TEXAS INSTRUMENTS

Calculation of TMS320LC54x Power Dissipation

Application Report

1997 Digital Signal Processing Solutions

Texas Instruments

Printed in U.S.A., June 1997

SPRA164

Calculation of TMS320LC54x Power Dissipation

Application Report

Clay Turner
Digital Signal Processing Solutions — Semiconductor Group

SPRA164
June 1997

PRINTED WITH SOY INK™

TEXAS INSTRUMENTS

Printed on Recycled Paper

IMPORTANT NOTICE

Texas Instruments (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 the specifications applicable at the time of sale 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.

Certain applications using semiconductor products may involve potential risks of death, personal injury, or severe property or environmental damage ("Critical Applications").

TI SEMICONDUCTOR PRODUCTS ARE NOT DESIGNED, INTENDED, AUTHORIZED, OR WARRANTED TO BE SUITABLE FOR USE IN LIFE-SUPPORT APPLICATIONS, DEVICES OR SYSTEMS OR OTHER CRITICAL APPLICATIONS.

Inclusion of TI products in such applications is understood to be fully at the risk of the customer. Use of TI products in such applications requires the written approval of an appropriate TI officer. Questions concerning potential risk applications should be directed to TI through a local SC sales office.

In order to minimize risks associated with the customer's applications, adequate design and operating safeguards should be provided by the customer to minimize inherent or procedural hazards.

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.

Copyright © 1997, Texas Instruments Incorporated

Contents

List of Figures

List of Tables

Calculation of TMS320LC54x Power Dissipation

ABSTRACT

Digital signal processor (DSP) applications continue to demand more from less: more features while consuming less power. The TMS320LC54x meets these challenges by combining high processing capability with the greatly reduced power consumption that battery-powered and portable applications demand. This application report describes the power-saving features of the TMS320LC54x and presents techniques for analyzing system and device conditions to determine operating current levels and power dissipation. From this information, informed decisions can be made regarding power supply requirements and thermal management considerations.

1 Introduction

The TMS320LC54x devices are 16-bit fixed-point processors with enhanced processing capabilities. Architecture, design, and process enhancements have produced a generation of DSPs that provide high performance while maintaining low power dissipation.

The TMS320LC54x DSPs are currently capable of processing speeds as high as 100 million instructions per second (MIPS) to handle a wide variety of high performance applications. In addition to its superior performance, the device exhibits very low power dissipation and offers flexible power management features which allow further reductions in power requirements.

The static CMOS technology used in fabrication of the TMS320LC54x family of devices combines high density with low power dissipation. Because CMOS devices ideally draw current only when switching, this technology offers the potential for fully static devices with standby modes, thereby exhibiting very low current drain. These characteristics make the TMS320LC54x devices uniquely well-suited to portable, power-sensitive, and battery-operated applications such as digital cellular telephones, laptop modems, voice mail pagers, etc.

Improved fabrication processes used in the manufacture of the TMS320LC54x yield typical active current requirements of 0.7 mA per MIPS for 3-V operation. These characteristics are further improved with the following power-management features:

- Flexible low-power modes (IDLE instructions) conserve power by halting sections of the device when their use is not required. Operation of the central processing unit (CPU), the on-chip peripherals, and the clock-generation circuitry can be halted independently.

- CLKOUT switching allows the external CLKOUT signal to be disabled, therefore, providing power savings when external clock synchronization is not necessary.
- Bus holder circuitry integrated into the device prevents floating buses, eliminating the need for external pullup resistors

This application report describes techniques for analyzing system and device conditions to determine operating current levels. From this analysis, power dissipation for the device can be determined. Knowledge of power dissipation can, in turn, be used to determine device thermal-management requirements.

2 CMOS Power Consumption

In CMOS logic, internal node voltages swing completely from one power supply rail to the other. The voltage change on a gate capacitance requires charge transfer, and therefore causes power consumption. Once the gate capacitance is charged, the gate can maintain a DC voltage level without any additional charge movement and does not consume current. For this reason, CMOS circuitry consumes power only when switching states.

