
**Welcome to E-XFL.COM**

## Understanding Embedded - DSP (Digital Signal Processors)

Embedded - DSP (Digital Signal Processors) are specialized microprocessors designed to perform complex mathematical computations on digital signals in real-time. Unlike general-purpose processors, DSPs are optimized for high-speed numeric processing tasks, making them ideal for applications that require efficient and precise manipulation of digital data. These processors are fundamental in converting and processing signals in various forms, including audio, video, and communication signals, ensuring that data is accurately interpreted and utilized in embedded systems.

## Applications of Embedded - DSP (Digital Signal Processors)

| Details | |
|---|---|
| Product Status | Obsolete |
| Type | SC140 Core |
| Interface | DSI, Ethernet, RS-232 |
| Clock Rate | 500MHz |
| Non-Volatile Memory | External |
| On-Chip RAM | 1.436MB |
| Voltage - I/O | 3.30V |
| Voltage - Core | 1.20V |
| Operating Temperature | 0°C ~ 90°C (TJ) |
| Mounting Type | Surface Mount |
| Package / Case | 431-BFBGA, FCBGA |
| Supplier Device Package | 431-FCPBGA (20x20) |
| Purchase URL | https://www.e-xfl.com/product-detail/nxp-semiconductors/kmsc8126vt8000 |

**Table 3.** Buses and Memory Controller

| Feature | Description |
|---|---|
| **60x-Compatible System Bus** | • 64/32-bit data and 32-bit address 60x bus.<br>• Support for multiple-master designs.<br>• Four-beat burst transfers (eight-beat in 32-bit wide mode).<br>• Port size of 64, 32, 16, and 8 controlled by the internal memory controller.<br>• Bus can access external memory expansion or off-device peripherals, or it can enable an external host device to access internal resources.<br>• Slave support, direct access by an external host to internal resources including the M1 and M2 memories.<br>• On-device arbitration between up to four master devices. |
| **Direct Slave Interface (DSI)** | A 32/64-bit wide slave host interface that operates only as a slave device under the control of an external host processor.<br>• 21–25 bit address, 32/64-bit data.<br>• Direct access by an external host to internal and external resources, including the M1 and the M2 memories as well as external devices on the system bus.<br>• Synchronous and asynchronous accesses, with burst capability in the synchronous mode.<br>• Dual or single-strobe modes.<br>• Write and read buffers improve host bandwidth.<br>• Byte enable signals enables 1, 2, 4, and 8 byte write access granularity.<br>• Sliding window mode enables access with reduced number of address pins.<br>• Chip ID decoding enables using one $\overline{CS}$ signal for multiple DSPs.<br>• Broadcast $\overline{CS}$ signal enables parallel write to multiple DSPs.<br>• Big-endian, little-endian, and munged little-endian support. |
| **3-Mode Signal Multiplexing** | • 64-bit DSI, 32-bit system bus.<br>• 32-bit DSI, 64-bit system bus.<br>• 32-bit DSI, 32-bit system bus. |
| **Memory Controller** | Flexible eight-bank memory controller:<br>• Three user-programmable machines (UPMs), general-purpose chip-select machine (GPCM), and a page-mode SDRAM machine.<br>• Glueless interface to SRAM, page mode SDRAM, DRAM, EPROM, Flash memory, and other user-definable peripherals.<br>• Byte enables for either 64-bit or 32-bit bus width mode.<br>• Eight external memory banks (banks 0–7). Two additional memory banks (banks 9, 11) control IPBus peripherals and internal memories. Each bank has the following features:<br>—32-bit address decoding with programmable mask.<br>—Variable block sizes (32 KB to 4 GB).<br>—Selectable memory controller machine.<br>—Two types of data errors check/correction: normal odd/even parity and read-modify-write (RMW) odd/even parity for single accesses.<br>—Write-protection capability.<br>—Control signal generation machine selection on a per-bank basis.<br>—Support for internal or external masters on the system bus.<br>—Data buffer controls activated on a per-bank basis.<br>—Atomic operation.<br>—RMW data parity check (on system bus only).<br>—Extensive external memory-controller/bus-slave support.<br>—Parity byte select pin, which enables a fast, glueless connection to RMW-parity devices (on the system bus only).<br>—Data pipeline to reduce data set-up time for synchronous devices. |

**Table 7.** Coprocessors

| Feature | Description |
|---|---|
| **VCOP** | • Fully programmable feed-forward channel decoding, feed-forward channel equalization and traceback sessions.<br>• Up to 400 3GPP 12.2kbps AMR channels (channel decoding, number of channels linear to frequency).<br>• Up to 200 blind transport format detect (BTFD) channels according to the 3GPP standard. Number of channels linear to frequency.<br>• For channel decoding:<br>—Constraint length between K = 5 and K = 9.<br>—Puncture Codes.<br>—Rate 1/2, 1/3, 1/4 and 1/6.<br>—Four fully programmable polynomials (rate 1/6 is implemented by three polynomials only).<br>—History buffer with up to 768 stages for 3G standards.<br>—Input symbols are 8-bit (256 levels) signed soft symbols.<br>—Output is hard decision (1-bit).<br>• For GSM channel equalization:<br>—Fully programmable 4 to 6 estimated channel autocorrelation coefficients (S-Parameters).<br>—History buffer with up to 4090 stages for GSM.<br>—Matched filter input is 8-bit (256 levels).<br>—SOVA assist algorithm.<br>—Output 8-bit coded delta values for SOVA assist algorithm, 1-bit hard decision traceback and history buffer or recursive traceback.<br>• Fully programmable block length for all sessions.<br>• Programmable traceback methods of Max Path, Min Path or End State.<br>• Programmable learning period length for the traceback session.<br>• Supports the start of feed-forward according to a presaved PM memory content. However the history buffer is not saved. Therefore the traceback is according to the current block only.<br>• Each SC140 can program the VCOP parameters while the VCOP is in IDLE mode and then the VCOP can run independently on the whole block of data.<br>• Dumping path metrics to the internal memory on up to 12 predefined stages; this is needed for BTFD applications.<br>• Interrupt lines and status bits notify the cores on session completion.<br>• Performance monitoring unit with 6 monitored behaviors. |
| **TCOP** | • Full support of 3GPP and CDMA2000 standards in Turbo decode.<br>• Up to 20 turbo-coding 384 kbps channels.<br>• 8 state PCCC with polynomial as supported by the 3G standards.<br>• Iterative decoding structure based on Maximum A-Posteriori probability (MAP), with calculations performed in the LOG domain.<br>• Encoding rate of 1/2, 1/3, 1/4, 1/5 with programmable puncturing for the parity symbols.<br>• Full flexibility interleave function via a look-up table.<br>• Flexible block size (1–32767 bits).<br>• MAX log MAP and log MAP (MAX*) approximation.<br>• Programmable MAX* using linear approximation.<br>• Programmable number of iterations, with resolution of half iteration (one MAP).<br>• Fully automatic execution when the GO command executes.<br>• High data rates (for multi-channel systems or multiple channel accumulating to high data rates).<br>• Can stop processing after every MAP when soft lambda all reach a programmable quality threshold.<br>• Minimum and maximum number of iterations to execute in conjunction with the stop criteria.<br>• The SC140 core or host can stop the processing after every MAP during run time.<br>• Automatic, internal normalization for $\alpha$, $\beta$ overflow handling, with zero overhead.<br>• Automatic, internal $\Lambda$ clipping for $\Lambda$ overflow handling, with zero overhead.<br>• Additional least significant bit in $\alpha$, $\beta$, $\gamma$ arithmetic guarding against precision loss during the gamma calculation due to the division by 2. |

**Table 8.** Power and Packaging

| Feature | Description |
|---|---|
| **Reduced Power Dissipation** | • Low-power CMOS design.<br>• Separate power supply for internal logic (1.2 V for 400 MHz or 500 MHz) and I/O (3.3 V).<br>• Low-power standby modes.<br>• Optimized power management circuitry (instruction-dependent, peripheral-dependent, and mode-dependent). |
| **Packaging** | • 0.8 mm pitch Flip-Chip Plastic Ball-Grid Array (FC-PBGA).<br>• 431-connection (ball).<br>• 20 mm × 20 mm. |

