Newsletter Issue #83, December 2009

Next ICAP/4 Release and DSP Designer

Intro

Since our last newsletter #82 Intusoft has devoted a continued flow of R&D and marketing resource to its revolutionary DSP Designer capability. Several things have been added or completed in 2009. We've also enhanced the upcoming Q1/2010 ICAP/4 release with lots of new features to accommodate DSP Designer, as well as heightening other aspects of the software. More on the enhancements to ICAP/4 can be found in this newsletter.

The block diagram in Figure 1 illustrates a design flow between the DSP development and evaluation board environment, and the Intusoft ICAP/4 PC environment.

In This Issue

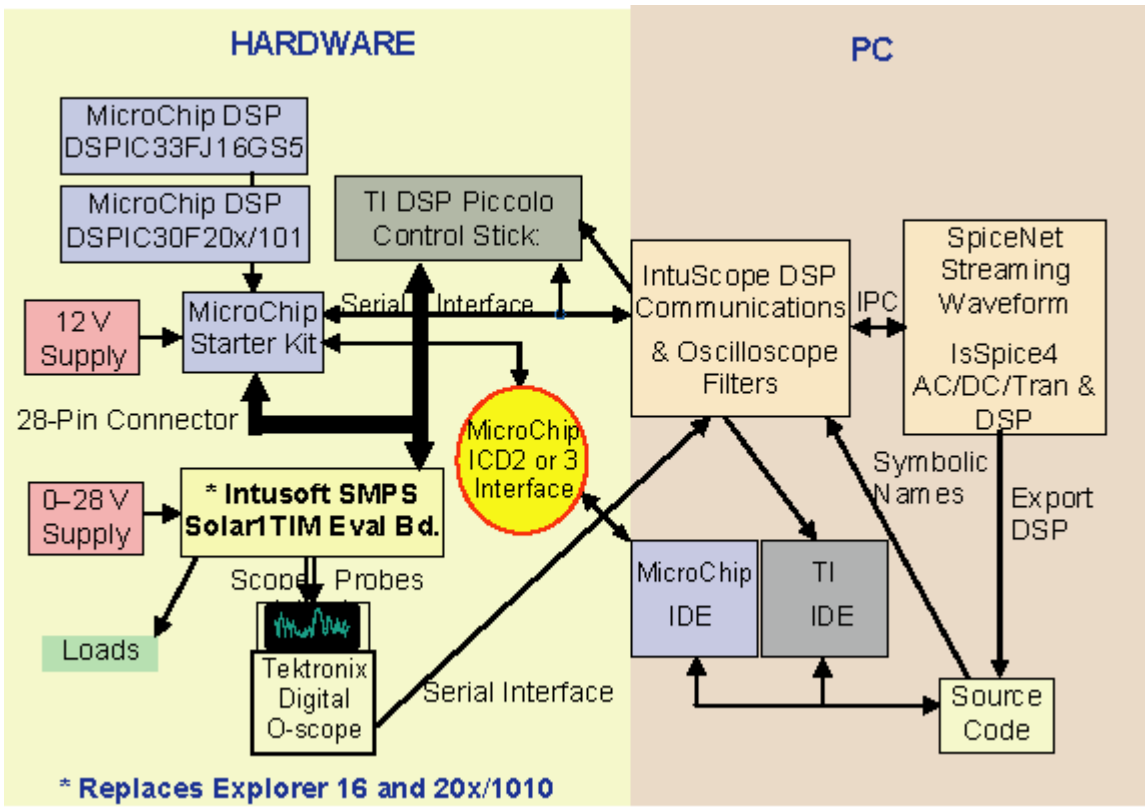


Figure 1, Block of DSP Development and PC Environments

Block Summary

Starting on the left side, the Microchip DSP development board plugs into the MicroChip 16/Pin-28/Bit Starter Kit, which in turn plugs directly into the Intusoft Solar1TiM evaluation board. The Texas Instruments DSP Piccolo stick development system plugs directly into the Solar1TiM PCB. The Microchip starter kit allows the user to develop and validate Microchip dsPIC30F/PIC24F/dsPIC33F devices. DsPIC33F devices include Digital Signal Controllers (DSCs) for common, multi-loop Switch-Mode Power Supplies, including the dsPIC33FJ16GS504 44-pin device shown in the block diagram. The starter board contains a socketed 28-pin DIP, USB port, power supply regulator, crystal oscillator, connectors for the ICD 2 In-circuit Debugger/Programmer and PICkit 2, single UART communication channel via USB bridge, header for access to all device I/O pins, and circuit prototyping area including pads for SOIC and SOT-23 devices. The MicroChip DSPIC33FJ16GS504 DSP Micro-controller is a high-performance SMPS & digital power conversion, 16-bit digital signal controller. It supports applications such as: AC to DC Converters; DC to DC Converters; Power Factor Correction (PFC); Uninterruptible power supply (UPS); Inverters; Embedded Power-Supply Controllers; Circuit Breakers; Arc Fault Detection. More can be found at:

http://www.microchip.com/wwwproducts/Devices.aspx?dDocName=en537165 and http://ww1.microchip.com/downloads/en/DeviceDoc/70318D.pdf.

