

(51) International Patent Classification:
G06F 17/50 (2006.01) **G06F 9/44** (2006.01)(21) International Application Number:
PCT/US2010/026493(22) International Filing Date:
8 March 2010 (08.03.2010)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
61/160,220 13 March 2009 (13.03.2009) US
12/423,962 15 April 2009 (15.04.2009) US(71) Applicant (for all designated States except US): **SYNOPSIS, INC.** [US/US]; 700 E. Middlefield Road, Mountain View, CA 94043 (US).

(72) Inventors; and

(75) Inventors/Applicants (for US only): **POTEMSKI, Andrew Stanley** [US/US]; 903 Brookside Court, Round Rock, TX 78681 (US). **TYSON, John Scott** [US/US]; 7616 Rockpoint Drive, Austin, TX 78731 (US).(74) Agent: **YAO, Shun**; Park, Vaughan & Fleming LLP, 2820 Fifth Street, Davis, CA 95618-7759 (US).

(81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PE, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— without international search report and to be republished upon receipt of that report (Rule 48.2(g))

(54) Title: METHOD AND APPARATUS FOR ACCELERATING PROJECT START AND TAPE-OUT

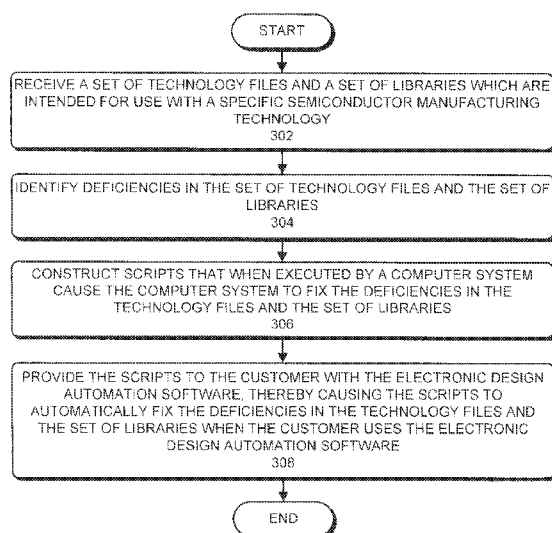


FIG. 3

(57) Abstract: Some embodiments provide systems and techniques that accelerate project start and tape-out. During operation, a system can receive a set of technology files and a set of libraries. Next, the system can identify deficiencies in the set of technology files and the set of libraries. The system can then construct update utilities that when executed by a computer system cause the computer system to fix the deficiencies in the technology files and the set of libraries. Further, a system can receive a set of checks that are performed by a foundry. Next, the system can construct tape-out scripts that when executed by a computer cause the computer to perform the set of checks on the circuit design. The update utilities and the tape-out scripts can then be provided to a customer with an electronic design automation software to accelerate project start and tape-out.

METHOD AND APPARATUS FOR ACCELERATING PROJECT START AND TAPE-OUT

5 **Inventors:** Andrew Stanley Potemski and John Scott Tyson

BACKGROUND

Field

10 [0001] This disclosure generally relates to electronic design automation. More specifically, the disclosure relates to methods and apparatuses for accelerating project start and tape-out.

Related Art

15 [0002] Rapid advances in computing technology have made it possible to perform trillions of computational operations each second on data sets that are sometimes as large as trillions of bytes. These advances can largely be attributed to the dramatic improvements in semiconductor design and manufacturing technologies that have made it possible to integrate
20 tens of millions of devices onto a single chip.

 [0003] Integration densities continue to increase at a rapid pace to keep up with the insatiable demand for smaller, faster, and more complex electronic devices and computers. As technology nodes move to 65nm and below, circuit designers and project managers are facing a number of new and difficult challenges. Specifically, circuit designers and project
25 managers are finding it increasingly difficult to meet project deadlines. Hence, what are needed are systems and techniques for reducing the circuit design time and for managing the circuit design process better.

SUMMARY

30 [0004] Some embodiments of the present invention provide systems and techniques that accelerate project start. During operation, a system can receive a set of technology files and a set of libraries, which are intended for use with a specific semiconductor manufacturing technology. Next, the system can identify deficiencies in the set of technology files and the set of libraries. The system can then construct scripts that when executed by a computer

system cause the computer system to fix the deficiencies in the technology files and the set of libraries. Next, the scripts can be provided to a customer with an electronic design automation (EDA) software. The scripts can be automatically executed by the EDA software to fix the deficiencies in the technology files and the set of libraries when the customer uses the electronic design automation software.

[0005] Some embodiments of the present invention provide systems and techniques to accelerate tape-out of a circuit design. During operation, a system can receive a set of checks that are performed by a foundry when the foundry receives a circuit design. Next, the system can construct scripts that when executed by a computer cause the computer to perform the set of checks on the circuit design. The scripts can then be provided to a customer with an EDA software. The scripts can be used to automatically perform the set of checks on the circuit design when the customer uses the EDA software to tape-out the circuit design.

BRIEF DESCRIPTION OF THE FIGURES

[0006] FIG. 1 illustrates various stages in the design and fabrication of an integrated circuit in accordance with an embodiment of the present invention.

[0007] FIG. 2A illustrates how project start can be accelerated by using update utilities in accordance with an embodiment of the present invention.

[0008] FIG. 2B illustrates how tape-out can be accelerated by performing checks during tape-out in accordance with an embodiment of the present invention.

[0009] FIG. 3 presents a flowchart that illustrates a process to accelerate project start for a circuit design project which uses electronic design automation software in accordance with an embodiment of the present invention.

[0010] FIG. 4 presents a flowchart that illustrates how scripts can be updated in response to receiving updates for a set of technology files and/or a set of libraries in accordance with an embodiment of the present invention.

[0011] FIG. 5 presents a flowchart that illustrates a process to check a circuit design during tape-out in accordance with an embodiment of the present invention.

[0012] FIG. 6 illustrates a computer system in accordance with an embodiment of the present invention.

DETAILED DESCRIPTION

[0013] The following description is presented to enable any person skilled in the art to make and use the invention, and is provided in the context of a particular application and

its requirements. Various modifications to the disclosed embodiments will be readily apparent to those skilled in the art, and the general principles defined herein may be applied to other embodiments and applications without departing from the spirit and scope of the present invention. Thus, the present invention is not limited to the embodiments shown, but
5 is to be accorded the widest scope consistent with the principles and features disclosed herein.

