

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
13 April 2006 (13.04.2006)

PCT

(10) International Publication Number
WO 2006/039286 A1

(51) International Patent Classification:
G06F 17/50 (2006.01)

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

(21) International Application Number:
PCT/US2005/034637

(22) International Filing Date:
28 September 2005 (28.09.2005)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
10/956,327 30 September 2004 (30.09.2004) US

(71) Applicant (for all designated States except US): SYNPPLICITY, INC. [US/US]; 600 W. California Street, Sunnyvale, CA 94086 (US).

(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, HU, IE, IS, IT, LT, LU, LV, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

(72) Inventor; and

Published:

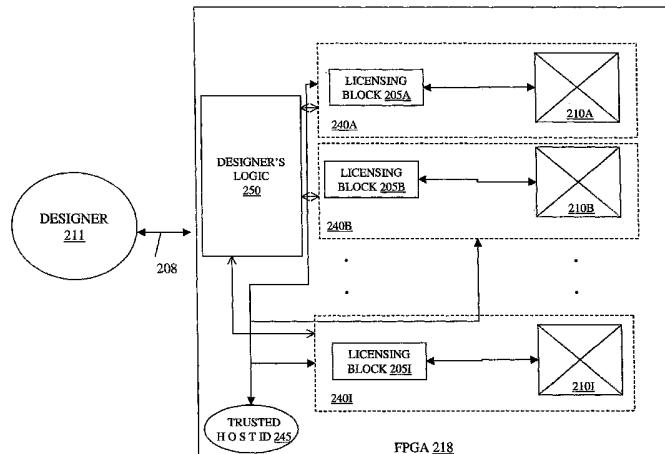
(75) Inventor/Applicant (for US only): MCELVAIN, Kenneth, S. [US/US]; 1209 Via Huerta, Los Altos, CA 94024 (US).

- with international search report
- before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments

(74) Agents: SCHELLER, James, C. et al.; Blakely, Sokoloff, Taylor & Zafman LLP, 12400 Wilshire Boulevard, 7th floor, Los Angeles, CA 90025 (US).

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: APPARATUS AND METHOD FOR LICENSING PROGRAMMABLE HARDWARE SUB-DESIGNS USING A HOST-IDENTIFIER



WO 2006/039286 A1

(57) Abstract: Methods and apparatuses for enforcing terms of a licensing agreement between a plurality of parties involved in a particular hardware design through the use of hardware technologies. According to one embodiment, a hardware subdesign includes a license verification sub-design that is protected from user modification by encryption. In one embodiment, a license is generated based on a trusted host identifier within an external hardware device. In one embodiment, each trusted host identifier is unique, and no two integrated circuits share the same trusted host identifier. In another embodiment, the integrated circuit is a field programmable gate array or an application specific integrated circuit. In one embodiment, a license determines how long the hardware sub-design will operate when the hardware sub-design is implemented within an integrated circuit having a trusted host identifier.

APPARATUS AND METHOD FOR LICENSING PROGRAMMABLE
HARDWARE SUB-DESIGNS USING A HOST-IDENTIFIER

FIELD OF THE INVENTION

[0001] The present invention relates generally to the field of designing integrated circuits, and more particularly to hardware technologies for enforcing terms of a licensing agreement between a plurality of parties involved in a particular hardware design.

BACKGROUND OF THE INVENTION

[0002] The design of complex hardware systems involves a combination of creating new hardware sub-designs (e.g., portions of a proprietary hardware design code and supporting documentation) from scratch and purchasing hardware sub-designs that are prepared by an IP block owner. In many cases, a designer purchases hardware sub-designs from an IP block owner (e.g., organizations/persons which own the intellectual property rights to hardware sub-designs) rather than develop a particular hardware sub-design internally. IP block owners face several challenges in licensing and maintaining control of their intellectual property (e.g., hardware sub-designs). One of the biggest challenges is the unauthorized use of and lack of effective ways of licensing hardware sub-designs on a per use basis. Unfortunately, FPGAs (field-programmable gate arrays) and ASICs (application specific integrated circuits) do not have any trusted and fixed unique identifiers, and as a result hardware sub-designs cannot be easily be licensed on a per use basis. (e.g., a per use license would require an accounting of every device a particular hardware sub-design is implemented on, and a lack of trusted unique identifiers makes it difficult to peg a particular license to a particular FPGA). Instead, IP block owners must often sell global one-time licenses (e.g., a global one-time license to use a particular hardware sub-design on any and all FPGAs at a designer) unless the IP block owner trusts (e.g., through a long standing business relationship) that a designer will properly account for and pay for each use of the hardware sub-design. Furthermore, rampant theft pervades the industry because of a lack of ways to track where hardware sub-designs are copied and applied (e.g., once a hardware sub-design is sold, an IP block owner is unable to monitor and determine how many instances of a particular hardware sub-design

were made on FPGAs). Thus, IP block owners are unable to maximize their leverage of intellectual property rights in hardware sub-designs, and are unable to control where their hardware sub-designs are ultimately utilized.

[0003] Figure 1A illustrates a FPGA 118 having a plurality of IP blocks 110A-110I. The FPGA 118 is coupled to designer 111 through bus 108. A designer 111 (e.g., a person or company who is creating a hardware system) transfers hardware sub-designs (e.g., the designer 111 may reuse a particular hardware sub-design that was previously created by the designer 111, may develop a custom design, may use a design in the public domain, or may use a design purchased from or created by a third party IP block owner) to the FPGA 118 through bus 108 (e.g., the designer 111 may transfer a representation of the design to the FPGA 118 to program one or more IP blocks 110A-110I and/or the designer's logic 150).

[0004] The FPGA 118 includes a designer's logic 150 and a plurality of IP blocks 110A-110I. The designer's logic 150 includes hardware sub-designs that have been created by a designer 111. Each IP block 110A-110I within FPGA 118 includes a hardware sub-design that has been purchased from a third-party. (e.g., the hardware sub-design within a particular IP block 110A may be purchased from a third party IP block owner 100 as illustrated in Figure 1B for example).

[0005] Figure 1B shows a prior art transaction flow diagram between a designer 111, IP block owner 100, and FPGA provider 120 for the implementation of a design. In Figure 1B, a designer 111 communicates first with the FPGA provider 120 as illustrated in circled 1. The communication, circled 1, may be a request to purchase a FPGA device 118 from the FPGA provider 120 (e.g., the FPGA provider may be any commercial supplier of FPGAs such as a distributor, retailer, manufacturer, and/or wholesaler). The FPGA provider 120 then ships to the designer 111 a FPGA 118 in circled 2. (e.g., the FPGA provider 120 may perform a credit check, generate an invoice, and accept an offer as part of a business transaction between designer 111 and FPGA provider 120).

[0006] After the designer 111 receives the FPGA 118 from the FPGA provider 120 in circled 2, the designer 111 sends a request to an IP block owner 100 to purchase a proprietary hardware sub-design that the designer 111 requires to complete his hardware system in circled 3. (e.g., an IP block owner 100 may have ownership rights in the proprietary hardware sub-design that the designer 111

needs to complete his hardware system within the time and cost constraints afforded to him). After the designer 111 provides a request to the IP block owner 100 in circled 3 (e.g., the designer 111 may electronically send to the IP block owner 100 an offer to purchase the proprietary hardware sub-design in the form of a purchase order or may telephone the IP block owner 100 to purchase the proprietary hardware sub-design), the IP block owner 100 provides the proprietary hardware sub-design to the designer 111 in circled 4 (e.g., the IP block owner 100 may enter into a contract for a global license and provide the HDL and/or RTL implementation design details and corresponding documentation necessary to implement the proprietary hardware sub-design to the designer 111 after accepting the designer 111's offer to purchase a global intellectual property license for the proprietary hardware sub-design owned by the IP block owner 100).

[0007] The designer 111 then incorporates the proprietary hardware sub-design received from IP block owner 100 into the FPGA 118 (e.g., the designer 111 may program the FPGA 118 which has been purchased by the designer 111 from the FPGA provider 120 through the bus 108 as described previously in Figure 1A). Unfortunately, the IP block owner 100 loses control over his intellectual property after the proprietary hardware sub-design details are sent to the designer 111. Even if the IP block owner 100 encrypts a portion of their proprietary hardware sub-design, the designer 111 can freely replicate and utilize it for as many FPGAs 118 as he/she desires (e.g., by replicating the license code in addition to the encrypted circuit). Furthermore, the IP block owner 100 is unable to determine whether the designer 111 has misappropriated the proprietary hardware sub-design that the IP block owner 100 has provided and is unable to monitor whether the designer 111 has resold the proprietary hardware sub-design to others (e.g., the IP block owner 100 may have invested millions of dollars in the original design of his proprietary hardware sub-design, and may have difficulty in recouping his investment but for misappropriation and/or theft of the proprietary hardware sub-design by the designer 111 because the IP block owner 100 is unable to license his intellectual property to others). Thus, it can be seen that it is desired to provide an improved apparatus and method for protecting proprietary hardware sub-designs created by IP block owners.

SUMMARY

[0008] The present invention discloses methods and apparatuses for enforcing terms of a licensing agreement between a plurality of parties involved in a particular hardware design through the use of hardware technologies. According to one aspect of the present invention, a hardware sub-design includes a license verification sub-design that is protected from user modification by encryption. In one aspect, connections between the licensing verification sub-design and portions of the logic code within the hardware sub-design for disabling an IP block are also encrypted to prevent modification. In one aspect, a license is generated based on a trusted host identifier within an external hardware device. In another aspect, a user cannot modify the trusted host identifier. In one aspect of the invention, each trusted host identifier is unique, and no two integrated circuits share the same trusted host identifier. In another aspect of the invention, the integrated circuit is a field programmable gate array or an application specific integrated circuit.

[0009] According to another aspect of the present invention, an exemplary method uploads a first hardware sub-design that is designed by a designer onto a programmable gate array having a trusted host identifier, as well as uploads a second hardware sub-design that is provided by a third party and which can only function on the programmable gate array having a particular trusted host identifier. In another aspect of the invention, a license is generated based on a group of trusted host identifiers. In one aspect, the license determines how long the hardware sub-design will operate when the hardware sub-design is implemented within an integrated circuit having a trusted host identifier. The present invention also discloses apparatuses, including software media, which may be used to design integrated circuits. For example, the present invention includes digital processing systems which are capable of designing integrated circuits according to the present invention, and the invention also provides machine readable media which, when executed on a digital processing system, such as a computer system, causes the digital processing system to execute a method for designing integrated circuits.

[0010] Other features of the present invention will be apparent from the accompanying drawings and from the detailed description that follows.

BRIEF DESCRIPTION OF THE DRAWINGS

[0011] The present invention is illustrated by way of example and not limitation in the figures of the accompanying drawings in which like references indicate similar elements.

[0012] Figure 1A illustrates a prior art FPGA having a plurality of IP blocks.

[0013] Figure 1B shows a prior art transaction flow diagram between a designer, an IP block owner, and a FPGA provider for implementation of a hardware design.

[0014] Figure 2A illustrates a FPGA having a trusted host identifier, a plurality of licensing blocks, and a plurality of IP blocks associated with each licensing block, according to one exemplary embodiment of the present invention.

[0015] Figure 2B shows a transaction flow diagram between a designer, an IP block owner, and a trusted host ID provider for the implementation of a hardware design using a trusted host identifier and a license code, according to one exemplary embodiment of the present invention.

[0016] Figure 3 shows a hardware diagram of the use of a trusted host identifier and a license code between a computer, a license generator, and a trusted ID provider for the implementation of a design, according to one exemplary embodiment of the present invention.

[0017] Figure 4A is a block diagram of a computer that may be used to implement embodiments of the present invention.

[0018] Figure 4B is a flow chart illustrating operations within a computer at a designer that are used to implement a hardware sub-design within a FPGA having a trusted host identifier, according to one exemplary embodiment of the present invention.

[0019] Figure 4C is a flow chart illustrating operations within a computer at a designer that are used to implement a hardware sub-design within an ASIC using an external hardware device having a trusted host identifier, according to one exemplary embodiment of the present invention.

[0020] Figure 5 illustrates a FPGA having a trusted host identifier, a plurality of licensing blocks, a plurality of clock memories, and a plurality of IP blocks associated with each licensing block and clock memories according to one exemplary embodiment of the present invention.

[0021] Figure 6 is a flow chart illustrating encryption of a license verification sub-design within a hardware sub-design with a trusted host identifier at an IP block owner to prevent modification, according to one exemplary embodiment of the present invention.

[0022] Figure 7 is a flow chart illustrating the generation of a hardware sub-design license code by a license generator at an IP block owner after receiving a trusted host identifier from a designer, according to one exemplary embodiment of the present invention.

[0023] Figure 8 is a flow chart illustrating the generation of a hardware sub-design license code having a time limit by a license generator at an IP block owner after receiving a trusted host identifier from a designer, according to one exemplary embodiment of the present invention.

