

TMS320C6410/13 Power Consumption Summary

Gustavo Martinez

DSP Hardware Applications

ABSTRACT

This document discusses the power consumption of the Texas Instruments TMS320C6410™ and TMS320C6413™ digital signal processors (DSPs). Power consumption on C64x 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 and in the accompanying spreadsheet was measured from units at the maximum end of power consumption for production units; no production units will have average power consumption that exceeds the spreadsheet values. Thus, the spreadsheet values may be used for board thermal analysis and power supply design as a maximum long-term average. All power measurements were taken from silicon revision 1.1 units of the TMS320C6410 and TMS320C6413.

This application report contains a spreadsheet that can be downloaded from <http://www.ti.com/lit/zip/SPRAA59>.

Table 1. Typical Activity

Device	Core Voltage	CPU Frequency	Power Per Fre-	Power at Frequency (W) ⁽²⁾		
			quency (mW/MHz) ⁽¹⁾	CPU & L1	Internal Logic	IO
C6410	1.2V	400 MHz	0.58	0.57	0.44	1.0
C6413	1.2V	500 MHz	0.58	0.69	0.44	1.1

⁽¹⁾ Assumes 60% CPU utilization.

⁽²⁾ Assumes the following conditions: 60% CPU utilization; 100 MHz EMIF at 50% utilization, 50% writes, 32 bits, 50% bit switching and ECLKOUT1 enabled; two 2-MHz McBSP at 100 utilization; two 50-MHz Timers at 100% utilization; device configured for HPI32 mode.

Contents

1	Activity-Based Models.....	2
2	Using the Power Estimation Spreadsheet	3
3	Using the Results.....	5
4	Example.....	5
5	References	6

List of Tables

1	Typical Activity.....	1
---	-----------------------	---

1 Activity-Based Models

TMS320C64x power consumption can vary widely depending on the use of on-chip resources. Thus, power consumption cannot 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 C6410 and C6413 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.

C6410 and C6413 devices provide the capability to disable peripherals that are not being used. When a peripheral is disabled, its clock is turned off thereby reducing the power consumption of the device. The spreadsheet that accompanies this application report includes a field from which the state (enabled or disabled) of peripherals with this feature can be specified. The device-specific data manual should be consulted for more information on disabling peripherals on these DSPs.

1.3 Modules

The C6410/13 power estimation spreadsheet contains the following modules with adjustable parameters:

- CPU
- EMIF
 - AECLKOUT1
 - AECLKOUT2
 - CLKOUT4
 - CLKOUT6
- HPI
- McBSP0
- McBSP1
- McASP0
- McASP1
- I2C0
- I2C1
- Timer0
- Timer1
- Timer2

EDMA is not listed as a separate module because the module activity models include necessary EDMA service. For available peripherals and peripheral configuration, please refer to the device-specific data manual.

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:

- Select the case temperature for which you want to estimate power
- Specify the input clock frequency for the device (this is used for McASP power estimation since this modules run off the DSP input clock)
- Choose the HPI configuration selected at boot time: HPI32, HPI16, or HPI disabled
- Fill in the appropriate module use parameters

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

As the spreadsheet is being configured, the settings will be checked for conflicts, e.g. peripheral clock frequency out of allowed range, etc. For best results, the information should be entered from left to right starting at the top and moving downward.

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. Peripherals that cannot be disabled through hardware or software but that are completely unused and kept in their reset state should use 0 for frequency to prevent any idle power from that peripheral 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-intensive 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% ($25\% \times 50\% + 90\% \times 50\%$). If the balance were changed to 25% control code and 75% DSP code, the weighted average would be about 75% utilization ($25\% \times 25\% + 90\% \times 75\%$). 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% ($25\% \times 25\% + 75\% \times 75\%$). 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

As mentioned earlier, C6410 and C6413 devices provide the capability to disable peripherals that are not being used. When a peripheral is disabled, its clock is turned off thereby reducing the power consumption of the device.

