Application Report
SPRAB31–March 2009

TMS320DM357 Power Consumption Summary

Catalog DSPS Applications

ABSTRACT

This application report discusses the power consumption of the Texas Instruments TMS320DM357 digital media System-on-Chip (DMSoC). Power consumption on the DM357 device is highly application-dependent; therefore, a spreadsheet that estimates power consumption is provided along with this document. To obtain good results from the spreadsheet, realistic usage parameters must be entered (see Section 2.1). 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 the accompanying spreadsheet was measured from strong units, representative of devices at the maximum end of power consumption for production units. No production units will have average power consumption that exceeds the spreadsheet values; therefore, the spreadsheet values may be used for board thermal analysis and power supply design as a maximum long-term average.

The spreadsheet discussed in this application report can be downloaded from the following URL: http://www.ti.com/lit/zip/SPRAB31.

Table 1. Nominal Activity

Core Voltage	Device Configuration	ARM Core Frequency	Power (mW) (1)			
			Core	IO18	IO33	Total
1.2 V	Static	0 MHz	252	36	20	308
	Standby	13.5 MHz	273	38	20	330
	540 MHz	270 MHz	693	180	20	893

(1) Assumes the following conditions: Static power consumption varies with process, temperature and voltage. In standby power consumption, both static and clocking power is considered. For 540 MHz configuration, HMJCP utilization is 60% for video encode and decode application, ARM doing typical activity (peripheral configurations, other housekeeping activities,) DDR2 at 50% utilization (135 MHz), 50% writes, 32 bits, 50% bit switching, 2-MHz ASP at 100% utilization; Timer0 at 100% utilization. The readings shown here are taken at room temperature (25°C).

Contents

List of Tables

1 Model Activity

1.1 Activity-Based Models

Power consumption for the TMS320DM357 can vary widely depending on the use of on-chip resources. Therefore, you cannot estimate power consumption accurately without an understanding of the components of the DMSoC in use and the usage patterns for those components. By providing the usage parameters that describe how and what on the DMSoC 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, various applications that use the DMSoC differently can get accurate predictions across the spectrum of possible power consumption on the DM357.

1.2 Baseline Power

Baseline power consumption is power that is not dependent on chip activity such as static power (leakage), phase-locked loop (PLL), oscillator power, DDR2 DLL consumption, and clock tree power to various subsystem components (e.g., SCR) that cannot be turned off via the on-chip power management module. While independent of SoC activity, baseline power is dependent on the device operating frequency, voltage, and temperature. Therefore, you can affect baseline power only by changing the PLL(s) output frequency, the core voltage, or the operating temperature (affect the leakage current).

1.3 Activity Power

Activity consumption is power that is consumed by active parts of the DMSoC: CPU(s), EMIF, peripherals, etc. Activity power is independent of temperature, but dependent on voltage and activity levels. In the spreadsheet, activity power is separated by the major modules within the device. Therefore, the individual module power consumption can be estimated independently; 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. 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.
- Status: Indicates whether the module is in an enabled or disabled state.
- % Utilization: The relative amount of time the module is active or in use versus off or idle.
- % 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 will change state from one cycle to the next.

Not all modules include all of the parameters.

1.4 Modules

The DM357 power estimation spreadsheet contains the following modules with adjustable parameters:

- ARM MM (megamodule) (see Section 5)
- H.264/MPEG4/JPEG Coprocessor (HMJCP)
- Video Processing Subsystem (VPSS)
  - Video Processing Front End (VPFE)
  - Video Processing Back End (VPBE)
- DDR2 Memory Controller
- External Memory Interfaces (EMIFA)
- Ethernet Media Access Controller (EMAC)
- Universal Serial Bus (USB) 2.0 (see Section 5 for limitations)
- Audio Serial Port (ASP)
- MultiMedia Card/Secure Data Memory Card (MMC/SD)

- Serial Peripheral Interface (SPI)
- Inter-Integrated Circuit (I2C)
- Timer (0 and 1)
- Watchdog Timer (WDT)
- Pulse Width Modulator (PWM) (0 and 1)
- Universal Asynchronous Receiver/Transmitter (UART) (0, 1 and 2)
- Host Port Interface (HPI)
- General-Purpose Input/Output (GPIO)
- Enhanced Direct Memory Access (EDMA3)

