TEXAS INSTRUMENTS

Calculation of TMS320C40 Power Dissipation

Application Report

1993 Digital Signal Processing Products

Texas Instruments

Printed in U.S.A., November 1993 SPRA032

Calculation of TMS320C40 Power Dissipation Application Report

Jenny Hong
Digital Signal Processing Products - Semiconductor Group
Texas Instruments

SPRA032
November 1993

Printed on Recycled Paper

IMPORTANT NOTICE

Texas Instruments Incorporated (TI) reserves the right to make changes to its products or to discontinue any semiconductor product or service without notice, and advises its customers to obtain the latest version of relevant information to verify, before placing orders, that the information being relied on is current.

TI warrants performance of its semiconductor products and related software to current specifications in accordance with TI's standard warranty. Testing and other quality control techniques are utilized to the extent TI deems necessary to support this warranty. Specific testing of all parameters of each device is not necessarily performed, except those mandated by government requirements.

Please be aware that TI products are not intended for use in life-support appliances, devices, or systems. Use of TI product in such applications requires the written approval of the appropriate TI officer. Certain applications using semiconductor devices may involve potential risks of personal injury, property damage, or loss of life. In order to minimize these risks, adequate design and operating safeguards should be provided by the customer to minimize inherent or procedural hazards. Inclusion of TI products in such applications is understood to be fully at the risk of the customer using TI devices or systems.

TI assumes no liability for applications assistance, customer product design, software performance, or infringement of patents or services described herein. Nor does TI warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right of TI covering or relating to any combination, machine, or process in which such semiconductor products or services might be or are used.

Contents

List of Illustrations

List of Tables

Introduction

The Texas Instruments **TMS320C40** Digital Signal Processor (DSP) is a high-performance 32-bit floating-point parallel microprocessor implemented in CMOS technology. The TMS320C40 is the world's first parallel processing DSP with on-chip communication ports for processor-to-processor communication. The 'C40 can execute up to 50 million floating-point operations per second (MFLOPS).

The power supply current requirements (I_{DD}) of the 'C40 vary with the specific application and the device program activity. Additionally, due to the inherent characteristics of CMOS technology, the current requirements depend on clock rates, output loadings, and data patterns.

This application report presents the information you need to determine power supply current requirements for the TMS320C40 under various operating conditions. After you make this determination, you can then calculate the device power dissipation, and, in turn, thermal management requirements.

The major topics discussed in this application report are:

- Capacitive and Resistive Loading
- Basic Current Consumption
  - Current Components
  - Current Dependency
  - Algorithm Partitioning
  - Test Setup Description
- Current Requirements of Internal Components
  - Quiescent
  - Internal Operations
  - Internal Bus Operations
- Current Requirements of Output Driver Components
  - Local or Global Bus
  - DMA
  - Communication Port
  - Data Dependency
- Calculation of Total Supply Current
  - Combining Supply Current Due to All Components
  - Supply Voltage, Operation Frequency, and Temperature Dependencies
  - Design Equation
  - Average Current
  - Thermal Management Considerations
- Example Supply Current Calculations
- Design Considerations
- Summary

Capacitive and Resistive Loading

In CMOS devices, the internal gates swing completely from one supply rail to the other. The voltage change on the gate capacitance requires a charge transfer, and therefore causes power consumption.

The required charge for a gate's capacitance is calculated by the following equation:

$$Q_{gate} = V_{DD} \times C_{gate} \ (coulombs)$$

where:

Q_{gate} is the gate's charge,

V_{DD} is the supply voltage, and

C_{gate} is the gate's capacitance.

Since current is coulombs per second, the current can then be obtained from:

$$I = coul / sec = V_{DD} \times C_{gate} \times Frequency$$

where:

I is the current.

For example, the current consumed by an 80-pF capacitor being driven by a 10-MHz CMOS level square wave is calculated as follows:

$$I = 5 \ (volts) \times 80 \times 10^{-12} (farads) \times 10 \times 10^{6} (charges/sec)$$

$$= 4 \ mA \ @ \ 10 \ MHz$$

Furthermore, if the total number of gates in a device is known, the effective total capacitance can be used to calculate the current for any voltage and frequency. For a given CMOS device, the total number of gates is probably not known, but you can solve for a current at a particular frequency and supply voltage and later use this current to calculate for any supply voltage and operating frequency.

$$I_{device} = V_{DD} \times C_{total} \times f_{CLK}$$

where:

I_{device} is the current consumed by the device,

C_{total} is the total capacitance, and

f_{CLK} is the clock cycle.

Solving for power ($P = V \times I$), the equation becomes:

$$P_{device} = V_{DD}^2 \times C_{total} \times f_{CLK}$$

where:

P_{device} is the power consumed by the device.

In this case, C_{total} includes both internal and external capacitances. C_{total} can be effectively reduced by minimizing power-consuming internal operation and external bus cycles. Bipolar devices, pull-up resistors and other devices consume DC power that adds a constant offset unaffected by f_{CLK}. The effect of these DC losses depends on data, not frequency. This document assumes an all-CMOS approach in which these effects are minimal.

Another source of power consumption is the current consumed by a CMOS gate when it is biased in the linear region. Typically, if a gate is allowed to float, it can consume current. Pull-ups and pull-downs of unused pins are therefore recommended.

Basic Current Consumption

Generally, power supply current requirements are related to the system—for example, operating frequency, supply voltage, temperature, and output load. In addition, because the current requirement for a CMOS device depends on the charging and discharging of node capacitance, factors such as clocking rate, output load capacitance, and data values can be important.

Current Components

The power supply current has four basic components:

- Quiescent
- Internal operations
- Internal bus operations
- External bus operations

Current Dependency

The power supply current consumption depends on many factors. Four are system related:

- Operation frequency
- Supply voltage
- Operating temperature
- Output load

Several others are related to TMS320C40 operation:

- Duty cycle of operations
- Number of buses used
- Wait states
- Cache usage
- Data value

You can calculate the total power supply current requirement for a ’C40 device by using the equation below, which comprises the four basic power supply current components and three system-related dependencies described above.

$$I_{total} = (I_q + I_{iops} + I_{ibus} + I_{xbus}) \times F \times V \times T$$

where:

I_{total} is the total supply current,

I_q is the quiescent current component,

I_{iops} is the current component due to internal operations,

I_{ibus} is the current component due to internal bus usage, including data value and cycle time dependency,

I_{xbus} is the current component due to external bus usage, including data value wait state, cycle time, and capacitive load dependency,

F is a scale factor for frequency,

V is a scale factor for supply voltage, and

T is a scale factor for operating temperature.

This report describes in detail the application of this equation and determination of all the dependencies.

The minimum power supply current requirement is 130 mA. The typical current consumption for most algorithms is 350 mA, as described in the TMS320C40 data sheet, unless excessive data output is being performed.

CAUTION:

The maximum current requirement is 850 mA and occurs only under worst case conditions: writing alternating data (AAAA AAAA to 5555 5555) out of both external buses simultaneously, every cycle, with 80 pF loads and running at 50 MHz.

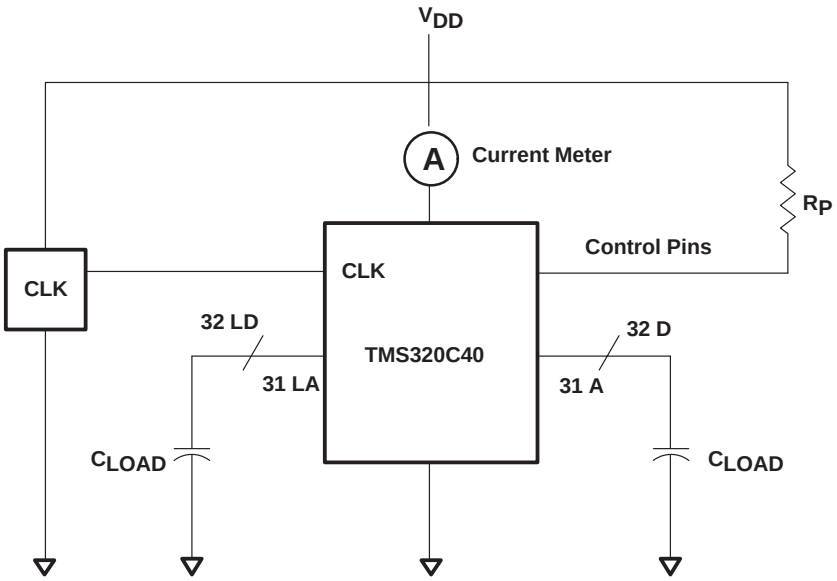
Algorithm Partitioning

Each part of an algorithm has its own pattern with respect to internal and external bus usage. To analyze the power supply current requirement, you must partition an algorithm into segments with distinct concentrations of internal or external bus usage. Analyze each program segment to determine its power supply current requirement. You can then calculate the average power supply current requirement from the requirements of each segment of the algorithm.

