66	D0/SDIN	D0/SDIN
67	D1/PIO	D1/PIO
68	PIO	PIO
69	PIO	PIO
70	PIO	PIO
71	PIO	PIO
72	GNDIO	GNDIO
73	N.C.	PIO
74	N.C.	PIO
75	D2/PIO	D2/PIO
76	D3/PIO	D3/PIO
77	VCC	VCC
78	GND	GND
79	D4/PIO	D4/PIO
80	D5/PIO	D5/PIO
81	VCCIO	VCCIO
82	GNDIO	GNDIO
83	D6/PIO	D6/PIO
84	D7/PIO	D7/PIO
85	N.C.	PIO
86	N.C.	PIO
87	PIO	PIO
88	PIO/XDONE	PIO/XDONE
89	GNDIO	GNDIO
90	WE-/SEN-	WE-/SEN-
91	SLAVE-	SLAVE-
92	OE-/SRST	OE-/SRST
93	CE-	CE-
94	RST-	RST-
95	VSYS	VSYS
96	VCCIO	VCCIO
97	GNDIO	GNDIO
98	A0/SCLK	A0/SCLK
99	A1/PIO	A1/PIO
100	A2/PIO	A2/PIO
101	A3/PIO	A3/PIO
102	GNDIO	GNDIO
103	A4/PIO	A4/PIO
104	A5/PIO	A5/PIO
105	A6/PIO	A6/PIO
106	A7/PIO	A7/PIO
107	PIO	PIO
108	PIO	PIO
109	GNDIO	GNDIO
110	VCC	VCC
111	GND	GND
112	A8/PIO	A8/PIO
113	A9/PIO	A9/PIO
114	A10/PIO	A10/PIO
115	A11/PIO	A11/PIO
116	VCCIO	VCCIO
117	PIO	PIO
118	PIO	PIO
119	A12/PIO	A12/PIO
120	A13/PIO	A13/PIO
121	GNDIO	GNDIO
122	PIO	PIO
123	PIO	PIO
124	A14/PIO	A14/PIO
125	A15/PIO	A15/PIO
126	A16/PIO	A16/PIO
127	A17/PIO	A17/PIO
128	VCCIO	VCCIO

128-pin Thin PQFP Package Pins by Type

Parallel Mode

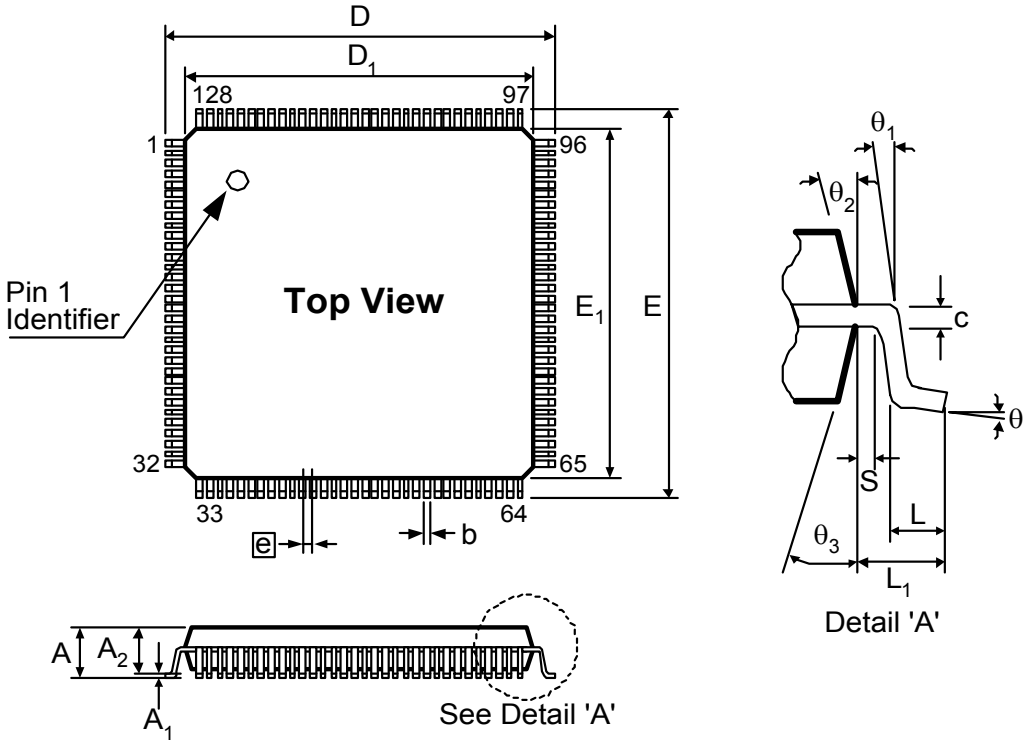
Type	TE502	TE505 TE512
PIO*	52	60
VCC	3	3
VCCIO	8	8
GND	3	3
GNDIO	16	16
D[7:0]	8	8
A[17:0]	18	18
JTAG	4	4
N.C.	8	0
Others	8	8

Serial Mode

Type	TE502	TE505 TE512
PIO*	76	84
VCC	3	3
VCCIO	8	8
GND	3	3
GNDIO	16	16
SDIN	1	1
SCLK	1	1
JTAG	4	4
N.C.	8	0
Others	8	8

PIO* includes pure PIO pins and those with alternate functions

128-pin Thin PQFP Package Mechanical Drawing

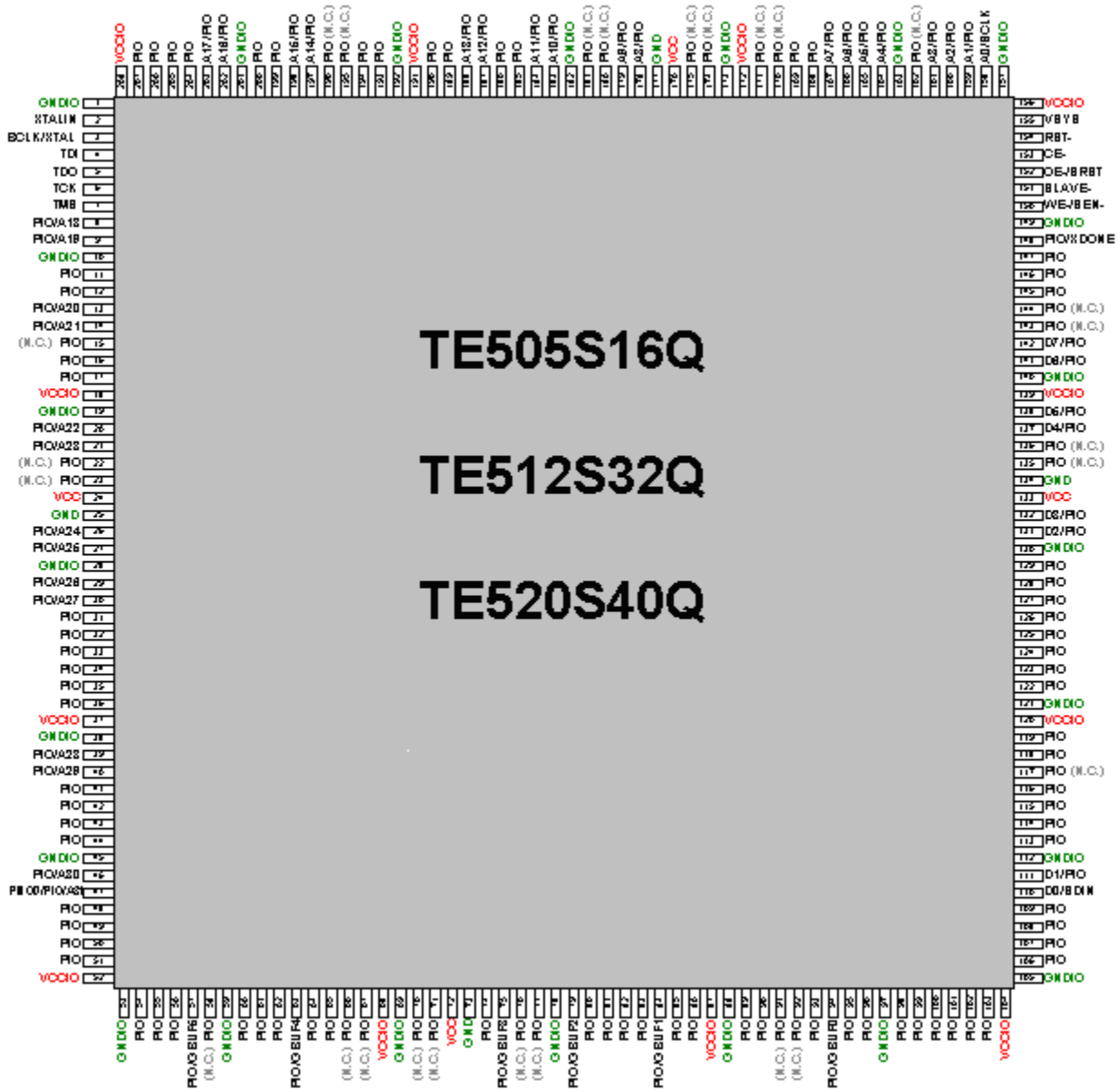


Symbol	Millimeters		
	Min.	Nom.	Max.
A	—	—	1.60
A_1	0.05	—	—
A_2	1.35	1.40	1.45
b	0.13	0.18	0.23
c	0.09	—	0.20
D	15.85	16.00	16.15
D_1	13.90	14.00	14.10
E	15.85	16.00	16.15
E_1	13.90	14.00	14.10
e	0.40 BSC		
L	0.45	0.60	0.75
L_1	1.00 REF		
S	0.20	—	—
θ	0°	3.5°	7°
θ_1	0°	—	—
θ_2	12° TYP		
θ_3	12° TYP		

1. All dimensions and tolerances conform to ASME Y14.5M-1994.
2. All dimensions are in millimeters.
3. Dimensions D1 and E1 do not include mold protrusion. Allowable mold protrusion shall not exceed 0.25 mm per side.
4. Package top dimensions may be smaller than bottom dimensions.

JEDEC Reference Drawing:	MS-026C
JEDEC File Location:	http://www.jedec.org/DOWNLOAD/pub95/MS-026c.pdf
Land Pattern Dimensions:	See 128-pin and 208-pin PQFP Package Land Pattern Dimensions
Assembly Vendor Drawing:	SPIL, D128-SW1-A

208-pin PQFP Package Footprint, top view



(N.C.) indicates a pin that is not connected on the TE505S16Q device in this package.

208-pin PQFP Package Pinout Tables

Shaded cells represent a pin that is not connected on the TE505 device but is connected on all other devices.

Pin	TE505	TE512 TE520
1	GNDIO	GNDIO
2	XTALIN	XTALIN
3	BCLK/XTAL	BCLK/XTAL
4	TDI	TDI
5	TDO	TDO
6	TCK	TCK
7	TMS	TMS
8	PIO/A18	PIO/A18
9	PIO/A19	PIO/A19
10	GNDIO	GNDIO
11	PIO	PIO
12	PIO	PIO
13	PIO/A20	PIO/A20
14	PIO/A21	PIO/A21
15	N.C.	PIO
16	PIO	PIO
17	PIO	PIO
18	VCCIO	VCCIO
19	GNDIO	GNDIO
20	PIO/A22	PIO/A22
21	PIO/A23	PIO/A23
22	N.C.	PIO
23	N.C.	PIO
24	VCC	VCC
25	GND	GND
26	PIO/A24	PIO/A24
27	PIO/A25	PIO/A25
28	GNDIO	GNDIO
29	PIO/A26	PIO/A26
30	PIO/A27	PIO/A27
31	PIO	PIO
32	PIO	PIO
33	PIO	PIO
34	PIO	PIO
35	PIO	PIO
36	PIO	PIO
37	VCCIO	VCCIO
38	GNDIO	GNDIO
39	PIO/A28	PIO/A28
40	PIO/A29	PIO/A29
41	PIO	PIO
42	PIO	PIO
43	PIO	PIO
44	PIO	PIO
45	GNDIO	GNDIO
46	PIO/A30	PIO/A30
47	PMOD/PIO/A31	PMOD/PIO/A31
48	PIO	PIO
49	PIO	PIO
50	PIO	PIO
51	PIO	PIO
52	VCCIO	VCCIO
53	GNDIO	GNDIO
54	PIO	PIO
55	PIO	PIO
56	PIO	PIO
57	PIO/GBUF5	PIO/GBUF5
58	N.C.	PIO
59	GNDIO	GNDIO
60	PIO	PIO
61	PIO	PIO
62	PIO	PIO
63	PIO/GBUF4	PIO/GBUF4
64	PIO	PIO
65	PIO	PIO
66	N.C.	PIO
67	N.C.	PIO
68	VCCIO	VCCIO
69	GNDIO	GNDIO
70	N.C.	PIO
71	N.C.	PIO
72	VCC	VCC
73	GND	GND
74	PIO	PIO
75	PIO/GBUF3	PIO/GBUF3
76	N.C.	PIO
77	N.C.	PIO
78	GNDIO	GNDIO
79	PIO/GBUF2	PIO/GBUF2
80	PIO	PIO
81	PIO	PIO
82	PIO	PIO
83	PIO	PIO
84	PIO/GBUF1	PIO/GBUF1
85	PIO	PIO
86	PIO	PIO
87	VCCIO	VCCIO
88	GNDIO	GNDIO
89	PIO	PIO
90	PIO	PIO
91	N.C.	PIO
92	N.C.	PIO
93	PIO	PIO
94	PIO/GBUF0	PIO/GBUF0
95	PIO	PIO
96	PIO	PIO
97	GNDIO	GNDIO
98	PIO	PIO
99	PIO	PIO
100	PIO	PIO
101	PIO	PIO
102	PIO	PIO
103	PIO	PIO
104	VCCIO	VCCIO
105	GNDIO	GNDIO
106	PIO	PIO
107	PIO	PIO
108	PIO	PIO
109	PIO	PIO
110	D0/SDIN	D0/SDIN
111	D1/PIO	D1/PIO
112	GNDIO	GNDIO
113	PIO	PIO
114	PIO	PIO
115	PIO	PIO
116	PIO	PIO
117	N.C.	PIO
118	PIO	PIO
119	PIO	PIO
120	VCCIO	VCCIO
121	GNDIO	GNDIO
122	PIO	PIO
123	PIO	PIO
124	PIO	PIO
125	PIO	PIO
126	PIO	PIO
127	PIO	PIO
128	PIO	PIO
129	PIO	PIO
130	GNDIO	GNDIO
131	D2/PIO	D2/PIO
132	D3/PIO	D3/PIO
133	VCC	VCC
134	GND	GND
135	N.C.	PIO
136	N.C.	PIO
137	D4/PIO	D4/PIO
138	D5/PIO	D5/PIO
139	VCCIO	VCCIO
140	GNDIO	GNDIO
141	D6/PIO	D6/PIO
142	D7/PIO	D7/PIO
143	N.C.	PIO
144	N.C.	PIO
145	PIO	PIO
146	PIO	PIO
147	PIO	PIO
148	PIO/ XDONE	PIO/ XDONE
149	GNDIO	GNDIO
150	WE-/SEN-	WE-/SEN-
151	SLAVE-	SLAVE-
152	OE-/SRST	OE-/SRST
153	CE-	CE-
154	RST-	RST-
155	VSYS	VSYS
156	VCCIO	VCCIO
157	GNDIO	GNDIO

Pin	TE505	TE512 TE520
158	A0/SCLK	A0/SCLK
159	A1/PIO	A1/PIO
160	A2/PIO	A2/PIO
161	A3/PIO	A3/PIO
162	N.C.	PIO
163	GNDIO	GNDIO
164	A4/PIO	A4/PIO
165	A5/PIO	A5/PIO
166	A6/PIO	A6/PIO
167	A7/PIO	A7/PIO
168	PIO	PIO
169	PIO	PIO
170	N.C.	PIO
171	N.C.	PIO
172	VCCIO	VCCIO
173	GNDIO	GNDIO
174	N.C.	PIO
175	N.C.	PIO
176	VCC	VCC
177	GND	GND
178	A8/PIO	A8/PIO
179	A9/PIO	A9/PIO
180	N.C.	PIO
181	N.C.	PIO
182	GNDIO	GNDIO
183	A10/PIO	A10/PIO
184	A11/PIO	A11/PIO
185	PIO	PIO
186	PIO	PIO
187	A12/PIO	A12/PIO
188	A13/PIO	A13/PIO
189	PIO	PIO
190	PIO	PIO
191	VCCIO	VCCIO
192	GNDIO	GNDIO
193	PIO	PIO
194	PIO	PIO
195	N.C.	PIO
196	N.C.	PIO
197	A14/PIO	A14/PIO
198	A15/PIO	A15/PIO
199	PIO	PIO
200	PIO	PIO
201	GNDIO	GNDIO
202	A16/PIO	A16/PIO
203	A17/PIO	A17/PIO
204	PIO	PIO
205	PIO	PIO
206	PIO	PIO
207	PIO	PIO
208	VCCIO	VCCIO

208-pin PQFP Package Pins by Type

Parallel Mode

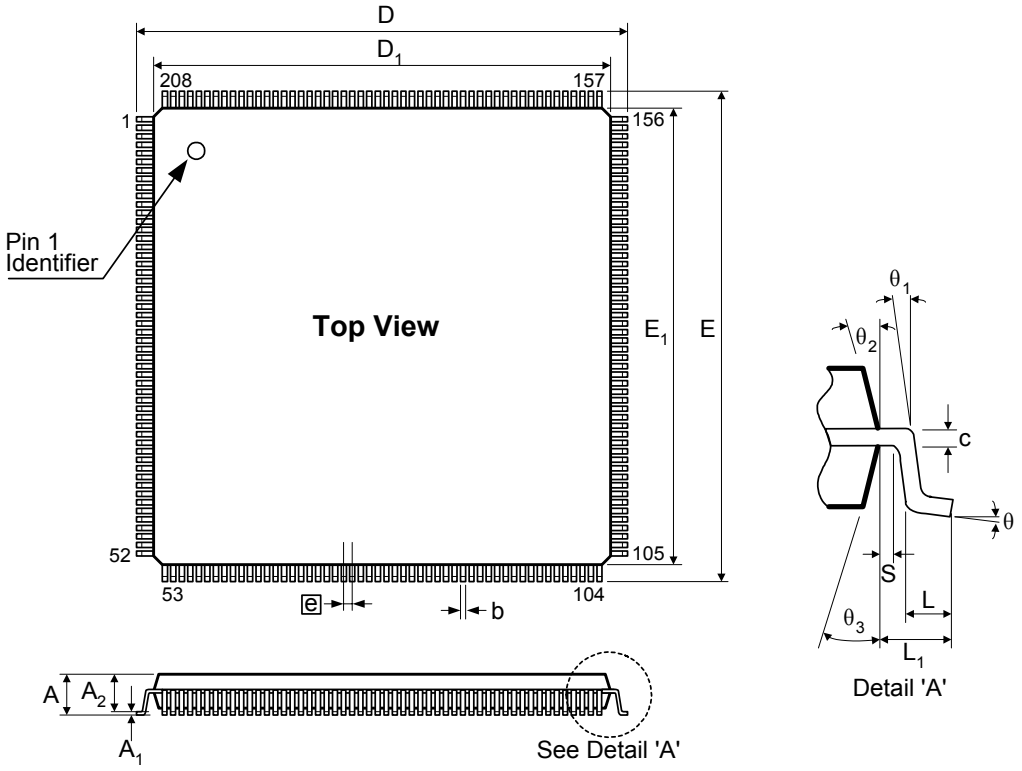
Type	TE505	TE512 TE520
PIO*	100	126
VCC	4	4
VCCIO	12	12
GND	4	4
GNDIO	24	24
D[7:0]	8	8
A[17:0]	18	18
JTAG	4	4
N.C.	26	0
Others	8	8

Serial Mode

Type	TE505	TE512 TE520
PIO*	124	150
VCC	4	4
VCCIO	12	12
GND	4	4
GNDIO	24	24
SDIN	1	1
SCLK	1	1
JTAG	4	4
N.C.	26	0
Others	8	8

PIO* includes pure PIO pins and those with alternate functions.

