Understanding FLEX 10K Timing

May 1999, ver. 2 Application Note 91

Introduction

Altera® devices provide predictable performance that is consistent from simulation to application. Before configuring a device, you can determine the worst-case timing delays for any design. You can use the timing models provided in this application note along with the timing parameters listed in the *FLEX 10K Embedded Programmable Logic Family Data Sheet* and *FLEX 10KE Embedded Programmable Logic Device Family Data Sheet* in this data book to estimate design performance.

For the most precise timing 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 note defines FLEX® 10K (including FLEX 10KA and FLEX 10KE) device internal and external timing parameters, and illustrates the timing model for the FLEX 10K device family.

Familiarity with the FLEX 10K architecture and characteristics is assumed. Refer to the *FLEX 10K Embedded Programmable Logic Family Data Sheet* and *FLEX 10KE Embedded Programmable Logic Device Family Data Sheet* for a complete description of the FLEX 10K architecture and for specific values for timing parameters listed in this application note.

Internal Timing Micropara-meters

The timing delays contributed by individual FLEX 10K architectural elements, called internal timing microparameters, cannot be measured explicitly. All internal timing microparameters are shown in italic type. The following sections define the internal timing microparameters for the FLEX 10K device family.

I/O Element Timing Microparameters

The following list defines the I/O element (IOE) timing microparameters for the FLEX 10K device family.

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, enable, or clear inputs, or for the output enable control of the IOE's tri-state buffer.

t_{IOFD}	IOE register feedback delay. The time required for the output of an IOE register to reach a row or column channel of the FastTrack Interconnect.
t_{INCOMB}	IOE input pad and buffer to FastTrack Interconnect 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_{INREG}	IOE input pad and buffer to IOE register delay. The time required for a signal on an I/O pin, used as an input, to reach an IOE register data input.
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 for data and enable signals before clock. The minimum time a signal must be stable at the I/O register's data and enable inputs before the register clock's rising edge to ensure that the register correctly stores the input data. t_{IOSU} is also the minimum recovery time between deassertions of clear and the rising edge of the clock.
t_{IOH}	I/O register hold time for data and enable signals after clock. The minimum time a signal must be stable at the I/O register's data and enable inputs 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 a logic low.
t_{OD1}	Output buffer and pad delay with the slow slew rate logic option turned off and $V_{CCIO} = V_{CCINT}$.
t_{OD2}	Output buffer and pad delay with the slow slew rate logic option turned off and V_{CCIO} = low voltage.
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 $V_{CCIO} = V_{CCINT}$. 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} = low voltage. 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. The delay required for the output signal to appear at the output pin after the tri-state buffer's enable control is enabled.

Interconnect Timing Microparameters

The following list describes the routing timing microparameters for the FLEX 10K device family.

$t_{SAMELAB}$ Logic element (LE) to LE in same logic array block (LAB) delay. The delay incurred by a signal routed between LEs in the same LAB.

$t_{SAMEROW}$ FastTrack Interconnect same row delay. The delay incurred by a row IOE, LE, or embedded array block (EAB) driving a row IOE, LE or EAB in the same row. The $t_{SAMEROW}$ delay is a function of fan-out and the distance between the source and destination. The value shown in the *FLEX 10K Embedded Programmable Logic Family Data Sheet* and *FLEX 10KE Embedded Programmable Logic Device Family Data Sheet* is the longest delay possible for an LE with a fan-out of four loads. However, the value generated by the MAX+PLUS II Timing Analyzer is more accurate because it considers fan-out and the relative locations of the source and destination in the design.

☞ For more information, see "Timing Model vs. MAX+PLUS II Timing Analyzer" on page 891.

$t_{SAMECOLUMN}$ FastTrack Interconnect same column delay. The delay incurred by an LE driving an IOE in the same column. The $t_{SAMECOLUMN}$ delay is a function of fan-out and the distance between the source and destination. The value shown in the *FLEX 10K Embedded Programmable Logic Family Data Sheet* and *FLEX 10KE Embedded Programmable Logic Device Family Data Sheet* is the longest delay possible for an LE with a fan-out of four loads. However, the value generated by the MAX+PLUS II Timing Analyzer is more accurate because it considers fan-out and the relative locations of the source and destination in the design.

For more information, see "Timing Model vs. MAX+PLUS II Timing Analyzer" on page 891.

$t_{DIFFROW}$ FastTrack Interconnect delay between different rows. The delay incurred by a column IOE, LE, or EAB driving a row IOE, LE, or EAB in a different row via one column and one row channel. The $t_{DIFFROW}$ delay is a function of fan-out and the distance between the source and destination. The value shown in the *FLEX 10K Embedded Programmable Logic Family Data Sheet* and *FLEX 10KE Embedded Programmable Logic Device Family Data Sheet* is the longest delay possible for an LE with a fan-out of four loads. However, the value generated by the MAX+PLUS II Timing Analyzer is more accurate because it considers fan-out and the relative locations of the source and destination in the design.

For more information, see "Timing Model vs. MAX+PLUS II Timing Analyzer" on page 891.

$t_{TWOROWS}$ FastTrack Interconnect delay between two rows. The delay incurred by a row IOE, LE, or EAB driving a row IOE, LE, or EAB in a different row via a row channel, a column channel, and another row channel. The $t_{TWOROWS}$ delay is a function of fan-out and the distance between the source and destination. The value shown in the *FLEX 10K Embedded Programmable Logic Family Data Sheet* and *FLEX 10KE Embedded Programmable Logic Device Family Data Sheet* is the longest delay possible for an LE with a fan-out of four loads. However, the value generated by the MAX+PLUS II Timing Analyzer is more accurate because it considers fan-out and the relative locations of the source and destination in the design.

For more information, see "Timing Model vs. MAX+PLUS II Timing Analyzer" on page 891.

$t_{LEPERIPH}$ Peripheral bus delay. Routing delay for an LE or IOE driving a control signal of an IOE via the peripheral control bus. The value shown in the *FLEX 10K Embedded Programmable Logic Family Data Sheet* and *FLEX 10KE Embedded Programmable Logic Device Family Data Sheet* is the longest delay possible for an LE with a fan-out of four loads. However, the value generated by the MAX+PLUS II Timing Analyzer is more accurate because it considers fan-out and the relative locations of the source and destination in the design.

For more information, see "Timing Model vs. MAX+PLUS II Timing Analyzer" on page 891.

$t_{LABCARRY}$ Carry chain delay to a different LAB. The routing delay for a carry-out signal of an LE driving the carry-in signal of an LE in a different LAB in the same row. A carry chain longer than one LAB skips either from one even-numbered LAB to another even-numbered LAB, or from one odd-numbered LAB to another odd-numbered LAB.

$t_{LABCASC}$ Cascade chain delay to a different LAB. The routing delay for a cascade-out signal of an LE driving the cascade-in signal of an LE in a different LAB in the same row. A cascade chain longer than one LAB skips either from one even-numbered LAB to another even-numbered LAB, or from one odd-numbered LAB to another odd-numbered LAB.

$t_{DIN2IOE}$ Delay from dedicated input pin to IOE control input. The time required for a signal on a dedicated input pin to reach an IOE control input.

t_{DIN2LE} Delay from dedicated input pin to LE or EAB control input. The time required for a signal on a dedicated input pin to reach an LE or EAB control input.