More on the TI stick can be found at: here

The Solar1TiM PCB is supplied by a 0-28 volt source, as would come from a solar panel in real application. The "Loads" box represents loading for the board. For Microchip, the ICD2 (or 3) Inter-process Communications Device box serves as interface between the Integrated Development Environment (IDE) software loaded on the PC. The TI IDE does the same for the TI Piccolo Stick DSP development system. Series 1000 and 2000 Tektronix scopes are supported and connected to the Solar1TiM board or development board. They feed into a serial plug-in to the ICAP/4 environment to view lab waveforms alongside IntuScope's simulation waveforms. The IntuScope DSP Communications and filters, for DC, assign a waveform vector name associated with a DSP address. A special interface to SpiceNet displays DSP op values using SpiceNet's voltage test point. A DSP "op setup" dialog lets the user select parameters such as refresh rate, display fonts and virtual instrument skin.

The SpiceNet streaming waveform box averages DC data 256 times for high accuracy. The streaming data displays the DC op point using a new simulated 7-segment digital voltmeter in SpiceNet. The user can also have multiple DVMs displaying not only external DC values on the development board, but also internal measurements such as virtual current or duty ratio. This saves a considerable amount of money as DVMs, oscilloscopes and transfer function analyzers cost around $50k. The AC/TRAN/DC "stimulation" of the development board is done by way of the IsSpice4 matrix solution and special scripts in IntuScope. The matrix solution is used to extract coefficients for the difference equations that describe the DSP control system. IsSpice4 produces a matrix that the DSP can understand. For op point, a program (browser) reads this data and code is created for the DSP. AC & TRAN analyses are performed via automated code-generation scripts in IntuScope, which in turn are used to automatically drive (stimulate) the DSP board by way of sine/cosine generators (for AC). Here, the DSP can be turned into a transfer function analyzer to extract a Bode Plot - measuring stability margins. IntuScope in turn can also detect the real and imaginary signal components at each frequency. For transient signal generation and a pulsed signal is connected to a MOSFET on the Solar1TIM PCB as stimulus.

TOP

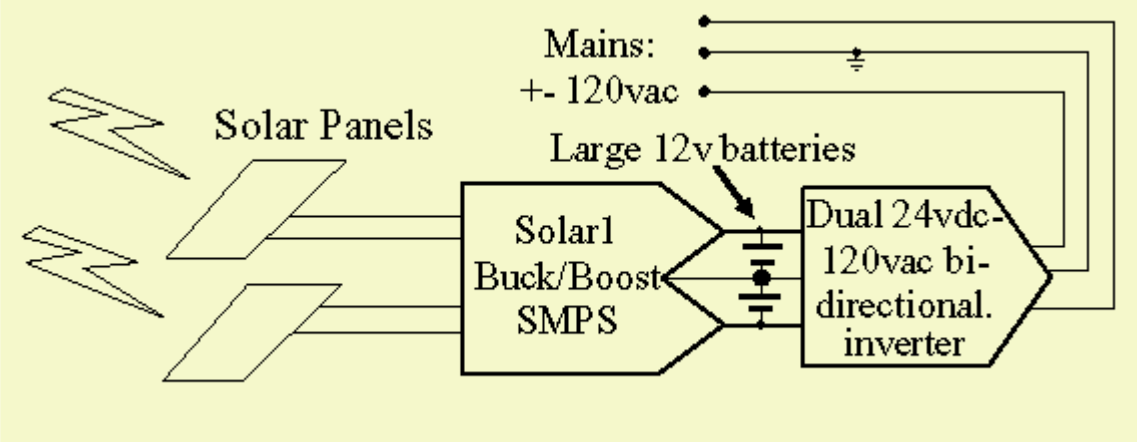
Intusoft's Solar1TiM Evaluation Board

Continuing the introduction in Newsletter #82, Intusoft has now completed its Solar1TiM Evaluation

Board. This serves as a replacement to the former work done on the MicroChip dsPIC30F202x/1010 SMPS Buck development board and the Explorer 16 development board. Now the Microchip 16bit-28pin starter kit plugs directly into Intusoft's Solar1TiM PCB via a 32-pin connector, as does the TI piccolo stick development system. Intusoft has strived to make this a universal evaluation board for multiple DSP providers. Others will be supported in the future.

