TMS320DM6446/3 Power Consumption Summary

Mukul Bhatnagar *Device Applications*

ABSTRACT

This document discusses the power consumption of the Texas Instruments TMS320DM6446 and TMS320DM6443 digital media System-on-Chip (DMSoC). Power consumption on the DM6446/3 devices is highly application-dependent, so spreadsheets are 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 devices to operate with industry-leading performance, while maintaining a low power-to-performance ratio.

The data presented in the accompanying spreadsheets was measured from strong units (representative of maximum end of power consumption for production units); no production units 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 data presented in this document is actual measured power consumption for the DM6446 and DM6443 silicon revision 2.1 and earlier devices.

This application report contains spreadsheets that can be downloaded from http://www.ti.com/lit/zip/SPRAAD6.

Table 1. Nominal Activity

			Power at Frequency (W)[1]			
Core Voltage	**CPU Frequency**	**Process[2]**	**Internal Logic**	**IO18**	**IO33**	**Total**
1.2 V	594 MHz	STRONG	1.05	0.16	0.02	1.23
		TYPICAL	0.92	0.16	0.02	1.1
1.2 V	459 MHz	STRONG	0.89	0.15	0.02	1.06
		TYPICAL	0.76	0.15	0.02	0.93
1.3 V	810 MHz	STRONG	1.37	0.16	0.02	1.55

(1) Assumes the following conditions: 60% DSP CPU utilization; 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. At room temp (25°C)

(2) The difference in the current consumption is essentially due to the difference in static or leakage current across manufacturing process. Variations will also occur depending on the core voltage and operating temperature.

Contents

List of Tables

1 Activity-Based Models

TMS320DM6446/3 power consumption can vary widely depending on the use of on-chip resources. Therefore, power consumption cannot be estimated 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, different applications that use the DMSoC differently can get accurate predictions, all across the spectrum of possible power consumption on DM6446/3 .

1.1 Baseline Power

Baseline power consumption is power that is not dependent on chip activity. This includes items such as static power (leakage), PLL(s) , oscillator power, DDR2 DLL power, 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 (effects the leakage current).

1.2 Activity Power

Activity consumption is power that is consumed by active parts of the DMSoC (the CPU(s), external memory interfaces, 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. 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.
- *Status* indicates whether the module is enabled or disabled state.
- *% 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 device versus reading data into the device.
- *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.

However, each module may not include all of the parameters.

1.3 Modules

The DM6446/3 power estimation spreadsheet contains the following modules with adjustable parameters:

- Digital Signal Processor (DSP)
- ARM MM (megamodule) [see Section 5]
- Video Imaging Co-Processor (VICP)
- Video Processing Front End (VPFE)
- Video Processing Back End (VPBE)
- Enhanced Direct Memory Access (EDMA3)
- DDR2 Memory
- External Memory Interface (EMIFA)
- AT Attachment/ Compact Flash (ATA/CF)
- Ethernet Media Access Controller (EMAC)
- VLYNQ™
- Universal Serial Bus (USB 2.0) [see Section 5]
- Audio Serial Port (ASP)

- Multimedia Card (MMC)
- Secure Digital (SD)
- Serial Peripheral Interface (SPI)
- General-Purpose Input/Output (GPIO)
- Universal Asynchronous Receiver/Transmitter (UART0/1/2)
- Timer 0/1
- Watchdog Timer (WDT)
- Inter-Integrated Circuit (I2C)
- Pulse Width Modulator (PWM 0/1/2)
- Host Port Interface (HPI) [see Section 5]

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

Using the power estimation spreadsheet involves 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
- Fill in the appropriate module use parameters

The spreadsheet takes the provided information and displays the details of power consumption for that 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 bits 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, ATA, EMAC, etc.). The utilization, read/write balance, and bit switching require estimation and a good understanding of the user application to choose appropriate values. To avoid enabling mutually exclusive peripherals/configurations at the same time, remember the pin multiplexing configuration for the device. For available peripherals and peripheral configuration, see the device-specific data manual.

2.1.1 % Utilization

For modules except DSP (C64x+™) and VICP, 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%. Similarly for ATA , when DM6446 is running at 594 MHz, the max theoretical bandwidth possible is 99 Mbytes/sec in UDMA5 mode, so for an application using UDMA4 mode (expected bandwidth of around 66 Mbytes/sec), the % utilization would be 66.6%. In some cases, the max bandwidth allowed maybe dependent on the device (SYSCLK1) frequency, then the user should factor this while calculating the utilization if doing the estimation at different device frequencies.

