Early/Late Gate Synchronizer Megafunction

Solution Brief 17 **June 1997, ver. 1**

Target Applications:
Communications
Digital Signal Processing

Family: FLEX 10K & FLEX 8000

Vendor:

Nova Engineering, Inc.
5 Circle Freeway Drive
Cincinnati, OH 45246
Tel. (513) 860-3456
Fax (513) 860-3535
E-mail sales@nova-eng.com
WWW http://www.nova-eng.com

Features

- Complete closed-loop synchronizer
- Variable loop filter bandwidth
- Balanced gate/dual integrator design
- Optimized for the Altera® FLEX® 10K and FLEX 8000 device architectures
- Applications
  - Digital receivers
  - Phase-locked loops (PLLs)

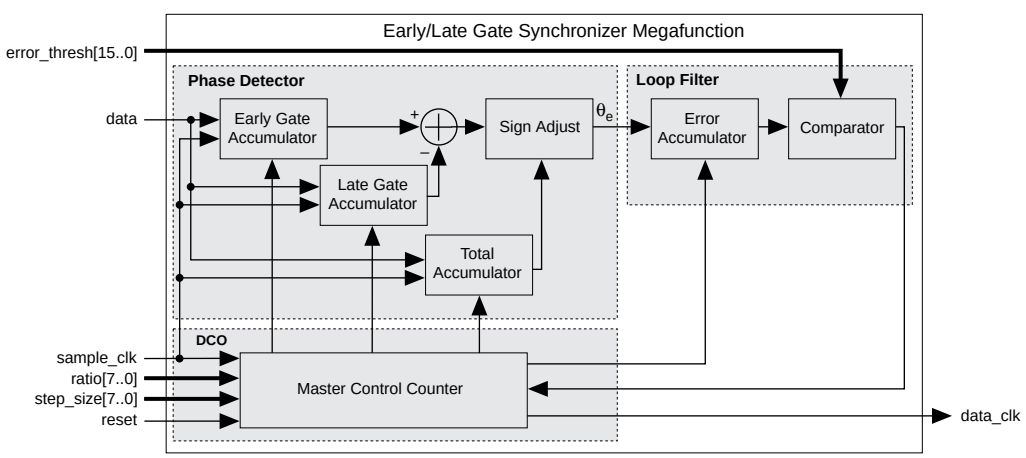
General Description

The early/late gate synchronizer megafunction contains all the functions necessary for a complete, first-order, closed-loop synchronizer. The synchronizer includes a phase detector, an up-down counter loop filter, and a digitally controlled oscillator (DCO). The phase detector is a balanced early/late gate, dual integrator design. The output of the phase detector is the difference, or phase error, between the data clock and the input data stream. The phase error (θ_e) output from the phase detector is accumulated in an up-down counter, which increments and decrements according to the sign and magnitude of the phase error.

The DCO advances or retards the phase of the locally generated data clock whenever the error accumulator exceeds a specified error threshold. The error threshold is programmable and is used to control the bandwidth of the loop filter. The loop bandwidth can be narrowed by increasing the error threshold and widened by decreasing the error threshold. Small error thresholds allow the filter to respond to rapid changes in the phase error.

The DCO also adjusts the phase of the locally generated data clock in programmable step sizes. The step size, or magnitude, of the phase adjustment determines the loop acquisition time and data clock jitter. Large step sizes can be used to minimize acquisition times, since large phase steps can quickly correct large phase errors. Small step sizes can be used to minimize clock jitter when the loop is locked. See Figure 1.

Figure 1. Early/Late Gate Synchronizer Megafunction Block Diagram



Functional Description

The phase detector requires a high-speed clock to sample the input binary data stream. The sampling clock must be an even multiple of the data rate and is typically 16 times the data rate (i.e., `ratio[7..0]` = 16). The relationship between the data and sampling clocks is described below:

$$T_{data\_clk} = \texttt{ratio[7..0]} \times T_{sample\_clk}$$

where: T_{sample_clk} = Period of the sample clock
T_{data_clk} = Period of the data clock
`ratio[7..0]` = Even integer between 4 and 254

