Understanding FLEX 8000 Timing

March 1995, ver. 1 Application Brief 143

Introduction

Altera FLEX 8000 devices provide predictable performance that is consistent from simulation to application. Before configuring a device, you can determine 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 brief and the timing parameters listed in the *FLEX 8000 Programmable Logic Device Family Data Sheet* in this data book.

☞ For the most precise results, you should use the MAX+PLUS II Timing Analyzer, which accounts for the effects of secondary factors such as placement and fan-out.

This application brief defines device internal delay parameters and AC timing characteristics and shows the timing model for Altera FLEX 8000 devices.

Familiarity with FLEX 8000 architecture and characteristics is assumed. Refer to the *FLEX 8000 Programmable Logic Device Family Data Sheet* in this data book for a complete description of the FLEX 8000 architecture and for specific values for timing parameters.

Internal FLEX 8000 Delay Parameters

Timing delays contributed by individual architectural elements are called internal delay parameters, or microparameters. All microparameters are shown in italics. The following list defines microparameters for FLEX 8000 devices.

t_{IN} I/O input pad and buffer delay. The time required for a signal on an I/O pin used as an input to reach a row or column channel of the FastTrack Interconnect.

t_{DIN_D} Dedicated input data delay. The time required for a signal used as a data input to reach a logic element (LE) from a dedicated input pin. The t_{DIN_D} delay is a function of fan-out and the distance between the source pin and destination LEs. The value shown in the *FLEX 8000 Programmable Logic Device Data Sheet* is the longest delay possible for a pin with a fan-out of four LEs. The value generated by the MAX+PLUS II Timing Analyzer is more accurate because it includes information on the fan-out and

	the relative locations of the source pin and destination LEs of the design.
t_{DIN_C}	Dedicated input control delay. The delay of a signal coming from a dedicated input pin that is used as an LE register control. These signals include the Clock, Clear, and Preset inputs to the LE register.
t_{DIN_IO}	Dedicated input I/O control delay. The delay of a signal from a dedicated input pin that is used as an I/O element (IOE) register control. These signals include the Clock and Clear inputs to the IOE register, in addition to the Output Enable control of the IOE's tri-state buffer.
t_{COL}	FastTrack Interconnect column delay. The delay incurred by a signal that requires routing through a column channel in the FastTrack Interconnect.
t_{ROW}	FastTrack Interconnect row delay. The delay incurred by a signal that requires routing through a row channel in the FastTrack Interconnect. The t_{ROW} delay is a function of fan-out and the distance between the source and destination LEs. The value shown in the *FLEX 8000 Programmable Logic Device Data Sheet* is the longest delay possible for an LE with a fan-out of four LEs. The value generated by the MAX+PLUS II Timing Analyzer is more accurate because it includes information on the fan-out and the relative locations of the source and destination LEs of the design.
t_{LOCAL}	Local interconnect delay. The delay incurred by a signal routed between LEs in the same Logic Array Block (LAB).
$t_{LABCARRY}$	Carry chain delay to the next LAB. The delay incurred by a carry-out signal that carries into the next LAB in the row.
$t_{LABCASC}$	Cascade chain delay to the next LAB. The delay incurred by a cascade-out signal that cascades into the next LAB in the row.
t_{LUT}	Look-up table (LUT) delay. The delay incurred by generating an LUT output from a signal from the local LAB interconnect.
t_{RLUT}	LUT for LE feedback delay. The time required for the output of an LE to be fed back and used to generate the LUT output in the same LE.

t_{CLUT}	LUT for carry chain delay. The delay incurred by a carry chain signal that is used to generate the LUT output.
t_{CGEN}	Carry-out generation delay. The delay incurred by generating a carry-out signal from a local LAB interconnect signal.
t_{CGENR}	Carry-out generation using LE feedback delay. The delay incurred by generating a carry-out signal from the feedback of the LE.
t_{CICO}	Carry-in, carry-out delay. The delay incurred by generating a carry-out signal that uses the carry-in signal from the previous LE.
t_{C}	Register control delay. The time required for a signal to be routed to the Clock, Preset, or Clear input of an LE register.
t_{GATE}	Cascade gate delay. The time required for a signal to pass through the cascade-generating AND gate in the LE. This delay is incurred, regardless of whether or not the cascade output is used.
t_{CASC}	Cascade chain delay. The time required for a cascade-out signal to be routed to the next LE in the same LAB. This delay, along with $t_{LABCASC}$, is also used to calculate the delay for a cascade-out signal to be routed to an LE in the next LAB in the row.
t_{CO}	LE Clock-to-output delay. The delay from the rising edge of the LE register's Clock to the time the data appears at the register output.
t_{COMB}	Combinatorial output delay. The time required for a combinatorial signal to bypass the LE register and become the output of the LE.
t_{SU}	LE register setup time. The time that a signal is required to be stable at the LE register input before the register Clock's rising edge to ensure that the register correctly stores the input data.
t_{H}	LE register hold time. The time that a signal is required to be stable at the LE register input after the register Clock's rising edge to ensure that the register correctly stores the input data.

