

- None yet.

---

This section covers the fixed bugs and added features as well as the known bugs and missing features in the latest available release from ASH WARE.

## 2.1 Version 2.10 Build A, B, and C (2015 Nov 1)

- Fixed numerous (>45) minor 'nuisance' bugs since Version 2.10 Build A.
- Added ability to view a variable in the Waveform Window (shows value over time) directly from the source code window without having to (first) add it to the 'Watch' window.
- Numerous Watch Window improvements including the ability to view Script Variables in the watch window. (Version 2.10 Build B only)
- Variable debug windows (e.g., Watch Window, Local Variable window, Channel Frame Window, etc) now highlight variables (color red) that have changed since the last time the simulator was stopped.
- Added an persistent 'Auto-Fit' setting to the Circuit and State Machine windows in which they are maximally zoomed to fully fit into the entire window, every time the window size changes. (Version 2.10 Build B only)
- Fix a waveform window issues with multi-target environments that prevented easy change of waveform node names and other issues. Also fix some waveform display of data variable issues.
- In multi-target environments, fix a spurious pop-up dialog warning message that occurred when a window that didn't support the active target was updated.
- Improve support for multiple-target script and vector handling. Note that the recommended script configuration is a single script file associated with the first target in the project.
- Right-clicking on an in-scope variable during simulation now provides the option to add it directly to the waveform window for graphical viewing.
- The MC33816 Simulator has been released to production.
- Added an important new demo that illustrates end-to-end 'Moving to Hardware.' This demo co-simulates a Host CPU and in fact drives the MC33816 using only a series of optimized-for-speed SPI accesses. The MC33816 is fully controlled by the host CPU code using only SPI accesses, thereby fully illustrating every required aspect of running the MC33816 from the host. Of particular importance is the compiling of the MC33816's startup script (that fully initializes the MC33816 registers at startup prior to running the MC33816) into a series of optimized SPI block writes and word writes of the registers and any initialized data RAM variables, including banked data RAM variables. (Version 2.10 Build B only)
- Added a eTPU/MC33816 demo that illustrates co-simulation of an eTPU and MC33816. The MC33816 uses one of it's channels to drive four injectors. The four injectors are organized into two banks, and each MC33816 controls one bank of two injectors. The eTPU monitors an incoming CAM signal to generate the four 'START' pulses. (Version 2.10 Build B only)

- 
- Added an `_ENTRY` tag to MC33816 Assembler. This is used to mark certain labels as code entry points such that no warning is generated for an unused code label. This is similar to the existing `_ISR` tag that (Version 2.10 Build B only)
  - Complete Simulation to Hardware path is supported. All register initialization under simulation is done using a startup script, and this same startup script can be compiled in a 'C' compiler and converted to a SPI Block writes. The Data RAM's initialized values are output from the assembler into a data file that also can be compiled in 'C' and transferred across the SPI bus. Run time scripts, when written as SPI writes (the new way of scripting) can also be compiled in 'C' and written across the SPI bus. In conjunction, these capabilities allow the exact simulation input to be run on the hardware, thereby 'Connecting the Dots between Simulator and Hardware'. (Version 2.10 Build B only)
  - Less than the guaranteed two-cycle availability on reads and writes as well as not keeping the SPI address register (`spi_add`) constant result in warnings as do invalid access addresses and access of invalid locations result in warnings. These warning can be disabled in the message dialog. Note that these violations actually generally work ok, the issue appears to be sporadic unavailability based on SPI bus activity resulting in occasional collisions.
  - Support for use of the IR register in SPI Backdoor reads/writes now supported.
  - The Bootstrap sequence is now supported in that the value of the 'Bootstrap\_charged' register (0x1A5) reflects the injector connections. However, the simulation begins after the bootstrap sequence has already completed so the `bootstrap_init_timer` field contains 0x34 and the `hsx_src_1V` and `hsx_bs_charged` fields reflect the presence/absence of a charging path (due to presence of injectors, high side driver shorts or injector opens.)
  - Supporting many additional simulation configurations:
    - SINGLE CORES:
      - Ch1Uc0
      - Ch1Uc1
      - Ch2Uc0
      - Ch2Uc1
    - DUAL CORES:
      - Ch1Uc0, Ch1Uc1
      - Ch2Uc0, Ch2Uc1
      - Ch1Uc0, Ch2Uc0
    - QUAD CORE:
      - Ch1Uc0, Ch1Uc1, Ch2Uc0, Ch2Uc1
  - Note that there was no missing feature associated with testing (using for example `'jocr <SomeLabel> ocur;'` or similar test with `'wait'`;). The manual is confusing, however there is no 'microcode enablement of these tests required for these tests to yield proper results based on the value of the registers `Curr_block_access_1` (0x188) or `Curr_block_access_2` (0x189) as described in the FSL MC33816, revision 4, dated 4/2014 on page 186.
  - Support MPC5676R and MPC5777C 3-eTPU system simulation configurations (System DevTool).
  - Add the Rainier (MPC5746R) eTPU configuration.
-

## 2.2 eTPU Simulation Model Issues

These issues that have not yet been fixed.

- This script command is not working with this value. `write_tcr1_prescaler(1);` (full speed TCR1 clocking.)
- The angle mode hardware cannot be clocked by the TCR2 prescaler in eTPU2+.
- The Coherent Dual-Parameter Controller (CDC) is not implemented

## 2.3 MC33816 Simulator Issues

The following are the known issues.

- Limitations on interrupts (automatic return mode not functional.)
- Instruction 'steoa' is not supported. Instruction 'stfw' (set freewheeling mode) is not yet supported. Note that instruction 'stslw' had been incorrectly listed in the previous release as not being supported. instruction 'steoa' has been moved to the 'Support not planned in production release' list.
- The simulation model starts its simulation AFTER the bootstrap capacitor charge sequence has completed and as such does not support simulation of the actual charging sequence itself. However, the initial simulation state does present the post-charging sequence correctly including all fields within the 'Bootstrap\_charged' register (0x1A5) based on presence of injectors, injector opens, and shorts on high side drivers. Faults such as injector open, high side short, etc, are not modeled.
- The strong versus weak load biasing (pull-up, pull-down, strong/weak bias, etc) are not yet modeled. Specifically, the strong and weak load biasing are modeled identically. See See MC33816 User Manual Rev 4, 2014-April page 51, section, Load Biasing Structures.
- Feedback filter comparators (both diagnostic and current sense) only support the default 'type 1' filter type. Note that the filter demands that a counter reach a programmable count before allowing a change in the comparator to have an effect. Type 1 resets that counter on every comparator output toggle. Type 2 decrement the counter (thereby delaying some ... but not as much) the counter when the comparator toggles.
- The ability control the Output Driver gates directly from the input pins using 'Output Routing'. See the Lsx\_output\_config Registers (0x142, 0x145, 0x148, 0x14B, 0x14E, 0x151) on page 159 and the Hsx\_output\_config Registers (0x155, 0x158, 0x15B, 0x15E, 0x161) on page 161.
- The ability to configure the OA\_x pins as analog input/output signals is not modeled. See MC33816 User Manual Rev 4, 2014-April page 66.
- Diagnostics only supports a "filter\_type" of 0.