**Table 9.** Software Support

| Feature | Description |
|---|---|
| **Real-Time Operating System (RTOS)** | The real-time operating system (RTOS) fully supports device architecture (multi-core, memory hierarchy, ICache, timers, DMA controller, interrupts, peripherals), as follows:<br>• High-performance and deterministic, delivering predictive response time.<br>• Optimized to provide low interrupt latency with high data throughput.<br>• Preemptive and priority-based multitasking.<br>• Fully interrupt/event driven.<br>• Small memory footprint.<br>• Comprehensive set of APIs. |
| **Multi-Core Support** | • One instance of kernel code in all four SC140 cores.<br>• Dynamic and static memory allocation from local memory (M1) and shared memory (M2). |
| **Distributed System Support** | Transparent inter-task communications between tasks running inside the SC140 cores and the other tasks running in on-board devices or remote network devices:<br>• Messaging mechanism between tasks using mailboxes and semaphores.<br>• Networking support; data transfer between tasks running inside and outside the device using networking protocols.<br>• Includes integrated device drivers for such peripherals as TDM, UART, and external buses. |
| **Software Support** | • Task debugging utilities integrated with compilers and vendors.<br>• Board support package (BSP) for the application development system (ADS).<br>• Integrated development environment (IDE):<br>—C/C++ compiler with in-line assembly so developers can generate highly optimized DSP code. Translates C/C++ code into parallel fetch sets and maintains high code density.<br>—Librarian. User can create libraries for modularity.<br>—A collection of C/C++ functions for developer use.<br>—Highly efficient linker to produce executables from object code.<br>—Seamlessly integrated real-time, non-intrusive multi-mode debugger for debugging highly optimized DSP algorithms. The developer can choose to debug in source code, assembly code, or mixed mode.<br>—Device simulation models enable design and simulation before hardware availability.<br>—Profiler using a patented binary code instrumentation (BCI) technique helps developers identify program design inefficiencies.<br>—Version control. Metrowerks® CodeWarrior® includes plug-ins for ClearCase, Visual SourceSafe, and CVS. |
| **Boot Options** | • External memory.<br>• External host.<br>• UART.<br>• TDM.<br>• $I^2C$ |

**Table 1-5.** DSI, System Bus, Ethernet, and Interrupt Signals (Continued)

| Signal Name | Type | Description |
|---|---|---|
| **$\overline{\text{IRQ1}}$** | Input | **Interrupt Request 1**<br>One of fifteen external lines that can request a service routine, via the internal interrupt controller, from the SC140 core. |
| DP1 | Input/ Output | **System Bus Data Parity** 1<br>The agent that drives the data bus also drives the data parity signals. The value driven on the data parity 1 signal should give odd parity (odd number of ones) on the group of signals that includes data parity 1 and D[8–15]. |
| $\overline{\text{DACK1}}$ | Output | **DMA Acknowledge 1**<br>The DMA controller drives this output to acknowledge the DMA transaction on the bus. |
| $\overline{\text{EXT\_BG2}}$ | Output | **External Bus Grant 2**[2]<br>The MSC8126 asserts this signal to grant bus ownership to an external bus master. |
| **$\overline{\text{IRQ2}}$** | Input | **Interrupt Request 2**<br>One of fifteen external lines that can request a service routine, via the internal interrupt controller, from the SC140 core. |
| DP2 | Input/ Output | **System Bus Data Parity 2**<br>The agent that drives the data bus also drives the data parity signals. The value driven on the data parity 2 signal should give odd parity (odd number of ones) on the group of signals that includes data parity 2 and D[16–23]. |
| $\overline{\text{DACK2}}$ | Output | **DMA Acknowledge 2**<br>The DMA controller drives this output to acknowledge the DMA transaction on the bus. |
| $\overline{\text{EXT\_DBG2}}$ | Output | **External Data Bus Grant 2**[2]<br>The MSC8126 asserts this signal to grant data bus ownership to an external bus master. |
| **$\overline{\text{IRQ3}}$** | Input | **Interrupt Request 3**<br>One of fifteen external lines that can request a service routine, via the internal interrupt controller, from the SC140 core. |
| DP3 | Input/ Output | **System Bus Data Parity 3**<br>The agent that drives the data bus also drives the data parity signals. The value driven on the data parity 3 signal should give odd parity (odd number of ones) on the group of signals that includes data parity 3 and D[24–31]. |
| DREQ2 | Input | **DMA Request 2**<br>Used by an external peripheral to request DMA service. |
| $\overline{\text{EXT\_BR3}}$ | Input | **External Bus Request 3**[2]<br>An external master should assert this signal to request bus ownership from the internal arbiter. |
| **$\overline{\text{IRQ4}}$** | Input | **Interrupt Request 4**<br>One of fifteen external lines that can request a service routine, via the internal interrupt controller, from the SC140 core. |
| DP4 | Input/ Output | **System Bus Data Parity 4**<br>The agent that drives the data bus also drives the data parity signals. The value driven on the data parity 4 signal should give odd parity (odd number of ones) on the group of signals that includes data parity 4 and D[32–39]. |
| $\overline{\text{DACK3}}$ | Output | **DMA Acknowledge 3**<br>The DMA controller drives this output to acknowledge the DMA transaction on the bus. |
| $\overline{\text{EXT\_DBG3}}$ | Output | **External Data Bus Grant 3**[2]<br>The MSC8126 asserts this signal to grant data bus ownership to an external bus master. |

**Table 1-5.** DSI, System Bus, Ethernet, and Interrupt Signals (Continued)

| Signal Name | Type | Description |
|---|---|---|
| **$\overline{\text{IRQ5}}$** | Input | **Interrupt Request 5**<br>One of fifteen external lines that can request a service routine, via the internal interrupt controller, from the SC140 core. |
| DP5 | Input/ Output | **System Bus Data Parity 5**<br>The agent that drives the data bus also drives the data parity signals. The value driven on the data parity 5 signal should give odd parity (odd number of ones) on the group of signals that includes data parity 5 and D[40–47]. |
| $\overline{\text{DACK4}}$ | Output | **DMA Acknowledge 4**<br>The DMA controller drives this output to acknowledge the DMA transaction on the bus. |
| $\overline{\text{EXT\_BG3}}$ | Output | **External Bus Grant 3**[2]<br>The MSC8126 asserts this signal to grant bus ownership to an external bus. |
| **$\overline{\text{IRQ6}}$** | Input | **Interrupt Request 6**<br>One of fifteen external lines that can request a service routine, via the internal interrupt controller, from the SC140 core. |
| DP6 | Input/ Output | **System Bus Data Parity 6**<br>The agent that drives the data bus also drives the data parity signals. The value driven on the data parity 6 signal should give odd parity (odd number of ones) on the group of signals that includes data parity 6 and D[48–55]. |
| DREQ3 | Input | **DMA Request 3**<br>Used by an external peripheral to request DMA service. |
| **$\overline{\text{IRQ7}}$** | Input | **Interrupt Request 7**<br>One of fifteen external lines that can request a service routine, via the internal interrupt controller, from the SC140 core. |
| DP7 | Input/ Output | **System Bus Data Parity 7**<br>The agent that drives the data bus also drives the data parity signals. The value driven on the data parity 7 signal should give odd parity (odd number of ones) on the group of signals that includes data parity 7 and D[56–63]. |
| DREQ4 | Input | **DMA Request 4**<br>Used by an external peripheral to request DMA service. |
| $\overline{\text{TA}}$ | Input/ Output | **Transfer Acknowledge**<br>Indicates that a data beat is valid on the data bus. For single-beat transfers, $\overline{\text{TA}}$ assertion indicates the termination of the transfer. For burst transfers, $\overline{\text{TA}}$ is asserted eight times to indicate the transfer of eight data beats, with the last assertion indicating the termination of the burst transfer. |
| $\overline{\text{TEA}}$ | Input/ Output | **Transfer Error Acknowledge**<br>This signal indicates a failure of the data tenure transaction.The masters within the MSC8126 monitor the state of this signal. The MSC8126 internal bus monitor can assert this signal if it identifies a bus transfer that does not complete. |
| $\overline{\text{NMI}}$ | Input | **Non-Maskable Interrupt**<br>When an external device asserts this line, it generates an non-maskable interrupt in the MSC8126, which is processed internally (default) or is directed to an external host for processing (see $\overline{\text{NMI\_OUT}}$). |
| $\overline{\text{NMI\_OUT}}$ | Output | **Non-Maskable Interrupt Output**<br>An open-drain signal driven from the MSC8126 internal interrupt controller. This output indicates whether a non-maskable interrupt is pending in the MSC8126 internal interrupt controller, waiting to be handled by an external host. |
| $\overline{\text{PSDVAL}}$ | Input/ Output | **Port Size Data Valid**<br>Indicates that a data beat is valid on the data bus. The difference between the $\overline{\text{TA}}$ signal and the $\overline{\text{PSDVAL}}$ signal is that the $\overline{\text{TA}}$ signal is asserted to indicate data transfer terminations, while the $\overline{\text{PSDVAL}}$ signal is asserted with each data beat movement. When $\overline{\text{TA}}$ is asserted, $\overline{\text{PSDVAL}}$ is always asserted. However, when $\overline{\text{PSDVAL}}$ is asserted, $\overline{\text{TA}}$ is not necessarily asserted. For example, if the DMA controller initiates a double word ($2 \times$ 64 bits) transaction to a memory device with a 32-bit port size, $\overline{\text{PSDVAL}}$ is asserted three times without $\overline{\text{TA}}$ and, finally, both signals are asserted to terminate the transfer. |