The required charge to change voltage levels on the gate is described by equation (1).

$$Q_{gate} = C_{gate} \times V_{DD} \tag{1}$$

where:

Q_{gate} is the charge required to change states (coulombs)
V_{DD} is the power supply voltage (volts)
C_{gate} is the gate capacitance (farads)

Repeated switching generates a current proportional to the switching frequency. Since current is defined in terms of coulombs per second (amperes), the current can be calculated as shown in equation (2).

$$I = Q_{gate} \times \text{frequency} = (C_{gate} \times V_{DD}) \times \text{frequency} \tag{2}$$

where:

I is the current in amperes (coulombs per second)

For example, the current consumed by an 80-pF capacitor being driven by a 10-MHz CMOS level square wave with 3-V V_{DD} is calculated as shown in equation (3).

$$I = (80 \times 10^{-12}\ \text{F}) \times (3\ \text{V}) \times \left(10 \times 10^{6}\ \text{Hz}\right) = 2.4\ \text{mA} \tag{3}$$

This approach can be generalized to include all of the internal node capacitances in a device to determine the total device current. From equation (3), the current is clearly proportional to the supply-voltage level, the total number of charging nodes and their capacitance, and the switching frequencies of those nodes.

$$I_{device} = C_{total} \times V_{DD} \times f_{clock} \quad (4)$$

where

I_{device} is the total device current

C_{total} is the total node capacitance of all internal switching nodes

f_{clock} is the switching frequency of the device clock

Since knowing the number and capacitance of all of the internal switching nodes is an unmanageable task, the current under different conditions can be determined empirically by measuring the current level for a particular algorithm under known frequency and supply voltage conditions, and then scaling the current value to determine the behavior under different conditions. This is the method utilized in this report. Current was measured under known conditions and scaled to represent a frequency-dependent current factor usually expressed in milliamperes (mA) per megahertz (MHz) or mA per MIPS depending on the situation. Then the current factor can be multiplied by the operating frequency to determine the total current.

The total power dissipation is dependent on internal operation as well as external bus cycles and the loads associated with the external buses. Bipolar devices (transistor-to-transistor logic [TTL] loads), pullup resistors, and other devices consume dc power that adds a constant offset to the total current. The effects of these additional components are examined in other sections of this report.

Determination of total power dissipation and calculation of total supply current are covered extensively in section 6, while system design considerations for minimizing power dissipation are covered extensively in section 7.

3 General Device Current Characteristics

In general, device current requirements vary according to several system- and device-related considerations. Among these considerations are supply voltage, temperature, and device program activity.

The TMS320LC54x devices exhibit power supply requirements consistent with CMOS device characteristics. Aspects of these characteristics necessary for analysis of device power supply current requirements are discussed in detail in this application report. A thorough working knowledge of device architecture and operation is critical to understand the power analysis described in this document. Detailed information regarding TMS320LC54x device architecture and operation is found in the *TMS320LC54x Reference Set Volume 1* (literature number SPRU131).

3.1 Current Components

The V_{DD} supply to the TMS320LC54x devices is internally separated into four sections, each of which supplies a different set of internal circuitry. The four groups are:

- Address
- Data
- Control
- Internal

The address and data groups supply the output drivers on the address and data buses respectively. The control group supplies all other outputs and the internal group supplies all of the internal device logic. This internal power structure isolates noise generated in the high current output drivers for the address and data buses, preventing the noise from propagating through the internal logic.

In most applications, all of the V_{DD} connections are connected together externally and can be considered a single supply. Most device activity involves the internal portion of this supply, but the use of the external buses causes contribution from address, data, and control lines.

3.2 Current Dependencies

The actual power supply current depends on many factors. These factors can be considered as two groups: system-related, and device- or algorithm-related factors. Some system-related factors are as follows:

- Operating frequency
- V_{DD} supply-voltage levels
- Direct current (dc) loads
- Operating temperature

Of the system-related factors, the most significant is operating frequency. As previously shown, CMOS current is directly proportional to the switching rate. If the switching rate doubles, the frequency-dependent component of the device current also doubles. V_{DD} supply-voltage levels also strongly affect the device current, but not as significantly as operating speed. The dc loads (such as TTL loads) introduce a constant offset in the device current and are not frequency dependent. Operating temperature affects the total device current less significantly than any of the other factors.

Total device current is also affected by device- and algorithm-related factors such as:

- Program activity
- Internal and external bus-data switching patterns
- Utilization of external buses

Since the total number of switching nodes affects the current, it is clear that the nature of the program activity strongly affects the current. Instructions that perform several parallel actions consume more current than simpler instructions. Instructions that involve moving data between the CPU and on-chip memory use internal address and data buses, and therefore, use more current than instructions that involve only the CPU.