- User-defined external circuit .DLL not supported.
- The 'Automatic Offset Compensation' is not modeled, nor is the compensation itself. Note that this feature is used to improve the current sense block accuracy, especially to reduce temperature effects on the offset. Execution of the 'stoc' command has no affect. Note that this feature will not be in the first production release.
- Driving of the OA\_1 and OA\_2 pins as an analog output pin with one of the internal analog signals is not supported. Nor is the 'OaSelect' (analog multiplexer) nor is the programmable gain, OAGainx(1:0) supported.
- The use of the current sense DAC's as Analog to Digital converters using the 'stadc' command is not supported. Specifically, 'stadc on sssc;' will result in a warning which can be disabled but the affected current sense block will remain in 'adc off' mode.
- Vcc5 (external 5 volt supply used to generate the 2.5V reference and the 1.27V Bandgap reference from linear regulators within the MC33816) under-voltage and over-voltage are not modeled in the simulation and the Vcc5 voltage is modeled as being valid coming out of reset . See MC33816 User Manual Rev 4, 2014-April section 6.2.3 pages 20 and 21. Note that this means that the uv\_vcc5 field within the Driver Status register cannot get set and that the interrupt request (if enabled in the Driver Config register) cannot be generated.
- Vccp (Internal/external 7V power supply) undervoltage is not modeled. See MC33816 User Manual Rev 4, 2014-April section 6.2.3 pages 23. This means that the vccp\_irq\_en bit of the Driver\_status register cannot get set high and that no interrupt request can be issued to the host microcontroller via the IRQB pin.
- The DRVEN input pin is not modeled such that it does not disable any of the output pre-drivers even if programmed to do so. Specifically, the LS7 pre-driver is not inhibited by the DRVEN pin.ls7\_ovr bit of the Driver\_config register (0x1C5) is set to '0' (reset value is '1'). See MC33816 User Manual Rev 4, 2014-April section 6.2.3 pages 24.
- The cksys\_drven signal, which monitors the external clock and can shutdown the external drivers on 'loss of clock' is not modeled. The corresponding bit in the Driver Status register always reads 0 (no loss of clock) 0x1D2. See MC33816 User Manual Rev 4, 2014-April pages 26, 34, 42.
- The DrvEn input, which shuts down output drivers when low, is not modeled. Additionally, there are certain overrides that allow output drivers to be driven even with the DrvEn pin low, (e.g. the the hs5\_ls36\_en\_ovr bit of the Driver\_config register (0x1C5)) these are not modeled either. See MC33816 User Manual Rev 4, 2014-April pages 34 and 42.
- The slew rate at the gate, which is controlled by the slew rate registers, is not modeled. (However the reading/writing of the slew rate register via the SPI bus and the stslew instruction are modeled.)
- Non VBoost mode where the VBatt is connected to the VBoost pin and the Vboost divider is set to 1/4 (instead of 1/32.) Note that in this mode, the allowed Vboost DAC value is 0x9 to 0xF8. See MC33816 User Manual Rev 4, 2014-April section 6.2.7 page 31.
- Detection of missing ground connection is not supported. Signal uv\_vccp is not modeled and therefore can never be asserted. See MC33816 User Manual Rev 4, 2014-April section 6.2.9 page 32.

- 
- Temperate monitoring is not supported. The over\_temp bit in the Driver Status register cannot be set and no host MCU interrupt via that driver\_config register can be generated from an over-temperature. See MC33816 User Manual Rev 4, 2014-April section 6.2.10 page 33.
  - The bootstrap capacitors which must be charged before the high side drivers can be powered are modeled as charging instantaneously. This is modeled this way to avoid the (up to) 51 ms of simulation time for these capacitors to charge. Also, only the default start-up bootstrap sequence and path (capacitor is charged through the external load using the low-side driver's bias current) is modeled. See MC33816 User Manual Rev 4, 2014-April section 6.3.2.2 page 40.
  - The current sense resistor's differential amplifiers output's ability to be routed to the OA\_1 and OA\_2 is not modeled. Similarly, the oa\_gain which provides a programmable amplification prior to the OA\_1 and OA\_2 pins is also not modeled. See MC33816 User Manual Rev 4, 2014-April page 52.
  - The loss of clock on the 'cksys' pin and switch to the backup (1MHz) clock is not modeled. See MC33816 User Manual Rev 4, 2014-April page 69.
  - The 'bandgap ok' signal (bg\_ok) is not modeled and the bandgap reference is always modeled as being ok. Similarly, the PORResetB signals which monitors the VCC2P5 voltage is modeled as always being valid. Similarly the SPIResetB signal is modeled as always being not in reset (valid). Similarly, the RSTB signal (which is 'anding' of these other signals) is modeled as always being not in reset. The simulation can be thought of as beginning when the various internal blocks are all valid. See MC33816 User Manual Rev 4, 2014-April page 79.
  - The vboost\_disable\_en signal which detects an undervoltage on the VBOOST circuit and turns of the output drivers on and undervoltage is not modeled. See MC33816 User Manual Rev 4, 2014-April page 80.
  - The SPI Bus is modeled at the bus level but not at the signal level. In general, the script commands can be thought of as being SPI bus commands that take no time to execute. Along those lines, the script commands within the script commands file can be exported as a sequence of SPI bus commands suitable for driving the actual hardware. See the miso\_slew signal on MC33816 User Manual Rev 4, 2014-April page 82 for an instance of the SPI bus not being supported at the signal level.
  - Using the four current sense DAC's in their (alternate) Analog to Digital (A/D) mode is not supported. See MC33816 User Manual Rev 4, 2014-April page 194.
  - The Trace\_unit Block is not supported. This block normally allows the code flow to be tracked external to the MC33816. See MC33816 User Manual Rev 4, 2014-April page 107.
  - The Global Reset Registers is not supported. This allows the device to be forced into reset by the host MCU by writing specific values (0xF473, 0x57A1) to this register using the SPI bus.
  - Writing of the PLL Configuration Register is not modeled. The PLL is assumed to be functional. See MC33816 User Manual Rev 4, 2014-April page 91.
  - The 'Built In Self Test (BIST) block is not supported. See MC33816 User Manual Rev 4, 2014-April, section 6.5.14) on page 93.
  - The Device Lock Block is only partially supported. Not all registers are protected against inadvertent

---

writes. Also, the lock of the last 16 data ram words is not supported such that this area can be written even when locked. See MC33816 User Manual Rev 4, 2014-April section 6.15.15.1 page 94.

- The reset behavior register is not supported. This allows certain fault conditions to be automatically reset when read by the external SPI bus. For instance, a diagnostics error can be reset when read across the SPI bus. See MC33816 User Manual Rev 4, 2014-April page 95.
- The Cipher Unit Block is not supported. This decrypts the executable code image as it is downloaded into the cores' code memory. Note that the simulator's code image is extracted from the ELF/DWARF file, though the encrypted 'CIPHER' file suitable for download into the actual hardware is also available from the ASH WARE toolset.

## 2.4 Bugs and GUI Issues

These issues that have not yet been fixed.