Test Setup Description

All TMS320C40 supply current measurements were performed on the test setup shown in Figure 1. The test setup consists of a TMS320C40, capacitive loads on all data and address lines, but no resistive loads. A Tektronix digital multimeter measures the power supply current. Unless otherwise specified, all measurements are made at a supply voltage of 5 V, an input clock frequency of 50 MHz, a capacitive load of 80 pF, and an operating temperature of 25°C. Note that the current consumed by the oscillator and pull-up resistors does not flow through the current meter. This current is considered part of the system's resistive loss (see *Capacitive and Resistive Loading* on page 2).

Figure 1. Test Setup



Current Requirement of Internal Components

The power supply current requirement for internal circuitry consists of three components: quiescent, internal operations, and internal bus operations. Quiescent and internal operations are constants, whereas the internal bus operations component varies with the rate of internal bus usage and the data values being transferred.

Quiescent

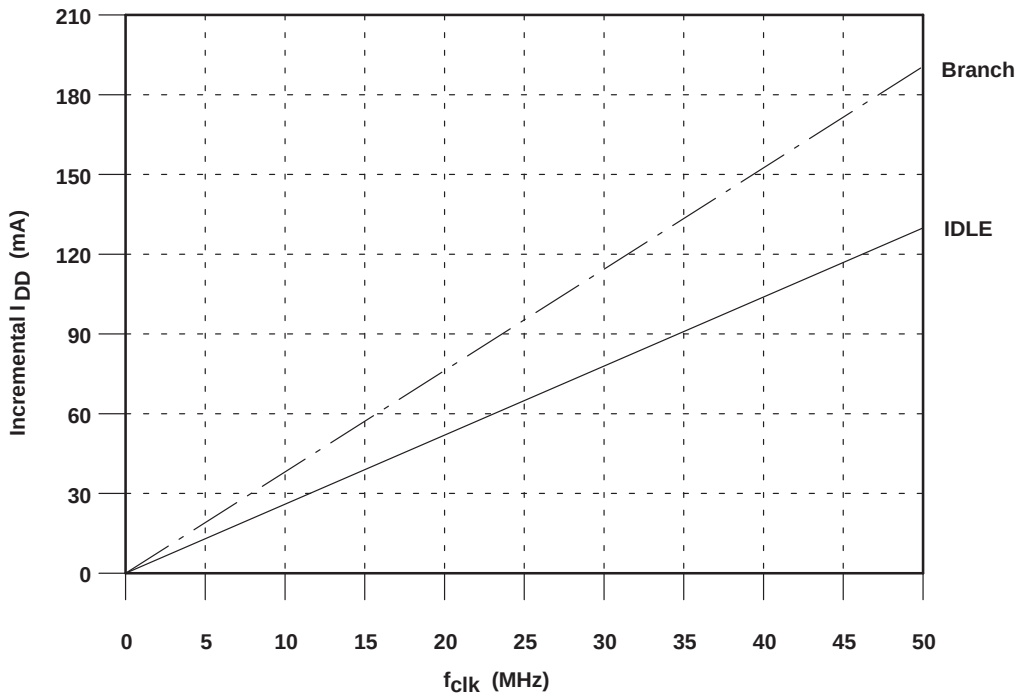
Quiescent refers to the baseline supply current drawn by the TMS320C40 during minimal internal activity, such as executing the IDLE instruction or reset. The quiescent requirement for the TMS320C40 is 130 mA while in IDLE. Examples of quiescent current include:

- Maintaining timer and oscillator
- Executing the IDLE instruction
- Holding the TMS320C40 in reset

Internal Operations

Internal operations include register-to-register multiplication, ALU operations, and branches, but not external bus usage or significant internal bus usage. Internal operations add a constant 60 mA above the quiescent requirement so that the total contribution of quiescent and internal operation is 190 mA. Note, however, that internal and/or external program operations executed via an RPTS instruction do not contribute an internal operations power supply current component. During an RPTS instruction, program fetch activity other than the instruction being repeated is suspended; therefore, power supply current is related only to the data operations performed by the instruction being executed.

Figure 2. Internal and Quiescent Current Components

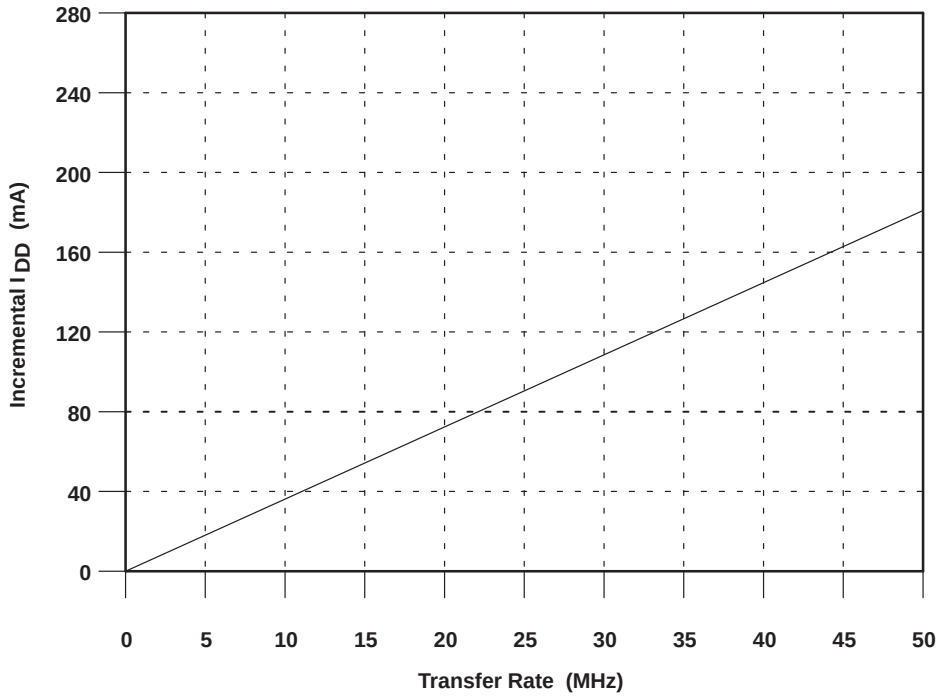


Internal Bus Operations

The internal bus operations include all operations that utilize the internal buses extensively, such as internal RAM accesses every cycle. No distinction is made between internal reads or writes, such as instruction or operand fetches from internal memory, because internally they are equal. Significant use of internal buses adds a data-dependent term to the equation for the power supply current requirement. Recall that switching requires more current. Hence, changing data at high rates requires higher power supply current.

Pipeline conflicts, use of cache, fetches from external wait-state memory, and writes to external wait-state memory all affect the internal and external bus cycles of an algorithm executing on the TMS320C40. Therefore, you must determine the algorithm's internal bus usage in order to accurately calculate power supply current requirements. The TMS320C4x software simulator and XDS emulator both provide benchmarking and timing capabilities that help you determine bus usage.

Figure 3. Internal Bus Current Versus Transfer Rate



The current resulting from internal bus usage varies linearly with transfer rates. Figure 3 shows internal bus current requirements for transferring alternating data (AAAA AAAAh to 5555 5555h) at several frequencies. Note that transfer rates greater than the TMS320C40's MIPS rating are possible because of the internal parallelism.

The data set AAAA AAAAh to 5555 5555h exhibits the maximum internal bus current for data transfer operations. The current required for transferring other data patterns may be derated accordingly, as described later in this subsection.

As the transfer rate decreases (that is, transfer cycle time increases) the incremental I_{DD} approaches 0 mA. This figure represents the incremental I_{DD} due to internal bus operations and is added to quiescent and internal operations current values.