**Table 1-7.** GPIO, TDM, UART, Ethernet, and Timer Signals (Continued)

| Signal Name | Type | Description |
|---|---|---|
| **GPIO23** | Input/ Output | **General-Purpose Input Output 23**<br>One of 32 GPIO signals used as GPIO or as one of two dedicated inputs or one of two dedicated outputs. For details, refer to the *MSC8126 Reference Manual* GPIO programming model. |
| TDM0TDAT | Input/ Output | **TDM0 Serial Transmitter Data**<br>The transmit data signal for TDM 0. As an output, this can be the DATA_D data signal for TDM 0. For configuration details, refer to the *MSC8126 Reference Manual* chapter describing TDM operation. |
| $\overline{\text{IRQ13}}$ | Input | **Interrupt Request 13**<br>One of fifteen external lines that can request a service routine, via the internal interrupt controller, from the SC140 core. |
| **GPIO24** | Input/ Output | **General-Purpose Input Output 24**<br>One of 32 GPIO signals used as GPIO or as one of two dedicated inputs or one of two dedicated outputs. For details, refer to the *MSC8126 Reference Manual* GPIO programming model. |
| TDM0RSYN | Input/ Output | **TDM0 Receive Frame Sync**<br>The receive sync signal for TDM 0. As an input, this can be the DATA_B data signal for TDM 0. For configuration details, refer to the *MSC8126 Reference Manual* chapter describing TDM operation. |
| $\overline{\text{IRQ14}}$ | Input | **Interrupt Request 14**<br>One of fifteen external lines that can request a service routine, via the internal interrupt controller, from the SC140 core. |
| **GPIO25** | Input/ Output | **General-Purpose Input Output 25**<br>One of 32 GPIO signals used as GPIO or as one of two dedicated inputs or one of two dedicated outputs. For details, refer to the *MSC8126 Reference Manual* GPIO programming model. |
| TDM0RCLK | Input/ Output | **TDM0 Receive Clock**<br>The receive clock signal for TDM 0. As an input, this can be the DATA_C data signal for TDM 0. For configuration details, refer to the *MSC8126 Reference Manual* chapter describing TDM operation. |
| $\overline{\text{IRQ15}}$ | Input | **Interrupt Request 15**<br>One of fifteen external lines that can request a service routine, via the internal interrupt controller, from the SC140 core. |
| **GPIO26** | Input/ Output | **General-Purpose Input Output 26**<br>One of 32 GPIO signals used as GPIO or as one of two dedicated inputs or one of two dedicated outputs. For details, refer to the *MSC8126 Reference Manual* GPIO programming model. |
| TDM0RDAT | Input/ Output | **TDM0 Serial Receiver Data**<br>The receive data signal for TDM 0. As an input, this can be the DATA_A data signal for TDM 0. For configuration details, refer to the *MSC8126 Reference Manual* chapter describing TDM operation. |
| **GPIO27** | Input/ Output | **General-Purpose Input Output 27**<br>One of 32 GPIO signals used as GPIO or as one of two dedicated inputs or one of two dedicated outputs. For details, refer to the *MSC8126 Reference Manual* GPIO programming model. |
| DREQ1 | Input | **DMA Request 1**<br>Used by an external peripheral to request DMA service. |
| URXD | Input | **UART Receive Data** |
| **GPIO28** | Input/ Output | **General-Purpose Input Output 28**<br>One of 32 GPIO signals used as GPIO or as one of two dedicated inputs or one of two dedicated outputs. For details, refer to the *MSC8126 Reference Manual* GPIO programming model. |
| DREQ2 | Input | **DMA Request 2**<br>Used by an external peripheral to request DMA service. |
| UTXD | Output | **UART Transmit Data** |

**Table 1-8.** Dedicated Ethernet Signals

| Signal Name | Type | Signal Description |
|---|---|---|
| **ETHCRS** | Input | **Carrier Sense**<br>In MII mode, indicates that either the transmit or receive medium is non-idle. |
| ETHRXD | Input | **Ethernet Receive Data**<br>In SMII mode, used for the Ethernet receive data. |

# 1.8 EOnCE Event and JTAG Test Access Port Signals

The MSC8126 uses two sets of debugging signals for the two types of internal debugging modules: EOnCE and the JTAG TAP controller. Each internal SC140 core has an EOnce module, but they are all accessed externally by the same two signals EE0 and EE1. The MSC8126 supports the standard set of test access port (TAP) signals defined by **IEEE** Std 1149.1 Standard Test Access Port and Boundary-Scan Architecture specification and described in **Table 1-9**.

**Table 1-9.** JTAG TAP Signals

| Signal Name | Type | Signal Description |
|---|---|---|
| EE0 | Input | **EOnCE Event Bit 0**<br>Puts the internal SC140 cores into Debug mode. |
| EE1 | Output | **EOnCE Event Bit 1**<br>Indicates that at least one on-device SC140 core is in Debug mode. |
| TCK | Input | **Test Clock**<br>Synchronizes the JTAG test logic. |
| TDI | Input | **Test Data Input**<br>A test data serial signal for test instructions and data. TDI is sampled on the rising edge of TCK and has an internal pull-up resistor. |
| TDO | Output | **Test Data Output**<br>A test data serial signal for test instructions and data. TDO can be tri-stated. The signal is actively driven in the shift-IR and shift-DR controller states and changes on the falling edge of TCK. |
| TMS | Input | **Test Mode Select**<br>Sequences the test controller state machine, is sampled on the rising edge of TCK, and has an internal pull-up resistor. |
| $\overline{\text{TRST}}$ | Input | **Test Reset**<br>Asynchronously initializes the test controller; must be asserted during power up. |

# 1.9 Reserved Signals

**Table 1-10.** Reserved Signals

| Signal Name | Type | Signal Description |
|---|---|---|
| TEST | Input | **Test**<br>For manufacturing testing. You *must* connect this signal to GND. |

**Table 2-4.** DC Electrical Characteristics

| Characteristic | Symbol | Min | Typical | Max | Unit |
|---|---|---|---|---|---|
| Input high voltage[1], all inputs except CLKIN | $V_{IH}$ | 2.0 | — | 3.465 | V |
| Input low voltage[1] | $V_{IL}$ | GND | 0 | 0.4 | V |
| CLKIN input high voltage | $V_{IHC}$ | 2.4 | 3.0 | 3.465 | V |
| CLKIN input low voltage | $V_{ILC}$ | GND | 0 | 0.4 | V |
| Input leakage current, $V_{IN} = V_{DDH}$ | $I_{IN}$ | –1.0 | 0.09 | 1 | µA |
| Tri-state (high impedance off state) leakage current, $V_{IN} = V_{DDH}$ | $I_{OZ}$ | –1.0 | 0.09 | 1 | µA |
| Signal low input current, $V_{IL} = 0.4$ V[2] | $I_L$ | –1.0 | 0.09 | 1 | µA |
| Signal high input current, $V_{IH} = 2.0$ V[2] | $I_H$ | –1.0 | 0.09 | 1 | µA |
| Output high voltage, $I_{OH} = –2$ mA, except open drain pins | $V_{OH}$ | 2.0 | 3.0 | — | V |
| Output low voltage, $I_{OL} = 3.2$ mA | $V_{OL}$ | — | 0 | 0.4 | V |
| Internal supply current:<br>• Wait mode<br>• Stop mode | <br>$I_{DDW}$<br>$I_{DDS}$ | <br>—<br>— | <br>375[3]<br>290[3] | <br>—<br>— | <br>mA<br>mA |
| Typical power at 400 MHz[4] | P | — | 1.15 | — | W |

