CLAIMS

1. A system for providing a secure boot image loading, comprising:

a microcontroller;

a plurality of physically modifiable internal components (PMIC) disposed on the microcontroller, wherein each of the plurality of PMICs can only be modified one time;

an image loader configured to load a boot image from memory of the microcontroller;

a checksum calculator configured to calculate a checksum value for the boot image; and

a checksum burner configured to modify the plurality of PMICs to create a binary representation of the checksum value for the boot image; and

wherein the image loader is configured to call the checksum calculator to calculate a second checksum value for the boot image each time the boot image is loaded, and to compare the second checksum value with the checksum value in the plurality of PMICs to determine if the second checksum value and the checksum value are the same.

2. The system of claim 1 wherein the second checksum value and the checksum value are the same, wherein the image loader is configured to continue to allow the boot image to load.

3. The system of claim 1 wherein the second checksum value and the first checksum value are not the same, wherein the image loader is configured to execute a mismatch response.

4. The system of claim 3 wherein the mismatch response is burned into the plurality of PMICs.

5. The system of claim 4 wherein the mismatch response prevents further loading of the boot image.

6. The system of claim 4 wherein the mismatch response alerts a user of the mismatch.

7. The system of claim 6 wherein the mismatch response further prevents loading of the boot image until a response is received from the user.

8. The system of claim 4 wherein the mismatch response that is burned into the plurality of PMICs is a partial response.

9. The system of claim 8 wherein the partial response is a pointer to an address where a complete mismatch response is stored.

10. A method comprising:

calculating a checksum value for a boot image;

writing the checksum value to a microcontroller;

burning in the checksum value into the microcontroller; and

burning in responses to a checksum mismatch into the microcontroller.

11. The method of claim 10 wherein the checksum value is burned into a plurality of physically modifiable internal components (PMIC) disposed on the microcontroller, wherein each of the plurality of PMICs can only be modified one time.

12. The method of claim 11 wherein the checksum value is burned into the plurality of PMICs as a binary representation of the checksum value.

13. The method of claim 10 wherein the response to a checksum mismatch are burned into a plurality of physically modifiable internal components (PMIC) disposed on the microcontroller, wherein each of the plurality of PMICs can only be modified one time.

14. A method of securing a boot image load, comprising:

loading the boot image from a microcontroller;

calculating a checksum value for the boot image;

comparing the checksum value with a stored checksum value, wherein the stored checksum value is permanently burned into the microcontroller;

allowing the boot image to continue loading in response to a match when comparing the checksum value with the stored checksum value; and

executing a mismatch response in response to a mismatch when comparing the checksum value with the stored checksum value, wherein the mismatch response is permanently burned into the microcontroller.

15. The method of claim 14 wherein the mismatch response stops further loading of the boot image.

16. The method of claim 14 wherein the mismatch response alerts a use of the mismatch.

17. The method of claim 16 wherein the mismatch response further stops further loading of the boot image until a response is received from the user.

18. The method of claim 14 wherein the mismatch response is a partial response as a pointer to an address where a complete mismatch response is stored.