Understanding MAX 5000 & Classic Timing

Introduction

Altera® devices provide performance that is consistent from simulation to application. Before programming a device, you can determine the worst-case timing delays for any design. You can calculate propagation delays either with the MAX+PLUS® II Timing Analyzer or with the timing models given in this application note and the timing parameters listed in individual device data sheets. Both methods yield the same results.

This application note defines internal and external timing parameters, and illustrates the timing models for the MAX® 5000 (including MAX 5000A), and Classic™ device families.

Familiarity with device architecture and characteristics is assumed. Refer to the device family data sheets in this data book for complete descriptions of the architectures, and for the specific values of the timing parameters listed in this application note.

Internal Timing Parameters

Within a device, the timing delays contributed by individual architectural elements are called internal timing parameters, which cannot be measured explicitly. All internal timing parameters are shown in italic type. The following section defines the internal timing parameters for MAX 5000 and Classic devices, and applies to both device families unless otherwise indicated. Classic devices include the EP610, EP610I, EP910, EP910I , and EP1810 devices only.

t_{IN} The time required for a dedicated input pin to drive the true and complement data input signal into the logic array(s).

t_{IO} I/O input pad and buffer delay. The t_{IO} delay applies to I/O pins used as inputs. In multi-LAB MAX 5000 devices, t_{IO} is the delay from the I/O pin to the PIA. In MAX 5000 devices with a single logic array block (LAB), t_{IO} is the delay from the I/O pin to the logic arrays. In Classic devices, t_{IO} is the delay added to t_{IN}.

t_{PIA} Programmable interconnect array (PIA) delay. The delay incurred by signals that require routing through the PIA. Multi-LAB MAX 5000 devices only.

t_{SEXP} Shared expander array delay. The delay of a signal through the AND–NOT structure of the shared expander product-term array that is fed back into the logic array. MAX 5000 devices only.

t_{ICS} Global clock delay. The delay from the dedicated clock pin to a register's clock input.

t_{LAC} Logic array control delay. The AND array delay for register control functions such as preset, clear, and output enable. MAX 5000 devices only.

t_{IC} Array clock delay. The delay through a macrocell's clock product term to the register's clock input.

t_{CLR} Register clear time. The delay from the assertion of the register's asynchronous clear input to the time the register output stabilizes at logical low.

t_{PRE} Register preset time. The delay from the assertion of the register's asynchronous preset input to the time the register output stabilizes at logical high.

t_{LAD} Logic array delay. The time a logic signal requires to propagate through a macrocell's AND–OR–XOR structure.

t_{RD} Register delay. The delay from the rising edge of the register's clock to the time the data appears at the register output. MAX 5000 devices only.

t_{COMB} Combinatorial buffer delay. The delay from the time when a combinatorial logic signal bypasses the programmable register to the time it becomes available at the macrocell output. MAX 5000 devices only.

t_{LATCH} Latch delay. The propagation delay through the programmable register when t_{LATCH} is configured as a flow-through latch. MAX 5000 devices only.

t_{SU} Register setup time. The time required for a signal to be stable at the register input before the register clock's rising edge to ensure that the register correctly stores the input data.

t_{H} Register hold time. The time required for a signal to be stable at the register input after the register clock's rising edge to ensure that the register correctly stores the input data.

t_{FD} Feedback delay. In single-LAB MAX 5000 devices, t_{FD} is the delay of a macrocell output fed back into the logic array. In multi-LAB MAX 5000 devices, t_{FD} is the delay of a macrocell output fed back into the LAB's logic array or to a PIA input. In Classic devices, t_{FD} is the delay of a macrocell output fed back into the logic array.

t_{OD} Output buffer and pad delay.

t_{XZ} Output buffer disable delay. The delay required for high impedance to appear at the output pin after the output buffer's enable control is disabled.

t_{ZX} Output buffer enable delay. The delay required for the output signal to appear at the output pin after the tri-state buffer's enable control is enabled.