208-pin PQFP Package Mechanical Drawing

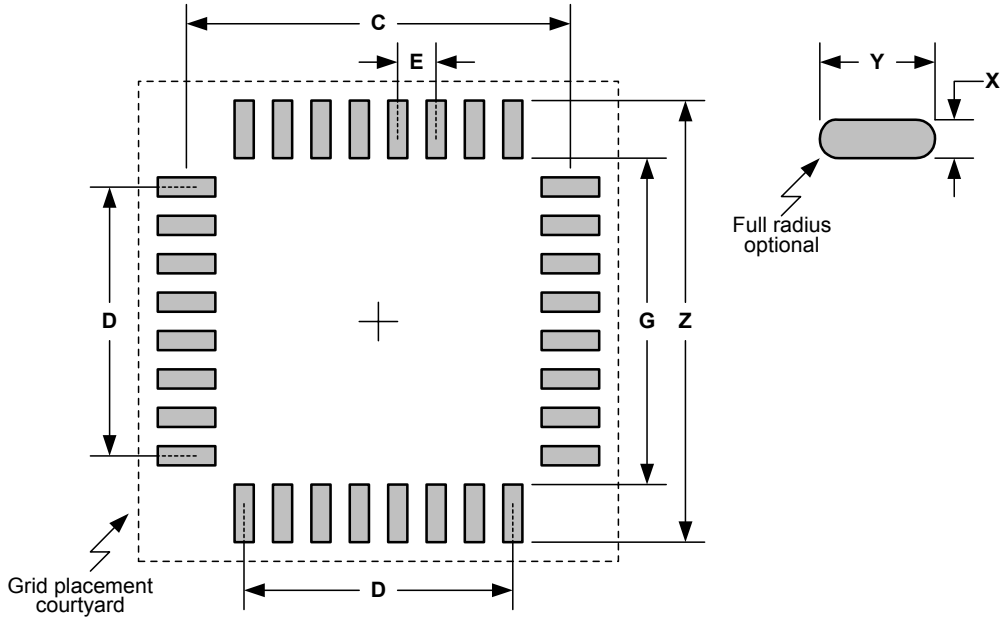


Symbol	Millimeters		
	Min.	Nom.	Max.
A	3.92	—	4.07
A_1	0.25	—	—
A_2	3.15	3.23	3.30
b	0.18	—	0.28
c	0.13	—	0.23
D	30.35	30.60	30.85
D_1	27.90	28.00	28.10
E	30.35	30.60	30.85
E_1	27.90	28.00	28.10
e	0.50 BSC		
L	0.35	0.50	0.65
L_1	1.30 REF		
S	0.20	—	—
θ	0°	—	7°
θ_1	0°	—	—
θ_2	12° TYP		
θ_3	12° TYP		

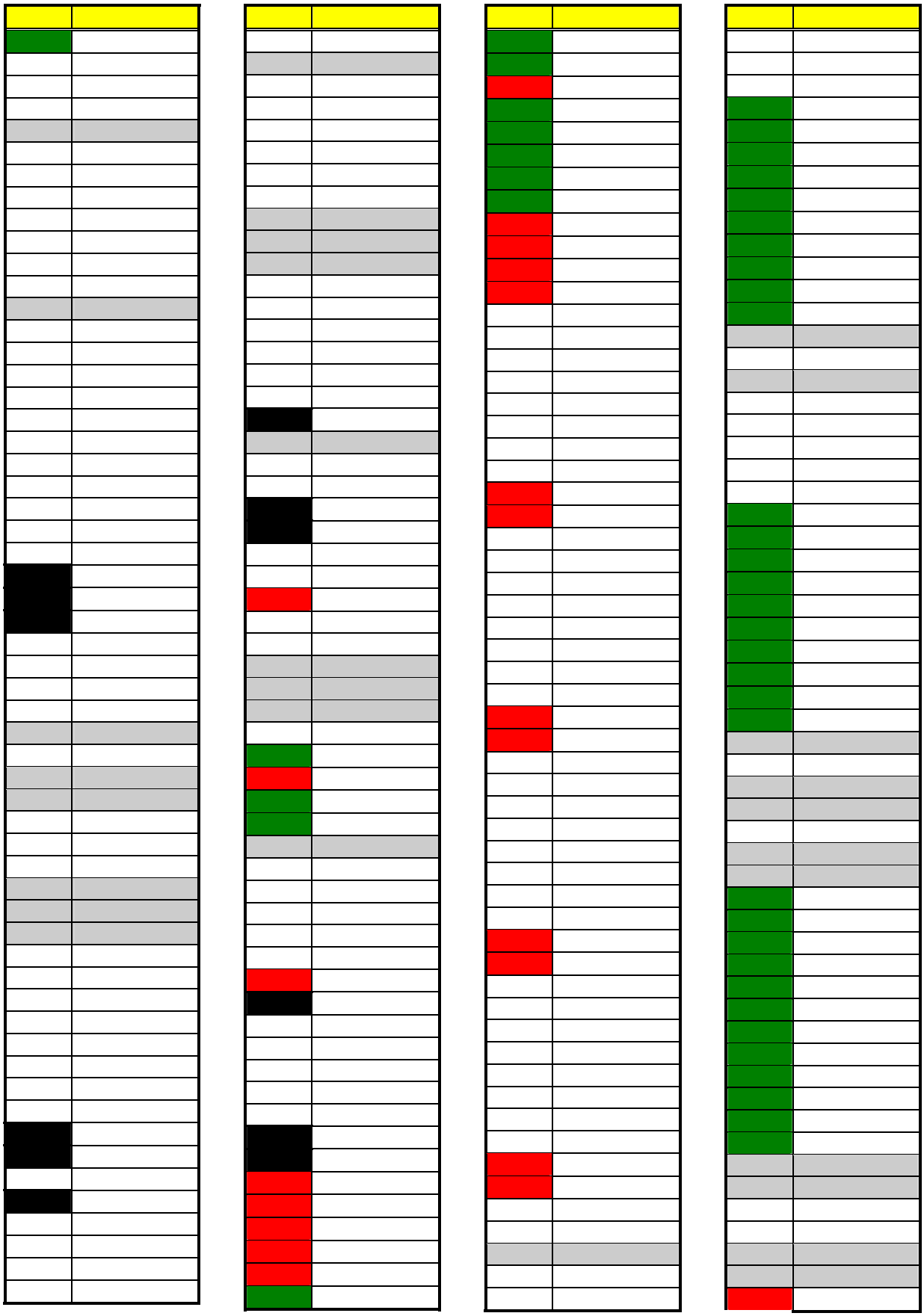
1. All dimensions and tolerances conform to ASME Y14.5M-1994.
2. All dimensions are in millimeters.
3. Dimensions D1 and E1 do not include mold protrusion. Allowable mold protrusion shall not exceed 0.25 mm per side.
4. Package top dimensions may be smaller than bottom dimensions.

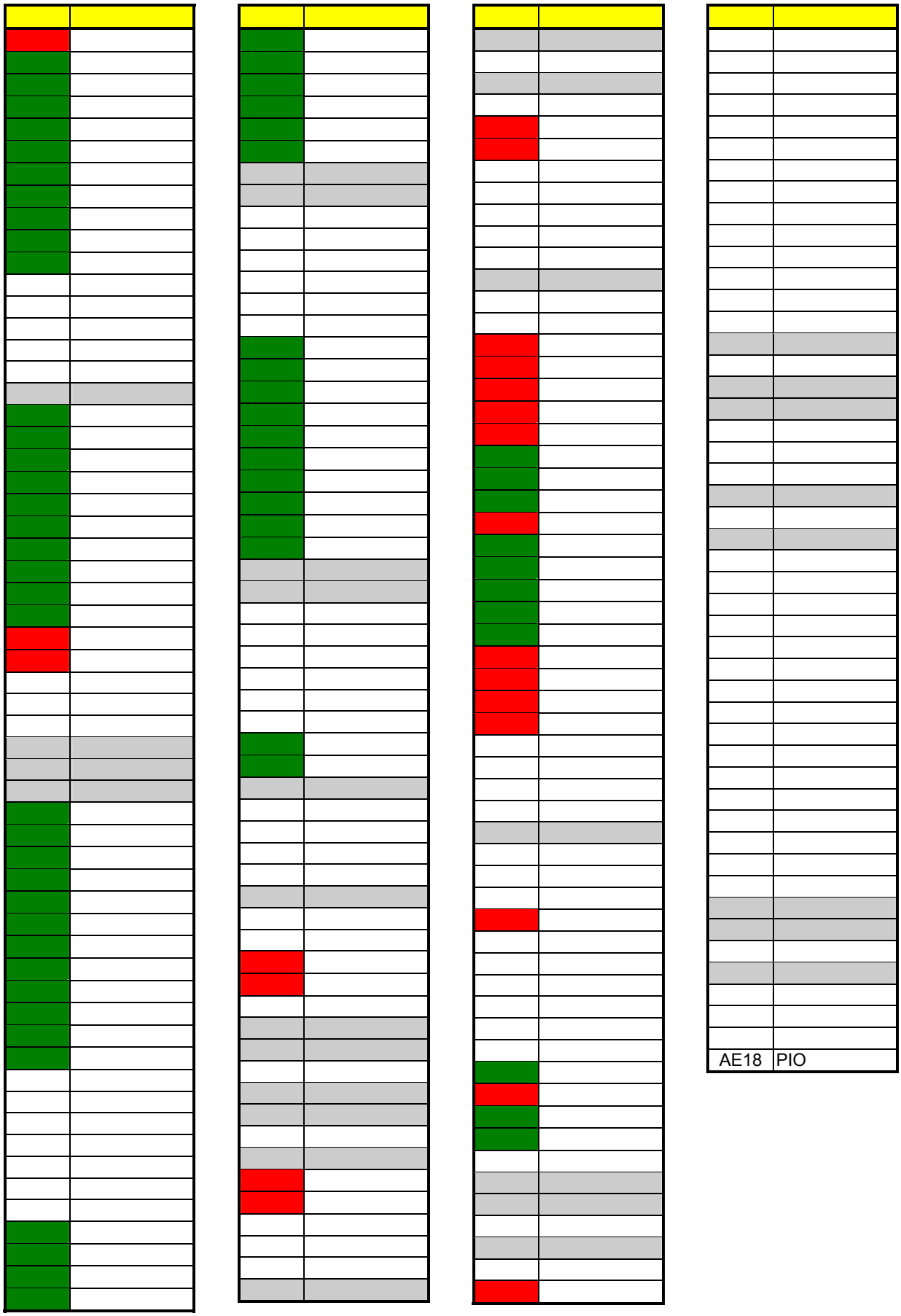
JEDEC Reference Drawing:	MS-029A-FA-1
JEDEC File Location:	http://www.jedec.org/DOWNLOAD/pub95/ms-029a.pdf
Land Pattern Dimensions:	See 128-pin and 208-pin PQFP Package Land Pattern Dimensions
Assembly Vendor Drawing:	SPIL, Q208-SW2-C

128-pin and 208-pin PQFP Package Land Pattern Dimensions



Symbol	128-pin LQFP (Code=L)			208-pin PQFP (Code=Q)			Units
	Min	Ref	Max	Min	Ref	Max	
Z	—	—	16.8	—	—	30.8	mm
G	13.6	—	—	27.6	—	—	
X	—	—	0.25	—	—	0.3	
Y	—	1.6	—	—	1.6	—	
C	—	15.2	—	—	29.2	—	
D	—	12.4	—	—	25.5	—	
E	—	0.4	—	—	0.5	—	




AE18
PIO

Electrical and Timing Characteristics

Absolute Maximum Ratings

Symbol	Parameter	Min	Max	Units
V_{CC}	Supply voltage relative to GND	-0.5	4.0	V
V_{IN}	Input voltage relative to GND [1]	-0.5	6.5	V
V_{TS}	Voltage applied to three-state output [1]	-0.5	5.5	V
T_{STG}	Storage temperature (ambient)	-60	+150	° C
T_{SOL}	Maximum soldering temperature (10 s at 1/16 in. = 1.5 mm)		+260	° C
T_J	Junction temperature, plastic packages		+125	° C

Recommended Operating Conditions/DC Characteristics

Symbol	Parameter	Min	Max	Units
V_{CC}	Supply voltage relative to GND	3.0	3.6	V
T_J	Operating junction temperature, commercial [3]	0	+85	° C
	Operating junction temperature, industrial [3]	-40	+100	° C
V_{CCR}	Longest supply voltage rise time from 1V to 3V [4]		200	ms
V_{IL}	Input Low voltage	-0.5	30% V_{CC}	V
V_{IH}	Input High voltage	50% V_{CC}	5.5	V
V_S	Schmitt Hysteresis, hysteresis mode	5% VCC	20% VCC	V
V_{ESD}	Electro-static discharge protection, human body model	2 000		V
T_{IN}	Input signal transition time		250	ns
V_{OL}	Output Low voltage, I_{OL} = -12 mA, V_{CC} min (TTL) [5]		0.4	V
	Output Low voltage, I_{OL} = -1.5 mA, V_{CC} min (LVCMOS)		10% VCC	V
V_{OH}	Output High voltage, I_{OH} = 6 mA, V_{CC} min (TTL) [5]	2.4		V
	Output High voltage, I_{OH} = 0.5 mA, V_{CC} min (LVCMOS)	90% VCC		V
V_{DR}	Data retention supply voltage (below which initialization data may be lost)	2.5		V
I_{OL}	Output Low current, output in highest drive strength mode (TTL), V_{OL} max, V_{CC} min		-12.0	mA
	Output Low current, output in lowest current drive mode (TTL), V_{OL} max, V_{CC} min		-4.0	mA
	Output Low current (LVCMOS), V_{OL} max, V_{CC} min		-1.5	mA
I_{OH}	Output High current, output in highest drive strength mode (TTL), V_{OH} min, V_{CC} min		+6.0	mA
	Output High current, output in lowest drive strength mode (TTL), V_{OH} min, V_{CC} min		+2.0	mA
	Output High current (LVCMOS), V_{OH} min, V_{CC} min		+0.5	mA
I_{IL}	Input leakage current	-10	+10	µA
C_{IO}	Pin capacitance [7]		10	pF
L_{IO}	Pin inductance [7]		20	nH

Note 1: Maximum DC overshoot above VCC or undershoot below GND must be limited to either 0.5 V or 10 mA, whichever is easier to achieve. During transitions, device pins may undershoot to -2.0 V or overshoot to +5.0 V, provided this condition lasts less than 20 ns and with less than 100 mA forcing current.

Note 2: Stresses beyond those listed under Absolute Maximum Ratings may cause permanent device damage. The values listed are stress ratings only. Functional operation of the device at these or any conditions exceeding those listed under Recommended Operating Conditions is not implied. Exposure to Absolute Maximum Ratings conditions for extended periods may affect device reliability.

Note 3: Typical ambient operating conditions are between 0 to +70° C for commercial devices and -40 to +85° C for industrial devices, depending on the application's power consumption, package style, and airflow.

Note 4: Ramp rate can be extended by asserting RST- until V_{CC} reaches the minimum specified value.

Note 5: Output in highest drive strength mode.

Note 6: Continuous static loads must fall within the I_{OH}, I_{OL} limits in this section.

Note 7: Capacitance and inductance is sample-tested only.

Triscend E5 Switching Characteristic Guidelines

All Triscend devices are 100% functionally tested. These parameters are modeled after the testing methods described by MIL-M-38510/605. The values listed below are ***representative, guideline values*** extracted from measured internal test patterns. Actual values may depend on application-specific use. ***The FastChip development system reports specific, worst-case guaranteed values in the Timing Analysis section of the project report.***

All timing values shown assume worst-case operating conditions, including process technology, power supply voltage, and junction temperature.

Product Status Definitions

These specifications include a status designation as defined below.