DETAILED DESCRIPTION

[0024] Methods and apparatuses for hardware technologies used to enforce terms of a licensing agreement between a plurality of parties involved in a particular hardware design are described herein. In the following description, for purposes of explanation, numerous specific details are set forth in order to provide a thorough understanding of the present invention. It will be evident, however, to one skilled in the art that the present invention may be practiced without these specific details. In other instances, well-known structures, processes and devices are shown in block diagram form or are referred to in a summary manner in order to provide an explanation without undue detail. Many of the methods of the present invention may be performed with a digital processing system, such as a conventional general-purpose computer system.

[0025] Figure 2A illustrates a FPGA 218 having a trusted host identifier 245, a plurality of licensing blocks 205A-205I, and a plurality of IP blocks 210A-210I associated with each licensing block 205A-205I, according to one exemplary embodiment of the present invention. A designer 211 communicates with the FPGA 218 through bus 208 (e.g., the designer 211 may compile all parts of a hardware sub-design by synthesizing, placing, routing, and generating a bit stream before downloading to an FPGA through bus 208). The designer 211 may be a hardware system designer who uses hardware sub-designs created internally as well as that purchased from third parties, according to one embodiment (e.g., the

designer 211 may purchase a particular hardware sub-design from an IP block owner 200 and combine it with the designer's logic 250 to make a complete design within a FPGA 218 through bus 208). The bus 208 may be a direct connect local bus (e.g., a computer connected directly to the FPGA through a serial and/or parallel port). Alternatively, the bus 208 may be a local or wide area network from which designer 211 communicates to FPGA 218, according to another embodiment.

[0026] The FPGA 218 includes a designer's logic 250, a trusted host ID 245, and one or more IP locations 240A–240I. The designer's logic 250 includes hardware sub-designs that have been solely created by a designer 211. (e.g., the designer's logic 250 might include hardware sub-designs that have been verified and already tested, or may include hardware sub-designs that have been previously used in other designs). The designer's logic 250 is separated from the IP locations 240A-240I because IP locations 240A-240I may be used for hardware sub-designs that have been designed by a third party in one embodiment.

[0027] A trusted host ID 245 (trusted host identifier) within the FPGA 218 maintains a unique, trusted host identifier for FPGA 218. The trusted host ID 245 may be implemented within a field programmable gate array using a built-in serial number (e.g., an FPGA may be manufactured a built-in serial number or the built-in serial number may be a one time programmable serial number) plus a design component that can be used in a license verification sub-design (e.g., a license verification sub-design within a hardware sub-design) to access that serial number without user visible connections in one embodiment (e.g., to protect the license verification portion of the sub-design from modification by the user). The trusted host ID 245 may be a unique number that has been permanently associated with FPGA 218 in one embodiment (e.g., the trusted host ID may be a number that was preset by a trusted host ID provider such as an FPGA manufacturer during fabrication). In one embodiment, a portion of the trusted host ID 245 may be common for a group of FPGAs (e.g., a mask having a fixed upper set of bits and the trusted host ID may be combined) that have been sold to a designer 211 by a trusted host ID provider 220 as shown in Figure 2B. (e.g., a portion of the numeric host ID may be associated with a particular batch of FPGAs sold by a trusted host ID provider 220 to designer 211 so as to indicate the manufacturing time and/or location of manufacture of the FPGA 218). The identity of the trusted

host ID 245 may be retrievable by designer 211 through bus 208 by querying the FPGA 218 in one embodiment. (e.g., a software program owned by designer 211, or a software program may allow designer 211 to read the contents of the trusted host ID 245). It should be noted that the trusted host ID 245 is unalterable without completely disabling FPGA 218 (e.g., the trusted host ID 245 may be protected against tampering through a circuit breaker that disables the FPGA 218 if someone attempts to change the identity of trusted host ID 245). In one embodiment, the trusted host ID 245 is external to a field programmable gate array and various known techniques can be used for secure communication over insecure channels (e.g., virtual private networking protocol).

[0028] In one embodiment, a designer 211 may compile all parts of a hardware sub-design by synthesizing, placing, routing, and generating a bit stream before downloading to an FPGA 218. Alternatively, in another embodiment for partial reconfiguration, each IP block 205A-205I may be downloaded separately to different locations on FPGA 218. Each IP location 240A-240I includes a licensing block 205A-205I and an IP hardware design block 210A-210I (e.g., or IP block 210A-210I). Each IP location 240A-240I may be connected to designer logic 250. (e.g., to allow a particular IP location 240A to communicate and share hardware sub-design instructions with designer logic 250). The IP locations 240A-240I may be enabled and disabled when a new hardware sub-design is added and/or deleted from within an IP block 210A-210I in one embodiment. (e.g., an IP location 240A may be a logical association of a particular licensing block 205A to an IP block 210A that is created when proprietary hardware sub-designs purchased from an IP block owner as shown in Figure 2B are added and/or deleted within the FPGA 218, as opposed to a physical IP location). The licensing blocks 205A-205I may be used to enable IP blocks 210A-210I in one embodiment. (e.g., a particular licensing block 205A may unlock the contents of the IP block 210A to reveal a hardware sub-design that is provided by an IP block owner 200 as illustrated in circled 4 of Figure 2B).

[0029] The licensing block 205A may receive a license and analyze a trusted host identifier 245 to decide which features of an IP block to enable. In one embodiment, the licensing blocks 205A-205I include license verification hardware (e.g., a license verification sub-design within the hardware sub-design that is encrypted while generating the hardware sub-design) that validates consistency of a

license code received from an IP block owner 200 (as shown in Figure 2B) with the trusted host identifier 245 to enable operation of the hardware sub-design. As such, the licensing blocks 205A-205I may operate as electronic locks which keep out unlicensed designers of intellectual property from accessing hardware sub-designs within the IP blocks 210A-210I by comparing a license code received from an IP block owner (such as an IP block owner 200 as shown in Figure 2B) with an internal encryption scheme within the licensing blocks 205A-205I that is dependent upon the trusted host ID 245 (e.g., the license code may work only on a FPGA having a particular trusted host ID 245 in one embodiment. In one embodiment, the licensing block 205A may receive a license code (e.g. a string of binary data) through bus 208 from the designer 211 who received the license code from an IP block owner 200 (as shown in circled 4 on Figure 2B). In another embodiment, a licensing block 205A may compare the license code with the trusted host ID 245 and the encryption scheme within the licensing block 205A. In one embodiment, the license code may be embedded within an IP block 210A-210I and/or may be embedded within the designer's logic 250.

[0030] The IP blocks 210A-210I may be hardware sub-designs that have been provided by a third party IP block owner, according to one embodiment. (e.g., IP block owner 200 as later will be described in Figure 2B). Different IP block owners may have sold or licensed their hardware sub-designs to a designer 211 in another embodiment to form the IP blocks 210A-210I. (e.g., a first IP block owner may provide the hardware sub-design within IP block 210A, whereas a different IP block owner may provide the hardware sub-design within IP block 210B). Multiple ones of IP blocks 210A-210I may share a common licensing block 205A-205I in another embodiment and/or may allow a common license code to enable their contents. (e.g., the same license code may operate to unlock a licensing block 205A as well as licensing block 205B). The licensing block 205A-205I may include a license verification sub-design that has been encrypted to prevent modification. In one embodiment, the IP blocks 210A-210I will only function on the FPGA 218 having a particular host ID 245 value, and will not function on a different FPGA that has a different host ID value. The licensing blocks 205A-205I may make a decision to operate based on a combination of a trusted host identifier 245 and a license provided by the IP block owner 200 in one embodiment. In the case that the license is not valid and verification fails, the

operation of the hardware sub-design may be prevented by asserting signals that interfere (e.g., causing state machines to reset and memories to never write) with the logic of the hardware-sub-design in one embodiment. The license verification sub-design may be encrypted so that the user is not able to modify any portion of the license verification sub-design (e.g., the design tool flow may prevent editing of this part of the hardware sub-design). In one embodiment, the hardware sub-design is encrypted only for the purpose of protecting a license verification sub-design within the hardware sub-design from modification. In one embodiment, connections between the licensing verification sub-design and portions of the logic code within the hardware sub-design for disabling the IP block may be encrypted to prevent modification.

[0031] Figure 2B shows a transaction flow diagram between a designer 211, an IP block owner 200, and a trusted host ID provider 220 for the implementation of a hardware design using a trusted host ID 245 and a license code, according to one exemplary embodiment of the present invention. In Figure 2B, a designer 211 sends a request to purchase a FPGA with a host ID 218 to a trusted host ID provider 220. In one embodiment, the trusted host ID provider 220 may be a trusted manufacturer such as a FPGA manufacturer that has developed a trustworthy business reputation sufficient so that the IP block owner 200 believes that each FPGA manufactured by the trusted manufacturer has a unique and uneditable host ID 218.

[0032] Alternatively, in another embodiment, the trusted host ID provider 220 may be a company that manufactures an external hardware device (e.g., a dongle and/or microchip), which provides a trusted host identifier when connected to a computer (e.g., such as computer 311 in Figure 3). In circled 1, the designer 211 orders a trusted host ID (or FPGA with trusted host identifier 218) from the trusted host ID provider 220. (e.g., the designer 211 may place an order by sending a purchase order to the trusted host ID provider 220 for the purchase of an FPGA 218 with a built in trusted host identifier 245 for use in a design of a customized integrated circuit). The designer 211 may also inform the trusted host ID provider 220 which IP block owner 200's hardware sub-design the designer 211 intends to use in his/her hardware system in circled 1 (e.g., the designer 211 may communicate to the trusted host ID provider that he/she intends to use a hardware sub-design created by IP block owner 200 for example).

[0033] In circled 2, the trusted host ID provider 220 ships a FPGA with a trusted host ID 218 to the designer 211, or alternatively provides an external hardware device having a trusted host identifier (e.g., the trusted host ID provider 220 may enter into a binding contract with designer 211 and produce a unique host-ID for each FPGA 218 ordered by the designer 211 for example). In one embodiment, the external hardware device may be directly connected to the FPGA and it may not required that the external hardware device be connected to a computer. In one exemplary embodiment, if the trusted host ID provider 220 provides an external hardware device, the designer 211 may need to connect the external hardware device to a computer connected to an ASIC and/or FPGA on which a hardware sub-design owned by an IP block owner 200 is used. (e.g., the designer 211 may connect a peripheral device having a trusted host identifier to a USB port on a computer that the designer 211 may use to write to an ASIC and/or FPGA that uses the hardware sub-design owned by the IP block owner 200). Alternatively, the serial number of a computer may be used as the host identifier in one embodiment. It should be noted that the IP block owner 200 must trust (e.g., have faith) that the host identifiers generated by the external hardware device are unique and unmodifiable. As such, the IP block owner 200 may need to pre-qualify each trusted host ID provider 220 that wishes to generate trusted host identifiers in one embodiment. In another embodiment, an independent trade organization may provide guidelines and qualification of trusted host ID providers.

[0034] Next, in circled 3, the designer 211 informs the IP block owner 200 that the designer 211 has made a purchase of the FPGA w/host ID 218 (or an external device having a trusted host identifier), and the designer 211 places an order for a hardware sub-design owned by IP block owner 200 (e.g., the designer 211 may purchase part of a complex integrated circuit design from the IP block owner 200 so that the designer 211 can save time rather than designing from scratch for example). The designer 211 may also inform the IP block owner 200 of the identity of the trusted host ID (e.g., the designer 211 may run a software program that determines the identity of the trusted host identifier by reading a fixed register bit within the FPGA with host ID 218 and/or on an external hardware device).

[0035] Next, the IP block owner 200 combines the trusted host ID information it receives from designer 211 in circled 3 to generate a trusted host ID

specific license for designer 211 that is customized for the hardware sub-design that the designer 211 wishes to purchase. (e.g., the IP block owner 200 may encrypt the license verification sub-design within the hardware sub-design using the trusted host ID 245 as shown in Figure 2A, and generate a license code that requires that only a particular host identifier 218 be used to validate and enable one or more features within the hardware sub-design provided by the IP block owner 200). In one embodiment, the license verification sub-design may make a decision to operate based on a combination of a trusted host identifier 245 and a license provided by the IP block owner 200. In the case that the license is not valid and verification fails, the operation of the hardware sub-design may be prevented by asserting signals that interfere with the logic of the hardware-sub-design in one embodiment.

[0036] Then, in circled 4, the IP block owner 200 provides the hardware sub-design (e.g., the hardware sub-design owned by the IP block owner 200) along with a license code (e.g., a license code that unlocks one or more features within the hardware sub-design owned by the IP block owner 200) that is customized for a particular FPGA with a unique host ID 218 to designer 211. In one embodiment, the IP block owner 200 generates a license verification sub-design within the hardware sub-design that is encrypted to prevent modification. In one embodiment, a license is generated for a group of trusted host IDs for a variety of FPGAs that the designer 211 wishes to use the hardware sub-design owned by IP block owner 200 on. (e.g., the IP block owner 200 may receive an order from designer 211 for placing the hardware sub-design owned by the IP block owner 200 on a group of FPGAs, each having a unique host ID, and the IP block owner 200 may authorize the use of his hardware sub-design to all qualifying host IDs that the designer 211 has purchased).