External Timing Parameters

External timing parameters represent actual pin-to-pin timing characteristics. Each external timing parameter consists of a combination of internal timing parameters. The data sheet for each device gives the values of the external timing parameters. These external timing parameters are worst-case values, derived from extensive performance measurements and ensured by testing. All external timing parameters are shown in bold type. The following list defines external timing parameters for MAX 5000 and Classic devices. Classic devices include the EP610, EP610I, EP910, EP910I , and EP1810 devices only.

t_{PD1} Dedicated input pin to non-registered output delay. The time required for a signal on any dedicated input pin to propagate through the combinatorial logic in a macrocell and appear at an external device output pin.

t_{PD2} I/O pin input to non-registered output delay. The time required for a signal on any I/O pin input to propagate through the combinatorial logic in a macrocell and appear at an external device output pin.

t_{PZX} Tri-state to active output delay. The time required for an input transition to change an external output from a tri-state (high-impedance) logic level to a valid high or low logic level.

t_{PXZ} Active output to tri-state delay. The time required for an input transition to change an external output from a valid high or low logic level to a tri-state (high-impedance) logic level.

t_{CLR} Time to clear register delay. The time required for a low signal to appear at the external output, measured from the input transition.

t_{SU} Global clock setup time. The time that data must be present at the input pin before the global (synchronous) clock signal is asserted at the clock pin.

t_{H} Global clock hold time. The time that data must be present at the input pin after the global clock signal is asserted at the clock pin.

t_{CO1} Global clock to output delay. The time required to obtain a valid output after the global clock is asserted at the clock pin.

t_{CNT} Minimum global clock period. The minimum period maintained by a globally clocked counter.

t_{ASU} Array clock setup time. The time data must be present at an input pin before an array (asynchronous) clock signal is asserted at the input pin.

t_{AH} Array clock hold time. The time data must be present at an input pin after an array clock signal is asserted at the input pin.

t_{ACO1} Array clock to output delay. The time required to obtain a valid output after an array clock signal is asserted at an input pin.

t_{ACNT} Minimum array clock period. The minimum period maintained by a counter when it is clocked by a signal from the array.

Timing Models

Timing models are simplified block diagrams that illustrate the propagation delays through Altera devices. Logic can be implemented on different paths. You can trace the actual paths used in your design by examining the equations listed in the MAX+PLUS II Report File (**.rpt**) for the project. You can then add up the appropriate internal timing parameters to calculate the propagation delays through the device.

MAX 5000 Devices

The MAX 5000 architecture supports many functions. The macrocell array provides registered, combinatorial, or flow-through latch operation. The registers can be clocked from a global clock or through product-term array clocks, and can be asynchronously preset and cleared. Separate product terms control the output enable and logic inversion signals. The array of shared expander product terms provides additional product terms to implement complex logic.

The MAX 5000 family has single- and multi-LAB devices. Figure 1 shows the timing model for the single-LAB EPM5032 device.

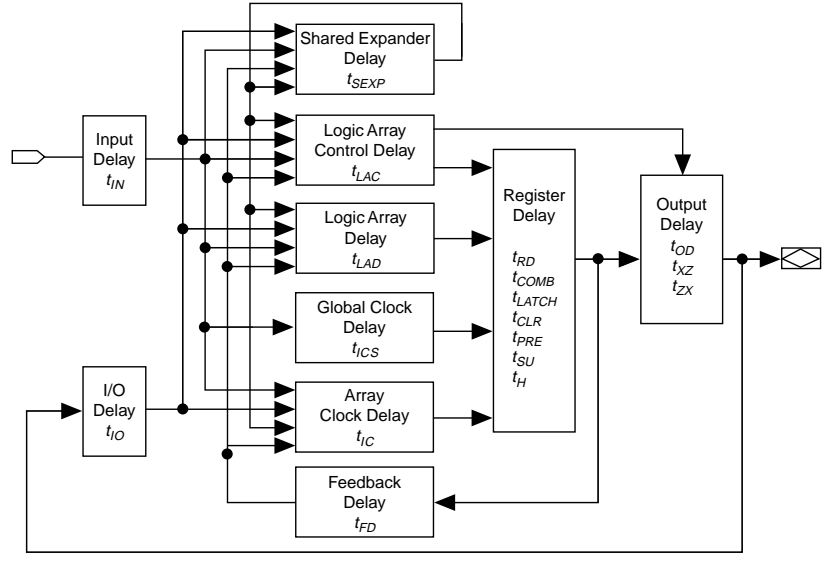*Figure 1. Single-LAB MAX 5000 Device Timing Model*

