TMS320VC5510 Power Consumption Summary

Ryan Verret *C5000 Applications*

ABSTRACT

This document assists in the estimation of power consumption for the TMS320C5510 digital signal processor (DSP). As power consumption can vary widely on this device, a spreadsheet was developed to provide a better estimate. This allows the user to tailor the prediction to their particular application. It also allows designers the ability to test the efficiency of different configurations before any hardware is assembled or any code is written. Project collateral discussed in this document can be downloaded from http://www.ti.com/lit/zip/SPRA972.

Contents

1 Activity-Based Models

Power consumption of the TMS320C5510 DSP is extremely application dependent due to its many peripherals and variety of power saving modes. It is because of this that the application to be measured must be well understood before a power estimate can be given. The activity of each peripheral must be known in order to configure the power estimation spreadsheet for accurate results which can be used in power supply design or battery life prediction.

The model used in this spreadsheet is based on two modes of power consumption: static and activity power. With this model, each active component can be isolated and accurately modeled to determine its contribution to the static power.

1.1 Static Power

Static power is the power consumption inherent to the CMOS technology itself (sometimes called leakage power). Static power is not affected by the device activity or operating frequency. This power component is measured when the PLL is not enabled and nothing on the device is being clocked. Static power is affected by core voltage, IO voltage, and the device operating temperature.

1.2 Activity Power

Activity power is the consumption of the active parts of the DSP. These include the CPU, EMIF, PLL, peripherals, etc. Power consumption is based on voltage, frequency and the given configuration of each module. In order to provide more accurate power estimation, each block can be characterized independently and its relative contribution to overall consumption is provided. This aids in system design for greater efficiency.

Each module has parameters used to describe its activity. These include frequency, idle status, utilization, read/write balance, bus size and switching probability, trace length, and load capacitance. Each module may not include all of these parameters, however.

- *Frequency* is the operating frequency of a module, or the operating frequency of the interface to that module.
- *Idle Status* indicates whether the module is idle or active.
- *%Utilization* is the percentage of activity in a module relative to its maximum.
- *%Writes* is the percentage of writes relative to the total number of transfers (writes and reads).
- *Bits* is the number of data bits in use on a interface that supports variable-width busses.
- *% Switch* is the probability that a data bit will switch from one cycle to the next.
- *Trace Length* is the total board trace length being driven by each pin of the interface.
- *Load Capacitance* is the sum of the input capacitances at the end of each trace.

1.3 Modules

Each of the following modules is user-configurable in the C5510 power estimation spreadsheet within realistic operating parameters.

- DPLL
- CPU
- CLKOUT
- EMIF
- DMA (6 channels)
- HPI (as part of DMA)
- Instruction Cache
- McBSP0
- McBSP1
- McBSP2
- Timer0
- Timer1
- GPIO

2 Using the Power Estimation Spreadsheet

To use the spreadsheet, you simply enter the usage parameters into the white cells. To ensure that the data validation feature limits the input to realizable configurations, enter values from left to right, top to bottom. The spreadsheet will then take the provided information and display the details of power consumption for that configuration.

To simplify its operation, the spreadsheet does not support more than one idle configuration at any given time. For applications where the idle status is constantly changing, each possible configuration should be estimated independently, then the maximum taken for power supply design or all cases time-averaged to predict battery life.

2.1 Choosing Appropriate Values

It is critical to choose the correct values for the power estimation spreadsheet, to produce accurate results. Each module must be considered in isolation and you must account for all activity included in a given operation. For instance, ‘EMIF current’ is not included in the instruction cache ‘miss current.’ This must be included separately in the EMIF section.

2.1.1 Temperature

The static power is affected by the operating temperature of the device. Select the desired temperature (in degrees Celsius) by entering the temperature in the “Temp” cell at the top of the spreadsheet.

2.1.2 Frequency

The module frequency should be straightforward to determine. It will either be the DPLL frequency, or some integer fraction of it. For some modules, such as the timers, the frequency is automatically set to the DPLL frequency for core power consumption and set to the specified frequency for IO power consumption.

2.1.3 Idle Status

Idle status for a given module is simply whether or not that module is configured by software to be in its idle state. The spreadsheet only supports on idle configuration at any time.

2.1.4 % Utilization

Utilization is explicitly defined for each module in order to provide a more accurate estimate of power consumption. If a module is not listed, then it is assumed to be in use whenever it is not idle.

- **CPU** – Since there are varying degrees of activity for the CPU, it is more difficult to provide a utilization number. Whenever the CPU is active (non-idle) it is repeatedly executing instructions. For this reason, 0% activity will be defined as a repeated NOP instruction– the smallest amount of power the CPU can consume while active. Conversely, 100% activity will be defined as the most power intensive instruction – the dual multiply and accumulate. All other instructions will fall somewhere in between. No single algorithm will achieve 100% utilization, but some highly optimized functions can come close. In addition, the CPU must also be used for control oriented tasks that consume far less current.

