Application Report
SPRA967E– January 2005

TMS320C6412 Power Consumption Summary

Ivan Garcia — *DSP C6000 Hardware Applications*

ABSTRACT

This document discusses the power consumption of the Texas Instruments TMS320C6412 digital signal processor (DSP). Power consumption on these devices is highly application dependent, so a spreadsheet is provided to model power consumption for a user's application. To get good results from the spreadsheet, realistic usage parameters must be entered. The low core voltage and other power design optimizations allow these DSPs to operate with industry-leading performance, while maintaining a low power-to-performance ratio.

The data presented in this document is actual measured power consumption for the C6412 silicon revision 1.0.

This application report contains a spreadsheet that can be downloaded from http://www.ti.com/lit/zip/SPRA967.

Table 1. Typical Activity

Power Per Frequency (mW/MHz)(1)				Power at Frequency (W)(2)		
Device	**Core Voltage**	**Frequency**	**CPU and L1**	**Internal Logic**	**IO**	**Total**
C6412	1.4 V	600 MHz	0.93	1.25	0.65	1.90
C6412	1.2 V	500 MHz	0.66	0.75	0.55	1.30
C6412	1.4 V	720 MHz	0.93	1.50	0.65	2.15

(1) Assumes 60% CPU utilization
(2) Assumes 60% CPU utilization, 50% EMIF utilization (133 MHz for 1.4 v, 100 MHz for 1.2 v), 50% writes, 64-bits, 50% bit switching, 2 - 2MHz McBSP at 100% utilization, and 2 - 50MHz Timers at 100% utilization

Contents

1 Activity-Based Models 2
2 Using the Power Estimation Spreadsheet 3
3 Using the Results 4
4 Example 5

List of Tables

1 Typical Activity 1

1 Activity-Based Models

TMS320C6412 power consumption can vary widely depending on the use of on-chip resources. Thus, power consumption can not be estimated accurately without an understanding of the components of the DSP in use and the usage patterns for those components. By providing the usage parameters that describe how and what on the DSP is being used, accurate consumption numbers can be obtained for power supply and thermal analysis.

This model breaks down power consumption into two major components: baseline power and activity power. Using this model, different applications that use the DSP differently can get accurate predictions, all across the spectrum of possible power consumption on the C6412 devices.

1.1 Baseline Power

Baseline consumption is power that is consumed that is not dependent on chip activity. This includes things like static power, PLL power, and clock tree power. While independent of activity, baseline power is dependent on the device operating frequency, voltage, and temperature. Thus, the user can affect baseline power only by changing the device operating frequency (through the CPU frequency), the CVDD voltage, or the operating temperature.

1.2 Activity Power

Activity consumption is power that is consumed by active parts of the DSP (the CPU, EDMA, peripherals, etc.). Activity power is independent of temperature, but dependent on voltage and activity levels. Activity power is separated by the major modules of the device, so that their contribution can be measured independently of each other. This helps with tailoring power consumption to specific applications. The parameters used to determine the activity level of a module are frequency, utilization, read/write balance, bus size, and switching probability. Module activity power includes necessary EDMA service for peripherals that require it. Note that not all parameters apply to all modules.

- *Frequency* is the operating frequency of a module or the frequency of external interface to that module.
- *% Utilization* is the relative amount of time the module is active or in use versus off or idled.
- *% Write* is the relative amount of time (considering active time only) the module is sending data out of the DSP versus reading data into the DSP.
- *Bits* is the number of data bits being used in a selectable-width interface.
- *% Switch* is the probability that any one data bit will change state from one cycle to the next.

1.3 Modules

The C6412 power estimation form contains the following modules with adjustable parameters

- CPU
- EMIF (includes AECLKOUT1)
  - AECLKOUT2
  - CLKOUT4
  - CLKOUT6
- McBSP 0
- McBSP 1
- Timer 0
- Timer 1
- EMAC
- I2C0
- HPI
- PCI

EDMA is not listed as a separate module because the module activity models include necessary EDMA service. The peripheral configuration is documented in *TMS320C6412 Fixed-Point Digital Signal Processor* (SPRS219).