- The 'Trace' window is not implemented in DevTool. This window shows past instructions and other events.
- The 'Fast Text Window' style windows are not editable in DevTool. These windows include the memory window, the configuration window, etc. Note that these windows were editable in Mtdt.
- The 'circuit' windows ('DC DC Converter' and 'Injector') are not editable.
- The syntax highlighter can make the source code, vector, and script commands file window sluggish.
- The scripts commands window does not support auto-code completion.
- The mouse-hover hints that explain how the MC33816 instruction works requires that the line contain a terminating semi-colon to show. This prevents the user from partially typing the line and then hovering over the instruction to determine the remainder of the syntax.
- The mouse-hover hints that explain how the MC33816 instruction works are disabled when the IDE first launches. It is required to simulate a bit before they are enabled.
- The section in the 'registers' window where the Timer Counter frequency is shown need to also show their frequency and period!!!!
- When zoomed in on just a small portion of a waveform and using the 'auto-range view' function, the function should (but does not always) extrapolate the sides of the waveform to find the min, and the max. Instead, the points just outside the view edge are used as the 'min' and 'max.'
- When changing projects, the old environment's open files don't get fully closed such that if a file changes on disk, the file monitor of the closed file remains active and can cause a spurious fault resulting in a spurious message. SOLUTION: When changing projects, close/re-open the new project. This clears out the old file watchers.
- When changing projects, if the old environment had more than the default number of panes (which is 5)

---

then the old environment's panes are not dismantled such that the new environment will have all the old environment's panes.

- In mixed-asm/source view, when single stepping through the entry table, multiple lines may be highlighted at one time.
- When single stepping a state machine function, when single stepping to the start of a function, the highlighted red 'current line' sometimes appears at line of the first source code and not at the entry table.
- In certain cases, the waveform is not visible even though the value is shown in popup box and is between the min and max y axis values.
- Clean up behavior of state machine view when transition from sim -> edit mode, but have not hit reset.
- If an error or warning is clicked on in the Output Window while the simulation is running or paused, all source code line highlighting (current line, breakpoint line, etc) lose their color and become white.
- When pasting text above a breakpoint or bookmark, the breakpoint/bookmark is not tracking its associated text correctly.
- The 'Tab' at the top of the editable file, especially when the filename is long, can get messed up when closing the file.
- Although the ISR scripts and Startup Scripts do execute correctly, it is not possible to set breakpoints or single step them in the GUI as is possible with primary ISR scripts.
- Because each analog and discrete node has its own buffer, the buffers can fill (and wrap) at different times, and thus in total hold data covering varying lengths of simulation time. When the waveform window view is set to large time scales, this can result in waveform data starting to draw at different times for different signals.

## 2.5 Multi-Target Issues

These issues that have not yet been fixed.

- Various debug features (step state, goto state, etc) do not work well when multiple targets/cores are using the same state machine.
- The vector file is target centric such that it can only drive a wave command unto pins from a single core.
- The project window needs ability to view/hide vector and script files.



---

This section enumerates features added and bugs fixed in previous releases.

### 3.1 Version 2.00 Build E (2015 Sep 2)

- Fixed a bug in which grammatically-incorrect script commands were being accepted. In certain cases, script commands were allowed when the number of closed parentheses did not match the number of open parentheses. For example, the following was allowed even though it is invalid.

```
write_reg5(0x17), REG_CHAN); // Missing close-parentheses
```

Note that the above is now (properly) generating an error message. The proper script command is below.

```
write_reg5(0x17), REG_CHAN));
```

- Fix/improve handling of 'overwrite vs. insert' mode in editor windows. In overwrite mode, cursor becomes a solid box to make it obvious that the mode has changed.
- Fix a USB license dongle issue with MC33816 DevTool wherein when a project had circuit windows, the application could take a minute or more to open.
- Properly supporting registers 'Flags\_source' (0x1C3), 'Flags\_direction' (0x1C1) and 'Flags\_polarity' (0x1C2) as well as the I/O pins and flags associated with those registers (FLAG0-FLAG2, START1-START6, IRQB, OA\_1, OA\_2, DBG, internal flags FLAG12-FLAG15.) As part of this implementation/fix the following issues were addressed:
  - Use of the input pins and output pins as 'flags' is now supported. This includes using flags to read input pins, write output pins. See MC33816 User Manual Rev 4, 2014-April page 96. This also includes the OA\_x pins being used as input/output flags.
  - Fixed an issue with the Flags\_polarity Register (0x1C2) not getting applied to the start pins as part of start management.
  - The IRQB pin was being driven with the IRQ state even when configured as a generic I/O pin. Note that for the IRQB state to be driven unto this pin, register 'Flags\_source' (0x1C3) bit 9 must be a zero. However the default is for the IRQB pin to be a generic input pin.
  - When driving one of the pins associated with a flag (E.g., the IRQB pin is associated with FLAG 9) as a generic Output pin, then the pin follows the 'AND' logic of the four cores' version of flag 9. In other words, the pin is a '1' only if all four cores have executed the 'stf high b9'. instruction. However, the pin will be a '0' if ANY of the cores has executed the 'stf low b9;' instruction. Note that this logic can be easily missed because the internal flags for all four cores reset to all 1's.
  - When a pin is configured to be a generic input pin instead of it's normal function (e.g., Start 1 is a generic input pin) and the invert bit is set (Flags\_polarity Register (0x1C2) bit 3 is set) then the value read by cp flags ir; or tested by jocr flag3; should be the inversion of the input pin, but it is not. The invert bit was having no affect in this case.
  - Flags\_source Register (0x1C3) bits 1, 2, and 3 (corresponding to the FLAG0, FLAG1, FLAG2) are

'hard-wired' as 1's. However, these values were being allowed to be cleared to 0's which was not how it works. These bits now remain 1's, even if written to 0's.

- Renamed the 'I/O Pins' window to be the 'Flag Pins' window and greatly improved the window in terms of fully providing all the information required to determine what is going on with the pins in all cases. Moved the 'DRVEN' and the 'RESETB' pins to the 'Config' window as they no longer are appropriate for the 'Flag Pins' window ... not actually being flags or following the patterns of those pins.
- Added documentation for the MC33816\_SPI\_SPACE 2 which can be used to access registers and the data RAM's.
- Provide labels as auto-complete options for ldjr1/2 instructions.
- Fixed a visualization issue with the Voltage seen in the Injector window. The VBatt Vds is showing the incorrect voltage. Note that this is just a visualation bug ... (importantly!) the simulation model is functionally correct.
- Added File 'StdDefMc33816.h' is missing the Err\_ucXchY\_1 Registers (0x162, 0x164, 0x166, 0x168). NOTE: This was actually fixed in the Version 2.00 Build 'C' version released on 2015-Aug-3 but was not documented in that release.
- In the state machine designer, users can now specify user-friendly aliases for conditionals on a state-by-state basis. The aliases display instead of the underlying conditional name on the state machine graph.

### 3.2 Version 2.00 Build C (2015, Aug 3)

- Added File 'StdDefMc33816.h' is missing the Err\_ucXchY\_1 Registers (0x162, 0x164, 0x166, 0x168). NOTE: Previous versions of these release notes fail to mention that this was actually fixed in this release.
- Added the ability to initialize global variables and databank variables.
- Added the ability to load the address of a variable into the 'ir' register using the 'LOAD\_IR @<VariableName>;' syntax. Note that the address of a variable, databank, and databank member can all be loaded.
- Add the ability to specify initial values for variables and databanks. Initial values are output in auto-generated 'C'-compliant files for use in script files and host code.
- Fix the output of databank type definitions to the ELF file, so that databank variables can be properly viewed in the watch window.
- Added an eTPU 'Hello World' demo.
- Improved the import of Mtdt System Simulator project files. User notifications are more accurate. Source search paths now convert correctly.
- Improved source code search path dialog. Note that this is ONLY USED ON EXTERNAL BUILDS OF CPU CODE! Search paths can now be located in project file or the environment file (user's choice.)