$t_{DIN2DATA}$ Delay from dedicated input pin to LE or EAB data input. The time required for a signal on a dedicated input pin to reach an LE or EAB data input. The value shown in the *FLEX 10K Embedded Programmable Logic Family Data Sheet* and *FLEX 10KE Embedded Programmable Logic Device Family Data Sheet* is the longest delay possible for a pin with a fan-out of four loads. However, the value generated by the MAX+PLUS II Timing Analyzer is more accurate because it considers fan-out and the relative locations of the source and destination in the design.

For more information, see "Timing Model vs. MAX+PLUS II Timing Analyzer" on page 891.

$t_{DCLK2IOE}$ Delay from dedicated clock pin to IOE clock. The time required for a signal on a dedicated clock pin to reach an IOE clock input.

$t_{DCLK2LE}$ Delay from dedicated clock pin to LE or EAB clock. The time required for a signal on a dedicated clock pin to reach an LE or EAB clock input.

Logic Element Timing Microparameters

The following list describes the LE timing microparameters for the FLEX 10K device family.

t_{LUT} Look-up table (LUT) delay. The delay incurred by generating an LUT output from an LAB local interconnect signal.

t_{RLUT} LUT using LE feedback delay. The time required for the output of an LE register to be fed back and used to generate the LUT output in the same LE.

t_{CLUT} Carry-chain LUT delay. The delay incurred by a carry chain signal used to generate the LUT output.

t_{PACKED} Data-in to packed register delay. The delay incurred by routing an LAB local interconnect signal around the LUT to the LE register data input.

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_{EN}	LE register enable delay. The time required for a signal to be routed to the enable input of an LE register.
t_{CGENR}	Carry-out generation using LE feedback delay. The time required for the output of an LE register to be fed back and used to generate the carry-out signal in the same LE.
t_{CGEN}	Carry-out generation delay. The delay incurred by generating a carry-out signal from an LAB local interconnect signal.
t_{CICO}	Carry-in to carry-out delay. The delay incurred by generating a carry-out signal that uses the carry-in signal from the previous LE.
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 for data and enable signals before clock. The minimum time a signal must be stable at the LE register's data and enable inputs before the register clock's rising edge to ensure that the register correctly stores the input data. The t_{SU} parameter is also the minimum recovery time between deasserting the clear or preset and the rising edge of the clock.
t_{H}	LE register hold time for data and enable signals after clock. The minimum time a signal must be stable at the LE register's data and enable inputs 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 logic 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 logic low.
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 a different LAB in the same row.

EAB Timing Microparameters

The following list describes the EAB timing microparameters for the FLEX 10K device family.

$t_{EABDATA1}$ — Data or address delay to EAB for combinatorial input. The time required for a signal at the boundary of an EAB to reach an EAB data or address combinatorial input.

$t_{EABDATA2}$ — Data or address delay to EAB for registered input. The time required for a signal at the boundary of an EAB to reach an EAB data or address registered input.

t_{EABWE1} — Write enable delay to EAB for combinatorial input. The time required for a signal at the boundary of an EAB to reach an EAB write enable (WE) combinatorial input.

t_{EABWE2} — Write enable delay to EAB for registered input. The time required for a signal at the boundary of an EAB to reach an EAB WE registered input.

t_{EABCLK} — EAB register clock delay. The time required for a signal at the boundary of an EAB to be routed to the clock input of an EAB register.

t_{EABCO} — EAB register clock-to-output delay. The delay from the rising edge of the EAB register's clock to the time the data appears at the register output.

$t_{EABBYPASS}$ — Bypass register delay. The time required for a combinatorial signal to bypass an EAB register.

t_{EABSU} — EAB register setup time before clock. The minimum time a signal must be stable at the EAB register's data input before the register clock's rising edge to ensure that the register correctly stores the input data.

t_{EABH} — EAB register hold time after clock. The minimum time a signal must be stable at the EAB register's data input after the register clock's rising edge to ensure that the register correctly stores the input data.

t_{EABCH}	Clock high time. The minimum time an EAB clock signal must remain at a logic high for proper clocking of the EAB registers.
t_{EABCL}	Clock low time. The minimum time an EAB clock signal must remain at a logic low for proper clocking of the EAB registers.
t_{AA}	Address access delay. The delay from an EAB RAM address input change to an EAB RAM data output change.
t_{DD}	Data-in to data-out valid delay. The time required for data at the EAB RAM data input to propagate to the EAB RAM data output during a write cycle.
t_{WP}	Write pulse width. The minimum time WE must be held at a logic high to ensure that the EAB RAM correctly stores the input data.
t_{WDSU}	Data setup time before falling edge of write pulse. The minimum time a signal must be stable at the EAB RAM data input before the falling edge of WE to ensure that the RAM correctly stores the input data.
t_{WDH}	Data hold time after falling edge of write pulse. The minimum time a signal must be stable at the EAB RAM data input after the falling edge of WE to ensure that the RAM correctly stores the input data.
t_{WASU}	Address setup time before rising edge of write pulse. The minimum time that a signal is required to be stable at the EAB RAM address input before the rising edge of WE to ensure that the RAM correctly stores the input data.
t_{WAH}	Address hold time after falling edge of write pulse. The minimum time that a signal is required to be stable at the EAB RAM address input after the falling edge of WE to ensure that the RAM correctly stores the input data.
t_{WO}	Write enable to data output valid delay. The delay from the rising edge of WE to data, which was just written to RAM, appearing at the EAB RAM data output.
t_{EABOUT}	Data-out delay. The time required for the output of an EAB to reach a row or column channel of the FastTrack Interconnect.

Internal EAB Timing Macroparameters

Internal EAB timing macroparameters are combinations of FLEX 10K internal EAB timing microparameters. These macroparameters cannot be measured explicitly. The following list defines the internal EAB timing macroparameters for the FLEX 10K device family.

t_{EABAA} EAB address access delay. The delay from an EAB address input change to an EAB data output change.

$t_{EABRCCOMB}$ EAB asynchronous read cycle time. The minimum time required to complete an asynchronous read cycle.

$t_{EABRCREG}$ EAB synchronous read cycle time. The minimum time required to complete a synchronous read cycle.

t_{EABWP} EAB write pulse width. The minimum time the EAB WE input must be held at a logic high to ensure that the EAB RAM correctly stores the input data.

$t_{EABWCCOMB}$ EAB asynchronous write cycle time. The minimum time required to complete an asynchronous write cycle.

$t_{EABWCREG}$ EAB synchronous write cycle time. The minimum time required to complete a synchronous write cycle.

t_{EABDD} EAB data-in to data-out valid delay. The amount of time required for data at the EAB data input to propagate through the EAB RAM to the EAB data output during a write cycle.

$t_{EABDATACO}$ EAB clock-to-output delay when using output registers. The delay from the rising edge of the EAB output register clock input to the time the data appears at the EAB data output.

$t_{EABDATASU}$ EAB data/address setup time before clock when using input register. The minimum time a signal must be stable at the EAB data/address input before the EAB input register clock's rising edge to ensure that the input register correctly stores the input data.

$t_{EABDATAH}$ EAB data/address hold time after clock when using input register. The minimum time a signal must be stable at the EAB data/address input after the EAB input register clock's rising edge to ensure that the input register correctly stores the input data.