2 Using the Power Estimation Spreadsheet

Using the power estimation spreadsheet involves simply entering appropriate usage parameters. Cells that are designed for user input are white in color. To use the spreadsheet, simply:

1. Choose the appropriate operating voltage for your desired part.
2. Enter the case temperature for which you want to estimate power.
3. Choose the boot time peripheral selection mode: HPI32, PCI, or EMAC (HPI disabled).
4. Fill in the appropriate module use parameters.

The spreadsheet will take the provided information and display the details of power consumption for that configuration.

2.1 Choosing Appropriate Values

The frequency and bits user values are determined by design and it will be clear what the correct values to enter are. Modules that are completely unused and unclocked should use 0 for frequency to prevent any idle power from that module from being included. The utilization, read/write balance, and bit switching require estimation and a good understanding of the user application to choose appropriate values.

2.1.1 Utilization

For modules except CPU, utilization is simply the percentage of time the module spends doing something useful, versus being unused or idle. For these peripherals, there are not various degrees of use, so the value is just an average over time. For example, the EMIF performs reads and writes one-quarter of the time and has no data to move for the other three-quarters of the time (though it continues to perform background tasks like refresh), this would be 25% utilization.

For peripherals with IO, utilization can be estimated by comparing used bandwidth with theoretical maximum bandwidth. If, for example, an application must transfer 160 Kb/s via the I2C port using the Fast Mode, with a theoretical 400 Kb/s maximum, the I2C port utilization would be about 40%.

The CPU utilization is not as straightforward, because there are varying degrees of use for the CPU. Here, 0% utilization means the CPU is idle and does no useful work, where 100% utilization means all 8 functional units are active every cycle and the maximum amount of data is brought in from L1P and L1D every cycle. Few DSP algorithms will achieve 100% utilization, because this requires everything to be used every cycle, with no stalls. Even DSP-intense applications do not spend all of the time in such highly parallel loops. Time is typically also spent executing control code or less demanding DSP algorithms. These types of code may execute only a few instructions in parallel and significantly reduce the IO of the CPU, and thus reduce overall utilization. Thus the balance of CPU use for the application must be considered, and entering 100% utilization is not practical for real applications.

For example, an application that executes control code (estimated at 25% of CPU capability) half of the time, and very dense DSP code (estimated at 90% of CPU capability) the other half, would have an average utilization of about 60% (25% x 50% + 90% x 50%). If the balance were changed to 25% control code and 75% DSP code, the weighted average would be about 75% utilization (25% x 25% + 90% x 75%). If the 25%/75% ratio is kept, but the DSP code does not fully use all the CPU resources (estimate now at 75% of CPU capability) then the overall utilization returns to about 60% (25% x 25% + 75% x 75%). Using estimates of intensity and duration of blocks of code in the application, an estimate of the overall CPU utilization can be obtained.

System level issues may also reduce utilization. Though the spreadsheet will accept 100% utilization for all peripherals, this is not possible in reality. As memory and EDMA bandwidth is consumed, peripheral activity is throttled back due to these bottlenecks and therefore do not achieve 100% utilization. In applications with a lot of memory and/or EDMA usage, individual module utilization numbers should be entered keeping this overall limitation in mind.

2.1.2 % Writes

Peripherals that move data out of the DSP as much as they move data into the DSP have 50% writes (the spreadsheet will assume the remaining 50% of the time is spent on reads). In some applications, peripherals move data in only one direction, or have a known balance of data movement. In these cases, % writes should be changed to 0%, 100%, or the known ratio as appropriate for the cases when the DSP is reading all the time, writing all the time, or a combination of the two, respectively. Otherwise, 50% is a typical number that should be used.

2.1.3 % Switching

Random data has a 50% chance any bit will change from one cycle to the next. Some applications may be able to predict this chance using some a priori information about the data set. If there is a property of the algorithm that allows prediction of the bit changes, the application-specific probability can be used. All other applications should use the default number of 50%.

2.2 Peripheral Enabling and Disabling

The C6412 device offers power savings features by disabling peripherals when not in use. The power estimation spreadsheet provides the options of selecting either the HPI (32-bit), PCI, or the EMAC (HPI disabled) peripherals, which can be enabled at device reset.