**Notes:**
1. See **Figure 2-1** for undershoot and overshoot voltages.
2. Not tested. Guaranteed by design.
3. Measured for 1.2 V core at 25°C junction temperature.
4. The typical power values were measured using an EFR code with the device running at a junction temperature of 25°C. No peripherals were enabled and the ICache was not enabled. The source code was optimized to use all the ALUs and AGUs and all four cores. It was created using CodeWarrior® 2.5. These values are provided as examples only. Power consumption is application dependent and varies widely. To assure proper board design with regard to thermal dissipation and maintaining proper operating temperatures, evaluate power consumption for your application and use the design guidelines in **Chapter 4** of this document and in *MSC8102, MSC8122, and MSC8126 Thermal Management Design Guidelines* (AN2601).

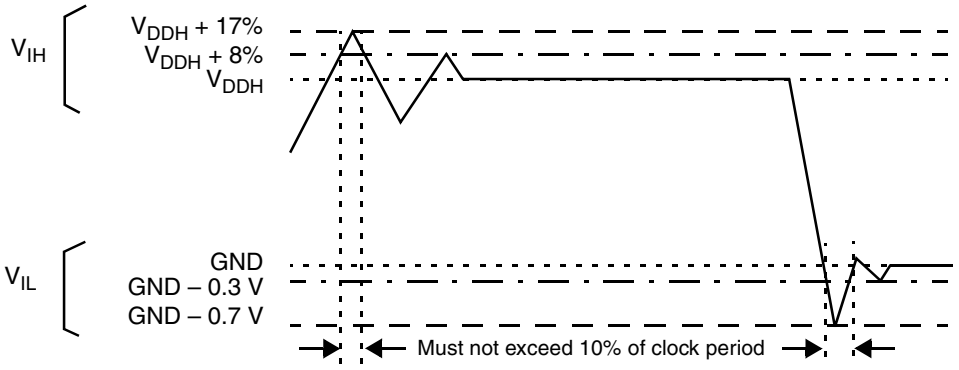


**Figure 2-1.** Overshoot/Undershoot Voltage for $V_{IH}$ and $V_{IL}$

# 2.5 AC Timings

The following sections include illustrations and tables of clock diagrams, signals, and parallel I/O outputs and inputs. When systems such as DSP farms are developed using the DSI, use a device loading of 4 pF per pin. AC timings are based on a 20 pF load, except where noted otherwise, and a 50 Ω transmission line. For loads smaller than 20 pF, subtract 0.06 ns per pF down to 10 pF load. For loads larger than 20 pF, add 0.06 ns for SIU/Ethernet/DSI delay and 0.07 ns for GPIO/TDM/timer delay. When calculating overall loading, also consider additional RC delay.

## 2.5.1 Output Buffer Impedances

**Table 2-5.** Output Buffer Impedances

| Output Buffers | Typical Impedance (Ω) |
|---|---|
| System bus | 50 |
| Memory controller | 50 |
| Parallel I/O | 50 |
| **Note:** These are typical values at 65°C. The impedance may vary by ±25% depending on device process and operating temperature. | |

## 2.5.2 Start-Up Timing

Starting the device requires coordination among several input sequences including clocking, reset, and power. **Section 2.5.3** describes the clocking characteristics. **Section 2.5.4** describes the reset and power-up characteristics. You must use the following guidelines when starting up an MSC8126 device:

- $\overline{\text{PORESET}}$ and $\overline{\text{TRST}}$ must be asserted externally for the duration of the power-up sequence. See **Table 2-10** for timing.
- If possible, bring up the $V_{DD}$ and $V_{DDH}$ levels together. For designs with separate power supplies, bring up the $V_{DD}$ levels and then the $V_{DDH}$ levels (see **Figure 2-3**).
- CLKIN should start toggling at least 16 cycles (starting after $V_{DDH}$ reaches its nominal level) before $\overline{\text{PORESET}}$ deassertion to guarantee correct device operation (see **Figure 2-2** and **Figure 2-3**).
- CLKIN must not be pulled high during $V_{DDH}$ power-up. CLKIN can toggle during this period.

The following figures show acceptable start-up sequence examples. **Figure 2-2** shows a sequence in which $V_{DD}$ and $V_{DDH}$ are raised together. **Figure 2-3** shows a sequence in which $V_{DDH}$ is raised after $V_{DD}$ and CLKIN begins to toggle as $V_{DDH}$ rises.

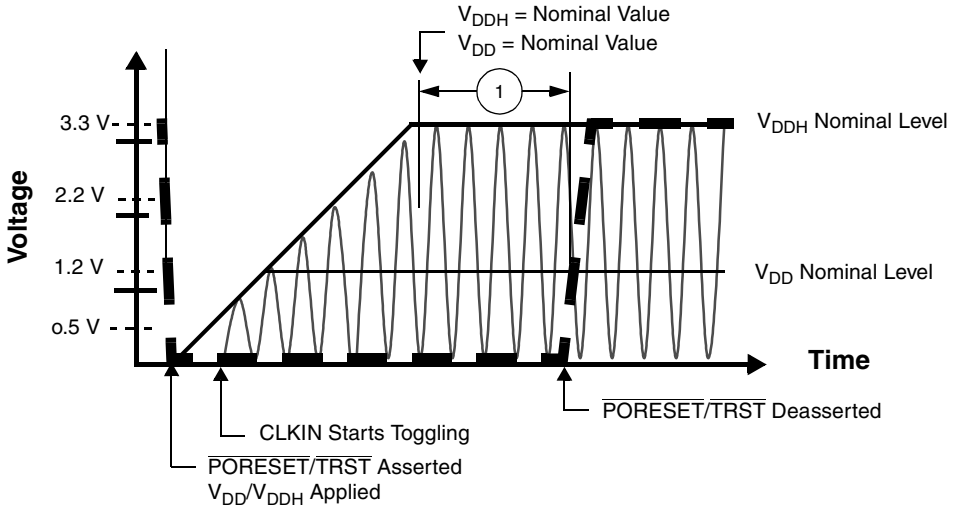


**Figure 2-2.** Start-Up Sequence with $V_{DD}$ and $V_{DDH}$ Raised Together

### 2.5.4.3 Reset Timing Tables

**Table 2-11** and **Figure 2-4** describe the reset timing for a reset configuration write through the direct slave interface (DSI) or through the system bus.

**Table 2-11.** Timing for a Reset Configuration Write through the DSI or System Bus

| No. | Characteristics | Expression | Min | Max | Unit |
|---|---|---|---|---|---|
| 1 | Required external $\overline{\text{PORESET}}$ duration minimum<br>• CLKIN = 20 MHz<br>• CLKIN = 133 MHz (400 MHz core)<br>• CLKIN = 166 MHz (500 MHz core) | 16/CLKIN | <br>800<br>120<br>96 | <br>800<br>—<br>— | <br>ns<br>ns<br>ns |
| 2 | Delay from deassertion of external $\overline{\text{PORESET}}$ to deassertion of internal $\overline{\text{PORESET}}$<br>• CLKIN = 20 MHz to 166 MHz | 1024/CLKIN | <br>6.17 | <br>51.2 | <br>µs |
| 3 | Delay from de-assertion of internal $\overline{\text{PORESET}}$ to SPLL lock<br>• CLKIN = 20 MHz (RDF = 1)<br>• CLKIN = 133 MHz (RDF = 2) (400 MHz core)<br>• CLKIN = 166 MHz (RDF = 2) (500 MHz core) | 6400/(CLKIN/RDF)<br>(PLL reference clock-division factor) | <br>320<br>96<br>77 | <br>320<br>96<br>77 | <br>µs<br>µs<br>µs |
| 5 | Delay from SPLL to $\overline{\text{HRESET}}$ deassertion<br>• REFCLK = 40 MHz to 166 MHz | 512/REFCLK | 3.08 | 12.8 | µs |
| 6 | Delay from SPLL lock to $\overline{\text{SRESET}}$ deassertion<br>• REFCLK = 40 MHz to 166 MHz | 515/REFCLK | 3.10 | 12.88 | µs |
| 7 | Setup time from assertion of $\overline{\text{RSTCONF}}$, CNFGS, DSISYNC, DSI64, CHIP_ID[0–3], BM[0–2], SWTE, and MODCK[1–2] before deassertion of $\overline{\text{PORESET}}$ | | 3 | — | ns |
| 8 | Hold time from deassertion of $\overline{\text{PORESET}}$ to deassertion of $\overline{\text{RSTCONF}}$, CNFGS, DSISYNC, DSI64, CHIP_ID[0–3], BM[0–2], SWTE, and MODCK[1–2] | | 5 | — | ns |
| **Note:** | Timings are not tested, but are guaranteed by design. | | | | |

