WHAT IS CLAIMED IS:

1. A method executing in a computer-aided logic design system for designing logic circuitry, wherein the method performs register retiming by moving a register in a logic design across one or more delay elements of the design, the method comprising:

providing a logic design file incorporating the logic design in computer readable format;

finding in the logic design a design condition comprising a first register being clocked using an unrelated clock domain to a second register; and

disallowing moving of the first and second register across delay elements when performing register retiming.

2. A method executing in a computer-aided logic design system for designing logic circuitry, wherein the method performs register retiming by moving a register in a logic design across one or more delay elements of the design, the method comprising:

providing a logic design file incorporating the logic design in computer readable format;

finding in the logic design a design condition comprising a first register having an input directly fed by a pin; and

disallowing moving of the first register across delay elements when performing register retiming.

3. The method of claim 2 wherein the design condition is replaced with a first register having an output directly feeding a pin.

4. The method of claim 2 wherein the design condition is replaced with a first register feeding an asynchronous signal to a second register.

5. The method of claim 1 wherein the design condition is replaced with a first register being clocked using a first clock domain and a second register being clocked using a second clock domain, different clock domain from the first clock domain.

6. The method of claim 2 wherein the design condition is replaced with a first register where a user has placed a timing constraint.

7. The method of claim 6 further comprising:

permitting a user to place a timing constraint on one or more registers of the logic design.

8. The method of claim 2 wherein the design condition is replaced with a first register where a user has indicated as "don't touch."

9. The method of claim 8 further comprising:

permitting a user to assign a "don't touch" indication on one or more registers of the logic design.

10. The method of claim 1 wherein a delay element comprises at least one of a logic gate, AND, OR, XOR, NOR, NAND, INVERT, look-up table, RAM, or DSP block.

11. The method of claim 2 wherein a delay element comprises at least one of a logic gate, AND, OR, XOR, NOR, NAND, INVERT, look-up table, RAM, or DSP block.

12. The method of claim 3 wherein a delay element comprises at least one of a logic gate, AND, OR, XOR, NOR, NAND, INVERT, look-up table, RAM, or DSP block.

13. The method of claim 4 wherein a delay element comprises at least one of a logic gate, AND, OR, XOR, NOR, NAND, INVERT, look-up table, RAM, or DSP block.

14. The method of claim 5 wherein a delay element comprises at least one of a logic gate, AND, OR, XOR, NOR, NAND, INVERT, look-up table, RAM, or DSP block.

15. The method of claim 6 wherein a delay element comprises at least one of a logic gate, AND, OR, XOR, NOR, NAND, INVERT, look-up table, RAM, or DSP block.

16. The method of claim 8 wherein a delay element comprises at least one of a logic gate, AND, OR, XOR, NOR, NAND, INVERT, look-up table, RAM, or DSP block.

17. The method of claim 1 wherein the logic design file is for a programmable logic integrated circuit.

18. The method of claim 6 wherein the logic design file is for a programmable logic integrated circuit.

19. The method of claim 8 wherein the logic design file is for a programmable logic integrated circuit.

20. A method executing in a computer-aided logic design system for designing logic circuitry, wherein the method performs register retiming by moving a register in a logic design across one or more delay elements of the design, the method comprising:

providing a logic design file incorporating the logic design in computer readable format;

providing a design condition where two registers are compatible if they have the same clock enable;

determining whether a first register and a second register are compatible; and

permitting forward retiming of the first and second register when the first and second registers are compatible.

21. The method of claim 20 wherein the design condition is replaced with where two registers are compatible if they have their asynchronous signal triggered at the same time.

22. The method of claim 20 wherein the design condition is replaced with where two registers are compatible if they have the same list of signals that trigger asynchronous events.

23. The method of claim 20 wherein the design condition is replaced with where two registers are compatible if they have the same asynchronous signal load.

24. The method of claim 20 wherein the design condition is replaced with where two registers are compatible if they have either the same asynchronous data or one has no asynchronous data.

25. The method of claim 20 wherein the design condition further comprises two registers are compatible if they have their asynchronous signal triggered at the same time, and two registers are compatible if they have the same asynchronous signal load.

26. A method comprising:

providing a high-level design language representation of an integrated circuit design;

executing a design synthesis process on the high-level design language representation of the integrated circuit design to obtain a netlist for the integrated circuit design;

controlling a flow of the design synthesis process using script commands provided to the design synthesis process in a script file, wherein the script commands comprise a retiming command;

causing a retiming operation to be performed on netlist of the integrated circuit design by including the retiming command in the script file; and

during the retiming operation, relocating a register from first position in a logic path to a second position in the logic path, different from the first position.

27. The method of claim 26 wherein the integrated circuit design is a programmable logic integrated circuit design.

28. The method of claim 26 wherein a first logical function provided by the logic path with the register in the first position is equivalent to a second logical function provided by the logic path with the register in the second position.

29. The method of claim 26 wherein causing the retiming operation comprises:

finding a logic path in the netlist comprising an input coupled to a first delay element coupled to a first register coupled to a second delay element coupled to a second register coupled to a third delay element coupled to an output; and

determining timing delays between the input to first register, first register to second register, and second register to output.

