TMS320C6455/C6454 Power Consumption Summary

Martinez, Gustavo *Device Applications*

ABSTRACT

This application report discusses the power consumption of the Texas Instruments TMS320C6455 and TMS320C6454 digital signal processor (DSPs). The power consumption on the C645x devices is highly application-dependent, therefore, a power spreadsheet that predicts power consumption is provided along with this application note. The power spreadsheet can be used for the purpose of modeling power consumption for user applications such as power supply design, thermal design etc. To achieve good results from the spreadsheet, realistic usage parameters must be entered. The low-core voltage and other power design optimizations allow these devices 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 devices at the maximum end of the power consumption for production devices; therefore, no production devices will have average power consumption that exceeds the spreadsheet values. Therefore, the spreadsheet values can be used for board thermal analysis and power supply design as a maximum long-term average.

This application report contains a spreadsheet that can be downloaded from http://www.ti.com/lit/zip/SPRAAE8.

Table 1. Typical Activity

		Power at Frequency (W)[1]		
Core Voltage	CPU Frequency	Internal Logic	IO	Total
1.25 V	1200 MHz	1.76	0.54	2.30
1.25 V	1000 MHz	1.66	0.53	2.19
1.2 V	850 MHz	1.41	0.53	1.94
1.2 V	720 MHz	1.29	0.52	1.81

(1) Assumes the following conditions: 60% CPU utilization; DDR2 at 50% utilization (250 MHz), 50% writes, 32 bits, 50% bit switching; two 2-MHz McBSPs at 100% utilization, 50% switching; two 75-MHz Timers at 100% utilization; device configured for HPI32 mode with pull-up resistors on HPI pins; room temperature (25°C).

Contents

List of Tables

Trademarks

RapidIO is a registered trademark of RapidIO Trade Associations.

All other trademarks are the property of their respective owners.

1 Activity-Based Models

The power consumption on the C645x DSPs can vary widely depending on the use of on-chip resources. Therefore, the 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 power consumption numbers can be obtained for power-supply and thermal analysis. By choosing the peripherals in use you can determine expected power consumption for worse case, i.e., 100% utilization.

The power spreadsheet divides the power consumption into two major components: baseline power and activity power. Using the power spreadsheet the power consumption can be predicted based on CPU frequency, temperature, % utilization, % writes, and % switching.

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. Therefore, you can affect the baseline power only by changing the device operating frequency (through the CPU frequency), the CV_{DD} voltage, or the operating temperature.

1.2 Activity Power

Activity consumption is power that is consumed by active parts of the DSP (the CPU, Enhanced Direct Memory Access (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: The operating frequency of a module or the frequency of external interface to that module
- Modes: Selects specific operating modes for certain peripherals.
- % Utilization: The relative amount of time the module is active or in use versus off or idled.
- % Write: The relative amount of time (considering active time only) the module is transmitting versus receiving.
- Bits: The number of data bits being used in a selectable-width interface.
- % Switch: The probability that any one data bit on the relative data bus will change state from one cycle to the next.

The C6455 and C6454 devices have the capability to disable peripherals that are not being used. When a peripheral is disabled, its clock is gated and the peripheral is held in reset thereby reducing the power consumption of the device. The spreadsheet that accompanies this application report includes a field that allows you to disable unused peripherals thus reducing the power consumption. The device data sheet should be consulted for more information on disabling or enabling the peripherals on the device.

1.3 Modules

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

- CPU
- Asynchronous External Memory Interface (EMIFA)
- SYSCLKOUT

- DDR2 Memory Controller
- Host-Post Interface (HPI)
- Multi-Channel Buffered Serial Port (McBSP0\1)
- Turbo-Decoder Coprocessor (TCP2)
- Viterbi-Decoder Coprocessor (VCP2)
- Timer0\1
- Ethernet Media Access Controller (EMAC)
  - RGMII
  - GMII
  - RMII
  - MII
- Serial RapidIO® (SRIO)
- Universal Test and Operations PHY Interface for ATM (UTOPIA)

The C6454 does not support all the peripherals listed here, but the spreadsheet is written in such a manner that non-applicable peripherals can be easily disabled.