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 will 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. These types of code may execute only a few instructions in parallel and significantly reduce the I/O of the CPU, and thus reduce overall utilization. Therefore, 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 very dense CPU intensive code (estimated at 90% of CPU capability) half of the time, and for the other half low activity, like waiting for commands from ARM , along with some other house keeping activities (estimated at 10% of CPU capability) ,would have an average utilization of about 50% (10% x 50% + 90% x 50%). If the balance were changed to 25% low activity code and 75% DSP code, the weighted average would be about 70% utilization (25% x 10% + 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 59% (25% x 10% + 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 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 , VP_{SS}, 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 individual module utilization numbers should be entered keeping this overall limitation in mind.

2.1.2 % Writes

Peripherals that move data out of the device as much as they move data into the device 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 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 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, DM6446/3 devices provide 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 thereby 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.

For a given application, if a module is not used, then it is typically recommended to keep it in disabled state.

It is plausible that the module is kept enabled but has no activity, this can be appropriately programmed by programming the % utilization and/or the frequency fields to a value of 0, in which case the numbers in the module's row will be indicative of the power consumed by clocking the module.

For the DSP, the spreadsheet provides three states: disabled, idled and active.

- Disabled : Indicates the “DSP” power domain OFF (DSP Subsystem clock gated off). This is also the default power on reset condition in “Host” boot mode or DSP_BT = 0 (see the data sheet).
- Idled : Indicates the DSP sleep mode using the C64x+ internal power down controller (PDC)
- Active : Indicates the DSP state when it is not idled or powered down, you should put an appropriate % utilization in this mode.

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 should 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 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 graphs page contains graphs that provide a visual breakdown of power consumption. Shown are a comparison of active power (based on the parameters supplied) and the baseline power, and pie charts showing the relative contributions of each peripheral to the core and I/O power consumption.

3 Using the Results

The results presented by the spreadsheets are based on measured power consumption for the DM6446 and DM6443 silicon revision 2.1 and earlier devices.

The intent of the power estimation spreadsheet s are to provide estimates of the upper bounds in an application specific loading and peripheral utilization scenario. Therefore, the measured units were selected to be strong unit at the maximum end of power consumption for production units; no production units have average power consumption that exceeds the spreadsheet values. The spreadsheet data may be considered maximum average power consumption, however, 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, 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 I/O Power Results

I/O power is dependent not only on the device and it’s 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. If the target system has very different I/O loading, the spreadsheet results may be scaled 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

Here are some examples 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 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.

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

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 macro buttons *459 MHz*, *594 MHz* and *810 MHz* allow you to quickly visualize the power numbers shown in Table 1.

4.2 594 MHz Sample Application (Video Encoder)

For this example, the VPFE is reading in video stream of D1 4:2:2 (720x480) video data @ 30 fps to the DDR2. The same data is read for the DDR2 displayed on the LCD using the VPBE (720x480 @ 30 fps).The data is being read from DDR2 into C64x+ using EDMA3 along with reference frames D1 4:2:0 (720x480 + 360x240 + 360x240), to perform video compression. For audio, the ASP feeds in an audio stream directly to the C64x+ (L2 memory) using EDMA3, 96 KHz sampling rate with 8 bits of data for 2 channels. The encoded audio/video data is written to the DDR2. This data is stored on a compact Flash via ATA, at a bit rate of 15.5 mbps for video and 384 kbps for audio. 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, 594 MHz DSP, 297 MHz ARM at 40°C. The DDR2 is running at 162 MHz with a 32-bit bus. All other peripherals are disabled.

- Voltage: 1.2 V (CV_{DD} and CV_{DDDSP})
- Case Temperature: 40°C
- CLKIN: 27 MHz
- DSP: 594 MHz , 50% utilization. Estimated based on the following factors:
  - Video/Audio Encode: 60% of the CPU utilization for 80% of the time.
  - Background Activities (Waiting for commands from ARM , polling for completion, etc.) : 10% of CPU capability for 20% of the time
- VICP : Enabled, 50% Utilization
- ARM: *Typical Activity*, which involves configuring peripherals, and other house keeping activities.
- VPSS : Enabled (Enables VPFE and VPBE)
- VPFE : 10 Mpixel/sec , 100% utilization, see Appendix A.
  - 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) do not use 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 (approximately 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 DSP : (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.
- ATA : 1.99 Mbytes/sec, 2 % utilization , 50% switching.
  - Video : 15.5 mpbs
  - Audio : 384 kbps
  - Total bandwidth : 15.884 mbps : 1.99 Mbytes/sec
  - ATA max bandwidth : 99 Mbytes/sec (UDMA5, with DM6446 running at 594 MHz)
- 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 1244 mW for core (CV_{DD} and CV_{DDDSP} combined) and for 156 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 1417 mW.