The nature of the data being moved by internal buses also affects the current. Data buses are generally fabricated as a set of connections routed together. Each of these lines has a characteristic capacitance with respect to the silicon substrate on which they are fabricated. Since these bus lines frequently are routed adjacent to each other, they also possess an intersignal capacitance, or a capacitance between the adjacent bus lines. When the voltage on a bus line changes, these capacitances also become charged and discharged as described previously. Since some of these capacitances exist between signal lines, the necessity to charge or discharge depends on the voltage levels on both lines. Therefore, the data patterns on the bus affect how many of these characteristic capacitances must charge, and consequently, affect the total device current. For this reason, driving a bus with an alternating pattern of AAAAh/5555h consumes more current than driving 0000h/FFFFh. In both cases, all 16 lines are switching so the current contribution due to each line's capacitance with respect to the silicon substrate is the same. However, in the former case, the capacitance between the bus lines must charge and discharge, which consumes current. In the latter case, this does not occur since all of the bus lines follow the same voltage.

External buses have similar effects on the internal buses mentioned previously with some additional considerations. External buses have greater intrinsic capacitance than internal buses because of the nature of the packaging and the presence of high-output current drivers for the pins. External buses also experience the load of the other external devices to which they are connected. Greater capacitance results in greater current. External buses also have data-dependent current levels and use of the external buses instead of internal buses significantly adds to the total device current.

Considering the contribution of the factors mentioned earlier, the total I_{DD} power supply current can be described as shown in equation (5).

$$I_{total} = (I_{int} + I_{addr} + I_{data} + I_{cntl}) \times V_S \times T \quad (5)$$

where

I_{total}	is the total I_{DD} supply current
I_{int}	is the current component due to all internal circuitry
I_{addr}	is the current component due to external address bus activity
I_{data}	is the current component due to external data bus activity
I_{cntl}	is the current component due to external control line activity
V_S	is a scale factor due to supply voltage
T	is a scale factor due to operating temperature

3.3 Algorithm Partitioning

The total current consumed by the device varies, based on program activity. Some sections use more current due to external bus usage or data patterns. Other sections may use less because the activity is completely confined to the CPU, or there is no activity such as in a power-down or idle state. Analysis of the device power consumption is greatly simplified by considering each section, or partition, of an algorithm that exhibits distinct concentrations of activity separately. These partitions of significant device activity can be considered the major contributors to the overall device current and shorter periods of less significant activity can be ignored. Each partition can be analyzed to determine its current use and these contributions can then be time-averaged to determine the total device current. This approach generally simplifies the current analysis compared to a more detailed analysis that may not yield significantly increased accuracy for the increased analysis complexity.

3.4 Test Setup Description

All of the TMS320LC54x power measurements were performed on the test setup shown in Figure 1. Both LC545 and LC548 devices were tested. Pullup resistors were added to inputs as necessary. The power supply for the pullup resistors was not connected through the current meter to avoid inclusion of this current in the measurements because, in a system, the power to drive the TMS320LC54x inputs high is supplied by an external device. Capacitive loads were included on outputs when relevant to the measurement and were varied to determine the power supply dependency on this parameter. The maximum rated output load capacitance for the device is 80 pF.

The clock to the device was provided by a Sony/Tektronix AFG2020 function generator. Unless otherwise indicated, all measurements were made with the device in PLL multiply-by-one clock mode. Various frequencies were used as indicated.

Measurements were made using a Tektronix DM 501 digital multimeter and, unless otherwise specified, were made at a supply voltage of 3.0 V and at an ambient temperature of 25°C. The I_{DD} that supplies the internal logic was measured separately from the I_{DD} that supplies the address, data, and control output buffers. Current traces for the example code were made using a Tektronics AM 503 current probe.