The Solar1TiM board has a mount provision for >10 amp output loads. The fan is not included with the board. It is controlled "on/off" in software via a test point on the board.

Following is a block representation of the Solar1TiM PCB being used as an SMPS supply for a bi-directional power inverter for solar power application. The system works in conjunction with bi-directional inverters between the large storage batteries and the power mains.



TOP

Marketing Strategy for DSP Designer

Marketing strategy for DSP Designer is ongoing. First, Intusoft presented a technical session that featured the advanced aspects of DSP Designer at the 2009 "PCIM Europe" conference in Nuremberg, Germany. Also, a demonstration of the new capability was performed at the German cluster for power electronics. For 2010 we are arranging for DSP Designer to be showcased in both presentation and exposition venues at PCIM China in 2010.

Secondly, the DSP Designer portal at: http://www.intusoft.com/DSPSimulation.htm is being updated to include new developments, technical articles and slide presentations.

Other marketing-development measures include continued effort to foster collaborations with MicroChip and Texas Instruments. Intusoft will also be providing its first DSP Designer seminar near Intusoft's corporate headquarters, immediately following the APEC 2010 convention in Palm Springs, California. This is an intensive 1-day applied seminar covering DSP Designer operation and application from a technical level, including demonstration of its interaction with the Solar1TiM PCB and DSP development boards. Other seminar venues will be explored in 2010.

Finally, in 2010 we will be attempting interface with other DSP providers such as Analog Devices Blackfin processors

TOP

Matrix Solution

The DSP control equations can be expressed using matrix algebra as shown in Figure 3. Assume there are j states that need to be evaluated, with k of them having a delay history. The equations can be arranged as shown with all trivial solutions at the bottom of the matrix. Let Hn be the history value Hn. Then the j+k by j sub matrix at the top will have its right hand side equal to zero. After solving the matrix the Vn values are substituted into the Hn RHS for the next iteration. There may be more states than history because some of the states may include input and outputs. The main diagonal is scaled to be 1 so that there is no divide required in the solution. For large j, the matrix coefficients should be sparse and non zero values should be near the main diagonal. That's equivalent to having a number of blocks with a single input and output cascaded. If the original matrix had non-zero coefficients below the main diagonal, then the matrix solves an algebraic set of simultaneous equations. DSP's can be made to have to all zero values below the main diagonal by judicious use of backward euler integration to break up the signal flow. That has the side effect of adding delays and reducing controller bandwidth.

LU decomposition, following the forward substitution gives us exactly what's needed. Then backward substitution is a multiply accumulate series for all non zero coefficients followed by division by the main diagonal value. If mixed precision is used, the main diagonal can be normalized to unity; eliminating the division. If integer or fractional scaling is used, the result can be multiplied by a predetermined constant, formed by dividing the scaling value by the main diagonal value, then applying the inverse of the scaling value to the outputs. The solution proceeds from the jth row and j+1 column, summing the products of the non-zero coefficient with their associated states. An array of coefficients is made in the order they will be used and a corresponding array of state-pointers can be made to make maximum use of the DSP multiply accumulate capability.

```
const int16   coef[numRowCoef];
iInt16 * varptr[numRowCoef];
int16 Vn;
while (numRowCoef--)
      Vn += *Coef++ *   *(*varptr++));
```

C compilers will figure this out; but there's always hand coded assembly language to fall back on.

For Reduced Instruction Set Computers, RISC, it may be necessary to limit the range of variable index change from one computation to the next. This can be accomplished by moving the rows with H coefficient up until they are just below the first coefficient used in that column, and the moving the column left to place the unit value on the main diagonal. Such a movement doesn't change the Lower triangle zero condition; but it tends to cluster coefficients along the main diagonal. Then the varptr usage shown above is replaced as shown below:

```
const int16  coef[numRowCoef];
iInt16  offset[numRowCoef];
int16 *varptr;
int16 Vn;
while (numRowCoef--)
       Vn += *Coef++ *  *(varptr + *offset++);
```

This form may need some adjustment depending on how the C compiler does its optimization. If the user identifies states that need a solution, then unwanted states can be eliminated by matrix manipulation. That reduces the number of MAC initializations and result storage; making a faster solution.