[0037] By providing a license that can only work on an FPGA with a trusted host ID 245, the IP block owner 200 is able to better regulate the ways the hardware sub-design, owned by the IP block owner 200, is used (e.g., the IP block owner 200 can restrict the use of his/her hardware sub-design only to certain FPGAs by providing a license to the designer 211 that works with only a certain group of trusted host IDs). Alternatively, the IP block owner 200 may provide a license code that requires an external hardware device that includes a trusted host identifier be connected to a computer at the designer 211 in one embodiment. For

example, the IP block owner 200's hardware sub-design may operate only on an ASIC and/or an FPGA that has the external hardware device connected. As such, the IP block owner 200 is able to charge a variable amount for his/her intellectual property (e.g., hardware sub-designs), by charging on a per use basis rather than for a global license. (e.g., the IP block owner 200 can charge for each FPGA on which their hardware sub-design is used because there is an enforcement mechanism in that every instance of the hardware sub-design is known, as opposed to a one-time fee to license a hardware sub-design for all FPGAs). In addition, by restricting the operation of the hardware sub-design owned by the IP block owner 200 to only licensees having a trusted host identifier, the IP block owner 200 is able to prevent unauthorized sublicensing and/or theft of the hardware sub-designs owned by the IP block owner 200. As such, the designer 211 is unable to generate unauthorized copies of the hardware sub-design(s) that the designer 211 has licensed from the IP block owner 200 without getting prior permission from the IP block owner 200. The designer 211 must purchase a license that explicitly allows the designer 211 to use additional copies because the hardware sub-designs that the designer 211 receives from the IP block owner 200 will only work in conjunction with a particular trusted host identifier. (e.g., the IP block owner 200 is able to prevent the spread of the hardware sub-design owned by the IP block owner 200 because the licenses granted by the IP block owner 200 may be restricted only to FPGAs with particular trusted host IDs).

[0038] Figure 3 shows a hardware diagram of the use of a trusted host identifier and a license code between a computer, a license generator, and a trusted host ID provider for the implementation of a design, according to one exemplary embodiment of the present invention. Figure 3 is similar to Figure 2B in that all the processes occurring in Figure 2B occur in Figure 3, except that Figure 3 illustrates the particular hardware used by the parties shown in Figure 2B. A computer 311 resides within a designer 211 as previously described in Figure 2B (e.g., the computer 311 may communicate with an FPGA with host ID 218 through bus 208 as described in Figure 2A). The computer 311 receives a trusted host ID 245 from a trusted host ID provider 220 in the communications that are illustrated circled 1 and 2, and as previously described in Figure 2B (e.g., a FPGA 218 may be shipped by a trusted host ID provider 220 to a designer 211 as shown in Figure 2B for use in a computer 311 after the parties enter into a contract). A license

generator 300 within the IP block owner 200, in circled 3 and 4, then generates a license that is specific to the trusted host identifier 245 (e.g., the license generator 300 may generate a license that depends on a particular host identifier 245 for a hardware sub-design to be unlocked). In one embodiment, the license generator 300 is an application program that is used by the IP block owner 200 to generate an encrypted license verification sub-design within the hardware sub-design and a customized host ID specific license code that is dependant on the trusted host ID 245 within a particular FPGA 218. In circled 4, the trusted host ID specific license code as well as the hardware sub-design is provided to computer 311 (the computer 311 at the designer 211 as described in Figure 2B) from the license generator 300.

[0039] Figure 4A is a block diagram of a computer system 311 that may be used to implement embodiments of the present invention. The computer system 311 is used to perform logic synthesis of a design that is described in an HDL code. The computer system 311 includes a processor 402 that is coupled through a bus 401 to a random access memory 404 and a read-only memory 406 and a mass storage device 407. Mass storage device 407 represents a persistent data storage device such as a floppy-disk drive, a fixed disk drive (e.g., magnetic drive, optical drive, or the like). Processor 402 may be embodied in a general-purpose processor (such as the Intel Pentium® processors) a special purpose processor and/or a specially programmed logic device. Display 420 is coupled to the processor 402 through bus 401 and provides graphical output for the computer system. This graphical output is typically a graphical user interface that may be used to control the operation of the computer system. Keyboard 421 and cursor control device 422 are coupled to bus 401 for communicating information and command selections to processor 402. The cursor control device 422 will typically be a mouse or other cursor control device that will be used to control a cursor displayed on the display device 420. Also coupled to processor 402 through bus 401 is an input/output interface 423 which can be used to control and transfer data to and from electrical devices such as printers and other computers that are coupled to the computer system 311. The trusted host identifier 245 may also be coupled to the computer system 311. In one embodiment, the trusted host identifier 245 is an external hardware device. In another embodiment, the trusted host identifier is within a FPGA 218 having a trusted host identifier 245. The trusted host identifier

245 may also be coupled to a particular input/output interface 423 rather than directly to the computer system 311.

[0040] It should be noted that the architecture of Figure 4 is provided for purposes of illustration only and that a computer system or other digital processing system used in conjunction with the present invention is not limited to this specific architecture.

[0041] A general example of certain embodiments of the present invention will now be provided while referring to Figure 4B. Figure 4B is a flow chart illustrating operations within a computer 311 at a designer 211 that are used to implement a hardware sub-design within a FPGA 218 having a trusted host identifier 245, according to one exemplary embodiment of the present invention. While most embodiments of the present invention are intended for use in HDL design synthesis software, the invention is not necessarily limited to such use. It should be noted that in one embodiment, the hardware sub-design could in fact be simply represented as a simple netlist with appropriate parts obscured by encryption. The hardware sub-design does not have to be HDL or a behavioral description. Although use of other languages in computer programs is possible, embodiments of the present invention will be described in the context of use in HDL synthesis systems, and particularly those designed for use with integrated circuits which have vendor specific technology/architectures.

[0042] As is well known, a supplier of programmable ICs typically determines the target architecture. An example of a target architecture is the programmed look-up tables (LUTs) and associated logic of field programmable gate arrays from Xilinx, Inc., of San Jose, California. Other examples of target architecture/technology include those well-known architectures in FPGAs and complex programmable logic devices from vendors such as Altera, Lucent Technologies, Advanced Micro Devices, and Lattice Semiconductor. For certain embodiments, the present invention may also be employed with ASICs.

[0043] The method of Figure 4B begins in operation 411 in which a designer (such as a designer 211 as shown in Figure 2B) prepares a design representation (e.g., a behavioral description of a desired circuit) including license validation hardware protected from modification. In operation 412, the design is displayed with the IP block, which is hidden, and changes are allowed to the design except for the IP block which is not editable (e.g., the designer's logic 250

may be editable whereas the IP locations 240A-240I having hardware sub-designs provided by an IP block owner 200 may be hidden and may be partly or completely uneditable as shown in Figure 2A). The design representation is then compiled by an HDL compiler in operation 413 to generate a technology independent RTL netlist. This netlist is independent of the particular vendor's technology which means that it is independent of the library of building blocks (e.g., look-up tables, multiplexers, AND/OR arrays, and so on) that is used by a particular target architecture. At least a portion of the design representation is secure from modification because it has been encrypted.

[0044] In operation 414, the design representation is optimized (e.g., the logic is optimized to improve timing parameters of the logic). This operation is optional and is used to remove duplicative interconnects and logic elements according to optimization routines. Then in operation 415, conventional place and route software tools are used to create a design of circuitry in the target architecture, such as a Xilinx or Altera FPGA. Next in operation 416, configuration data for programming a portion of an FPGA to create an IP block (e.g., an IP block having a hardware-sub-design purchased from an IP block owner 200) including design representation license validation hardware protected from modification is generated.

[0045] Figure 4C is a flow chart illustrating operations within a computer 311 at a designer 211 that are used to implement a hardware sub-design within an ASIC using an external hardware device having a trusted host identifier 245, according to one exemplary embodiment of the present invention. The implementation of hardware sub-designs owned by an IP block owner 200 within an ASIC requires that the designer 211 purchase an external hardware device as previously discussed in Figure 2B. The operations 411-415 shown in Figure 4C are the same as operations shown in Figure 4B. In operation 426 on Figure 4C, a layout of a circuit for implementing within an ASIC an IP block including design representation license validation hardware protected from modification is performed because the hardware sub-design is licensed for use on an ASIC rather than an FPGA.

[0046] Figure 5 illustrates a FPGA 518 having a trusted host identifier 545, a designer's logic 550, a plurality of licensing blocks 505A-505I, a clock circuit 525, and a plurality of IP blocks 510A-510I associated with each licensing block 505A-505I and the clock circuit 525 according to one exemplary embodiment of the

present invention. The FPGA 518 is similar to FPGA 218 as previously described in Figure 2A. However, the FPGA 518 illustrated in Figure 5 includes a clock circuit 525 that communicates with each licensing block 505A-I within IP location 540A-540I. The clock circuit 525 may store a new value (e.g., a count of how many times a particular hardware sub-design has been executed, or the current date/time, etc.) only if it is larger than the previously stored value in one embodiment. In one embodiment the clock circuit 525 cannot be reset and is used only to make sure that recorded time continues to move forward.

[0047] Each IP location 540A-540I includes a licensing block 505A-505I and an IP block 510A-510I. The clock circuit 525 stores information that governs how long a particular IP location 540A-540I will operate (e.g., the clock circuit 525 may store information about how long the IP location 540A may be operational based upon a license code provided to a licensing block 505A by a IP block owner 200 as shown in Figure 2A). In one embodiment, when the limit for a particular operation period is reached, the clock circuit 525 may disable one or more IP blocks 510A-510I. (e.g., similarly as discussed previously in Figure 2A, the IP location 540A-540I may be a logical module that can be enabled for the period of time the IP block 510A-510I associated with the IP location 540A-540I is accessible by a designer 211, and disabled when the license to access the hardware sub-design within a IP hardware design block 510A-510I is expired).

[0048] In one embodiment, the clock circuit 525 may store the last time a particular IP block 510A-510I associated with the clock circuit 525 was updated (e.g., the last time a designer 211 uploaded a particular third party hardware sub-design to an IP block 510A-510I through a bus 508). In another embodiment, the clock circuit 525 may store a countdown timer that determines how long the licensing block 505A is allowed to operate (e.g., how long before the IP location associated with the IP block is disabled). In another embodiment of the present invention, the clock circuit 525 may include the last updated time of a group of IP locations 540A-540I (e.g., if an IP block owner 200 had previously licensed a group of hardware sub-designs to the designer 211, then the clock circuit 525 may store the last updated time for a group of IP locations 540A and 540B for a partial reconfiguration scenario for example). In one embodiment, an external hardware device having host ID 545 may include a clock circuit (e.g., similar to clock circuit 525 for example) that is powered by a battery and allows the hardware sub-designs

within the IP blocks 510A-510I to operate for a predetermined amount of time. In another embodiment, memory within an external hardware device having a trusted host identifier 545 may include all of the embodiments discussed for the clock circuit 525 as discussed above.

[0049] In one embodiment, a designer 211 may use a software application which itself uses a software licensing subsystem (e.g., GLOBETROTTER® by Macromedia, Inc. of Santa Clara, California) to receive an evaluation license for a hardware sub-design from the IP block owner 200. The evaluation license may include a software-based trusted host identifier (e.g., the evaluation license may include the software-based trusted host identifier within additional bits of data sent along with a software license for the application software). The designer 211, using the software application, may then transfer (e.g., via a JTAG serial communication scheme) the software-based trusted host identifier and the evaluation license to the licensing block 505A within the FPGA 518 as shown in Figure 5, so that the licensing block 505A can enable one or more features within the hardware sub-design using the software-based trusted host-identifier. In one embodiment, a trusted host identifier 245 used by an encrypted portion (e.g., the license verification sub-design within the hardware sub-design) of the IP block 210A or the licensing block 205A, may be generated to enable a remaining portion of the IP block 210A (e.g., the hardware sub-design), when the trusted host identifier 245 is provided by software at the time of programming or manufacturing the integrated circuit. In another embodiment, trusted host identifier is provided by the IP block owner 200 to the designer 211 through software at the time of programming or manufacturing an integrated circuit to enable the IP block 210A for example. This software may be run by the designer 211 at the time that the integrated circuit is being programmed, and this software would typically include executable computer program code which, through an authorization message or signal provided by the IP block owner, would cause the integrated circuit to be programmed or manufactured with an enabled IP block. This authorization message or signal may be provided in an encrypted portion of the software or data structures which are utilized by the software, or this authorization message or signal may be provided through a network connection from a data processing system controlled by the IP block owner.

[0050] Figure 6 is a flow chart illustrating encryption of the license verification sub-design within a hardware sub-design with a trusted host identifier 245 at an IP block owner to prevent modification, according to one exemplary embodiment of the present invention. In operation 601, a hardware sub-design for an IP block (e.g., such as IP block 210A in Figure 2A) is designed by an IP block owner 200 as described previously in Figure 2B). The hardware sub-design may be a design that was originally created by an IP block owner 200 in one embodiment. In another embodiment, an IP block owner 200 may own the hardware sub-design but the hardware sub-design originally was created by another company. In operation 602, data is generated to program a FPGA to implement the hardware sub-design within an IP block. A license generator 300 at an IP block owner 200 as described in Figure 3 may be used to generate data to program a FPGA 218 to implement the hardware sub-design within one or more of the IP blocks 210A-210I as shown in Figure 2A in one embodiment.