$t_{EABWESU}$ EAB WE setup time before clock when using input register. The minimum time a signal must be stable at the EAB WE input before the EAB input register clock's rising edge to ensure that the input register correctly stores the input data.

t_{EABWEH} EAB WE hold time after clock when using input register. The minimum time a signal must be stable at the EAB WE input after the EAB input register clock's rising edge to ensure that the input register correctly stores the input data.

$t_{EABWDSU}$ EAB data setup time before falling edge of write pulse when not using input registers. The minimum time a signal must be stable at the EAB data input before the falling edge of the EAB WE input to ensure that the RAM correctly stores the input data.

t_{EABWDH} EAB data hold time after falling edge of write pulse when not using input registers. The minimum time a signal must be stable at the EAB data input after the falling edge of the EAB WE input to ensure that the RAM correctly stores the input data.

$t_{EABWASU}$ EAB address setup time before rising edge of write pulse when not using input registers. The minimum time a signal must be stable at the EAB address input before the rising edge of the EAB WE input to ensure that the RAM correctly stores the input data.

t_{EABWAH} EAB address hold time after falling edge of write pulse when not using input registers. The minimum time a signal must be stable at the EAB address input after the falling edge of the EAB WE input to ensure that the RAM correctly stores the input data.

t_{EABWO} EAB write enable to data output valid delay. The delay from the rising edge of the EAB WE input to data, which was just written into RAM, appearing at the EAB data output.

External Timing Parameters

External timing parameters represent actual pin-to-pin timing characteristics. Each external timing parameter consists of a combination of internal delay elements. They are worst-case values, derived from extensive performance measurements, and are ensured by device testing or characterization. All external timing parameters are shown in bold type. For example, $\mathbf{t_{DRR}}$ is the AC operating specification. Other external timing parameters can be estimated by using the timing model or the equations in "Calculating Timing Delays" on page 884 of this application note.

t_{DRR} Register-to-register delay. The time required for the signal from one register to pass through four LEs via three row interconnects and four local interconnects to reach the D input of a second register. 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 feeds another register in another LAB. Figures 1 through 4 show this path for different FLEX 10K devices. The test circuit files are available from Altera Applications.

Figure 1. t_{DRR} Circuit Path for 24-Column FLEX 10K Devices

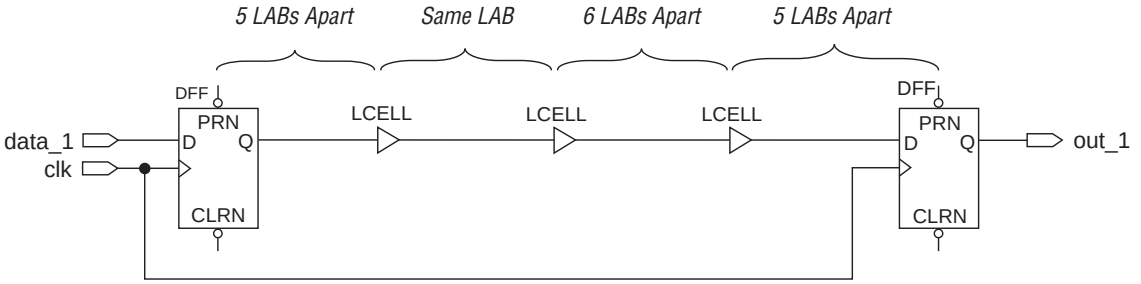


Figure 2. t_{DRR} Circuit Path for 36-Column FLEX 10K Devices

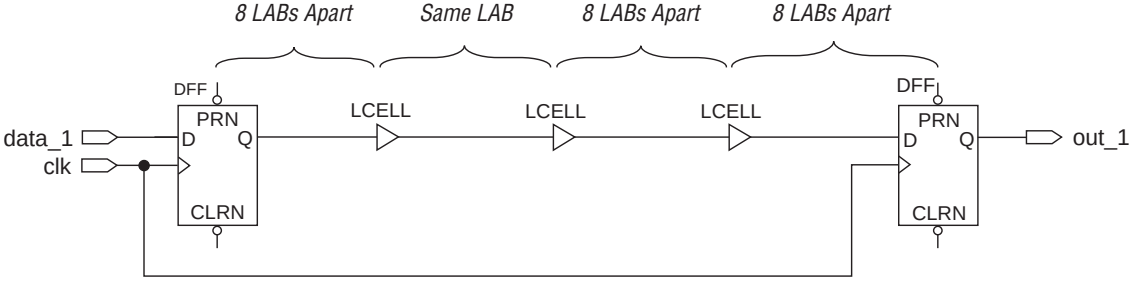


Figure 3. t_{DRR} Circuit Path for 52-Column FLEX 10K Devices

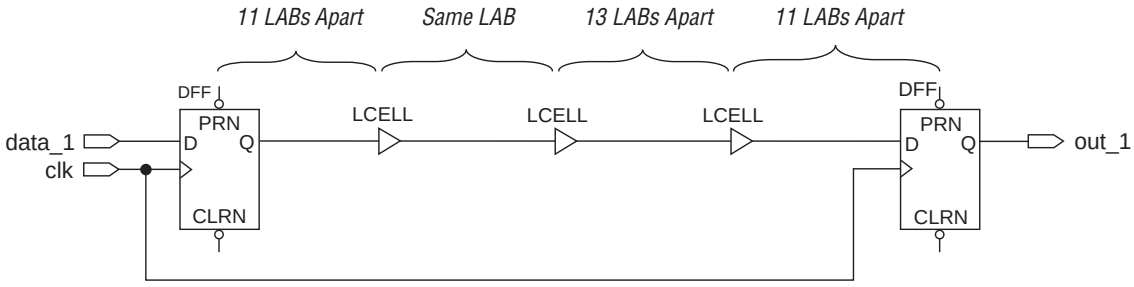
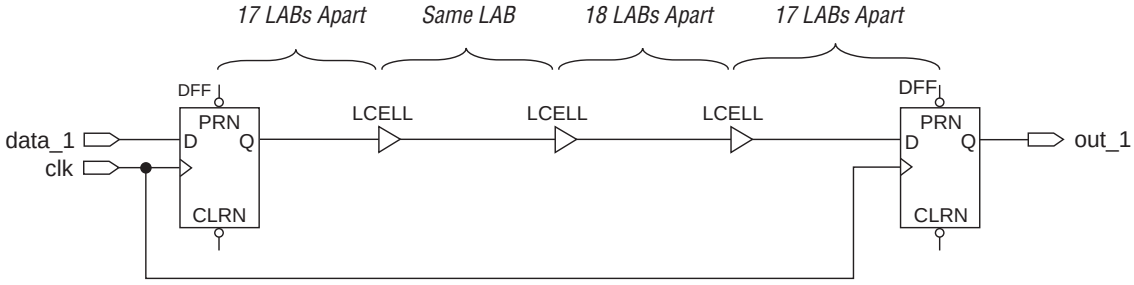


Figure 4. t_{DRR} Circuit Path for 76-Column FLEX 10K Devices



t_{INSU} Setup time with global clock at IOE register. The minimum time a signal must be stable at a pin driving an IOE register before a rising edge is applied to the global clock pin to ensure that the register correctly stores the input data.

t_{INH} Hold time with global clock at IOE register. The minimum time a signal must be stable at a pin driving an IOE register after a rising edge is applied to the global clock pin to ensure that the register correctly stores the input data.

t_{OUTCO} Clock-to-output delay with global clock at IOE register. The delay from a rising edge on the global clock pin to data appearing at an output pin driven by an IOE register.

t_{ODH} Output data delay. The minimum time a registered output pin will remain at its previous value after a rising edge is applied to the clock input pin. This parameter applies for both global and non-global clocking, and for LE, EAB, and IOE registers.