4.3 810 MHz Sample Application (Video Encoder)

For this example, the VPFE is reading in video stream of D1 4:2:2 (720x480) video data @ 30 fps to the DDR2. The same data is read for the DDR2 displayed on the LCD using the VPBE (720x480 @ 30 fps).The data is being read from DDR2 into C64x+ using EDMA3 along with reference frames D1 4:2:0 (720x480 + 360x240 + 360x240), to perform video compression. For audio, the ASP feeds in an audio stream directly to the C64x+ (L2 memory) using EDMA3, 96 KHz sampling rate with 8 bits of data for 2 channels. The encoded audio/video data is written to the DDR2. This data is stored on a compact Flash via ATA, at a bit rate of 15.5 mbps for video and 384 kbps for audio. 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 1.3 V, 810 MHz DSP, 405 MHz ARM. The DDR2 is running at 162 MHz with a 32-bit bus. All other peripherals are disabled.

- Voltage: 1.3 V (CV_{DD} and CV_{DDDSP})
- Case Temperature: 40°C
- CLKIN: 27 MHz
- DSP: 810 MHz , 50% utilization. Estimated based on the following factors:
  - Video/Audio Encode: 60% of the CPU utilization for 80% of the time.
  - Background Activities (Waiting for commands from ARM , polling for completion, etc.) : 10% of CPU capability for 20% of the time
- VICP : Enabled, 50% Utilization
- ARM: *Typical Activity*, which involves configuring peripherals, and other house keeping activities.
- VPSS : Enabled (Enables VPFE and VPBE)
- VPFE : 10 Mpixel/sec , 100% utilization, see Appendix A.
  - 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) do not use 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 (approximately 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 DSP : (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.
- ATA : 1.99 Mbytes/sec, 2% utilization , 50% switching.
  - Video : 15.5 mpbs
  - Audio : 384 kbps
  - Total bandwidth : 15.884 mbps : 1.99 Mbytes/sec
  - ATA max bandwidth : 99 Mbytes/sec (UDMA5, with DM6446 running at 810 MHz)
- 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 1668 mW for core (CV_{DD} and CV_{DDDSP} combined) and for 129 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 1817 mW.

5 Limitations

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

- ARM CPU power representation and reporting : ARM megamodule power consumption estimation does not follow the traditional CPU usage parameters (like the DSP), and currently is broken into three main categories:
  - Sleep Mode : Special "wait-for-interrupt" sleep mode, see *TMS320DM644x DMSoC ARM Subsystem Reference Guide* (Literature Number SPRUE14)
  - Low Activity : This is representative of ARM performing just low activity instructions (NOPs, etc.)
  - Typical Activity : This is representative of ARM performing typical activity which would typically 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 of modeled estimates and not actual measurements. Currently the spreadsheet just allows the user to estimate the core power consumed on enabling the USB2.0 module via the PSC (and variations with device frequency). For USB I/O 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

1. *TMS320DM6446 Digital Media System-on-Chip* Data Manual (Literature Number SPRS283)

2. *TMS320DM6443 Digital Media System-on-Chip* Data Manual (Literature Number SPRS282)
3. *TMS320DM644x DMSoC ARM Subsystem* Reference Guide (Literature Number SPRUE14)
4. *TMS320C64x+ Megamodule Reference Guide* (Literature Number SPRU871)
5. *TMS320DM64x Power Consumption Summary* Application Report (Literature Number SPRA962)

Appendix A

Table 2 provides some common video formats and frame size along with the equivalent Mpixel/sec (for given frame rates). This could be used to program the “Frequency” parameter on the VPFE and VPBE modules.

Table 2. Common Video Formats

Format	Frame Size or Resolution	Mpixel/sec
NTSC - D1 @ 30fps	720 x 480	10.37
PAL - D1 @ 25fps	720 x 576	10.37
VGA @ 30fps	640 x 480	9.22
NTSC - SIF @ 30fps	352 x 240	2.53
PAL - CIF @ 25fps	352 x 288	2.53
QVGA @ 30fps	320 x 240	2.30
ATSC - 1080i @ 30fps	1920 x 1080	62.21
ATSC - 720p @ 60fps	1280 x 720	55.30
ATSC - 480p @ 60fps	720 x 480	20.74

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