[0051] Next, in operation 603, the license verification sub-design within the hardware sub-design is encrypted at HDL and/or RTL levels and possibly other levels with host ID info (e.g., to ensure that the hardware sub-design provided by the IP block owner 200 only works on a particular FPGA having the trusted host ID 245 as described in Figure 2A). Lastly, in operation 604, hardware sub-design information is provided to a designer 211 of an FPGA system who will implement the hardware sub-design provided by the IP block owner 200 into an FPGA 218 having the trusted host ID 245 as discussed in Figure 2B, along with the designer's logic 250 that was separately developed. (e.g., the hardware sub-designs created by the designer such as the designer's logic 250 in Figure 2A).

[0052] Figure 7 is a flow chart illustrating the generation of a hardware sub-design license code by a license generator 300 at an IP block owner 200 after receiving a trusted host identifier 245 from a designer 211, according to one exemplary embodiment of the present invention. In operation 701, a license generator 300 receives a request for a hardware sub-design from a computer 311 at a designer 211. (e.g., the designer 211 may have entered into a contract with an IP block owner 200 to supply a hardware sub-design owned by the IP block owner 200). In operation 702, the IP block owner 200 sends a request for a trusted host ID 245 to a designer 211 (e.g., the request for a trusted host ID 245 may be sent by the IP block owner 200 to determine how much to charge the designer 211 for a

particular license). In operation 703, a trusted host ID 245 is received from the designer 211. (e.g., so that the IP block owner 200 can generate a license to use the hardware sub-design only on a particular FPGA 218 having the trusted host ID 245).

[0053] In operation 704, a unique IP block license code is generated based on the trusted host ID 245. (e.g., a unique IP block license code may be generated by the license generator 300). In operation 705, the license verification sub-design within the hardware sub-design owned by the IP block owner 200 may be encrypted. In operation 706, a unique IP block license code and hardware sub-design is provided to a designer 211 from the license generator 300 at the IP block owner 200 (e.g., to fulfill the obligations under a contract between the designer 211 and the IP block owner 200 for example, as described in detail in Figure 2B).

[0054] Figure 8 is a flow chart illustrating the generation of a unique hardware sub-design license having a time limit by a license generator 800 at an IP block owner 200 after receiving a trusted host identifier 245 from a designer 211, according to one exemplary embodiment of the present invention. Figure 8 differs from Figure 7 in that the license generator 800 additionally generates a time limit for a license based on a request from a designer 211 (e.g., a time limit for how long a particular hardware sub-design owned by an IP block owner 200 will operate and that is stored within clock circuit 525 as described in Figure 5 and/or on an external hardware device). Operations 701-704 are exactly the same as previously described in Figure 7. In operation 801, a time limit for a license based on a request from a designer 211 is generated. (e.g., the time limit may be based on how much a designer is willing to pay an IP block owner 200). In operation 802, a license verification sub-design within a hardware sub-design owned by the IP block owner 300 is encrypted to prevent modification, similar to operation 705 as previously described in Figure 7. In operation 803, a unique IP block license having a time limit, and the hardware sub-design is provided to the designer 211. As such, Figure 8 differs from Figure 7 in that a time limit license based on a request from a designer is generated. Furthermore, a unique IP block license is provided to the designer 211 in Figure 8 that includes the time limit.

[0055] The operations of the various methods of the present invention may be implemented by a processing unit in a digital processing system which executes sequences of computer program instructions which are stored in a memory which

may be considered to be a machine-readable storage media. The memory may be random access memory, read only memory, a persistent storage memory, such as mass storage device or any combination of these devices. Execution of the sequences of instruction causes the processing unit to perform operations according to the present invention. The instructions may be loaded into memory of the computer from a storage device or from one or more other digital processing systems (e.g., a server computer system) over a network connection. The instructions may be stored concurrently in several storage devices (e.g., DRAM and a hard disk, such as virtual memory). Consequently, the execution of these instructions may be performed directly by the processing unit. In other cases, the instructions may not be performed directly or they may not be directly executable by the processing unit. Under these circumstances, the executions may be executed by causing the processor to execute an interpreter that interprets the instructions, or by causing the processor to execute instructions which convert the received instructions to instructions which can be directly executed by the processor. In other embodiments, hard-wired circuitry may be used in place of or in combination with software instructions to implement the present invention. Thus, the present invention is not limited to any specific combination of hardware circuitry and software, nor to any particular source for the instructions executed by the computer or digital processing system. Furthermore, "Encryption" as used in this document refers to any mechanism that can be used to prevent modification by the designer of the portion of the sub-design involved in licensing. The hardware sub-design may for instance be pre-compiled into a binary database for which no de-compile is allowed.

[0056] Although the present invention has been described with reference to specific exemplary embodiments, it will be evident that various modifications and changes may be made to these embodiments without departing from the broader spirit and scope of the invention as set forth in the claims. Accordingly, the specification and drawings are to be regarded in an illustrative rather than a restrictive sense.

CLAIMS

What is claimed is:

1. A method of generating an encrypted hardware sub-design, comprising:
 - generating a first hardware sub-design at a first designer;
 - generating a second hardware sub-design at a second designer;
 - uploading said first hardware sub-design and said second hardware sub-design to an integrated circuit having a trusted host identifier; and
 - enabling said second hardware sub-design by applying a license that is based on said trusted host identifier.
2. The method as in claim 1, further comprising: encrypting a license verification sub-design within said second hardware sub-design while generating said second hardware sub-design.
3. The method as in claim 1, wherein said license is based on a group of trusted host identifiers.
4. The method as in claim 1, wherein said license determines how long said second hardware sub-design will operate.
5. The method as in claim 1, wherein said trusted host identifier is a unique value to said integrated circuit.
6. The method as in claim 1, wherein an external hardware device provides said trusted host identifier.
7. A machine readable medium containing a plurality of executable instructions, which when executed on a digital processing system cause said digital processing system to perform a method of generating an encrypted hardware sub-design, said method comprising:
 - generating a first hardware sub-design at a first designer;
 - generating a second hardware sub-design at a second designer;

uploading said first hardware sub-design and said second hardware sub-design to an integrated circuit having a trusted host identifier; and enabling said second hardware sub-design by applying a license that is based on said trusted host identifier.

8. The machine-readable medium as in claim 7, further comprising encrypting a license verification sub-design within said second hardware sub-design while generating said second hardware sub-design.
9. The machine-readable medium as in claim 7, wherein said license is based on a group of trusted host identifiers.
10. The machine-readable medium as in claim 7, wherein said license determines how long said second hardware sub-design will operate.
11. The machine-readable medium as in claim 7, wherein said trusted host identifier is a unique value to said integrated circuit.
12. The machine-readable medium as in claim 7, wherein an external hardware device provides said trusted host identifier.
13. A system of generating an encrypted hardware sub-design, comprising:
means for generating a first hardware sub-design at a first designer;
means for generating a second hardware sub-design at a second designer;
means for uploading said first hardware sub-design and said second hardware sub-design to integrated circuit having a trusted host identifier; and
means for enabling said second hardware sub-design by applying a license that is based on said trusted host identifier.
14. The system as in claim 13, further comprising:
means for encrypting a license verification sub-design within said second hardware sub-design while generating said second hardware sub-design.

15. The system as in claim 13, wherein said license is based on a group of trusted host identifiers.
16. The system as in claim 13, wherein said license determines how long said second hardware sub-design will operate.
17. The system as in claim 13, wherein said trusted host identifier is a unique value to said integrated circuit.
18. The system as in claim 13, wherein an external hardware device provides said trusted host identifier.
19. A method, comprising:
 - uploading an encrypted circuit design to an integrated circuit having a trusted host identifier; and
 - enabling said encrypted circuit design based on a trusted host identifier specific license.
20. The method as in claim 19, further comprising:
 - uploading a non-encrypted circuit design to said integrated circuit having said trusted host identifier.
21. The method as in claim 19, wherein said trusted host identifier specific license is based on a group of trusted host identifiers.
22. The method as in claim 19, wherein said trusted host identifier specific license determines how long said encrypted circuit design will operate.
23. The method as in 19, wherein said trusted host identifier is a unique value to said integrated circuit.
24. The method as in claim 19, wherein said integrated circuit is a field programmable gate array or an application specific integrated circuit.

25. A machine readable medium containing a plurality of executable instructions, which when executed on a digital processing system cause said digital processing system to perform a method, said method comprising:
 - uploading an encrypted circuit design to an integrated circuit having a trusted host identifier; and
 - enabling said encrypted circuit design based on a trusted host identifier specific license.
26. The machine readable medium as in claim 25, wherein said method further comprising:
 - uploading a non-encrypted circuit design to said integrated circuit having said trusted host identifier.
27. The machine-readable medium as in claim 25, wherein said trusted host identifier specific license is based on a group of trusted host identifiers.
28. The machine-readable medium as in claim 25, wherein said trusted host identifier specific license determines how long said encrypted circuit design will operate.
29. The machine-readable medium as in claim 25, wherein said trusted host identifier is a unique value to said integrated circuit.
30. The machine-readable medium as in claim 25, wherein said integrated circuit is a field programmable gate array or an application specific integrated circuit.
31. A system, comprising:
 - means for uploading an encrypted circuit design to an integrated circuit having a trusted host identifier; and
 - means for enabling said encrypted circuit design based on a trusted host identifier specific license.

32. The system as in claim 31, further comprising:
means for uploading a non-encrypted circuit design to said integrated circuit
having said trusted host identifier.
33. The system as in claim 31, wherein said trusted host identifier specific license
is based on a group of trusted host identifiers.
34. The system as in claim 31, wherein said trusted host identifier specific license
determines how long said encrypted circuit design will operate.
35. The system as in claim 31, wherein said trusted host identifier is a unique
value to said integrated circuit.
36. The system as in claim 31, wherein said integrated circuit is a field
programmable gate array or an application specific integrated circuit.
37. A method of applying constraints to intellectual property within an integrated
circuit based a trusted host identifier, comprising:
generating a trusted host identifier specific license that enables a hardware
sub-design which can only run for a specific amount of time on said
integrated circuit having said trusted host identifier.
38. The method as in claim 37, further comprising:
uploading said hardware sub-design to said integrated circuit having said
trusted host identifier along with another hardware sub-design that
does not require a license.
39. The method as in claim 37, wherein said trusted host identifier specific license
is based on a group of trusted host identifiers.
40. The method as in claim 37, wherein said trusted host identifier specific license
determines how long said hardware sub-design will operate.

41. The method as in claim 37, wherein said trusted host identifier is a unique value to said integrated circuit.
42. The method as in claim 37, wherein said integrated circuit is a field programmable gate array or an application specific integrated circuit.
43. A machine readable medium containing a plurality of executable instructions, which when executed on a digital processing system cause said digital processing system to perform a method of applying constraints to intellectual property within an integrated circuit based on a trusted host identifier, said method comprising:
generating a trusted host identifier specific license that enables a hardware sub-design which can only run for a specific amount of time on said integrated circuit having said trusted host identifier.
44. The machine-readable medium as in claim 43, said method further comprising: uploading said hardware sub-design to said integrated circuit having said trusted host identifier along with another hardware sub-design that does not require a license.
45. The machine-readable medium as in claim 43, wherein said trusted host identifier specific license is based on a group of trusted host identifiers.
46. The machine-readable medium as in claim 43, wherein said trusted host identifier specific license determines how long said hardware sub-design will operate.
47. The machine-readable medium as in claim 43, wherein said trusted host identifier is a unique value to said integrated circuit.
48. The machine-readable medium as in claim 43, wherein said integrated circuit is a field programmable gate array or an application specific integrated circuit.

49. A system of applying constraints to intellectual property within an integrated circuit based on a trusted host identifier, comprising:
 - means for generating a trusted host identifier specific license that enables a hardware sub-design which can only run for a specific amount of time on said integrated circuit having said trusted host identifier.
50. The system as in claim 49, further comprising:
 - means for uploading said hardware sub-design to said integrated circuit having said trusted host identifier along with another hardware sub-design that does not require a license.
51. The system as in claim 49, wherein said trusted host identifier specific license is based on a group of trusted host identifiers.
52. The system as in claim 49, wherein said trusted host identifier specific license determines how long said hardware sub-design will operate.
53. The system as in claim 49, wherein said trusted host identifier is a unique value to said integrated circuit.
54. The system as in claim 49, wherein said integrated circuit is a field programmable gate array or an application specific integrated circuit.
55. A method, comprising :
 - manufacturing an integrated circuit having a trusted host identifier that cannot be modified by a user.
56. The method as in claim 55, wherein said integrated circuit is a field programmable gate array or an application specific integrated circuit.
57. The method as in claim 55, wherein said integrated circuit includes an encrypted license verification sub-design within a hardware sub-design.