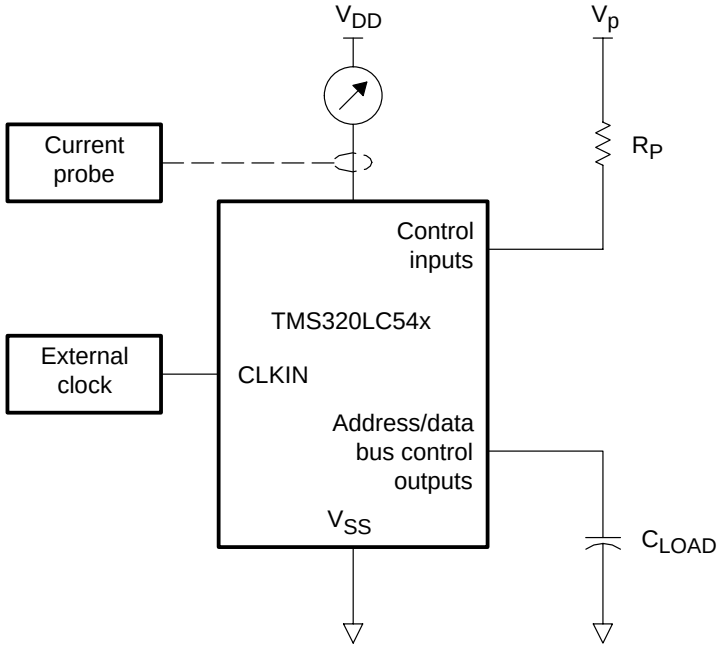

Figure 1. Test Setup

4 Current Due to Internal Components

This section goes into detail explaining the current (mA) use of 320LC54x DSP internal components. The following components are covered:

- Clock-generation circuitry
- Power-down modes
- CPU
- Memory usage
- On-chip peripherals

4.1 Clock Generation Circuitry

The 320LC54x digital signal processors are fabricated in a fully static CMOS technology. The current used by the device is entirely due to switching currents and virtually no current is consumed when the device is not switching. This provides the ability to conserve power by operating the device at very low clock rates, or by stopping the clock, without corruption of data (assuming all timing requirements have been met as outlined in the data sheet). References to “processor clock” or “device clock” in this document refer to CLKOUT.

4.2 Clock Modes and Their Effect on Power Consumption

A portion of the total device power is consumed by the circuitry that generates and distributes the internal processor clock. The TMS320LC54x family provides several methods for generating the processor clock:

- **Internally generated clock using an external crystal**

  This option allows the user to construct a simple crystal oscillator using as little as three external components (a crystal and two capacitors) in conjunction with an internal inverting amplifier on the TMS320LC54x. The oscillator, powered by the DSP power supply, generates an input frequency at X2/CLKIN. This frequency is then multiplied by a scaling factor to generate CLKOUT.

- **Externally generated clock**

  Instead of using a crystal, the user may also inject an external clock from an integrated circuit oscillator or other source into X2/CLKIN with X1 left unconnected. In this case, less power is consumed by the DSP since it is not driving a crystal, but power is consumed somewhere else in the system to generate the input clock signal.

- **Phase-locked loop (PLL) clock multiplication**

  The TMS320LC54x has an internal phase-locked loop (PLL) which can lock on to an input clock signal and generate CLKOUT as a multiple of the frequency of the input signal. Multiplication factors range from 0.25 to 15 depending on which TMS320LC54x device is being used. The user should See the *TMS320LC54x Reference Set, Volume 1*, or the data sheet for detailed information on available clock modes on each device.

- **Stop mode**

  The TMS320LC54x includes an additional clock mode called stop mode which disables the PLL and the internal clock signals to the CPU and peripherals. Stop mode performs a similar function to the IDLE3 mode mentioned in the next section but can be initiated through hardware control.

The clock generation circuitry basically uses a constant amount of current at all times, unless the input clock speed is changed or stopped. This current forms background current which does not change due to CPU activity.

4.2.1 Power-Down Modes

Using software and hardware control features, the TMS320LC54x devices provide several power-down modes to aid in conservation of power during periods of reduced device activity. The IDLE instruction provides a means to cause the device to enter a power-down state through software control. The three versions of the IDLE instruction operate as follows: IDLE1 disables the processor clock to the CPU, but allows all on-chip peripherals to remain active, including the timer, standard serial ports, and TDM serial ports. IDLE2 disables the processor clock to the CPU, the timer, the standard serial ports, and the TDM serial ports, but the buffered serial port (BSP) and host-port interface (HPI) remain active. IDLE3 stops the clock to the CPU and all on-chip peripherals, and disables the PLL, resulting in the lowest power state of the three IDLE modes. In IDLE3 mode, the BSP and HPI can continue to operate under special conditions (these conditions are covered in more detail in later sections describing the BSP and the HPI). The IDLE states can be released through hardware control using interrupts.