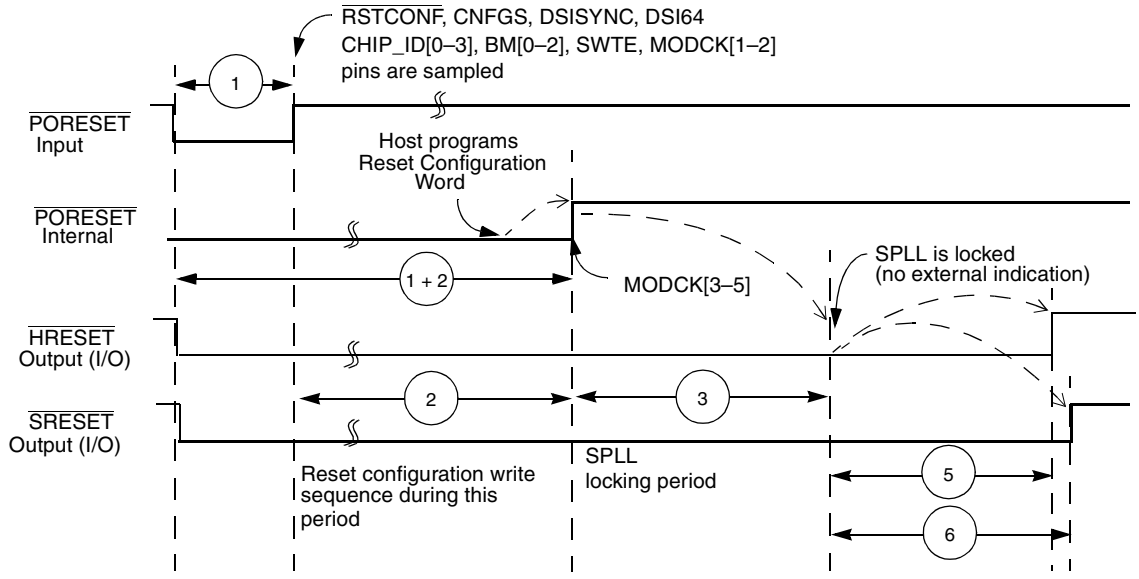


**Figure 2-4.** Timing Diagram for a Reset Configuration Write

The UPM machine and GPCM machine outputs change on the internal tick determined by the memory controller configuration. The AC specifications are relative to the internal tick. SDRAM machine outputs change only on the REFCLK rising edge.

**Table 2-13.** AC Timing for SIU Inputs

| No. | Characteristic | Value for Bus Speed in MHz | | | Units |
|---|---|---|---|---|---|
| | | Ref = CLKIN | | Ref = CLKOUT | |
| | | 133 | 166 | 133 | |
| 10 | Hold time for all signals after the 50% level of the REFCLK rising edge | 0.5 | 0.5 | 0.5 | ns |
| 11a | $\overline{\text{ARTRY}}/\overline{\text{ABB}}$ set-up time before the 50% level of the REFCLK rising edge | 3.0 | 3.0 | 3.0 | ns |
| 11b | $\overline{\text{DBG}}/\overline{\text{DBB}}/\overline{\text{BG}}/\overline{\text{BR}}/\overline{\text{TC}}$ set-up time before the 50% level of the REFCLK rising edge | 3.3 | 3.3 | 3.3 | ns |
| 11c | $\overline{\text{AACK}}$ set-up time before the 50% level of the REFCLK rising edge | 2.9 | 2.9 | 2.9 | ns |
| 11d | $\overline{\text{TA}}/\overline{\text{TEA}}/\overline{\text{PSDVAL}}$ set-up time before the 50% level of the REFCLK rising edge<br>• Data-pipeline mode<br>• Non-pipeline mode | <br>3.4<br>4.0 | <br>3.4<br>4.0 | <br>3.4<br>4.0 | <br>ns<br>ns |
| 12 | Data bus set-up time before REFCLK rising edge in Normal mode<br>• Data-pipeline mode<br>• Non-pipeline mode | <br>1.8<br>4.0 | <br>1.7<br>4.0 | <br>1.8<br>4.0 | <br>ns<br>ns |
| 13 | Data bus set-up time before the 50% level of the REFCLK rising edge in ECC and PARITY modes<br>• Data-pipeline mode<br>• Non-pipeline mode | <br>2.0<br>7.3 | <br>2.0<br>7.3 | <br>2.0<br>7.3 | <br>ns<br>ns |
| 14 | DP set-up time before the 50% level of the REFCLK rising edge<br>• Data-pipeline mode<br>• Non-pipeline mode | <br>2.0<br>6.1 | <br>2.0<br>6.1 | <br>2.0<br>6.1 | <br>ns<br>ns |
| 15a | $\overline{\text{TS}}$ and Address bus set-up time before the 50% level of the REFCLK rising edge<br>• Extra cycle mode (SIUBCR[EXDD] = 0)<br>• No extra cycle mode (SIUBCR[EXDD] = 1) | <br>3.6<br>5.0 | <br>3.6<br>5.0 | <br>3.8<br>5.0 | <br>ns<br>ns |
| 15b | Address attributes: TT/$\overline{\text{TBST}}$/TSZ/$\overline{\text{GBL}}$ set-up time before the 50% level of the REFCLK rising edge<br>• Extra cycle mode (SIUBCR[EXDD] = 0)<br>• No extra cycle mode (SIUBCR[EXDD] = 1) | <br>3.5<br>4.4 | <br>3.5<br>4.4 | <br>3.5<br>4.4 | <br>ns<br>ns |
| 16[1] | PUPMWAIT signal set-up time before the 50% level of the REFCLK rising edge | 3.7 | 3.7 | 3.7 | ns |
| **Note:** | Values are measured from the 50% TTL transition level relative to the 50% level of the REFCLK rising edge. | | | | |

**Table 2-14.** AC Timing for SIU Outputs

| No. | Characteristic | Value for Bus Speed in MHz | | | Units |
|---|---|---|---|---|---|
| | | Ref = CLKIN | | Ref = CLKOUT | |
| | | 133 | 166 | 133 | |
| 30[2] | Minimum delay from the 50% level of the REFCLK for all signals | 0.8 | 0.8 | 1.0 | ns |
| 31 | $\overline{\text{PSDVAL}}/\overline{\text{TEA}}/\overline{\text{TA}}$ max delay from the 50% level of the REFCLK rising edge | 4.9 | 4.9 | 5.8 | ns |
| 32a | Address bus max delay from the 50% level of the REFCLK rising edge<br>• Multi-master mode (SIUBCR[EBM] = 1)<br>• Single-master mode (SIUBCR[EBM] = 0) | <br>5.5<br>4.2 | <br>5.5<br>3.9 | <br>6.4<br>5.1 | <br>ns<br>ns |
| 32b | Address attributes: TT[0–1]/$\overline{\text{TBST}}$/TSZ/$\overline{\text{GBL}}$ max delay from the 50% level of the REFCLK rising edge | 5.1 | 5.1 | 6.0 | ns |
| 32c | Address attributes: TT[2–4]/TC max delay from the 50% level of the REFCLK rising edge | 5.7 | 5.7 | 6.6 | ns |
| 32d | BADDR max delay from the 50% level of the REFCLK rising edge | 4.2 | 4.2 | 5.1 | ns |
| 33a | Data bus max delay from the 50% level of the REFCLK rising edge<br>• Data-pipeline mode<br>• Non-pipeline mode | <br>3.9<br>6.1 | <br>3.7<br>6.1 | <br>4.8<br>7.0 | <br>ns<br>ns |
| 33b | DP max delay from the 50% level of the REFCLK rising edge<br>• Data-pipeline mode<br>• Non-pipeline mode | <br>5.3<br>6.5 | <br>5.3<br>6.5 | <br>6.2<br>7.4 | <br>ns<br>ns |
| 34 | Memory controller signals/ALE/$\overline{\text{CS}}$[0–4] max delay from the 50% level of the REFCLK rising edge | 4.2 | 3.9 | 5.1 | ns |
| 35a | $\overline{\text{DBG}}/\overline{\text{BG}}/\overline{\text{BR}}/\overline{\text{DBB}}$ max delay from the 50% level of the REFCLK rising edge | 4.7 | 4.7 | 5.6 | ns |
| 35b | $\overline{\text{AACK}}/\overline{\text{ABB}}/\overline{\text{TS}}/\overline{\text{CS}}$[5–7] max delay from the 50% level of the REFCLK rising edge | 4.5 | 4.5 | 5.4 | ns |