- Fixed several misc. spurious 'Internal Diagnostic Fault' type messages such as when an external .bat file is specified for an external build.
- Fixed the OutputWindow such that 'Click-To-Error' works even when the directory name has a '(', ') or '-' character. These can occur when the user makes a second copy of a directory.
- In the watch window, when the <add symbol> marker is clicked on a dropdown combo box appears that provides register and global variable options for watch selection.
- Support code references (script function and variables) in script files.
- Fix an editor crash in the case where multiple lines are selected and then a keystroke is made.
- Remove an unhelpful 'invalid target designator' warning in the editor.

### 3.3 Version 2.00 Build A (2015 June 23)

- **Misc**

- Improved 'snappiness' of editing source code files by improving the performance of syntax highlighting.
- Fixed a bug in which if a file is changed on disk and is reloaded into the simulator, it fails to get re-parsed if it was a script file, and fails to trigger a rebuild if it is a source code file.
- On an automated test it is possible to override 'fake' a passing '0' exit code by closing using the files->exit menu. This now results in a '1' (fail) exit code.

- **Scripting**

- Support enum type definitions and literal usage in the scripting environment.
- Improved script file stepping and breakpoint handling such that when breaking at an instruction following a timing instruction (e.g. 'wait\_time') the timing instruction fully executes before simulation is halted.

- **State Machines**

- Added multiple-state/transition selection capability via dragging the mouse
- Added ability to horizontally align all selected states and transitions (previously, only 'vertical align' was supported)
- Added ability to create snapshot image files (.png, jpg, etc) of the state machine
- Added ability to scroll view via mouse drag
- Added ability to zoom-in and zoom-out

**• Multi-Target/Core Simulation**

- Support co-simulation of an eTPU with a MC33816! (Added build batch file for 1 eTPU target + 1 MC33816 core, a second model with 1 eTPU and the full 4-MC33816 cores to follow)
- Added a model for co-simulation of the eTPU and MC33816 together (e.g. allow eTPU's output channels to drive the MC33816's 'start' input pins.)
- Added extended scripts for buffers, gates, 'OR', 'AND', 'XOR', 'NOR', 'NAND' and 'NXOR' to drive pins between targets. (Example: the eTPU's output pin 4 and output pin 5 could be connected to an 'OR' gate and the 'OR' gate's output could drive the MC33816's 'START3' pin.
- Fixed multi-core/target debugging issues (e.g. stepping, breakpoints, goto cursor, etc) in the case where the cores are different executable image files (often the case.) The source level debugging is now updated relative to the active core so that things like single stepping/breakpoints/etc will occur for whichever core is active (but NOT for the inactive core!)
- Mixed assembly view is now done relative to the active core/target such that if a source code file is not used by the active target then in dis-assembly view will show there being no dis-assembly. This is consistent with the approach of making DevTool more 'active target/core' focussed such that information for any inactive cores/targets is less 'in the way' thereby reducing confusion.
- In multi-target simulation, fixed a problem in which only the first target/core's signals would display correctly in the waveform window.
- Changed the GUI's 'Multi-Target' behavior of script commands files when only a single target has a script commands file (which is generally the case). Single-stepping the script commands file can be done from ANY target and does NOT change the active target. Also, breakpoints, goto cursor, etc, work even if the script commands file is not in the active target.
- In multi-target simulation mode, added the ability on the 'Memory' and 'Wait Rows' (33816 only) windows to specify the target/core for which the window should render information.
- Fixed a bug in which the analog portion of the simulation would not occur unless Channel 2 Core 1 is enabled to execute code (33816 only). This includes simulation of circuits (DC/DC Converter, Injectors, etc, as well as an external gates placed for (for example) drive the start pins from an eTPU's output pins.
- Fixed a Waveform Window bug in which there was only a single copy of the 'wait', 'timer EOC', and 'Start Event' indicators. Each core now has it's own copy

**• Circuit Windows**

- Added ability to edit fields (as applicable)
- Redesigned ALL the schematics. The HS, LS, and Current Sense schematics have a complete makeover.

- Fields that have changed since the last simulation pause, single step, goto cursor, etc, are highlighted red. Remainder are now blue.
- In the waveform window, fixed an issue with auto-ranging analog signals, esp. when highly zoomed.
- Fixed a bug in the 'Replace Text' dialog such that when the replacement text is a superset of the text being replaced, it doesn't try to replace the same text over and over again.
- Fixed a bug wherein the running of simulation after creation of a discrete node could sometimes cause a crash.
- When auto-running the simulator as part of a test suite, a failure to load a script file would halt the test suite (bad.) Instead, the DevTool now exits with an error code thereby allowing the test suite to proceed (but with errors.)
- The 'Timers' window is not function yet was showing up the 'View' menu. It is now removed from the view menu to avoid confusion. Note that the 'Timers' window allowed 'start' and 'stop' tags to be embedded in the source code such that the amount of time it takes to traverse from one spot in the code to another can be measured. This will be enabled in a future version of software when it is correctly supported in DevTool.
- Fixed bug in which ALL feedback registers (0x180, 0x181, 0x182, and 0x183) were getting reset to 0x4403. The four cores' feedback registers are now being correctly reset to 0x440C, 0x880C, 0x1030, and 0x43C0, respectively.
- When exporting SPI-commands files, made the 'Unsupported script command' message disable-able so that one can create the spi-commands file without having to click through zillions of dialog boxes.
- Improved several MC33816 error/warning messages in various cases (e.g. on an 'invalid immediate value' error would report 'missing jump destination' instead', etc.)
- Modified the MC33816 memory windows such that the address field shows word addressing rather than byte addressing.
- Removed the spurious warning when an output driver is configured as 'undef.' Also, added warning when a diagnostic comparator is queried using 'wait' or 'jocf'.
- Extended Buffers and Gates capability such they can 'drive'/'be driven by' the MC33816. For example, the MC33816's IRQB pin could drive (say) the flag0 input via placement of an logic device using the 'place\_buffer()'; command.
- Added the capability to drive the rest of the MC33816's binary pins via script commands (e.g. the OA\_2 pin when configured to be a digital I/O pin, etc.)
- Added several missing pins from the MC33816's 'I/O' pins window. Also, added important details such as (e.g., OA\_2 is configured as 'Analog Output', Input, or Digital output pin, etc.)

- Fixed a spurious diagnostic message that can occur when a row in the wait table is enabled that has not yet been initialized.
- Fixed a bug with the joslr and joslf instructions when testing the 'start1, start2, ..., start6' pin states. The instruction had been returning 'true' on matches on start pins matching a 'high' pins, whereas the 'low' pins needed to also match. Consider joslr Start12Set start12; This should only return 'true' if both start1 and start2 pins are high. The problem is that it should return 'false' if (say) start3 is also high. However only 'pins that should be 1' were being tested and the 'pins that should be zero' were not being tested.
- Fixed bugs associated with viewing multiple eTPU Channel variables as Discrete's in the waveform window at once. There were problems associated with auto-ranging and with saving/restoring the project.