Integrated Circuit (IC) Design Flow

[0014] FIG. 1 illustrates various stages in the design and fabrication of an integrated
10 circuit in accordance with an embodiment of the present invention.

[0015] The process can start with a product idea (step 100) which can be realized using an integrated circuit that is designed using an EDA process (step 110). After the integrated circuit is taped-out (event 140), it can undergo a fabrication process (step 150) and a packaging and assembly process (step 160) to produce chips 170.

15 [0016] The EDA process (step 110) comprises steps 112-130, which are described below for illustrative purposes only and are not meant to limit the present invention. Specifically, the steps may be performed in a different sequence than the sequence described below.

[0017] During system design (step 112), circuit designers can describe the
20 functionality that they want to implement. They can also perform what-if planning to refine functionality, check costs, etc. Hardware-software architecture partitioning can also occur at this stage. Exemplary EDA software products from Synopsys, Inc. that can be used at this step include Model Architect, Saber®, System Studio, and DesignWare®.

25 [0018] During logic design and functional verification (step 114), the VHDL or Verilog code for modules in the system can be written and the design can be checked for functional accuracy, e.g., the design can be checked to ensure that it produces the correct outputs. Exemplary EDA software products from Synopsys, Inc. that can be used at this step include VCS®, Vera®, DesignWare®, Magellan™, Formality®, ESP and Leda®.

30 [0019] During synthesis and design for test (step 116), the VHDL/Verilog can be translated to a netlist. Further, the netlist can be optimized for the target technology, and tests can be designed and implemented to check the finished chips. Exemplary EDA software products from Synopsys, Inc. that can be used at this step include Design Compiler®, Physical Compiler®, Test Compiler, Power Compiler™, FPGA Compiler, TetraMAX®, and DesignWare®.

[0020] During netlist verification (step 118), the netlist can be checked for compliance with timing constraints and for correspondence with the VHDL/Verilog source code. Exemplary EDA software products from Synopsys, Inc. that can be used at this step include Formality®, PrimeTime®, and VCS®.

5 [0021] During design planning (step 120), an overall floorplan for the chip can be constructed and analyzed for timing and top-level routing. Exemplary EDA software products from Synopsys, Inc. that can be used at this step include Astro™ and IC Compiler products.

10 [0022] During physical implementation (step 122), circuit elements can be positioned in the layout (placement) and can be electrically coupled (routing). Exemplary EDA software products from Synopsys, Inc. that can be used at this step include Astro™ and IC Compiler products.

15 [0023] During analysis and extraction (step 124), the circuit's functionality can be verified at a transistor level and parasitics can be extracted. Exemplary EDA software products from Synopsys, Inc. that can be used at this step include AstroRail™, PrimeRail, PrimeTime®, and Star-RCXT™.

20 [0024] During physical verification (step 126), the design can be checked to ensure correctness for manufacturing, electrical issues, lithographic issues, and circuitry. Hercules™ is an exemplary EDA software product from Synopsys, Inc. that can be used at this step.

[0025] During resolution enhancement (step 128), geometric manipulations can be performed on the layout to improve manufacturability of the design. Exemplary EDA software products from Synopsys, Inc. that can be used at this step include Proteus/Progen, ProteusAF, and PSMGen.

25 [0026] During mask data preparation (step 130), the design can be "taped-out" to produce masks which are used during fabrication. Exemplary EDA software products from Synopsys, Inc. that can be used at this step include the CATS® family of products.

Configurable Foundry Ready System

30 [0027] Integration densities have increased at a rapid pace to keep up with the insatiable demand for smaller, faster, and more complex electronic devices and computers. As semiconductor technology moves to 65nm, and further to 45/40nm and below, circuit designers and project managers are facing a number of difficult challenges. First, circuit designers need highly advanced design techniques to design circuits at these nodes. Second,

due to the complex dependency between different stages in the electronic design flow, and due to the highly sophisticated nature of the processing involved in each stage, it is very difficult for project managers to accurately measure progress and to accurately predict when a circuit design project will complete. Third, the libraries and the foundry requirements are very complicated, which causes project teams to spend large amounts of time setting up the circuit design project and then taping-out the circuit design to ensure that it meets the foundry's requirements. Fourth, it is difficult to ensure that third party circuit designs (often referred to as "IP" in the semiconductor industry) are clean and are compatible with one another. In light of the above challenges, it is not surprising that 69% of tape-outs (according to some estimates) miss their deadlines.

[0028] Some embodiments of the present invention provide customers a low risk path to move to 65nm technologies and below. These embodiments reduce the turnaround time and improve predictability and productivity. Specifically, these embodiments provide an automated way to set up library data for electronic design automation software by providing scripts and templates to create a complete library and technology file database for use in a circuit design flow. Further, some embodiments help the customer ensure that a taped-out circuit design will meet the foundry's requirements.

[0029] A customer typically receives a set of technology files and a set of libraries. The technology files can include design rule check files and LVS (lithography versus silicon) check files. The technology files can also include information that describes the semiconductor manufacturing technology that is going to be used to manufacture the chip. During circuit design, the EDA software can use these parameter values to predict the behavior of the circuit design and to ensure that the final circuit design will manufacture properly. If a deficiency in the technology file goes unnoticed, it can cause substantial delays in the time to market. Deficiencies in the technology files can include invalid syntax, missing pieces that need to be generated, incorrect format, or incorrect unit. Hence, when a circuit design project is started, it is critical to ensure that the technology files do not have any deficiencies.

[0030] Circuit designers often use off-the-shelf circuit designs that implement low-level logic functions (e.g., an AND gate, an OR gate, etc.), macros, cells, etc. These off-the-shelf circuit designs are typically provided as a library, e.g., standard cell libraries, and can be provided by foundries or third party vendors. For example, foundries often provide memory modules to the customer. Note that libraries are supposed to speed up the circuit design process because they can prevent the circuit designers from "reinventing the wheel."

However, if the libraries contain deficiencies, the project team can end up wasting precious resources trying to fix the deficiencies. Hence, before a library is used in a circuit design, it is critical to ensure that the library does not have any deficiencies.

[0031] During tape-out, a circuit design must meet a set of very stringent criteria so that the manufactured chips will behave as desired. Specifically, a foundry typically performs a large number of tests on the circuit design before accepting the circuit design for manufacturing. If the foundry determines that the circuit design does not meet its requirements, the foundry notifies the circuit designers, who then modify the circuit so that it conforms to the foundry's requirements. Unfortunately, this iterative process between the foundry and the circuit designers can cause substantial delays in the time-to-market for the circuit design. Hence, before a circuit design is taped-out and provided to a foundry for manufacturing, it is critical to ensure that the circuit design meets all of the foundry's requirements.