Figure 2 shows the timing model for the multi-LAB MAX 5000 devices: the EPM5064, EPM5128, EPM5130, and EPM5192 devices. In multi-LAB devices, the PIA routes signals between different LABs. All I/O inputs enter the logic array through the PIA. Signals routed through the PIA incur an additional delay.

*Figure 2. Multi-LAB MAX 5000 Device Timing Model*
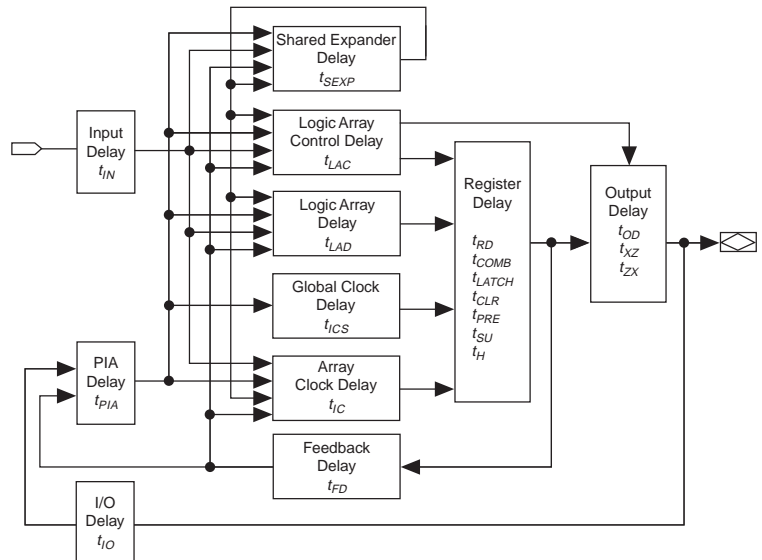
Classic Devices

The architecture for the Classic device family, which includes the EP610, EP610I, EP910, EP910I , and EP1810 devices, provides registered and combinatorial capabilities. Registers can be clocked from a global clock or through a product-term array clock, and can be asynchronously cleared. When the global clock is used, the output enable signal can be controlled by a product term. Figure 3 shows the timing model for these Classic devices.

Figure 3. Classic Device Timing Model

If the register is bypassed, the delay between the logic array and the output buffer is zero.
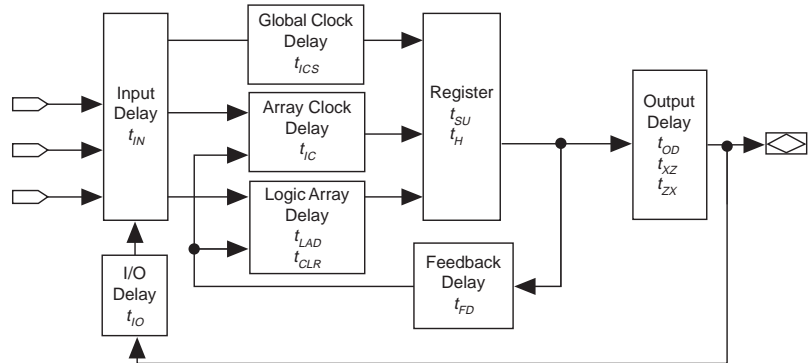
Calculating Timing Delays

You can calculate pin-to-pin timing delays for any device with the appropriate timing model and internal timing parameters. Each external timing parameter is calculated from a combination of internal timing parameters. Figure 4 shows the external timing parameters for the MAX 5000 and Classic device families. Classic devices include the EP610, EP610I, EP910, EP910I , and EP1810 devices only. To calculate the delay for a signal that follows a different path through the device, refer to the timing models shown in Figures 1 through 3 to determine which internal timing parameters to add together.