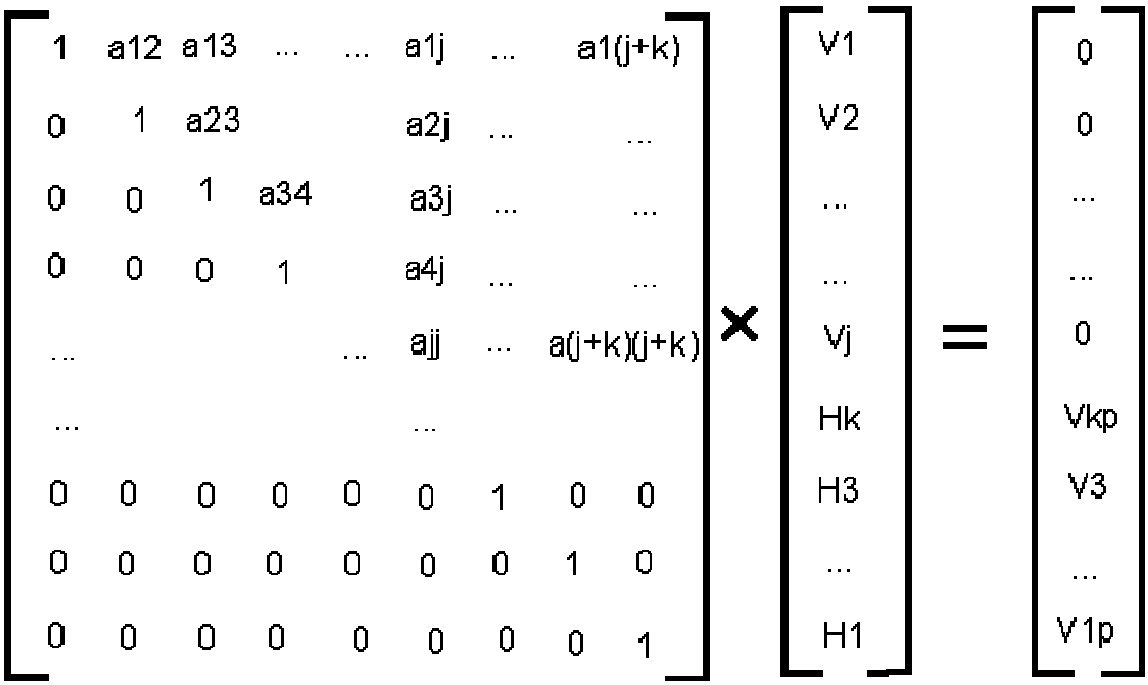


Figure 3, A matrix solution has RHS(0 thru j)=0

[TOP]

Comparing Microchip *dsPIC33FJ16GS504* and Texas Instruments Piccolo, *TMS320F28027* Digital Signal Processors for Switch Mode Power Supply (SMPS) Applications

DSP Architecture

In general, both Texas Instruments and Microchip architectures have optimized the use of their respective processes to achieve a low cost, low power solution for SMPS applications. Clearly, it's possible to make a Silicon solution that runs much faster even using 64 bit floating point to avoid scaling issues. But that approach scales up cost and power leaving the target market behind. Cost and power are minimized as word length and processor speed are reduced. That places a premium on design ingenuity to compete in this market place. Most contemporary low cost power electronics are produced using single sided PC boards. SMT technology is just emerging for these applications. Yes, that's way behind the technology we'd like to use, however, price competition is a major consideration. DSP based designs are on the threshold of breaking into the low cost SMPS market. Both Microchip

and Texas Instruments have a number of modules that perform a variety of peripheral operations like serial I/O; A/D conversion, pulse width modulation... Over time, nearly anything that used to be an external part has been drawn into the DSP so that very few peripheral parts need to be added. The PWM outputs still need drivers to interface with the power MOSFETS and there's the need for interface buffers for communication with the outside world.

Texas Instrument Piccolo DSP: The Piccolo DSP has more RAM than Microchip. It takes advantage of the added RAM to run its program code from RAM for faster execution. TI uses a pipelined architecture, requiring a full pipeline to achieve throughput equal to its clock speed. The actual execution speed using a 60 mip chip is very close to the Microchip 40 mip chip.

The Piccolo A/D converter is 12 bits wide and has an effective accuracy of about 10.6 bits. Noise appears to be normally distributed so that accuracy is improved by averaging many samples, thus increasing accuracy by the square root of the number of samples.

Microchip dsPIC33 DSP: Microchip has divided the RAM into XRAM and YRAM for the purpose of implementing the MAC instruction. This sets on top of an MPU architecture borrowed from the original Intel 80xxx series. The XRAM, YRAM approach tends to make all memory look like registers; however, they still retain a preferred set of registers used for indirect addressing and multiplier, multiplicand storage.