### 3.4 Version 1.01 Build D (2014 Oct 1)

- Fixed the 'error color' (making it far lighter) such that the line of text where the build-error occurs can still be read. Previously, the 'error color' was so dark that it could not be read on certain monitors.
- The injector model multi-injector in a bank bug has been fixed. The diagnostic feedback comparator outputs are now being simulated correctly.
- The set\_fault() script command now supports all planned faults including 'device open' and 'device short' for HS, LS, and INJECTORS. Injectors additionally support shorts (either injector side) to the BATTERY, to GROUND, and to the BOOST supply.
- The injector resistance and the resistance of the HS and LS driver's Rds-on is now modeled and can be specified via script commands.
- The ability to test the flags using the 'wait', 'jocr', or 'jocf' instruction has been added and the flags register is now (correctly) global to all cores as it was previously (and incorrectly) modeled as being local to each core.
- In the watch window, fixed the display of banked variables. The watch window was calculating the wrong bank-address and displaying the wrong address's variables when displaying the bank-variables.
- Changed the verify\_current() script command. The current sense current resistors are identified using the 'SENSE1' through 'SENSE4' syntax. Previously it was documented as such, however the actual script command used the 'VSENSE1' through 'VSENSE4'. ALSO, added the ability to verify the current through the high side and low side drivers (HS1-HS5, and LS1-LS7) using the 'verify current' script command. Note that previously only the ability to verify the sense current was supported.
- The help file can now be accessed for extended instructions (right-click and select 'Open help...').
- The instruction examples can now be copied to the Windows' clipboard (right-click and select 'Copy Example...').

- An auto-code helper capability has been added for writing MC33816 assembly. When starting to type in an MC33816 source code window, assembly instruction options appear in a floating list box, allowing auto-completion of if instructions and their parameters. More detailed help hints appear when the mouse is hovered over the auto-complete options.
- Improve source code syntax highlighting in cases where `/*` multi-line comments change to `///  
//` line comments.
- Include files meant to be used in simulation script files (e.g., `StdDefMc33816.h`, `etec_sim_autodef.h`) are now installed into an Include directory under the executable install path. When script files are processed, this directory is automatically searched.

### 3.5 Version 1.01 Build C (2014 Sep 6)

- Changed the default font to 10pt Courier New. Users can now configure the font on a project-by-project basis in the IDE settings. Font changes affect all editor windows and the simulation register/status windows.
- Allow tabs within DevTool to be floated into their own window, and can later be re-docked into the main DevTool application. The floating/docking control is selected by right-clicking in the tab and selecting the float/dock option in the context menu. Window types that can be floated include the Waveform, Watch, Local Variable and Channel Frame window types, as well as all of the simulation register/status windows.
- Hovering the mouse over a variable in the code will show its current value (in-scope variables only for given execution point)
- The simulation detects a large number of suspicious situations (such as setting the VBoost DAC above 0xD0 ... not recommended per FSL literature.) These messages can be disabled within the Message Options dialog box. However, it was often difficult to figure out exactly which message to disable within the dialog box because there are so many possible messages and because the dialog box was poorly organized. The dialog organization has been improved and the messages themselves now generally indicate exactly where, within the messages dialog, to find the message to disable. The message ID is also now listed in the dialog in case the user wants to disable the message from the command line (when the IDE is launched ... useful for automated testing.)
- When compiling (or building in the MC33816) using the `<alt F9>` hotkey, changed files were not getting saved prior to the compile/build. This has been fixed.
- Fully-implemented the 'rstreg' instruction which was missing several of the TargetReg field values (e.g. 'all' was not supported.)
- Support for 'wait', 'jocr', 'jocf' testing of the the Vds and Vsrc shortcut diagnostic comparators (`sc1v-sc3v`, `sc1s-sc3s`, etc) as well as 'Operation Done' is now supported. (Note that previous versions supported 'Operation Done' using 'jarf' and 'jarr' ... just not 'wait', 'jocr', 'jocf')

- Vastly improved the DC/DC Converter (VBoost) window and added much needed missing information.
- Added detection and warning when multiple low side drivers in same bank are 'ON' at same time.
- Injector simulation model now functional for multi-bank simulations.
- Implemented most missing instructions (sto, jfbkf, etc.) See missing features list for remaining unsupported instructions.
- Script commands for simulating opens/device-shorts/shorts-to-battery, etc, have been implemented. Note that the 'set\_fault();' script command is used which is a different script command than was originally documented and intended.
- The set\_reset\_source() script command has been added to allow users to configure the reset\_source register. Any read of the register clears all its bits.
- The Circuit Windows did not draw correctly when the window is resized smaller such that the scroll bar appears. The draw function didn't work in this case and the window became illegible. This has been fixed.
- Vastly improved the Injector schematic such that much, much more important information is provided. Also, fixed numerous misinformation.
- Numerous validity checks now performed on the 'place\_injector()' script including that VBoost High Side Driver must be connected to HS2 or HS4, that any shared current sense drivers also share the same current sense resistor, that any shared high side driver must share both Boost and Battery high side drivers, etc.
- The DAC settling time register (0x1A9) and current filter registers (0x198, 0x199, and 0x19A) are now implemented. This prevents a wait instruction from returning 'true' when waiting for a DAC output state.
- Added a warning if the VBoost pin voltage exceeds the maximum allowed (72V.) Note that (similar to all messages) this warning can be disabled.
- Support simulation of the DAC Comparator filters (registers Current\_filter12 Register (0x198), Current\_filter34l Register (0x199), Current\_filter4h4neg Register (0x19A), and Boost\_filter Register (0x19D)) Dac4h4negFiltersAddr. Note that these filters de-glitch the comparator output by not allowing the comparator output to change unless the unfiltered output is stable for the number of clocks specified in the filter.
- Support simulation of the DAC settling time of 0.9 microseconds. The DAC output voltage is simulated as having linear slew from the previous DAC output voltage to the new DAC output voltage taking 0.9 micro-seconds.
- Support viewing of DAC1-DAC4L and DAC4Neg voltages in the waveforms window.
- Removed OA\_1 and OA\_2 pin voltage from being viewable in the waveforms window. Note that these voltages are not going to be simulated in the first production release which was confusing.
- When single stepping by hitting the F8 key at the 'wait' instruction, the simulator only advances by one



clock, but instead should simulate until past the wait instruction.

- There is not warning (and should be) if the VBoost DAC is given a value outside its allowed range of 0x9 to 0xD0. See See MC33816 User Manual Rev 4, 2014-April page 31, very top of page.
- Fixed an MC33816 simulation bug wherein the create wait table entry relative (cwer) was incorrectly calculating the address when the offset was negative.
- Added a set of extended instructions to the ASH WARE MC33816 assembler that relieve the developer of figuring out whether they need to use far or relative addressing for jumps and when creating wait table entries. The new instruction names are JUMP, JUMP\_ARITHMETIC, JUMP\_CONTROL, JUMP\_STATUS, JUMP\_START, JUMP\_CONDITION, JUMP\_FEEDBACK, JUMP\_CORE\_ID, CALL, CREATE\_WAIT\_ENTRY.
- When a multi-instruction cycle instruction is executed (such as the shri instruction, see instruction sequence below) then any subsequent ALU instruction (add, subtract, load immediate, swap, toc2, and toint) to that get executed while the mult-instruction cycle instrucion is still executing are supposed to be ignored. The bug is that (instead) the alu instructions get executed instead of ignored. For instance, the variable named 'Signature' in the sequence gets written with the value 0x9D97. whreas the 'ldirh' ALU instruction should be ignored, and is not.

```
ldirl 20h rst;
cp ir r2;
shri r2 4;
ldirh 9Dh _rst; // Should be ignored, but was not
ldirl 97h _rst; // Should be ignored, but was not
store ir Signature _ofs;
```

- [NOTE: This bug was still listed in the previous release even though it had actually been fixed.] Instruction ldfd does not write the output drivers, but it does perform the RAM operation.