**Notes:**
1. Values are measured from the 50% level of the REFCLK rising edge to the 50% signal level and assume a 20 pF load except where otherwise specified.
2. The load for specification 30 is 10 pF. The load for the other specifications in this table is 20 pF. For a 15 pF load, subtract 0.3 ns from the listed value.
3. The maximum bus frequency depends on the mode:
   • In 60x-compatible mode connected to another MSC8126 device, the frequency is determined by adding the input and output longest timing values, which results in the total delay for 20 pF output capacitance. You must also account for other influences that can affect timing, such as on-board clock skews, on-board noise delays, and so on.
   • In single-master mode, the frequency depends on the timing of the devices connected to the MSC8126.
   • To achieve maximum performance on the bus in single-master mode, disable the $\overline{\text{DBB}}$ signal by writing a 1 to the SIUMCR[BDD] bit. See the SIU chapter in the *MSC8122 Reference Manual* for details.

# 2.5.10 Ethernet Timing

## 2.5.10.1 Management Interface Timing

**Table 2-23.** Ethernet Controller Management Interface Timing

| No. | Characteristics | Min | Max | Unit |
|---|---|---|---|---|
| 801 | ETHMDIO to ETHMDC rising edge set-up time | 10 | — | ns |
| 802 | ETHMDC rising edge to ETHMDIO hold time | 10 | — | ns |

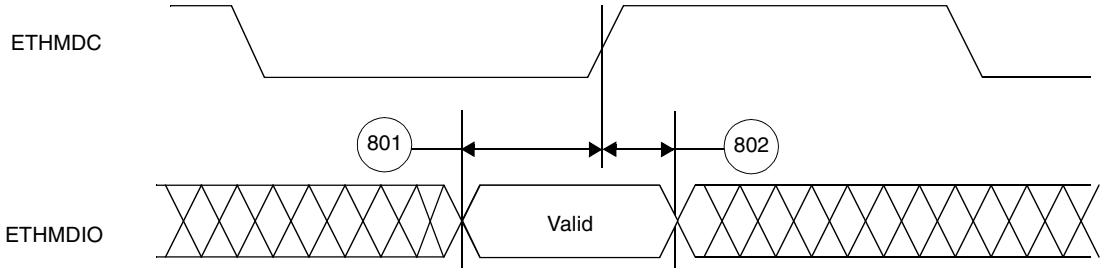


**Figure 2-18.** MDIO Timing Relationship to MDC

## 2.5.10.2 MII Mode Timing

**Table 2-24.** MII Mode Signal Timing

| No. | Characteristics | Min | Max | Unit |
|---|---|---|---|---|
| 803 | ETHRX_DV, ETHRXD[0–3], ETHRX_ER to ETHRX_CLK rising edge set-up time | 3.5 | — | ns |
| 804 | ETHRX_CLK rising edge to ETHRX_DV, ETHRXD[0–3], ETHRX_ER hold time | 3.5 | — | ns |
| 805 | ETHTX_CLK to ETHTX_EN, ETHTXD[0–3], ETHTX_ER output delay | 1 | 12.6 | ns |