58. A machine readable medium containing a plurality of executable instructions, which when executed on a digital processing system cause said digital processing system to perform a method, the method comprising :
manufacturing an integrated circuit having a trusted host identifier that cannot be modified by a user.
59. The machine-readable medium as in claim 58, wherein said integrated circuit is a field programmable gate array or an application specific integrated circuit.
60. The machine-readable medium as in claim 58, wherein said integrated circuit includes an encrypted license verification sub-design within a hardware sub-design.
61. A system, comprising :
means for manufacturing an integrated circuit having a trusted host identifier that cannot be modified by a user.
62. The system as in claim 61, wherein said integrated circuit is a field programmable gate array or an application specific integrated circuit.
63. The system as in claim 61, wherein said integrated circuit includes an encrypted license verification sub-design within a hardware sub-design.
64. An integrated circuit having a trusted host identifier, comprising:
a first hardware sub-design that is enabled independent of said trusted host identifier by a first designer; and
a second hardware sub-design that is enabled based on said trusted host identifier by a second designer.
65. The integrated circuit as in claim 64, wherein said second hardware sub-design is enabled based on a group of trusted host identifiers.
66. The integrated circuit as in claim 64, wherein a license verification sub-design within said second hardware sub-design is encrypted to prevent modification.

67. The integrated circuit as in claim 64, wherein said trusted host identifier is a unique value to said integrated circuit.
68. The integrated circuit as in claim 64, wherein said integrated circuit is a field programmable gate array or an application specific integrated circuit.
69. A location within an integrated circuit having a trusted host identifier, comprising:
 - a licensing block that enables an encrypted circuit design inside said integrated circuit; and
 - a clock circuit that determines how long said encrypted circuit design will operate.
70. The location as in claim 69, wherein said licensing block enables at least a portion of said encrypted circuit design based on a group of trusted host identifiers.
71. The location as in claim 69, wherein said licensing block determines how long said encrypted circuit will operate.
72. The location as in claim 69, wherein said trusted host identifier is a unique value to said integrated circuit.
73. The location as in claim 69, wherein said integrated circuit is a field programmable gate array or an application specific integrated circuit.
74. A digital processing system for use in designing a plurality of integrated circuits (ICs), said digital processing system comprising:
 - a display device;
 - a memory;
 - a processor coupled to said memory and to said display device, said processor to generate a trusted host identifier specific license that

enables a hardware sub-design to operate for a specific amount of time on an integrated circuit having a trusted host identifier.

75. The digital processing system as in claim 74, wherein said processor to upload said hardware sub-design to said integrated circuit having said trusted host identifier along with another hardware sub-design that does not require a license.
76. The digital processing system as in claim 74, wherein said trusted host identifier specific license is based on a group of trusted host identifiers.
77. The digital processing system as in claim 74, wherein said trusted host identifier specific license determines how long said hardware sub-design will operate.
78. The digital processing system as in claim 74, wherein said trusted host identifier is a unique value to said integrated circuit.
79. The digital processing system as in claim 74, wherein said integrated circuit is a field programmable gate array or an application specific integrated circuit.
80. A method at a designer, comprising:
incorporating a first hardware design and a second encrypted design which enables a second hardware design through a trusted host identifier.
81. The method as in claim 80, wherein said second encrypted design is enabled through a trusted host identifier specific license that is based on a group of trusted host identifiers.
82. The method as in claim 81, wherein said trusted host identifier specific license determines how long said encrypted circuit design will operate.
83. The method as in claim 80, wherein said trusted host identifier is a unique value to said integrated circuit.

84. The method as in claim 80, wherein said first hardware design and said second encrypted design is incorporated within a field programmable gate array or an application specific integrated circuit.
85. A method at an IP block owner, comprising:
 - generating an encrypted design for incorporation into an integrated circuit by another designer; and
 - authorizing data for a trusted host identifier portion of said integrated circuit.
86. The method as in claim 85, wherein said encrypted design is enabled through a trusted host identifier specific license that is based on a group of trusted host identifiers.
87. The method as in claim 86, wherein said trusted host identifier specific license determines how long said encrypted design will operate.
88. The method as in claim 85, wherein said integrated circuit includes a trusted host identifier that is a unique value to said integrated circuit.
89. The method as in claim 85, wherein said integrated circuit is a field programmable gate array or an application specific integrated circuit.
90. A method at a trusted host identifier provider, comprising:
 - producing a programmable gate array or an application specific integrated circuit which can be configured by a designer; and
 - incorporating a trusted host identifier portion in said programmable gate array or said application specific integrated circuit.
91. The method as in claim 90, wherein said trusted host identifier portion includes an encrypted license verification sub-design within a hardware sub-design.

92. The method as in claim 90, wherein said trusted host identifier portion is enabled through a trusted host identifier specific license that is based on a group of trusted host identifiers.
93. The method as in claim 92, wherein said trusted host identifier specific license determines how long said encrypted design will operate.
94. The method as in claim 90, wherein said programmable gate array or said application specific integrated circuit includes a trusted host identifier that is a unique value to said programmable gate array or said application specific integrated circuit.
95. A method at a designer, comprising:
generating a trusted host identifier used by an encrypted portion of an IP block to enable a remaining portion of said IP block, where said trusted host identifier is provided by software at the time of programming or manufacturing an integrated circuit.
96. The method as in claim 95, wherein said encrypted portion is enabled through a trusted host identifier specific license that is based on a group of trusted host identifiers.
97. The method as in claim 96, wherein said trusted host identifier specific license determines how long said encrypted circuit design will operate.
98. The method as in claim 95, wherein said trusted host identifier is a unique value for said integrated circuit.
99. The method as in claim 95, wherein said integrated circuit is a field programmable gate array or an application specific integrated circuit.
100. A method at an IP block owner, comprising:
generating an encrypted circuit design associated with an IP block for incorporation into an integrated circuit by another designer; and

providing a trusted host identifier to said another designer through software at the time of programming or manufacturing an integrated circuit to enable said IP block.

101. The method as in claim 100, wherein said encrypted circuit design is enabled through a trusted host identifier specific license that is based on a group of trusted host identifiers.
102. The method as in claim 101, wherein said trusted host identifier specific license determines how long said encrypted circuit design will operate.
103. The method as in claim 100, wherein said trusted host identifier is a unique value to said integrated circuit.
104. The method as in claim 100, wherein said integrated circuit is a field programmable gate array or an application specific integrated circuit.

1/12

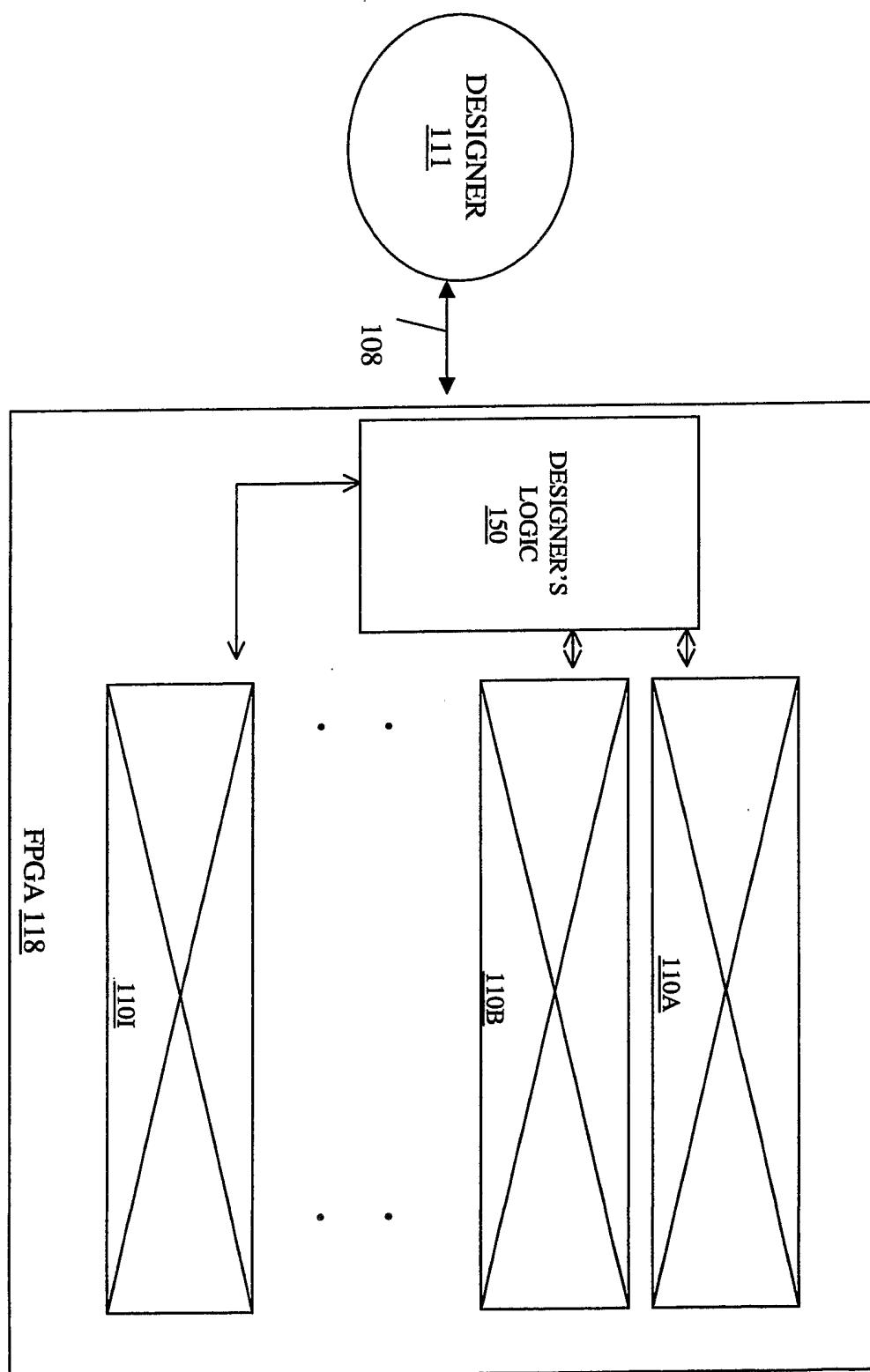


FIGURE 1A (PRIOR ART)