30. The method of claim 29 further comprising:

reducing one of the timing delays by relocating the first or second register

31. A method comprising:

providing a first design for a programmable logic integrated circuit comprising a first logic element comprising a first combinatorial block and a first register block and a second logic element comprising a second combinatorial block and a second register block, where in the first design an output of the first combinatorial block is configured to couple to an input of the second logic element and an output of the second register block is configured to couple to a third logic element; and

performing a retiming operation on the first design for the programmable logic integrated circuit to generate a second design for the programmable logic integrated circuit, where in the second design an output of the first register block is configured to couple to an input of the second logic element and an output of the second combinatorial block is configured to couple to the third logic element.

32. A method comprising:

providing a first design for a programmable logic integrated circuit comprising a first logic element comprising a first combinatorial block and a first register block and a second logic element comprising a second combinatorial block and a second register block, where in the first design an output of the first register block is configured to couple to an input of the second logic element and an output of the second combinatorial block is configured to couple to a third logic element; and

performing a retiming operation on the first design for the programmable logic integrated circuit to generate a second design for the programmable logic integrated circuit, where in the second design an output of the first combinatorial block is configured to couple to an input of the second logic element and an output of the second register block is configured to couple to the third logic element.

33. The method of claim 31 wherein the second design is functionally equivalent to the first design.

34. The method of claim 32 wherein the second design is functionally equivalent to the first design.

35. The method of claim 31 wherein the programmable logic integrated circuit further comprises a fourth logic element comprising a fourth combinatorial block and a fourth register block and in the first design an output of the fourth combinatorial block is configured to couple to an input of the second logic element, and

after the retiming operation, in the second design an output of the fourth register block is configured to couple to an input of the second logic element.

36. The method of claim 32 wherein the programmable logic integrated circuit further comprises a fourth logic element comprising a fourth combinatorial block and a fourth register block and in the first design an output of the fourth register block is configured to couple to an input of the second logic element, and

after the retiming operation, in the second design an output of the fourth combinatorial block is configured to couple to an input of the second logic element.

37. The method of claim 31 where in the second design the output of the first register block is configured to couple to the same input of the second logic element that the output of the first combinatorial block was configured to couple to in the first design.

38. The method of claim 32 where in the second design the output of the first combinatorial block is configured to couple to the same input of the second logic element that the output of the first register block was configured to couple to in the first design.

39. The method of claim 31 where in the second design the output of the second combinatorial block is configured to couple to the same input of the third logic element that the output of the second register block was configured to couple to in the first design.

40. The method of claim 32 where in the second design the output of the second register block is configured to couple to the same input of the third logic element that the output of the second combinatorial block was configured to couple to in the first design.

41. The method of claim 20 further comprising:

performing a retiming operation on the first and second registers to add a third register and remove the first and second registers from the design;

adding logic gates to asynchronous input paths of the third register so a functionality provided by the third register is equivalent to a functionality provided by the first and second registers.

42. The method of claim 21 further comprising:

performing a retiming operation on the first and second registers to add a third register and remove the first and second registers from the design;

adding logic gates to asynchronous input paths of the third register so a functionality provided by the third register is equivalent to a functionality provided by the first and second registers.

43. The method of claim 22 further comprising:

performing a retiming operation on the first and second registers to add a third register and remove the first and second registers from the design;

adding logic gates to asynchronous input paths of the third register so a functionality provided by the third register is equivalent to a functionality provided by the first and second registers.

44. The method of claim 23 further comprising:

performing a retiming operation on the first and second registers to add a third register and remove the first and second registers from the design;

adding logic gates to asynchronous input paths of the third register so a functionality provided by the third register is equivalent to a functionality provided by the first and second registers.

45. The method of claim 24 further comprising:

performing a retiming operation on the first and second registers to add a third register and remove the first and second registers from the design;

adding logic gates to asynchronous input paths of the third register so a functionality provided by the third register is equivalent to a functionality provided by the first and second registers.

46. The method of claim 24 further comprising:

performing a retiming operation on the first and second registers to add a third register and remove the first and second registers from the design;

adding logic gates to asynchronous input paths of the third register so a functionality provided by the third register is equivalent to a functionality provided by the first and second registers.

47. A method comprising:

performing a retiming operation on a first register of a first design;

relocating the first register through a delay element to obtain a second and third register for a second design, equivalent to the first design; and

adding a logic structure coupled to the second register and third register of the second design depending on the delay element.

48. The method of claim 47 wherein when the delay element comprises a AND or an OR gate, no logic structure is added to input paths of the second and third registers.

49. The method of claim 47 wherein when the delay element comprises a NOT gate, the logic structure comprises a NOT gate added to input paths of the second and third registers.

50. The method of claim 47 wherein when the delay element comprises an XOR gate or look-up table, the logic structure comprises a multiplexer coupled having first input coupled to an output of the second register and a select input coupled to an output of the third register.

51. The method of claim 50 wherein the multiplexer further comprises a second input coupled to an output of an XOR gate when the delay element comprises an XOR gate.

52. The method of claim 50 wherein the multiplexer further comprises a second input coupled to an output of a look-up table when the delay element comprises a look-up table.