CLAIMS

What is claimed is:

1. A method of implementing a modified priority routing of an input/output (I/O) interruption, comprising:

determining, by a processor, whether the I/O interruption is pending for a core;

determining, by the processor, whether any of a plurality of guest threads of the core is enabled for guest thread processing of the interruption in response to determining that the PO interruption is pending;

determining, by the processor, whether at least one of the plurality of guest threads enabled for the guest thread processing is in a wait state,

wherein each guest thread comprises an interrupt subclass mask that includes a control bit, a status bit, and a zone mask table,

wherein the corresponding zone mask table comprises a plurality of enablement bits per supported zone,

wherein the wait state for each guest thread is indicated by the status bit and the control bit associated with each guest thread,

wherein the status and control bits are configured to enable the associated guest thread for the guest thread processing of the interruption;

initiating a delay for routing the I/O interruption to a host based upon determining that none of the plurality of guest threads are enabled for the guest thread processing to prevent the core from leaving an emulation mode;

based on determining that more than one of the plurality of guest threads enabled for the guest thread processing are in the wait state, selecting via a routing algorithm an enabled guest thread from the more than one of the plurality of guest threads enabled for the guest thread processing; and

routing, by the processor, the I/O interruption to the enabled guest thread,

wherein the I/O interruption is pending for a particular interrupt subclass within a particular zone of the core when the guest program-status word bit is on and an interrupt subclass is on in a corresponding zone mask table,

wherein each of the plurality of enablement bits corresponds directly to the bits in a pending interruption array of the core that is simultaneously enabled in both the enabled guest thread and the host.

2. The method of claim 1, wherein the I/O interruption is indicated as pending in a pending interrupt array.

3. The method of claim 1, further comprising: based upon determining that none of the plurality of guest threads enabled for the guest thread processing are in the wait state, routing the I/O interruption to one of the plurality of guest threads enabled for the guest thread processing.

4. The method of claim 1, further comprising: after the initiating of the delay, determining whether any of the plurality of guest threads of the core are enabled for the guest thread processing.

5. A system for implementing a modified priority routing of an input/output (I/O) interruption, comprising a processor and a memory:

the processor being configured to:

determine whether the I/O interruption is pending for a core;

determine whether any of a plurality of guest threads of the core is enabled for guest thread processing of the interruption based upon determining that the I/O interruption is pending;

determine whether at least one of the plurality of guest threads enabled for the guest thread processing is in a wait state,

wherein each guest thread comprises an interrupt subclass mask that includes a control bit, a status bit, and a zone mask table,

wherein the corresponding zone mask table comprises a plurality of enablement bits per supported zone,

wherein the wait state for each guest thread is indicated by the status bit and the control bit associated with each guest thread,

wherein the status and control bits are configured to enable the associated guest thread for the guest thread processing of the interruption;

initiating a delay for routing the I/O interruption to a host based upon determining that none of the plurality of guest threads are enabled for the guest thread processing to prevent the core from leaving an emulation mode;

based on determining that more than one of the plurality of guest threads enabled for the guest thread processing are in the wait state, selecting via a routing algorithm an enabled guest thread from the more than one of the plurality of guest threads enabled for the guest thread processing; and

route the I/O interruption to the enabled guest thread,

wherein the I/O interruption is pending for a particular interrupt subclass within a particular zone of the core when the guest program-status word bit is on and an interrupt subclass is on in a corresponding zone mask table,

wherein each of the plurality of enablement bits corresponds directly to the bits in a pending interruption array of the core that is simultaneously enabled in both the enabled guest thread and the host.

6. The system of claim 5, wherein the I/O interruption is indicated as pending in a pending interrupt array.

7. The processor of claim 5, further configured to: based upon determining that none of the plurality of guest threads enabled for guest thread processing are in the wait state, route the I/O interruption to one of the plurality of guest threads enabled for the guest thread processing.

8. The processor of claim 5, further configured to: after the initiation of the delay, determine whether any of the plurality of guest threads of the core are enabled for the guest thread processing.

9. A computer program product, the computer program product comprising a non-transitory computer readable storage medium having program instructions implementing a modified priority routing of an input/output (I/O) interruption embodied therewith, the program instructions executable by a processor to cause the processor to:

determine whether the I/O interruption is pending for a core;

determine whether any of a plurality of guest threads of the core is enabled for guest thread processing of the interruption based upon determining that the I/O interruption is pending;

determine whether at least one of the plurality of guest threads enabled for the guest thread processing is in a wait state,

wherein each guest thread comprises an interrupt subclass mask that includes a control bit, a status bit, and a zone mask table,

wherein the corresponding zone mask table comprises a plurality of enablement bits per supported zone,

wherein the wait state for each guest thread is indicated by the status bit and the control bit associated with each guest thread,

wherein the status and control bits are configured to enable the associated guest thread for the guest thread processing of the interruption;

initiating a delay for routing the I/O interruption to a host based upon determining that none of the plurality of guest threads are enabled for the guest thread processing to prevent the core from leaving an emulation mode;

based on determining that more than one of the plurality of guest threads enabled for the guest thread processing are in the wait state, selecting via a routing algorithm an enabled guest thread from the more than one of the plurality of guest threads enabled for the guest thread processing; and

route the I/O interruption to the enabled guest thread,

wherein the I/O interruption is pending for a particular interrupt subclass within a particular zone of the core when the guest program-status word bit is on and an interrupt subclass is on in a corresponding zone mask table,

wherein each of the plurality of enablement bits corresponds directly to the bits in a pending interruption array of the core that is simultaneously enabled in both the enabled guest thread and the host.

10. The computer program product of claim 9, wherein the I/O interruption is indicated as pending in a pending interrupt array.

11. The computer program product of claim 9, wherein the program instructions are further executable by the processor to cause the processor to: based upon determining that none of the plurality of guest threads enabled for guest thread processing are in the wait state, route the I/O interruption to one of the plurality of guest threads enabled for the guest thread processing.

12. The computer program product of claim 9, wherein the program instructions are further executable by the processor to cause the processor to: initiate a delay for a routing of the I/O interruption to a host based upon determining that none of the plurality of guest threads are enabled for the guest thread processing.

13. The computer program product of claim 9, wherein the program instructions are further executable by the processor to cause the processor to: after the initiation of the delay, determine whether any of the plurality of guest threads of the core are enabled for the guest thread processing.

14. The system of claim 5, wherein the control bit comprises a guest control register, and the status bit comprises a guest program-status word bit.