# SimNow™ Software version 4.6.2 - Release Notes

## 4.6.2 Enhancements/Bug Fixes

- Initial support for AMD SB900 Southbridge in the NDA package.
- Fixed an issue with Firmware hub strapping for SB800/SB900 models
- Add additional SST LPC part support to the Memory device.
- Users of the Family12h BIOS PI 0.0.5.2 will want to use the following BSDs to workaround current simulator issues:
    - barb_family12h_0.0.5.2.bsd: Single-core BSD with memory timing workarounds.
    - seabream_family12h_0.0.5.2.bsd: Contains memory timing workarounds
    - torpedo_family12h_0.0.5.2.bsd: Contains memory timing workarounds to allow the BIOS to fully POST. However there is no video output due to simulator issues.

## FAQ for AVX instruction support in SimNow

1) *Which BSDs support AVX?*

Please use vp_bd_phase1.bsd.

2) *Which instruction sets are enabled in vp_bd_phase1.bsd, beyond those already supported by Family10h models?*

SSSE3, SSE4.1, SSE4.2, AES (and PCLMULQDQ), XSAVE/XRSTOR, AVX, XOP and CVT16 are supported.

3) *When I run my AVX application in SimNow, the application crashes with undefined opcode exceptions. What is the problem?*

Most operating systems do not yet support YMM state management using XSAVE. The OS is expected to set 'XFEATURE_ENABLED_MASK[2:1]=11b' and 'CR4.OSXAVE=1'. When these bits are not set, AVX instructions generate undefined opcode exceptions when executed.
You can work around this by setting these bits manually in the debugger after the OS has booted and before you run your AVX-enabled application. Do (r xcr0=7). Then read CR4 (r cr4), or in bit 18 (0x40000), and write the new CR4 (r cr4=40xxx). However, without OS support for YMM state management, the YMM registers will not be saved or restored on context switches. Due to this, we suggest running only one single-threaded AVX-enabled application at a time. Also, the upper 128-bits of the YMM registers will be undefined when your application begins - without YMM OSXSAVE support, the YMM registers will not be cleared by the OS like they might with OSXSAVE support. Linux kernel 2.6.30 supports YMM state management. If you boot a guest in SimNow that uses Linux 2.6.30+, then the guest OS will properly

enable and use XSAVE to manage YMM state. In this case, you will not need to set XCR0 or CR4.OSXSAVE manually.

4) *Are there any known bugs in SimNow's AVX implementation*?

Yes. If a 256-bit floating point operation generates a floating point exception, and MXCSR exception bits should be set from both the upper and lower 128-bits of the operation, then only a subset of the MXCSR exception bits may be set in the simulated processor.

5) *How do I read the contents of the new AVX register state through the debugger?*

Read the lower 128 bits of YMM registers with (r ymml0). Read the upper 128 bits with (r ymmh0). Read XCR0 with (r xcr0).

**Known Issues**

- The IDE controller(ide:0) does not work on "drachma_peso_1p_family10h.bsd", The SATA controllers(sata:0 sata:1) must be used to boot an OS.
- There is a known incompatibility with Windows XP SP2 product activation and SimNow. If you boot Windows XP SP2 in SimNow, you will receive an error like “A problem is preventing Windows from accurately checking the license for this computer”. The SimNow team is looking into the root cause for the problem.
- SuSE installation media containing both 32-bit and x64 installations will default to the 32-bit installation. WORKAROUND: To perform an x64 installation, type “linux64 initrd=initrd64” at the initial text mode screen’s “boot:” prompt. Note that this workaround is not necessary when using the Matrox® G400 video adapter.
- The floppy device model has known issues with the Linux kernels.
- PCI interrupts from devices on the subordinate buses of the AMD-8131 and AMD-8132 chipsets aren’t routed correctly when in legacy PIC mode.