CLAIMS

1. A method for improving the timing performance of electronic circuits designed using electronic design automation, the method comprising:

carrying out an initial wire-length-driven placement for an integrated circuit design embodied in an unplaced netlist, using a computerized placer, to obtain a data structure representing initial placements of logic gates;

identifying at least one timing-critical source-sink path between at least one pair of source-sink endpoints in said data structure representing said initial placements;

creating a new pseudo two-pin net, the new pseudo two-pin net lacking a corresponding electrical signal in the electronic circuit, for each pair of said at least one pair of source-sink endpoints to create an updated netlist; and

carrying out a revised wire-length-driven placement on said updated netlist to obtain a data structure representing revised placements.

2. The method of Claim 1, wherein said at least one timing-critical source sink path comprises a plurality of timing-critical source-sink paths and said at least one pair of source-sink endpoints comprises a plurality of pairs of source-sink endpoints.

3. The method of Claim 2, further comprising identifying at least one timing-critical source-sink path between at least one pair of source-sink endpoints in said data structure based on the revised placement, creating at least one new pseudo two-pin net for each pair of said plurality of pairs of endpoints identified based on the revised placement, and repeating said step of carrying out said revised wire-length-driven placement using said at least one new pseudo two-pin net.

4. The method of Claim 3, wherein carrying out said initial wire-length-driven placement and carrying out said revised wire-length-driven placement each comprise applying a half-perimeter wire length-driven placement, and wherein said computerized placer comprises a half-perimeter wire length-driven computerized placer.

5. The method of Claim 3, wherein identifying said plurality of timing-critical source-sink paths between said pairs of source-sink endpoints in said data structure representing said initial placements comprises obtaining input from a human subject matter expert, using a computerized user interface.

6. The method of Claim 3, wherein identifying said plurality of timing-critical source-sink paths between said pairs of source-sink endpoints in said data structure representing said initial placements comprises using a computerized timing estimation routine.

7. The method of Claim 6, wherein using said computerized timing estimation routine comprises using said computerized timing estimation routine to obtain results that are relatively accurate among said pairs of source-sink endpoints.

8. The method of Claim 3, wherein carrying out said revised wire-length-driven placement on said updated netlist to obtain said data structure representing revised placements comprises moving said pairs of endpoints closer to each other.

9. The method of Claim 3, wherein carrying out said revised wire-length-driven placement on said updated netlist to obtain said data structure representing revised placements comprises lowering tortuosity of paths between said pairs of endpoints.

10. The method of Claim 9, wherein carrying out said revised wire-length-driven placement on said updated netlist to obtain said data structure representing revised placements further comprises moving said pairs of endpoints closer to each other.

11. The method of Claim 9, wherein, in carrying out said revised wire-length-driven placement on said updated netlist to obtain said data structure representing revised placements, said pairs of endpoints are fixed relative to each other.

12. The method of Claim 3, further comprising retaining said pseudo two-pin nets during said plurality of total iterations.

13. The method of Claim 3, further comprising:

carrying out logic synthesis to obtain said integrated circuit design embodied in said unplaced netlist; and

subsequent to said plurality of total iterations, carrying out, on a final version of said data structure representing said revised placements, virtual timing optimization, clock optimization, wire synthesis and optimization, and routing.

14. The method of Claim 13, further comprising:

based on results of said routing, repeating said steps of identifying said plurality of timing-critical source-sink paths, creating said new pseudo two-pin net for each pair of said pairs of endpoints, and carrying out said revised wire-length-driven placement for a new plurality of total iterations.

15. The method of Claim 13, further comprising fabricating a physical integrated circuit based on said routing.

16. The method of Claim 3, further comprising:

clustering and fixing memory elements on a final version of said data structure representing said revised placements;

carrying out a further revised wire-length-driven placement based on said clustering and fixing; and

repeating said steps of identifying said plurality of timing-critical source-sink paths, creating said new pseudo two-pin net for each pair of said pairs of endpoints, and carrying out said revised wire-length-driven placement for a new plurality of total iterations subsequent to said further revised wire-length-driven placement based on said clustering and fixing.