Some peripherals can be disabled or enabled either through hardware or through software. An example of a peripheral that can be disabled through hardware is the HPI. The HPI can be disabled, enabled in 16-bit mode, or enabled in 32-bit mode at boot time through specific pin settings. On the other hand, the I2Cs can only be enabled or disabled through software.

The spreadsheet accommodates this power saving feature by including fields from which a peripheral can be specified as disabled or enabled.

For more information on peripheral enabling please refer to the device datasheet.

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 assume the EMIF is idle.

2.4 Graphs

The graphs page of the spreadsheet 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 1.1 silicon of the C6410 and C6413. 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 4 inches of 50 ohm trace. The HPI was loaded with approximately 3 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 DSP is being used to process audio data from two sources: a CD audio source and a base band I&Q data source. The DSP receives base band I&Q data at a 40 MHz bit rate using a serializer on a McASP and, after processing the data, transmits processed stereo data a bit rate of 1.536 MHz through another serializer on the same McASP; data size is 32-bits. CD audio data (48 KHz sampling rate) is received and transmitted through the second McASP; data size is 24-bits. The DSP is communicating with a microcontroller through the I2C at a rate of 400Kb/s. Both McBSPs are used to interface to other devices in the system; they are operated at frequency of 12.5 MHz with a utilization of 15%. Furthermore, the application requires buffer writes at 10 Mbytes/sec and buffer reads at 5 Mbytes/sec. The device is running at 1.2V, 500 MHz CPU with the HPI disabled, at a case temperature of 25C. The EMIF is running at 100 MHz with a 32-bit bus. All other peripherals are disabled.

- Voltage: 1.2V
- Case Temperature: 25C
- CPU: 500 MHz, 40% utilization

References

- CPU utilization is based on the following factors:
 - Base band data processing: 75% of CPU capability for 40% of the time
 - CD audio data processing: 20% of CPU capability for 30% of the time
 - Background: 10% of CPU capability for 40% of the time
 - CLKOUT4: disabled
 - CLKOUT6: disabled
 - EMIF: 100 MHz, 10% utilization, 60% writes, 32 bits, 50% switching
 - EMIF utilization is based on the following factors:
 - External Buffer Writes: 20Mbytes/sec
 - External Buffer Reads: 15 Mbytes/sec
 - Total: 35 Mbytes/sec (360 Mbytes/sec average maximum for 32-bit EMIF at 100 MHz), 20 Mbytes/sec of which are writes
 - AECLKOUT1: enabled
 - AECLKOUT2: disabled
 - HPI is disabled
 - McBSP0: 12.5 MHz, 15% utilization, 50% switching
 - McBSP1: 12.5 MHz, 15% utilization, 50% switching
 - McASP: 40 MHz, 100% utilization, 50% switching, 2 serializers
 - Note that McASP0 transmits data at a rate of 1.536 MHz and also receives data at a rate of 40 MHz. Since the power spreadsheet only accommodates for one clock rate and since the power for running an extra serializer at 40 MHz minimal, McASP0 is programmed with two serializers running at 40 MHz.
 - McASP1: 2.304 MHz bit clock rate, 100% utilization, 50% switching, 2 serializers
 - Bit clock rate is calculated as follows: 48 KHz frame sync rate in I2S mode where each frame contains a left channel and a right channel of 24-bit data yields $48K * 2 * 24 = 2.304$ MHz
 - I2C0: 400Kb/s, 100% utilization
 - I2C1: disabled
 - All other modules are disabled or not being used (0% utilization)
- Entering these values into the spreadsheet gives a maximum power consumption of about 631 mW for core and 396 mW for IO, for a total of 1027 mW.

5 References

1. *TMS320C6414/5/6 Power Consumption Summary* (SPRA811).
2. *TMS320C6413, TMS320C6410 Fixed-Point Digital Signal Processors* (SPRS247).

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
Copyright © 2019, Texas Instruments Incorporated