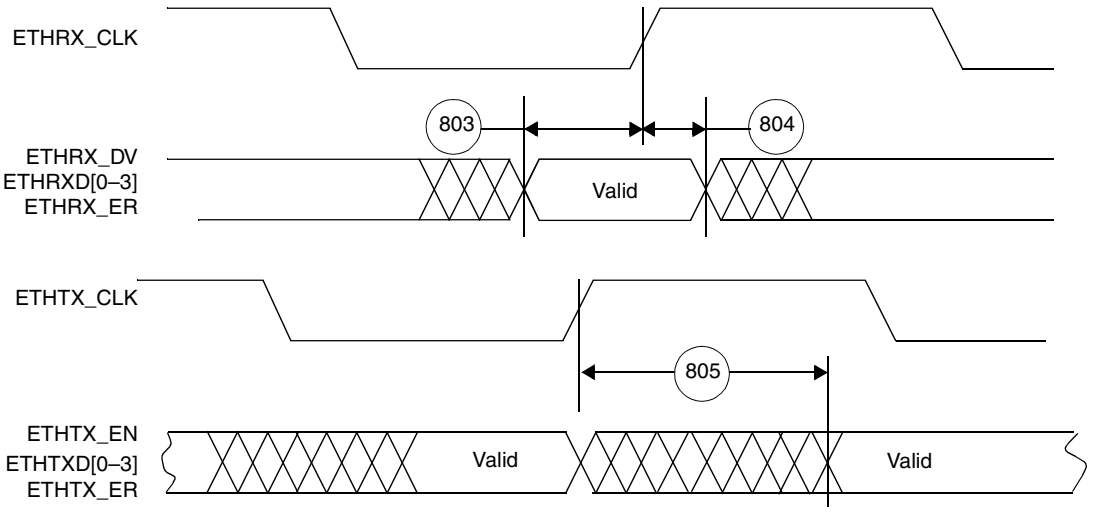


**Figure 2-19.** MII Mode Signal Timing

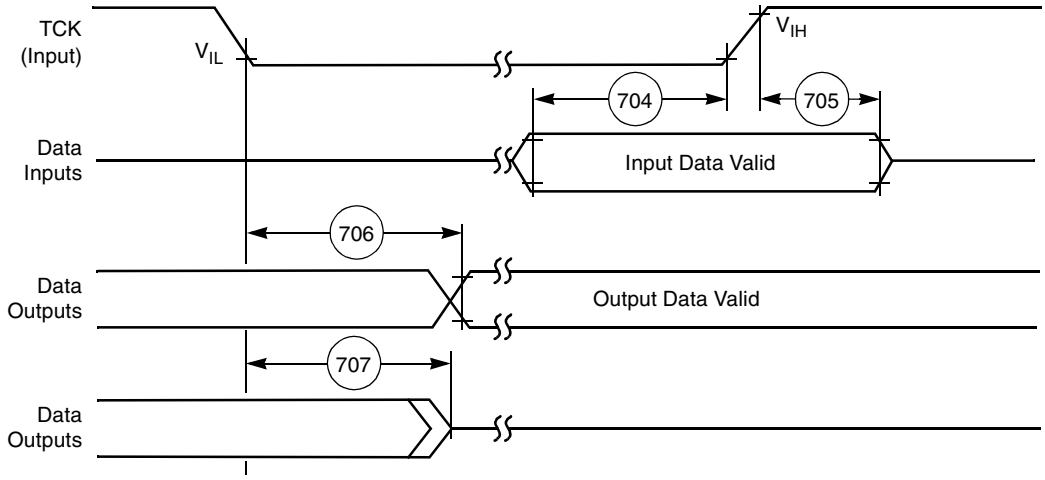


**Figure 2-25.** Boundary Scan (JTAG) Timing Diagram

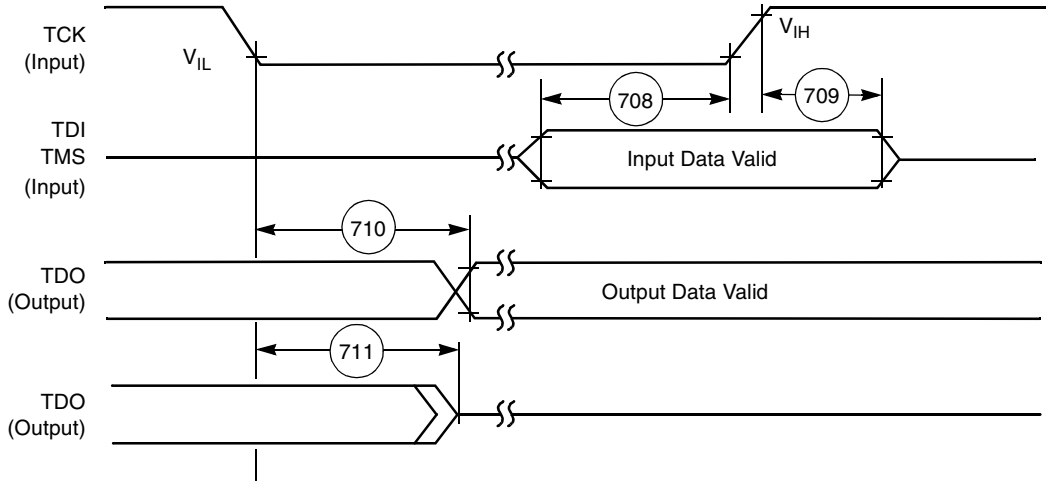


**Figure 2-26.** Test Access Port Timing Diagram

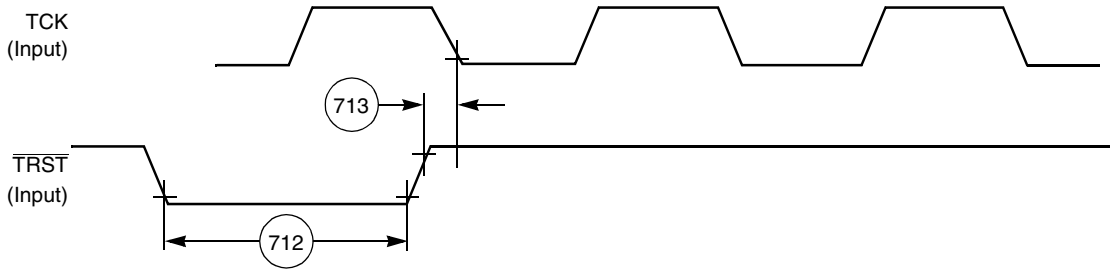


**Figure 2-27.** $\overline{TRST}$ Timing Diagram

**Table 3-1.** MSC8126 Signal Listing By Name (Continued)

| Signal Name | Location Designator | Signal Name | Location Designator |
|---|---|---|---|
| GND | B13 | GND | L14 |
| GND | B15 | GND | L16 |
| GND | B17 | GND | L17 |
| GND | B22 | GND | M5 |
| GND | C2 | GND | M6 |
| GND | C8 | GND | M10 |
| GND | C10 | GND | M14 |
| GND | C12 | GND | M19 |
| GND | C14 | GND | N10 |
| GND | C15 | GND | N14 |
| GND | D5 | GND | P10 |
| GND | D9 | GND | P13 |
| GND | D11 | GND | P14 |
| GND | D13 | GND | P21 |
| GND | D21 | GND | R4 |
| GND | E8 | GND | T20 |
| GND | E10 | GND | V4 |
| GND | E12 | GND | V15 |
| GND | E14 | GND | W5 |
| GND | E15 | GND | W6 |
| GND | E17 | GND | W9 |
| GND | E18 | GND | W13 |
| GND | F7 | GND | W19 |
| GND | F11 | GND | W20 |
| GND | F13 | GND | Y9 |
| GND | G20 | GND | Y12 |
| GND | J6 | GND | Y15 |
| GND | J14 | GND | Y17 |
| GND | J20 | GND | AA8 |
| GND | K10 | GND | AA13 |
| GND | K11 | GND | AA16 |
| GND | K12 | GND | AB2 |
| GND | K13 | $GND_{SYN}$ | P11 |
| GND | K19 | GPIO0 | B19 |
| GND | L9 | GPIO1 | C18 |
| GND | L10 | GPIO2 | C17 |

**Table 3-1.** MSC8126 Signal Listing By Name (Continued)

| Signal Name | Location Designator | Signal Name | Location Designator |
|---|---|---|---|
| GPIO3 | C20 | HA14 | L3 |
| GPIO4 | D19 | HA15 | K2 |
| GPIO5 | C21 | HA16 | K4 |
| GPIO6 | C22 | HA17 | G6 |
| GPIO7 | C19 | HA18 | J2 |
| GPIO8 | D22 | HA19 | H5 |
| GPIO9 | E19 | HA20 | H2 |
| GPIO10 | E21 | HA21 | K3 |
| GPIO11 | F20 | HA22 | F6 |
| GPIO12 | E22 | HA23 | G5 |
| GPIO13 | E20 | HA24 | G2 |
| GPIO14 | F21 | HA25 | G4 |
| GPIO15 | G19 | HA26 | J3 |
| GPIO16 | F19 | HA27 | G3 |
| GPIO17 | G21 | HA28 | H3 |
| GPIO18 | F18 | HA29 | F5 |
| GPIO19 | F22 | $\overline{\text{HBCS}}$ | N9 |
| GPIO20 | F17 | $\overline{\text{HBRST}}$ | M16 |
| GPIO21 | H19 | HCID0 | E7 |
| GPIO22 | G22 | HCID1 | C7 |
| GPIO23 | J19 | HCID2 | D7 |
| GPIO24 | H18 | HCID3 | D8 |
| GPIO25 | J21 | HCLKIN | P9 |
| GPIO26 | N20 | $\overline{\text{HCS}}$ | N17 |
| GPIO27 | E6 | HD0 | T5 |
| GPIO28 | C6 | HD1 | T4 |
| GPIO29 | D17 | HD2 | U4 |
| GPIO30 | C16 | HD3 | V2 |
| GPIO31 | D16 | HD4 | W4 |
| HA7 | R14 | HD5 | W3 |
| HA8 | D8 | HD6 | W2 |
| HA9 | W11 | HD7 | Y2 |
| HA10 | W10 | HD8 | AB5 |
| HA11 | L4 | HD9 | Y5 |
| HA12 | L2 | HD10 | AA5 |
| HA13 | J5 | HD11 | AB4 |

**Table 3-1.** MSC8126 Signal Listing By Name (Continued)

| Signal Name | Location Designator | Signal Name | Location Designator |
|---|---|---|---|
| HRESET | E5 | IRQ5 | L8 |
| HRW | N15 | IRQ5 | T16 |
| HTA | H14 | IRQ6 | C17 |
| HWBE0 | N8 | IRQ6 | D22 |
| HWBE1 | P8 | IRQ6 | R16 |
| HWBE2 | P7 | IRQ7 | E19 |
| HWBE3 | P6 | IRQ7 | G14 |
| HWBE4 | R7 | IRQ7 | R15 |
| HWBE5 | T7 | IRQ8 | E21 |
| HWBE6 | R6 | IRQ9 | F20 |
| HWBE7 | T6 | IRQ10 | E22 |
| HWBS0 | N8 | IRQ11 | E20 |
| HWBS1 | P8 | IRQ12 | F21 |
| HWBS2 | P7 | IRQ13 | J19 |
| HWBS3 | P6 | IRQ14 | H18 |
| HWBS4 | R7 | IRQ15 | J21 |
| HWBS5 | T7 | MODCK1 | V2 |
| HWBS6 | R6 | MODCK2 | W4 |
| HWBS7 | T6 | HWBE4 | R7 |
| INT_OUT | G14 | HWBE5 | T7 |
| IRQ1 | C20 | HWBE6 | R6 |
| IRQ1 | R10 | HWBE7 | T6 |
| IRQ1 | T18 | NC | E21 |
| IRQ2 | D19 | NC | F21 |
| IRQ2 | K8 | NMI | F4 |
| IRQ2 | R19 | NMI_OUT | B6 |
| IRQ3 | C21 | PBS0 | G7 |
| IRQ3 | G10 | PBS1 | K6 |
| IRQ3 | R17 | PBS2 | N6 |
| IRQ4 | B19 | PBS3 | K5 |
| IRQ4 | C22 | PBS4 | R7 |
| IRQ4 | G12 | PBS5 | T7 |
| IRQ4 | T17 | PBS6 | R6 |
| IRQ5 | C18 | PBS7 | T6 |
| IRQ5 | C19 | PGPL0 | J17 |
| IRQ5 | H13 | PGPL1 | N19 |

