CLAIMS

1. A system comprising:

a plurality of machines comprising a first generation machine and a second generation machine, each of the plurality of machines comprising a machine version, the first generation machine executing a first virtual machine and a virtual architecture level, the second generation machine executing a second virtual machine and the virtual architecture level,

wherein the virtual architecture level provides a compatibility level for a complex interruptible instruction to the first and second virtual machines, the compatibility level comprising a local parameter block format for the complex interruptible instruction for each of the plurality of machines, the compatibility level being architected for indicating a lowest common denominator machine version across the plurality of machines;

wherein the lowest common denominator machine version is propagating from one of the machines of the plurality of machines having the lowest common denominator version, to the remaining machines of the plurality of machines;

wherein, based on the propagating, the remaining machines of the plurality of machines implement the complex interruptible instruction as an architecture specific instruction with a parameter block format corresponding to the propagated lowest common denominator machine version;

wherein the architecture specific instruction is executed on the remaining machines of the plurality of machines, wherein the execution of the architecture instruction causes the machine to determine the operation to be performed based on the parameter block size, wherein the operation is one of: query, generate, compress, and expand;

wherein upon completion of the execution, a condition code resulting from the execution is determined; and

wherein one of completion or further operation processing is performed based on the condition code.

2. The system of claim 1, wherein the lowest common denominator indicator is generated by a machine of the plurality of machines to first execute the complex interruptible instruction.

3. The system of claim 1, wherein the lowest common denominator indicator is propagated from a machine of the plurality of machines to first execute the complex interruptible instruction to a remaining number of the plurality of machines.

4. The system of claim 1, wherein the lowest common denominator indicator within the compatibility level is controlled through a series of facility bits that identify which functions are available in a particular virtual machine.

5. The system of claim 1, wherein the complex interruptible instruction comprises a DEFLATE Conversion Call instruction.

6. The system of claim 1, wherein the complex interruptible instruction comprises an instruction from a complex set of instructions running on an accelerator.

7. A method comprising:

executing a first virtual machine and a virtual architecture level by a first generation machine of a plurality of machines, the plurality of machines comprising the first generation machine and a second generation machine, each of the plurality of machines comprising a machine version;

executing a second virtual machine and the virtual architecture level by the second generation machine;

providing, by the virtual architecture level, a compatibility level for a complex interruptible instruction to the first and second virtual machines, the compatibility level comprises a local parameter block format for the complex interruptible instruction for each of the plurality of machines, the compatibility level being architected for indicating a lowest common denominator machine version across the plurality of machines;

propagating the lowest common denominator machine version, from one of the machines of the plurality of machines having the lowest common denominator version, to the remaining machines of the plurality of machines;

based on the propagating, implementing, by the remaining machines of the plurality of machines, the complex interruptible instruction as an architecture specific instruction with a parameter block format corresponding to the propagated lowest common denominator machine version;

executing the architecture specific instruction on the remaining machines of the plurality of machines, wherein the execution of the architecture instruction causes the machine to determine the operation to be performed based on the parameter block size, wherein the operation is one of: query, generate, compress, and expand;

upon completion of the execution, determining a condition code resulting from the execution; and

performing one of completion or further operation processing based on the condition code.

8. The method of claim 7, wherein the lowest common denominator indicator is generated by a machine of the plurality of machines to first execute the complex interruptible instruction.

9. The method of claim 7, wherein the lowest common denominator indicator is propagated from a machine of the plurality of machines to first execute the complex interruptible instruction to a remaining number of the plurality of machines.

10. The method of claim 7, wherein the lowest common denominator indicator within the compatibility level is controlled through a series of facility bits that identify which functions are available in a particular virtual machine.

11. The method of claim 7, wherein the complex interruptible instruction comprises a DEFLATE Conversion Call instruction.

12. The method of claim 7, wherein the complex interruptible instruction comprises an instruction from a complex set of instructions running on an accelerator.

13. A computer program product comprising a computer readable storage medium having program instructions embodied therewith, the program instructions executable to cause operations comprising:

executing a first virtual machine and a virtual architecture level by a first generation machine of a plurality of machines, the plurality of machines comprising a first generation machine and a second generation machine, each of the plurality of machines comprising a machine version;

executing a second virtual machine and the virtual architecture level by the second generation machine;

providing, by the virtual architecture level, a compatibility level for a complex interruptible instruction to the first and second virtual machines, the compatibility level comprises a local parameter block format for the complex interruptible instruction for each of the plurality of machines, the compatibility level being architected for indicating a lowest common denominator machine version across the plurality of machines;

propagating the lowest common denominator machine version, from one of the machines of the plurality of machines having the lowest common denominator version, to the remaining machines of the plurality of machines;

based on the propagating, implementing, by the remaining machines of the plurality of machines, the complex interruptible instruction as an architecture specific instruction with a parameter block format corresponding to the propagated lowest common denominator machine version;

executing the architecture specific instruction on the remaining machines of the plurality of machines, wherein the execution of the architecture instruction causes the machine to determine the operation to be performed based on the parameter block size, wherein the operation is one of: query, generate, compress, and expand;

upon completion of the execution, determining a condition code resulting from the execution; and

performing one of completion or further operation processing based on the condition code.

14. The computer program product of claim 13, wherein the lowest common denominator indicator is generated by a machine of the plurality of machines to first execute the complex interruptible instruction.

15. The computer program product of claim 13, wherein the lowest common denominator indicator is propagated from a machine of the plurality of machines to first execute the complex interruptible instruction to a remaining number of the plurality of machines.

16. The computer program product of claim 13, wherein the lowest common denominator indicator within the compatibility level is controlled through a series of facility bits that identify which functions are available in a particular virtual machine.

17. The computer program product of claim 13, wherein the complex interruptible instruction comprises a DEFLATE Conversion Call instruction.

18. The computer program product of claim 13, wherein the complex interruptible instruction comprises an instruction from a complex set of instructions running on an accelerator.