2/12

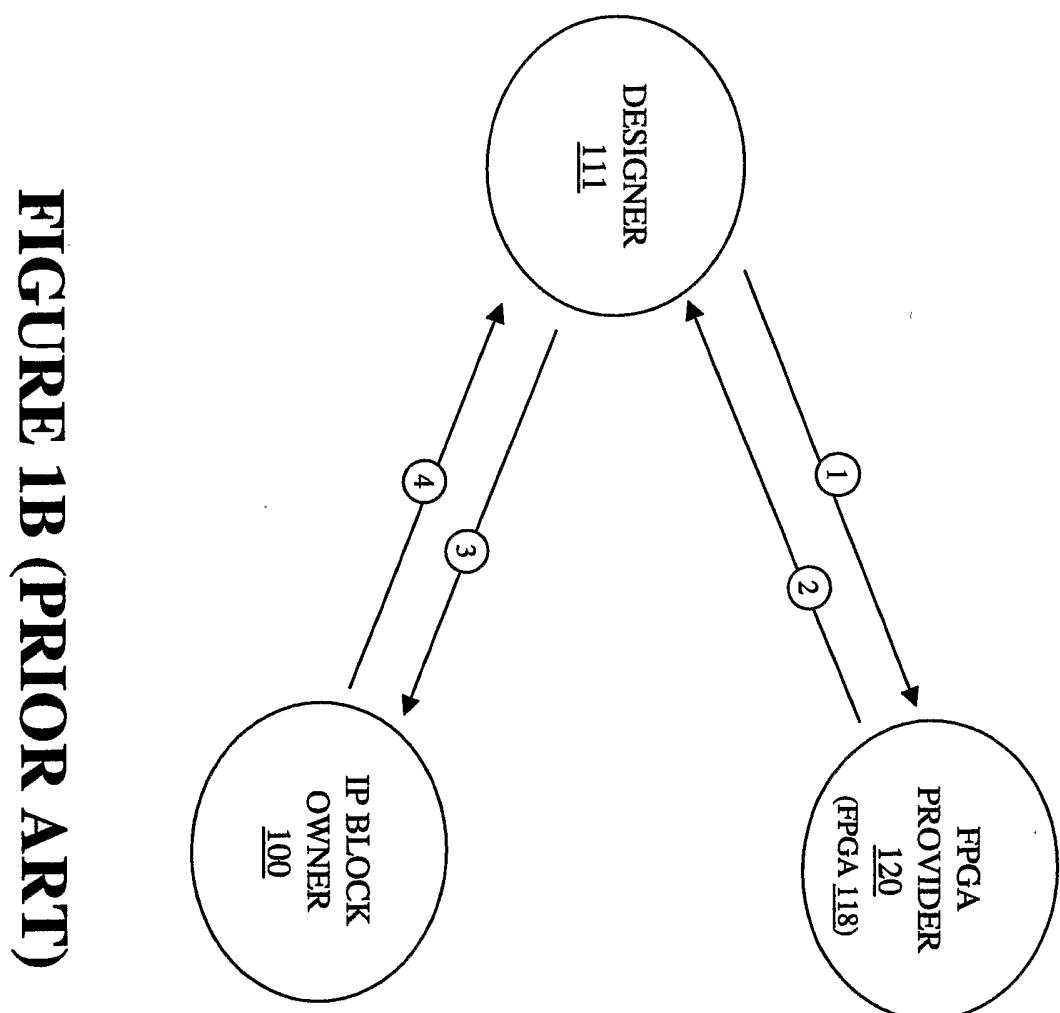
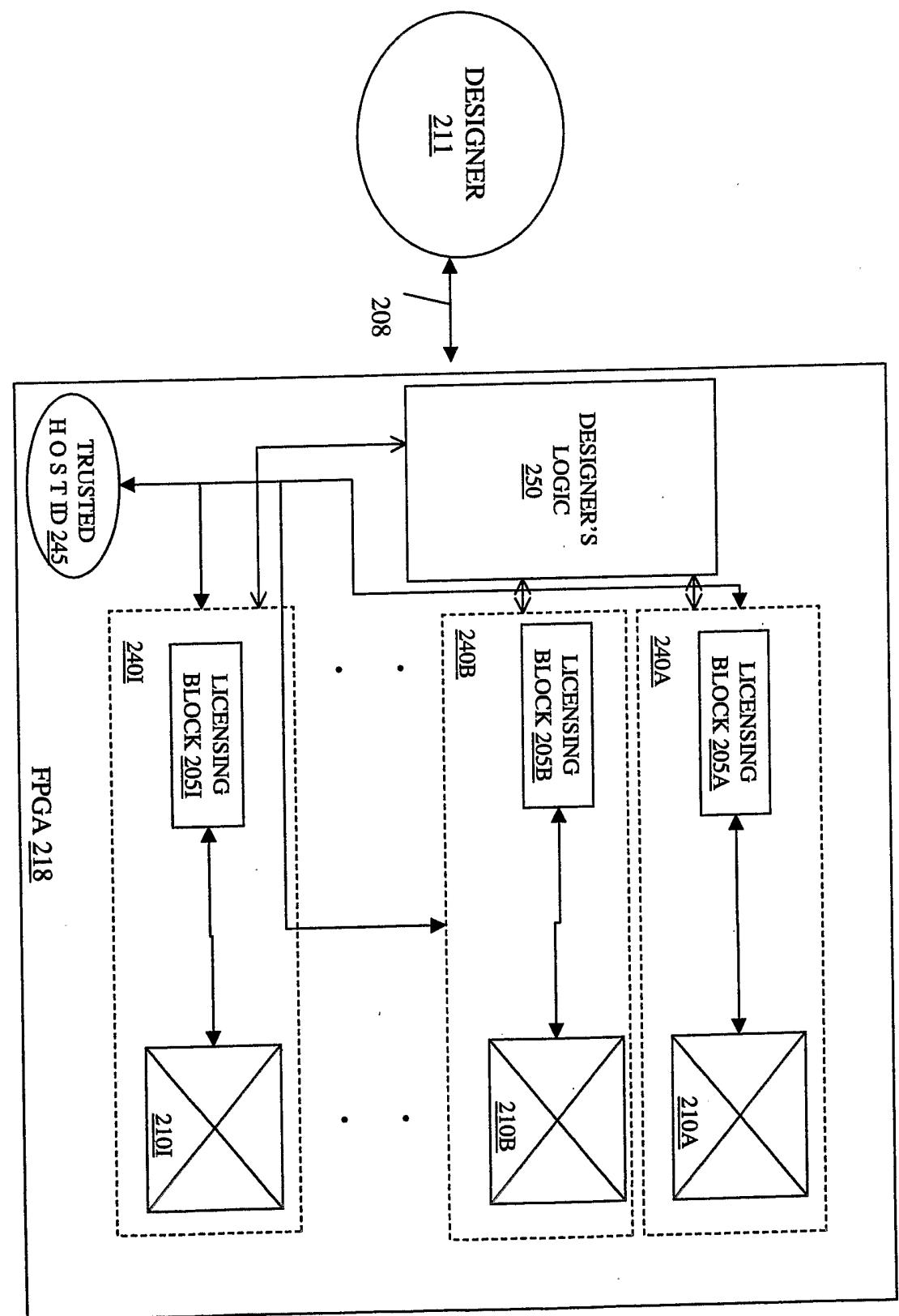


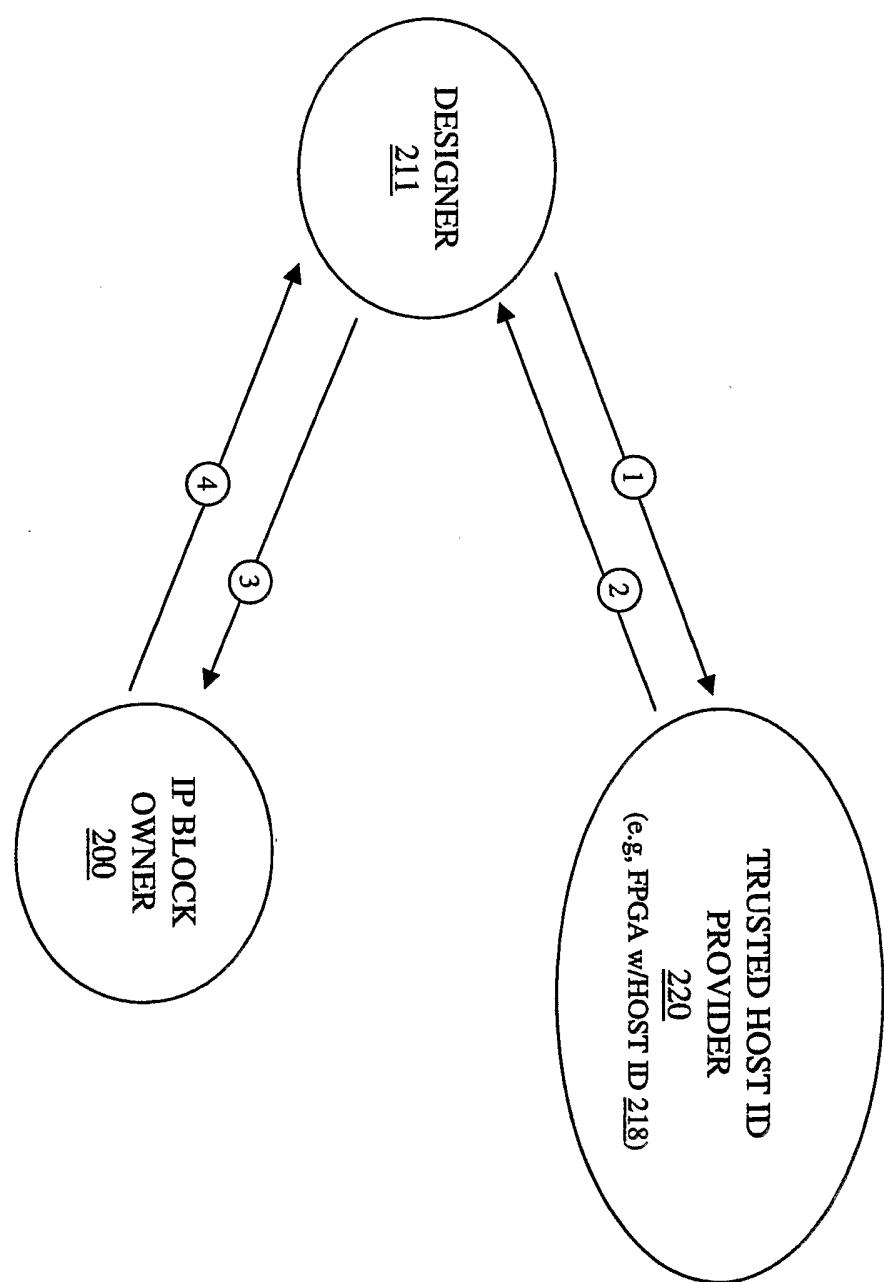
FIGURE 1B (PRIOR ART)