17. A computer comprising:

a memory; and

at least one processor, coupled to said memory, and operative to improve the timing performance of electronic circuits designed using electronic design automation by:

carrying out an initial wire-length-driven placement for an integrated circuit design embodied in an unplaced netlist, using a computerized placer, to obtain a data structure representing initial placements of logic gates;

identifying at least one timing-critical source-sink path between at least one pair of source-sink endpoints in said data structure representing said initial placements;

creating a new pseudo two-pin net, the new pseudo two-pin net lacking a corresponding electrical signal in the electronic circuit, for each pair of said at least one pair of source-sink endpoints to create an updated netlist; and

carrying out a revised wire-length-driven placement on said updated netlist to obtain a data structure representing revised placements.

18. The computer of Claim 17, wherein said at least one timing-critical source sink path comprises a plurality of timing-critical source-sink paths and said at least one pair of source-sink endpoints comprises a plurality of pairs of source-sink endpoints, and where said at least one processor is further operative to repeat said steps of identifying said plurality of timing-critical source-sink paths, creating said new pseudo two-pin net for each pair of said plurality of pairs of endpoints, and carrying out said revised wire-length-driven placement for a plurality of total iterations.

19. The computer of Claim 18, wherein said at least one processor is further operative to:

subsequent to said plurality of total iterations, carry out, on a final version of said data structure representing said revised placements, virtual timing optimization, clock optimization, wire synthesis and optimization, and routing; and

prepare a layout based on said routing;

instantiate said layout as a design structure; and

provide said design structure to fabrication equipment to facilitate fabrication of a physical integrated circuit in accordance with said design structure.

20. A computer program product comprising one or more computer readable storage media having stored thereon:

first program instructions executable by a computer system to cause the computer system to carry out an initial wire-length-driven placement for an integrated circuit design embodied in an unplaced netlist, using a computerized placer, to obtain a data structure representing initial placements of logic gates;

second program instructions executable by the computer system to cause the computer system to identify at least one timing-critical source-sink path between at least one pair of source-sink endpoints in said data structure representing said initial placements;

third program instructions executable by the computer system to cause the computer system to create a new pseudo two-pin net, the new pseudo two-pin net lacking a corresponding electrical signal in the electronic circuit, for each pair of said at least one pair of source-sink endpoints to create an updated netlist; and

fourth program instructions executable by the computer system to cause the computer system to carry out a revised wire-length-driven placement on said updated netlist to obtain a data structure representing revised placements.

21. The computer program product of Claim 20, wherein said at least one timing-critical source sink path comprises a plurality of timing-critical source-sink paths and said at least one pair of source-sink endpoints comprises a plurality of pairs of source-sink endpoints, further comprising fifth program instructions executable by the computer system to cause the computer system to repeat said steps of identifying said plurality of timing-critical source-sink paths, creating said new pseudo two-pin net for each pair of said plurality of pairs of endpoints, and carrying out said revised wire-length-driven placement for a plurality of total iterations.

22. The computer program product of Claim 21, further comprising:

sixth program instructions executable by the computer system to cause the computer system to, subsequent to said plurality of total iterations, carry out, on a final version of said data structure representing said revised placements, virtual timing optimization, clock optimization, wire synthesis and optimization, and routing; and

seventh program instructions executable by the computer system to cause the computer system to prepare a layout based on said routing;

eighth program instructions executable by the computer system to cause the computer system to instantiate said layout as a design structure; and

ninth program instructions executable by the computer system to cause the computer system to provide said design structure to fabrication equipment to facilitate fabrication of a physical integrated circuit in accordance with said design structure.

23. A method for improving the timing performance of electronic circuits designed using electronic design automation, the method comprising:

obtaining, from a computerized placer, results of an initial wire-length-driven placement for an integrated circuit design embodied in an unplaced netlist, said results comprising a data structure representing initial placements of logic gates;

obtaining, from a computerized timer, at least one timing-critical source-sink path between at least one pair of source-sink endpoints in said data structure representing said initial placements;

creating a new pseudo two-pin net, the new pseudo two-pin net lacking a corresponding electrical signal in the electronic circuit, for each pair of said at least one pair of source-sink endpoints to create an updated netlist; and

providing said updated netlist to said computerized placer to facilitate said computerized placer carrying out a revised wire-length-driven placement on said updated netlist to produce a data structure representing revised placements.

24. The method of Claim 23, wherein said at least one timing-critical source sink path comprises a plurality of timing-critical source-sink paths and said at least one pair of source-sink endpoints comprises a plurality of pairs of source-sink endpoints.

25. The method of Claim 24, further comprising repeating said steps of obtaining said plurality of timing-critical source-sink paths, creating said new pseudo two-pin net for each pair of said plurality of pairs of endpoints, and providing said updated netlist for a plurality of total iterations.