There are two power-down modes that are initiated under hardware control. Stop mode causes the processor to enter a state similar to IDLE3 mentioned above. It is initiated by changing the clock-mode pins appropriately. Hold mode is similar to IDLE1 mode mentioned above and is initiated by pulling the $\overline{\text{HOLD}}$ input low. During this state, the external buses are in high impedance state and the device enters IDLE1 mode.

CLKOUT remains active (if it is selected) during IDLE1 and HOLD mode. During IDLE2 mode, IDLE3 mode and Stop mode, CLKOUT is always disabled. Table 1 summarizes the effects of each of the power-down modes on the activity of CPU, the peripherals, and the CLKOUT pin.

Table 1. TMS320LC54x Activity During Low-Power Modes

MODE	CPU	PLL	TIMER	STANDARD SERIAL PORTS	BUFFERED SERIAL PORT	HOST PORT INTERFACE	CLKOUT
Normal	•	•	•	•	•	•	•
IDLE1		•	•	•	•	•	•
Hold		•	•	•	•	•	•
IDLE2		•			•	•	
IDLE3					•†	•†	
Stop					•†	•†	

† Under special conditions

The current required to operate the PLL can be observed directly in IDLE2 mode. In this state, when the BSP and the HPI are disabled, the PLL is the only active section of the TMS320LC54x. The current required to operate the PLL contains both a frequency-independent component (due to biasing requirements of the PLL) and a frequency-dependent component (due to node switching in all of the other circuitry on the device). The clock generation circuitry is the only circuitry on the processor that contributes a frequency-independent current component. The frequency-dependent component is expressed in mA per MHz of CLKOUT. The measurements given here do not include the current required to drive the CLKOUT pin. Inclusion of the current due to CLKOUT is discussed later in the report.

Table 2 shows the power consumption of the device in each of the low-power modes under different clock-generation conditions. These measurements were obtained by measuring the device current in the specified IDLE mode with all peripherals disabled.

Table 2. TMS320LC54x Power Consumption During Low-Power Modes

Divide-by-two external clock mode: (PLL disabled)	IDLE1	0.005 mA + 0.12 mA per MHz of CLKOUT
	IDLE2	0.005 mA + 0.03 mA per MHz of CLKOUT
	IDLE3	<1 μA
PLL external clock mode:	IDLE1	0.80 mA + 0.12 mA per MHz of CLKOUT
	IDLE2	0.80 mA + 0.03 mA per MHz of CLKOUT
	IDLE3	<1 μA

4.3 Internal CPU Activity

The power use of the TMS320LC54x devices is directly related to the level of CPU activity. Many factors affect the CPU current use including the instruction complexity (the number of parallel operations being performed by the instruction), the utilization of the internal buses (including the data patterns on the buses), and the effects of using the repeat option on instructions. These effects are examined in the sections that follow.

4.3.1 Internal CPU Functional Blocks

The TMS320LC54x CPU is composed of several hardware blocks that perform specific functions. The function of these blocks is described in detail in the *TMS320LC54x Reference Set, Volume 1*. These blocks also have specific sets of instructions associated with their functions. These CPU functions have been placed in the following groups for consideration of the power use:

- Accumulator and arithmetic/logic unit functions include shift operations, Boolean operations, non-multiplication arithmetic operations, and accumulator load/store/modify operations.
- Auxiliary register and auxiliary register arithmetic unit (ARAU) functions include auxiliary register load/store/modify operations, that do not occur as a parallel operation in other CPU instructions.
- Multiplier functions include T-register operations and multiply/ accumulate operations.
- Branch and control functions include branches, calls, and returns.

4.3.2 Instruction Complexity

The current used during specific CPU instructions is directly related to their complexity. Instructions that perform more parallel operations consume more current. Instructions that load or shift internal registers generally require the least current. Instructions that minimize use of the internal and external data buses also use less current. A more detailed examination of the effects of memory usage is covered later.