Microchip's A/D converter is 10 bits wide and performs adequately for PWM control, achieving 12 or 13 bits accuracy when averaged over 256 samples.

Instruction set:

The MAC instruction: Both TI and Microchip have special hardware to implement the multiply accumulate, MAC operation. This added hardware is an important DSP feature that reduces the execution time needed to solve difference equations by a factor of about 3 over optimized C compiler execution. The implementation is slightly different, favoring Microchip for fewer clock cycles. The C-compilers for both products fail to use the MAC instruction for the following code, requiring assembly language coding to get the speed benefit. Microchip cannot shift right more than 8 bits; so a left shift (16-radix) is used instead by using the "sac" instruction that saves the upper accumulator word in a 16-bit register. The Microchip MAC, CLR and MOVSAC instructions are complex. After performing the operation, the multiplier and multiplicand registers are loaded from the pointer registers and then the pointer registers can be incremented or decremented by 2, 4, 6. Microchip addressing is byte oriented so that +=2 is equivalent to +=1 for C language or TI Assembly language. After performing a MAC

product, the Microchip pointer registers are several memory locations ahead and the multiplier and multiplicand registers are holding data from the previous pointers. This makes assembly coding for Microchip very error prone. The TI MAC operation sums the previous product, requiring an extra summation at the end. Moreover, the TI shift operation requires loading the T register so that it really requires 2 instructions. The examples below are representative of what's done by the Intusoft code generators. Further optimization may be possible by reusing the TI T and DP registers, combining pointer arithmetic for long sequences of null coefficients and taking advantage of the Microchip increment-decrement range. These added optimizations produce only modest speed improvements for the code examples provided with the DSP Designer software.

C Language

```
Accum = 0;
Accum += *pState++ * *pCoef++;
...
Accum += *pState * *pCoef;
Accum = Accum >> radix;
```

Texas Instruments Assembly (n+4 + [1])

```
ZAPA
MAC P, *+XAR4[i],XAR7++
...
MAC P, *+XAR4[j],XAR7
ADDL ACC, P
MOV  T,  #8
ASRL   ACC, T
```

Microchip Assy (n+2+[1])

```
clr   A, [w8]+=2, w5, [w10]+=2, w6
mac   w5*w6, A, [w8]+=2, w5,
 [w10]+=2, w6
...
mac w5*w6, A, [w8], w5, [w10],w6
sac  a,  #-6 , w0
```

[1] Added instructions needed to reach non-sequential state locations

Moving Data: Both DSP's have mov instructions. TI's syntax is

```
<mov destination, source> ,
while Microchip is
<mov source,destination>.
```

TI can't reach all memory with the mov instruction. So it becomes necessary on occasion to load the DP (Data Page) register in order to get the data ram in the proper page. The <@> symbol identifies the addressing method as DP relative. This is not very serious since most DSP instructions rely on registers and pointers that do access the entire RAM.

C Language

Accum = DSP_RAM_VALUE;

Texas Instruments

MOVW DP, DSP_REGISTER_BASE
MOV AL, @VALUE_OFFSET

Microchip

MOV DSP_REGISTER_VALUE, w0

Limiting Data: Keeping data within established limits is required to prevent arithmetic overflows. TI has MIN and MAX instructions that aid in limiting. However, each limit for TI takes 2 instructions while Microchip requires 3 instructions. HI and LO limits are usually required for each state variable, making the limit operations as costly as the MAC operations.

C Language

Accum > Hi ? Accum = Hi : Accum;
Accum < Lo ? Accum = Lo : Accum;

Texas Instruments

```
MOV  AH, #HI ; biggest input in AH
MIN  AL, AH  ; if AL > AH, AL=AH
MOV  AH, #LO ; smallest input in AH
MAX  AL,  AH    ; if AL < AH, AL=AH
```

Microchip

```
mov    #HI,w2 ; biggest input in w2
cpsgt  w2, w0
; skip if biggest < measured
mov    w2, w0   ; set lim val
mov    #LO,w2   ; smallest in in w2
cpslt   w2,  w0
; skip if smallest > measured
mov     w2, w0   ; set lim val
```

Comparative Test Results

Each DSP was connected to the Intusoft Solar1TiM evaluation board. The 2 vendor DSP evaluation boards plug directly into connectors provided on the Intusoft evaluation board. Software for initialization and memory control was written in the C programming language. Interrupt service routines for the sine-cosine generator, the MPID control law and the Plant model were done in assembly language. In general, the C compilers worked well when the MAC capabilities were not required. Scaling used a mixed integer/fractional word. The binary point location was optimized for the problem at hand. For example, the sine-cosine generator used a 1.14 representation, where the number to the right is the number of binary bits in the fractional part and the left side accounts for the integer bits excluding the sign bit. The MPID algorithm for TI used 7.8 scaling and for Microchip 5.10 scaling was used. MPID stands for Modified PID (Proportional Integral, Differential). Modifications are for limiting that provides automatic soft-start behavior. The control algorithms for the sine-cosine generator was hand crafted. Both the MPID and Plant models used DSP Designer Assembly language code generators

The Sine-cosine generator is used to generate the excitation signal for transfer function analysis and for the in-phase and quadrature signals used for signal detection.