[0032] Unfortunately, project deadlines are often missed because of deficiencies in the technology files and the libraries, and/or because the circuit design did not satisfy a foundry requirement. Some embodiments of the present invention provide systems and techniques that help to identify deficiencies in the technology files and/or the libraries during the project start phase. Further, some embodiments enable circuit designers to ensure that the taped-out circuit design meets all of the foundry's requirements.

[0033] FIG. 2A illustrates how project start can be accelerated by using update utilities in accordance with an embodiment of the present invention. (Note that FIG. 2A and FIG. 2B illustrate technology files and libraries 204 as a single object for the sake of clarity. Technology files and libraries are usually separate objects, which may be provided by different entities.)

[0034] Foundry 202 can provide technology files and libraries 204 to customer 206. Unfortunately, technology files and libraries 204 often contain deficiencies which can cause substantial delays in circuit design projects. To overcome this problem, EDA software vendor 208 can receive a copy of technology files and libraries 204. Next, EDA software vendor 208 can check technology files and libraries 204 to identify any deficiencies. EDA software vendor 208 can then construct update utilities 210 that fix the deficiencies. For example, update utilities 210 can be a set of Tcl/Tk scripts that when executed modify the technology files and libraries to fix the deficiencies. Next, EDA software vendor 208 can provide update utilities 210 with EDA software 212 to customer 206. When customer 206

uses EDA software 212, update utilities 210 can be automatically executed whenever they are required to fix deficiencies in the technology files and libraries 204.

[0035] FIG. 2B illustrates how tape-out can be accelerated by performing checks during tape-out in accordance with an embodiment of the present invention.

5 [0036] Foundry 202 can provide the foundry's requirements 252 to customer 206. Customer 206 can provide taped-out circuit design 256 to foundry 202, which can then perform a variety of checks. Unfortunately, the customer's circuit design often fails the checks that the foundry performs when it receives the circuit design from the customer. To overcome this problem, EDA software vendor 208 can receive the foundry's requirements
10 252. Next, EDA software vendor 208 can construct tape-out scripts 254 (e.g., Tcl/Tk scripts) which check whether a circuit design meets the foundry's requirements 252. EDA software vendor 208 can then provide tape-out scripts 254 with EDA software 212 to customer 206. When customer 206 uses EDA software 212, tape-out scripts 254 can be automatically executed during tape-out to ensure that the circuit design meets the foundry's requirements
15 252.

Accelerating Project Start

[0037] FIG. 3 presents a flowchart that illustrates a process to accelerate project start for a circuit design project which uses electronic design automation software in accordance
20 with an embodiment of the present invention.

[0038] The process can begin by receiving a set of technology files and a set of libraries which are intended for use with a specific semiconductor manufacturing technology (block 302). The technology files usually describe parameters for a semiconductor manufacturing technology, and the set of libraries usually include pre-designed circuit
25 designs. Further, the technology files are usually received from a foundry that is expected to manufacture chips based on the circuit design, and the set of libraries can be received from the foundry or a third party library vendor.

[0039] The system can then identify deficiencies in the set of technology files and the set of libraries (block 304). The libraries can include standard cells, I/O cells, and memories.
30 A deficiency in a library can be a design rule violation. Note that the foundry creates design rules. The library vendors typically try to provide cells in the libraries so that they satisfy the design rules. However, the libraries usually contain deficiencies, e.g., design rule violations, which need to be identified and fixed before the library can be used in the circuit design process. In the absence of update utilities, this process of identifying and fixing deficiencies

in the libraries and the technology files can take up a large amount of resources. Hence, the update utilities dramatically reduce the amount of time that customers need to spend on fixing such deficiencies.

[0040] Next, the system can construct scripts that when executed by a computer system cause the computer system to fix the deficiencies in the technology files and the set of libraries (block 306). Specifically, the scripts (e.g., Tcl/Tk scripts) can be constructed by configuring script templates using parameter values that are associated with a specific semiconductor manufacturing technology. The script templates can include instructions that fix the most common types of deficiencies that are found in technology files and libraries. However, to fix the technology files and/or the libraries for a particular foundry, the script templates need to be populated with parameter values that are specific to the semiconductor manufacturing technology that is used by the foundry.

[0041] The system can then provide the scripts to the customer with the electronic design automation software, thereby causing the scripts to automatically fix the deficiencies in the technology files and the set of libraries when the customer uses the electronic design automation software (block 308). Specifically, the update utilities can also automatically set up the design flow for the customer so that the libraries and technology files are fixed by update utilities before they are used by the tools in the design flow. A repair or rebuilding process may need to be performed after the update utility fixes the deficiencies in the technology files and/or libraries.

[0042] When customers receive updated technology files and libraries, it can cause new errors, e.g., DRC errors, LVS errors, timing problems, etc., in the circuit design. The update utilities can significantly accelerate solving these problems. The circuit designers are typically hard pressed to meet their own deadlines when they receive the updates, and hence, using an update to a library that can cause new errors can be very disruptive to the circuit design process. At the same time, the updated technology files and/or libraries may improve performance of the system, and hence, by not using the updated technology files and/or libraries, the circuit designers may forgo performance improvements in the chip design. By using update utilities, the circuit designers can enjoy the benefits of the updated library without worrying about errors in the library because the update utility has been specifically constructed to fix any deficiencies in the updated library.

[0043] FIG. 4 presents a flowchart that illustrates how scripts can be updated in response to receiving updates for a set of technology files and/or a set of libraries in accordance with an embodiment of the present invention.

[0044] The process can begin by receiving an update to the set of technology files and/or the set of libraries (block 402).

[0045] Next, the system can apply the update to the set of technology files and/or the set of libraries to obtain a set of updated technology files and/or a set of updated libraries (block 404).

[0046] The system can then identify deficiencies in the set of updated technology files and/or the set of updated libraries (block 406).

[0047] Next, the system can update the scripts based on the deficiencies in the set of updated technology files and/or the set of updated libraries to obtain updated scripts that when executed by the computer system cause the computer system to fix deficiencies in the set of updated technology files and/or the set of updated libraries (block 408).

[0048] The system can then provide the updated scripts to the customer, thereby causing the updated scripts to automatically fix the deficiencies in the updated technology files and/or the set of updated libraries when the customer uses the electronic design automation software (block 410).