FLEX 10K Timing Model

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 FLEX 10K device by examining the equations listed in the MAX+PLUS II Report File (**.rpt**) for your project. You can then add up the appropriate internal timing parameters to calculate the approximate propagation delays through the FLEX 10K device. However, the MAX+PLUS II Timing Analyzer provides the most accurate timing information. Figures 5 through 8 show the timing models for FLEX 10K devices.

Figure 5. FLEX 10K Device Timing Model

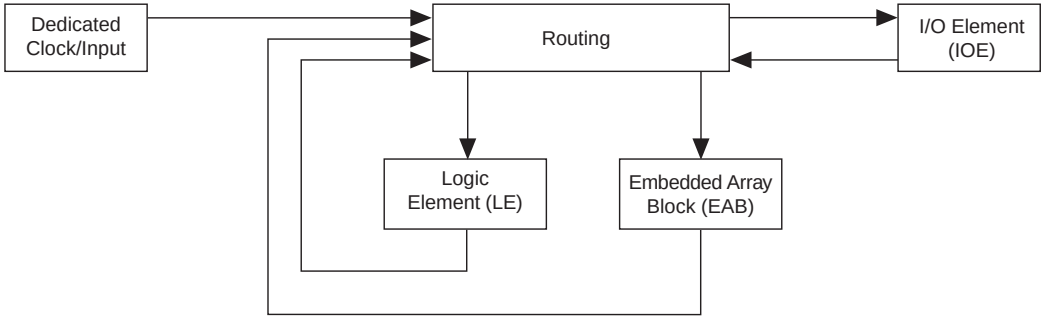


Figure 6. FLEX 10K Device LE Timing Model

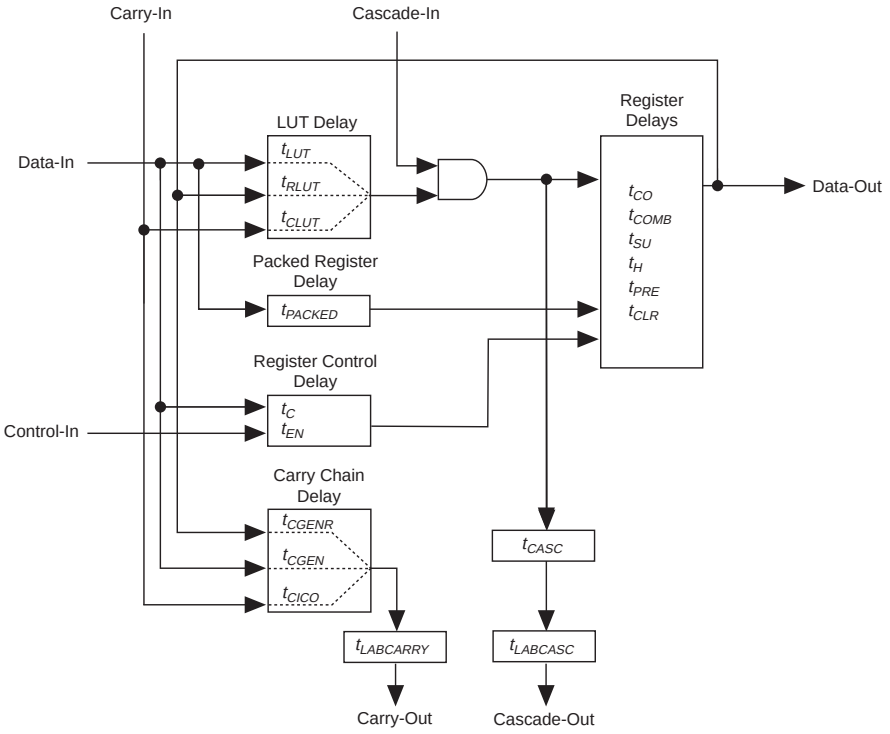


Figure 7. FLEX 10K Device IOE Timing Model

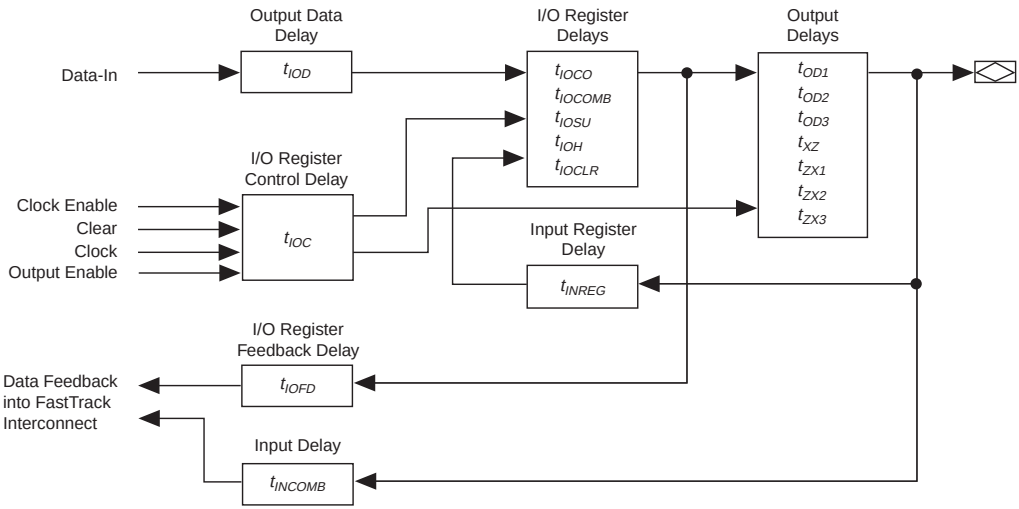
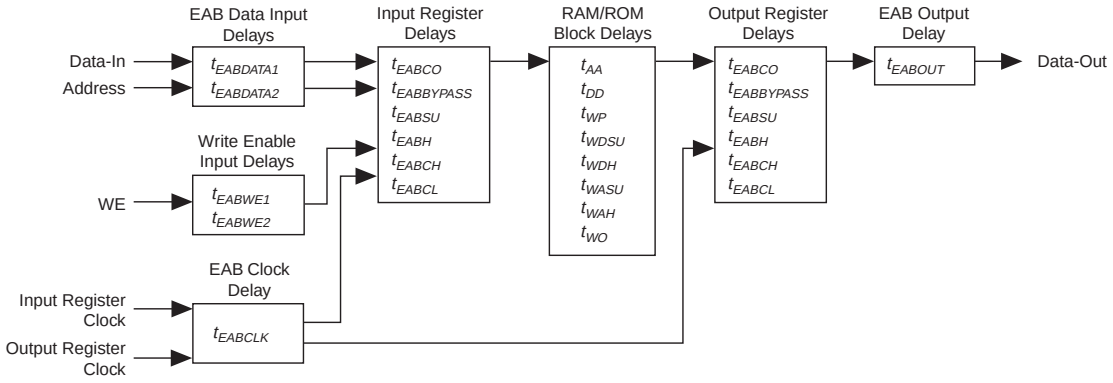


Figure 8. FLEX 10K Device EAB Timing Model



Calculating Timing Delays

You can calculate approximate pin-to-pin timing delays for FLEX 10K devices with the timing models shown in Figures 5 through 8 and the internal timing parameters listed in the *FLEX 10K Embedded Programmable Logic Family Data Sheet* in this data book. Each timing delay is calculated from a combination of internal timing parameters. Figure 9 shows the FLEX 10K device family LE timing delays. To calculate the delay for a signal that follows a different path through the FLEX 10K device, refer to the FLEX 10K timing models to determine which internal timing parameters to add together.