Preview:	Initial estimates based on simulation or extrapolated data from other speed grades, devices, families, or process technologies. These values are subject to change without notice. These values are estimates and should not be used for production.
Preliminary:	Based on preliminary or partial device characterization. Though further changes are not expected, these values are subject to change without notice. These values are safe for producing prototypes but should not be used for high-volume production.
Final or Unmarked:	Specifications are final. These specifications may be used for volume production. Values cannot change without notice.
Guideline:	A worst-case value or a range of worst-case values based on example applications under a variety of conditions. Actual worst-case values are reported for the specific use within an application by the FastChip development system in the Timing Analysis section of the project report. The value reported by FastChip may not match the values shown herein. The value reported by FastChip supercedes any value shown below.

General E5 Timing Characteristics

			Final			
		Speed Grade	**All**	**-25**	**-40**	
Description	**Symbol**	**Device**	**Min**	**Max**	**Max**	**Units**
Bus Clock frequency	F_{BCLK}	All	0	25.0	40.0	MHz
Bus Clock high time	T_{BCH}	All	12.0			ns
Bs Clock low time	T_{BCL}	All	12.0			ns
Bus Clock rise time	T_{BCRT}	All		5.0	5.0	ns
Bus Clock fall time	T_{BCFT}	All		5.0	5.0	ns
32 kHz crystal start-up time	T_{32STU}	All		200.0	200.0	ms
2–24 MHz Crystal start-up time	T_{XSTU}	All		5.0	5.0	ms
Internal ring oscillator frequency	F_{ROSC}	All	5.0	20.0	20.0	MHz
RST- pulse width	T_{RSTW}	All	20.0			ns
Power-on reset delay after power valid	T_{RSTW}	All	13	52	52	µs
			Final			

JTAG Interface Timing Characteristics

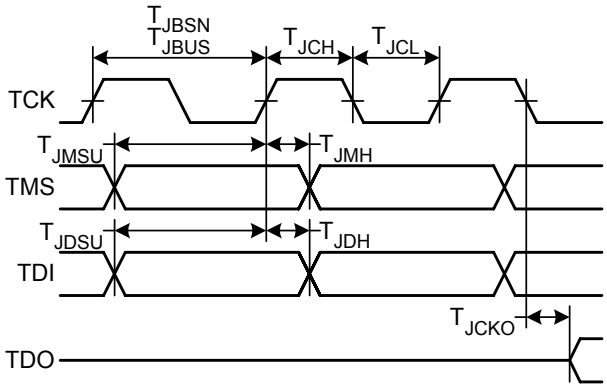


Figure 47. JTAG timing diagram.

JTAG Timing Characteristics

				Final		Final		
			Speed Grade	-25		-40		
Description	Symbol	Fig.	Device	Min	Max	Min	Max	Units
TCK frequency, boundary scan	F_{JBSN}	47	All	0	10.0	0	10.0	MHz
TCK boundary scan cycle time	T_{JBSN}	47	All	100	∞	100	∞	ns
TCK frequency, bus access	F_{JBUS}	47	All	0	25.0	0	40.0	MHz
TCK bus access cycle time	T_{JBUS}	47	All	40.0	∞	25.0	∞	ns
TCK High time	T_{JCH}	47	All	11.0		11.0		ns
TCK Low time	T_{JCL}	47	All	11.0		11.0		ns
TCK rise time	T_{JCRT}	47	All		5.0		5.0	ns
TCK fall time	T_{JCFT}	47	All		5.0		5.0	ns
TDI setup time before TCK rising edge	T_{JDSU}	47	All	5.0		5.0		ns
TDI hold time after TCK rising edge	T_{JDH}	47	All	2.0		2.0		ns
TMS setup time before TCK rising edge	T_{JMSU}	47	All	5.0		5.0		ns
TMS hold time after TCK rising edge	T_{JMH}	47	All	2.0		2.0		ns
TDO valid after TCK falling edge	T_{JCKO}	47	All	1.0	15.7	1.0	11.5	ns
				Final		Final		

Pin-to-Pin Guaranteed Timing Specifications

All Triscend devices are 100% functionally tested. These parameters are modeled after the testing methods described by MIL-M-38510/605. Pin-to-pin timing parameters are derived by measuring external and internal test patterns. The values listed below are representative for worst-case pin locations and clock loading. Actual values may depend on application-specific use. FastChip reports specific, worst-case guaranteed values in the Timing Analysis section of the project report.

All timing values shown assume worst-case operating conditions, including process technology, power supply voltage, and junction temperature.

Bus Clock Input to Output Delay

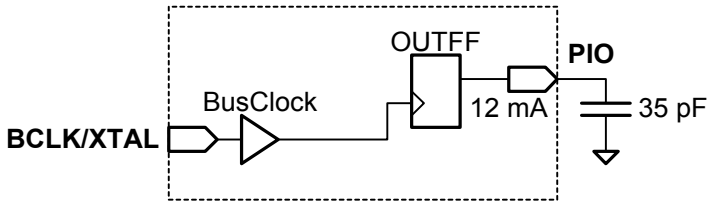


Figure 48. Bus Clock to Output from output flip-flop, 12 mA driver.

Description	Symbol	Speed Grade / Device	All / Min [1]	-25 / Max	-40 / Max	Units
Bus Clock Input (BCLK/XTAL) pad to output delay using output flip-flop.	T_{BPCO}	TE502	2.0	21.8	16.0	ns
		TE505	2.0	21.8	16.0	ns
		TE512	2.0	22.7	16.7	ns
		TE520	2.0	24.0	17.5	ns
			Final			

Note 1: Not tested. Guaranteed by design.

Bus Clock Input to Output Enable Delay

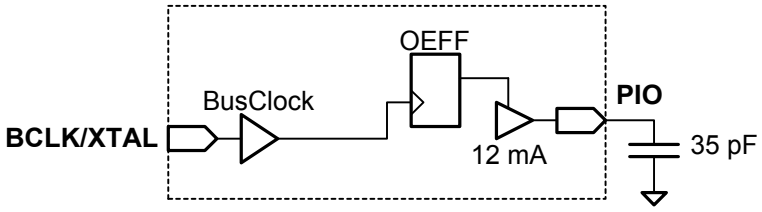


Figure 49. Bus Clock to Output from output flip-flop, 12 mA driver.

Description	Symbol	Speed Grade / Device	All / Min [1]	-25 / Max	-40 / Max	Units
Bus Clock Input (BCLK/XTAL) pad to output valid using output-enable flip-flop.	T_{BPEO}	TE502	2.0	20.0	15.0	ns
		TE505	2.0	20.0	15.0	ns
		TE512	2.0	20.0	15.0	ns
		TE520	2.0	20.0	15.0	ns
Bus Clock Input (BCLK/XTAL) pad to output high-impedance (hi-Z) using output-enable flip-flop.	T_{BPEZ}	TE502	2.0	26.0	20.0	ns
		TE505	2.0	26.0	20.0	ns
		TE512	2.0	26.0	20.0	ns
		TE520	2.0	32.0	22.0	ns
			Final			

Note 1: Not tested. Guaranteed by design.

Bus Clock, Input Set-Up and Hold

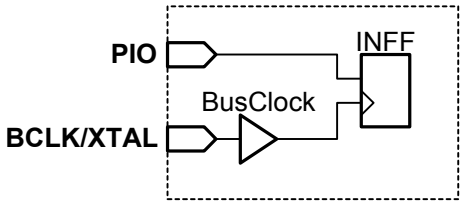


Figure 50. PIO setup to input flip-flop before Bus Clock.

Description	Symbol	Speed Grade / Device	-25 / Min	-40 / Min	Units
Set-Up: Delayed Input Data set-up time before active clock edge to input flip-flop or latch using the Bus Clock Input as clock, delayed input path.	T_{BPSU}	TE502	3.5	2.7	ns
		TE505	4.1	3.2	ns
		TE512	5.0	3.8	ns
		TE520	6.2	4.7	ns
Set-Up: No Delayed Input Data set-up time before active clock edge to input flip-flop or latch using the Bus Clock Input as clock, no delayed input path.	T_{BPSN}	TE502	0	0	ns
		TE505	0	0	ns
		TE512	0	0	ns
		TE520	0	0	ns
Hold: Delayed Input Data hold time after active clock edge to input flip-flop or latch using the Bus Clock Input as clock, delayed input path.	T_{BPHT}	TE502	0	0	ns
		TE505	0	0	ns
		TE512	0	0	ns
		TE520	0	0	ns
		TE502	4.5	3.5	ns

Set-Up: No Delayed Input Data hold time after active clock edge to input flip-flop or latch using the Bus Clock Input as clock,	T_{BPHN}	TE502	4.5	3.5	ns
		TE505	5.8	4.2	ns
		TE520	6.1	4.6	ns
			Final		

Global Buffer Input to Output Delay

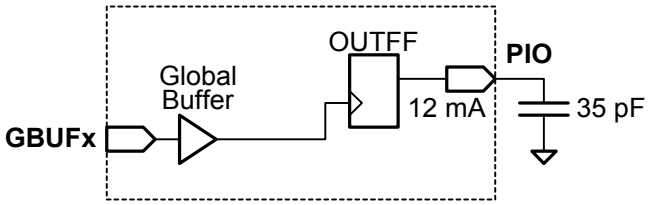


Figure 51. Global Buffer clock to Output from output flip-flop, 12 mA driver.

		Speed Grade	All	-25	-40	
Description	**Symbol**	**Device**	**Min** [1]	**Max**	**Max**	**Units**
Global Buffer (GBUFx) pad to output delay using output flip-flop.	T_{GPCO}	TE505	2.0	19.5	14.5	ns
		TE505	2.0	19.5	14.5	ns
		TE512	2.0	19.5	14.5	ns
		TE520	2.0	22.7	16.7	ns
				Final		

Note 1: Not tested. Guaranteed by design.

Bus Clock Input to Output Enable Delay

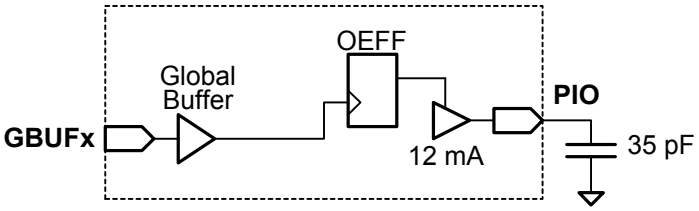


Figure 52. Bus Clock to Output from output flip-flop, 12 mA driver.

		Speed Grade	All	-25	-40	
Description	**Symbol**	**Device**	**Min** [1]	**Max**	**Max**	**Units**
Bus Clock Input (BCLK/XTAL) pad to output valid using output-enable flip-flop.	T_{GPEO}	TE502	2.0	15.0	11.0	ns
		TE505	2.0	15.8	11.7	ns
		TE512	2.0	15.8	11.7	ns
		TE520	2.0	18.3	13.5	ns
		TE502	2.0	24.3	18.5	ns
		TE505	2.0	24.3	18.5	ns
		TE512	2.0	24.3	18.5	ns
		TE520	2.0	25.5	19.3	ns
				Final		

Note 1: Not tested. Guaranteed by design.

Global Buffer, Input Set-Up and Hold

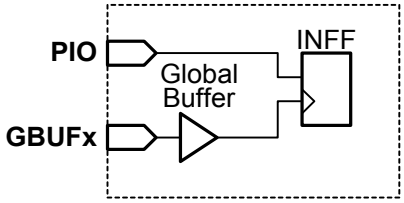


Figure 53. PIO setup to input flip-flop before Global Buffer clock.

		Speed Grade	-25	-40	Units
Description	**Symbol**	**Device**	**Min**	**Min**	
Set-Up: Delayed Input Data set-up time before active clock edge to input flip-flop or latch using a Global Buffer as clock, delayed input path.	T_{GPSU}	TE502	4.8	3.6	ns
		TE505	5.2	4.0	ns
		TE512	6.5	5.0	ns
		TE520	7.5	5.5	ns
Set-Up: No Delayed Input Data set-up time before active clock edge to input flip-flop or latch using a Global Buffer as clock, no delayed input path.	T_{GPSN}	TE502	0	0	ns
		TE505	0	0	ns
		TE512	0	0	ns
		TE520	0	0	ns
Hold: Delayed Input Data hold time after active clock edge to input flip-flop or latch a Global Buffer as clock, delayed input path.	T_{GPHT}	TE502	0	0	ns
		TE505	0	0	ns
		TE512	0	0	ns
		TE520	0	0	ns
Hold: No Delayed Input Data hold time after active clock edge to input flip-flop or latch a Global Buffer as clock, delayed input path.	T_{GPHN}	TE502	3.4	2.6	ns
		TE505	3.6	2.8	ns
		TE512	3.6	2.8	ns
		TE520	3.9	3.0	ns
			Final		

Memory Interface Unit (MIU) Timing Characteristics

The Memory Interface Unit on the E5 has user-defined control timing. Various control registers define the setup, strobe width, and hold time for output-enable, write-enable, and chip-enable signals. These registers are configured from within the FastChip development system.

Memory Interface Unit (MIU) Functional Diagram

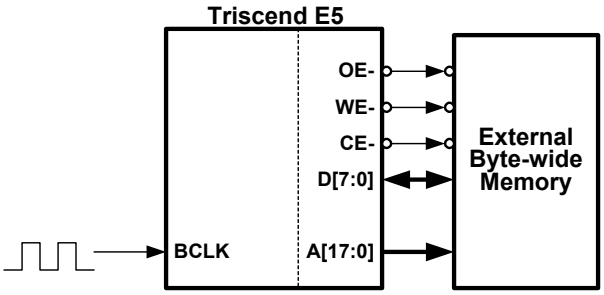


Figure 54. Memory Interface Unit.

Example Memory Interface Unit (MIU) Waveforms

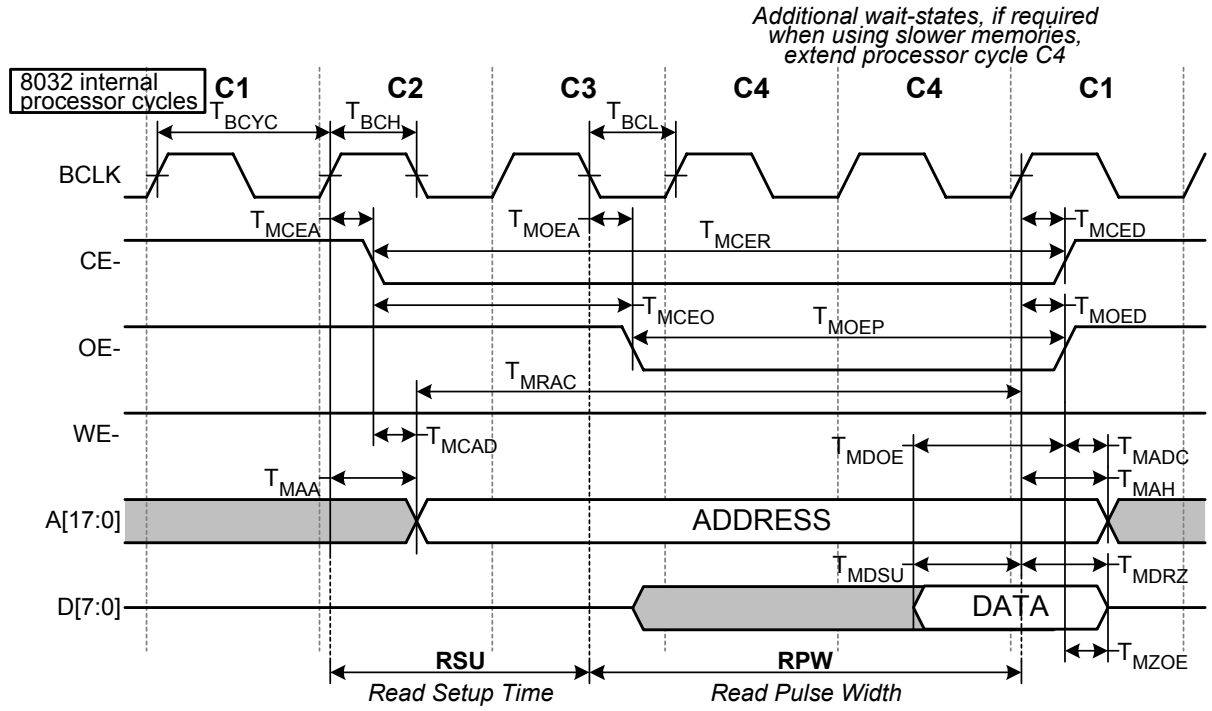


Figure 55. 8051 instruction-fetch (read) from external memory via Memory Interface Unit (MIU). In the figure, the read setup time is 1.5 Bus Clock cycles (RSU=1) and the read strobe pulse width is asserted for 2.5 Bus Clock cycles (RPW=2). Note that the MIU extends the microcontroller's C4 instruction cycle if the E5 is connected to a slow external memory.

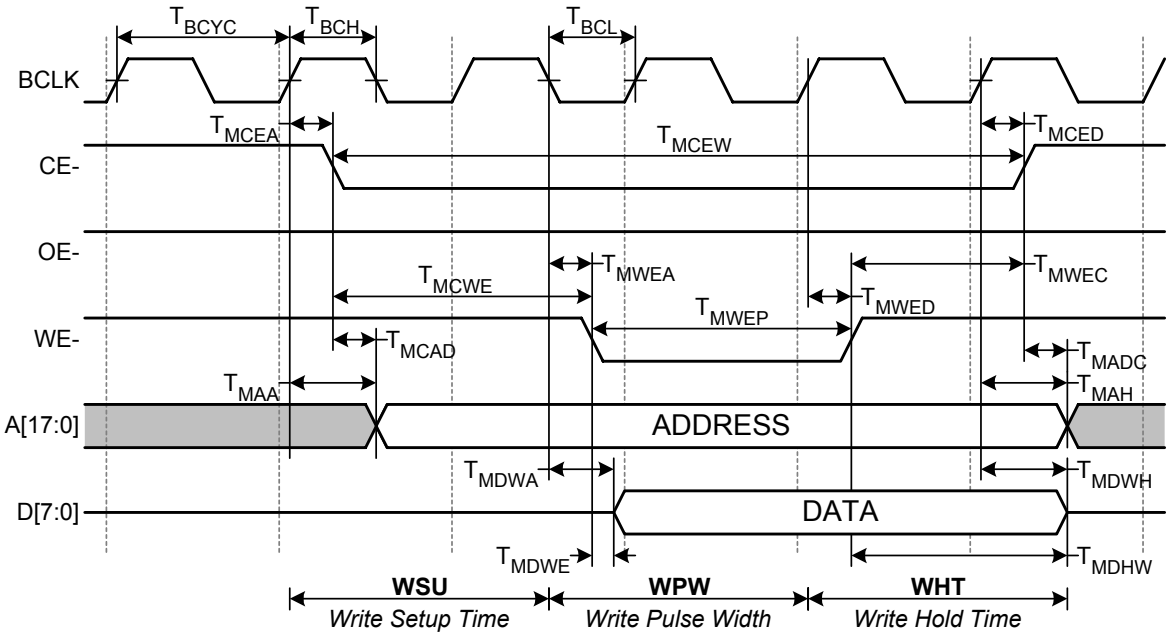


Figure 56. MIU write cycle to external memory. In the figure, the write setup time is 1.5 Bus Clock cycles (WSU=1), the write strobe pulse width is asserted for 1.5 Bus Clock cycles (WPW=1), and the write data hold time is 1.0 Bus Clock cycles (WHT=1).