t_{PRE}	LE register Preset delay. The delay from the assertion of the LE register's asynchronous Preset input to the time the register output stabilizes at a logical high.
t_{CLR}	LE register Clear delay. The delay from the assertion of the LE register's asynchronous Clear input to the time the register output stabilizes at a logical low.
t_{IOD}	Output data delay. The delay incurred by a signal routed from the FastTrack Interconnect to an IOE.
t_{IOC}	IOE control delay. The delay for a signal used to control the I/O register's Clock or Clear input, or for the Output Enable control of the IOE's tri-state buffer.
t_{IOCO}	I/O register Clock-to-output delay. The delay from the rising edge of the I/O register's Clock to the time the data appears at the register output.
t_{IOCOMB}	I/O register bypass delay. The delay for a combinatorial signal to bypass the I/O register.
t_{IOSU}	I/O register setup time. The time required for a signal to be stable at the I/O register input before the register Clock's rising edge to ensure that the register correctly stores the input data.
t_{IOH}	I/O register hold time. The time required for a signal to be stable at the I/O register input after the register Clock's rising edge to ensure that the register correctly stores the input data.
t_{IOCLR}	I/O register Clear delay. The delay from the time the I/O register's asynchronous Clear input is asserted to the time the register output stabilizes at logical low.
t_{OD1}	Output buffer and pad delay with the Slow Slew Rate logic option turned off and V_{CCIO} = 5.0 V.
t_{OD2}	Output buffer and pad delay with the Slow Slew Rate logic option turned off and V_{CCIO} = 3.3 V.
t_{OD3}	Output buffer and pad delay with the Slow Slew Rate logic option turned on.

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

t_{ZX1} Output buffer Enable delay with the Slow Slew Rate logic option turned off and V_{CCIO} = 5.0 V. The delay required for the output signal to appear at the output pin after the tri-state buffer's Enable control is enabled.

t_{ZX2} Output buffer Enable delay with the Slow Slew Rate logic option turned off and V_{CCIO} = 3.3 V. The delay required for the output signal to appear at the output pin after the tri-state buffer's Enable control is enabled.

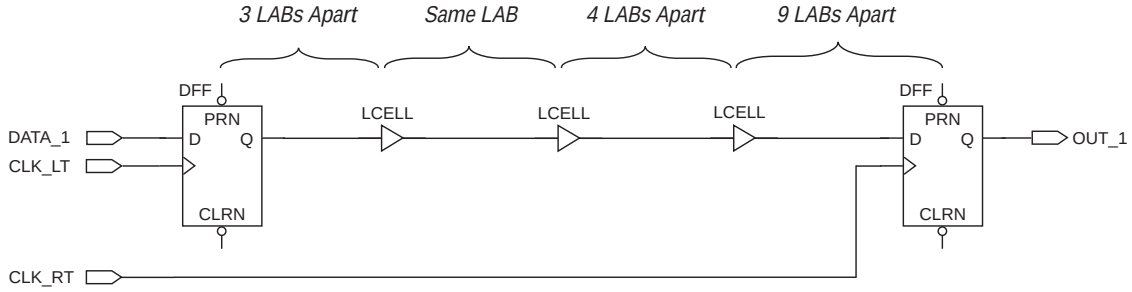
t_{ZX3} Output buffer Enable delay with the Slow Slew Rate logic option turned on and V_{CCIO} = 5.0 V or 3.3 V. The delay required for the output signal to appear at the output pin after the tri-state buffer's Enable control is enabled.

External AC Timing Characteristics

External AC timing characteristics, called macroparameters, represent actual pin-to-pin timing characteristics. Each macroparameter consists of a combination of internal delay elements (microparameters). All macroparameters are shown in bold type. One timing macroparameter, **t_{DRR}**, characterizes the AC operating specifications. This is a worst-case value, derived from extensive performance measurements and guaranteed by testing. Other macroparameters can be estimated by using the timing model or the equations in "Calculating Timing Delays" later in this application brief.

t_{DRR} Register-to-register delay. The average time required for the signal from one register to pass through four LEs via three row interconnects and four local interconnects. The test circuit used for this parameter is a register with an output that goes through three LCELL primitives in two different LABs; the last LCELL goes to another register in another LAB. Figure 1 shows this path. The full circuit contains multiple copies of this path, and the registers and LCELL primitives are assigned with the same relationship in each of the paths. That relationship is also described in Figure 1.