Current use increases with utilization of arithmetic functions such as addition, subtraction, and Boolean functions. Current use further increases when the hardware multiplier is used. The TMS320LC54x processors provide several multiplication instructions which perform different levels of parallel operations. For example, these instructions can automatically load two data operands from memory, multiply the operands, accumulate the product, and automatically control pointers to the operands for the next execution. The power use of the instructions increases as the number of parallel operations increases. The most power intensive instructions on the TMS320LC54x devices are various forms of multiplications, which multiply two 16-bit values, accumulate (add) the product to the accumulator, and manage data in two input memory arrays all in one clock cycle.

4.3.3 Power Effects of Repeated Instructions

The TMS320LC54x family of processors provide an automatic means for repeating, or “looping”, an instruction without the pipeline overhead associated with implementing a loop through the use of a branch instruction. The branch instruction disrupts the pipeline flow by causing a discontinuity in the program address sequence. The discontinuity causes the pipeline to flush, and consequently causes the application to incur extra clock cycles for each pass of the loop. Since many computations and logical operations may require repetition of the same instruction, a branched loop becomes an inefficient method for repeating a single instruction in terms of pipeline overhead.

The TMS320LC54x processor solves this problem by offering a “repeat” instruction (RPT) that repeats the single instruction, following it up to 256 times with zero overhead. Assuming the pipeline is full when the RPT instruction is encountered, the instruction that follows it is repeated the specified number of times with zero pipeline overhead. In other words, the repeated instruction behaves much as it would if it were explicitly listed many times. But instead of consuming up to 256 words of memory (for straight-line code), the loop can be implemented in only two instructions.

Although some instructions are not useful to repeat, such as instructions that load or store values in the accumulator or auxiliary registers, many other instructions may be repeated including arithmetic and logic instructions, multiply/accumulate operations, and “no operation” (NOP) instructions repeated for the purpose of implementing delays.

Repeated instructions deserve consideration in terms of power because they may consume significant lengths of time; therefore, the action being repeated changes the power consumption required to execute the instruction. Most instructions require less current to execute when they are repeated because, although the same action is performed repetitively, the instruction is fetched only once. Consequently, the power required to re-fetch the instruction is saved. Some multiply/accumulate instructions require more current to execute when repeated because the CPU automatically increments an address pointer to a table of operands. The auto-increment does not occur if the instruction is repeated as straight-line code. An example of code illustrating these two cases is shown in Figure 2.

```
RPT      #99                    MACD     *AR3–, coeff
MACD     *AR3–, coeff           MACD     *AR3–, coeff+1
                                MACD     *AR3–, coeff+2
                                         . . .
                                MACD     *AR3–, coeff+98
                                MACD     *AR3–, coeff+99

words:   3                      words:   200
cycles:  101†                   cycles:  300†
power:   1.0 mA per MIPS @      power:   0.90 mA per MIPS @
         3 V†                            3 V†
             (a)                             (b)
```

† Cycle counts and power use vary due to locations of operands in SARAM, DARAM, or ROM

Figure 2. Example of Repeated (a) vs. Straight-Line (b) Code Use of MACD

4.3.4 Current Use of the TMS320LC54x CPU

Measurements performed on the power characteristics of the TMS320LC54x instruction set are included In Appendix B. The measurements were performed by configuring the TMS320LC54x device in a known state, repeating instructions and measuring the current use during the repetition of the instruction. All measurements were made in PLL multiply-by-one clock mode unless otherwise specified and the current readings include the current used by the clock generation circuitry. All instruction set current measurements were made with the peripherals (timer, serial ports) inactive and with address visibility disabled. The effects of address visibility are discussed in more detail later in this report. The measurements were performed with the test code executing from on-chip dual-access RAM (DARAM).

Instructions were repeated by two methods. In the “straight-line” case, the instruction under test was explicitly repeated many times followed by a branch instruction back to the top of the block of the instructions. In the “RPT” case, the RPT instruction was used to generate many repetitions of the instruction under test followed by a branch instruction to return control back to the RPT instruction. The two test cases are illustrated in Figure 3.

```
testloop                    testloop
       I1                           RPT    #255
       I1                           I1
              (256 times)           B      testloop
       I1
       I1
       B      testloop
(a) straight-line method           (b) RPT method
```

Figure 3. Loop Implementation for Instruction Current Measurements

Table 3 includes a general comparison of the power dissipation performance of the TMS320LC54x CPU for various activities. The measurements in Table 3 are intended to give a quantitative point of reference for the power characteristics of the TMS320LC54x instruction set. Actual observed current may vary somewhat due to the pattern of data being manipulated, code source memory type, and internal versus external bus usage. The effects on instruction current use due to memory-related factors are considered in more detail in the next section. All of the measurements are given in units of mA per MHz of CLKOUT. All measurements were performed using an addressed operand (either defined data memory address or indirect addressing) unless otherwise specified. A more detailed table of TMS320LC54x CPU power dissipation by instructions is included in Appendix B.