An advanced virtual current controller used the plant model to estimate inductor current using a Kalman Filter approach and closes the "differential" loop using the average inductor current. The estimated current is also used to provide current limiting; thereby eliminating the need for current sense components.

Timing was measured using an oscilloscope to measure the length of marker signals that were output at the beginning and end of each task.

Vendor	Microchip	Texas Instruments

Function		
Sine/cosine	1.44usec	0.96 usec
MPID	1.72usec	1.24usec
Plant	3.44usec	2.88usec
Total	6.60usec	5.08usec

Table 1, Timing Comparison

While the total in table 1 has no meaning, it's a convenient way to measure the "average" performance difference. That turns out to be about 30% in favor of Texas Instruments or about ½ the advantage one would predict based on clock speed. Figure 4 shows how the 2 loops respond to a step change. The "noise" is the switching ripple, which can be seen because the load switch is synchronized to the PWM clock. There is no noise in the simulation because average modeling techniques were used. The difference in test vs. simulation is due largely to capacitor modeling.

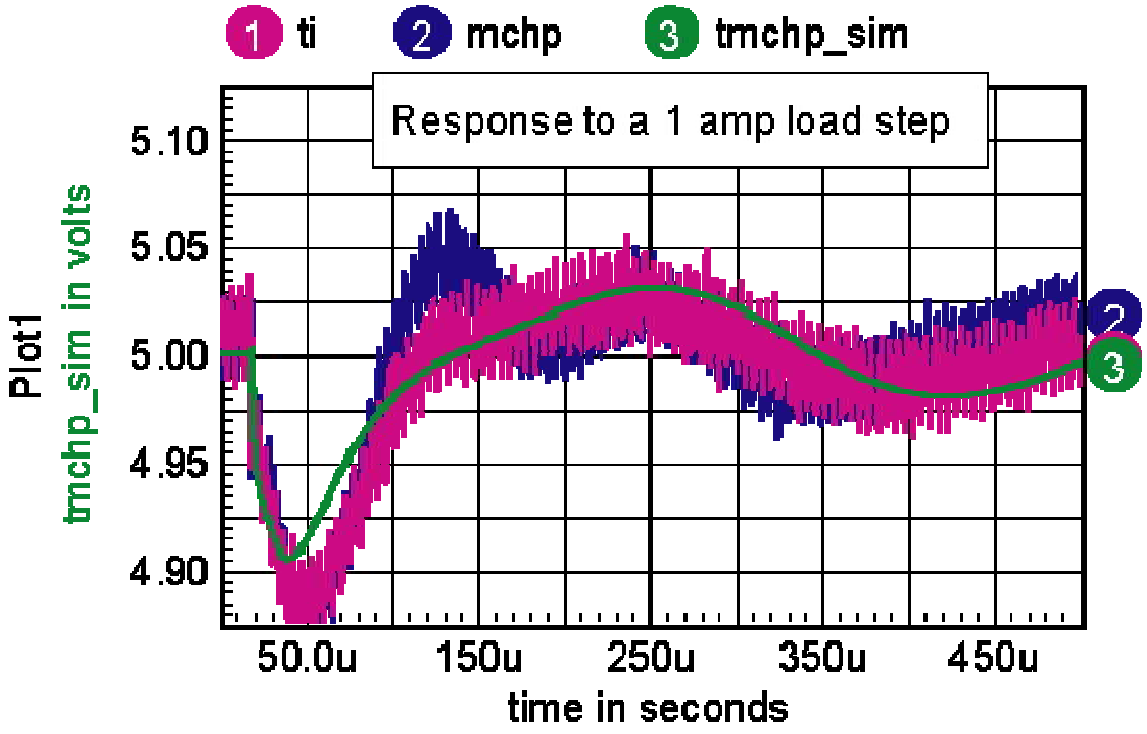


Figure 4 , Comparing MPID response

Working with the Development Environment

Both TI and Microchip offer a free suite of tools (C-compiler, assembler and Integrated Development

Environment, IDE). They both would like you to buy their respective C-Compilers. TI will not provide technical support if you don't send them money and Microchip disables C-Compiler optimization unless you pay them. They must make the basic compiler free because they have borrowed the code from public licensed software that prevents them from selling it.