Figure 9. Logic Element Timing Delays (Part 1 of 4)

Combinatorial Delay

From Row I/O Inputs:

t_1

t_2

$t_1 = t_{INCOMB} + t_{SAMEROW} + t_{LUT} + t_{COMB} + t_{SAMECOLUMN} + t_{IOD} + t_{IOCOMB} + t_{OD1}$

$t_2 = t_{INCOMB} + t_{SAMEROW} + t_{LUT} + t_{COMB} + t_{SAMEROW} + t_{IOD} + t_{IOCOMB} + t_{OD1}$

From Dedicated Inputs:

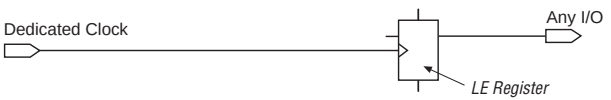
t_1

t_2

$t_1 = t_{DIN2DATA} + t_{LUT} + t_{COMB} + t_{SAMECOLUMN} + t_{IOD} + t_{IOCOMB} + t_{OD1}$

$t_2 = t_{DIN2DATA} + t_{LUT} + t_{COMB} + t_{SAMEROW} + t_{IOD} + t_{IOCOMB} + t_{OD1}$`

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



$t_{CO} = t_{DCLK2LE} + t_C + t_{CO} + (t_{SAMEROW}$ or $t_{SAMECOLUMN}) + t_{IOD} + t_{IOCOMB} + t_{OD1}$

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

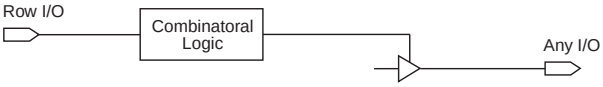
$t_{ACO} = t_{INCOMB} + t_{SAMEROW} + t_C + t_{CO} + (t_{SAMEROW}$ or $t_{SAMECOLUMN}) + t_{IOD} + t_{IOCOMB} + t_{OD1}$

Figure 9. Logic Element Timing Delays (Part 2 of 4)

Tri-State Enable/Disable Delay

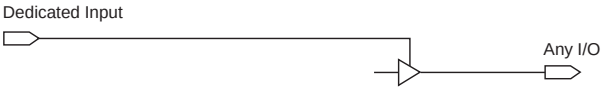
t_{XZ} or t_{ZX}

From Row I/O Inputs through Logic:



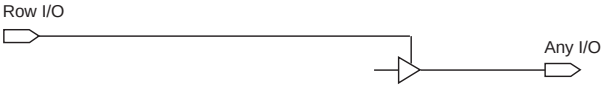
$t_{XZ}, t_{ZX} = t_{INCOMB} + t_{SAMEROW} + t_{LUT} + t_{COMB} + t_{LEPERIPH} + t_{IOC} + (t_{XZ}$ or $t_{ZX1})$

Directly from Dedicated Inputs:



$t_{XZ}, t_{ZX} = t_{DIN2IOE} + t_{IOC} + (t_{XZ}$ or $t_{ZX1})$

Directly from Row I/O Inputs:



$t_{XZ}, t_{ZX} = t_{INCOMB} + t_{LEPERIPH} + t_{IOC} + (t_{XZ}$ or $t_{ZX1})$

Figure 9. Logic Element Timing Delays (Part 3 of 4)

LE Register Clear & Preset Time

From Row I/O Inputs to Row or Column Outputs:

t_{CLR}

t_{PRE}

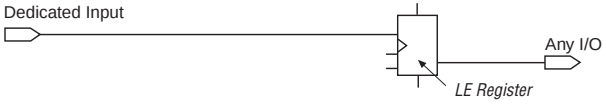
$t_{CLR} = t_{INCOMB} + t_{SAMEROW} + t_C + t_{CLR} + (t_{SAMEROW}$ or $t_{SAMECOLUMN}) + t_{IOD} + t_{IOCOMB} + t_{OD1}$

$t_{PRE} = t_{INCOMB} + t_{SAMEROW} + t_C + t_{PRE} + (t_{SAMEROW}$ or $t_{SAMECOLUMN}) + t_{IOD} + t_{IOCOMB} + t_{OD1}$

From Dedicated Inputs to Row or Column Outputs:

t_{CLR}

t_{PRE}

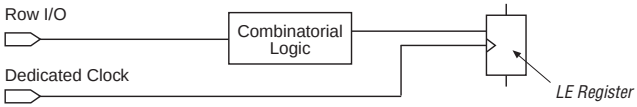


$t_{CLR} = t_{DIN2LE} + t_C + t_{CLR} + (t_{SAMEROW}$ or $t_{SAMECOLUMN}) + t_{IOD} + t_{IOCOMB} + t_{OD1}$

$t_{PRE} = t_{DIN2LE} + t_C + t_{PRE} + (t_{SAMEROW}$ or $t_{SAMECOLUMN}) + t_{IOD} + t_{IOCOMB} + t_{OD1}$

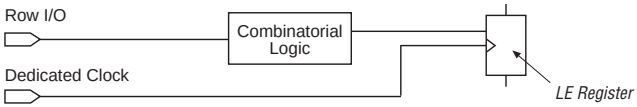
Figure 9. Logic Element Timing Delays (Part 4 of 4)

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



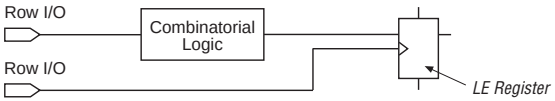
$$t_{SU} = (t_{INCOMB} + t_{SAMEROW} + t_{LUT}) - (t_{DCLK2LE} + t_C) + t_{SU}$$

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



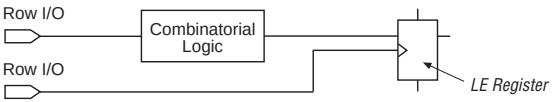
$$t_H = (t_{DCLK2LE} + t_C) - (t_{INCOMB} + t_{SAMEROW} + t_{LUT}) + t_H$$

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



$$t_{ASU} = (t_{INCOMB} + t_{SAMEROW} + t_{LUT}) - (t_{INCOMB} + t_{SAMEROW} + t_C) + t_{SU}$$

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



$$t_{AH} = (t_{INCOMB} + t_{SAMEROW} + t_C) - (t_{INCOMB} + t_{SAMEROW} + t_{LUT}) + t_H$$

Figure 10 shows the FLEX 10K device family IOE timing delays. To calculate the delay for a signal that follows a different path through the FLEX 10K device, refer to the FLEX 10K timing model to determine which internal timing parameters to add together.