Memory Interface Unit (MIU) Timing Characteristic Guidelines, Stand-alone Operation (SLAVE- = High)

All Triscend devices are 100% functionally tested. These parameters are modeled after the testing methods described by MIL-M-38510/605. Pin-to-pin timing parameters are derived by measuring external and internal test patterns. The values listed below are representative for typical pin locations and normal clock loading. For these tests, Bus Clock is supplied externally via the BCLK pin. Actual values may depend on application-specific use. FastChip reports specific, worst-case guaranteed values in the Timing Analysis section of the project report.

All timing values shown assume worst-case operating conditions, including process technology, power supply voltage, and junction temperature.

				Preliminary		*Preliminary*		
			Speed Grade	-25		-40		
Description	**Symbol**	**Fig.**	**Device**	**Min**	**Max**	**Min**	**Max**	**Units**
Bus Clock								
Bus Clock frequency	F_{BCLK}	54–56	All	0	25.0	0	40.0	MHz
Bus Clock cycle period	T_{BCYC}	54–56	All	40.0	∞	25.0	∞	ns
Bus Clock High time	T_{BCH}	54–56	All	12.0		12.0		ns
Bus Clock Low Time	T_{BCL}	54–56	All	12.0		12.0		ns
Enables								
Bus Clock rising edge to CE- asserted	T_{MCEA}	54–56	All		24.0		18.0	ns
Bus Clock rising edge to CE- de-asserted	T_{MCED}	54–56	All		22.0		16.5	ns
Bus Clock falling edge to OE- asserted	T_{MOEA}	54,55	All		22.0		16.5	ns
Bus Clock to rising edge OE- de-asserted	T_{MOED}	54,55	All		20.0		15.0	ns
Bus Clock falling edge to WE- asserted	T_{MWEA}	54,56	All		22.0		16.0	ns
Bus Clock rising edge to WE- de-asserted	T_{MWED}	54,56	All		20.0		14.5	ns
Data during Read Operation								
Setup time on Data before Bus Clock, read operation	T_{MDSU}	54,55	All	3.0		2.0		ns
Data hold time after read, relative to Bus Clock	T_{MDRZ}	54,55	All	13.0		10.0		ns
Data during Write Operation								
Data valid after Bus Clock, write operation	T_{MDWA}	54,56	All		22.5		16.0	ns
Address								
Address valid after Bus Clock	T_{MAA}	54–56	All		25.0		18.0	ns

Table 46. MIU Control Values.

Value	Description	Legal Values
RSU	Read Setup. Controlled by bits MIUCTRL1[3:1] in the MIU control register.	0, 1, 2, 3, 4, 5, 6, 7
RPW	Read Pulse Width. Controlled by bits MIUCTRL1[6:4] in the MIU control register.	0, 1, 2, 3, 4, 5, 6, 7
WSU	Write Setup. Controlled by bits MIUCTRL0[2:0] in the MIU control register.	0, 1, 2, 3, 4, 5, 6, 7
WPW	Write Pulse Width. Controlled by bits MIUCTRL0[5:3] in the MIU control register.	0, 1, 2, 3, 4, 5, 6, 7
WHT	Write Hold Time. Controlled by bits MIUCTRL1[0] and MIUCTRL0[7:6] in the MIU control register.	0, 1, 2, 3, 4, 5, 6, 7

Asynchronous Memory Interface Timing

The following values are derived from MIU tests where Bus Clock is provided via an external source. While using the internal crystal oscillator amplifier or the internal ring oscillator, Bus Clock may not be externally visible. The following values provide relative timing referencing only signals available on the MIU. The timing of the MIU control signals is determined by the bit settings in the MIU control register, as shown in Table 46.

For supported memory devices, FastChip automatically configures the proper pulse width and strobe settings based on the speed of the selected device and the frequency of Bus Clock.

					Preliminary		*Preliminary*		
				Speed Grade	**-25**		**-40**		
Description	**Symbol**	**Fig.**	**Device**	**Start Here → Add Min or Max — Variable Timing [1]**	**Min**	**Max**	**Min**	**Max**	**Units**
Address valid after CE- asserted	T_{MCAD}	54–56	All	N/A	0	2.0	0	2.0	ns
Address hold time after CE- de-asserted	T_{MADC}	54–56	All	N/A	0	3.0	0	1.5	ns
Read Timing									
8051 fetch (read) access time [2]	T_{MRAC}	54,55	All	[(RSU+RPW+1)• T_{BCYC}] + Min	-28.0		-20.0		ns
CE- pulse width, read	T_{MCER}	54,55	All	[(RSU+RPW+1)• T_{BCYC}] + Max		-2.0		-1.5	ns
OE- asserted after CE- asserted, read [3]	T_{MCOE}	54,55	All	(RSU• T_{BCYC}) + T_{BCH} + Min	-2.0		-1.5		ns
OE- pulse width, read [3]	T_{MOEP}	54,55	All	(RPW• T_{BCYC}) + T_{BCL} + Max		-2.0		-1.5	ns
Read data on D[7:0] setup time before OE- de-asserted	T_{MDOE}	54,55	All	N/A	23.0		17.0		ns
CE- de-asserted after OE- de-asserted	T_{MOEC}	54,55	All	N/A		2.0		1.5	ns
Data hold time (float) after OE- de-asserted	T_{MZOE}	54,55	All	N/A	0		0		ns
Write Timing									
CE- pulse width, write	T_{MCEW}	54,56	All	[(WSU+WPW+WHT+1)• T_{BCYC}]+ Max		-2.0		-1.5	ns
WE- asserted after CE- asserted, write [3]	T_{MCWE}	54,56	All	(WSU• T_{BCYC}) + T_{BCH} + Min	-2.5		-1.5		ns
WE- pulse width, write [3]	T_{MWEP}	54,56	All	(WPW• T_{BCYC}) + T_{BCL} + Max		-2.0		-1.5	ns
Write data valid on D[7:0] after WE-asserted	T_{MDWE}	54,56	All	NA	0	2.0	0	2.0	ns
Write data hold time after WE- de-asserted	T_{MDHW}	54,56	All	(WHT• T_{BCYC}) + Min or Max	0	+2.5	0	+1.5	ns
CE- de-asserted after WE- de-asserted	T_{MWEC}	54,56	All	(WHT• T_{BCYC}) + Max or Max	0	+2.0	0	+2.0	ns
					Preliminary		*Preliminary*		

Note 1: The MIU control register controls the timing for the read setup time (RSU), the read pulse width (RPW), the write setup time (WSU), the write pulse width (WPW), and the write hold time (WHT). Table 46 shows the legal values for each parameter. The MIU control register timings do not affect external accesses controlled by Selectors in the CSL matrix.

Note 2: The 8051 microcontroller operates at maximum performance—*i.e.*, no wait-states—when RSU=0 and RPW=2, resulting in a 3 clock-cycle read strobe.

Note 3: If the source for Bus Clock is an external clock driver with known characteristics, use the actual minimum clock High and clock Low times for the clock source. In all other cases, assume the minimum times, T_{BCH} and T_{BCH}.

Configurable System Interconnect (CSI) Socket Timing Guidelines

The Triscend Configurable System Interconnect (CSI) bus socket provides a family-independent interface between the processor, its peripherals, and the Configurable System Logic (CSL) matrix. The following values provide a typical range of worst-case delays based on test designs. The actual values depend on a variety of factors including how the CSL logic cells connect to the CSI bus in the application, the utilization of the CSL matrix, and the number of modules in the design. The tables also show how much additional logic delay, or slack time, is allowed in each type of logic path while operating at the maximum bus clock frequency, F_{BCLK}.

CSI Socket Functional Diagram

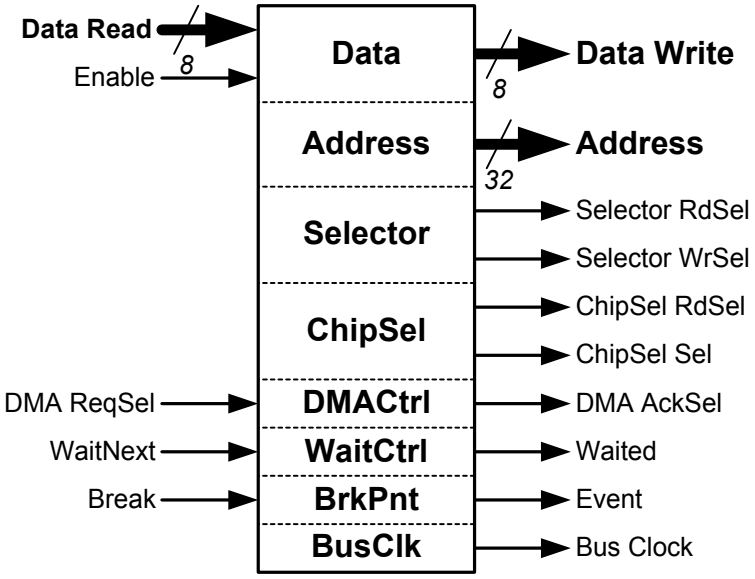


Figure 57. Configurable System Interconnect (CSI) bus socket.

CSI Socket Timing Benchmark Circuits

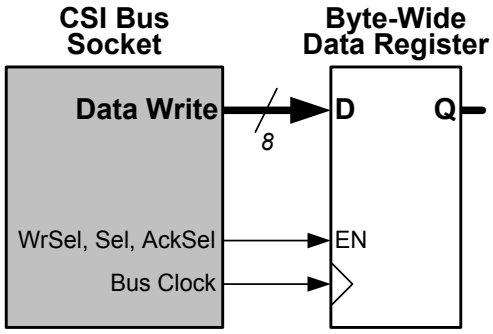


Figure 58. CSI bus write to byte-wide register.

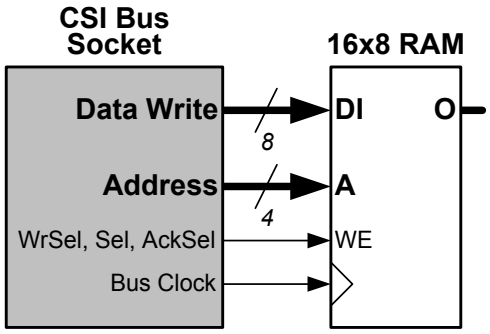


Figure 59. CSI bus write to 16x8 RAM.

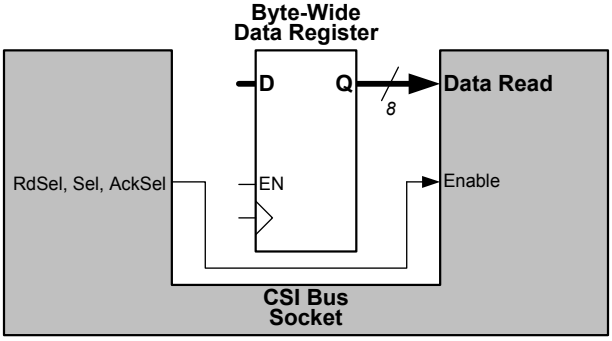


Figure 60. CSI bus read from byte-wide register.

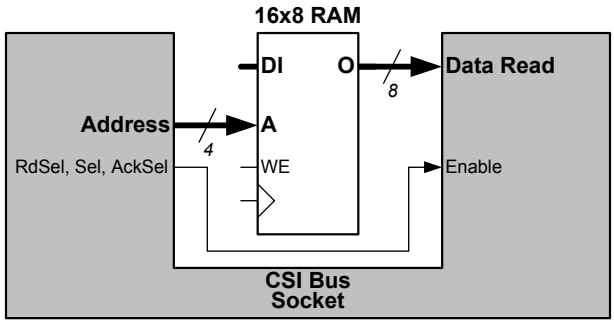


Figure 61. CSI bus read from 16x8 RAM.

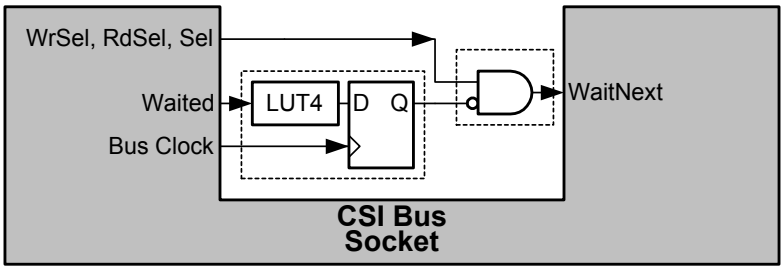


Figure 62. Wait-state control circuit.

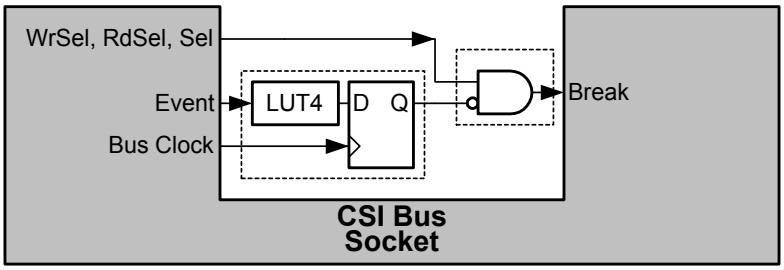


Figure 63. CSL Breakpoint control circuit.

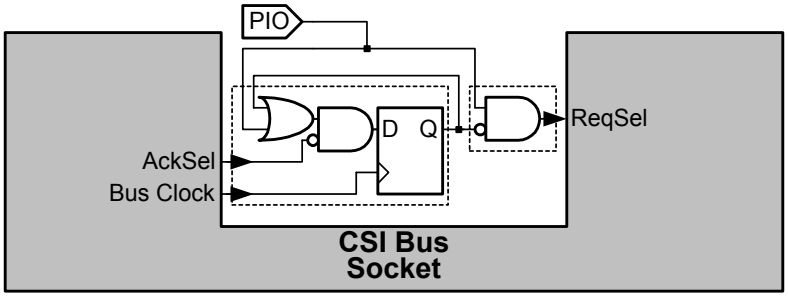


Figure 64. CSL DMA request/acknowledge circuit.

CSI Socket Timing Characteristic Guidelines

The values listed below are representative, guideline values extracted from measured internal test patterns. Actual values may depend on application-specific use. FastChip reports specific, worst-case guaranteed values in the Timing Analysis section of the project report.

				Guideline	*Guideline*	
			Speed Grade	*-25*	*-40*	
Description	**Symbol**	**Fig.**	**Device**	**Typical**	**Typical**	**Units**
Bus Write Operations						
Data Write valid, distributed and setup before writing to byte-wide register, through 4-input LUT	T_{CDWD}	58	All	8.0 ⇔ 16.0	6.0 ⇔ 12.0	ns
Additional logic and interconnect delay allowed in Data Write path to byte-wide register, through 4-input LUT, operating at maximum clock frequency, F_{BCLK}	T_{BCYC} - T_{CDWD}	58	All	24.0 ⇔ 32.0	13.0 ⇔ 19.0	ns
Data Write valid, distributed and setup to 16x8 RAM	T_{CDWR}	59	All	8.0 ⇔ 16.0	6.0 ⇔ 12.0	ns
Additional logic and interconnect delay allowed in Data Write path to 16x8 RAM, operating at maximum clock frequency, F_{BCLK}	T_{BCYC} - T_{CDWR}	59	All	24.0 ⇔ 32.0	13.0 ⇔ 19.0	ns
Selector output valid (WrSel, Sel, or AckSel), distributed and setup before writing to byte-wide register	T_{CDWS}	58	All	12.0 ⇔ 28.0	7.0 ⇔ 15.0	ns
Additional logic and interconnect delay allowed in Selector path to byte-wide register, operating at maximum clock frequency, F_{BCLK}	T_{BCYC} - T_{CDWS}	58	All	12.0 ⇔ 28.0	10.0 ⇔ 18.0	ns
Selector output valid (WrSel, Sel, or AckSel), distributed and setup before writing to 16x8 RAM	T_{CDWE}	59	All	12.0 ⇔ 28.0	7.0 ⇔ 15.0	ns
Additional logic and interconnect delay allowed in Selector path to 16x8 RAM, operating at maximum clock frequency, F_{BCLK}	T_{BCYC} - T_{CDWE}	59	All	12.0 ⇔ 28.0	10.0 ⇔ 18.0	ns
Address valid, distributed and setup before writing to 16x8 RAM	T_{CDWA}	59	All	15.0 ⇔ 19.0	8.0 ⇔ 12.0	ns
Additional logic and interconnect delay allowed in Address path to 16x8 RAM, operating at maximum clock frequency, F_{BCLK}	T_{BCYC} - T_{CDWA}	59	All	11.0 ⇔ 25.0	13.0 ⇔ 17.0	ns
Bus Read Operations						
Data output Q valid on byte-wide register, distributed and setup before read operation	T_{CDRD}	60	All	5.0 ⇔ 10.0	5.0 ⇔ 10.0	ns
Additional logic and interconnect delay allowed in Data Read path from Q output on byte-wide register, operating at maximum clock frequency, F_{BCLK}	T_{BCYC} - T_{CDRD}	60	All	15.0 ⇔ 20.0	15.0 ⇔ 20.0	ns
Data output O valid on 16x8 RAM, distributed and setup before read operation	T_{CDRR}	61	All	9.0 ⇔ 19.0	7.0 ⇔ 14.0	ns
Additional logic and interconnect delay allowed in Data Read path from O output on 16x8 RAM, operating at maximum clock frequency, F_{BCLK}	T_{BCYC} - T_{CDRR}	61	All	11.0 ⇔ 21.0	11.0 ⇔ 18.0	ns
Selector output valid (RdSel, Sel, or AckSel), distributed and setup before read operation	T_{CDRS}	60, 61	All	12.0 ⇔ 28.0	9.0 ⇔ 14.0	ns
Additional logic and interconnect delay allowed in Data Read, Selector output to Enable, operating at maximum clock frequency, F_{BCLK}	T_{BCYC} - T_{CDRS}	60, 61	All	12.0 ⇔28.0	11.0 ⇔ 16.0	ns
Address valid, distributed to 16x8 RAM, data output O valid, distributed and setup before read operation	T_{CDRC}	61	All	20.0 ⇔ 30.0	15.0 ⇔ 17.0	ns
Additional logic and interconnect delay allowed in Data Read, Address through 16x8 output O, operating at maximum clock frequency, F_{BCLK}	T_{BCYC} - T_{CDRC}	61	All	10.0 ⇔ 20.0	8.0 ⇔ 10.0	ns
				Guideline	***Guideline***	