### 3.6 Version 1.01 Build B (2014 July 11)

These features have been added and bugs fixed in Version 1.01 Build B which was released on 2014 July 11.

- Enforced access protection of output drivers by certain instructions (e.b. 'bias') thereby causing the effect of the instruction to be blocked. These can be enabled by setting the appropriate bits in the 'Out\_acc\_ucX\_chX' registers (0x184, 0x185, 0x186, and 0x187.)
- Asm816: Added various instruction extensions including an optimized constant load to the 'ir' register.
- Asm816: Added support for `-I=<path>` and `-d=<macro>` cmd line options.
- The Boost voltage comparator's filter is now modeled (0x19D). See MC33816 User Manual Rev 4, 2014-April section 6.2.3 pages 24.
- Fixed a bug in which, when switching project files, any non-standard panels in the previous project file

were being retained by the new project.

- Fixed a bug in which when switching projects, windows used for viewing a file (e.g. source code window or script window) from the old project were not fully closing such that if the file on disk (from the OLD project) were to change, an annoying diagnostic message was firing.
- The build define (e.g.. "MPC5554\_B") is now automatically injected into the script command environment as a macro.
- Fix a symbolic debug issue where when reading/writing a struct/union member value, if the member had a typedef'd type, the read/write could fail.
- In the target node settings of the project tree, settings for controlling the directory for object (intermediate) files and the directory for auto-generated code output from the state machine compiler have been added. The default is that object files get placed in a sub-directory of the project directory called "obj", and by default state machine auto-generated code files are output into the same directory as the final executable.

### 3.7 Version 1.01 Build A (2014 June 16)

These features have been added and bugs fixed in Version 1.01 Build A which was released on 2014 June 16.

- Fixed a bug in which the file being built (usually a '.elf' file) cannot be renamed (when right on the file in the project window) if the .elf file does not exist to begin with. This makes sense (say) for a source code file (because there needs to be an existing source code file to rename). But this restriction makes no sense for an .elf file because it is generated by DevTool, AND it makes it squirrely to rename the generated files.
- The Current Sense Block amplifier gain was ~20 mv/LSB. It is now now modeled correctly at 9.763 mv/LSB.
- Current sense block has a -0.25V amplifier output offset was not modeled and now is.
- Current Sense Amplifier bug in which the amplifier stops working altogether if the gain is changed (stgn instruction is executed) has been fixed.
- Diagnostics is mostly supported.
- SPI Loopback is partially functional. However, it acts just like SPI interface with the restrictions and core access limitations mostly not enforced. Also, the two-cycle availability on reads is not enforced.
- Watch window displays bank and global variable display.
- The ASH WARE assembler now produces the ciphered .bin file needed to load code into the device (using the SPI interface.) Note that for simulation an industry-standard .elf file is included.

- Fixed a bug in when testing 'ocur' or '\_ocur' in a 'wait' or 'jocr' or 'jocf' instructions.
- Fixed a bug in which execution of the 'Change Opamp Gain' instruction ('stgn') would set the gain to 0.
- Fixed a bug in which the Amplifier settling time, which is supposed to be 2 micro-seconds max, is clock dependent and changes if the core's clock frequency changes or due pretty much anything such as a change in the system clock frequency or a change in the clock divide or if the core clock is 1/4 the system clock, etc.
- The MC33816 resets to the frequency of 16.778 MHz. The 'set\_clk\_period(41666666);' script command should be used to set the device to the more typical 24 MHz. In a multi-core simulation this script needed to be executed on every core (e.g. ch2\_core0.set\_clk\_period(41666666);, ch2\_core1.set\_clk\_period(41666666);, etc). This has been changed such that a change to the system clock frequency on any core changes the system clock frequency for all cores.
- Fixed a 'duplicate DAC4L bug' in which the DAC4L value accessed across the SPI bus(e.g. using script commands) was not the same as the DAC4L value read/written in the UcReg's 'dac\_osoc' when run in Ch1.Uc0, or 'dac\_sssc' when run under Ch2.Uc0, etc.
- Fixed a bug in which the vector frequency is incorrect if the Ck\_per register is set to anything except '1' (Core clock must be half of the system clock for the vector frequency to be correct.) This has been fixed.
- Waveform \_CUR4L\_FBK was swapped with \_CURR4H\_FBK.
- Fixed a bug in which the keyboard's 'UP' and 'DOWN' arrow keys were not changing the active waveform in the waveform window.
- Do not allow organizer nodes to be selected in waveform channel selection dialog.
- Fix syntax highlight issue with comments on same line as preprocessor directives.
- Apply syntax highlighting to new text that hasn't yet been highlighted, when the cursor moves. (text was being left un-highlighted longer than necessary)
- Make several improvements to waveform discrete nodes, including fixing an issue where channel frame variables could be unassociated with a channel making their data signal behave strangely.
- In the MC33816 IDE, added an MC33816 assembler manual to the installation available in .pdf format from the Windows menu or in .chm format from DevTool's help menu.
- Fixed a bug in which the circuit windows definition file was not getting loaded thereby preventing viewing of the circuit.
- Fixed a bug in which the single step (F8 or F7) would stall at the 'wait' instruction, progressing the simulation only one clock cycle for each step. When at a 'wait' a step now runs the simulation until the first instruction past the wait, even when it takes many clocks until the wait-ending event appears. Fix a bug that causes an exception when editing MC33816 state machine transition properties. If the transition condition and then the transition priority are modified, an exception occurs when exiting the

dialog with an "Ok".

- Fixed a discrete node bug in the waveform. New discrete nodes that represented eTPU channel frame variables were not displaying.
- Fixed an eTPU graphical state machine editor bug that caused crashes in the both the editor and compiler (SMC). The case where a transition was left undefined, saved out to file, and then restored, was not being handled correctly.
- The cam/crank state machine demo was not configured correctly resulting in incorrect fuel and spark signals - this has been corrected.

### Legacy File Conversion

- Support conversion when using the windows 'open' function from (say) windows explorer. Previously, a Windows-induced 'open' would cause the project file to be treated as a text file. A conversion was required from the command line required the '-p=LegacyProjectFileName' option to be used.
- Support conversion of Waveform Window sizes and vertical scrollbars. Note that the 'Waveform Window' was formally refereed to as the 'Logic Analyzer Window'.
- Support conversion even if the executable image (.elf/.cod/coff, etc) file does not exist. Previously, the conversion would fail because those files are used to identify the build's source code files. However, when no executable image file can be found, the user must modify that portion of the project file.
- Support conversion of the thread groups.

### MC33816 Assembler 'ASM816.exe' Fixed Issues

- Added support for Cipher algorithm (non-ciphered .bin files would not work on the actual hardware)
- Added support for -msgStyle and -msgPath command line format (GNU, PSPAD, etc message format)
- Support for variables (sint16 and uint16) included indexed (banked) variables. Variable addresses are exported into the auto-defines header file for reference by the host code. Bank base index.
- Fully locates data including banked data. Macros support setting the bank address for defined banks.
- Added 'C-Like' preprocessor support.
- Removed 'missing preprocessor' work-arounds from state machine
- Fixed misc elf/dwarf executable image format issues.