Figure 10. I/O Element Timing Delays (Part 1 of 2)

I/O Element Clear Time

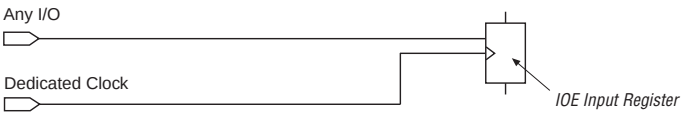
From Row I/O Inputs:

$$t_{CLR} = t_{INCOMB} + t_{LEPERIPH} + t_{IOC} + t_{IOCLR} + t_{OD1}$$

From Dedicated Inputs:

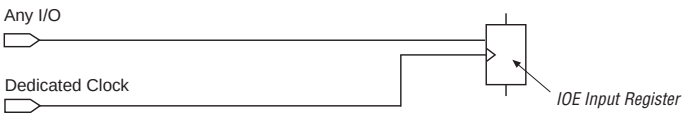
$$t_{CLR} = t_{DIN2IOE} + t_{IOC} + t_{IOCLR} + t_{OD1}$$

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



$$t_{SU} = t_{INREG} - (t_{DCLK2IOE} + t_{IOC}) + t_{IOSU}$$

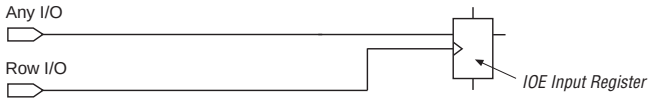
Hold Time from a Global Clock & Any I/O Data Input



$$t_{H} = (t_{DCLK2IOE} + t_{IOC}) - t_{INREG} + t_{IOH}$$

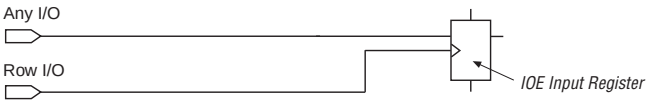
Figure 10. I/O Element Timing Delays (Part 2 of 2)

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



$$t_{ASU} = t_{INREG} - (t_{INCOMB} + t_{LEPERIPH} + t_{IOC}) + t_{IOSU}$$

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



$$t_{AH} = (t_{INCOMB} + t_{LEPERIPH} + t_{IOC}) - t_{INREG} + t_{IOH}$$

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

$$t_{CO} = t_{DCLK2IOE} + t_{IOC} + t_{IOCO} + t_{OD1}$$

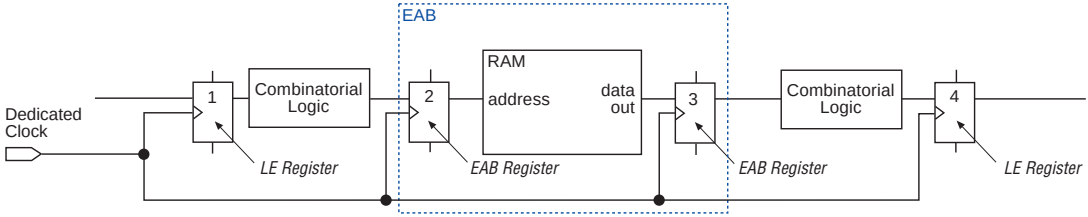
Clock-to-Output Delay from a Row I/O Clock to Any Output

$$t_{ACO} = t_{INCOMB} + t_{LEPERIPH} + t_{IOC} + t_{IOCO} + t_{OD1}$$

Figure 11 shows FLEX 10K device family EAB timing delays for a sample circuit. To calculate the delay for a different circuit, refer to the FLEX 10K timing model to determine which internal timing parameters to add together.

Figure 11. EAB Timing Delays

Cycle Time with a Global Clock



$t_{CYC1TO2} = t_C + t_{CO} + t_{SAMEROW} + t_{LUT} + t_{COMB} + t_{SAMEROW} + t_{EABDATASU}$

$t_{CYC2TO3} = t_{EABRCREG}$

$t_{CYC3TO4} = t_{EABDATACO} + t_{SAMEROW} + t_{LUT} + t_{COMB} + t_{SAMEROW} + t_{LUT} + t_{SU} - t_C$

Timing Model vs. MAX+PLUS II Timing Analyzer

Hand calculations based on the timing model provide a good estimate of a design's performance. However, the MAX+PLUS II Timing Analyzer always provides the most accurate information on design performance because it takes into account three secondary factors that influence the routing microparameters:

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

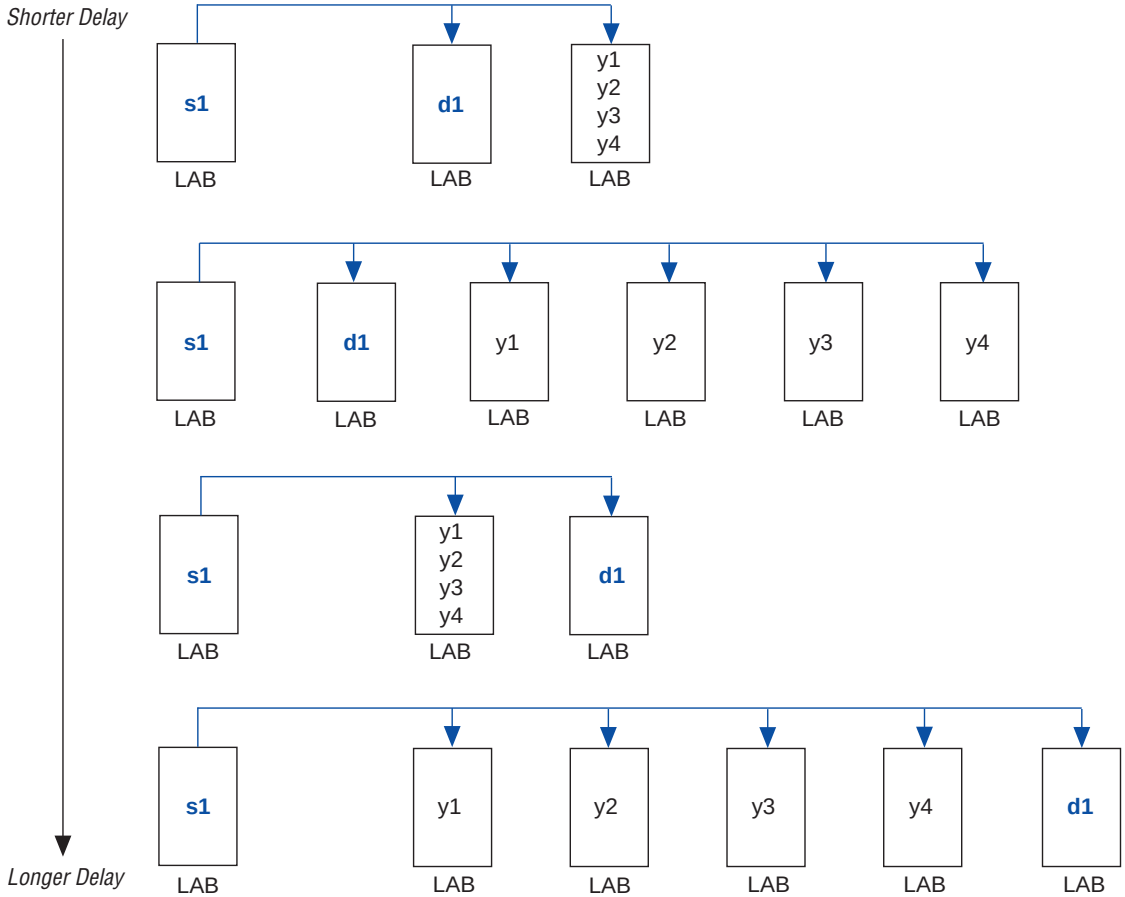
Fan-Out

The more loads a signal has to drive, the longer the $t_{SAMEROW}$, $t_{SAMECOLUMN}$, $t_{DIFFROW}$, $t_{TWOROWS}$, $t_{DIN2DATA}$, and $t_{LEPERIPH}$ delays become. These delays are functions of the number of LABs that a signal source has to drive, as well as 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.