[0049] Note that some embodiments of the present invention are based on the following insight: technology files and libraries often have the same types of deficiencies that need to be fixed. Hence, if the update utility scripts are designed so that they are configurable, they can be quickly configured to fix the deficiencies for different foundries or third party library vendors. Additionally, the updates to the technology files and/or libraries may also contain deficiencies that are similar to those that were present in the original technology files and/or libraries. Hence, at least some portions of the update utilities can be re-used for fixing deficiencies in the updates to technology files and/or libraries.

Additionally, as mentioned above, the update utilities are highly configurable so that when the EDA software vendor receives technology files for a different foundry, the EDA software vendor can use configurable templates to quickly generate update utilities for that particular foundry.

Accelerating Tape-out

[0050] The EDA design flow can include a large number of configuration variables which are specifically designed to understand all of the different design parameters and analysis parameters that need to be checked to tape-out to the foundry. For example, the foundry may require that the customer run a timing analysis at four different operating corners, e.g., four different temperature and voltage corners. Configuration variables in the

EDA design flow can allow a user to program these operating corners. Similarly, other parameters may be related to timing, power, design for test, manufacturability.

[0051] FIG. 5 presents a flowchart that illustrates a process to check a circuit design during tape-out in accordance with an embodiment of the present invention.

5 [0052] The process can begin by receiving a set of checks that are performed by a foundry when the foundry receives a circuit design (block 502). The checks can include netlist structure checks, formal verification checks, floorplan checks, library checks, static timing analysis checks, power integrity checks, signal integrity checks, and physical verification checks.

10 [0053] Next, the system can construct scripts which when executed by a computer cause the computer to perform the set of checks on the circuit design (block 504).

[0054] The system can then provide the scripts to a customer with the electronic design automation software, thereby causing the scripts to automatically perform the set of checks on the circuit design when the customer uses the electronic design automation
15 software to tape-out the circuit design (block 506).

[0055] When the user uses the electronic design automation software, the system can automatically perform the set of checks on the circuit design during tape-out to obtain a set of tape-out check results (block 508). Alternatively, the customer can manually execute some or all of the scripts during tape-out to ensure that the taped-out circuit design (e.g., GDS-II
20 data) meets the foundry's requirements.

[0056] Next, the system can generate a report which includes the set of tape-out check results (block 510). The circuit design can be modified if it fails one or more of the tape-out checks. Once the circuit design passes all of the checks, the circuit design can be taped-out, and the taped-out data can be provided to the foundry. In some embodiments, the
25 system can provide the report to the foundry (block 512). Note that, the tape-out is accelerated because the customer ensures that the circuit design passes the tape-out check scripts before providing the taped-out circuit design to the foundry.

[0057] A foundry may perform its checks regardless of whether the customer has performed the tape-out checks or not. However, if the customer indicates that certain tape-out checks have been performed, the foundry can better prioritize its checks, e.g., the foundry
30 may perform checks that the customer has not performed before it performs the checks that the customer has already performed.

[0058] In some embodiments, the EDA software may digitally sign the results of the tape-out checks. The customer may then present the digitally signed results to the foundry to

prove that the tape-out checks were actually performed, and that the circuit design actually passed the tape-out checks.

Computer System

[0059] FIG. 6 illustrates a computer system in accordance with an embodiment of the present invention.

[0060] Computer system 602 comprises processor 604, memory 606, and storage 608. Computer system 602 can be coupled with display 614, keyboard 610, and pointing device 612. Storage 608 can store instructions and/or data that when processed by processor 604 can cause computer system 602 to help accelerate project start and tape-out.

[0061] Specifically, storage 608 can store EDA software 616, circuit design 618, technology files 620, libraries 622, update utilities 624, and tape-out scripts 626. During operation, EDA software 616 can be used to create circuit design 618, and technology files 620 and libraries 622 can be used during the circuit design process. Specifically, EDA software 616 can use update utilities 624 to fix any deficiencies in technology files 620 and libraries 622. Once the design is ready to be taped-out, EDA software 616 can execute tape-out scripts 626 to check whether circuit design 618 satisfies the foundry's requirements. The modules shown in FIG. 6 are for illustrative purposes only and are not intended to limit the invention to the forms disclosed.

CONCLUSION

[0062] The data structures and code described in this detailed description are typically stored on a computer-readable storage medium, which may be any device or medium that can store code and/or data for use by a computer system. The computer-readable storage medium includes, but is not limited to, volatile memory, non-volatile memory, magnetic and optical storage devices such as disk drives, magnetic tape, CDs (compact discs), DVDs (digital versatile discs or digital video discs), or other media capable of storing computer-readable media now known or later developed.

[0063] The methods and processes described in the detailed description section can be embodied as code and/or data, which can be stored in a computer-readable storage medium as described above. When a computer system reads and executes the code and/or data stored on the computer-readable storage medium, the computer system performs the methods and processes embodied as data structures and code and stored within the computer-readable storage medium.

[0064] Furthermore, the methods and processes described below can be included in hardware modules. For example, the hardware modules can include, but are not limited to, application-specific integrated circuit (ASIC) chips, field-programmable gate arrays (FPGAs), and other programmable-logic devices now known or later developed. When the hardware modules are activated, the hardware modules perform the methods and processes included within the hardware modules.

[0065] The foregoing descriptions of embodiments of the present invention have been presented only for purposes of illustration and description. They are not intended to be exhaustive or to limit the present invention to the forms disclosed. Accordingly, many modifications and variations will be apparent to practitioners skilled in the art. Additionally, the above disclosure is not intended to limit the present invention. The scope of the present invention is defined by the appended claims.

What Is Claimed Is:

1. A method to accelerate project start for a circuit design project which uses electronic design automation software, the method comprising:

5 receiving a set of technology files and a set of libraries, which are intended for use with a specific semiconductor manufacturing technology;

identifying deficiencies in the set of technology files and the set of libraries;

constructing scripts that when executed by a computer system cause the computer system to fix the deficiencies in the technology files and the set of libraries; and

10 providing the scripts to the customer with the electronic design automation software, thereby causing the scripts to automatically fix the deficiencies in the technology files and the set of libraries when the customer uses the electronic design automation software.

2. The method of claim 1, wherein the technology files describe parameters for a semiconductor manufacturing technology, and wherein the libraries include pre-designed
15 cells.