Description	Symbol	Fig.	Speed Grade: Device	Guideline -25: Typical	Guideline -40: Typical	Units
Wait-State Control						
Selector output valid (WrSel, RdSel, or Sel), through LUT4, distributed and setup to WaitNext	T_{CWNS}	62	All	12.0 ⇔ 28.0	9.0 ⇔ 16.0	ns
Additional logic and interconnect delay allowed in Selector path before WaitNext, operating at maximum clock frequency, F_{BCLK}	T_{BCYC} - T_{CWNS}	62	All	12.0 ⇔ 28.0	9.0 ⇔ 16.0	ns
CSL flip-flop output Q valid, through LUT4, distributed and setup to WaitNext	T_{CWNQ}	62	All	16.0 ⇔ 20.0	9.0⇔ 15.0	ns
Additional logic and interconnect delay allowed in CSL flip-flop Q output path before WaitNext, operating at maximum clock frequency, F_{BCLK}	T_{BCYC} - T_{CWNQ}	62	All	20.0 ⇔ 24.0	10.0 ⇔ 16.0	ns
Waited output valid, through LUT4, distributed and setup to CSL flip-flop	T_{CWTD}	62	All	12.0 ⇔ 28.0	7.0 ⇔ 15.0	ns
Additional logic and interconnect delay allowed in Waited path before CSL flip-flop, operating at maximum clock frequency, F_{BCLK}	T_{BCYC} - T_{CWTD}	62	All	12.0 ⇔ 28.0	10.0 ⇔ 18.0	ns
Breakpoint Control						
Selector output valid (WrSel, RdSel, or Sel), through LUT4, distributed and setup to Break	T_{CBPS}	63	All	12.0 ⇔ 28.0	9.0 ⇔ 16.0	ns
Additional logic and interconnect delay allowed in Selector path before Break, operating at maximum clock frequency, F_{BCLK}	T_{BCYC} - T_{CBPS}	63	All	12.0 ⇔ 28.0	9.0 ⇔ 16.0	ns
CSL flip-flop output Q valid, through LUT4, distributed and setup to Break	T_{CBPQ}	63	All	16.0 ⇔ 20.0	9.0⇔ 15.0	ns
Additional logic and interconnect delay allowed in CSL flip-flop Q output path before Break, operating at maximum clock frequency, F_{BCLK}	T_{BCYC} - T_{CBPQ}	63	All	20.0 ⇔ 24.0	10.0 ⇔ 16.0	ns
Event output valid, through LUT4, distributed and setup to CSL flip-flop	T_{CEVT}	63	All	12.0 ⇔ 28.0	7.0 ⇔ 15.0	ns
Additional logic and interconnect delay allowed in Event path before CSL flip-flop, operating at maximum clock frequency, F_{BCLK}	T_{BCYC} - T_{CEVT}	63	All	12.0 ⇔ 28.0	10.0 ⇔ 18.0	ns
DMA Request/Acknowledge						
PIO input valid, through LUT4, distributed and setup to ReqSel	T_{CRSP}	64	All	17.0 ⇔ 25.0	10.0 ⇔ 15.0	ns
Additional logic and interconnect delay allowed in PIO request path before ReqSel, operating at maximum clock frequency, F_{BCLK}	T_{BCYC} - T_{CRSP}	64	All	15.0 ⇔ 23.0	10.0 ⇔ 15.0	ns
CSL flip-flop output Q valid, through LUT4, distributed and setup to ReqSel	T_{CRSQ}	64	All	16.0 ⇔ 20.0	9.0⇔ 15.0	ns
Additional logic and interconnect delay allowed in CSL flip-flop Q output path before ReqSel, operating at maximum clock frequency, F_{BCLK}	T_{BCYC} - T_{CRSQ}	64	All	20.0 ⇔ 24.0	10.0 ⇔ 16.0	ns
AckSel output valid, through LUT4, distributed and setup to CSL flip-flop	T_{CACK}	64	All	12.0 ⇔ 28.0	7.0 ⇔ 15.0	ns
Additional logic and interconnect delay allowed in AckSel path before CSL flip-flop, operating at maximum clock frequency, F_{BCLK}	T_{BCYC} - T_{CACK}	64	All	12.0 ⇔ 28.0	10.0 ⇔ 18.0	ns
				Guideline	***Guideline***	

Sideband Signal Timing Characteristics

The sideband signals are controls to and from the embedded Accelerated 8051 microcontroller, unique to the Triscend E5 family. Each signal is associated with a specific dedicated resource inside the 8051 microcontroller.

The values below indicate the number of Bus Clock cycles required for the signal to be recognized by the microcontroller.

Sideband Signal Functional Diagram

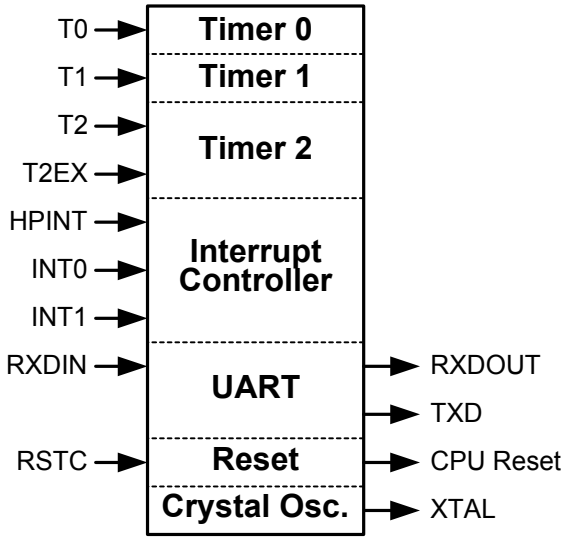


Figure 65. Embedded Accelerated 8051 microcontroller sideband signals.

Sideband Signal Timing Characteristic Guidelines

				Final		*Final*		
			Speed Grade	-25		-40		
Description	**Symbol**	**Fig.**	**Device**	**Min**	**Max**	**Min**	**Max**	**Units**
Timer 0								
T0 pulse width	T_{ST0P}	65	All	5		5		T_{BCYC}
Timer 1								
T1 pulse width	T_{ST1P}	65	All	5		5		T_{BCYC}
Timer 2								
T2 pulse width	T_{ST2P}	65	All	5		5		T_{BCYC}
T2EX pulse width	T_{ST2XP}	65	All	5		5		T_{BCYC}
Interrupts								
HPINT pulse width	T_{SHPP}	65	All	5		5		T_{BCYC}
INT0 pulse width	T_{SIT0P}	65	All	5		5		T_{BCYC}
INT1 pulse width	T_{SIT1P}	65	All	5		5		T_{BCYC}
Resets								
Reset input to 8051, RSTC pulse width	T_{SRSTC}	65	All	2		2		T_{BCYC}
				Final		*Final*		

Configurable System Logic (CSL) Cell (Combinatorial Logic Mode, Sequential Mode)

In combinatorial mode, a single CSL cell implements any possible logic function of zero to four inputs. Likewise, a single CSL cell behaves like a 16x1 read-only memory (ROM). The contents of the ROM are loaded during initialization.

Two CSL cells operating in tandem implement any possible logic function of zero to five inputs. Likewise, two CSL cells in tandem behave like a 32x1 read-only memory (ROM). The contents of the ROM are loaded during initialization.

Two CSL cells operating in tandem also implement a limited number of logic functions of between six to nine inputs.

Special logic allows the logic functions to be cascaded into wider functions.

In sequential mode, each CSL provides a 'D'-type, edge-triggered flip-flop with clock-enable control and an asynchronous set or clear control.

CSL Combinatorial Logic and Sequential Mode Functional Diagrams

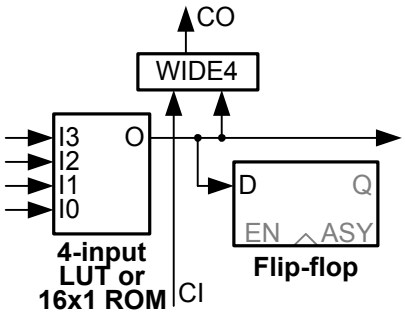


Figure 66. Four-input logic function.

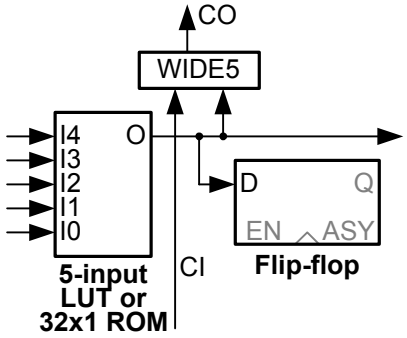


Figure 67. Five-input logic function, requires two CSL cells in tandem.

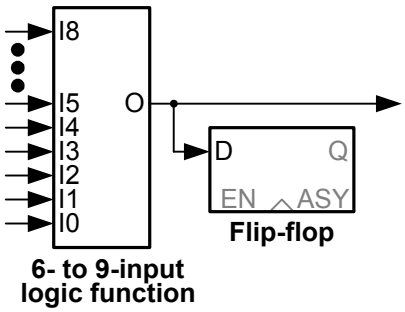


Figure 68. Six- to nine- input logic function, requires two CSL cells in tandem.

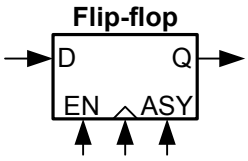


Figure 69. CSL cell flip-flop.

CSL Combinatorial Logic and Sequential Mode Timing Characteristic Guidelines

The values listed below are representative, guideline values extracted from measured internal test patterns. Actual values may depend on application-specific use. FastChip reports specific, worst-case guaranteed values in the Timing Analysis section of the project report.

All timing values shown assume worst-case operating conditions, including process technology, power supply voltage, and junction temperature.

				Guideline		Guideline		
			Speed Grade	-25		-40		
Description	Symbol	Fig.	Device	Min	Max	Min	Max	Units
Combinatorial Functions								
Four-input logic function or 16x1 ROM, inputs I3 - I0 to O output, no output driver delay [1]	T_{ILO}	66	All		3.1		2.4	ns
Four-input logic function or 16x1 ROM, inputs I3 - I0 to O output [2]	T_{LUT4}	66	All		4.5		3.1	ns
Five-input logic function or 32x1 ROM, inputs I4 - I0 to O output [2]	T_{LUT5}	67	All		5.5		4.0	ns
Six- to nine-input logic function, inputs I8 - I0 to O output [2]	T_{LOG9}	68	All		5.5		4.0	ns
Cascaded, Wide Functions								
Cascaded four-input logic function or 16x1 ROM, inputs I3 - I0 to CO output [2, 3]	T_{LCO4}	66	All		6.9		4.0	ns
Cascaded logic function WIDE4, input CI to CO output [2, 3]	T_{CCO4}	66	All		4.3		2.9	ns
Cascaded five-input logic function or 32x1 ROM, inputs I4 - I0 to CO output [2, 3]	T_{LCO5}	67	All		5.5		4.0	ns
Cascaded logic function WIDE5, input CI to CO output [2, 3]	T_{CCO5}	67	All		4.1		3.4	ns
Sequential Delays								
CSL cell clock input CK to flip-flop output Q [2]	T_{CKO}	69	All		4.0		2.9	ns
Setup Time before CSL Cell Clock CK								
LUT4 or 16x1 ROM inputs I3 - I0 to flip-flop D input	T_{LIK4}	66	All	5.1		4.0		ns
LUT5 or 32x1 ROM inputs I4 - I0 to flip-flop D input	T_{LIK5}	67	All	6.1		4.8		ns
Logic inputs I8 - I0 to flip-flop D input	T_{LIK9}	68	All	6.1		4.8		ns
Cascade input, CI	T_{CIIK}		All	4.9		3.7		ns
Direct input, DI, bypassing LUT	T_{DIK}	69	All	2.0		1.5		ns
Clock enable input EN	T_{CEK}	69	All	1.1		0.9		ns
Hold Time after CSL Cell Clock								
All hold times	T_{CQH}	69	All		0		0	ns
CSL Cell Clock								
Clock High time, internally generated	T_{CHC}	69	All	2.5		2.0		ns
Clock Low time, internally generated	T_{CLC}	69	All	2.5		2.2		ns
Asynchronous Control Input								
Pulse width (High)	T_{APW}	69	All	0.7		0.5		ns
Delay from Asynchronous input asserted to Q	T_{AQ}	69	All	1.9		1.5		ns
CSL Toggle Frequency								
Toggle frequency, Q output through local interconnect, through LUT4, to setup on flip-flop [4]	F_{TGL}	66	All		123.7		158.8	MHz
				Guideline		Guideline		

Note 1: Calculated using methods similar to other programmable logic vendors

Note 2: Includes output delay driving onto a single interconnect segment, T_{ZIP}

Note 3: If CO output drives CI of adjacent block, subtract output delay, T_{ZIP}, 1.4 ns for -25, 0.7 ns for -40

Note 4: Derived from FastChip test design

Configurable System Logic (CSL) Cell (Arithmetic Mode)

In arithmetic mode, a CSL implements a single bit preloadable adder or subtracter. Alternatively, a CSL implements a single-bit adder/subtracter with a separate add/subtract control. As a single-bit multiplier, a CSL cell has a partial-sum input from the previous multiplier stage.

As with combinatorial logic, the outputs from an arithmetic function can be stored in the CSL cell's flip-flop.

CSL Arithmetic Mode Functional Diagrams

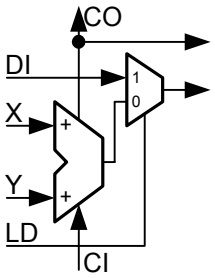


Figure 70. Preloadable Adder.

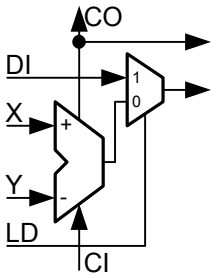


Figure 71. Preloadable Subtracter.

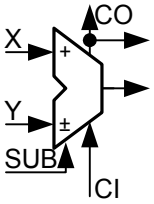


Figure 72. Adder/Subtracter.

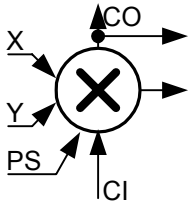


Figure 73. Multiplier.

CSL Arithmetic Mode Timing Characteristic Guidelines

The values listed below are representative, guideline values extracted from measured internal test patterns. Actual values may depend on application-specific use. FastChip reports specific, worst-case guaranteed values in the Timing Analysis section of the project report.

All timing values shown assume worst-case operating conditions, including process technology, power supply voltage, and junction temperature.

				Guideline		*Guideline*		
			Speed Grade	-25		-40		
Description	**Symbol**	**Fig.**	**Device**	**Min**	**Max**	**Min**	**Max**	**Units**
All Arithmetic Functions								
Operands X, Y to SUM output [1]	T_{OPS}	70–73	All		4.5		3.1	ns
CI input to SUM output [1]	T_{CIS}	70–73	All		4.9		3.3	ns
Operands X, Y to CO output [1,2]	T_{OPC}	70–73	All		5.5		3.8	ns
CI input to CO output [1,2]	T_{CCOA}	70–73	All		4.2		1.6	ns
Adder or Subtracter Only								
DI data input to SUM output [1]	T_{DIS}	70,71	All		3.9		2.6	ns
DI data input to CO output [1,2]	T_{DICO}	70,71	All		6.3		3.6	ns
LD load input to SUM output [1]	T_{LDS}	70,71	All		3.3		2.3	ns
LD load input to CO output [1,2]	T_{LDCO}	70,71	All		6.3		3.4	ns
Adder/Subtracter Only								
SUB add/subtract input to SUM output [1]	T_{SUS}	72	All		3.3		2.3	ns
SUB add/subtract input to CO output [1,2]	T_{SCO}	72	All		5.6		3.4	ns
Multiplier Only								
PS partial sum input to SUM output [1]	T_{PS}	73	All		4.5		3.0	ns
PS partial sum input to CO output [1,2]	T_{PCO}	73	All		5.5		3.3	ns
				Guideline		*Guideline*		

Note 1: Includes output delay driving onto a single interconnect segment, T_{ZIP}

Note 2: If CO output drives CI of adjacent block, subtract output delay, T_{ZIP}, 1.4 ns for -25, 0.7 ns for -40

Configurable System Logic (CSL) Cell (Memory Mode, Single-Port RAM)

In memory mode, a single CSL cell implements a 16x1, edge-triggered, single-port random-access memory (RAM). Two CSL cells in tandem operate as a 32x1, edge-triggered RAM. The outputs can be captured in the CSL cell's flip-flop.