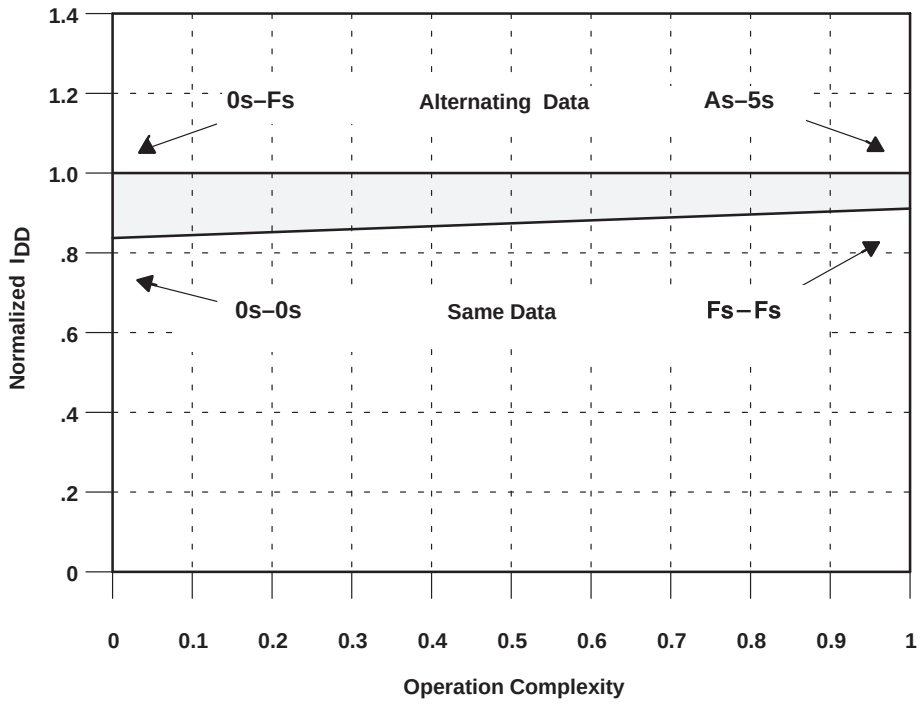
For example, the maximum transfer rate corresponds to three accesses every cycle (one program fetch and two data transfers) or an effective one-third H1 transfer cycle time. At this rate, 178 mA is added to the quiescent (130 mA) and internal operation (60 mA) current values for a total of 368 mA.

Figure 3 shows the internal bus current requirement when transferring As followed by 5s for various transfer rates. Figure 4 shows the data dependence of the internal bus current requirement when the data is other than As followed by 5s. The trapezoidal region bounds all possible data values transferred. The lower line represents the scale factor for transferring the same data. The upper line represents the scale factor for transferring alternating data (all 0s to all Fs or all As to all 5s, etc.).

The possible permutation of data values is quite large. The term relative data complexity refers to a relative measure of the extent to which data values are changing and the extent to which the number of bits are

changing state. Therefore, relative data complexity ranges from 0, signifying minimal variation of data, to a normalized value of 1, signifying greatest data variation.

Figure 4. Internal Bus Current Versus Data Complexity Derating Curve



If a statistical knowledge of the data exists, Figure 4 can be used to determine the exact power supply requirement on the basis of internal bus usage. For example, Figure 4 indicates a 89.5% scale factor when all Fs (FFFF FFFFh) are moved internally every cycle with two accesses per cycle (80 Mbytes per second). Multiplying this scale factor by 178 mA (from Figure 3) yields 159 mA due to internal bus usage. Therefore, an algorithm running under these conditions requires about 349 mA of power supply current (130 + 60 + 159).

Since a statistical knowledge of the data may not be readily available, a nominal scale factor may be used. The median between the minimum and maximum values at 50% relative data complexity yields a value of 0.93 and can be used as an estimate of a nominal scale factor. Therefore, this nominal data scale factor of 93% can be used for internal bus data dependency, adding 165.5 mA to 130 mA (quiescent) and 60 mA (internal operations) to yield 355.5 mA. As an upper bound, assume worst case conditions of three accesses of alternating data every cycle, adding 178 mA to 130 mA (quiescent) and 60 mA (internal operations) to yield 368 mA.

Current Requirement of Output Driver Components

The output driver circuits on the TMS320C40 are required to drive significantly higher DC and capacitive loads than internal device logic drivers. Because of this, output drivers impose higher supply current requirements than other sections of circuitry in the device.

Accordingly, the highest values of supply current are exhibited when external writes are being performed at high speed. During read cycles, or when the external buses are not being used, the TMS320C40 is not driving the data bus; this eliminates a significant component of the output buffer current. Furthermore, in many typical cases, only a few address lines are changing, or the whole address bus is static. Under these conditions, an insignificant amount of supply current is consumed. Therefore, when no external writes are being performed or when writes are performed infrequently, current due to output buffer circuitry can be ignored.

When external writes are being performed, the current required to supply the output buffers depends on several considerations:

- Data pattern being transferred,
- Rate at which transfers are being made,
- Number of wait states implemented (because wait states affect rates at which bus signals switch), and
- External bus DC and capacitive loading.

External bus operations involve external writes to the device and constitute a major power supply current component. The power supply current for the external buses, made up of four components, is summarized in the following equation:

$$I_{xbus} = (I_{base\ local} + I_{local}) + (I_{base\ global} + I_{global})$$

where:

$I_{base\ local/global}$ is the current consumed by the internal driver and pin capacitance,

I_{local} is the local bus current component, and

I_{global} is the global bus current component.

The remainder of this section describes in detail the calculation of external bus current requirements. Note that the DMA current component (I_{DMA}) and communication port current component (I_{CP}) should be included in the calculation of I_{xbus} if they are used in the operations.

Local or Global Bus

The current due to bus writes varies with write cycle time. As discussed in the previous section, to obtain accurate current values, you must first determine the rate and timing for write cycles to external buses by analyzing program activity, including any pipeline conflicts that may exist. To do this, you can use information from the TMS320C4x emulator or simulator as well as the *TMS320C4x User's Guide*. In your analysis, you must account for effects from the use of cache, because use of cache can affect whether or not instructions are fetched from external memory.

When evaluating external write activity in a given program segment, you must consider whether or not a particular level of external write activity constitutes significant activity. If writes are being performed at

a slow enough rate, they do not impact supply current requirements significantly and can be ignored. This is the case, however, only if writes are being performed at very slow rates on either the local or global bus.

When bus-write cycle timing has been established, Figure 5 can be used to determine the contribution to supply current due to bus activity. Figure 5 shows values of current contribution from the local or global bus for various transfer rates. This data was gathered when alternating values of 55555555h and AAAAAAAAh were written at a capacitive load of 80 pF per output signal line. This condition exhibits the highest current values on the device. The values presented in the figure represent the incremental current contributed by the local or global bus output driver circuitry under the given conditions. Current values obtained from this graph are later scaled and added to several other current terms to calculate the total current for the device. As indicated in the figure, the lower limit $I_{base} = I_q + I_{iops} + I_{ibus}$ is essentially I_{total} for transfer rates less than 1 Mword/second.

Figure 5. Local/Global Bus Current Versus Transfer Rate and Wait States

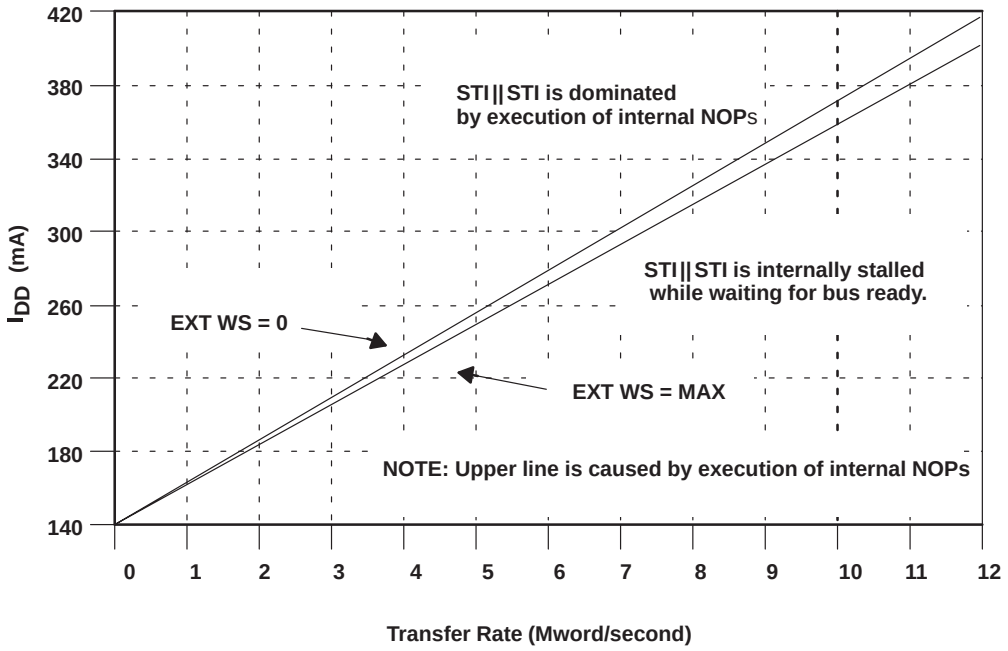
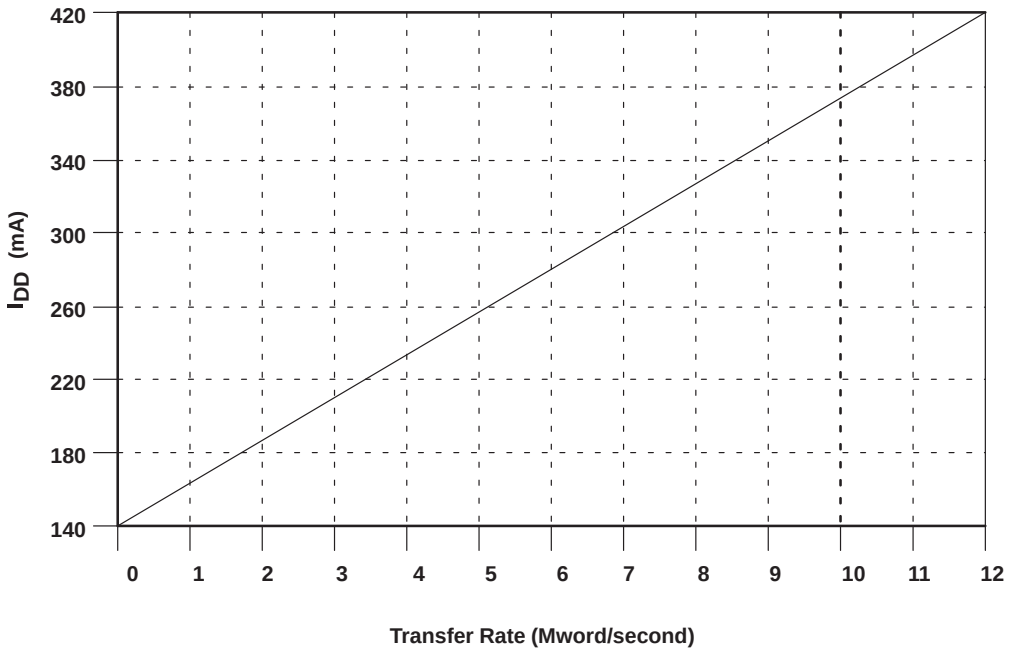