3. The method of claim 1, wherein the script is a Tcl/Tk script.

4. The method of claim 1, wherein constructing scripts involves configuring script templates using parameter values that are associated with the specific semiconductor manufacturing technology.

20 5. The method of claim 1, wherein the technology files are received from a foundry which is expected to manufacture chips based on the circuit design, and wherein the set of libraries are received from a third party library vendor.

6. The method of claim 1, further comprising:

receiving an update to the set of technology files;

25 applying the update to the set of technology files to obtain a set of updated technology files;

identifying deficiencies in the set of updated technology files;

updating the scripts based on the deficiencies in the set of updated technology files to obtain updated scripts that when executed by the computer system cause the computer system to fix deficiencies in the set of updated technology files; and

providing the updated scripts to the customer, thereby causing the updated scripts to automatically fix the deficiencies in the updated technology files when the customer uses the electronic design automation software.

7. The method of claim 1, further comprising:

receiving an update to the set of libraries;

applying the update to the set of libraries to obtain a set of updated libraries;

10 identifying deficiencies in the set of updated libraries;

updating the scripts based on the deficiencies in the set of updated libraries to obtain updated scripts that when executed by the computer system cause the computer system to fix deficiencies in the set of updated libraries; and

providing the updated scripts to the customer, thereby causing the updated scripts to automatically fix the deficiencies in the updated libraries when the customer uses the electronic design automation software.

8. A computer-readable storage medium storing instructions that when executed by a computer cause the computer to perform a method to accelerate project start for a circuit design project which uses electronic design automation software, the method comprising:

20 receiving a set of technology files and a set of libraries which are intended for use with a specific semiconductor manufacturing technology;

identifying deficiencies in the set of technology files and the set of libraries;

constructing scripts that when executed by a computer system cause the computer system to fix the deficiencies in the technology files and the set of libraries; and

25 providing the scripts to the customer with the electronic design automation software, thereby causing the scripts to automatically fix the deficiencies in the technology files and the set of libraries when the customer uses the electronic design automation software.

9. The computer-readable storage medium of claim 8, wherein the technology files describe parameters for a semiconductor manufacturing technology, and wherein the libraries include pre-designed cells.

10. The computer-readable storage medium of claim 8, wherein the script is a Tcl/Tk script.

11. The computer-readable storage medium of claim 8, wherein constructing scripts involves using parameters associated with the specific semiconductor manufacturing
5 technology to configure script templates.

12. The computer-readable storage medium of claim 8, wherein the technology files are received from a foundry which is expected to manufacture chips based on the circuit design, and wherein the set of libraries are received from a third party library vendor.

13. The computer-readable storage medium of claim 8, wherein the method
10 further comprises:
receiving an update to the set of technology files;
applying the update to the set of technology files to obtain a set of updated technology files;
identifying deficiencies in the set of updated technology files;
15 updating the scripts based on the deficiencies in the set of updated technology files to obtain updated scripts that when executed by the computer system cause the computer system to fix deficiencies in the set of updated technology files; and
providing the updated scripts to the customer, thereby causing the updated scripts to automatically fix the deficiencies in the updated technology files when the customer uses the
20 electronic design automation software.

14. The computer-readable storage medium of claim 8, wherein the method further comprises:
receiving an update to the set of libraries;
applying the update to the set of libraries to obtain a set of updated libraries;
25 identifying deficiencies in the set of updated libraries;
updating the scripts based on the deficiencies in the set of updated libraries to obtain updated scripts that when executed by the computer system cause the computer system to fix deficiencies in the set of updated libraries; and

providing the updated scripts to the customer, thereby causing the updated scripts to automatically fix the deficiencies in the updated libraries when the customer uses the electronic design automation software.

15. A method to accelerate tape-out of a circuit design, wherein the circuit design
5 is designed and taped-out using electronic design automation software, the method comprising:

receiving a set of checks that are performed by a foundry;

constructing scripts that when executed by a computer cause the computer to perform
the set of checks on the circuit design; and

10 providing the scripts to a customer with the electronic design automation software,
thereby causing the scripts to automatically perform the set of checks on the circuit design
when the customer uses the electronic design automation software to tape-out the circuit
design.

16. The method of claim 15, further comprising:

15 automatically performing the set of checks on the circuit design during tape-out to
obtain a set of tape-out check results;

generating a report which includes the set of tape-out check results; and

providing the report to the foundry.

17. The method of claim 15, wherein the set of checks includes:

20 netlist structure checks;

formal verification checks;

floorplan checks;

library checks;

static timing analysis checks;

25 power integrity checks;

signal integrity checks; and

physical verification checks.

18. A computer-readable storage medium storing instructions that when executed
by a computer cause the computer to perform a method to accelerate tape-out of a circuit

design, wherein the circuit design is designed and taped-out using electronic design automation software, the method comprising:

receiving a set of checks that are performed by a foundry;

constructing scripts which when executed by a computer cause the computer to

5 perform the set of checks on the circuit design; and

providing the scripts to a customer with the electronic design automation software, thereby causing the scripts to automatically perform the set of checks on the circuit design when the customer uses the electronic design automation software to tape-out the circuit design.

10 19. The computer-readable storage medium of claim 18, wherein the method further comprises:

automatically performing the set of checks on the circuit design during tape-out to obtain a set of tape-out check results;

generating a report which includes the set of tape-out check results; and

15 providing the report to the foundry, thereby accelerating tape-out of the circuit design.

20 20. The computer-readable storage medium of claim 18, wherein the set of checks includes:

netlist structure checks;

formal verification checks;

20 floorplan checks;

library checks;

static timing analysis checks;

power integrity checks;

signal integrity checks; and

25 physical verification checks.

