CLAIMS

1. A method of error checking for instructions, the method comprising:

creating an instruction packet with one or more instructions;

determining if a parity of the instruction packet matches a predesignated parity;

if the parity of the instruction packet does not match the predesignated parity, using a bit within the instruction packet to change the parity of the instruction packet to match the predesignated parity; and

storing the instruction packet with the predesignated parity in a memory.

2. The method of claim 1, further comprising determining if there is an error in the instruction packet retrieved from the memory, based on comparing the parity of the instruction packet to the predesignated parity.

3. The method of claim 2, further comprising determining there is an error in the instruction packet and generating an exception.

4. The method of claim 2, further comprising determining there is an error in the instruction packet and recording the error for analysis in a debugger.

5. The method of claim 2, comprising determining there is an error in the instruction packet during one or more pipeline stages including fetch, decode, one or more execution stages, or write back of the instruction packet in a processor.

6. The method of claim 1, wherein the instruction packet is a very long instruction word (VLIW) packet.

7. The method of claim 6, further comprising determining that the VLIW packet comprises less than a maximum possible number of instructions, inserting a no-operation (NOP) field in the VLIW packet, and using a NOP bit of the NOP field as the bit to change parity of the instruction packet.

8. The method of claim 6, wherein the bit belongs to a field of the VLIW packet, the field used for conveying information related to instructions of the VLIW packet.

9. The method of claim 1, wherein the bit is an unused bit of the instruction packet.

10. The method of claim 1, wherein the predesignated parity is one of an odd parity or an even parity.

11. An apparatus comprising:

a processor configured to:

create an instruction packet with one or more instructions;

determine if a parity of the instruction packet matches a predesignated parity; and

if the parity of the instruction packet does not match the predesignated parity, use a bit within the instruction packet to change the parity of the instruction packet to match the predesignated parity; and

a memory configured to store the instruction packet with the predesignated parity.

12. The apparatus of claim 11, wherein a parity check block of the processor is configured to determine if there is an error in the instruction packet retrieved from the memory, based on a comparison of the parity of the instruction packet to the predesignated parity.

13. The apparatus of claim 12, wherein the processor is further configured to generate an exception if an error is detected.

14. The apparatus of claim 12, wherein the processor is further configured to record the instruction packet for future analysis in a debugger if an error is detected.

15. The apparatus of claim 11, wherein the instruction packet is a very long instruction word (VLIW) packet.

16. The apparatus of claim 15, wherein an assembler is configured to determine that the VLIW packet comprises less than a maximum possible number of instructions, and insert a no-operation (NOP) field in the VLIW packet, wherein a NOP bit of the NOP field is used as the bit to match the predesignated parity.

17. The apparatus of claim 16, wherein the bit belongs to a field of the VLIW packet, the field used for conveying information related to instructions of the VLIW packet.

18. The apparatus of claim 11, wherein the bit is an unused bit of the instruction packet.

19. The apparatus of claim 11, wherein the predesignated parity is one of an odd parity or an even parity.

20. The apparatus of claim 11 integrated into a device selected from the group consisting of a set top box, music player, video player, entertainment unit, navigation device, communications device, personal digital assistant (PDA), fixed location data unit, and a computer.

21. An apparatus comprising:

means for creating an instruction packet with one or more instructions;

means for determining if a parity of the instruction packet matches a predesignated parity; and

if the parity of the instruction packet does not match the predesignated parity, means for using a bit within the instruction packet to change the parity of the instruction packet to match the predesignated parity; and

means for storing the instruction packet with the predesignated parity.

22. The apparatus of claim 21, further comprising means for determining if there is an error in the instruction packet retrieved from the means for storing, based on a comparison of the parity of the instruction packet to the predesignated parity.

23. The apparatus of claim 22, further comprising means for generating an exception if an error is detected.

24. The apparatus of claim 21, wherein the predesignated parity is one of an odd parity or an even parity.

25. A non-transitory computer readable storage medium comprising code, which, when executed by a processor, causes the processor to perform a method of error checking for instructions, the non-transitory computer readable storage medium comprising:

code for creating an instruction packet with one or more instructions;

code for determining if a parity of the instruction packet matches a predesignated parity;

if the parity of the instruction packet does not match the predesignated parity, code for using a bit within the instruction packet to change the parity of the instruction packet to match the predesignated parity; and

code for storing the instruction packet with the predesignated parity in a memory.

26. The non-transitory computer readable storage medium of claim 25, further comprising code for determining if there is an error in the instruction packet retrieved from the memory, based on comparing the parity of the instruction packet to the predesignated parity.

27. The non-transitory computer readable storage medium of claim 26, further comprising code for determining there is an error in the instruction packet and code for generating an exception.

28. The non-transitory computer readable storage medium of claim 26, further comprising code for determining there is an error in the instruction packet and code for recording the error for analysis in a debugger.

29. The non-transitory computer readable storage medium of claim 25, wherein the bit is an unused bit of the instruction packet.

30. The non-transitory computer readable storage medium of claim 25, wherein the predesignated parity is one of an odd parity or an even parity.