Texas Instruments IDE: This IDE is quite similar to the Microsoft Visual Studio. Using the same keystrokes for Run (F5), Build (F7) and set breakpoint (F9). You can set multiple break points and the debug window allows arithmetic expressions. Running your code requires a successful compile and connect to the debugger. Then you must load the program from the proper ".out" file locate in the release or debug folder. The code is loaded into RAM so that it's volatile. It's recommended that you go to main before beginning a session. To do it again you need to reset the DSP, recompile, load program, go main and run. Finding text uses the familiar Microsoft Visual Studio method with a find text box and up-down arrows in the toolbar.

The hardware interface and IDE are supplied with the Piccolo Control Stick, TMX20F28027 for $39.00. That includes the DSP but there is no auxiliary serial interface. Intusoft provides this interface on the Solar1TiM evaluation PCB ($200). Examples are well hidden in the TI website, but with some persistence and Google search, they can be found. The Intusoft Solar1TiM PCB comes with working examples and TI documentation. The Piccolo Control Stick is difficult to handle without touching the IC's making it less robust than the Microchip evaluation board.

Microchip IDE: This IDE isn't as full featured as TI. You can only set 2 break points and the function keys are quite different. Run uses F9 and Halt-Rest use the F5-F6 combination. You can add to the watch window only using pull down list selections. Debug arithmetic isn't allowed. To find text, you must enter it in the "Find" dialog and use F3 to find again. To run your software, you must open the IDE and connect to the debugger using a debug menu selection. Then after compiling, you load the DSP code using a debug menu selection. If you're code accesses the clock, you are informed that you must turn power to the DSP off and on again. Note that Microchip used flash memory so the powering down the DSP doesn't loose the code.

There are 3 pieces of hardware needed to run your code. First is the evaluation board that contains the DSP. The lowest cost version is the 16-bit 28-pin starter, DM300027 for $79. At his time it doesn't ship with the DSP, dsPIC33F16GS502-I/PT, which is available for about $5 from Microchip or Digi-Key. Then you need the MPLAB ICD 2, the In Circuit Debugger interface DV164005, selling for $199.99 or the recently released ICD 3, DV164035 for $189.99. Finally you need an SMPS evaluation card, DSP Designer supports the Solar1Tim PCB for $200. You'll need the C30 complier, which can be had for free, but the optimization goes away after 60 days. The Microchip evaluation board has proven to be

very robust, even when you plug the DSP in backwards.

Improvements

Here are some things that would make for a better user experience if they were corrected.

Microchip:

1. Only 2 breakpoints
2. Non Standard hot keys (F9 vs. F5...), an option to use the Microsoft Visual Studio hot key definitions is desirable.
3. No search toolbar.
4. Assembly breakpoints are inaccurate, code stops 2 instructions after the break, requiring insertion of "nop" instructions in some cases
5. Watch window doesn't allow math expressions
6. Can't use C/C++ header in assembly code.

Texas Instruments:

1. Restart is time consuming, requiring reset and reload; macro language is not a substitute because macros go away for new projects and updates.
2. Address limitation, only 7-bit data page.
3. Lower speed timer interrupts occasionally get skipped
4. C/C++ headers show up as nonsense (C,NOLIST) in project includes when using #include .cdecls, C,NOLIST,"cinfigure.h")
5. Unexpected location of variables (not in same order as declarations)... makes use of DP offset impossible for mixed C/C++ Assembly programming without resorting to the .map file which may change with future releases.
6. Production package size is more expensive to use for low cost applications.
7. Avoidance of using flash memory because RAM coding makes for faster execution

Both:

1. The MAC instruction in not used by the C/C++ compiler
2. Make/Build is not recognizing include files in assembler files.
3. Difficult to find technical documentation…incomplete with product download.
4. It would be desirable for copyright notice's to clearly assign sample code to the public domain.
5. The ".map" file does not expand struct members.

CONCLUSION

Both TI and Microchip DSP's can be used for low cost SMPS controllers. The following features and design techniques combine to make a DSP solution for SMPS design superior to using analog IC's.

1. The added capability of a DSP allows for tradeoff between execution speed and external hardware such as current sense components. Using these virtual components saves production cost.
2. An assembly code generator simplifies the design process, allowing the engineer to easily tradeoff control concepts.
3. The DSP can contain built-in test to measure its own transfer function and other properties that correspond to the SPICE simulator's AC, DC and TRAN simulations. Engineers gain confidence in a design by making the SPICE model agree with the hardware test data.
4. Moreover, using average models in a SPICE simulation yields faster than real time simulation results and accounts for nonlinear behavior encountered in startup and overload conditions.