Figure 5 demonstrates a feature of the 'C40's external bus architecture known as a posted write. In general, data is written to a latch (or one deep FIFO) and held by the bus until the bus cycle is complete. Since the CPU may not require that bus again for some time, the CPU is free to perform operations on other buses until a conflict occurs. Conflicts include DMA, a second write, or a read to the bus.

In Figure 5, the upper line is applicable when STI || STI is not dominated by execution of internal NOPs and the external wait state is equal to zero. The lower line shows when STI || STI is internally stalled while waiting for the external bus to go ready because of wait states. The addition of NOPs between successive STI || STI operations contributes to internal bus current and therefore does not result in the lowest possible current.

Figure 6. Local/Global Bus Current Versus Transfer Rate at Zero Wait States



To further illustrate the relationship of current and write cycle time, Figure 6 shows the characteristics of current for various numbers of cycles between writes for zero wait states. The information on this graph can be used to obtain more precise values of current whenever zero wait states are used. Table 1 lists the number of cycles used for software generated wait states.

Table 1. Wait State Timing Table

Wait State	Read Cycles	Write Cycles
0	1	2
1	2	3
2	3	4
3	4	5

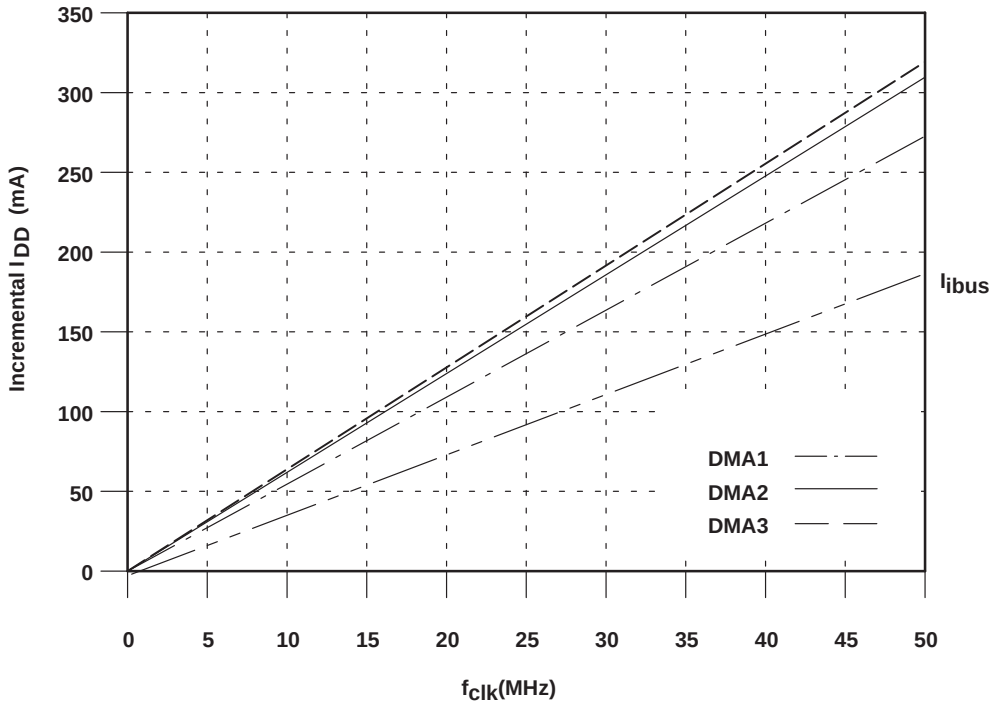
Once a current value has been obtained from Figure 5 or Figure 6, this value can be scaled by a data dependency factor if necessary, as described on page 14. This scaled value is then summed along with several other current terms to determine the total supply current.

DMA

Using DMA to transfer data consumes power that is data dependent. The current resulting from DMA bus usage (I_{DMA}) varies linearly with the transfer rate. Figure 7 shows DMA bus current requirements for transferring alternating data (AAAA AAAAh to 5555 5555h) at several transfer rates; it also shows that

current consumption increases when more DMA channels are used. However, as more DMA channels are used, the incremental change in current diminishes as the internal DMA bus becomes saturated. Note that DMA current is superimposed over I_{ibus} (internal bus) value.

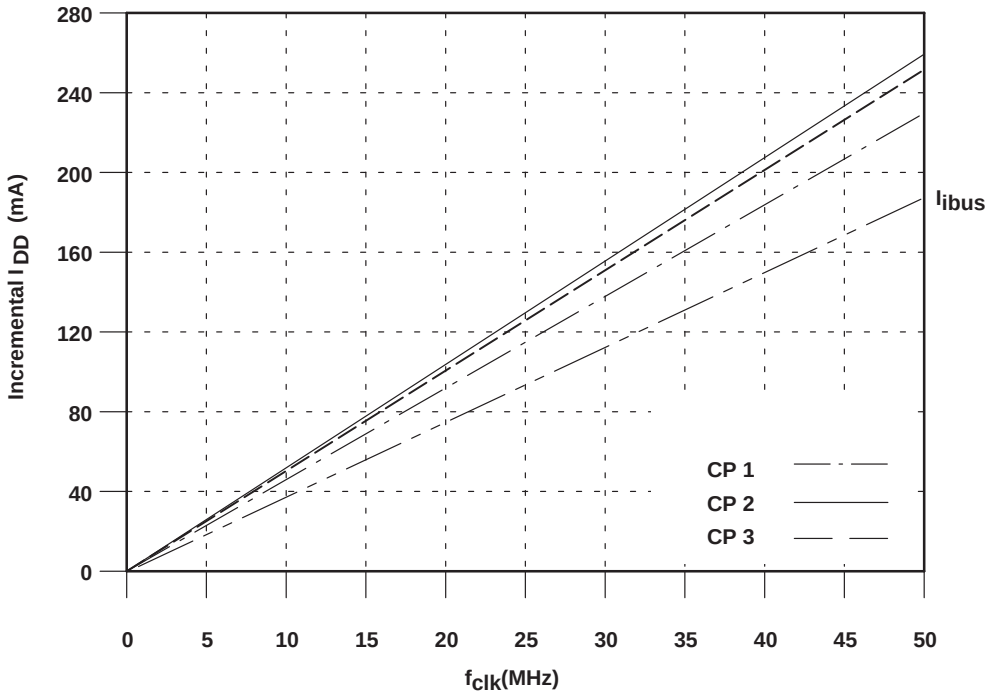
Figure 7. DMA Bus Current Versus Clock Rate



Communication Port

Communication port operations add a data-dependent term to the equation for the current requirement. The current resulting from communication port operation (I_{CP}) varies linearly with the transfer rate. Figure 8 shows communication port operation current requirements for transferring alternating data (AAAA AAAAh to 5555 5555h) at several transfer rates; it also shows that current consumption increases when more communication port channels are used. Similar to the DMA bus current consumption, adding communication ports eventually saturates the peripheral bus as more channels are added.