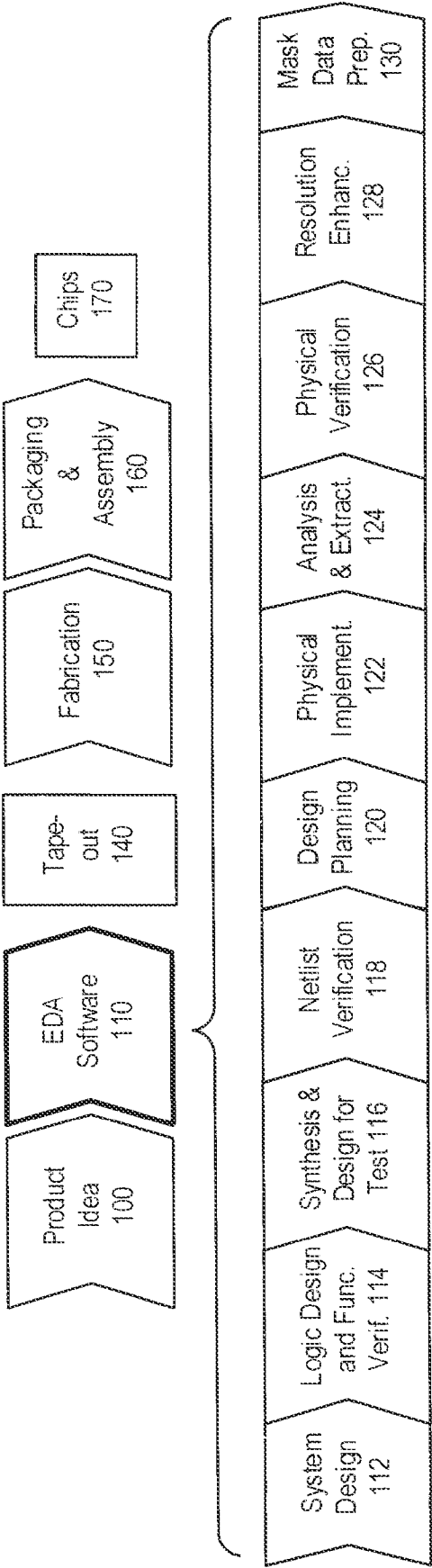


FIG. 1

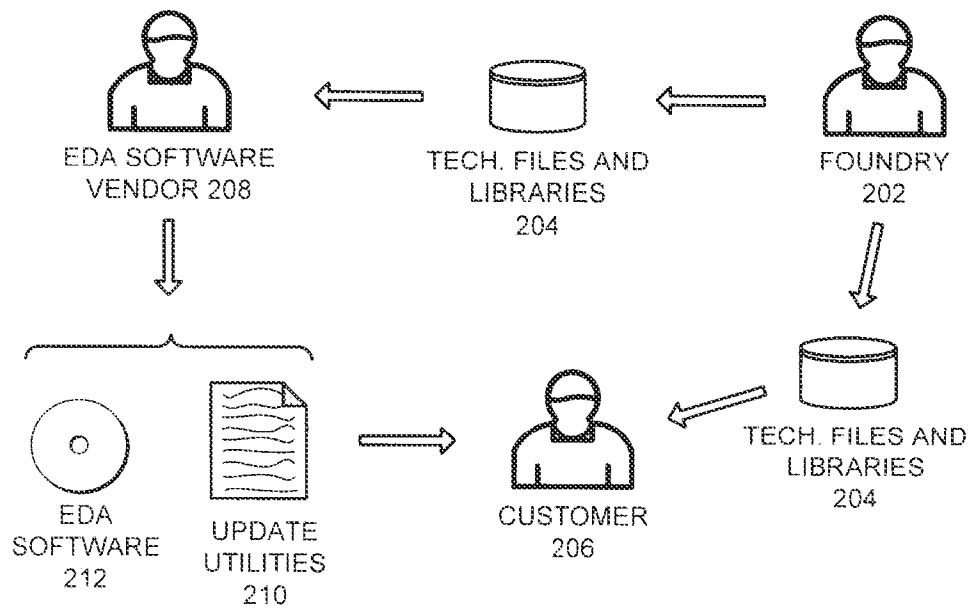


FIG. 2A

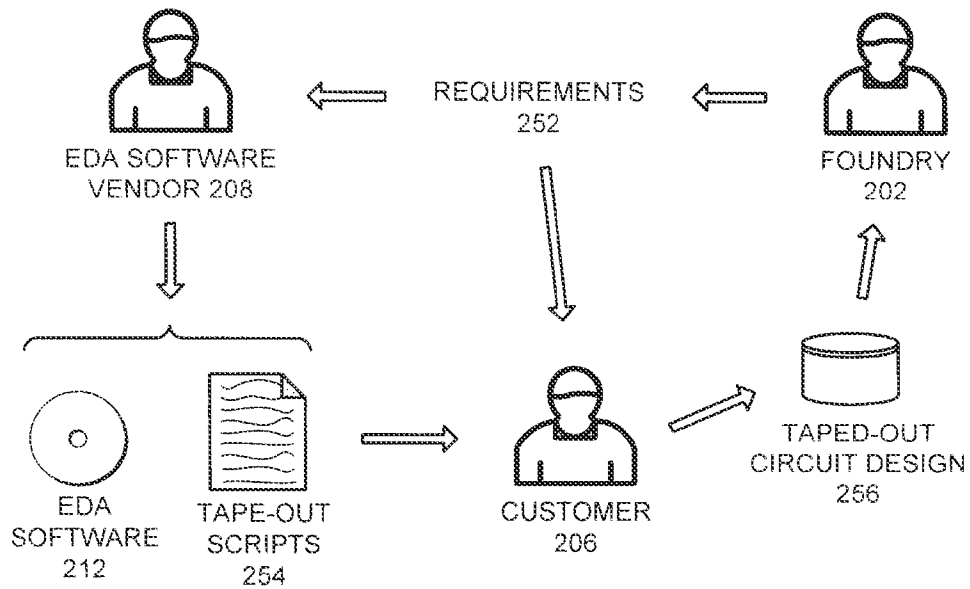
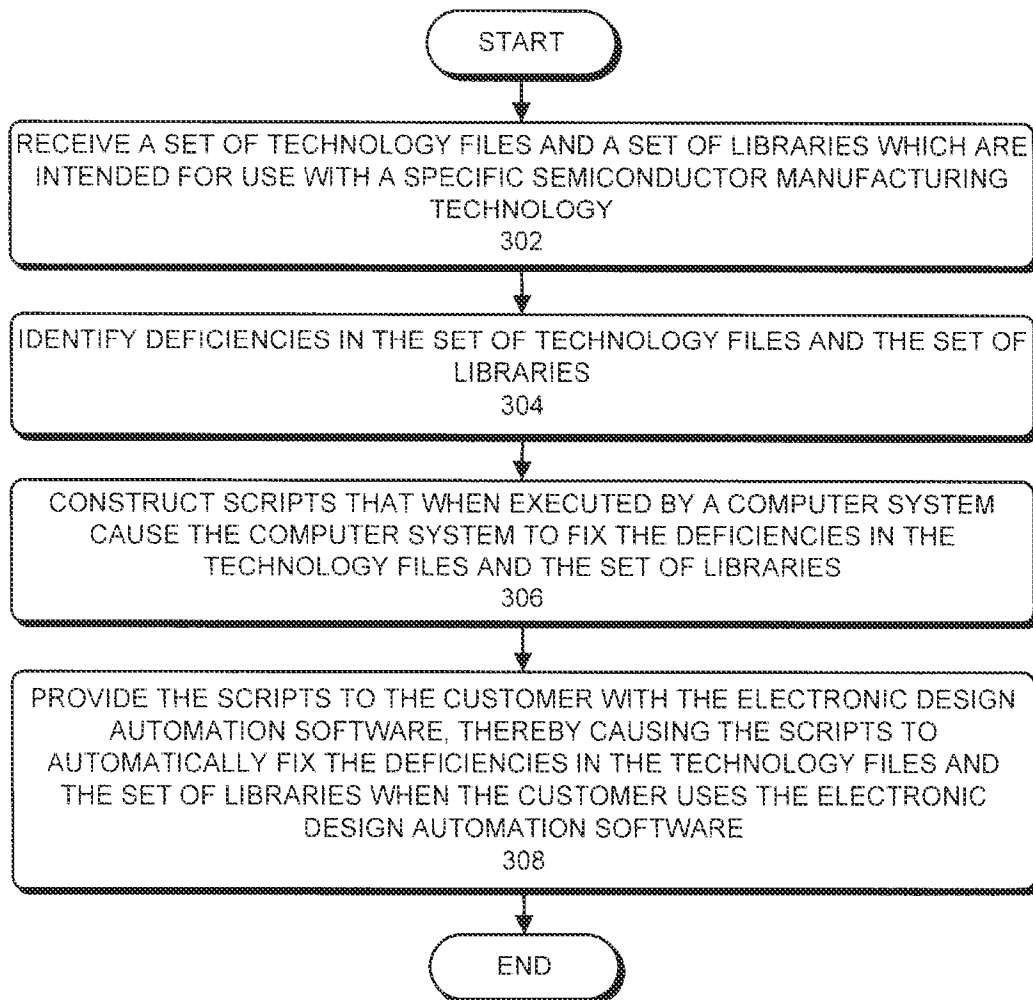