CSL Memory Mode, Single-Port RAM Functional Diagrams

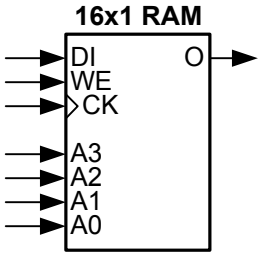


Figure 74. 16x1 single-port RAM.

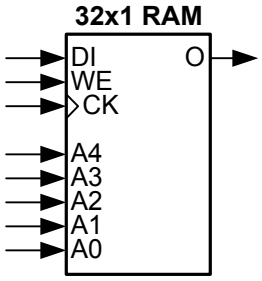


Figure 75. 32x1 single-port RAM.

CSL Memory Mode, Single-Port RAM Timing Characteristic Guidelines

The values listed below are representative, guideline values extracted from measured internal test patterns. Actual values may depend on application-specific use. FastChip reports specific, worst-case guaranteed values in the Timing Analysis section of the project report.

All timing values shown assume worst-case operating conditions, including process technology, power supply voltage, and junction temperature. Values include delay driving one interconnect segment.

				Guideline		*Guideline*		
			Speed Grade	*-25*		*-40*		
Description	**Symbol**	**Fig.**	**Size**	**Min**	**Max**	**Min**	**Max**	**Units**
Write Operation								
Address write cycle time (clock period)	T_{WC16} T_{WC32}	74,76 75,76	16x1 32x1	5.6 5.6		4.1 4.1		ns
CK clock pulse width (active edge)	T_{WP16} T_{WP32}	74,76 75,76	16x1 32x1	3.5 3.5		2.7 2.7		ns
Address setup time before CK	T_{AS16} T_{AS32}	74,76 75,76	16x1 32x1	0.8 0.8		0.6 0.6		ns
Address hold time after CK	T_{AH16} T_{AH32}	74,76 75,76	16x1 32x1		0 0		0 0	ns
DIN setup time before CK	T_{DS16} T_{DS32}	74,76 75,76	16x1 32x1	0.9 0.9		0.7 0.7		ns
DIN hold time after CK	T_{DH16} T_{DH32}	74,76 75,76	16x1 32x1		0 0		0 0	ns
WE setup time before CK	T_{WS16} T_{WS32}	74,76 75,76	16x1 32x1	1.6 1.6		1.2 1.2		ns
WE hold time after CK	T_{WH16} T_{WH32}	74,76 75,76	16x1 32x1		0 0		0 0	ns
Data valid at O after CK [1]	T_{WO16} T_{WO32}	74,76 75,76	16x1 32x1		7.0 8.0		5.2 6.3	ns
Read Operation								
Address read cycle time	T_{RC16} T_{RC32}	74,76 75,76	16x1 32x1	10.5 11.5		7.5 8.4		ns
Data valid after address change (no Write Enable) [1]	T_{AO16} T_{AO32}	74,76 75,76	16x1 32x1		4.5 5.5		3.0 4.0	ns
				Guideline		*Guideline*		

Note 1: Includes output delay driving onto a single interconnect segment, T_{ZIP}

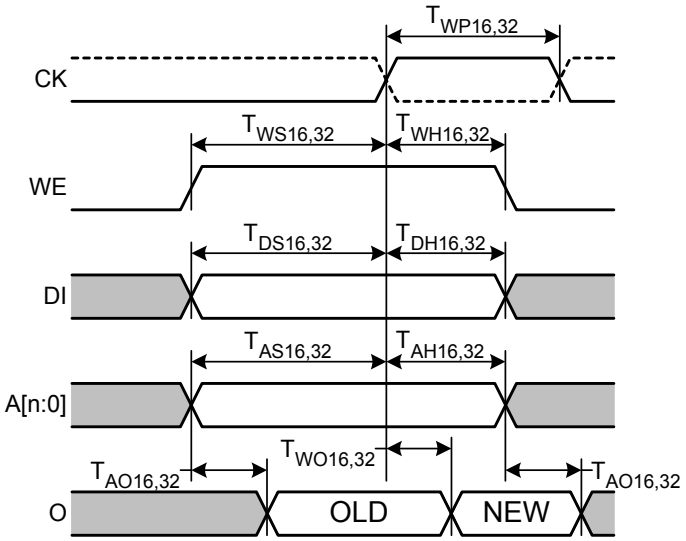


Figure 76. Single-port RAM timing diagram.

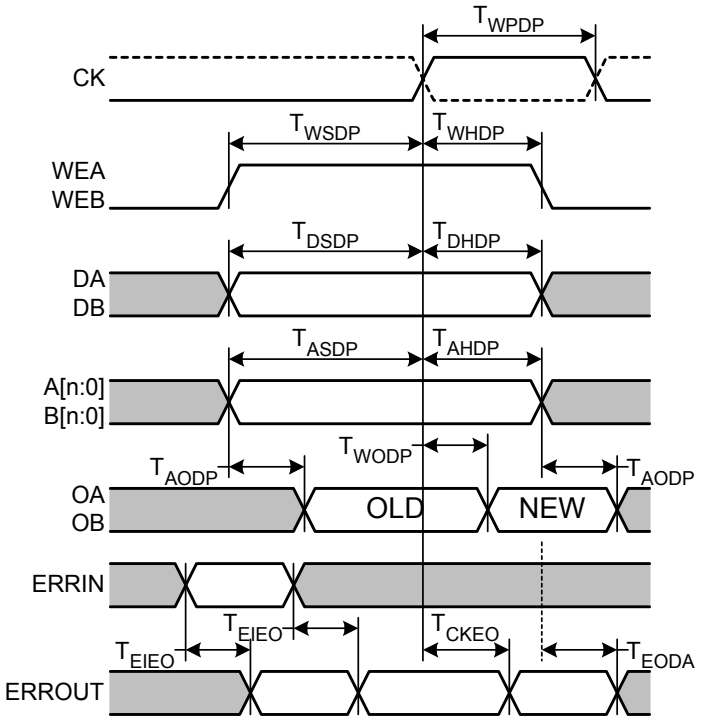


Figure 77. Dual-port RAM timing diagram.

Configurable System Logic (CSL) Cell (Memory Mode, Dual-Port RAM)

In memory mode, two CSL cell in tandem implement a 16x1, edge-triggered, dual-port random-access memory (RAM). The outputs can be captured in a flip-flop. Built-in circuitry flags an error when writing both ports with different data, at the same address.

CSL Memory Mode, Dual-Port RAM Functional Diagram

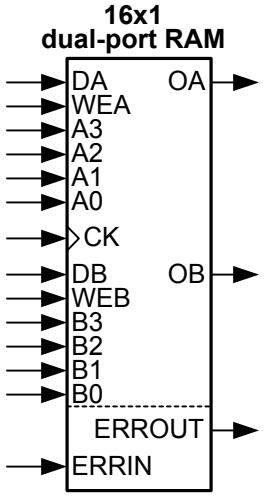


Figure 78. 16x1 dual-port RAM.

CSL Memory Mode, Dual-Port RAM Timing Characteristic Guidelines

The values listed below are representative, guideline values extracted from measured internal test patterns. Actual values may depend on application-specific use. FastChip reports specific, worst-case guaranteed values in the Timing Analysis section of the project report.

All timing values shown assume worst-case operating conditions, including process technology, power supply voltage, and junction temperature. Values include delay driving one interconnect segment.

				Guideline		*Guideline*		
			Speed Grade	*-25*		*-40*		
Description	**Symbol**	**Fig.**	**Size**	**Min**	**Max**	**Min**	**Max**	**Units**
Address write cycle time (clock period)	T_{WCDP}	77,78	16x1	9.6		7.4		ns
CK clock pulse width (active edge)	T_{WPDP}	77,78	16x1	8.1		6.2		ns
Address setup time before CK	T_{ASDP}	77,78	16x1	0.8		0.6		ns
Address hold time after CK	T_{AHDP}	77,78	16x1		0		0	ns
DA, DB setup time before CK	T_{DSDP}	77,78	16x1	0.9		0.7		ns
DA, DB hold time after CK	T_{DHDP}	77,78	16x1		0		0	ns
WEA, WEB setup time before CK	T_{WSDP}	77,78	16x1	1.6		1.2		ns
WEA, WEB hold time after CK	T_{WHDP}	77,78	16x1		0		0	ns
Data valid on OA, OB after CK [1]	T_{WODP}	77,78	16x1		7.0		5.2	ns
Data valid after address change (no Write Enable) [1]	T_{AODP}	77,78	16x1		5.5		3.0	ns
ERRIN to ERROUT valid [1,2]	T_{EIEO}	77,78	16x1		5.1		3.5	ns
ERROUT valid after write data conflict after CK [1,2]	T_{CKEO}	77,78	16x1		8.9		5.8	ns
				Guideline		*Guideline*		

Note 1: Includes output delay driving onto a single interconnect segment, T_{ZIP}

Note 2: If ERROUT output drives ERRIN of adjacent block, subtract output delay, T_{ZIP}, 1.4 ns for -25, 0.7 ns for -40

Configurable System Logic (CSL) Cell (Memory Mode, 8-bit Shift Register)

In memory mode, a CSL cell implements an 8-bit, serial-in/serial-out, preloadable shift register with clock-enable control and tap select control.

CSL Memory Mode, 8-bit Shift Register Functional Diagram

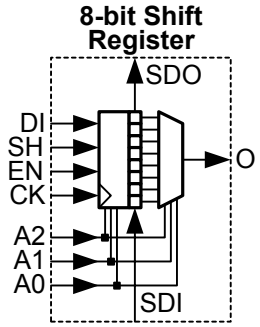


Figure 79. 8-bit Shift Register.

CSL Memory Mode, 8-bit Shift Register Timing Characteristic Guidelines

The values listed below are representative, guideline values extracted from measured internal test patterns. Actual values may depend on application-specific use. FastChip reports specific, worst-case guaranteed values in the Timing Analysis section of the project report.

All timing parameters assume worst-case operating conditions, including process technology, power supply voltage, and junction temperature. Values include delay driving one interconnect segment.

				Guideline		*Guideline*		
			Speed Grade	-25		-40		
Description	**Symbol**	**Fig.**	**Device**	**Min**	**Max**	**Min**	**Max**	**Units**
Combinatorial Delays								
Data valid on output O after address change (no load operation, SH=High) [1]	T_{AOSR}	79	All		4.5		3.0	ns
Sequential Delays								
CSL cell clock input CK to output O [1]	T_{COSR}	79	All		7.0		5.3	ns
CSL cell clock input CK to serial data output SDO [1,2]	T_{CSDSR}	79	All		6.4		5.9	ns
Setup Time before CSL Cell Clock CK								
Data input DI	T_{DSSR}	79	All	0.9		0.7		ns
Shift/load input SH	T_{SSSR}	79	All	0.5		0.4		ns
Clock enable input CE	T_{CSSR}	79	All	0.5		0.4		ns
Address inputs A2 - A0	T_{ASSR}	79	All	0.8		0.6		ns
Serial data input SDI	T_{ISSR}	79	All	1.2		0.9		ns
Hold Time after CSL Cell Clock								
All hold times	T_{HSR}	79	All		0		0	ns
CSL Cell Clock								
Clock High time	T_{CH}	79	All	2.5		2.0		ns
Clock Low time	T_{CL}	79	All	2.5		2.2		ns
				Guideline		*Guideline*		

Note 1: Includes output delay driving onto a single interconnect segment, T_{ZIP}

Note 2: If SDO output drives SDI of adjacent block, subtract output delay, T_{ZIP}, 1.4 ns for -25, 0.7 ns for -40

Bus Clock and Global Buffers

Bus Clock and Global Buffers Functional Diagram

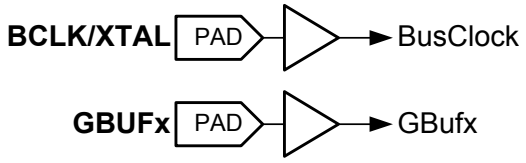


Figure 80. Bus Clock and Global Buffers.

Bus Clock and Global Buffers Timing Characteristic Guidelines

The values listed below are representative, guideline values extracted from measured internal test patterns. Actual values may depend on application-specific use. FastChip reports specific, worst-case guaranteed values in the Timing Analysis section of the project report.

All timing parameters assume worst-case operating conditions, including process technology, power supply voltage, and junction temperature.

				Guideline	*Guideline*	
			Speed Grade	**-25**	**-40**	
Description	**Symbol**	**Fig.**	**Device**	**Maximum**	**Maximum**	**Units**
Bus Clock						
From BCLK/XTAL input through bus clock buffer to any CSL or PIO clock input CK [1]	T_{BCLK}	80	TE502	5.7	4.3	ns
			TE505	5.7	4.3	ns
			TE512	5.7	4.3	ns
			TE520	5.7	4.3	ns
Global Buffers						
From GBUFx input through associated global buffer to any CSL or PIO input [2]	T_{GBUF}	80	TE502	7.1	5.5	ns
			TE505	7.1	5.5	ns
			TE512	7.1	5.5	ns
			TE520	7.1	5.5	ns
				Guideline	*Guideline*	

Note 1: Values for all devices based on data measured on a TE520. The delays for the TE502, TE505, and TE512 are expected to be less than the TE520.

Programmable Input/Output (PIO) Timing Guidelines

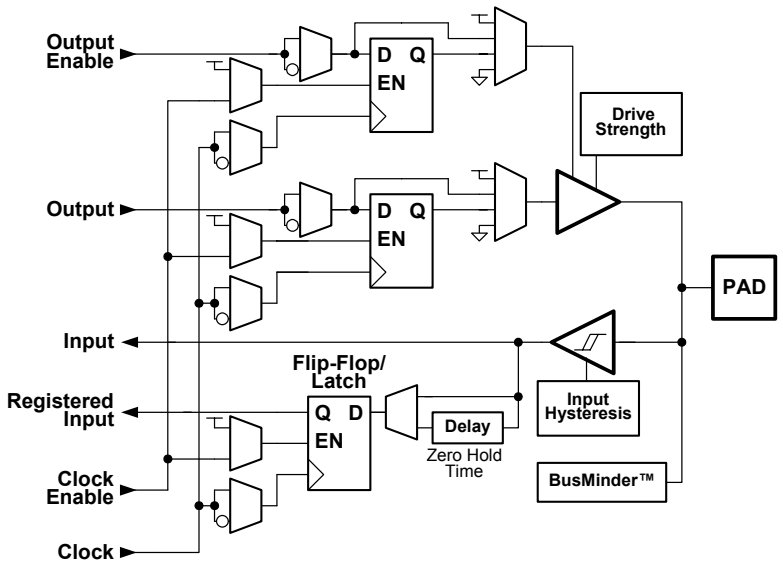


Figure 81. Programmable Input/Output (PIO).

Input Path Characteristics

The values listed below are representative, guideline values extracted from measured internal test patterns. Actual values may depend on application-specific use. FastChip reports specific, worst-case guaranteed values in the Timing Analysis section of the project report.

All timing parameters assume worst-case operating conditions, including process technology, power supply voltage, and junction temperature. Values include delay driving one interconnect segment.

				Guideline		*Guideline*		
			Speed Grade	-25		-40		
Description	**Symbol**	**Fig.**	**Device**	**Min**	**Max**	**Min**	**Max**	**Units**
Propagation Delays								
Pad to input [1]	T_{PID}	81	All		4.2		2.5	ns
Pad to registered input via transparent input latch, no delay [1]	T_{PIL}	81	All		4.5		3.5	ns
PIO clock input on to registered input IQ, flip-flop mode [1]	T_{PICQ}	81	All		4.5		2.9	ns
PIO clock input on to registered input IQ, transparent latch mode [1]	T_{PICL}	81	All		4.4		2.9	ns
Setup Time before PIO clock CK								
Clock enable	T_{PENS}	81	All	2.0		1.5		ns
Pad, zero hold time delay inserted	T_{PPS}	81	All	2.0		1.5		ns
Pad, no delay inserted	T_{PPSN}	81	All	0		0		ns
Hold Time after PIO clock CK								
Pad or clock enable, zero hold time delay inserted	T_{PENH}	81	All		0		0	ns
Pad or clock enable, no delay inserted	T_{PNHN}	81	All		3.1		2.3	ns
PIO Clock								
Clock High time	T_{PCH}	81	All	2.5		1.8		ns
Clock Low time	T_{PCL}	81	All	7.8		5.7		ns
				Guideline		*Guideline*		

Note 1: Includes output delay driving onto a single interconnect segment, T_{ZIP}

Output Path Characteristics

The values listed below are representative, guideline values extracted from measured internal test patterns. Actual values may depend on application-specific use. FastChip reports specific, worst-case guaranteed values in the Timing Analysis section of the project report.

All timing parameters assume worst-case operating conditions, including process technology, power supply voltage, and junction temperature. Values include delay driving one interconnect segment.

				Guideline		*Guideline*		
			Speed Grade	-25		-40		
Description	**Symbol**	**Fig.**	**Device**	**Min**	**Max**	**Min**	**Max**	**Units**
Propagation Delays								
Output to Pad	T_{OP}	81	All		8.7		5.5	ns
Output-enable to Pad high-impedance (Hi-Z)	T_{OPZ}	81	All		12.2		8.5	ns
Output-enable to Pad active and valid	T_{OE}	81	All		9.6		6.4	ns
PIO clock input on output register (OREG) to Pad	T_{KO}	81	All		9.4		6.3	ns
PIO clock input on output-enable register (OEREG) to Pad	T_{KZ}	81	All		15.1		9.9	ns
Setup Time before PIO clock CK								
Output	T_{POSU}	81	All	1.9		1.5		ns
Output-enable	T_{POOE}	81	All	1.9		1.5		ns
Clock enable	T_{POEN}	81	All	2.0		1.5		ns
Hold Time after PIO clock CK								
All hold times	T_{PQH}	81	All		0		0	ns
PIO Clock								
Clock High time	T_{CHIO}	81	All	2.5		2.5		ns
Clock Low time	T_{CLIO}	81	All	2.5		2.5		ns
				Guideline		*Guideline*		

Triscend E5 Power/Current Consumption Characteristics

Typical Power-Down Current Consumption (3.3 volts, room temperature)

Table 47 shows typical power-down current consumption, measured on various devices, with 3.3 volts applied to VCC and VCCIO, at room temperature. Devices under test were placed in the minimum power consumption mode. All power-down options selected.