Figure 8. Communication Port Current Versus Clock Rate



Note that since the communication ports are intended to communicate with other TMS320C40 communication ports over short distances, no additional capacitive loading was added. In this case, the transmission distance is about 6 inches without additional 80-pF loads. Note that communication port current is superimposed over I_{ibus} value.

Data Dependency

Data dependency of the current for the local and global buses is expressed as a scale factor that is a percentage of the maximum current exhibited by either of the two buses.

Figure 9. Local/Global Bus Current Versus Data Complexity

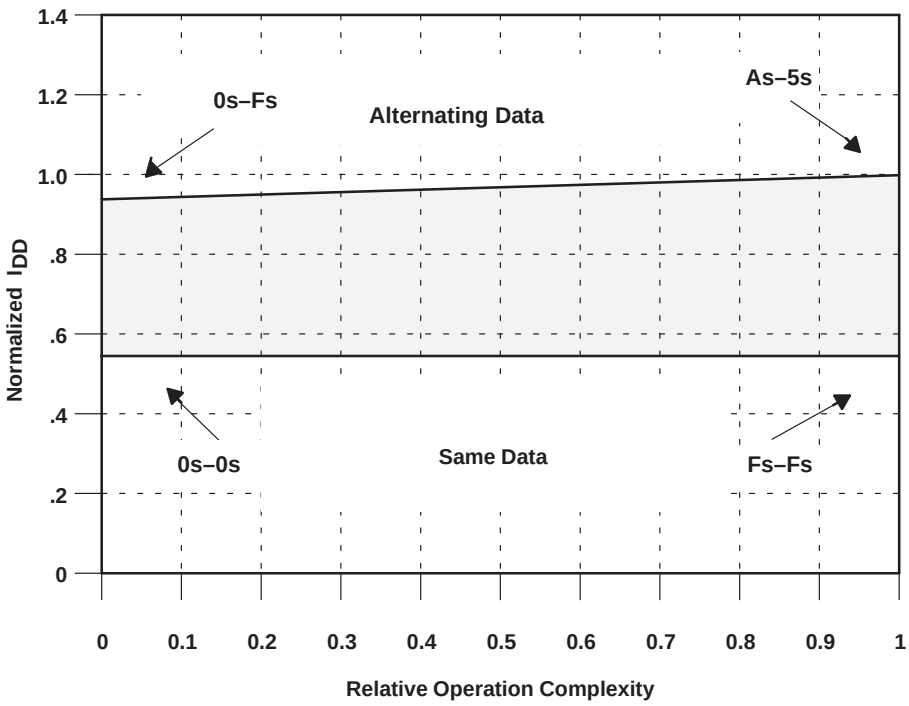


Figure 9 shows normalized weighting factors that can be used to scale current requirements on the basis of patterns in data being written on the external buses. The range of possible weighting factors forms a trapezoidal pattern bounded by extremes of data values. As the figure shows, the minimum current occurs when all zeros are written, while the maximum current occurs when alternating 5555 5555h and AAAA AAAAh are written. This condition results in a weighting factor of 1, which corresponds to using the values from Figure 5 and/or Figure 6 directly.

As with internal bus operations, data dependencies for the external buses are well defined, but accurate prediction of data patterns is often either impossible or impractical. Therefore, unless you have precise knowledge of data patterns, you should use an estimate of a median or average value for the scale factor. Assuming that data will be neither 5s and As nor all 0s and will be varying randomly, then a value of 0.80 is appropriate. Otherwise, if you prefer a conservative approach, you can use a value of 1.0 as an upper bound.

Regardless of the approach taken for scaling, once you determine the scale factor for the buses, apply this factor to the current values you determined with the graphs in the *Local and Global Buses* section, page 11.

For example, if a nominal scale factor of 0.80 for the buses is assumed, the current contribution from the two buses is as follows:

Local or Global : $0.80 \times 133\ mA = 106.4\ mA$

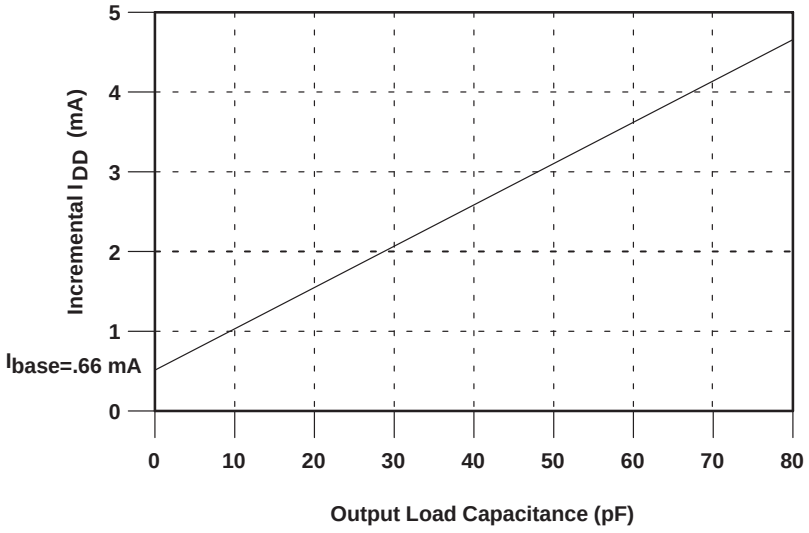
Capacitive Loading Dependence

Once cycle timing and data dependencies have been accounted for, capacitive loading effects should be calculated and applied. Figure 10 shows the current values obtained above as a function of actual load capacitance if the load capacitance presented to the buses is less than 80 pF.

In the previous example, if the load capacitance is 20 pF instead of 80 pF, the actual pin current would be 1.66 mA.

While the slope of the line in Figure 10 can be used to interpolate scale factors for loads greater than 80 pF, the TMS320C40 is specified to drive output loads less than 80 pF; interface timings cannot be guaranteed at higher loads. With data dependency and capacitive load scale factors applied to the current values for local and global buses, the total supply current required for the device for a particular application can be calculated, as described in the next section.

Figure 10. Pin Current Versus Output Load Capacitance (10 MHz)



Calculation of Total Supply Current

The previous sections have discussed currents contributed by different sources on TMS320C40. Because determinations of actual current values are unique and independent for each source, each current source was discussed separately. In an actual application, however, the sum of the independent contributions determines the total current requirement for the device. This total current value is exhibited as the total current supplied to the device through all of the V_{DD} inputs and returned through the V_{SS} connections.

Note that numerous V_{DD} and V_{SS} pins on the device are routed to a variety of internal connections, not all of which are common. Externally, however, all of these pins should be connected in parallel to 5 V and ground planes, providing very low impedance.

As mentioned previously, because of the inherent differences in operations between program segments, it is usually appropriate to consider current for each of the segments independently. In this way, peak current requirements are readily obtained. Further, you can make average current calculations to use in determining heating effects of power dissipation. These effects, in turn, can be used to determine thermal management considerations.

Combining Supply Current Due to All Components

To determine the total supply current requirements for any given program activity, calculate each of the appropriate components and combine them in the following sequence:

1. Start with 130 mA quiescent current requirement.
2. Add 60 mA for internal operations unless the device is dormant, such as when executing IDLE or using an RPTS instruction to perform internal and/or external bus operations (see *Internal Operations* section on page 7). Internal or external bus operations executed via RPTS do not contribute an internal operations power supply current component. Therefore, current components in the next two steps may still be required, even though the 60 mA is omitted.
3. If significant internal bus operations are being performed (see *Internal Bus Operations* on page 8), add the calculated current value.
4. If external writes are being performed at high speed (see *Current Requirements of Output Driver Component*s on page 11), then add the values calculated for local and global bus current components.
5. Add DMA and communication port current requirements if they are used.