FIG. 2B

**FIG. 3**

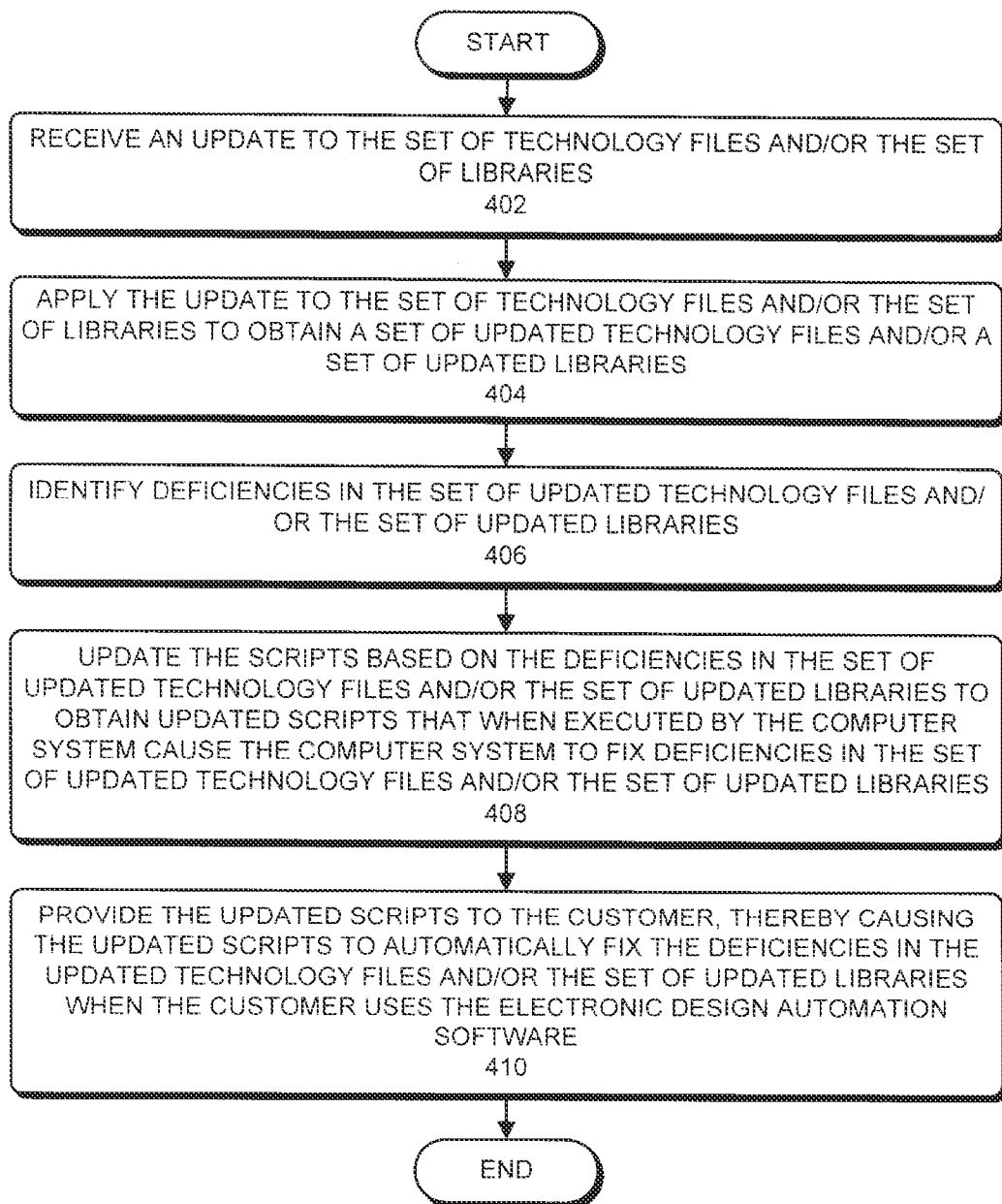
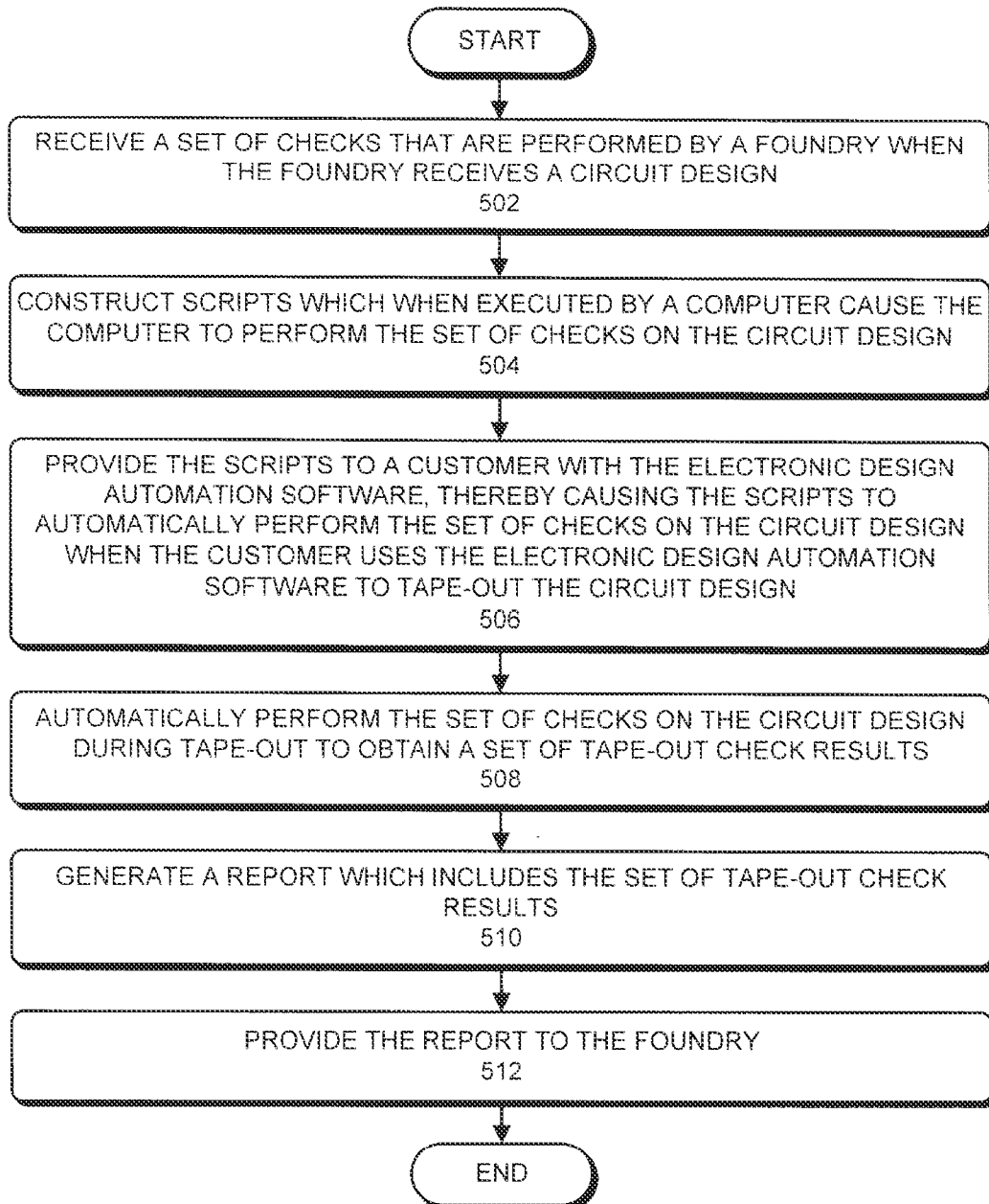