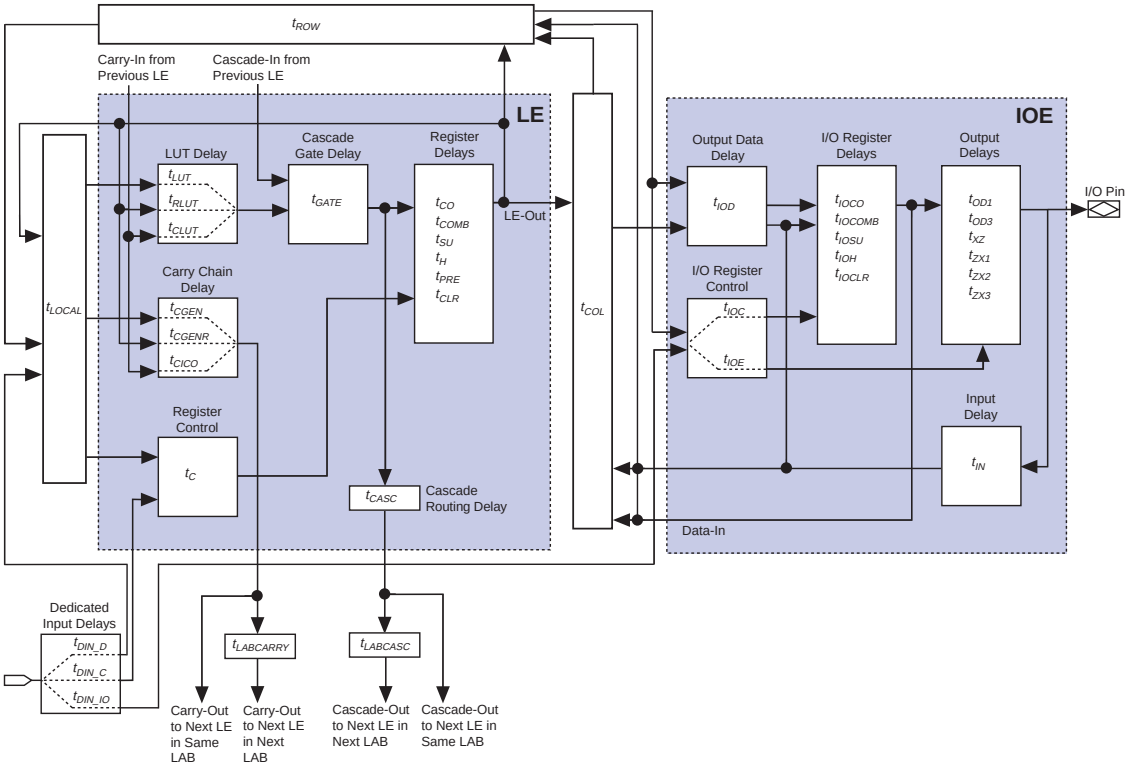
Figure 1. Path for t_{DRR} Circuit for 21-Column Devices



FLEX 8000 Timing Model

Timing models are simplified block diagrams that illustrate propagation delays through Altera devices. Logic can be implemented on different paths. You can trace the actual paths used in your FLEX 8000 device by examining the equations listed in the MAX+PLUS II Report File (**.rpt**) for the project. You can then add up the appropriate microparameters to calculate the approximate propagation delays through the FLEX 8000 device. However, for the most precise delay information, you should use the MAX+PLUS II Timing Analyzer. Figure 2 shows the timing model for FLEX 8000 devices.

Figure 2. FLEX 8000 Timing Model



Calculating Timing Delays

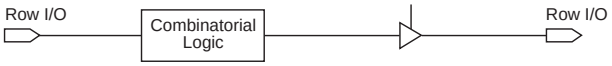
You can calculate approximate pin-to-pin timing delays for FLEX 8000 devices with the timing model and the internal delay parameters in the *FLEX 8000 Programmable Logic Device Family Data Sheet* in this data book. Each AC timing macroparameter is calculated from a combination of internal delays. Figure 3 shows the FLEX 8000 family LE macroparameters (shown in bold type). To calculate the delay for a signal that follows a different path through the FLEX 8000 device, refer to the timing model to determine which microparameters (shown in italic type) to add together.