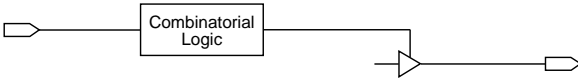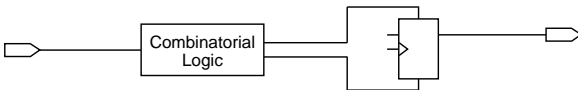*Figure 4. External Timing Parameters (Part 1 of 3)*

Combinatorial Delay
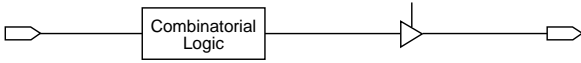
MAX 5000 (single-LAB)	t_{PD1}	=	$t_{IN} + t_{LAD} + t_{COMB} + t_{OD}$
	t_{PD2}	=	$t_{IO} + t_{LAD} + t_{COMB} + t_{OD}$
MAX 5000 (multi·LAB)	t_{PD1}	=	$t_{IN} + t_{LAD} + t_{COMB} + t_{OD}$
	t_{PD2}	=	$t_{IO} + t_{PIA} + t_{LAD} + t_{COMB} + t_{OD}$
Classic	t_{PD1}	=	$t_{IN} + t_{LAD} + t_{OD}$
	t_{PD2}	=	$t_{IO} + t_{IN} + t_{LAD} + t_{OD}$

Tri-State Enable/Disable Delay

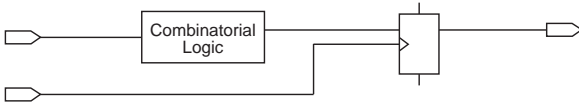| MAX 5000 | t_{PXZ}, t_{PZX} | = | $t_{IN} + t_{LAC} + (t_{XZ} \text{ or } t_{ZX})$ |
| Classic | t_{PXZ}, t_{PZX} | = | $t_{IN} + t_{LAD} + (t_{XZ} \text{ or } t_{ZX})$ |

Register Clear & Preset Time

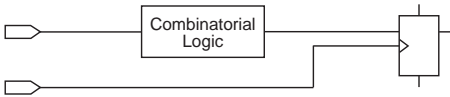| MAX 5000 | t_{PRE}, t_{CLR} | = | $t_{IN} + t_{LAC} + (t_{PRE} \text{ or } t_{CLR}) + t_{OD}$ |
| Classic | t_{CLR} | = | $t_{IN} + t_{CLR} + t_{OD}$ |

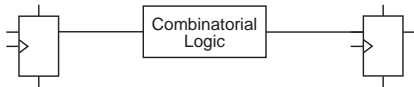*Figure 4. External Timing Parameters (Part 2 of 3)*

Setup Time

MAX 5000	t_{SU}	=	$(t_{IN} + t_{LAD}) - (t_{IN} + t_{ICS}) + t_{SU}$
Classic	t_{SU}	=	$(t_{IN} + t_{LAD}) - (t_{IN} + t_{ICS}) + t_{SU}$

Hold Time

MAX 5000	t_{H}	=	$(t_{IN} + t_{ICS}) - (t_{IN} + t_{LAD}) + t_{H}$
Classic	t_{H}	=	$(t_{IN} + t_{ICS}) - (t_{IN} + t_{LAD}) + t_{H}$

Counter Frequency

MAX 5000	t_{CNT}	=	$t_{RD} + t_{FD} + t_{LAD} + t_{SU}$
Classic	t_{CNT}	=	$t_{FD} + t_{LAD} + t_{SU}$