Load Distribution

The load distribution relative to the source and destination also affects the $t_{SAMEROW}$, $t_{SAMECOLUMN}$, $t_{DIFFROW}$, $t_{TWOROWS}$, $t_{LEPERIPH}$, and $t_{DIN2DATA}$ delays. Consider a signal `s1` that feeds destination `d1` and logic elements `y[4..1]`. If `y[4..1]` are in different LABs, `s1` has four additional loads. However, if the LEs are all in the same LAB, `s1` has four shorter-delay loads. Therefore, the row interconnect delay from `s1` to `d1` is greater when each load `y[4..1]` is in a different LAB. Figure 12 illustrates how variations in the position of `d1` and the distribution of `y[4..1]` change the routing delay.

Figure 12. Delay from s1 to d1 as a Function of Relative Position & Load Distribution



Distance

The distance between the source and destination LEs also affects the $t_{SAMEROW}$, $t_{SAMECOLUMN}$, $t_{DIFFROW}$, $t_{TWOROWS}$, $t_{DIN2DATA}$, and $t_{LEPERIPH}$ parameters. For example, if `s1` drives an LE in the same row, the delay from `s1` to the LE increases as the distance from `s1` to the LE increases.

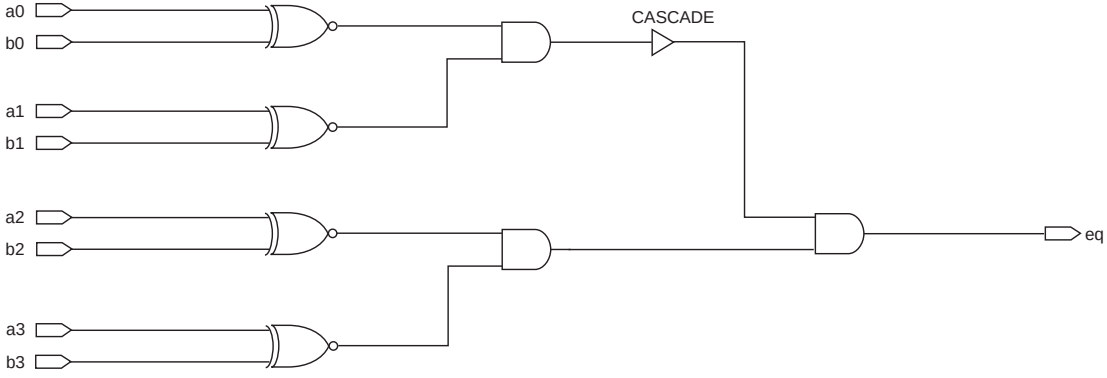
Examples

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

Example 1: 4-Bit Equality Comparator with a Cascade Chain

You can analyze the timing delays for circuits that have been subjected to minimization and logic synthesis. The synthesized equations are available in your project's MAX+PLUS II Report File (**.rpt**). These equations are structured so that you can quickly determine the logic implementation of any signal. For example, Figure 13 shows a 4-bit equality comparator.

Figure 13. 4-Bit Equality Comparator Circuit

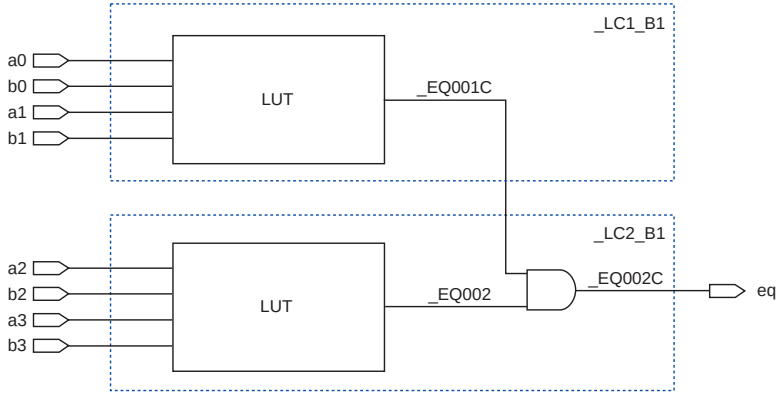


The MAX+PLUS II Report File for the circuit shown in Figure 13 gives the following equations for eq, the output of the comparator:

```
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;
_LC1_B1 =  LCELL( _EQ001C);
_EQ001C =  _EQ001;
_EQ001  =  a0 & a1 & b0 & b1
           # a0 & !a1 & b0 & !b1
           # !a0 & a1 & !b0 & b1
           # !a0 & !a1 & !b0 & !b1;
```

Figure 14 shows a synthesized 4-bit equality comparator.

Figure 14. Synthesized 4-Bit Equality Comparator



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

If a2 and eq are both row I/O pins, the timing delay from a2 to eq can be estimated by adding the following microparameters:

$$t_{INCOMB} + t_{SAMEROW} + t_{LUT} + t_{COMB} + t_{SAMEROW} + t_{IOD} + t_{IOCOMB} + t_{OD1}$$

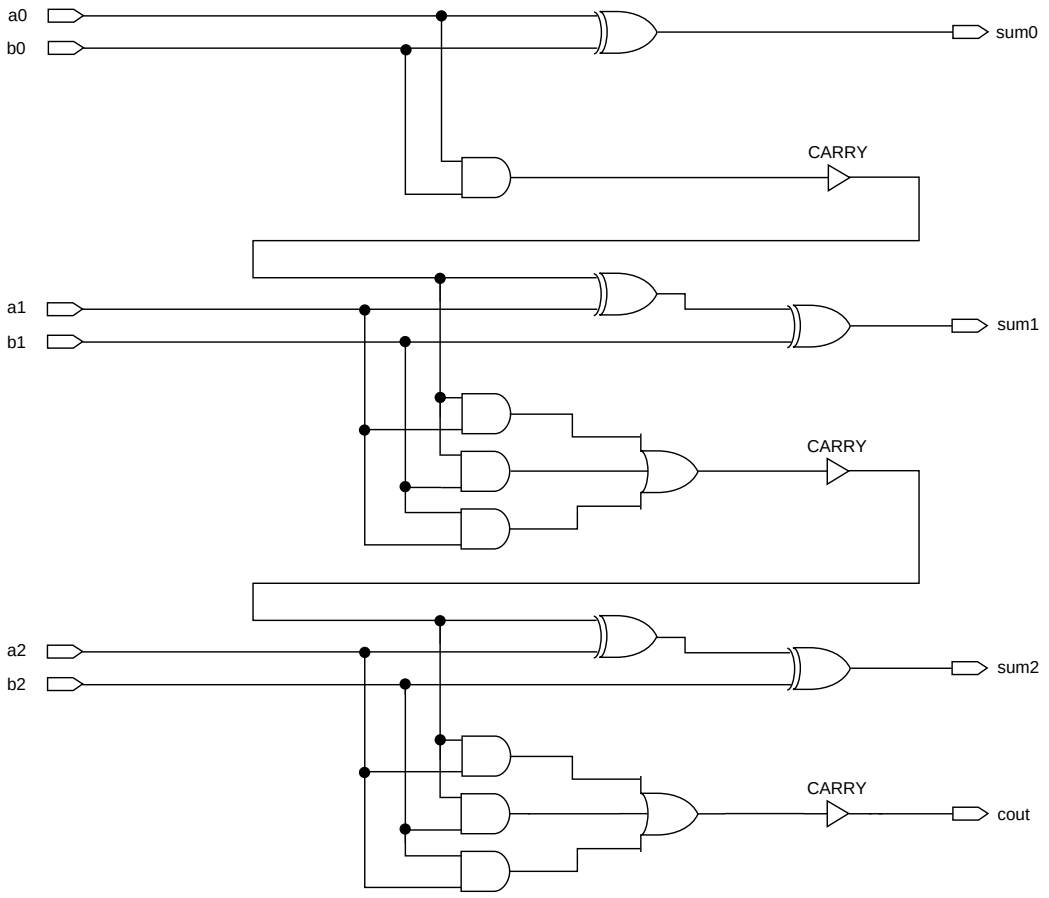
If a0 is a row I/O pin, the timing delay from a0 to eq can be estimated by adding the following microparameters:

$t_{INCOMB} + t_{SAMEROW} + t_{LUT} + t_{CASC} + t_{COMB} + t_{SAMEROW} + t_{IOD} + t_{IOCOMB} + t_{OD1}$

Example 2: 3-Bit Adder Using a Carry Chain

FLEX 10K 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 10K architecture has a built-in carry chain that performs this function. This example explains how to estimate the delay for a 3-bit adder that uses a carry chain (see Figure 15).

Figure 15. 3-Bit Adder Implemented with a Carry Chain

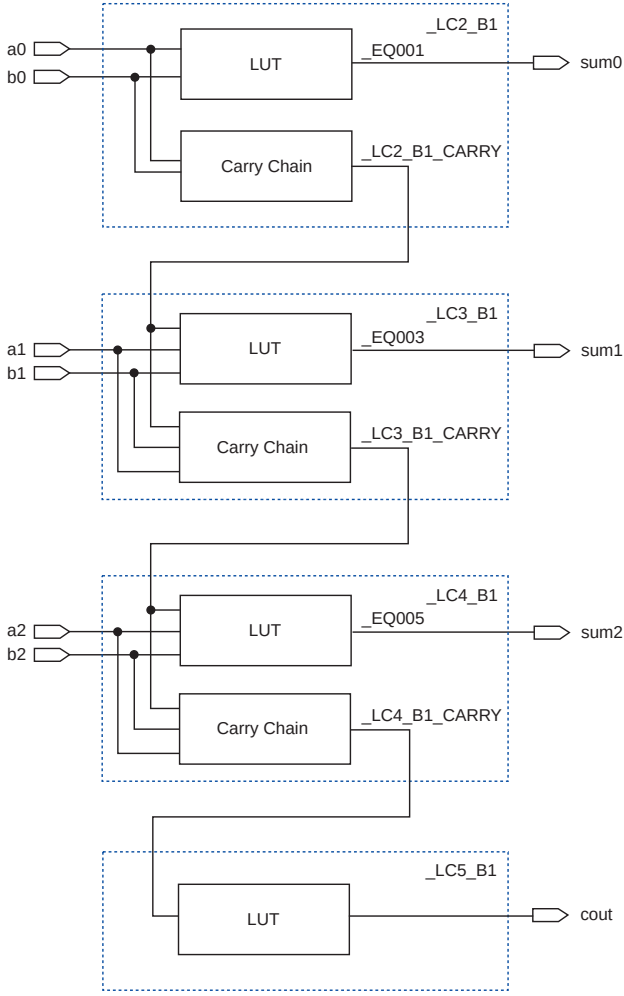


The MAX+PLUS II Report File contains the following equations for the 3-bit adder in Figure 15:

```
cout            = _LC5_B1;
sum0            = _LC2_B1;
sum1            = _LC3_B1;
sum2            = _LC4_B1;
_LC2_B1         = LCELL( _EQ001);
_EQ001          = !a0 & b0
                  # a0 & !b0;
_LC2_B1_CARRY   = CARRY( _EQ002);
_EQ002          = a0 & b0;
_LC3_B1         = LCELL( _EQ003);
_EQ003          = a1 & !b1 & !_LC2_B1_CARRY
                  # !a1 & !b1 & _LC2_B1_CARRY
                  # a1 & b1 & _LC2_B1_CARRY
                  # !a1 & b1 & !_LC2_B1_CARRY;
_LC3_B1_CARRY   = CARRY( _EQ004);
_EQ004          = a1 & _LC2_B1_CARRY
                  # a1 & b1
                  # b1 & _LC2_B1_CARRY;
_LC4_B1         = LCELL( _EQ005);
_EQ005          = a2 & !b2 & !_LC3_B1_CARRY
                  # !a2 & !b2 & _LC3_B1_CARRY
                  # a2 & b2 & _LC3_B1_CARRY
                  # !a2 & b2 & !_LC3_B1_CARRY;
_LC5_B1         = LCELL( _LC4_B1_CARRY);
_LC4_B1_CARRY   = CARRY( _EQ006);
_EQ006          = a2 & _LC3_B1_CARRY
                  # a2 & b2
                  # b2 & _LC3_B1_CARRY;
```

Figure 16 shows a synthesized 3-bit adder.

Figure 16. Synthesized 3-Bit Adder



In Figure 16, LE _LC2_B1 generates sum0 and a carry-out signal (_LC2_B1_CARRY) that feeds the carry-in of _LC3_B1. LE _LC3_B1 generates sum1 and a carry-out signal (_LC3_B1_CARRY) that feeds the carry-in of _LC4_B1. LE _LC4_B1 generates sum2 and cout using a2, b2, and _LC3_B1_CARRY. The cout signal must pass through _LC5_B1 because a carry buffer cannot directly feed a pin.

If a0 and sum2 are row I/O pins, the timing delay from a0 to sum2 can be estimated by adding the following microparameters:

$t_{INCOMB} + t_{SAMEROW} + t_{CGEN} + t_{CICO} + t_{CLUT} + t_{COMB} + t_{SAMEROW} + t_{IOD} + t_{IOCOMB} + t_{OD1}$

If a0 and cout are row I/O pins, the timing delay from a0 to cout can be estimated by adding the following microparameters:

$t_{INCOMB} + t_{SAMEROW} + t_{CGEN} + t_{CICO} + t_{CICO} + t_{CLUT} + t_{COMB} + t_{SAMEROW} + t_{IOD} + t_{IOCOMB} + t_{OD1}$

Conclusion

The FLEX 10K device architecture has predictable internal timing delays that can be estimated based on signal synthesis and placement. The MAX+PLUS II Timing Analyzer provides the most accurate timing information. However, you can also use the FLEX 10K timing model along with the timing parameters listed in the *FLEX 10K Embedded Programmable Logic Family Data Sheet* in this data book to estimate a design's performance before compilation. Both methods enable you to accurately predict your design's in-system timing performance.