---

## 3.8 Version 1.00 Build B (2014 March 12)

- Selecting vector file aliases for waveform channels was failing, resulting in a blank waveform button for the signal. The workaround was to select the underlying node (e.g. "\_tcrclk" rather than "Crank"). This has been fixed, however, the configuration of waveform channels will need to be re-done with this release.
- The editor response while syntax highlighting has been improved.
- Fixed a problem where the user was unable to disable message for HSR when HSR already set.
- DevTool automatically records all pin transition data, including the TCRCLK pin. In some cases this could lead to a failure in behavior verification when using previously generated .bv/.ebv files for which the TCRCLK pin logging was disabled. The workaround is disable behavior verification for the TCRCLK pin using the "disable\_ebehavior\_pin("\_tcrclk")" script command. Note that script file can be kept compatible between tools by making use of the #ifdef \_DEV\_TOOL\_ / #endif keepout.
- Right-clicking on a local variable or channel frame window column header could cause a crash-type failure - this has been fixed.
- The legacy project file import capability of DevTool has been greatly improved. Simulator message options get imported, logic analyzer nodes get read into the waveform window, and watch variable are transferred into the DevTool watch window.
- Settings to reload/reparse script and vector files on every reset have been added, as well as File menu options to explicitly reload and parse each file type.
- Hid a un-implemented 'All Targets' menu item from the breakpoints menu.
- Added a 'Reset at <time>' message to the output window so that the window does not go (disturbingly) completely blank when the simulator is reset in simulation mode.
- In the 'Build' menu added 'reload script' and a 'reload vector' options.
- Added options for ALWAYS re-reloading the script and vector files on reset. Note that the default is to reload only when these files have been edited in DevTool. (to adjust, in the project window, right click the vector/script files and select 'settings') Rational: this option is needed if the user has changed an included '.h' file.

### Crashes/Hangs

- Fixed a lockup problem that could occur when right-clicking the mouse in the waveform window.

### Help Documentation

- Help documentation improved, though still shortcomings.
- Help buttons in many dialogs have now been connected to the appropriate help documentation.
- Removed 'Mtdt' information from the DevTool manuals that was not applicable to DevTool. (Note that there are still known issues in this area.)

### Spurious Diagnostic Messages

- Fixed spurious diagnostic message that occurs when a text file that is open in the IDE changes on disk such that it gets reloaded.
- When the mouse is hovering over a panel divider and the user right-clicks, a spurious fatal diagnostic error occurs. This has been fixed.

### Installer

- When installing DevTool, no longer change Mtdt project associations to DevTool. Instead, these associations stay with Mtdt.

## 3.9 Version 1.00 Build A (2014 Feb 24)

The following features and bugs were added/fixed in this version.

Note that this release was given a '1.0' version designation because the DevTool eTPU Simulator is released to production. However, the MC33816 simulation is still pre-production.

- Added popop hint text added to MC33816 assembly code that describes the MC33816 opcode and in some cases the opcode fields.
- MC33816 assembler's `-version` command line option now produces the correct version.
- An errant "The default ETEC (eTPU) compiler installation cannot be found ..." message is getting printed when running the MC33816 simulator. The MC33816 simulator does not use this eTPU compiler so no such message is appropriate. Fixed.
- Complete redesign of the 'Text Search' dialog including adding ability for search to span multiple files. Many issues and shortcomings fixed and addressed including redo/undo tracking issues fixed.
- Waveform window was not allowing the cursor and other times to be drag-and-dropped into the wave view area to run to the specified time. Fixed.
- Fixed a problem in which the 'Goto State' was not working in the state machine window.
- When attempting to begin a simulation, but the build fails, then the output window is popped to the front so that the offending error can be readily viewed. This is good. However, if a script or vector file fails to load then the Output window could remain hidden, making it difficult to tell what the specific problem is with the script or vector file. This has been fixed.
- Fixed a bug in which when a file is changed outside of DevTool such that DevTool reloads the file, if the file was in mixed-asm-source view mode at the time of the reload, an errant diagnostics message was firing.
- Fixed a bug in which the code coverage boxes that show whether code has been traversed or not would sometimes not fully span the height of the code window.
- Fixed the `'-bd='` command line option. (overrides the build script ... typically used to change the simulation model)

- Support for eTPU2+

### 3.10 Version 0.82 Build F (2014 Feb 3)

The following features and bugs were added/fixes in this version.

- The following all appear to happen when the Current sense values are very large Waveforms "Auto range buffer" calculates an incorrect value which is less than "Auto range view"
- State Machine Compiler now is a stand-alone command line tool thereby allowing incorporation into an automated regression test. (Does not affect on GUI.)
- Added ability to specify the Channel Number (or in some cases Function Number) in windows such as the Channel Hardware window so they can be tied to the active channel (which changes for each thread) or a specific channel which does not change.
- Fix a crash that can occur if a needed .net service pack is not installed. Now detecting needed .net Service Pack, alert user, and gracefully shut down.
- Redirect 'Un-recoverable Error Reporting' to ASH WARE and no longer Microsoft.
- Added detection in the installer for .net.
- Added an Hour-Glass cursor and a moving status image that appear when building (compiling, assembling, linking, etc) so DevTool does not appear to be hung during lengthy builds.
- Fixed a crash that can occur in certain cases in the Waveforms window when the 'Auto range buffer' is used.
- In the Waveform window's waveform dialog, the auto-range calculation was not always correct because the very latest value was not always included in the calculation; this has been corrected.
- The ASH WARE MC33816 Assembler is now the default.
- Using the ASH WARE MC33816 assembler, multiple assembly (dfi) files can be built into a single executable image.
- Multiple MC33816 cores can now load the same executable code image. Note that only a single core can have 'build' enabled for each executable image. In words, only one executable core can have 'auto-build' enabled. Disable cores from building by right clicking on the project's 'target' node and selection 'Disable Build.'

### 3.11 Version 0.82 Build E (2013 Jan 13)

The following features and bugs were added/fixes in this version.

- In the Watch window, moved the '[register xyz]' information to the 'Type' column and not the 'Value' column where it was a bit confusing.

- In the Watch window, moved the '[register xyz]' information to the 'Type' column and not the 'Value' column where it was a bit confusing.
- Change the 'Edit Mode' of the 'value' field to occur on a single mouse click (a double-click had been required)
- Fixed issue in which swithing between mixed-source assembly and source-only view of a file could cause the file to be considered 'dirty' thereby triggering an (unneeded) rebuild when doing a 'make.'
- Added a 'NOP' to any new state created by the MC33816 Graphical State Machine tool. Previously there was no NOP and the issue was that the assembler generates an error on a line with a label but no opcode.
- Fixed an issue in which the 'Dongle.exe' and 'License.exe' utilities were not being installed on a License Server installation.
- Click-to-line on the Output Window was not working for errors reported by the ASH WARE State Machine Compiler when acting on the DFI files in the MC33816.
- Right clicking in space in the state machine window, pop now supports a 'Add State' capability.
- Deleting a state bug in MC33816 was leaving orphan transitiions.
- The delete keyboard key can now be used to delete states and transitions.
- Enabled the TrackWhell in the Waveforms window (changes selected channel.)
- When selecting the (unsupported) 'Threads' window it would crash the app.
- On the Edit menu, the 'block indent' and 'block un-indent' had the wrong hotkeys listed AND did not work.
- In the 'Options' menu the 'Waveform->Activate Right Cursor' submenu was not working.

### 3.12 Version 0.82 Build C (2013 Jan 7)

The following features and bugs were added/fixed in this version.