Although EDMA3 is listed as a separate module, the row essentially provides the power consumption on turning on the clock to the channel controller and the transfer controller(s) for a particular device frequency and voltage. The EDMA3 activity power is included in the activity power of the module/peripheral serviced by the EDMA3 (this includes MMC/SD, ASP, SPI, UARTs, I2C, EMIFA, DDR2). Therefore, in estimating power for peripherals that typically use EDMA3 for their transfers, the EDMA3 should be kept enabled. For available peripherals and peripheral configuration, see the device-specific data manual.

2 Using the Power Estimation Spreadsheet

The power estimation spreadsheet involves entering the appropriate usage parameters as input data in the spreadsheet. Cells that are designed for user input are white in color. The following steps explain how to use the spreadsheet:

1. Choose the case temperature for the estimated power
2. Fill in the appropriate module use parameters

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, e.g., peripheral clock frequency out of allowed range, etc. For best results, enter the information from left to right starting at the top and moving downward.

2.1 Choosing Appropriate Values

The frequency and bit user values are determined by design and it will be clear what the correct values to enter are. For some modules, the frequency field is used to input the data rates (for instance EMAC, etc). In order to choose appropriate values, you need a good understanding of read/write balance, bit switching required estimation, and utilization of the user application. You should also keep in mind the pin multiplexing configuration for the device to avoid enabling mutually exclusive peripherals/configurations at the same time. For available peripherals and peripheral configuration, see the device-specific data manual.

2.1.1 Utilization

Utilization is simply the percentage of time the module spends doing something useful, versus being unused or idle. For these modules, there are not various degrees of use, so the value is just an average over time. For example, the DDR2 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 I/O, 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, with a theoretical 400 Kb/s maximum, the I2C port utilization would be about 40%. In some cases, the max bandwidth allowed may be dependent on the device (SYSCLK1) frequency; factor this while calculating the utilization if doing the estimation at different device frequencies.

The HMJCP performs the computational operations required for image processing. The utilization is based on the algorithm implemented. The spreadsheet can be configured to get power for the following algorithm:

- Encode (D1 to H.264)
- Decode (H.264 to D1)
- Encode and Decode (D1 to H.264 to CIF)

System level issues may also reduce utilization. Though the spreadsheet accepts 100% utilization for all peripherals, this is not possible in reality. As concurrency in data movement increases and/or throughput requirements on high bandwidth modules (system DMA, VPSS, etc.,) increases, overall peripheral activity is throttled back due to bottlenecks created at various common end points. In such cases, peripherals might not achieve 100% utilization; therefore, you should enter individual module utilization numbers keeping this overall limitation in mind.

2.1.2 % Writes

Peripherals that transmit as much data as they receive 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 DMSoC 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 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 previously, the DM357 device provides the capability to disable modules that are not being used via the power sleep controller (PSC). When a peripheral is disabled, its clock is turned off reducing the power consumption of the device.

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

If a module is not used for a given application, then it is recommended to keep it in disabled state.

It is possible that the module is kept enabled but has no activity. To achieved this, program the % utilization and/or the frequency fields to a value of 0, then the numbers in the module's row will be indicative of the power consumed by clocking the module.

For VPFE and VPBE, the enable/disable functionality is controlled by the video processing sub system (VPSS) enable/disable switch in the spreadsheet. If the application requires the use of VPFE and/or VPBE, you can enable it by *enabling* the VPSS.

If an application requires the use of VPFE and VPBE in the system, this automatically implies that DDR2 is also enabled and active in the system. To estimate power for such scenarios, you need to make sure that the DDR2 fields are programmed with the appropriate frequency and utilization, read/write percentage, etc., required by the VPFE and/or VPBE.

In the current implementation of the spreadsheet, if the DDR2 is disabled, it also assumes that the PLL2 is powered down and is operating in bypass mode with the DDR2 clock being directly fed by the input reference clock (27 MHz CLKIN).