3/12

**FIGURE 2A**

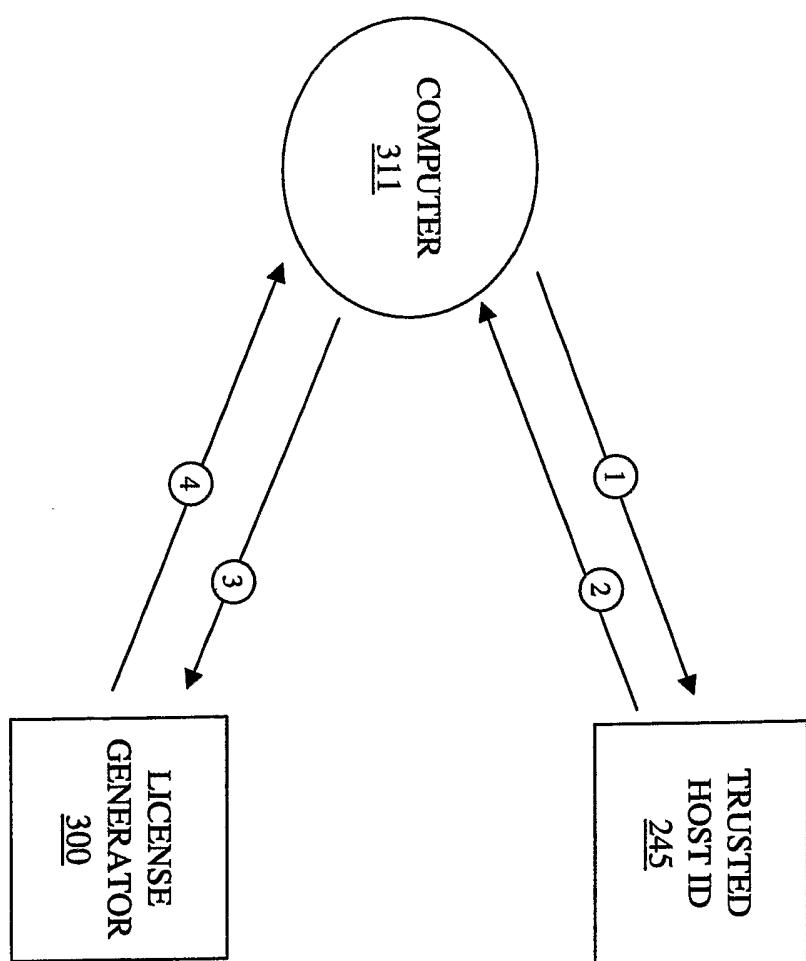
4/12

FIGURE 2B

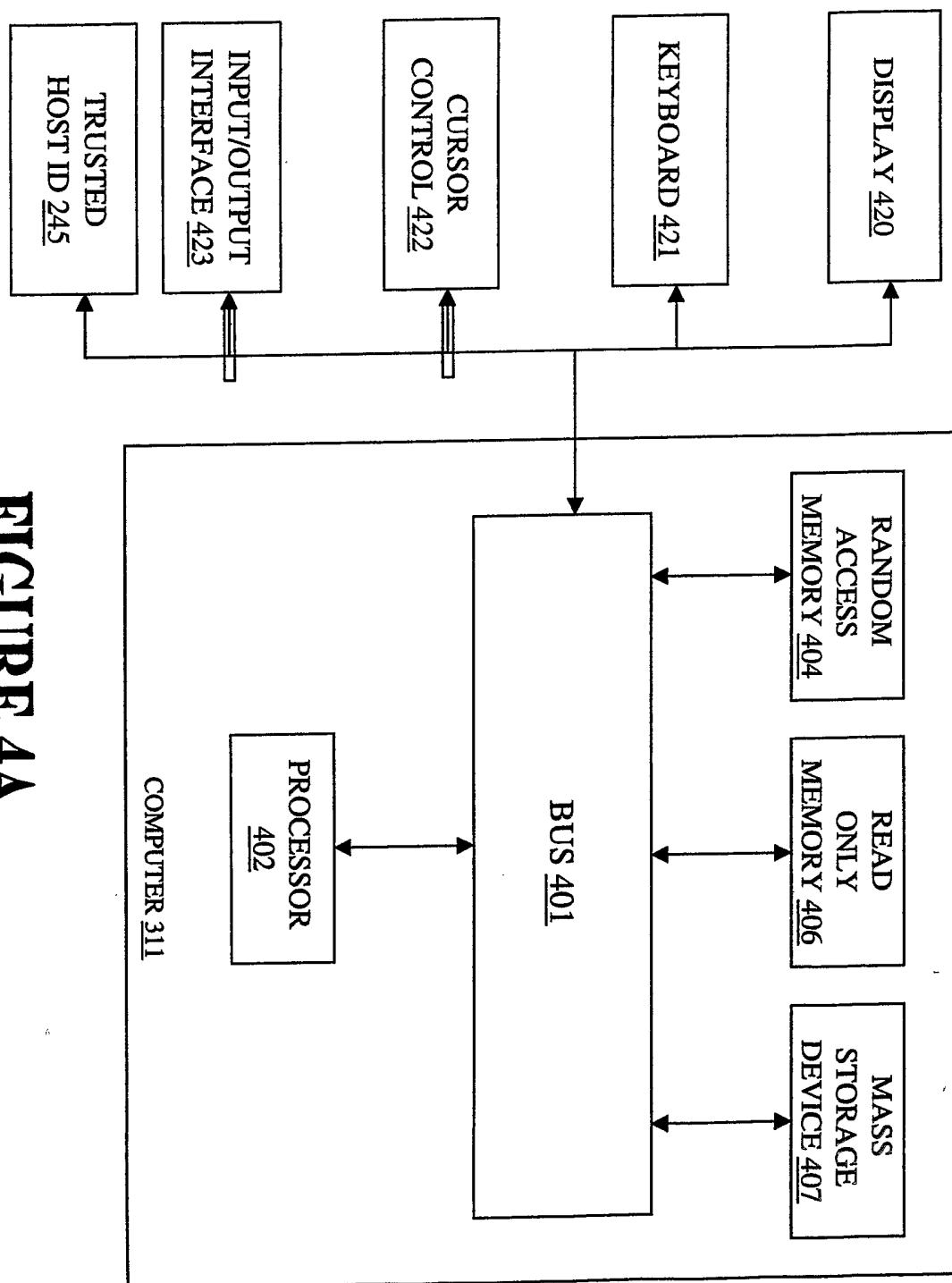


5/12

FIGURE 3



6/12

**FIGURE 4A**

7/12

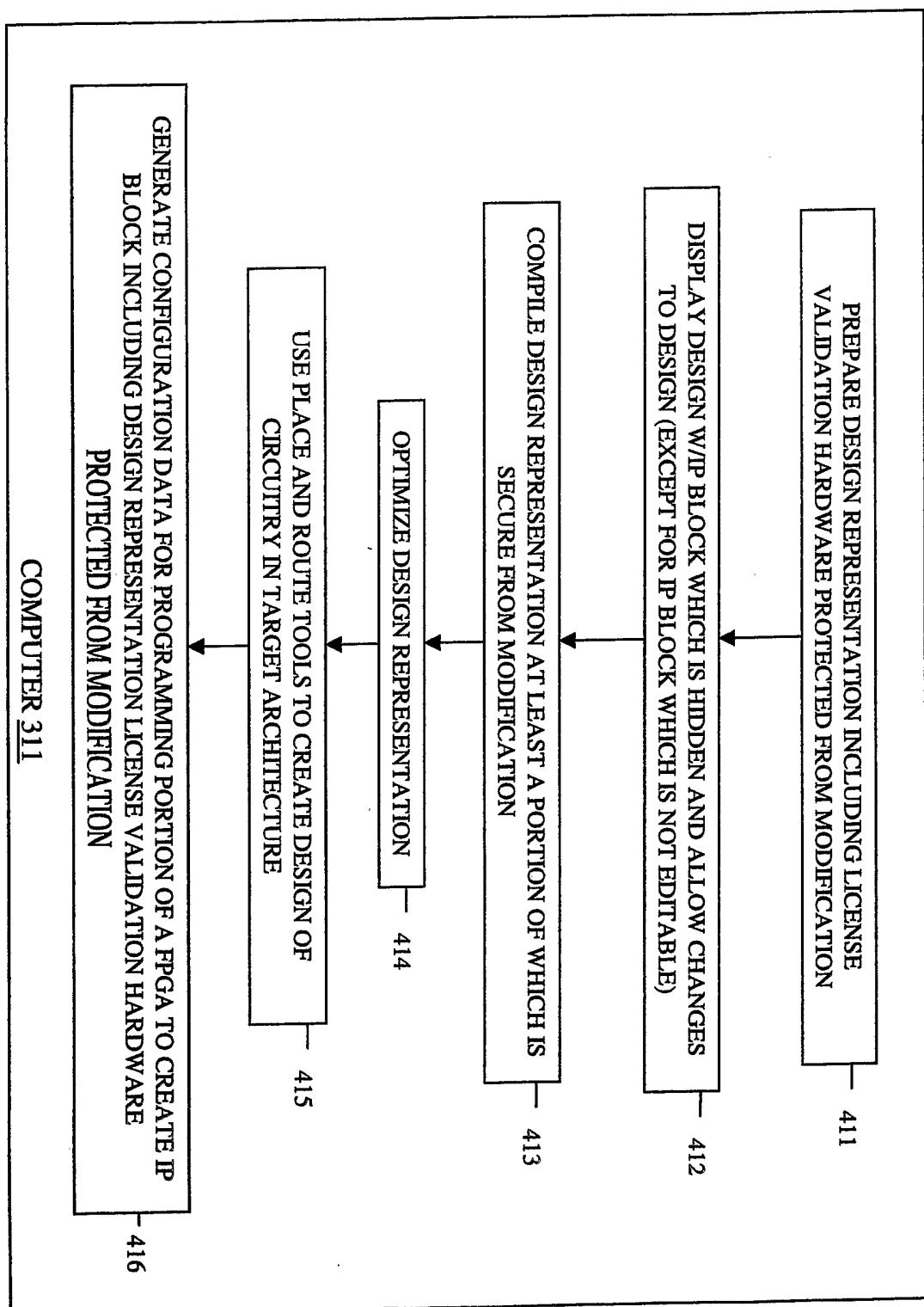
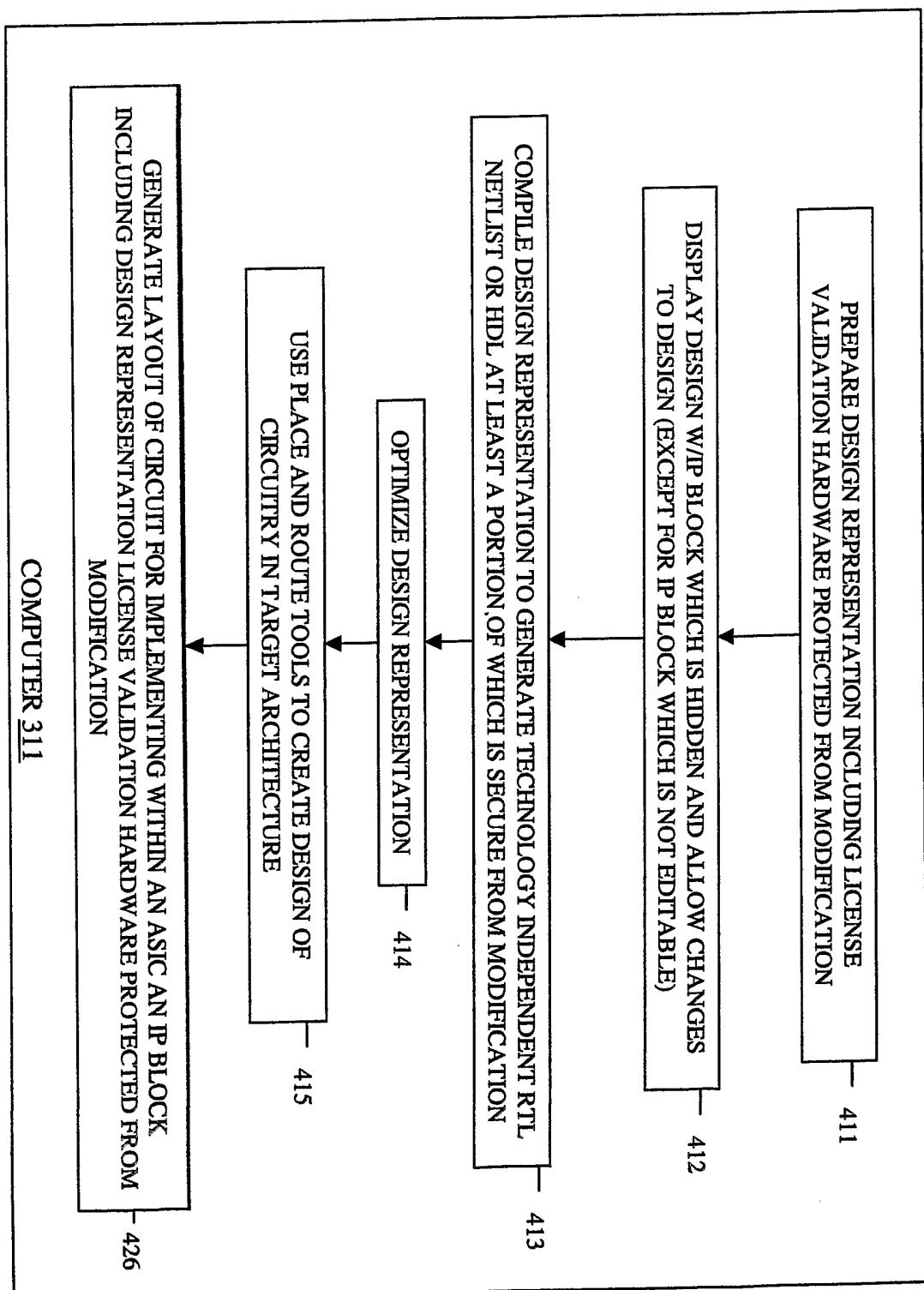