- Added recognition of .h, .map, .lst, and .report files as 'text files' such that when they are double clicked in the project's 'Reference' folder they open up in the editor. Unrecognized file types are no longer ignored when double clicked. Instead a dialog opens asking the user if they should be opened in the editor.
- Header files (.h) now getting syntax highlighted.
- When loading a project file the eTPU's default type (etpu1/etpu2) and code size are detected and these become the default type and code size passed to the ETEC Compiler.
- Fixed a bug in which the 'Click-To-Line' capability of the Output Window (and other cases) would scroll to the wrong line when the file was being viewed in 'Mixed-Assembly' view.
- Fixed bugs in which a view of a source code file in mixed-assembly mode was not getting saved/restored correctly in the project file and sometimes caused a spurious diagnostic warning. Also, improved text-



selection persistence when toggling between mixed-assembly and source-only view.

- When first non-whitespace character typed in the editor is a '#' (preprocessor) character then auto-align to the left hand side of the editor.
- When selecting a line by holding the END and SHIFT buttons down it seems like the trailing newline characters get included in the selection when they should not.
- Fixed bug in which if the first character typed into an editable file window is a backspace or delete (and perhaps a couple others) then the keystroke is lost, though the run-mode does switch to 'edit' such that the second keystroke is not lost.
- Fixed the toolbar's 'Goto Cursor' button.
- The popup menu that appears when editing a file or state machine now contains the 'Save' option.
- Improved the mouse behavior (selecting text, etc.) on the far-left side of the editable window where the Coverage boxes are seen when simulating.
- If file viewed in DevTool has a completely different path (down to root of drive) display the absolute path, not a (HUGE) relative path.
- Added support and documentation for disabling warning(s) from the command line using the '-ws=,...'
- Fixed issue with MC33816 'Direct Injection' and 'DC-DC Converter' demo not loading into non-licensed computer.
- Changed nomenclature 'Stop' to be 'Pause' instead to avoid confusion. Modified several buttons to match. (Thank you Tim!)
- Right-click on the build batch file should include make/build option(s)
- Fixed issue in which extremely large listings generated by external 'Mk.bat' style builds (I.E. when code is built by an external batch file) could cause DevTool to hang.
- Support 'Click-To-Line' support for listings where a path is 'relative' to the 'CWD'. For example 'Error ..\..\WebApps\SomeFile.c line 37'
- Fixed issue with specifying the .elf file following a creating of a new project. The .elf file cannot be specified
- New project problem. BC build - can't build with external batch file because executable is not set yet, and can't set executable because it hasn't been built yet!
- Fixed certain cases in which a dialog box would appear as a separate program within Windows.
- Added a verify\_version\_ex()' script command for DevTool to check DevTool and ETEC versions. Note that the old verify\_version() script command still works, but generates a warning, AND it checks the equivalent Mtdt Version rather than the new DevTool version which restart at version 1.0.
- On auto-run mode when a command line parameters is used as part of a regression test, ..., if the test is

interrupted and a different project is opened, the command line parameters used initially to launch the regression test are retained and stay in effect, whereas they should have been erased when a different project file was loaded.

- In an eTPU Target's settings, click on the 'Tools Directory' dropdown. Then click (Some Hotkey ???) This causes some sort of race condition and locks up DevTool. Bizarre. Note that this bug was not intentionally fixed (is it really gone?) However, some other changes were made to fix another bug that apparently fixed this bug also. In any case, can no longer reproduce.
- Fixed a bug in which, other than using drag & drop, it was not possible to add files to the reference folder.
- Fixed a crash that can occur when closing a panel. Note that DevTool consists of panels, within panels, within panels. Panels with child panels cannot be closed but this was not being enforced in all cases which could lead to a crash.
- Fixed a bug in which if partway through a simulation a change is made to the project file that should cause a code rebuild, the code doesn't (always) rebuild, thereby allowing the simulation to continue using a stale code build.
- Added an MRU approach to closing windows such that if a window is closed, the last-viewed window becomes active.
- After a re-build, script file needs to load on reset!! (because defines file may have changed)
- Files in the project (vector, script, 'c', .dfi, etc) can be renamed by right-clicking on them.
- Fixed a general weakness in that it was difficult to create and use header files or other generic files with non-supported file extension. This is now supported by both the main menu and project's popup menu.
- Fixed a bug in which files that are opened for viewing (e.g. '.LST' files) that are supposed to automatically be reloaded on a link were not (in certain situations) being automatically reloaded.
- Added ability to generate CodeCoverage, CodeLoad, SymbolTable and ExecutionTrace files from the Files, Write menu.
- Fixed an issue with creating new script files where the file would get created but would not get added to the project. Also there were spurious messages when creating new (template or blank) vector and primary script files that have been fixed.

The purpose of this section is to help invite customer feedback in order to help ASH WARE prioritize development of these features. The following features are planned for future releases.

- Memory Tool not implemented. Memory tool supports things like dumping memory to a dis-assembly or image file. [Workaround: use same feature via a script command, 'dump\_file();' ]
- The 'Help->Register Computer' menu item is missing. [Workaround: Open an explorer window and copy the license file, 'AshWareComputerKey.ack' from the installation directory into your email program (as an attachment) and email to ASH WARE.]
- The 'Threads' Window is not yet supported.
- The 'Memory' window is basic and feature sparse.
- Complex breakpoints (available in Mtdt) have not been enabled in DevTool yet.
- The 'Trace' window (available in Mtdt) has not been supported in DevTool yet.
- CPU32 Call Stack window (available in Mtdt) has not been supported in DevTool yet.
- The Timers are not supported (timers embedded into code addresses)
- Cannot view .HTML files such as the analyses file in the IDE
- It would be nice to view help videos (e.g. Youtube videos) in the IDE.
- Ability to modify certain windows such as the 'Registers', 'Channel Hardware', 'Scheduler', etc. Note that several windows that were editable in the legacy Mtdt Simulator are 'read-only' in DevTool.
- Add auto-messaging in Simulator when error handler is traversed, along with disable-able message in the Messages options.
- Breakpoints can only be Set or Deleted. The Enable or Disable of existing breakpoints is not supported. All breakpoints are Enabled.
- Ability to continue to edit during long builds (currently, an Hour-Glass appears and DevTool is disabled during long builds)
- Support the ability to edit code without ending simulation mode. Does not include ability to compile and run patches on the fly - to get the changes the simulation must be re-started after a re-build.
- A disable-able warning that gets issued when channels have overlapping channel frames
- Verification errors appear in the output window in a click-to-line format.
- Spell check within comments and strings
- Support a separate edit mode and simulation mode IDE configuration. Currently only one mode is supported and no window layout changes when switching between the modes.

The following items are notable changes to DevTool relative to its predecessor, Mtdt.

- The `load_executable()`, and `vector()` script commands have been deprecated. Workaround: use the `#ifdef _DEV_TOOL_` to make script files Dev-Tool specific.
- DevTool command line options generally have an '=' character, whereas Mtdt did not. For example, the following shows how DevTool and Mtdt specify the project file when launched from the command line.

```
DevTool:  
-p=Proj.ETpuIdeProj
```

```
Mtdt:  
-pProj.ETpuSimProject
```

- DevTool automatically records all pin transition data, including the TCRCLK pin. In some cases this could lead to a failure in behavior verification when using previously generated .bv/.ebv files for which the TCRCLK pin logging was disabled. The workaround is disable behavior verification for the TCRCLK pin using the `"disable_ebehavior_pin("_tcrclk")"` script command. Note that script file can be kept compatible between tools by making use of the `#ifdef _DEV_TOOL_ / #endif` keepout.