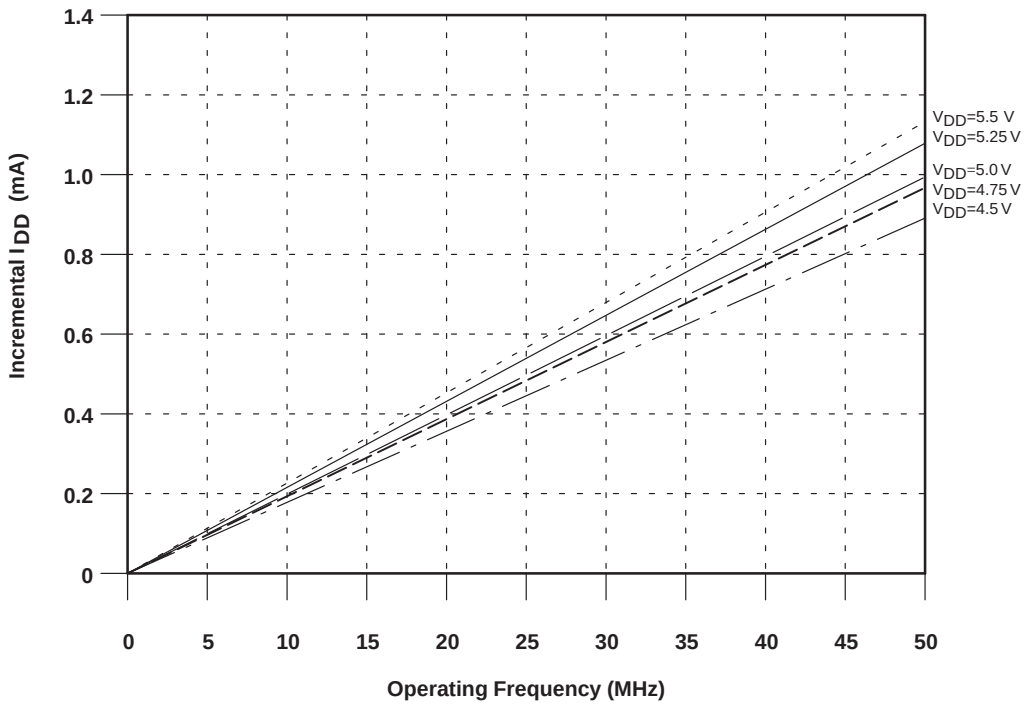
The current value resulting from summing these components is the total device current requirement for a given program activity.

Supply Voltage, Operating Frequency, and Temperature Dependencies

Three additional factors that affect current requirements are supply voltage level, operating temperature, and operating frequency. However, these considerations affect total supply current, not specific components (that is, internal or external bus operations). Note that supply voltages, operating temperature, and operating frequency must be maintained within required device specifications.

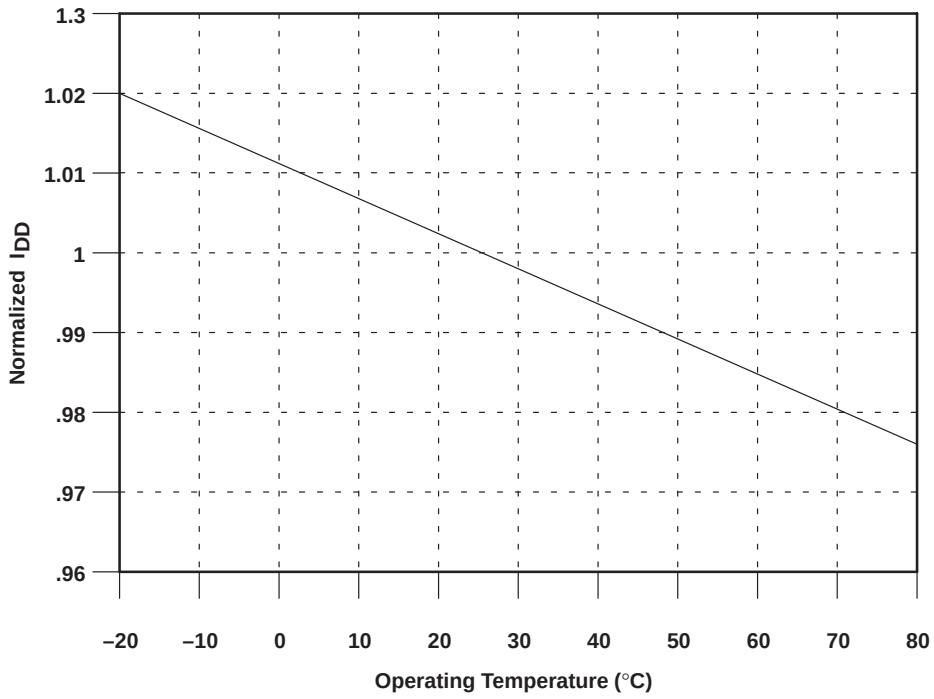
The scale factor for these dependencies is applied in the same manner as discussed in previous sections, once the total current for a particular program segment has been determined. Figure 11 shows the relative scale factors to be applied to the supply current values as a function of both V_{DD} and operating frequency.

Figure 11. Current Versus Frequency and Supply Voltage



Power supply current consumption does not vary significantly with operating temperature. However, you can use a scale factor of 2% normalized I_{DD} per 50°C change in operating temperature to derate current within the specified range noted in the TMS320C40 data sheet.

Figure 12. Change in Operating Temperature (°C)



This temperature dependence is shown graphically in Figure 12. Note that a temperature scale factor of 1.0 corresponds to current values at 25°C, which is the temperature at which all other references in the document are made.

Design Equation

The procedure for determining the power supply current requirement can be summarized in the following equation:

$$I_{total} = (I_q + I_{iops} + I_{ibus} + I_{xbus}) \times F \times V \times T$$

where:

I_q = 130 mA,

I_{iops} = 40 mA,

$I_{ibus} = D1 \times f1$ (see Table 2),

$I_{xbus} = I_{base} + I_{local} + I_{global} + I_{DMA} + I_{CP}$

with:

$I_{local} = D2 \times C2 \times F2$ (see Table 2),

$I_{global} = D3 \times C3 \times F3$ (see Table 2),

I_{DMA} = *f4* (see Table 2),

I_{CP} = *f5* (see Table 2),

and:

F is a scale factor for frequency,

V is a scale factor for supply voltage, and

T is a scale factor for operating temperature.

Table 2 describes the symbols used in the power supply current equation and gives either the value or the figure number from which the value can be obtained.

Table 2. Current Equation Symbols

Symbol	Description	Value
I_q	Quiescent Current	130 mA
I_{iops}	Internal Operations Current	60 mA
I_{bus}	Internal Bus Operations Current	
D_1	Internal Bus Data Scale Factor	Figure 4
f_1	Internal Bus Current Requirement	Figure 3
I_{xbus}	External Bus Operations Current	
$I_{base\ local}$	Local Bus Base Current	42.5 mA
I_{local}	Local Bus Operations Current	
D_2	Local Bus Data Scale Factor	Figure 9
C_2	Local Bus Cap Load Scale Factor	Figure 10
f_2	Local Bus Current Requirement	Figure 5 or Figure 6
$I_{base\ global}$	Global Bus Base Current	42.5 mA
I_{global}	Global Bus Operations Current	
D_3	Global Bus Data Scale Factor	Figure 9
C_3	Global Bus Cap Load Scale Factor	Figure 10
f_3	Global Bus Current Requirement	Figure 5 or Figure 6
I_{DMA}	DMA Bus Operations Current	
f_4	DMA Bus Current Requirement	Figure 7
I_{CP}	Comm Port Operations Current	
f_5	Comm Port Current Requirement	Figure 8
FV	Freq/Supply Voltage Scale Factor	Figure 11
T	Temperature Scale Factor	Figure 12

Average Current

Over the course of an entire program, some segments will typically exhibit significantly different levels of current for different durations. For example, a program may spend 80% of its time performing internal operations and draw a current of 250 mA; it may spend the remaining 20% of its time performing writes at full speed to both buses and drawing 790 mA.

While knowledge of peak current levels is important in order to establish power supply requirements, some applications require information about average current. This is particularly significant if periods of high peak current are short in duration. You can obtain average current by performing a weighted sum of the current due to the various independent program segments over time. You can calculate the average current for the example in the previous paragraph as follows:

$$I = 0.8 \times 250\ mA + 0.2 \times 790\ mA = 358\ mA$$

Using this approach, you can calculate average current for any number of program segments.

Thermal Management Considerations

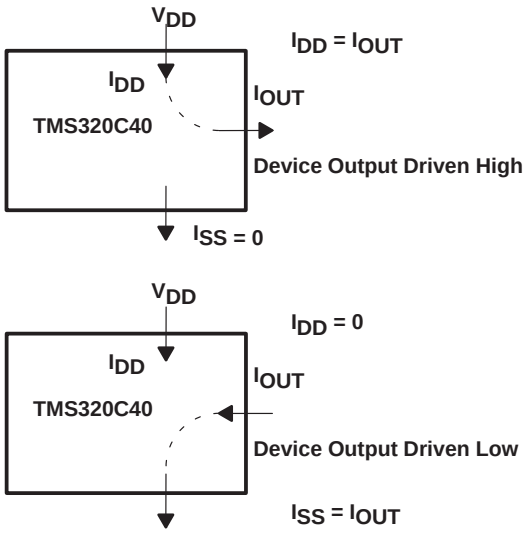
Heating characteristics of the TMS320C40 are dependent upon power dissipation, which, in turn, is dependent upon power supply current. When making thermal management calculations, you must consider the manner in which power supply current contributes to power dissipation and to the TMS320C40 package thermal characteristics' time constant.