Table 47. Typical Power-Down Consumption.

Device	Power-Down Current
TE502	5 µA
TE505	5 µA
TE512	20 µA
TE520	20 µA

Worst-Case Power-Down Current Consumption Characteristics

Figure 82 and Figure 83 shows the power-down current consumption measured on corner-lot devices at different voltages. The characterized devices were processed under worst-case parametric conditions. Devices were placed in the minimum power consumption mode. All power-down options were selected.

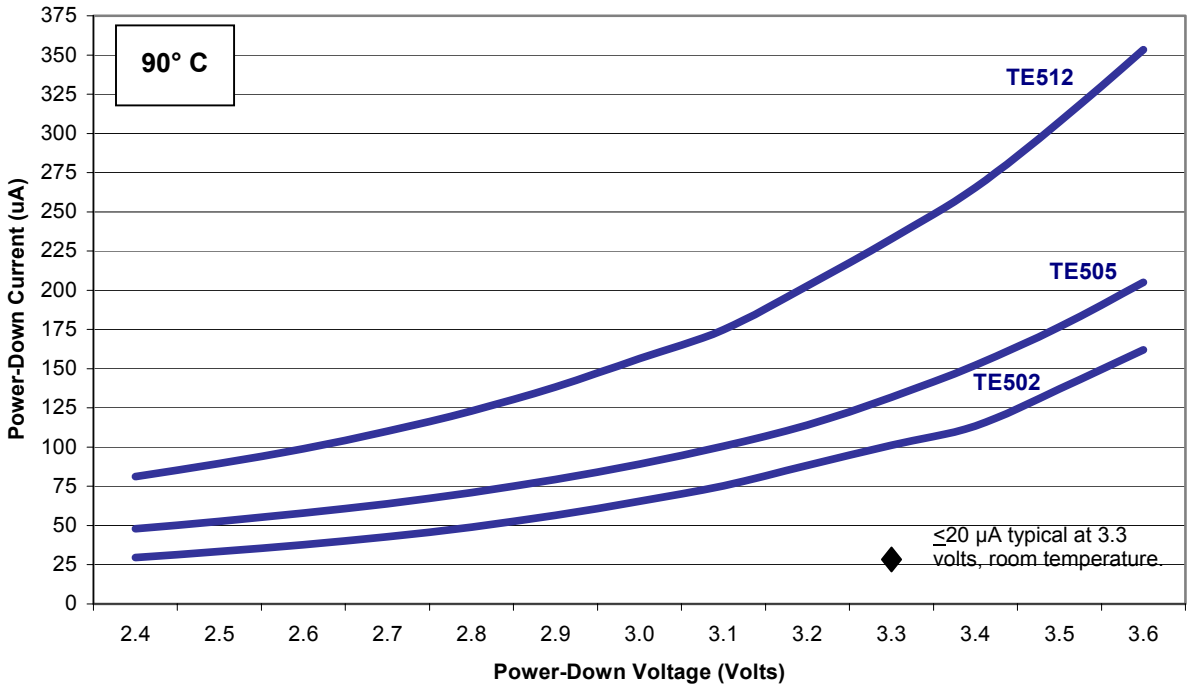


Figure 82. Worst-case power down current consumption characteristics for the TE502, TE505, and TE512, measured at 90° C.

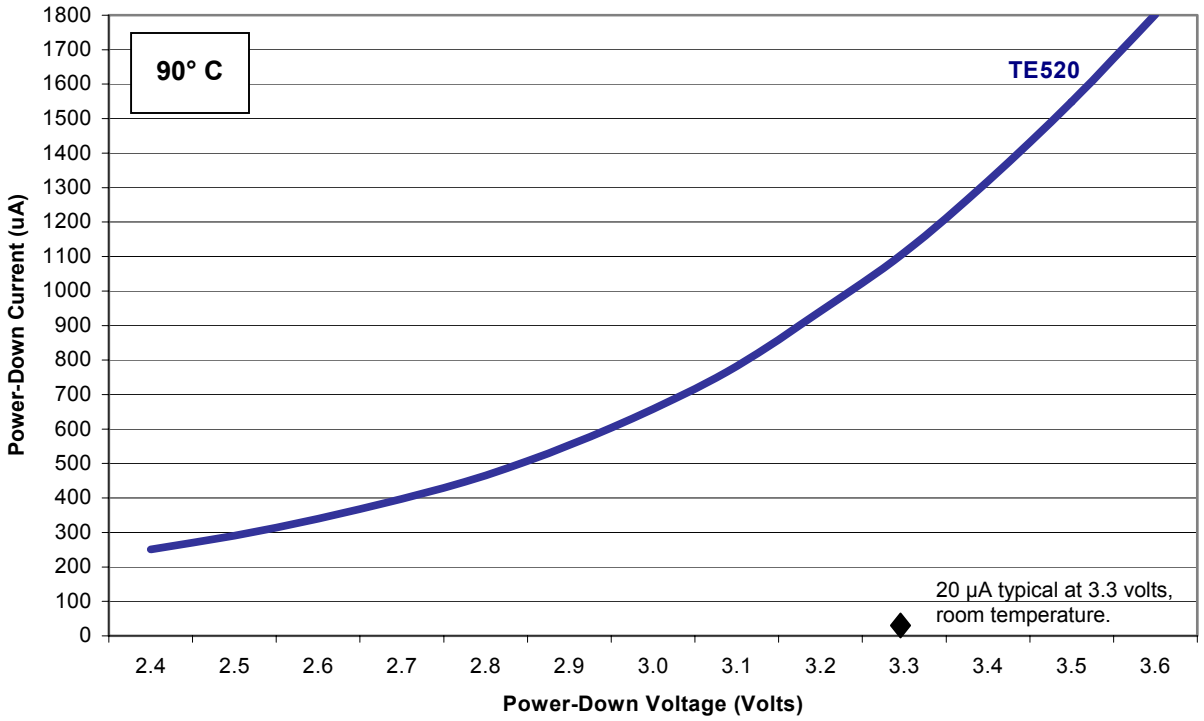


Figure 83. Worst-case power down current consumption characteristics for the TE520, measured at 90° C.

Typical Dynamic Power Consumption Component Estimates

This information should only be used to provide a rough power estimate for a design. This data does not substitute for actual power measurements from the finished E5 application. See Application Note 13 (AN-13), "***Estimating Triscend E5 Power Consumption***" for additional information.

The total dynamic power consumption consists of various components, including power-down mode, power consumed by the microcontroller sub-system, power consumed by the application operating in the configurable system logic (CSL), and the power consumed by the programmable input/output (PIO) pins.

Microcontroller Sub-system Component

The microcontroller sub-system represents all of the dedicated resources of the Triscend E5 device.

- The Accelerated 8051 microcontroller and its peripherals.
- The Memory Interface Unit (MIU) to access external Flash containing initialization data and the 8051 application code. The power consumed by the external memory is not included.
- The Configurable System Interconnect (CSI) bus logic, including the bus socket logic driving into the CSL matrix and the internal sys-tem RAM.
- The Bus Clock distribution network, which drives all the dedicated resources, the CSI bus, and is distributed into the CSL matrix.
- Various other circuits including the power-on reset (POR) circuitry, the internal ring oscillator, the internal crystal oscillator amplifier, etc.

The MCU sub-system is by far the largest power consumer on an E5 device. The size of the CSL matrix grows with the larger-density devices along with the size of the Bus Clock distribution network. Consequently, the larger-density devices consume more power.

If comparing against the original 8051microcontroller, consider that the accelerated 8051 microcontroller embedded within an E5 Customizable Microcontroller device has roughly three times more performance operating at the same frequency. Consequently, the accelerated 8051 microcontroller within an E5 de-

vice can operate at one-third the clock frequency while maintaining the same overall performance as the original 8051 design.

$$\text{MCU\_Power}_{(3.6\text{ volts}, 70^{O}\text{C, typical})} = \text{Dynamic} \bullet \text{Frequency} + 20\text{mW}$$

Dynamic is the microcontroller sub-system power consumption from Table 48.

Frequency is the bus clock frequency in MHz.

Table 48. Typical microcontroller sub-system dynamic power consumption at 3.6 volts, 70° C.

Device	Dynamic Power Consumption	Units
TE502*	4.5	mW/MHz
TE505*	6.7	
TE512*	12.7	
TE520	18.0	

*estimate, based on measured data from TE520.

CSL Power Component

The power consumed by the configurable system logic (CSL) portion of the Customizable Microcontroller device is the sum of all the CSL cells utilized in the application. Not every CSL cell switches on every clock cycle. The switch factor, ***k***, describes the switching activity for the CSL cell. In a typical application, only about 5% to 30% of the cells switch on a given clock edge. The Frequency is the clock frequency at the CSL cell clock input or the frequency at which a group of CSL cells changes, in MHz. It is easiest to compute the total CSL power component by examining the individual clock domains within an application.

$$\text{CSL\_Power}_{(3.6\text{ volts}, 70^{O}\text{C, typical})} = \sum k \bullet 0.15\text{mW/MHz} \bullet \text{Frequency}$$

PIO Power Component

The power consumed by the programmable input/output (PIO) cells on the Customizable Microcontroller device is the sum of all the PIO cells that are configured as outputs utilized in the application. Not every PIO cell switches on every clock cycle. The switch factor, ***k***, describes the switching activity for the PIO cell. In a typical application, only about 5% to 30% of the cells switch on a given clock edge. The Frequency is the frequency at which a group of PIO cells changes, in MHz.. It is easiest to compute the total PIO power component by examining the individual clock domains within an application. The PIO dynamic power depends on whether the PIO drives 4 mA or 12 mA, as shown in Table 49.

$$\text{PIO\_Power}_{(3.6\text{ volts}, 70^{O}\text{C, typical})} = \sum k \bullet \text{PIO\_Dynamic} \bullet \text{Frequency}$$

Table 49. Typical PIO output pin power consumption at 3.6 volts, 70° C.

Drive Strength	Dynamic Power Consumption	Units
4 mA	1.3	mW/MHz
12 mA	2.6	

Output Buffer Switching Characteristics

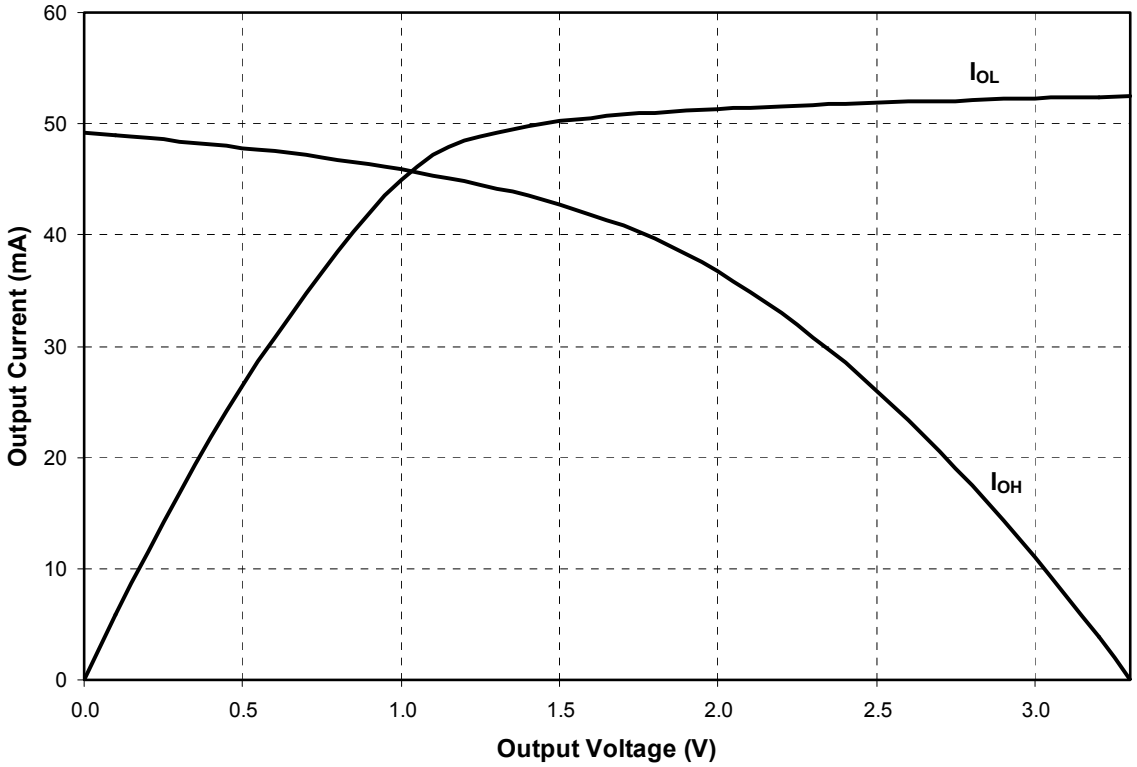


Figure 84. 12 mA output buffer characteristics (T_J= 25ºC, V_{CCIO}=3.3 Volts), HSpice simulation.

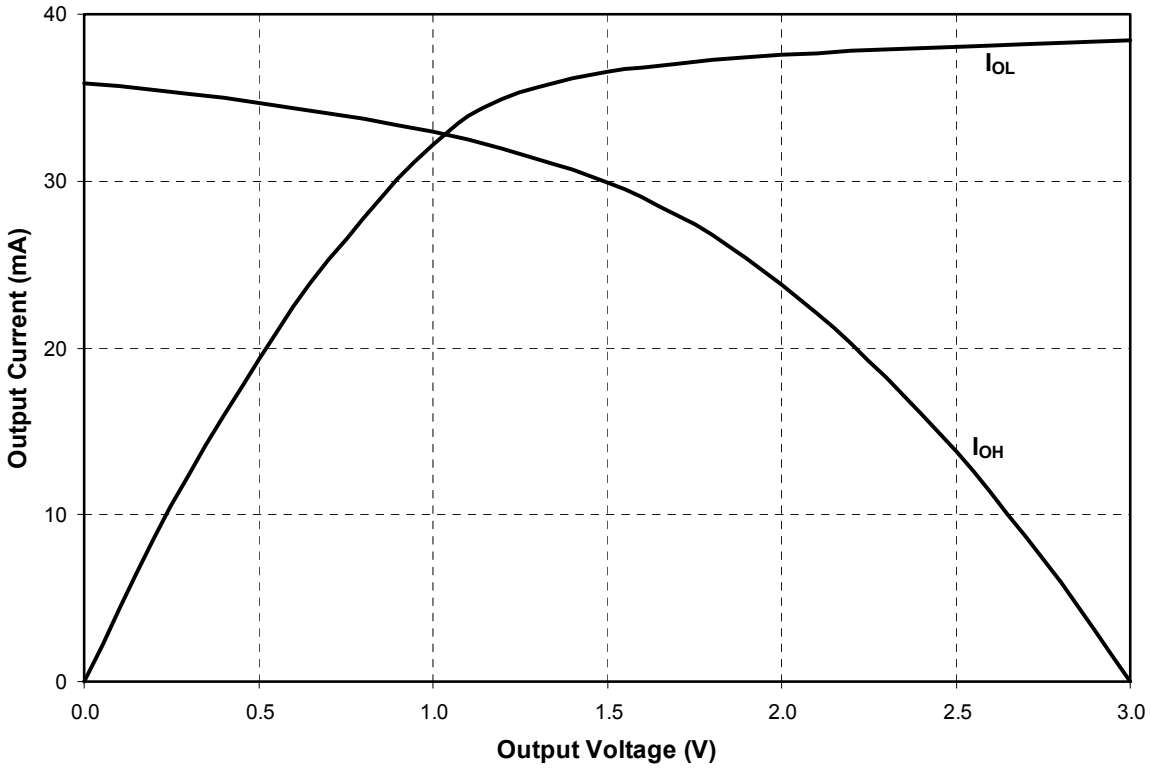


Figure 85. 12 mA output buffer characteristics (T_J= 85ºC, V_{CCIO}=3.0 Volts), HSpice simulation.

Appendix A: E5 Instruction Timing

Table 50 shows the instruction timing for the E5's embedded and accelerated 8051 microcontroller. The table also shows the performance of the E5 versus the original 8051 and the 80C320. All E5 instruction fetches are affected by the read timing settings in the Memory Interface Unit. The "E5 Clock Cycles" values in the table below assume an external memory device fast enough to perform a four-cycle fetch at the bus clock frequency, without wait-states. The number of E5 machine cycles is not affected.

All E5 instruction timing is potentially affected by wait-states on the bus. Wait-states may affect the number of E5 clock cycles required to complete an instruction. The number of machine cycles is not affected. The original 8051 did not support wait-states.

The E5 supports an op-code 'A5' instruction, which was not available on the original 8032. This is a software breakpoint function. The timing for this E5 instruction has not been characterized.

Table 50. E5 Instruction Timing.