Asynchronous Setup Time
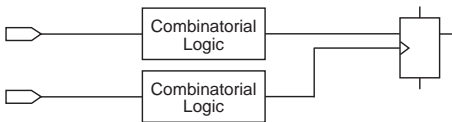
MAX 5000	t_{ASU}	=	$(t_{IN} + t_{LAD}) - (t_{IN} + t_{IC}) + t_{SU}$
Classic	t_{ASU}	=	$(t_{IN} + t_{LAD}) - (t_{IN} + t_{IC}) + t_{SU}$

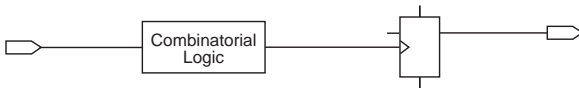*Figure 4. External Timing Parameters (Part 3 of 3)*

Asynchronous Hold Time
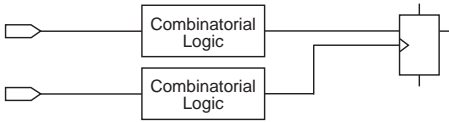
MAX 5000	t_{AH}	$=$	$(t_{IN} + t_{IC}) - (t_{IN} + t_{LAD}) + t_H$
Classic	t_{AH}	$=$	$(t_{IN} + t_{IC}) - (t_{IN} + t_{LAD}) + t_H$

Clock-to-Output Delay

MAX 5000	t_{CO1}	$=$	$t_{IN} + t_{ICS} + t_{RD} + t_{OD}$
Classic	t_{CO1}	$=$	$t_{IN} + t_{ICS} + t_{OD}$

Array Clock-to-Output Delay

MAX 5000	t_{ACO1}	$=$	$t_{IN} + t_{IC} + t_{RD} + t_{OD}$
Classic	t_{ACO1}	$=$	$t_{IN} + t_{IC} + t_{OD}$

Examples

The following examples show how to use internal timing parameters to calculate the delays for real applications.

Example 1: First Bit of 7483 TTL Macrofunction

You can analyze the timing delays for macrofunctions that have been subjected to minimization and logic synthesis. A MAX+PLUS II Report File that includes the optional Equations Section lists the synthesized logic equations for the project. These equations are structured so you can quickly determine the logic implementation of any signal.

For MAX 5000 devices, Figure 5 shows part of a 7483 TTL macrofunction (a 4-bit full adder). The Report File gives the following equations for s1, the least significant bit of the adder:

```
s1    = OUTPUT (_LC021 , VCC);
_LC021 = LCELL  (_EQ026 $ C0);
_EQ026 = b1 & !a1
         # !b1 & a1;
```

Figure 5. Adder Logic Timing for MAX 5000 Architecture
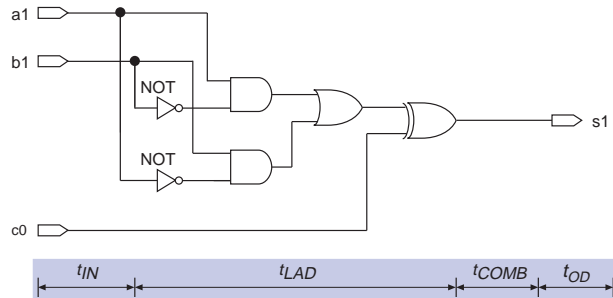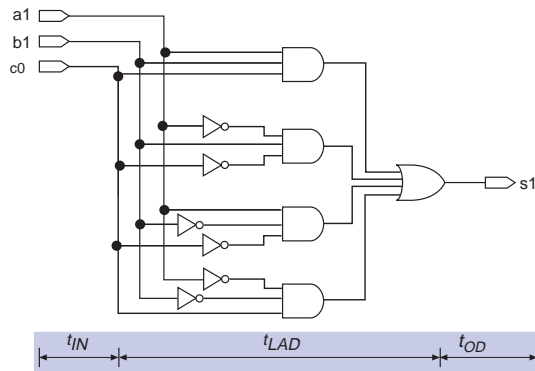

