Amendments to the Claims:

This listing of claims will replace all prior versions, and listings of claims in the application:

Listing of Claims:

Claims 1-76. (Cancelled)

Claim 77. (Currently amended) A method for designing logic circuitry implemented by a programmable logic device, the method comprising:

providing a design file incorporating the logic design circuitry in computer readable format; and with a computer aided design system:

generating from the design file a retiming graph for each of a plurality of clock domains associated with the logic design circuitry;

applying constraints to the retiming graph to determine which registers of the retiming graph are capable of being moved forward or backward across a combinational logic element of the logic design circuitry; and

retiming the logic design circuitry by moving compatible registers on the retiming graph backward or forward across the combinational logic elements until the difference between gate-to-gate delays for the logic design is minimized.

Claim 78. (Previously presented) The method of claim 77, wherein applying the constraints comprises determining which of the registers are compatible.

Claim 79. (Previously presented) The method of claim 78, wherein the registers are compatible if they have the same clock domain, the same clock-enable, and they have the same asynchronous load.

Claim 80. (Previously presented) The method of claim 77, wherein retiming comprises iteratively retiming the logic design circuitry to find a minimal difference between the gate-to-gate delays.

Claim 81. (Previously presented) The method of claim 77, wherein retiming comprises determining delays associated with a netlist and determining a required cycle time.

Claim 82. (Previously presented) The method of claim 77, wherein retiming the logic design circuitry comprises forming a host node to store the constraints.

Claim 83. (Previously presented) The method of claim 77, further comprising applying constraints for asynchronous signals by setting upper and lower boundaries on edges in the retiming graph.

Claim 84. (Previously presented) The method of claim 77, further comprising applying the retiming graph to a netlist based on user selectable criteria.

Claim 85. (Previously presented) The method of claim 84, wherein applying the retiming comprises modifying power-up signals, or reset signals, or asynchronous logic signals on the registers retimed.

Claim 86. (Previously presented) The method of claim 77, further comprising removing registers of the retiming graph that are not capable of being moved forward or backward across logic elements of the logic design circuitry.

Claim 87. (Currently amended) A method for designing logic circuitry implemented by a programmable logic device, the method comprising:

providing a design file incorporating the logic design circuitry in computer readable format; and with an electronic design automation system:

generating from the design file a retiming graph for each of a plurality of clock domains associated with the logic design circuitry;

applying constraints to the retiming graph to determine which registers of the retiming graph are compatible and are capable of being moved forward or backward across a combinational logic element of the logic design circuitry; and

retiming the logic design circuitry by moving compatible registers on the retiming graph backward or forward across the combinational logic elements until the difference between gate-to-gate delays for the logic design is minimized,

wherein the registers are compatible if they have the same clock-enable.

Claim 88. (Previously presented) The method of claim 87, wherein retiming comprises iteratively retiming the logic design circuitry to find a minimal difference between the gate-to-gate delays.

Claim 89. (Previously presented) The method of claim 87, further comprising applying constraints for asynchronous signals by setting upper and lower boundaries on edges in the retiming graph.

Claim 90. (Previously presented) The method of claim 87, further comprising applying the retiming graph to a netlist based on user selectable criteria.

Claim 91. (Previously presented) The method of claim 90, wherein applying the retiming comprises modifying power-up signals, or reset signals, or asynchronous logic signals on the registers retimed.

Claim 92. (Previously presented) The method of claim 87, further comprising removing registers of the retiming graph that are not capable of being moved forward or backward across logic elements of the logic design circuitry.

Claim 93. (Currently amended) A method for designing logic circuitry implemented by a programmable logic device, the method comprising:

providing a design file incorporating the logic design circuitry in computer readable format; and with a computer:

generating from the design file a retiming graph for each of a plurality of clock domains associated with the logic design circuitry;

applying constraints to the retiming graph to determine which registers of the retiming graph are compatible and are capable of being moved forward or backward across a combinational logic element of the logic design circuitry; and

retiming the logic design circuitry by moving compatible registers on the retiming graph backward or forward across the combinational logic elements until the difference between gate-to-gate delays for the logic design is minimized,

wherein the registers are compatible if they have they have the same asynchronous load.

Claim 94. (Previously presented) The method of claim 93, wherein retiming comprises iteratively retiming the logic design circuitry to find a minimal difference between the gate-to-gate delays.

Claim 95. (Previously presented) The method of claim 93, wherein retiming comprises determining delays associated with a netlist and determining a required cycle time.

Claim 96. (Previously presented) The method of claim 93, further comprising removing registers of the retiming graph that are not capable of being moved forward or backward across logic elements of the logic design circuitry.