A master counter generates the data clock and the early and late gate timing pulses. The early timing gate enables one accumulator to integrate the energy in the incoming signal during the first half of the symbol period. The late timing gate enables a second accumulator to integrate the signal energy during the second half of the symbol period. The total accumulator integrates the signal energy during the entire symbol period. The total accumulator determines whether the data bit was a binary one or zero, and adjusts the sign of the phase error appropriately. Assuming transitions in the data, the difference between the early and late gate integrators is proportional to the receiver phase error (θ_e). The phase detector output is defined as follows:

$$\theta_e = |\Sigma_{early}| - |\Sigma_{late}|$$

where: $|\Sigma_{early}| \leq \frac{\texttt{ratio[7..0]}}{2}$ and $|\Sigma_{late}| \leq \frac{\texttt{ratio[7..0]}}{2}$

The phase error is accumulated until:

$$|\Sigma\theta_e| > \texttt{error\_thresh[15..0]}$$

The DCO advances or retards the phase of the data clock whenever the accumulated error exceeds the programmed error threshold. The direction of the phase adjustment is determined by the sign of the accumulated phase error, while the magnitude of the correction is determined by the value programmed into `step_size[7..0]`.

Phase corrections, or phase steps, by the DCO can be represented in degrees or fractional cycles of the data clock. The following equation demonstrates how to calculate the phase step in degrees:

$$\text{phase step} = \frac{\texttt{step\_size[7..0]}}{\texttt{ratio[7..0]}} \times 360^\circ$$

The `step_size[7..0]` value should always be very small in relation to the `ratio[7..0]` value. Phase corrections greater than the `ratio[7..0]` value will make the loop unstable. The `step_size[7..0]` value should also be greater than zero, except when no phase corrections are desired. Setting the `step_size[7..0]` value to zero is equivalent to disabling the synchronizer.

The timing jitter, caused by phase adjustments to the data clock, can be calculated with the following equation:

$$\text{timing jitter} = \texttt{step\_size[7..0]} \times T_{sample\_clk}$$

Ports

The master control logic uses the `ratio[7..0]` input to determine the number of sample clocks per symbol, the number of sample clocks per early gate, and the number of sample clocks per late gate. The integration time for the early and late gate integrators is determined by the value of `ratio[7..0]`/2. The phase detector requires equal integration times for the early and late gate integrators. Therefore, the `ratio[7..0]` value must be an even integer between 4 and 254. Table 1 describes the ports for the early/late gate synchronizer megafunction.

Table 1. Early/Late Gate Synchronizer Megafunction Ports

Name	Type	Width (Bits)	Description
`data`	Input	1	Binary input data with a bit period equal to one cycle of the data clock.
`sample_clk`	Input	–	Sample clock.
`reset`	Input	–	Asynchronous reset, active high.
`ratio[7..0]`	Input	8	Defines the number of sample clocks per data clock.
`error_thresh[15..0]`	Input	16	Defines the error threshold at which a phase correction occurs.
`step_size[7..0]`	Input	8	Defines the magnitude of the phase corrections.
`data_clk`	Output	–	Output clock, phase aligned to input data.

Changes in the values of `ratio[7..0]`, `error_thresh[15..0]`, and `step_size[7..0]` must be synchronous to `sample_clk`. Specifically, these values must be stable and correct when the rising edge of `sample_clk` occurs.

Device Utilization

The early/late gate synchronizer megafunction is designed for both FLEX 10K and FLEX 8000 devices and does not require the use of the FLEX 10K embedded array blocks (EABs). Therefore, this megafunction performs equally well in both device families. Table 2 illustrates the device utilization and maximum clock frequency for the synchronizer.

Table 2. Typical Device Utilization

Implementation	Clock (f_{MAX})	Logic Cells	EABs
`error_thresh[15..0]` = arbitrary `ratio[7..0]` = arbitrary `step_size[7..0]` = arbitrary	35 MHz	260	0

Applications

The early/late gate synchronizer megafunction is fundamentally a digital phase-locked loop (DPLL). The synchronizer is designed to provide phase lock between an internally generated data clock and an input data stream. Moreover, it can perform the traditional task of providing phase lock between two clocks.

The synchronizer can be used as a PLL by connecting the reference clock to the data input. The reference clock appears as an alternating one and zero data pattern (i.e., 101010) to the synchronizer. Again, a high-speed sampling clock is needed to generate the internal timing and control. Because the data clock is twice the frequency of the reference clock, it is necessary to divide the output data clock by two.

101 Innovation Drive
San Jose, CA 95134
(408) 544-7000
http://www.altera.com

Copyright © 1997 Altera Corporation. Altera, AMPP, FLEX, FLEX 10K, and FLEX 8000 are trademarks and/or service marks of Altera Corporation in the United States and other countries. Other brands or products are trademarks of their respective holders. The specifications contained herein are subject to change without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera Corporation. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services. All rights reserved.