Furthermore, peripherals that can be enabled after device reset can be enabled on the power estimation spreadsheet such as the McBSP1, McBSP0, and I2C0, by entering a frequency greater than 0. For instance, if a given application does not enable McBSP1, then 0 must be entered in the corresponding frequency field for McBSP1. Otherwise, McBSP1 will be enabled.

For more information on peripheral enabling, refer to *TMS320C6412 Fixed-Point Digital Signal Processor* (SPRS219).

2.3 Power Down Modes

In addition to the power consumption for the supplied parameters, the power calculation spreadsheet gives the power consumption for the CPU power down modes. Note that the PD1 and IDLE power down modes include consumption by idle EMIF.

2.4 Graphs

The graphs page contains graphs that provide a visual breakdown of power consumption. Shown is a comparison of active power (based on the parameters supplied) and the power down modes, and pie charts showing the relative contributions of each module to the core and IO power consumption.

3 Using the Results

The results presented by the spreadsheet are based on measured data for revision 1.0 silicon. The measured units were selected to be at the maximum end of power consumption for production units; no production units will have average power consumption that exceeds the spreadsheet values. The spreadsheet data may therefore be considered maximum average power consumption. That is, transient currents may cause power to spike above the spreadsheet value for a small amount of time, but over a long period of time, the observed average consumption will be below the spreadsheet value. Thus, the spreadsheet value may be used for board thermal analysis and power supply design as a maximum long-term average.

3.1 Adjusting IO Power Results

IO Power is dependent not only on the DSP and activity, but also the load being driven. For loads with CMOS inputs, the power required to drive the trace dominates, and is a better measure of load than number of inputs or lumped load capacitance. For the data presented in the spreadsheet, the EMIF (with serial termination), McBSP, and Timer interfaces were loaded with approximately four inches of 50 ohm trace. The HPI and PCI were loaded with approximately three inches of 50 ohm trace, with no termination. If the target system has very different IO loading, the spreadsheet results may be scaled either up or down to compensate. For this reason, the spreadsheet allows the user to specify the approximate load on the IO pins for each module. This parameter is used to adjust the reported IO power numbers.

4 Example

Here is an example demonstrating how to choose appropriate values for a particular application. The values used in this example may be imported into the spreadsheet by clicking the appropriate macro button.

4.1 Sample Application

In this example, the CPU is running a particular algorithm approximately 40% of the time to process the data. The rest of the time the processor is performing non-optimized background tasks. The device is communicating with a microcontroller through the I2C at a rate of 350Kb/s. The EMAC is actively writing data 70% of the time at 100Mbps. Furthermore, the application requires external buffer reads at 5.25 MB/s and writes at 10 MB/s via the EMIF. The DSP (GDK package) is running at 1.2V, 500MHz CPU with the HPI enabled, at 60C. The EMIF is running at 100MHz, with a 64-bit bus. All other peripherals are disabled, if applicable. EMAC is selected at device reset.

- Voltage: 1.2v
- Temp: 60 C
- EMAC enabled at reset. HPI and PCI are disabled.
- CPU: 500 MHz, 35% utilization. CPU utilization is based on the following factors:
  - Optimized DSP algorithm: 75% of CPU capability for 40% of the time.
  - Background: 10% of CPU capability for 40% of the time
- EMIF: 100 MHz, 10% utilization, 55% writes, 64 bits, 50% switching
  - External Buffer Writes: 10 MB/s
  - External Buffer Reads: 8.5 MB/s
  - Total: 18.5 MB/s (190 MB/s max for 16-bit EMIF), 10 MB/s of which are writes
- AECLKOUT2: 0 MHz, 0% utilization (AECLKOUT1 is included with the EMIF activity)
- I2C: 350Kb/s, 100% utilization
- EMAC: 100Mbps, 70% utilization
- All other modules use 0 MHz, 0% utilization

Entering these values into the spreadsheet gives us a maximum power consumption of about 715 mW core, 310 mW I/O, for a total of 1025 mW.

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale (www.ti.com/legal/termsofsale.html) or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265