  For example, assume that a certain application executes control code half of the time and a highly optimized algorithm for the other half. If the control code is estimated to be at 30% utilization and the dense DSP code is estimated to be at 90% utilization, the overall utilization would be 60% (30% $\times$ 50% + 90% $\times$ 50%). If the application spent more time executing the optimized algorithm, utilization would obviously go up, and vice versa. By examining individual portions of an application and estimating the utilization and time spent in each, a more accurate CPU utilization percentage can be obtained.
- **CLKOUT** – The clock output utilization percentage is simply defined as the percentage of time that the DSP core clock is being output. On the 5510 design, the CLKOUT output buffer is operated by the CVdd supply (the spreadsheet credits power consumed by the CLKOUT correctly).
- **EMIF** – EMIF utilization is related to the maximum bandwidth of the EMIF. One hundred percent utilization corresponds to the maximum transfer rate for a given frequency. This number will be scaled down by both slower and less frequent transfers. As the minimum period for EMIF accesses is three cycles, the utilization percentage is defined as three times the average frequency of activity over the entire block of code being analyzed divided by the actual EMIF clock frequency. The average activity frequency is simply the average number of EMIF accesses per second and the EMIF clock frequency is some fraction of the DPLL frequency.

  Choose the desired memory type (synchronous or asynchronous) in the “Other” column.
- **HPI** – HPI utilization is also related to the maximum bandwidth of the HPI. The minimum period for HPI accesses is 14 CPU cycles + 10ns (as referenced in the 5510 data manual, literature number SPRS076) At 200 MHz, this is equivalent to 16 cycles, so the utilization percentage is defined as sixteen times the average activity frequency divided by the DPLL frequency.
- **DMA** – DMA utilization for a given channel is related to the maximum bandwidth of the port for each data resource, which is one transfer each cycle. This results in a utilization number that is simply the DMA channel’s average activity frequency divided by the DPLL frequency.
- **Instruction Cache** – Utilization of the instruction cache is defined as the percentage of time that the CPU executes instructions from external memory.

- **McBSP** – McBSP utilization is defined as the percentage of time that the McBSP is transferring data.
- **Timer** – Timer utilization is defined as the percentage of time that the timer is counting. The timer can be configured to use the TIN/TOUT pin as an output, an input or not used. If this pin is used as an output, select TOUT enabled in the “Other” column. Otherwise, select TOUT disabled.
- **GPIO/XF** – Utilization for general purpose outputs is the percentage of time that they are switching at their specified frequency. Select the number of GPIO pins used in the “Other” column of the spreadsheet.

2.1.5 % Writes

For modules that move data onto a bus, *% writes* is defined as the percentage of utilization in which the peripheral is putting data on the bus. Data is assumed to be taken off the bus for the remainder of the utilization period. For modules with equal input / output bandwidth, the write percentage should be 50%. And for the instruction cache, *% writes* is the hit percentage of instruction accesses.

2.1.6 Bits

Bits is the number of data bits on a variable width bus. The spreadsheet only permits valid bus sizes.

2.1.7 % Switch

Random data has a 50% chance that any bit will change from one cycle to the next. Some applications may be able to predict this change 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%.

2.1.8 Trace Length

The trace length parameter indicates the average length of controlled-impedance board trace of each output for a given interface (assumed 50Ω trace). This only affects IO power and further, only contributes when the modules are writing.

2.1.9 Load Capacitance

Load capacitance is the average of the sum of the input capacitances of all external devices connected to a given module’s outputs. This parameter should be obtained from the datasheets of the devices that are connected to the DSP.

2.1.10 Other

Some modules have other parameters that are self-explanatory. These include whether the EMIF is configured in synchronous or asynchronous mode, if the timers are configured as outputs, and if the number of GPIO pins configured as outputs.

2.2 Units

The results are estimated in the spreadsheet and displayed in milliamps (mA) or millliwatts (mW). Click on the units in the “Total” row of the calculated results and use the pull-down menu to select the desired units.

2.3 Graphs

The graphs included in the spreadsheet show the relative contribution of total core and IO power for each module. They provide a visual display of power usage and allow easy identification of the major power consumers.

3 Using the Results

The results presented in the spreadsheet are based on measured data for revision 2.2 silicon. The measured units were selected to be towards the high end of power consumption for production units. Most production units will have power consumption that is below the value given in the spreadsheet, if accurately modeled.

The spreadsheet produces an estimated average current (or power) for the specified configuration. Transient currents may cause power to exceed the estimated value for short periods, but long term power consumption should be at or below the spreadsheet value. This allows for better estimates of power supply requirements and more accurate battery life predictions.

4 References

1. *TMS320VC5510 Fixed-Point Digital Signal Process Data Manual* (SPRS076)

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