**Table 3-1.** MSC8126 Signal Listing By Name (Continued)

| Signal Name | Location Designator | Signal Name | Location Designator |
|---|---|---|---|
| $V_{DDH}$ | L15 | $V_{DDH}$ | W7 |
| $V_{DDH}$ | L19 | $V_{DDH}$ | W8 |
| $V_{DDH}$ | M4 | $V_{DDH}$ | W12 |
| $V_{DDH}$ | M7 | $V_{DDH}$ | W15 |
| $V_{DDH}$ | M9 | $V_{DDH}$ | W17 |
| $V_{DDH}$ | M15 | $V_{DDH}$ | Y4 |
| $V_{DDH}$ | M17 | $V_{DDH}$ | Y10 |
| $V_{DDH}$ | M18 | $V_{DDH}$ | Y13 |
| $V_{DDH}$ | M20 | $V_{DDH}$ | Y16 |
| $V_{DDH}$ | N7 | $V_{DDH}$ | Y20 |
| $V_{DDH}$ | P20 | $V_{DDH}$ | AA9 |
| $V_{DDH}$ | R3 | $V_{DDH}$ | AA12 |
| $V_{DDH}$ | U20 | $V_{DDH}$ | AA14 |
| $V_{DDH}$ | V3 | | |

Note: This table lists every signal name. Because many signals are multiplexed, an individual ball designator number may be listed several times.

**Table 3-2.** MSC8126 Signal Listing by Ball Designator (Continued)

| Des. | Signal Name | Des. | Signal Name |
|---|---|---|---|
| T6 | HWBS7/HDBS7/HWBE7/HDBE7/PWE7/PSDDQM7/PBS7 | U21 | A12 |
| T7 | HWBS5/HDBS5/HWBE5/HDBE5/PWE5/PSDDQM5/PBS5 | U22 | A13 |
| T8 | TSZ0 | V2 | HD3/MODCK1 |
| T9 | TSZ2 | V3 | $V_{DDH}$ |
| T10 | TBST | V4 | GND |
| T11 | $V_{DD}$ | V5 | D0 |
| T12 | D16 | V6 | D1 |
| T13 | TT1 | V7 | D4 |
| T14 | D21 | V8 | D5 |
| T15 | D23 | V9 | D7 |
| T16 | IRQ5/DP5/DACK4/EXT_BG3 | V10 | D10 |
| T17 | IRQ4/DP4/DACK3/EXT_DBG3 | V11 | D12 |
| T18 | IRQ1/DP1/DACK1/EXT_BG2 | V12 | D13 |
| T19 | D30 | V13 | D18 |
| T20 | GND | V14 | D20 |
| T21 | A15 | V15 | GND |
| T22 | A14 | V16 | D24 |
| U2 | HD16 | V17 | D27 |
| U3 | HD19 | V18 | D29 |
| U4 | HD2/DSI64 | V19 | A8 |
| U5 | D2 | V20 | A9 |
| U6 | D3 | V21 | A10 |
| U7 | D6 | V22 | A11 |
| U8 | D8 | W2 | HD6 |
| U9 | D9 | W3 | HD5/CNFGS |
| U10 | D11 | W4 | HD4/MODCK2 |
| U11 | D14 | W5 | GND |
| U12 | D15 | W6 | GND |
| U13 | D17 | W7 | $V_{DDH}$ |
| U14 | D19 | W8 | $V_{DDH}$ |
| U15 | D22 | W9 | GND |
| U16 | D25 | W10 | HDST1/HA10 |
| U17 | D26 | W11 | HDST0/HA9 |
| U18 | D28 | W12 | $V_{DDH}$ |
| U19 | D31 | W13 | GND |
| U20 | $V_{DDH}$ | W14 | HD40/D40/ETHRXD0 |

# 4.3 Connectivity Guidelines

Unused output pins can be disconnected, and unused input pins should be connected to the non-active value, via resistors to $V_{DDH}$ or GND, except for the following:

- If the DSI is unused (DDR[DSIDIS] is set), $\overline{\text{HCS}}$ and $\overline{\text{HBCS}}$ must pulled up and all the rest of the DSI signals can be disconnected.
- When the DSI uses synchronous mode, $\overline{\text{HTA}}$ must be pulled up. In asynchronous mode, $\overline{\text{HTA}}$ should be pulled either up or down, depending on design requirements.
- $\overline{\text{HDST}}$ can be disconnected if the DSI is in big-endian mode, or if the DSI is in little-endian mode and the DCR[DSRFA] bit is set.
- When the DSI is in 64-bit data bus mode and DCR[BEM] is cleared, pull up $\overline{\text{HWBS[1–3]}}$/$\overline{\text{HDBS[1–3]}}$/$\overline{\text{HWBE[1–3]}}$/$\overline{\text{HDBE[1–3]}}$ and $\overline{\text{HWBS[4–7]}}$/$\overline{\text{HDBS[4–7]}}$/$\overline{\text{HWBE[4–7]}}$/$\overline{\text{HDBE[4–7]}}$/$\overline{\text{PWE[4–7]}}$/$\overline{\text{PSDDQM[4–7]}}$/$\overline{\text{PBS[4–7]}}$.
- When the DSI is in 32-bit data bus mode and DCR[BEM] is cleared, $\overline{\text{HWBS[1–3]}}$/$\overline{\text{HDBS[1–3]}}$/$\overline{\text{HWBE[1–3]}}$/$\overline{\text{HDBE[1–3]}}$ must be pulled up.
- When the DSI is in asynchronous mode, $\overline{\text{HBRST}}$ and HCLKIN should either be disconnected or pulled up.
- The following signals must be pulled up: $\overline{\text{HRESET}}$, $\overline{\text{SRESET}}$, $\overline{\text{ARTRY}}$, $\overline{\text{TA}}$, $\overline{\text{TEA}}$, $\overline{\text{PSDVAL}}$, and $\overline{\text{AACK}}$.
- In single-master mode (BCR[EBM] = 0) with internal arbitration (PPC_ACR[EARB] = 0):
  - $\overline{\text{BG}}$, $\overline{\text{DBG}}$, and $\overline{\text{TS}}$ can be left unconnected.
  - $\overline{\text{EXT\_BG[2–3]}}$, $\overline{\text{EXT\_DBG[2–3]}}$, and $\overline{\text{GBL}}$ can be left unconnected if they are multiplexed to the system bus functionality. For any other functionality, connect the signal lines based on the multiplexed functionality.
  - $\overline{\text{BR}}$ must be pulled up.
  - $\overline{\text{EXT\_BR[2–3]}}$ must be pulled up if multiplexed to the system bus functionality.
- If there is an external bus master (BCR[EBM] = 1):
  - $\overline{\text{BR}}$, $\overline{\text{BG}}$, $\overline{\text{DBG}}$, and $\overline{\text{TS}}$ must be pulled up.
  - $\overline{\text{EXT\_BR[2–3]}}$, $\overline{\text{EXT\_BG[2–3]}}$, and $\overline{\text{EXT\_DBG[2–3]}}$ must be pulled up if multiplexed to the system bus functionality.
- In single-master mode, $\overline{\text{ABB}}$ and $\overline{\text{DBB}}$ can be selected as $\overline{\text{IRQ}}$ inputs and be connected to the non-active value. In other modes, they must be pulled up.

**Note:** The MSC8126 does not support DLL-enabled mode. For the following two clock schemes, ensure that the DLL is disabled (that is, the DLLDIS bit in the Hard Reset Configuration Word is set).

- If no system synchronization is required (for example, the design does not use SDRAM), you can use any of the available clock modes.
- In the CLKIN synchronization mode, use the following connections:
  - Connect the oscillator output through a buffer to CLKIN.
  - Connect the CLKIN buffer output to the slave device (for example, SDRAM) making sure that the delay path between the clock buffer to the MSC8126 and the SDRAM is equal (that is, has a skew less than 100 ps).
  - Valid clock modes in this scheme are: 0, 7, 15, 19, 21, 23, 28, 29, 30, and 31.