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.

The current version of the power estimation spreadsheet does not include the PCI peripheral. To estimate power consumption for the PCI, the HPI entry can be used by entering similar bus utilization and data throughput numbers.

2 Using the Power Estimation Spreadsheet

Using the power estimation spreadsheet involves entering the appropriate usage parameters as input data in the spreadsheet. To use the spreadsheet simply:

- Choose the appropriate operating CPU frequency (700 MHz to 1200 MHz).
  - Core and SRIO voltages is set based on selected operating CPU frequency.
  - When CPU = 851 MHz to 1200 MHz, core and SRIO voltages is set to 1.25 V.
  - When CPU = 850 MHz or less, core and SRIO voltages is set to 1.2 V.
- Choose the case temperature for which you want to estimate power 0°C to 90°C.
- Enable the appropriate peripherals used for you application including the mode, frequency, and bus width for that peripheral if necessary.
- Fill in the appropriate peripherals or modules % utilization, % writes, and % switching.

The spreadsheet takes the provided information and displays the details of power consumption for the chosen configuration.

As the spreadsheet is being configured, the settings are checked for conflicts (i.e., peripherals 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. Modules that are completely unused and disabled from the peripheral configuration register can be disabled in the spreadsheet by selecting the *Disabled* button/tab in the column labeled *Status*. 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 the time the module spends doing something useful, versus being unused or idle. For these peripherals, there are no various degrees of use, so the value is just the average over time. For example, the DDR2 memory controller performs read 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 refreshes), this would be considered 25% utilization.

The CPU utilization is not as straightforward, because there are varying degrees of use for the CPU. Here, 0% utilization means the CPU is active and does no useful work (NOP execution) , where 100% utilization is representative of high activity condition , with all 8 functional units active every cycle making use of the software pipelined (SPLOOP) buffer hardware ,with maximum amount of data is brought in every cycle. Few DSP algorithms achieve 100% utilization, because this requires everything to be used every cycle, with no stalls. Even 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 algorithms. Control type code may only execute 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 approximately 74% utilization (25% x 25% + 90% x 75%). If the 25%/75% relation 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 approximately 63% (25% x 25% + 75% x 75%). By 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 can also reduce utilization. Though the spreadsheet accepts 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 assumes 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, the C6455 and C6454 devices provide the capability to enable only the peripherals that will be used. When a peripheral is disabled, the peripheral's clock is gated and the peripheral is held in reset.

A peripheral can be enabled or disabled in the spreadsheet from the column labeled *Status*. If the peripheral is disabled, the core and IO power for the peripheral will be zero. If the peripheral is enable with 0% utilization, the core supply reflects the delta power associated with enabling the peripheral versus the peripheral disabled.

Note: The power associated with enabling the peripherals will scale with the CPU frequency. However if the peripherals are driven by a fixed external clock source the power will not scale unless you change the external clock source.

For more information regarding peripheral enabling, please refer to the device data sheet, *TMS320C6455 Fixed-Point Digital Signal Processor Data Manual* (SPRS276).

2.3 Graphs

The output/results graphs in the spreadsheet provide a visual breakdown of the power consumption for the following:

- Core and IO activity power consumption
- Total System-on-a-Chip (SoC) power consumption
  - Leakage power
  - Clocking power
  - Activity and dynamic power consumption
- Percent total SoC power consumption

3 Using the Results

The power data presented in this document and the accompanying spreadsheet was collected from devices considered to be at the maximum end of power consumption for production device; no production units have average power consumption that exceeds the spreadsheet values. The spreadsheet data may, therefore, be considered maximum average power consumption. Transient currents may cause power to spike above the spreadsheet values for a small amount of time over a long period of time, however, the observed average power consumption will be below the spreadsheet value. Therefore, 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 on 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 EMIFA, DDR2, McBSP, UTOPIA, EMAC, SRIO, and Timer interfaces were loaded with approximately 4.5 inches of 50 Ω trace, with serial termination. PCI and HPI pins were loaded with approximately 2.5 inches of 50 Ω trace.