Figure 3. Logic Element AC Timing Parameters (Part 1 of 3)

Combinatorial Delay

t_1

t_2



From I/O Inputs:

$t_1 = t_{IN} + t_{ROW} + t_{LOCAL} + t_{LUT} + t_{GATE} + t_{COMB} + t_{COL} + t_{IOD} + t_{IOCOMB} + t_{OD1}$

$t_2 = t_{IN} + t_{ROW} + t_{LOCAL} + t_{LUT} + t_{GATE} + t_{COMB} + t_{ROW} + t_{IOD} + t_{IOCOMB} + t_{OD1}$

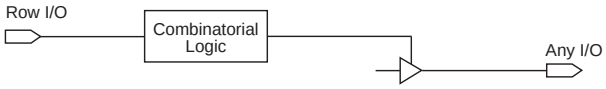
From Dedicated Inputs:

$t_1 = t_{DIN\_D} + t_{LOCAL} + t_{LUT} + t_{GATE} + t_{COMB} + t_{COL} + t_{IOD} + t_{IOCOMB} + t_{OD1}$

$t_2 = t_{DIN\_D} + t_{LOCAL} + t_{LUT} + t_{GATE} + t_{COMB} + t_{ROW} + t_{IOD} + t_{IOCOMB} + t_{OD1}$

Tri-State Enable/Disable Delay

t_{XZ} or **t_{ZX}**



From Row I/O Inputs:

$t_{XZ}, t_{ZX} = t_{IN} + t_{ROW} + t_{LOCAL} + t_{LUT} + t_{GATE} + t_{COMB} + t_{ROW} + t_{IOC} + (t_{XZ}$ or $t_{ZX1})$

From Dedicated Inputs:

$t_{XZ}, t_{ZX} = t_{DIN\_IO} + t_{IOE} + (t_{XZ}$ or $t_{ZX1})$

Figure 3. Logic Element AC Timing Parameters (Part 2 of 3)

LE Register Clear & Preset Time

t_{CLR}

t_{PRE}

From I/O Inputs to Row or Column Outputs:

$$t_{CLR} = t_{IN} + t_{ROW} + t_{LOCAL} + t_C + t_{CLR} + (t_{ROW} \text{ or } t_{COL}) + t_{IOD} + t_{IOCOMB} + t_{OD1}$$

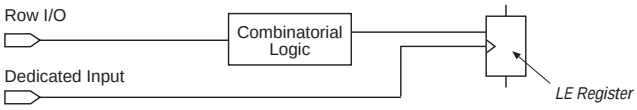
$$t_{PRE} = t_{IN} + t_{ROW} + t_{LOCAL} + t_C + t_{PRE} + (t_{ROW} \text{ or } t_{COL}) + t_{IOD} + t_{IOCOMB} + t_{OD1}$$

From Dedicated Inputs to Row or Column Outputs:

$$t_{CLR} = t_{DIN\_C} + t_C + t_{CLR} + (t_{ROW} \text{ or } t_{COL}) + t_{IOD} + t_{IOCOMB} + t_{OD1}$$

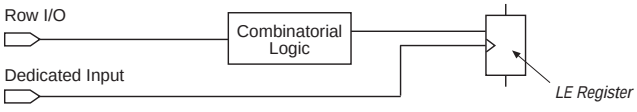
$$t_{PRE} = t_{DIN\_C} + t_C + t_{PRE} + (t_{ROW} \text{ or } t_{COL}) + t_{IOD} + t_{IOCOMB} + t_{OD1}$$

Register Setup Time from a Global Clock & Row I/O Data Input



$$t_{SU} = (t_{IN} + t_{ROW} + t_{LOCAL} + t_{LUT} + t_{GATE}) - (t_{DIN\_C} + t_C) + t_{SU}$$

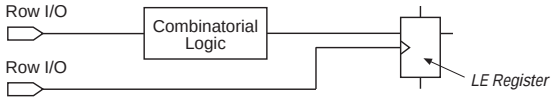
Register Hold Time from a Global Clock & Row I/O Data Input



$$t_H = (t_{DIN\_C} + t_C) - (t_{IN} + t_{ROW} + t_{LOCAL} + t_{LUT} + t_{GATE}) + t_H$$

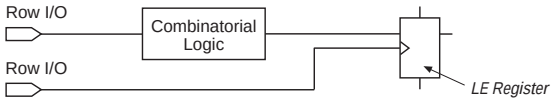
Figure 3. Logic Element AC Timing Parameters (Part 3 of 3)

Asynchronous Setup Time from a Row I/O Clock & Row I/O Data Input



$$t_{ASU} = (t_{IN} + t_{ROW} + t_{LOCAL} + t_{LUT} + t_{GATE}) - (t_{IN} + t_{ROW} + t_{LOCAL} + t_C) + t_{SU}$$

Asynchronous Hold Time from a Row I/O Clock & Row I/O Data Input



$$t_{AH} = (t_{IN} + t_{ROW} + t_{LOCAL} + t_C) - (t_{IN} + t_{ROW} + t_{LOCAL} + t_{LUT} + t_{GATE}) + t_H$$