FIG. 4

**FIG. 5**

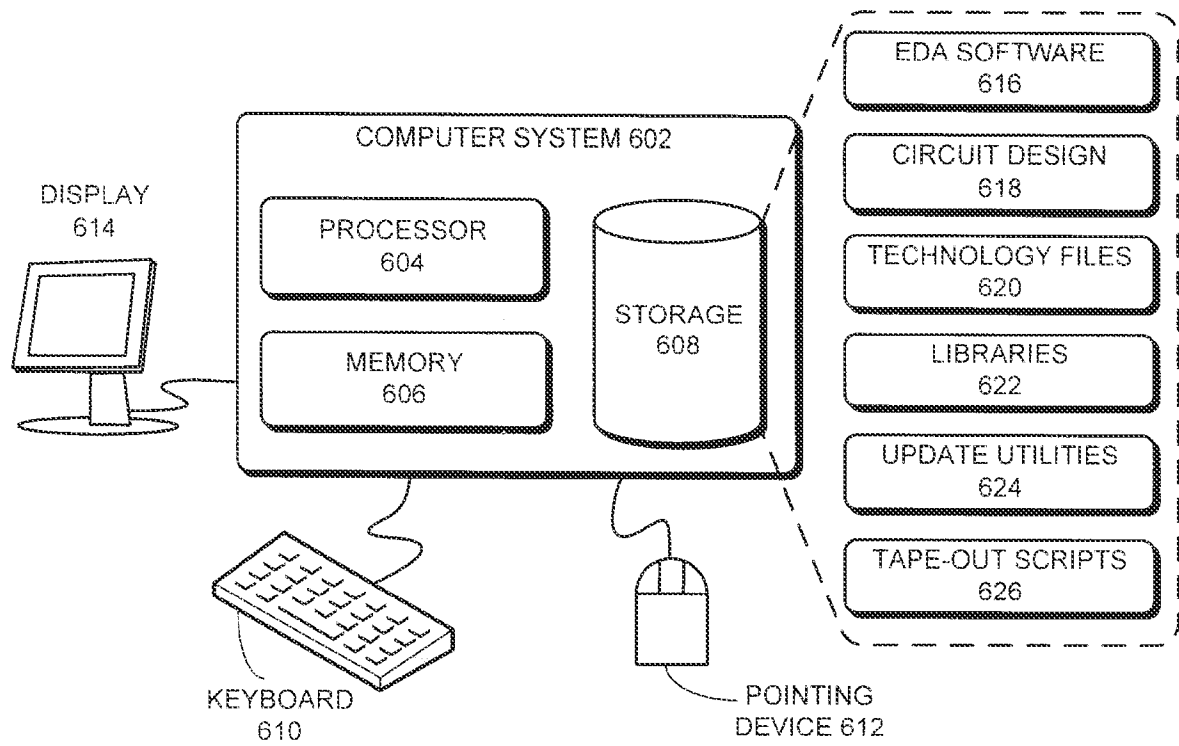


FIG. 6