Instruction	Hex Code	Bytes	Machine Cycles	E5 Clock Cycles	8051 Clock Cycles	E5 vs. 8051 Speed Ratio	80C320 Clock Cycles	E5 vs. 80C320 Speed Ratio
NOP	00	1	1	4	12	3	4	1
ADD A, RO	28	1	1	4	12	3	4	1
ADD A, R1	29	1	1	4	12	3	4	1
ADD A, R2	2A	1	1	4	12	3	4	1
ADD A, R3	2B	1	1	4	12	3	4	1
ADD A, R4	2C	1	1	4	12	3	4	1
ADD A, R5	2D	1	1	4	12	3	4	1
ADD A, R6	2E	1	1	4	12	3	4	1
ADD A, R7	2F	1	1	4	12	3	4	1
ADD A, @RO	26	1	1	4	12	3	4	1
ADD A, @R1	27	1	1	4	12	3	4	1
ADD A, direct	25	2	2	8	12	1.5	8	1
ADD A, #data	24	2	2	8	12	1.5	8	1
ADDC A, RO	38	1	1	4	12	3	4	1
ADDC A, R1	39	1	1	4	12	3	4	1
ADDC A, R2	3A	1	1	4	12	3	4	1
ADDC A, R3	3B	1	1	4	12	3	4	1
ADDC A, R4	3C	1	1	4	12	3	4	1
ADDC A, R5	3D	1	1	4	12	3	4	1
ADDC A, R6	3E	1	1	4	12	3	4	1
ADDC A, R7	3F	1	1	4	12	3	4	1
ADDC A, @R0	36	1	1	4	12	3	4	1
ADDC A, @R1	37	1	1	4	12	3	4	1
ADDC A, direct	35	2	2	8	12	1.5	8	1
ADDC A, #data	34	2	2	8	12	1.5	8	1
SUBB A, R0	48	1	1	4	12	3	4	1
SUBB A, R1	49	1	1	4	12	3	4	1
SUBB A, R2	4A	1	1	4	12	3	4	1
SUBB A, R3	4B	1	1	4	12	3	4	1
SUBB A, R4	4C	1	1	4	12	3	4	1
SUBB A, R5	4D	1	1	4	12	3	4	1
SUBB A, R6	4E	1	1	4	12	3	4	1
SUBB A, R7	4F	1	1	4	12	3	4	1
SUBB A, @R0	46	1	1	4	12	3	4	1
SUBB A, @R1	47	1	1	4	12	3	4	1
SUBB A, direct	45	2	2	8	12	1.5	8	1
SUBB A, #data	44	2	2	8	12	1.5	8	1
INC A	04	1	1	4	12	3	4	1
INC R0	08	1	1	4	12	3	4	1
INC R1	09	1	1	4	12	3	4	1
INC R2	0A	1	1	4	12	3	4	1
INC R3	0B	1	1	4	12	3	4	1

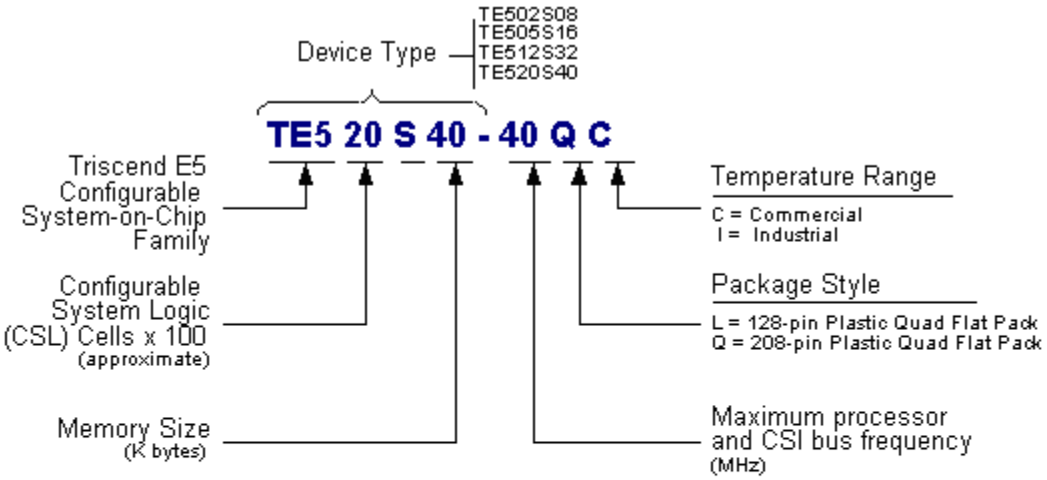
Instruction	Hex Code	Bytes	Machine Cycles	E5 Clock Cycles	8051 Clock Cycles	E5 vs. 8051 Speed Ratio	80C320 Clock Cycles	E5 vs. 80C320 Speed Ratio
INC R4	0C	1	1	4	12	3	4	1
INC R5	0D	1	1	4	12	3	4	1
INC R6	0E	1	1	4	12	3	4	1
INC R7	0F	1	1	4	12	3	4	1
INC @R0	06	1	1	4	12	3	4	1
INC @R1	07	1	1	4	12	3	4	1
INC direct	05	2	2	8	12	1.5	8	1
INC DPTR	A3	1	2	8	24	3	12	1.5
DEC A	14	1	1	4	12	3	4	1
DEC R0	18	1	1	4	12	3	4	1
DEC R1	19	1	1	4	12	3	4	1
DEC R2	1A	1	1	4	12	3	4	1
DEC R3	1B	1	1	4	12	3	4	1
DEC R4	1C	1	1	4	12	3	4	1
DEC R5	1D	1	1	4	12	3	4	1
DEC R6	1E	1	1	4	12	3	4	1
DEC R7	1F	1	1	4	12	3	4	1
DEC @R0	16	1	1	4	12	3	4	1
DEC @R1	17	1	1	4	12	3	4	1
DEC direct	15	2	2	8	12	1.5	8	1
Software Breakpoint	A5	1	?	?	-	-	-	1
MUL AB	A4	1	5	20	48	2.4	20	1
DIV AB	84	1	5	20	48	2.4	20	1
DA A	D4	1	1	4	12	3	4	1
ANL A, RO	58	1	1	4	12	3	4	1
ANL A, R1	59	1	1	4	12	3	4	1
ANL A, R2	5A	1	1	4	12	3	4	1
ANL A, R3	5B	1	1	4	12	3	4	1
ANL A, R4	5C	1	1	4	12	3	4	1
ANL A, R5	5D	1	1	4	12	3	4	1
ANL A, R6	5E	1	1	4	12	3	4	1
ANL A, R7	5F	1	1	4	12	3	4	1
ANL A, @R0	56	1	1	4	12	3	4	1
ANL A, @R1	57	1	1	4	12	3	4	1
ANL A, direct	55	2	2	8	12	1.5	8	1
ANL A, #data	54	2	2	8	12	1.5	8	1
ANL direct A	52	2	2	8	12	1.5	8	1
ANL direct, #data	53	3	3	12	24	2	12	1
ORL A, R0	48	1	1	4	12	3	4	1
ORL A, R1	49	1	1	4	12	3	4	1
ORL A, R2	4A	1	1	4	12	3	4	1
ORL A, R3	4B	1	1	4	12	3	4	1
ORL A, R4	4C	1	1	4	12	3	4	1
ORL A, R5	4D	1	1	4	12	3	4	1
ORL A, R6	4E	1	1	4	12	3	4	1
ORL A, R7	4F	1	1	4	12	3	4	1
ORL A, R7	4F	1	1	4	12	3	4	1
ORL A, @R0	46	1	1	4	12	3	4	1
ORL A, @R1	47	1	1	4	12	3	4	1
ORL A, direct	45	2	2	8	12	1.5	8	1
ORL A, #data	44	2	2	8	12	1.5	8	1
ORL direct, A	42	2	2	8	12	1.5	8	1
ORL direct, #data	43	3	3	12	24	2	12	1
XRL A, R0	68	1	1	4	12	3	4	1
XRL A, R1	69	1	1	4	12	3	4	1
XRL A, R2	6A	1	1	4	12	3	4	1

Instruction	Hex Code	Bytes	Machine Cycles	E5 Clock Cycles	8051 Clock Cycles	E5 vs. 8051 Speed Ratio	80C320 Clock Cycles	E5 vs. 80C320 Speed Ratio
XRL A, R3	6B	1	1	4	12	3	4	1
XRL A, R4	6C	1	1	4	12	3	4	1
XRL A, R5	6D	1	1	4	12	3	4	1
XRL A, R6	6E	1	1	4	12	3	4	1
XRL A, R7	6F	1	1	4	12	3	4	1
XRL A, @R0	66	1	1	4	12	3	4	1
XRL A, @R1	67	1	1	4	12	3	4	1
XRL A, direct	65	2	2	8	12	1.5	8	1
XRL A, #data	64	2	2	8	12	1.5	8	1
XRL direct, A	62	2	2	8	12	1.5	8	1
XRL direct, #data	63	3	3	12	24	2	12	1
CLR A	E4	1	1	4	12	3	4	1
CPL A	F4	1	1	4	12	3	4	1
RL A	23	1	1	4	12	3	4	1
RLC A	33	1	1	4	12	3	4	1
RR A	03	1	1	4	12	3	4	1
RRC A	13	1	1	4	12	3	4	1
SWAP A	C4	1	1	4	12	3	4	1
MOV A, R0	E8	1	1	4	12	3	4	1
MOV A, R1	E9	1	1	4	12	3	4	1
MOV A, R2	EA	1	1	4	12	3	4	1
MOV A, R3	EB	1	1	4	12	3	4	1
MOV A, R4	EC	1	1	4	12	3	4	1
MOV A, R5	ED	1	1	4	12	3	4	1
MOV A, R6	EE	1	1	4	12	3	4	1
MOV A, R7	EF	1	1	4	12	3	4	1
MOV A, @R0	E6	1	1	4	12	3	4	1
MOV A, @R1	E7	1	1	4	12	3	4	1
MOV A, direct	E5	2	2	8	12	1.5	8	1
MOV A, #data	74	2	2	8	12	1.5	8	1
MOV RO, A	F8	1	1	4	12	3	4	1
MOV R1, A	F9	1	1	4	12	3	4	1
MOV R2, A	FA	1	1	4	12	3	4	1
MOV R3, A	FB	1	1	4	12	3	4	1
MOV R4, A	FC	1	1	4	12	3	4	1
MOV R5, A	FD	1	1	4	12	3	4	1
MOV R6, A	FE	1	1	4	12	3	4	1
MOV R7, A	FF	1	1	4	12	3	4	1
MOV R0, direct	A8	2	2	8	12	1.5	8	1
MOV R1, direct	A9	2	2	8	12	1.5	8	1
MOV R2, direct	AA	2	2	8	12	1.5	8	1
MOV R3, direct	AB	2	2	8	12	1.5	8	1
MOV R4, direct	AC	2	2	8	12	1.5	8	1
MOV R5, direct	AD	2	2	8	12	1.5	8	1
MOV R6, direct	AE	2	2	8	12	1.5	8	1
MOV R7, direct	AF	2	2	8	12	1.5	8	1
MOV RO, #data	78	2	2	8	12	1.5	8	1
MOV RO, #data	78	2	2	8	12	1.5	8	1
MOV R1, #data	79	2	2	8	12	1.5	8	1
MOV R2, #data	7A	2	2	8	12	1.5	8	1
MOV R3, #data	7B	2	2	8	12	1.5	8	1
MOV R4, #data	7C	2	2	8	12	1.5	8	1
MOV R5, #data	7D	2	2	8	12	1.5	8	1
MOV R6, #data	7E	2	2	8	12	1.5	8	1
MOV R7, #data	7F	2	2	8	12	1.5	8	1
MOV @RO, A	F6	1	1	4	12	3	4	1

Instruction	Hex Code	Bytes	Machine Cycles	E5 Clock Cycles	8051 Clock Cycles	E5 vs. 8051 Speed Ratio	80C320 Clock Cycles	E5 vs. 80C320 Speed Ratio
MOV @R1, A	F7	1	1	4	12	3	4	1
MOV @RO, direct	A6	2	2	8	12	1.5	8	1
MOV @R1, direct	A7	2	2	8	12	1.5	8	1
MOV @RO, #data	76	2	2	8	12	1.5	8	1
MOV @R1, #data	77	2	2	8	12	1.5	8	1
MOV direct, A	F5	2	2	8	12	1.5	8	1
MOV direct, RO	88	2	2	8	12	1.5	8	1
MOV direct, R1	89	2	2	8	12	1.5	8	1
MOV direct, R2	8A	2	2	8	12	1.5	8	1
MOV direct, R3	8B	2	2	8	12	1.5	8	1
MOV direct, R4	8C	2	2	8	12	1.5	8	1
MOV direct, R5	8D	2	2	8	12	1.5	8	1
MOV direct, R6	8E	2	2	8	12	1.5	8	1
MOV direct, R7	8F	2	2	8	12	1.5	8	1
MOV direct, @R0	86	2	2	8	12	1.5	8	1
MOV direct, @R1	87	2	2	8	12	1.5	8	1
MOV direct, direct	85	3	3	12	24	2	12	1
MOV direct, #data	75	3	3	12	24	2	12	1
MOV DPTR, #data 16	90	3	3	12	24	2	12	1
MOVC A, @A+DPTR	93	1	2	8	24	3	12	1.5
MOVC A, @A+PC	83	1	2	8	24	3	12	1
MOVX A, @R0	E2	1	2	8	24	3	8-36*	1
MOVX A, @R1	E3	1	2	8	24	3	8-36*	1
MOVX A, @DPTR	E0	1	2	8	24	3	8-36*	1
MOVX @RO, A	F2	1	2	8	24	3	8-36*	1
MOVX @R1, A	F3	1	2	8	24	3	8-36*	1
MOVX @ DPTR, A	F0	1	2	8	24	3	8-36*	1
PUSH direct	C0	2	2	8	24	3	8	1
POP direct	D0	2	2	8	24	3	8	1
XCH A, R0	C8	1	1	4	12	3	4	1
XCH A, R1	C9	1	1	4	12	3	4	1
XCH A, R2	CA	1	1	4	12	3	4	1
XCH A, R3	CB	1	1	4	12	3	4	1
XCH A, R4	CC	1	1	4	12	3	4	1
XCH A, R5	CD	1	1	4	12	3	4	1
XCH A, R6	CE	1	1	4	12	3	4	1
XCH A, R7	CF	1	1	4	12	3	4	1
XCH A, @R0	C6	1	1	4	12	3	4	1
XCH A, @R1	C7	1	1	4	12	3	4	1
XCHD A @R0	D6	1	1	4	12	3	4	1
XCHD A @R1	D7	1	1	4	12	3	4	1
XCH A, direct	C5	2	2	8	12	1.5	8	1
CLR C	C3	1	1	4	12	3	4	1
CLR bit	C2	2	2	8	12	1.5	8	1
SETB C	D3	1	1	4	12	3	4	1
SETB bit	D2	2	1	4	12	3	8	1
CPL C	B3	1	1	4	12	3	4	1
CPL bit	B2	2	2	8	12	1.5	8	1
ANL C, bit	82	2	2	8	24	3	8	1
ANL C, /bit	B0	2	2	8	24	3	8	1
ORL C, bit	72	2	2	8	24	3	8	1
ORL C, /bit	A0	2	2	8	24	3	8	1
MOV C, bit	A2	2	2	8	12	1.5	8	1
MOV bit, C	92	2	2	8	24	3	8	1
ACALL addr11	11,31, 51,71, 91,B1,	2	3	12	24	2	12	1

Instruction	Hex Code	Bytes	Machine Cycles	E5 Clock Cycles	8051 Clock Cycles	E5 vs. 8051 Speed Ratio	80C320 Clock Cycles	E5 vs. 80C320 Speed Ratio
	D1,F1							
LCALL addr16	12	3	4	16	24	1.5	16	1
RET	22	1	2	8	24	3	16	1
RETI	32	1	2	8	24	3	16	1
AJMP ADDR11	01,21, 41,61, 81,A1, C1,E1	2	3	12	24	2	12	1
LJMP addr16	02	3	4	16	24	1.5	16	1
JMP @A+DPTR	73	1	2	8	24	3	12	1.5
SJMP rel	80	2	3	12	24	2	12	1
JZ rel	60	2	3	12	24	2	12	1
JNZ rel	70	2	3	12	24	2	12	1
JC rel	40	2	3	12	24	2	12	1
JNC rel	50	2	3	12	24	2	12	1
JB bit, rel	20	3	4	16	24	1.5	16	1
JBC bit, rel	30	3	4	16	24	1.5	16	1
JBC bit, rel	10	3	4	16	24	1.5	16	1
CJNE A, direct, rel	B5	3	4	16	24	1.5	16	1
CJNE A, #data, rel	B4	3	4	16	24	1.5	16	1
CJNE @RO, #data, rel	B6	3	4	16	24	1.5	16	1
CJNE @R1, #data, rel	B7	3	4	16	24	1.5	16	1
CJNE RO, #data, rel	B8	3	4	16	24	1.5	16	1
CJNE R1, #data, rel	B9	3	4	16	24	1.5	16	1
CJNE R2, #data, rel	BA	3	4	16	24	1.5	16	1
CJNE R3, #data, rel	BB	3	4	16	24	1.5	16	1
CJNE R4, #data, rel	BC	3	4	16	24	1.5	16	1
CJNE R5, #data, rel	BD	3	4	16	24	1.5	16	1
CJNE R6, #data, rel	BE	3	4	16	24	1.5	16	1
CJNE R7, #data, rel	BF	3	4	16	24	1.5	16	1
DJNZ RO, rel	D8	2	3	12	24	2	12	1
DJNZ R1, rel	D9	2	3	12	24	2	12	1
DJNZ R2, rel	DA	2	3	12	24	2	12	1
DJNZ R3, rel	DB	2	3	12	24	2	12	1
DJNZ R4, rel	DC	2	3	12	24	2	12	1
DJNZ R5, rel	DD	2	3	12	24	2	12	1
DJNZ R6, rel	DE	2	3	12	24	2	12	1
DJNZ R7, rel	DF	2	3	12	24	2	12	1
DJNZ direct, rel	D5	3	4	16	24	1.5	16	1

Ordering Information



Sales Offices

Headquarters

Triscend Corporation
301 North Whisman Road
Mountain View, CA 94043
USA

Tel: **1-650-968-8668**
Fax: 1-650-934-9393

Sales Representatives

Visit the Triscend web site for the Triscend sales representative in your area.

Web: **www.triscend.com**

Triscend

Triscend Corporation
301 N. Whisman Rd.
Mountain View, CA 94043-3969
Tel: 1-650-968-8668
Fax: 1-650-934-9393

E-mail: **info@Triscend.com**
Web: **www.triscend.com**

U.S. Distribution

All American Semiconductor
Web: **www.allamerican.com**

Unique Technologies
Tel: **1-800-556-0225**
Web:
www.unique-technologies.com/

Japan

Internix Incorporated
Tel: **(03) 3369-1101**
Fax: (03) 3366-8566
Web: **www.internix.co.jp**

Europe,Middle-East

Unique
Web:
www.unique.memec.com

Asia-Pacific, South Africa

Unique
Web: **www.unique.memec-asiapacific.com**

Triscend E5 ustomizable Microcontroller Family

CONTENTS