Clock-to-Output Delay from a Global Clock to Any Output

$$t_{CO} = t_{DIN\_C} + t_C + t_{CO} + (t_{ROW} \text{ or } t_{COL}) + t_{IOD} + t_{IOCOMB} + t_{OD1}$$

Asynchronous Clock-to-Output Delay from a Row I/O Clock to Any Output

$$t_{ACO} = t_{IN} + t_{ROW} + t_{LOCAL} + t_C + t_{CO} + (t_{ROW} \text{ or } t_{COL}) + t_{IOD} + t_{IOCOMB} + t_{OD1}$$

Figure 4 shows the FLEX 8000 family I/O element macroparameters. To calculate the delay for a signal that follows a different path through the FLEX 8000 device, refer to the timing model to determine which microparameters to add together.

Figure 4. I/O Element AC Timing Parameters (Part 1 of 2)

I/O Element Clear Time

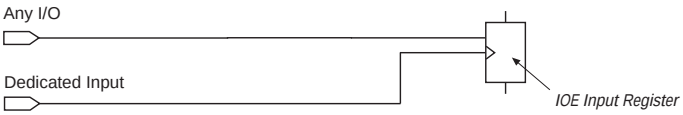
From I/O Inputs:

$$t_{CLR} = t_{IN} + t_{ROW} + t_{IOC} + t_{IOCLR} + t_{OD1}$$

From Dedicated Inputs:

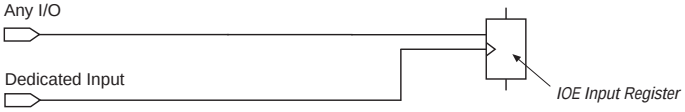
$$t_{CLR} = t_{DIN\_IO} + t_{IOC} + t_{IOCLR} + t_{OD1}$$

Register Setup Time from a Global Clock & Row I/O Data Input



$$t_{SU} = t_{IN} - (t_{DIN\_IO} + t_{IOC}) + t_{IOSU}$$

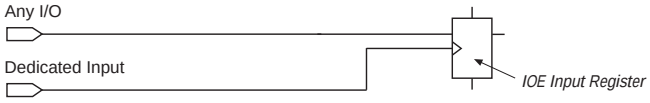
Register Hold Time from a Global Clock & Row I/O Data Input



$$t_{H} = (t_{DIN\_C} + t_{IOC}) - t_{IN} + t_{H}$$

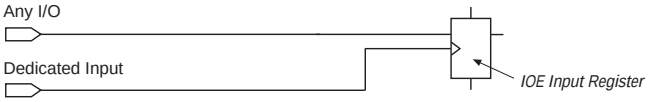
Figure 4. I/O Element AC Timing Parameters (Part 2 of 2)

Asynchronous Setup Time from a Row I/O Clock & Row I/O Data Input



$$t_{ASU} = t_{IN} - (t_{IN} + t_{ROW} + t_{IOC}) + t_{SU}$$

Asynchronous Hold Time from a Row I/O Clock & Row I/O Data Input



$$t_{AH} = (t_{IN} + t_{ROW} + t_{IOC}) - t_{IN} + t_{H}$$

Clock-to-Output Delay from a Global Clock to Any Output

$$t_{CO} = t_{DIN\_IO} + t_{IOC} + t_{IOCO} + t_{OD1}$$

Asynchronous Clock-to-Output Delay from a Row I/O Clock to Any Output

$$t_{ACO} = t_{IN} + t_{ROW} + t_{IOC} + t_{IOCO} + t_{OD1}$$

Timing Model vs. MAX+PLUS II Timing Analyzer

The MAX+PLUS II Timing Analyzer always provides the most accurate information on the performance of a design. However, hand calculations based on the timing model also provide a good estimate of the design performance. The MAX+PLUS II Timing Analyzer is more accurate because it takes into account three secondary factors that influence the t_{ROW} and t_{DIN_D} parameters:

- Fan-out for each signal in the delay path
- Positions of other loads relative to the source and destination
- Distance between signal source and destination