TOP

DSP Features for Q1/2010 Release

DSP Package

Software, Hardware, Interface

DSP Designer includes the following items:

1. Intusoft's Solar1TiM Dual Buck SMPS Board
2. Plug-in interface for TI Piccolo MCU
3. Plug-in Interface for Microchip 16-bit x 28 pin Starter Board
4. Interface adapter for any DSP
5. Real Time AC, DC and TRAN DSP interface
6. DSP Designer Getting Started Guide

SpiceNet

New DSP Analysis

A .DSP Analysis in invoked be selecting one of the options in the File menu, "ExportDSP...". The purpose of this selection is to generate code (DC, AC, TRAN) for the selected DSP.

New DSP Layers/Configurations

A DSP controller can be imbedded in SpiceNet using configurations and new layers added to SpiceNet. One new layer contains interface voltage sources and node names. The other has the z-transform controller portion of the schematic. A new configuration is created by the user, which includes interface parts for exporting signals to the DSP board.

DSP State Variables

State variables use the SpiceNet node names, importantly used in setting up the IsSpice4 matrix for code generation. ZDELAY element node names must have a "zi" prefix for inputs and a "zo" prefix for outputs. Outputs that control pulse width modulators have an embedded "pwm#" name, where "#' identifies the hardware PWM number.

New DSP Paramters

When the .DSP analysis is run, special parameters evaluated by SpiceNet are placed in a ".par" file. Parameters include:

T: The sample time
ref# : Reference voltage for "ind#" input
radix : Number of fractional bits
LIMIT_HO_+"node" : Low limit
LIMIT_LO_+"node" : High limit

Parameters are also used to calculate control coefficients so you can easily change things like bandwidth, plus calculate A/D converter and PWM output scaling.

New Digital VOM Meter

A new interface in SpiceNet displays DSP op values in the form of a virtual instrument using SpiceNet's voltage test point. If the IntuScope setup dialog contains the vectors node name, then it will be retrieved and displayed as a virtual instrument, complete with the units provided in the test point parameters. This is equivalent to replacing multiple DVM's on a workbench. Various parameters, including the refresh rate, display fonts and virtual instrument skin, can be selected using the DSP op setup dialog.

IntuScope

New AC and TRAN Scripts

New DSP AC and TRAN scripts are added in IntuScope's Calculator to display step load transient results and control system bode plots in hardware from the DSP Development board. An interface to

Tektronix Oscilloscopes lets you view/compare results for observable nodes. This includes nodes that are unobservable because they are mathematical representations inside the DSP control system! For AC, new sin/cosine generator scripts have been added to InsuScope. From these automatically driving the DSP, the DSP can be turned into a transfer function analyzer and used to extract a Bode Plot and stability margins.

New DC Capability for the DSP

Press "Browse" in the "Add Waveform" dialog and select "DSP Communication Filter." If the DSP is running, pressing "view all" in IntuScope displays DC average values in the Output window. The new special interface in SpiceNet displays DSP op values.

IsSpice4

Templates for DSP initialization

It is assumed the main portion of the DSP code is in the C programming language. Special templates are provided with DSP Designer include this code along with code necessary for Real Time Communication, all which add the appropriate initialization to the C code, along with any special code to detect faults and assist with startup.

New Code Generator

The IsSpice4 code generator requires that certain rules be followed, so several "ExportDSP..."" rule check errors and warnings have been added:

TOP

Additional ICAP/4 Improvements for Q1/2010 Release

- You can now use the find dialog without the mouse; just use the <Ctrl F> keys and type in the names. The Tab key switches the focus between the Part List Box and Node List Box. The up/down arrow keys select the next or previous items. Typing the name of a part or node selects the item in the list.
- Test points have been added to the find dialog's Part List, eliminating the need to search for them.
- B-elements have been modified by adding a check box that will link comments on the schematic to the expression. If you change a comment on the schematic it will change the expression, and vis versa.
- If you bring up .CIR, .ERR, .OUT, or .PAR files in IsEd, make a change in your circuit, and

re-run the simulation, IsEd will ask if it is ok to reload each opened file. A new option enables you to automatically reload all opened files in the future. To the Options menu was added the ability to return back the reload file dialog, in case it was checked for not showing. The new option removes the dialog box that asks if it is ok to load each individual file.

- A "Print All Pages" option was added to the Print Preview toolbar to print all drawing pages instead of each one individually. New text indicates what page you're viewing.
- In IsEd font format (font type, size, etc.) can be saved with the file, rather than only using the default setting.