

CLAIMS

What is claimed is:

1. A system for encrypting a high performance computing (HPC) application comprising:
 - an application code compiled into an executable file targeting a heterogenous computing environment, wherein the executable file runs on at least one host processor;
 - at least one FPGA device with designs compiled into associated FPGA netlists, wherein the netlists are targeted to the FPGA device;
 - a unique device identifier, wherein the unique device identifier is a manufacturer Chip ID associated with the FPGA device;
 - a License Manager utility, wherein the License Manager utility is provided via a Licenser to a Licensee to read the unique device identifier from the FPGA device;
 - an AES encryption algorithm using a Cyclic Block Chaining (CBC) and an Initialization Vector (IV); and
 - a Board Support Package (BSP), wherein the unique device identifier is embedded within the Board Support Package and is accessible to the host processor on every execution via the Board Support Package.
2. The system of claim 1, wherein the Board Support Package is Open Computing Language (OpenCL) compliant.
3. The system of claim 1, wherein the AES algorithm uses a 256-bit key.

4. The system of claim 1, wherein the IV is 128-bits.
5. The system of claim 3, wherein the AES key is randomly generated, stored, and maintained by the Licensor.
6. The system of claim 4, wherein the IV is randomly generated, stored, and maintained by the Licensor.
7. The system of claim 3, wherein the AES key rolls over with every update of the application code.
8. The system of claim 4, wherein the IV rolls over with every update of the application code.
9. The system of claim 1, wherein the unique device identifier is 64 bits.
10. A method for encrypting a HPC application, the method comprising:
 - reading a unique device identifier, wherein the unique device identifier is a manufacturer Chip ID from a FPGA device read via a License Manager utility;
 - concatenating FPGA netlists, an executable code, and the unique device identifier into the HPC application via a Licensor; and
 - encrypting the HPC application via an AES Cyclic Block Chaining (CBC) algorithm and an Initialization Vector (IV).

11. The method of claim 10, wherein an application programming interface (API) is used for a host to read the unique device identifier embedded within a Board Support Package.
12. The method of claim 10, wherein the AES algorithm uses a 256-bit key, and wherein the AES key is hard coded into the License Manager utility to encrypt.
13. The method of claim 10, wherein the AES algorithm uses a 128-bit IV, and wherein the IV is hard coded into the License Manager utility to encrypt.
14. The method of claim 10, wherein the AES algorithm uses a 256-bit key obtained by the License Manager utility from the Licensor via a web interface with a secure communication protocol.
15. The method of claim 10, wherein the AES algorithm uses a 128-bit IV obtained by the License Manager utility from the Licensor via a web interface with a secure communication protocol.
16. The method of claim 14, wherein the AES Key is randomly generated, stored, and maintained by the Licensor.
17. The method of claim 15, wherein the IV is randomly generated, stored, maintained by the Licensor.
18. The method of claim 14, wherein the AES Key rolls over with every update of the executable code.

19. The method of claim 15 wherein the IV rolls over with every update of the executable code.
20. The method of claim 13, wherein the unique device identifier is 64 bits.
21. The method of claim 10, wherein the unique device identifier, the FPGA netlists, and the executable code are concatenated in an arranged sequence of the unique device identifier, the FPGA netlists, and the executable code, and wherein the arranged sequence is encrypted as an executable file.
22. A system for decrypting a HPC application comprising:
 - an application code compiled into an executable file targeting a heterogeneous computing environment, wherein the executable file runs on at least one host processor;
 - at least one FPGA device with a design compiled into associated FPGA netlists, wherein the netlists are targeted to the FPGA device;
 - a unique device identifier, wherein the unique device identifier is a manufacturer Chip ID associated with the FPGA device;
 - a License Manager utility, wherein the License Manager utility is provided via a Licensor to a Licensee to read the unique device identifier from the FPGA device;
 - an AES encryption algorithm using a Cyclic Block Chaining (CBC) and an Initialization Vector (IV); and

a Board Support Package (BSP), wherein the unique device identifier is embedded within the Board Support Package and is accessible to the host processor on every execution via the Board Support Package.

23. The system of claim 22, wherein the Board Support Package is OpenCL compliant.
24. The system of claim 22, wherein the AES algorithm uses a 256-bit key, and wherein the key is hard coded into the License Manager utility to encrypt.
25. The system of claim 22, wherein the AES algorithm uses a 256-bit key obtained via the License Manager utility from the Licensor via a web interface with a secure communication protocol.
26. The system of claim 22, wherein the AES algorithm uses a 128-bit IV obtained by the License Manager utility from the Licensor via a web interface with a secure communication protocol.
27. The system of claim 22, wherein the AES algorithm uses a 128-bit IV, and wherein the IV is hard coded into the License Manager utility to encrypt.
28. The system of claim 24, wherein the AES Key is randomly generated, stored, and maintained by the Licensor.
29. The system of claim 26, wherein the IV is randomly generated, stored, and maintained by the Licensor.

30. The system of claim 24, wherein the AES Key rolls over with every update of the application code.
31. The system of claim 26, wherein the IV rolls over with every update of the application code.
32. The system of claim 22, wherein the unique device identifier is 64 bits.
33. A method for decrypting a HPC application comprising:
 - reading a first unique device identifier embedded within a BSP via a License Manager utility that is launched by a Licensee, wherein the first unique device identifier is a manufacturer Chip ID from a FPGA device;
 - decrypting a second unique device identifier from the HPC application via the License Manager utility with a static AES key and an IV; and
 - comparing the first unique device identifier against the second unique device identifier via the License Manager utility.
34. The method of claim 33, wherein a positive match of the first and the second unique device identifier proceeds with decrypting the remainder of the HPC application, outputting a decrypted executable file of a Host code netlist and a Kernel Code netlist, and launching the executable file.
35. The method of claim 33, wherein a negative match of the first and the second unique device identifier terminates a decryption process.