The s1 output is the output of macrocell 21 (_LC021), which contains combinatorial logic. The combinatorial logic LCELL(_EQ026 $ C0) represents the XOR of the intermediate equation _EQ026 and the carry-in, c0. In turn, _EQ026 is logically equivalent to the XOR of inputs b1 and a1. Therefore, the timing delay for s1 in MAX 5000 devices is as follows:

$$t_{IN} + t_{LAD} + t_{COMB} + t_{OD}$$

For Classic devices, Figure 6 shows part of a 7483 TTL macrofunction (a 4-bit full adder). The Report File gives the following equations for s1, the least significant bit of the adder:

```
S1      = LCELL(_EQ002);
_EQ002 = A1 & B1 &  C0
         # !A1 & B1 & !C0
         # A1 & !B1 & !C0
         # !A1 & !B1 & C0;
```

Figure 6. Adder Logic Timing for Classic Architecture

The s1 output is the output of the macrocell which contains the combinatorial logic. The _EQ002 represents the equation that logically represents the synthesized implementation of a1, b1, and c0. Therefore, the timing delay for s1 in Classic devices is as follows:

$$t_{IN} + t_{LAD} + t_{OD}$$

Example 2: Second Bit of 7483 TTL Macrofunction

For complex logic that requires expanders (represented as _X<*number*> in Report Files), the expander array delay, t_{SEXP}, is added to the delay element.

For MAX 5000 devices, the second bit of the 7483 adder macrofunction, s2, requires shared expanders. The equations are as follows:

```
s2      = _LC019;
_LC019  = LCELL(_EQ023 $ _EQ024);
_EQ023  = _X029 & _X030 & _X031;
_X029   = EXP(!b1 & !a1);
_X030   = EXP(!b1 & !c0);
_X031   = EXP(!a1 & !c0);
_EQ024  = _X032 & _X033;
_X032   = EXP(!b2 & a2);
_X033   = EXP(b2 & a2);
```

Figure 7 shows how you can map the logic structure onto the MAX 5000 architecture with these equations. The timing delay for s2 in MAX 5000 devices is shown below:

$$t_{IN} + t_{SEXP} + t_{LAD} + t_{COMB} + t_{OD}$$

Figure 7. Adder Equations Mapped to MAX 5000 Architecture
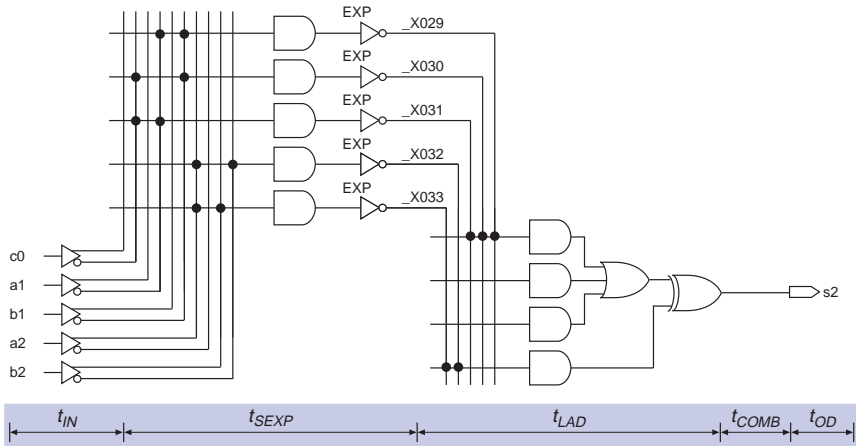
For Classic devices, the second bit of the 7483 adder macrofunction, s2, requires shared expanders. The equations are as follows:

```
S2   = LCELL(_EQ003);
_EQ003 = A2 & B1 & B2 & C0
          # A1 & A2 & B2 & !_LC017
          # !A2 & B1 & !B2 & C0
          # A1 & !A2 & !B2 & !_LC017
          # !A2 & !B1 & B2 & !_LC018
          # !A1 & !A2 & B2 & !C0
          # A2 & !B1 & !B2 & !_LC018
          # !A1 & A2 & !B2 & !C0;
_LC017 = LCELL(_EQ010);
_EQ010 = !B1 & !C0;
_LC018 = LCELL(_EQ011);
_EQ011 = A1 & C0;:
```