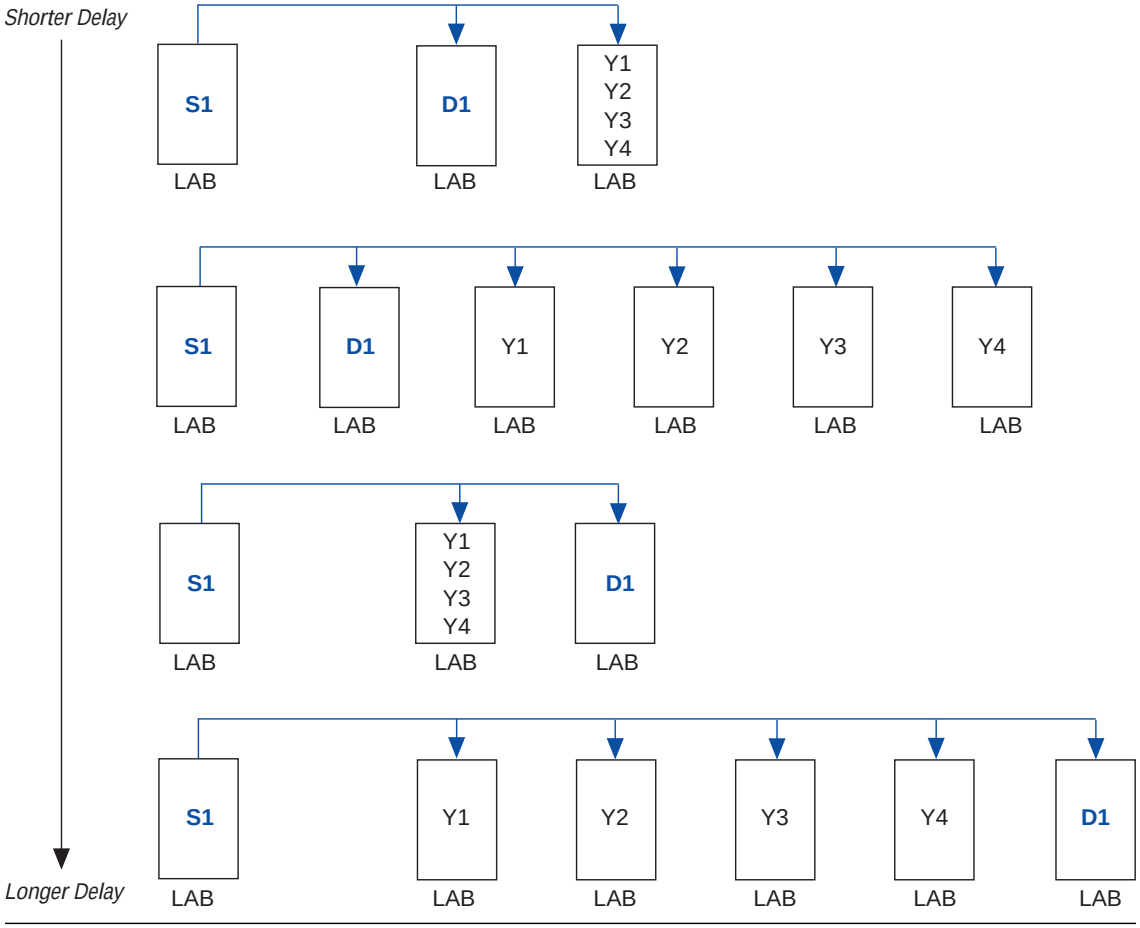
Fan-Out

The more loads a signal has to drive, the longer the delay across t_{ROW} and t_{DIN_D}. This loading is a function of the number of LABs that a signal source has to drive, as well as of the number of LEs in the LAB that use the signal. The number of LABs that a signal drives has a greater effect on the delay than the number of cells in the LAB that use the signal. For example, a signal `S1` feeds destination `D1` and feeds logic elements `Y[4..1]`. If `Y[4..1]` are in different LABs, then `S1` has four loads. If, however, they are all in the same LAB, then `S1` still has four loads, but has a shorter delay. Therefore, the delay from `S1` to `D1` is greater when each signal `Y[4..1]` is in a different LAB.

Load Distribution

The load distribution relative to the source and destination also affects the t_{ROW} and t_{DIN_D} delays. Figure 5 illustrates the change in the t_{ROW} and t_{DIN_D} delays caused by variations in the position of `D1` and the distribution of `Y[1..4]`.

Figure 5. Effect of Relative Position & Load Distribution on t_{ROW}



Distance

The distance between the source and destination LEs also affects the timing of t_{ROW} and t_{DIN_D}. For example, if S1 and D1 are pins on the left and right sides of a device, respectively, then the delay through one LCELL on the same row (i.e., the time required to traverse the length of the device) is the same no matter where the LCELL is placed. If, on the other hand, S1 and D1 are both on the left side, then the delay from S1 to D1 depends on where the LCELL is placed. If the LCELL is on the right side (far from S1 and D1), then the delay is longer than if it is on the left side (close to S1 and D1).