FIGURE 4B

8/12

**FIGURE 4C**

9/12

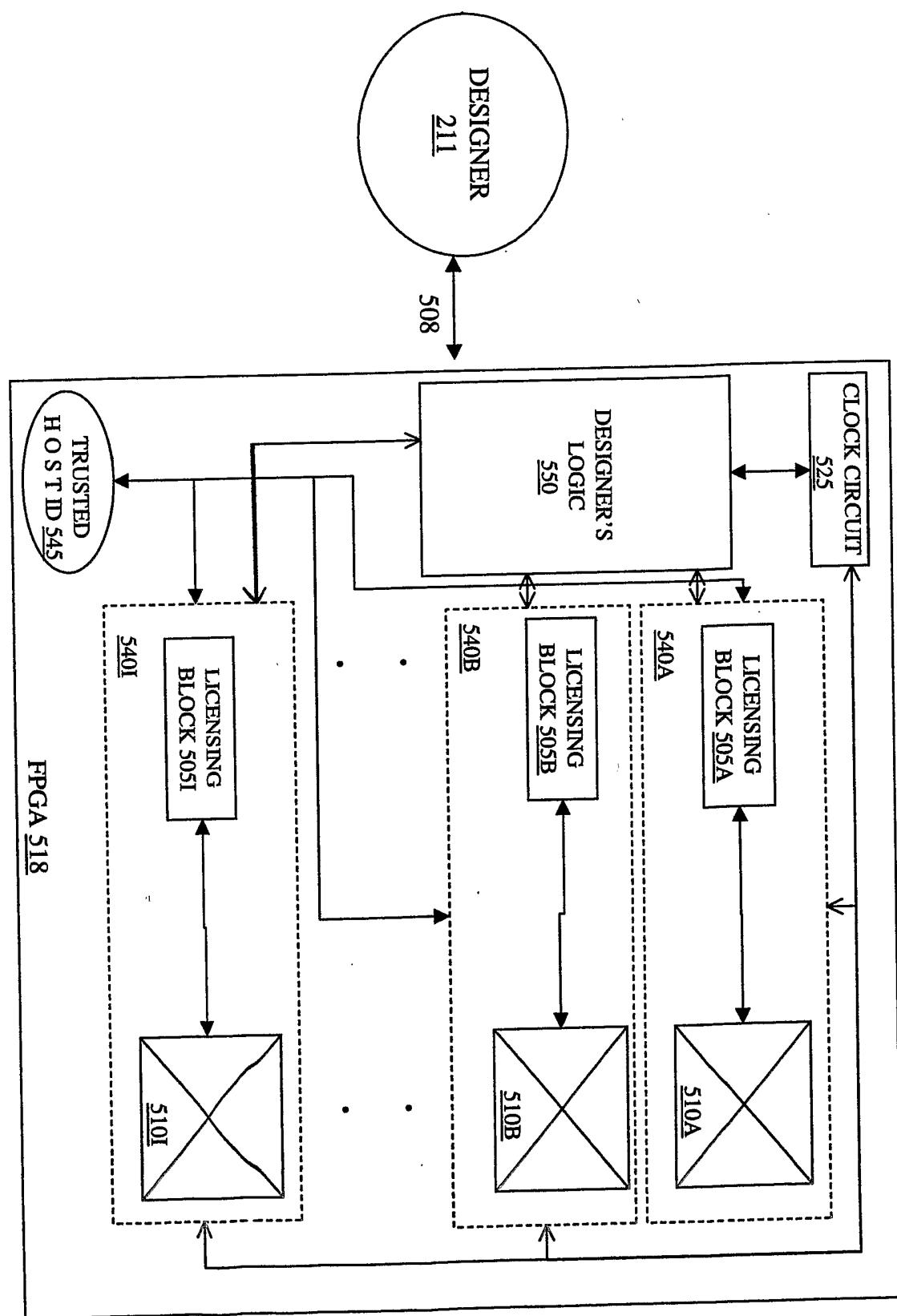
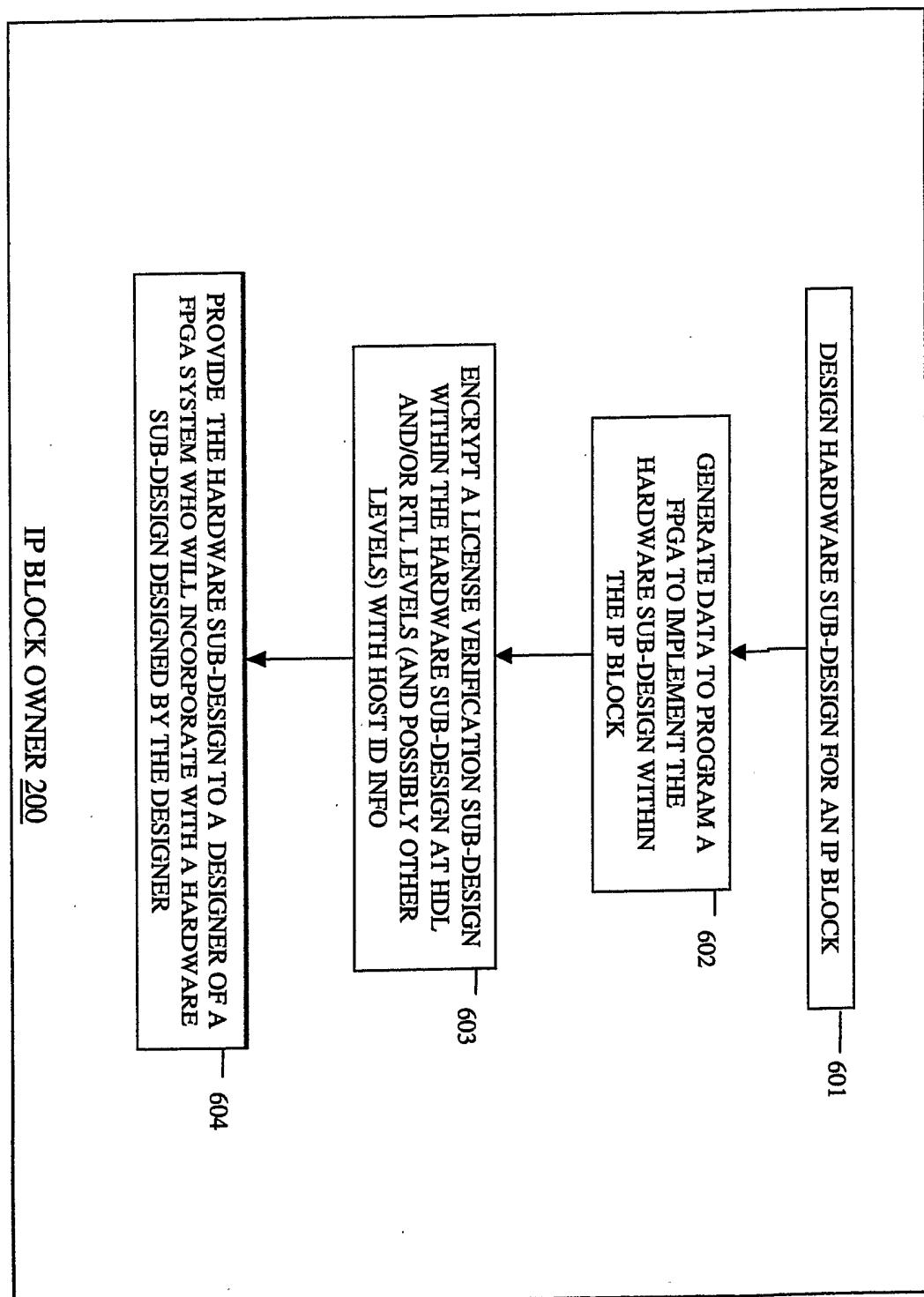
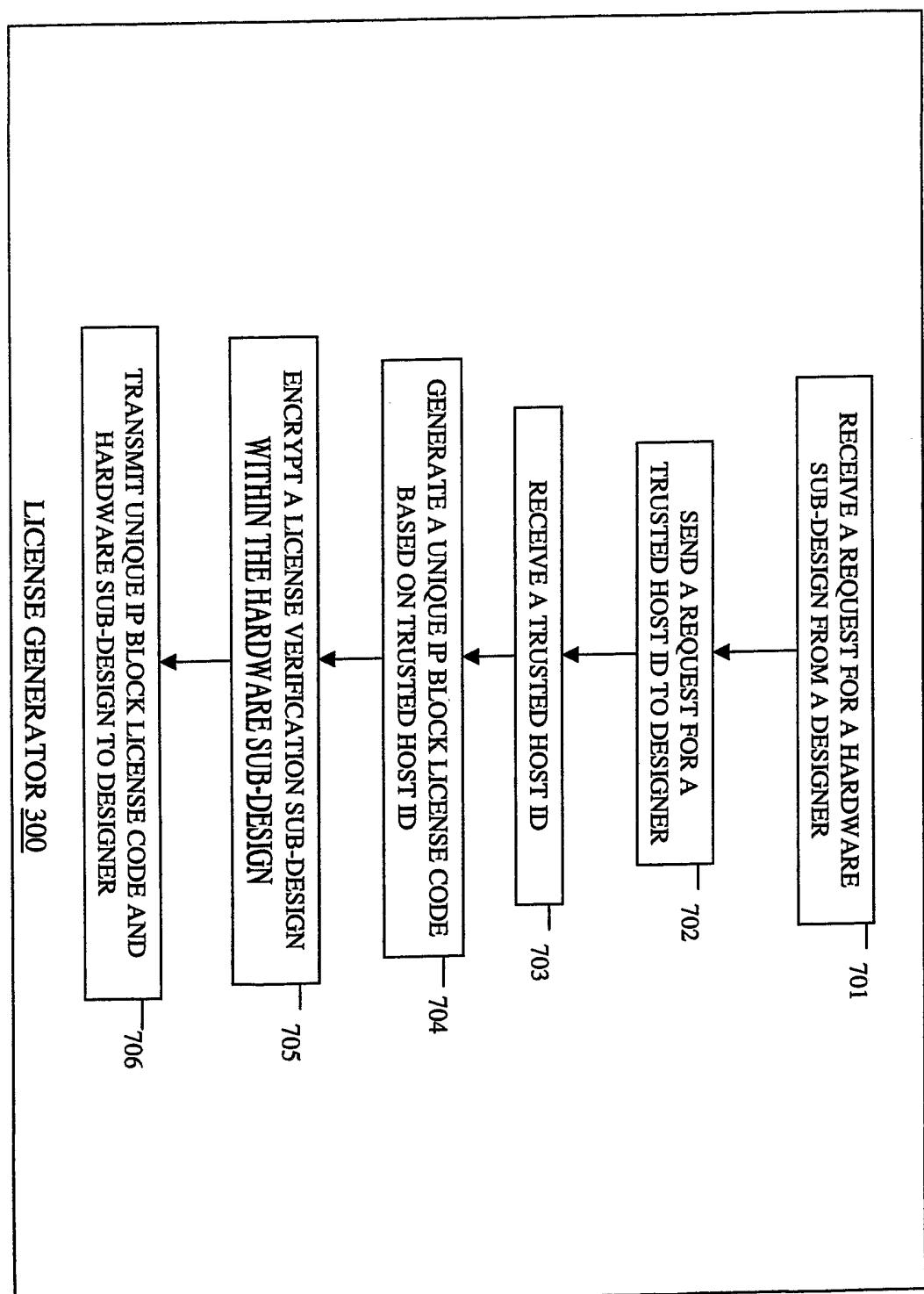


FIGURE 5

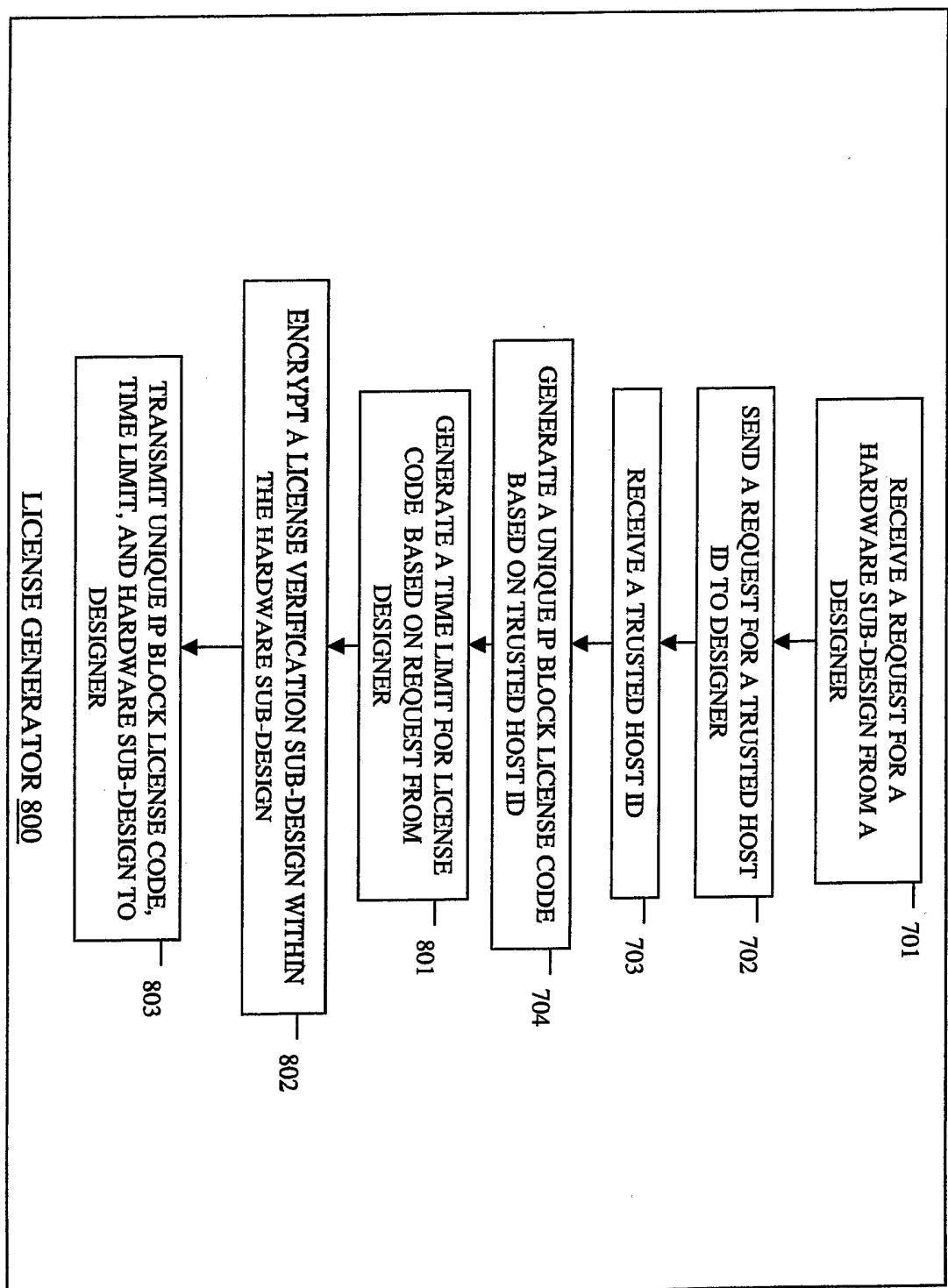
10/12

**FIGURE 6**

11/12

**FIGURE 7**

12/12

**FIGURE 8**

INTERNATIONAL SEARCH REPORT

International Application No

/US2005/034637

A. CLASSIFICATION OF SUBJECT MATTER
G06F17/50

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

EPO-Internal

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 6 357 037 B1 (BURNHAM JAMES L ET AL). 12 March 2002 (2002-03-12) column 1, line 30 – line 33 column 2, line 26 – column 4, line 19 ----- US 2003/126450 A1 (MASTER PAUL L ET AL) 3 July 2003 (2003-07-03) paragraph '0003! paragraph '0010! – paragraph '0015! ----- US 2003/140255 A1 (RICCHETTI MICHAEL ET AL) 24 July 2003 (2003-07-24) paragraph '0003! – paragraph '0016! ----- US 2002/199110 A1 (KEAN THOMAS A) 26 December 2002 (2002-12-26) paragraph '0001! paragraph '0005! – paragraph '0016! paragraph '0059! – paragraph '0079! -----	1-104 1-104 1-104 1-104



Further documents are listed in the continuation of box C.



Patent family members are listed in annex.

* Special categories of cited documents :

- *A* document defining the general state of the art which is not considered to be of particular relevance
- *E* earlier document but published on or after the international filing date
- *L* document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
- *O* document referring to an oral disclosure, use, exhibition or other means
- *P* document published prior to the international filing date but later than the priority date claimed

T later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

X document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

Y document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

& document member of the same patent family

Date of the actual completion of the international search

17 January 2006

Date of mailing of the international search report

30/01/2006

Name and mailing address of the ISA

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040, Tx. 31 651 epo nl,
Fax: (+31-70) 340-3016

Authorized officer

Anticoli, C

INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No

/US2005/034637

Patent document cited in search report	Publication date	Patent family member(s)			Publication date
US 6357037	B1	12-03-2002			NONE
US 2003126450	A1	03-07-2003	AU 2002361772 A1	WO 03058390 A2	24-07-2003 17-07-2003
US 2003140255	A1	24-07-2003			NONE
US 2002199110	A1	26-12-2002			NONE