Figure 8 shows how you can map the logic structure onto the Classic architecture with these equations. The timing delay for s2 in Classic devices is shown below:

$$t_{IN} + t_{LAD} + t_{FD} + t_{LAD} + t_{OD}$$

*Figure 8. Adder Equations Mapped to Classic Architecture*
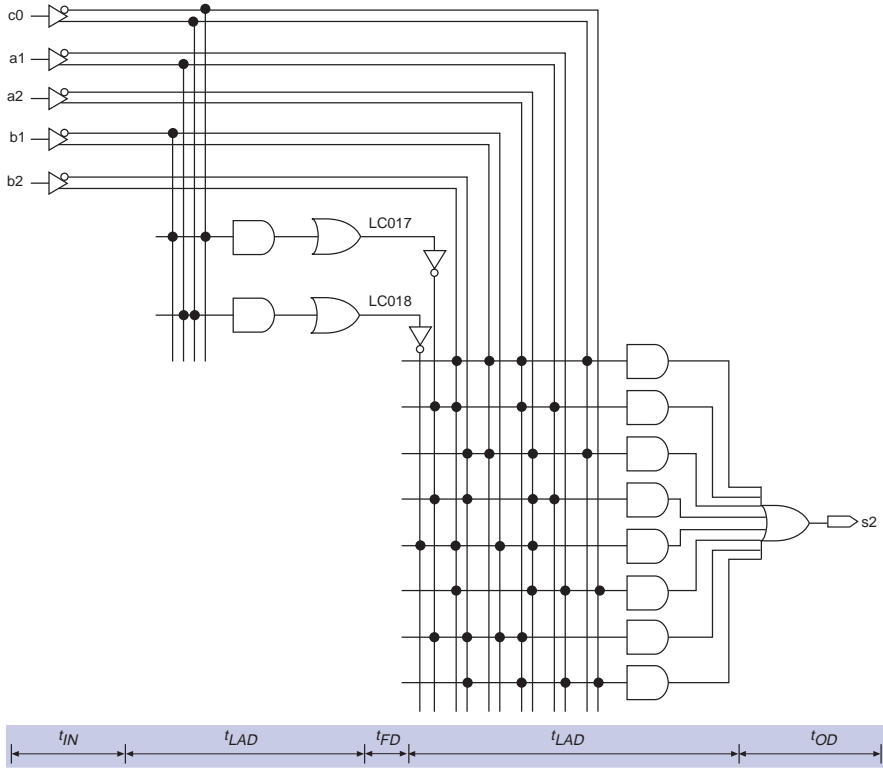


## Example 3: First Bit of 7483 TTL Macrofunction in Low-Power Mode (Classic Devices)

If a Classic device macrocell is set for low-power mode, you must add the low-power adder delay to the total delay through that macrocell. Thus, the s1 delay in Figure 6 is as follows:

$$t_{IN} + t_{LPA} + t_{LAD} + t_{OD}$$

**Conclusion**

The MAX 5000 and Classic device architectures have fixed internal timing delays that are independent of routing. Therefore, you can determine the worst-case timing delays for any design before programming a device. Total delay paths can be expressed as the sums of internal timing delays. Timing models illustrate the internal delay paths for devices and show how these internal timing parameters affect each other. You can use the MAX+PLUS II Timing Analyzer to automatically calculate delay paths, or hand-calculate delay paths by adding the internal timing parameters for an appropriate timing model. With the ability to predict worst-case timing delays, you can be confident of a design's in-system timing performance.