In order to reduce 3.3-V IO power with respect to HPI data buffers the spreadsheet has an option to enable 3.3-V IO pull-up resistors for HPI data pins HD[31:0]. You should be aware that enabling the pull-up resistor in the *Status* column significantly reduces the 3.3-V IO power consumption.

Note: The PCI specification prohibits the use of any kind of termination resistor on the PCI pins.

3.2 Spreadsheet Layout and Details

3.2.1 Baseline Section of the Spreadsheet

The results section of the baseline power portion of the power spreadsheet shown below consolidates the average power associated with leakage and clocking. The clocking power includes things like PLL power and the power consumed by active clocks in the system. The Total SoC (mW) column sums up the rows for leakage and clocking power.

The baseline portion of the srio-1.25/1.2 column shows the power consumption associated with enabling the SRIO macro. If the SRIO peripheral is disabled the macro power will be zero.

3.2.2 Activity Section of the Spreadsheet

The activity section of the spreadsheet contains the average power consumption associated with enabling a peripheral along with power consumed due to peripheral activity. The activity level of a peripheral is defined by the peripheral frequency, % utilization, % writes, % switching, bus width, and peripheral mode.

When the peripheral is disabled the core and IO power associated with that peripheral equals zero. If the peripheral is enabled with % utilization equal to zero, the core power associated with the peripheral is equal to the power consumption of the module when enabled. To increase power consumption is dependant on the module being enabled and CPU frequency. Configuring the % utilization and other associated parameters increases the power of the core and the IO to reflect the power consumption when using the peripheral.

The activity row for the SRIO only shows the power associated with actively transmitting and receiving data. To calculate the total SRIO power, the baseline portions of the srio-1.25/1.2 supplies should be considered as well. In other words, the total SRIO power is calculated by the following equation:

srio_total_power = srio_core_activity(Cvdd-1.25/1.2) + srio_io_baseline(srio-1.25/1.2) + srio_io_activity(srio-1.25/1.2)

3.2.3 Totals SoCSection of the Spreadsheet

The totals section provides the total in each column for each power supply and Baseline plus Activity power. The total (mW) is equal to the total power for core and IO all summed up i.e., total device power.

3.2.4 Idle Power Section of the Spreadsheet

The IDLE power is the power consumption associated with CPU PLL configured, CPU executing an “IDLE” instruction, and all peripherals disabled.

4 Voltage Supply Reference List

Table 2 provides a reference for the pins included under the different groups of core and IO voltages documented in the power spreadsheet.

Table 2. Power Pins

Group Name	Signal Name	Description
Cvdd-1.2/1.25	**CV_{DD}**	1.2-/1.25-V Core supply voltage
srio-1.2/1.25(1)	**DV_{DDRM}**	1.2-/1.25-V I/O supply voltage (SRIO interface supply)
	DV_{DD12}	1.2-/1.25-V I/O supply voltage (main SRIO supply)
	AV_{DDT}	1.2-/1.25-V I/O supply voltage (SRIO termination supply)
	AV_{DDA}	1.2-/1.25-V I/O supply voltage (SRIO analog supply)
Dvdd-3.3	**DV_{DD33}**	3.3-V I/O supply voltage
Dvdd-1.8	**DV_{DD18}**	1.8-V I/O supply voltage (DDR2 Memory Controller)
	DV_{DDR} (2)	1.8-V I/O supply voltage (SRIO regulator supply)
Dvdd-1.5	**DV_{DD15}**	1.5-V I/O supply voltage for the RGMII function of the EMAC
	$V_{REFHSTL}$	(DVDD15/2)-V reference for HSTL buffer (EMAC RGMII)
pll-1.8	PLLV1	1.8-V IO supply voltage for PLL1
	PLLV2	1.8-V IO supply voltage for PLL2
	AV $_{DLL1}$	1.8-V I/O supply voltage
	AV $_{DLL2}$	

(1) Applies to C6455 only. The SRIO field in the power consumption spreadsheet should be set to *disabled* when estimating power for the C6454 device.

(2) The DV_{DDR} pin consumes negligible power (less than 10 mA). Therefore, the power impact of this pin is not reflected on the ddr-1.8 column of the spreadsheet.

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