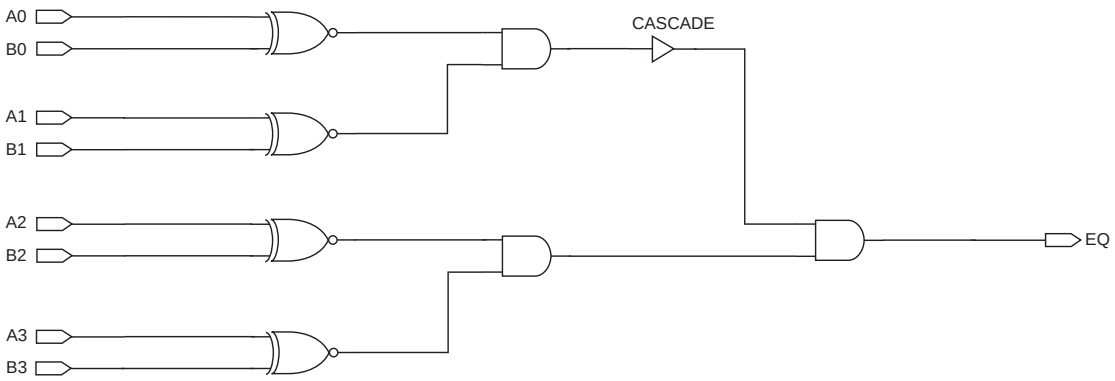
Examples

The following examples show how to use microparameters to estimate the delays for real applications.

Example 1: 4-Bit Equality Comparator with Cascade

You can analyze the timing delays for circuits that have been subjected to minimization and logic synthesis. The synthesized equations can be found in the MAX+PLUS II Report File (**.rpt**) for the project. These equations are structured so that you can quickly determine the logic configuration of any signal. For example, Figure 6 shows a 4-bit equality comparator.

Figure 6. 4-Bit Equality Comparator Circuit



The Report File for this circuit gives the equations for EQ, the output of the comparator:

```
EQ      =  _LC2_B1;
_LC2_B1 =  LCELL( _EQ002C);
_EQ002C =  _EQ002 & CASCADE( _EQ001C);
_EQ002  =  A2 & A3 & B2 & B3
           # A2 & !A3 & B2 & !B3
           # !A2 & A3 & !B2 & B3
           # !A2 & !A3 & !B2 & !B3;
```

The equation for _EQ001C cascades into the previous equation:

```
% _LC1_B1 =  LCELL( _EQ001C); %
_EQ001C   =  _EQ001;
_EQ001    =  A0 & A1 & B0 & B1
             # A0 & !A1 & B0 & !B1
             # !A0 & A1 & !B0 & B1
             # !A0 & !A1 & !B0 & !B1;
```

The output pin, EQ, is the output of the second LE of a cascade chain. The combinatorial LE, _LC1_B1, implements the comparison of the first two bits. The second two bits are implemented in the LUT of _LC2_B1. The outputs of these two LEs are then cascaded together to form the output of _LC2_B1.

If A2 and EQ are row I/O pins, then the timing delay from A2 to EQ can be estimated by adding the following parameters:

$$t_{IN} + t_{ROW} + t_{LOCAL} + t_{LUT} + t_{GATE} + t_{COMB} + t_{ROW} + t_{IOD} + t_{IOCOMB} + t_{OD1}$$

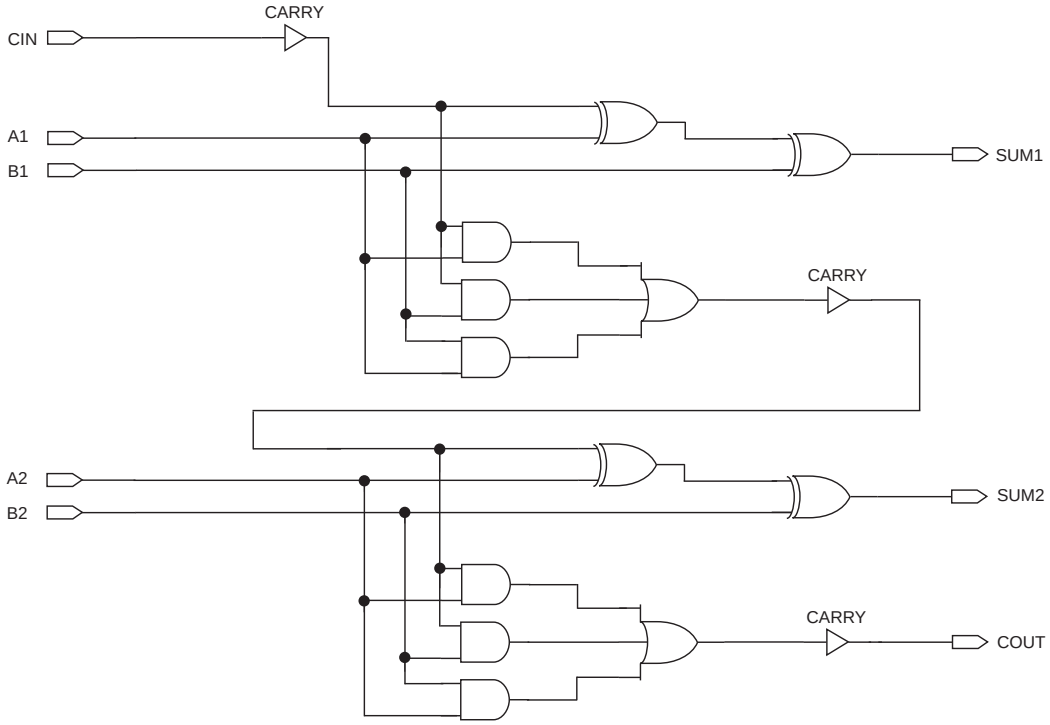
If A0 is a row I/O pin, the timing delay from A0 to EQ can be estimated by adding the following parameters:

$$t_{IN} + t_{ROW} + t_{LOCAL} + t_{LUT} + t_{GATE} + t_{CASC} + t_{GATE} + t_{COMB} + t_{ROW} + t_{IOD} + t_{IOCOMB} + t_{OD1}$$

Example 2: 2-Bit Adder Using Carry Chain

FLEX 8000 devices have specialized resources that implement complex arithmetic functions. For instance, adders and counters require a carry function to determine whether or not to increment the next significant bit. The FLEX 8000 architecture has a built-in carry chain that performs this function. The following example explains how to estimate the delay for a 2-bit adder that uses the carry chain, shown in Figure 7.

Figure 7. 2-Bit Adder Implemented with a Carry Chain



The MAX+PLUS II Report File contains the following equations for the 2-bit adder:

```
COUT            =  _LC4_B1;
SUM1            =  _LC2_B1;
SUM2            =  _LC3_B1;
_LC2_B1         =  LCELL( _EQ001);
_EQ001          =  !A1 & !B1 & _LC1_B1_CARRY
                   # A1 & !B1 & !_LC1_B1_CARRY
                   # A1 & B1 & _LC1_B1_CARRY
                   # !A1 & B1 & !_LC1_B1_CARRY;
_LC2_B1_CARRY   =  CARRY( _EQ002);
_EQ002          =  A1 & B1
                   # A1 & _LC1_B1_CARRY
                   # B1 & _LC1_B1_CARRY;
_LC3_B1         =  LCELL( _EQ003);
_EQ003          =  A2 & B2 & _LC2_B1_CARRY
                   # !A2 & B2 & !_LC2_B1_CARRY
                   # !A2 & !B2 & _LC2_B1_CARRY
                   # A2 & !B2 & !_LC2_B1_CARRY;
```

```
_LC4_B1         =  LCELL( _LC3_B1_CARRY);
_LC3_B1_CARRY   =  CARRY( _EQ004);
_EQ004          =  B2 & _LC2_B1_CARRY
                   # A2 & B2
                   # A2 & _LC2_B1_CARRY;
_LC1_B1_CARRY   =  CARRY( CIN);
```

The CIN input carries into _LC2_B1, and the output of the LE _LC2_B1 is SUM1 and _LC2_B1_CARRY. These signals are then used to generate SUM2 and _LC3_B1_CARRY. The output pin, COUT, must pass through the LCELL _LC4_B1 because a CARRY buffer cannot directly feed a pin.

If CIN to SUM1 are on a row IOE, the pin-to-pin delay between them can be estimated by adding the following parameters:

$t_{IN} + t_{ROW} + t_{LOCAL} + t_{CGEN} + t_{CLUT} + t_{GATE} + t_{COMB} + t_{ROW} + t_{IOD} + t_{IOCOMB} + t_{OD1}$

If CIN to COUT are on a row IOE, the pin-to-pin delay between them can be estimated by adding the following parameters:

$t_{IN} + t_{ROW} + t_{LOCAL} + t_{CGEN} + t_{CICO} + t_{CICO} + t_{CLUT} + t_{GATE} + t_{COMB} + t_{ROW} + t_{IOD} + t_{IOCOMB} + t_{OD1}$

Conclusion

The FLEX 8000 architecture has predictable internal timing delays that can be estimated based on signal synthesis and placement. With the FLEX 8000 timing model and the individual device parameters in the *FLEX 8000 Programmable Logic Device Family Data Sheet*, you can estimate the performance of a design before compilation. For the most accurate timing data, the MAX+PLUS II Timing Analyzer calculates delays based on secondary factors, such as capacitive loading and the distance across the FastTrack Interconnect. These methods enable you to accurately predict your design's in-system timing performance.