2.3 Graphs

The *Case Temp Measurement* sheet provides the worst case junction temperature and case temperature values for given inputs.

The *Heat Sink Calculation* sheet shows the impact of heat sink on case temperature for given inputs.

The *DM357_Graph* provides a visual breakdown of power consumption. A comparison is also provided for active power (based on the parameters supplied) and the baseline power, and pie charts show the relative contributions of each peripheral to the core and I/O power consumption.

3 Using The Results

The results presented by the spreadsheet are based on measured data for TMS320DM357 silicon revision 2.1 and earlier devices.

The intent of the power estimation spreadsheet is to provide estimates of the upper bounds in an application-specific loading and peripheral utilization scenario. The measured units were selected to be a strong unit at the maximum end of power consumption for production units. No production units will have average power consumption that exceeds the spreadsheet values; therefore, the spreadsheet data may be considered maximum average power consumption; the actual observed power may vary. That is, transient currents may cause power to spike above the spreadsheet value for a small amount of time; however, over a long period, the observed average consumption will be below the spreadsheet value. The spreadsheet value may be used for board thermal analysis and power supply design as a maximum long-term average.

3.1 Adjusting I/O Power Results

I/O power is dependent not only on the device 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 the number of inputs or lumped load capacitance. If the target system has very different I/O loading, scale the spreadsheet results either up or down to compensate. For this reason, the spreadsheet allows you to specify the approximate load on the I/O pins for each module by using the trace length field. This parameter is used to adjust the reported I/O power numbers.

4 Example

The following examples demonstrate how to choose appropriate values for a particular application. These values may be imported into the spreadsheet by clicking the appropriate macro button.

4.1 Basic Configuration(s)

The *Static* macro button reports the static power for the device, when the MXI/CLKIN is cut off and voltage is applied to the various core and I/O rails

The *Standby* macro button reports the power consumed with the 27 MHz CLKIN. PLL1 is powered down/disabled and the system is operating in bypass mode with the CLKIN as the system clock. PLL2 is also powered down/disabled and is operating in bypass mode with the CLKIN passed through to DDR2 clock. The ARM is in the *wait for interrupt* sleep mode, the DSP subsystem power domain is OFF and all peripherals are disabled.

The *270 MHz* macro button reports the power consumed for the following conditions: 60% HMJCP utilization (performing video encode and decode operations); ARM doing typical activity (peripheral configurations and other housekeeping activities), DDR2 at 50% utilization (135 MHz), 50% writes, 32 bits, 50% bit switching, 2-MHz ASP at 100% utilization; Timer0 at 100% utilization. At room temp (25°C)

Note: In *Static* configuration, a *finite* current component is shown on the I/O rails (DV_{DD18} and DV_{DD33}), these values are system dependent and will differ (could be lower or higher) based on the state of pull up or pull downs and load on the device I/O lines.

4.2 Sample Application(s)

The sample application contains the following configuration: the VPFE is reading in video stream of D1 4:2:2 (720x480) video data @ 30 fps from the DDR2, the HMJCP encodes it with H.264 standard and decodes it with PAL-CIF standard to display it on LCD through VPBE (352x288 @ 25fps). For audio, the ASP feeds in an audio stream directly to the SoC (internal memory) using EDMA3, 96 KHz sampling rate

with 8 bit of data for two channels. The encoded audio/video data is written to the DDR2. The ARM is primarily responsible for configuring peripherals and running some house keeping code. All three PWM modules (PWM 0/1/2) are enabled and running at 500 KHz (50% duty cycle). Additionally I2C is being used for control functions on the audio codec, etc., and Timer0 and Timer 1 are also enabled and running. The EMIFA module is enabled (AEMIF boot mode).

The device is running at 1.2 V, HMJCP on, 270 MHz ARM, at 40°C. The DDR2 is running at 162 MHz, with a 32-bit bus. All other peripherals are disabled.