Depending on the sources and destinations of current on the device, some current contributions to I_{DD} do not constitute a component of power dissipation at 5 volts. That is to say, the TMS320C40 may be acting only as a switch, in which case, the voltage drop is across a load and not across the 'C40. If the total current flowing into V_{DD} is used to calculate power dissipation at 5 volts, erroneously large values for package power dissipation will be obtained. The error occurs because the current resulting from driving a logic high level into a DC load appears only as a portion of the current used to calculate system power dissipation due to V_{DD} at 5 volts. Power dissipation is defined as:

$$P = V \times I$$

where P is power, V is voltage, and I is current. If device outputs are driving any DC load to a logic high level, only a minor contribution is made to power dissipation because CMOS outputs typically drive to a level within a few tenths of a volt of the power supply rails. If this is the case, subtract these current components out of the TMS320C40 supply current value and calculate their contribution to system power dissipation separately (see Figure 13).

Figure 13. Load Currents



Furthermore, external loads draw supply current (I_{DD}) only when outputs are being driven high, because when outputs are in the logic zero state, the device is sinking current through V_{SS}, which is supplied from an external source. Therefore, the power dissipation due to this component will not contribute through I_{DD} but will contribute to power dissipation with a magnitude of:

$$P = V_{OL} \times I_{OL}$$

where V_{OL} is the low-level output voltage and I_{OL} is the current being sunk by the output, as shown in Figure 13. The power dissipation component due to outputs being driven low should be calculated and added to the total power dissipation.

When outputs with DC loads are being switched, the power dissipation components from outputs being driven high and outputs being driven low should be averaged and added to the total device power dissipation. Power components due to DC loading of the outputs should be calculated separately for each program segment before average power is calculated.

Note that unused inputs that are left unconnected may float to a voltage level that will cause the input buffer circuits to remain in the linear region, and therefore contribute a significant component to power supply current. Accordingly, if you want absolute minimum power dissipation, you should make any unused inputs inactive by either grounding or pulling them high. If several unused inputs must be pulled high, they can be pulled high together through one resistor to minimize component count and board space.

When you use power dissipation values to determine thermal management considerations, use the average power unless the time duration of individual program segments is long. The thermal characteristics of the TMS320C40 in the 325-pin PGA package are exponential in nature with a time constant on the order of minutes. Therefore, when subjected to a change in power, the temperature of the device package will require several minutes or more to reach thermal equilibrium.

If the duration of program segments exhibiting high power dissipation values is short (on the order of a few seconds) in comparison to the package thermal characteristics' time constant, use average power calculated

in the same manner as average current described in the previous section. Otherwise, calculate maximum device temperature on the basis of the actual time required for the program segments involved. For example, if a particular program segment lasts for 7 minutes, the device essentially reaches thermal equilibrium due to the total power dissipation during the period of device activity.

Note that the average power should be determined by calculating the power for each program segment (including all considerations described above) and performing a time average of these values, rather than simply multiplying the average current by V_{DD}, as determined in the previous subsection.

Calculate specific device temperature by using the TMS320C40 thermal impedance characteristics included in the TMS320C40 data sheet in Chapter 14 of the *TMS320C4x User's Guide*.

Example Supply Current Calculations

An FFT represents a typical DSP algorithm. The FFT code in Appendix A processes data in the RAM blocks. The entire algorithm consists mainly of internal bus operations and hence includes quiescent and, in general, internal operations. At the end of the processing, the results are written out on the global and local bus. Therefore, the algorithm exhibits a higher current requirement during the write portion where the external bus is being used significantly.

Processing

The processing portion of the algorithm is 95% of the total algorithm. During this portion, the power supply current is required for the internal circuitry only. Data is processed in several loops that make up the majority of the algorithm. During these loops, two operands are transferred on every cycle. The current required for internal bus usage, then, is 60 mA (from Figure 3). The data is assumed to be random. A data value scale factor of 0.93 is used (from Figure 4). This value scales 60 mA, yielding 55.8 mA for internal bus operations. Adding 55.8 mA to the quiescent current requirement and internal operations current requirement yields a current requirement of 245.8 mA for the major portion of the algorithm.

$$I = I_q + I_{iops} + I_{ibus}$$

$$I = 130\ mA + 60\ mA + (60\ mA)\ (0.93)$$

$$= 245.8\ mA$$

Data Output

The portion of the algorithm corresponding to writing out data is approximately 5% of the total algorithm. Again, the data that is being written is assumed to be random. From Figure 4 and Figure 10, scale factors of 0.93 and 0.8 are used for derating due to data value dependency for internal and local buses, respectively. During the data dump portion of the code, a load and a store are performed every cycle; however, the parallel load/store instruction is in an RPTS loop. Therefore, there is no contribution due to internal operations, because the instruction is fetched only once. The only internal contributions are due to quiescent and internal bus operations. Figure 5 indicates a 23-mA current contribution due to writes every available cycle. Therefore, the total contribution due to this portion of the code is:

$$I = I_q + I_{ibus} + I_{xbus}$$

or

$$I = 130\ mA + (60\ mA)\ (0.93) + 85\ mA + (23\ mA)\ (0.8)$$

$$= 289.2\ mA$$

Average Current

The average current is derived from the two portions of the algorithm. The processing portion took 95% of the time and required about 245.8 mA; the data dump portion took the other 5% and required about 411.6 mA. The average is calculated as:

$$I_{avg} = (0.95)\ (245.8\ mA) + (0.05)\ (289.2\ mA)$$

$$= 247.97\ mA$$

From the thermal characteristics specified in the *TMS320C4x User's Guide*, it can be shown that this current level corresponds to a case temperature of 28°C. This temperature meets the maximum device specification of 85°C and hence requires no forced air cooling.

Experimental Results

A photograph of the power supply current for the FFT, using a 40-MHz system clock, is shown in Appendix A. During the FFT processing, the current varied between 190 and 220 mA. The current during external writes had a peak of 230 mA, and the average current requirement as measured on a digital multimeter was 205 mA. Scaling those results to the 50-MHz calculations yielded results that were close to the actual measured power supply current.

Design Considerations

Designing systems for minimum power dissipation involves reducing device operating current requirements due to signal switching rate, capacitive loading, and other effects. Selective consideration of these effects makes it possible to optimize system performance while minimizing power consumption. This section describes current reduction techniques based on operating current dependencies of the device as discussed in previous sections of this document.

System Clock and Signal Switching Rates

Since current (and therefore power) requirements of CMOS devices are directly proportional to switching frequency, one potential approach to minimizing operating power is to minimize system clock frequency and signal switching rates. Although performance is often directly proportional to system clock and signal switching rates, tradeoffs can be made in both areas to achieve an optimal balance between power usage and performance in the design of a system.

If reducing power is a primary goal, and a given system design does not have particularly demanding performance requirements, the system clock rate can be reduced with the corresponding savings in power. Minimum power is realized when system clock rates are only as fast as necessary to achieve required system performance. Additionally, if overall system clock rates cannot be reduced, an alternative approach to power reduction is to reduce clock speed wherever possible during periods of inactivity.

Also, the appropriate choice of clock generation approach will ensure minimum system power dissipation. The use of an external oscillator rather than the on-chip oscillator can result in lower power device and system power dissipation levels. As described previously, the internal oscillator can require as much as 10 mA when operating at 40 MHz. If you use an external oscillator that requires less than 10 mA for clock generation, overall system power is reduced.

When considering switching rates of signals other than the system clock, the main consideration is to minimize switching. Specifically, any unnecessary switching should be avoided. Outputs or inputs that are unused should either be disabled, tied high, or grounded, whichever is appropriate. Additionally, outputs connected to external circuitry should drive other power dissipation elements only when absolutely necessary.

Capacitive Loading of Signals

Current requirements are also directly proportional to capacitive loading. Therefore, all capacitive loading should be minimized. This is especially significant for device outputs.

The approaches to minimize capacitive loading are consistent with efficient PC board layout and construction practices. Specifically, signal runs should be as short as possible, especially for signals with high switching rates. Also, signals should not run long distances across PC boards to edge connectors unless absolutely necessary.

Note that the buffering of device outputs that must drive high capacitive loads reduces supply current for the TMS320C40, but this current is translated to the buffering device. Whether or not this is a valid tradeoff must be determined at the system level. The two main considerations are: 1) whether the power required by the buffers is more or less than the power required from the ’C40 to drive the load in question, and 2) whether or not off-loading the power to the buffers has any implications with respect to system power-down modes. It may be desirable to use buffers to drive high capacitive loads, even though they may require more current than the TMS320C40, especially in cases where part of the system may be powered down but the TMS320C40 is still required to interface to other low capacitance loads.

DC Component of Signal Loading

In order to achieve lowest device current requirements, the internal and external DC load component of device input and output signal loading must also be minimized .