The details of the peripherals used in this application are:

- Voltage: 1.2 V (CV_{DD})
- Case Temperature: 40°C
- CLKIN: 27 MHz
- HMJCP: Turned on, 50% utilization. Estimation based on the following factors:
  - Video/Audio Encode: 60% of the HMJCP utilization for 80% of the time
  - Background Activities (waiting for commands from ARM, polling for completion, etc.): 10% of HMJCP capability for 20% of the time
- ARM: *typical activity* that involves configuring peripherals and other house keeping activities
- VPSS: Enabled (enables VPFE and VPBE)
- VPFE: 10 Mpixel/sec , 100% utilization
  - CCDC to DDR2 , 720 x 480 @ 30 fps (approx 10 Mpix/sec or 20 Mbytes/sec with 2 bytes/pixel)
  - All other sub-modules (Histogram, H3A, Preview Engine, Resizer) not used DDR2
- VPBE: 10 Mpixel/sec, 100% utilization (continuous display), digital mode (to LCD)
  - OSD reading data from DDR2 at the rate of 720 x 480 @ 30 fps (approx 10 Mpix/sec or 20 Mbytes/sec with 2 bytes/pixel)
  - EDMA3: Enabled. Used for various memory-to-memory transfers and peripheral-to-memory transfers (like ASP)
- DDR2: 7.25% utilization, 23% writes, 50% switching
  - Writes from VPFE: 20 Mbytes/sec
  - Reads from VPBE: 20 Mbytes/sec
  - Reads issued by HMJCP: (720x480 + 720x480 + 360x240+ 360 x 240) x 16bits @ 30 fps: 50 Mbytes/sec
  - Writes (Output Video/Audio Out): 15.884 mbps : 1.985 Mbytes/sec
  - Reads (from ATA): 1.985 Mbytes/sec
  - Total: 94 Mbytes/sec (1296 Mbytes/sec max for 32 bit DDR2 at 162 MHz), 22 Mbytes/sec of which are writes
- ASP: 2 MHz (output frequency), 100 % utilization , 50% switching
  - 96 KHz 8 bit data , 2 channels : 96 x1024 x 8 x 2 = 1.56 Mbps
- I2C: 214 Kbps , 100% utilization
- Timer 0 and Timer 1: 100% utilization
- PWM0 , PWM1 and PWM2: 500 KHz , 100% utilization
- All other modules are not used and are disabled

Entering these values into the spreadsheet gives a maximum power consumption of about 800 mW for core (CV_{DD} and CV_{DDDSP} combined) and for 171 mW on 1.8 I/O rail (all I.8 V I/O's combined) and 20 mW on the 3.3 V I/O rail , for a total of 991 mW.

5 Limitations

The current implementation of the power estimation spreadsheet has the following limitations:

- ARM CPU power representation and reporting: ARM mega module power consumption estimation does not follow the traditional CPU usage parameters (like the DSP), and is currently broken into three main categories
  - Sleep Mode: Special *wait-for-interrupt* sleep mode, see the *TMS320DM644x DMSoC ARM*

Subsystem Reference Guide (SPRUE14)
  - Low Activity: This is representative of ARM just performing low activity instructions (NOPs, etc.)
  - Typical Activity: This is representative of ARM performing typical activity, which would include peripheral configurations, load/stores and other house keeping activities (waiting for transfer completions, interrupts, etc.)
- USB2.0 Power representation: The USB2.0 module power consumption is based on modeled estimates and not actual measurements. Currently, the spreadsheet just allows you to estimate the core power consumed on enabling the USB2.0 module via PSC (and variations with device frequency). For USB /IO power, typical worst case *expected* numbers are provided for different usage modes (high-speed receive/transmit and full-speed receive/transmit).
- All measurements have been performed with a 27 MHz CLKIN provided by an external oscillator. The spreadsheet does not provide a capability of estimating power based on a different CLKIN value.
- Due to limitations in the test setup, HPI data was calculated based on trends from other devices. The data provided is a realistic estimate of the HPI's power consumption.

6 References

- *TMS320DM644x DMSoC ARM Subsystem Reference Guide* (SPRUE14)
- *Thermal Considerations for TMS320DM64xx, TMS320DM64x, and TMS320C6000 Devices* (SPRAAL9)

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