Any device inputs that are unused and left floating may cause excessively high DC current to be drawn by their input buffer circuitry. This occurs because if an input is left unconnected, the voltage on the input may float to a level that causes the input buffer to be biased at a point within its range of linear operation. This can cause the input buffer circuit to draw a significant DC current directly from V_{DD} to ground. Therefore, any unused device inputs should be pulled up to V_{DD} via a resistor pull-up of nominally 20 kΩ, or driven high with an unused gate. Input-only pins that are not used can be pulled up in parallel with other inputs of the same type with a single gate or resistor to minimize system component count. In this case, up to 15 or more standard device inputs can be pulled up with a single resistor.

Any device I/O pins that are unused should be selected as outputs. This avoids the requirement for pull-ups (to ensure that the I/O input stage is not biased in the linear region) and therefore eliminates an unnecessary current component.

For any device output, any DC load present is directly reflected in the system's power supply current. Therefore, DC loading of outputs should be reduced to a minimum. If DC currents are being sourced from the address bus outputs, the address bus should be set to a level that minimizes the current through the external load. This can be accomplished by performing a dummy read from an external address.

For I/O pins that must be used in both the input and output modes, individual pull-up resistors of nominally 20 kΩ should be used to ensure minimum power dissipation if these pins are not always driven to a valid logic state. This is particularly true of the data bus pins. When the bus is not being driven explicitly, it is left floating, which can cause excessively high currents to be drawn on the input buffer section of all 64 bits of the bus. In this case, because all 64 data bus bits are normally used independently in most applications, each data bus pin should be pulled up with a separate resistor for minimum power.

Summary

The power supply current requirement for the TMS320C40 cannot be expressed simply in terms of operating frequency, supply voltage, and output load capacitance. A more complete specification, one based on device functionality, must be used to determine a more accurate power supply current requirement. This application note presents the information you need to determine power supply specifications. The specification is based on a knowledge of the algorithm and how it operates on the TMS320C40 in terms of internal and external bus usage. As devices become more complex, the approach presented in this document must be applied.

The power supply current requirement for the TMS320C40 depends on device functionality and system parameters. The components of current related to device functionality are those due to quiescent current, internal operations, internal bus operations, and external bus operations. The dependencies related to system parameters are those due to operating frequency, supply voltage, output load capacitance, and operating temperature. The typical power supply current requirement is 350 mA, and the minimum, or quiescent, is 130 mA.

Appendix A

FFT Assembly Code

```
            .globl FFT                  ; Entry point for execution
            .globl N                    ; FFT size
            .globl M                    ; LOG2 (N)
            .globl SINE                 ; Address of sine table
            .globl INP
            .globl OUTPUT
            .globl INPUT

            .sect ”freq”                ; Memory with output data
OUTP        .word 0

            .text
*           INITIALIZE
FFTSIZ      .word  N
LOGFFT      .word  M
SINTAB      .word  SINE
INPUT       .word  INP
OUTPUT      .word  OUTP
FFT:        LDP    FFTSIZ               ; Command to load data page pointer

            LDI    @FFTSIZ, R7          ; R7 = N2
            LSH3   –2,R7,IR1            ; IR1 = N/4, pointer for SIN/COS
                                        ; table
            LDI    @LOGFFT, R9          ; R9 holds the remain stage number
            LSH3   1,R7,IR0             ; IR0 = 2*NI (because of real/imag)
            LDI    1,R8                 ; Initialize repeat counter of first
                                        ; loop
            LDI    1,AR5                ; Initialize IE index (AR5 = IE)
            LDI    @INPUT, R10          ; R10 points to X(I)

            LSH    1,R7                 ; N2 = N2/2
            SUBI3  1,R8,RC              ; RC should be 1 less than desired #
*           OUTER LOOP
LOOP:       RPTBD  BLK1                 ; Setup for first loop
            LSH    –1,R7                ; N2 = N2 / 2
            LDI    R10,AR0              ; AR0 points to X(I)
```

```
          ADDI   R7,AR0, AR2          ; AR2 points to X(I)
*         FIRST LOOP
          ADDF   *AR0,*AR2,R0         ; R0 = X(I) + X(L)
          SUBF   *AR2++,*AR0++,R1     ; R1 = X(I) – X(L)
          ADDF   *AR2,*AR0,R2         ; R2 = Y(I) + Y(L)
          SUBF   *AR2,*AR0,R3         ; R3 = Y(I) – Y(L)
          STF    R2,*AR0––            ; Y(I) = R2 AND ...
||        STF    R3,*AR2––            ; Y(L) = R3
BLK1      STF    R1,*AR0++(IR0)       ; X(I) = R0 AND ...
||        STF    R1,*AR2++(IR0)       ; X(L) = R1 AND AR0, 2 = AR0, 2 + 2*N

*         IF THIS IS THE LAST STAGE,YOU ARE DONE
          SUBI   1,R9
          BZD    END
*         MAIN INNER LOOP
          LDI    2,AR1                ; INIT LOOP COUNTER FOR INNER LOOP
          LDI    @SINTAB,AR4          ; INITIALIZE IA INDEX (AR4 = IA)
          ADDI   AR5,AR4              ; IA = IA + E; AR4 POINTS TO COSINE
          ADDI   R10,AR1,AR0          ; (X(I), Y(I)) POINTER
          SUBI   1,R8,RC              ; RC SHD BE 1 LESS THAN DESIRED #
INLOP:    RPTB   BLK2                 ; Setup for second loop
          ADDI   2,AR1                ; Increase inner loop counter
          ADDI   R7,AR0,AR2           ; (X(L), Y(L)) pointer
          LDF    *AR4,R6              ; R6 = SIN

*         SECOND LOOP
          SUBF   *AR2,*AR0,R2         ; R2 = X(I) – X(L)
          SUBF   *+AR2,*+AR0,R1
*                                     ; R1 = Y(I) – X(L)
          MPYF   R2,R6,R0             ; R0 = R2 * SIN and
||        ADDF   *+AR2,*+AR0,R3
                                      ; R3 = Y(I) + Y(L)
||        STF    R3,*+AR0             ; Y(I) = Y(I) + Y(L)
          SUBF   R0,R3,R4             ; R4 = R1 * COS – R2 * SIN
          MPYF   R1,R6,R0             ; R0 = R1 * SIN and...
||        ADDF   *AR2,*AR0,R3         ; R3 = X(I) + X(L)

          MPYF   R2,*+AR4(IR1),R3     ; R3 = R2 * COS and...
||        STF    R3,*+AR0             ; Y(I) = Y(I) + Y(L)
```

```
            SUBF   R0,R3,R4             ; R4 = R1 * COS – R2 * SIN
            MPYF   R1,R6,R0             ; R0 = R1 * SIN and ...
            ADDF   *AR2,*AR0,R3         ; R3 = X(I) + X(L)
            MPYF   R2,*+AR4(IR1),R3     ; R3 = R2 * COS and ...
||          STF    R3,*+AR0++(IR0)
                                        ; X(I) = X(I) + X(L) and AR0 = AR0 +
                                        ; 2 * NI
            ADDF   R0,R3,R5             ; R5 = R2 * COS + R1 * SIN
BLK2        STF    R5,*AR2++(IR0)       ; X(L) = R2 * COS + R1 * SIN,
                                        ; incr AR2 and ...
||          STF    R4,*+AR2             ; Y(L) = R1 * COS – R2 * SIN

            CMPI   R7,AR1
            BNEAF  INLOP                ; Loop back to the inner loop
            ADDI   AR5,AR4              ; IA = IA + IE; AR4 points to cosine
            ADDI R10,AR1,AR0            ; (X(I), Y(I)) pointer
            SUBI   1,R8,RC              ; RC should be 1 less than desired #

            LSH    1,R8                 ; Increment loop counter for nxt time
            BRD    LOOP                 ; Next FFT stage (delayed)
            LSH    1,AR5                ; IE = 2 * IE
            LDI    R7,IR0               ; N1 = N2
            SUBI3  1,R8,RC              ; RC shd be 1 less than desired #

*           STORE RESULT OUT USING BIT–REVERSED ADDRESSING
END:        LDI    @FFTSIZ,IR0          ; IR0 = SIZE OF FFT = N
            SUBI3  2,IR0,RC             ; RC = N – 2
            LDI    2,IR1
            RPTBD  BITRV
            LDI    @INPUT,AR0
            LDI    @OUTPUT,AR1
            LDF    *+AR0(1),R0

*           BIT REVERSE LOOP
            LDF    *AR0++(IR0)B,R1
||          STF    R0,*+AR1 (1)

BITRV       LDF    *+AR0 (1),R0
||          STF    R1,*AR1++(IR1)
```

```
            LDF     *AR0++(IR0)B,R1
||          STF     R0,*+AR1 (1)
            STF     R1,*AR1++(IR1)

SELF        BR      SELF                    ; Branch to itself at the end

.end
```

Figure 14. Photo of I_{DD} for FFT