Table 3. TMS320LC54x CPU Power Dissipation Characteristics

ACTIVITY	CURRENT (mA per MIPS)	CURRENT AT 50 MIPS (mA)
IDLE3	0	0
IDLE2	0.03	1.5
IDLE1	0.12	6
Repeat NOPs	0.3	15
Inline NOPs	0.4	20
Block data transfer in on-chip DARAM using RPT	0.8	40
Repeat MAC with changing data (dual-operand addressing)	1.0	50
Inline MAC with changing data (dual-operand addressing)	1.2	60
Repeat MACD with changing data (single-operand addressing)	0.8	40
Inline MACD with changing data (single-operand addressing)	1.0	50
Repeated double-precision arithmetic instructions with changing data	0.9	45
Inline double-precision arithmetic instructions with changing data	1.1	55
Repeat FIRS with changing data	1.2	60
Inline FIRS with changing data	0.9	45
FIR filter	0.9	45
Full-rate GSM vocoder	1.03	51.5
Complex 256-point FFT	1.07	53.5

4.4 Effects of Memory Usage on Power Consumption

Although device power use for a given algorithm is based primarily on CPU power, the memory environment in which the algorithm is running also plays a role in the total power consumption. An understanding of the effects of memory and bus configuration on power consumption is valuable in optimizing system and software design for peak performance. In this section, memory types, use of on-chip vs. off-chip memory, utilization of internal buses, address visibility, and power characteristics of each of the memory types is examined in detail.

4.4.1 Memory Types

The TMS320LC54x processor family provides several on-chip memory options which can be utilized in conjunction with external memory to lower system cost and reduce system power use. The on-chip memory types available include single-access RAM (SARAM), dual-access RAM (DARAM), and ROM. General information regarding the TMS320LC54x memory structure is described here. For detailed information on memory maps and memory configuration, See the *TMS320LC54x Reference Set, Volume 1* (literature number SPRU131).

The TMS320LC54x SARAM is static RAM, which, under normal conditions, allows a single memory read or write access in one clock cycle. However, with an algorithm configured appropriately, dual access (two memory operations in one cycle) is possible. The location in the memory map and the size of this memory block are dependent on the processor being used. However, in each case, the total SARAM block is composed of 2k and 1k sub-blocks. For example, an 8k SARAM block is actually composed of four 2k sub-blocks. If the SARAM block is an odd multiple of 1k in size, it is composed of all 2k sub-blocks plus one 1k sub-block (that is, 7k = 3 x 2k + 1k). The SARAM is single-access if both accesses are targeted at memory locations within the same 2k (or 1k) sub-block because they cannot be performed in the same clock cycle. If the two accesses are located in different sub-blocks, they can be performed in the same clock cycle and the memory essentially becomes dual-access. Knowledge of this fact can be used in the design of the memory configuration to take advantage of this conditional dual-access capability and increase CPU throughput by eliminating memory-based pipeline delays. As CPU throughput increases, current also increases so this behavior should be considered when estimating power consumption of an algorithm.

The DARAM is always dual-access, regardless of the addresses of the memory operations.

The on-chip ROM (read-only memory) is memory that can be used as program memory or to store coefficient tables that are never altered. As with external ROM, the memory is non-volatile meaning it retains the data it contains even if device power is removed. This memory block is single-access only. The ROM is mask-programmed at the factory at the time of fabrication and cannot be reprogrammed or erased. For information on how to submit code for on-chip ROM production, see the *TMS320LC54x Reference Set, Volume 1*, or contact Texas Instruments.

Of these memory types, ROM is used primarily for storage of program code and constants. SARAM and DARAM can be configured as either program or data memory.

Internally, data is transported between the memory and the CPU via a series of four separate internal buses. The program bus carries address and data for program code fetches and immediate operands. The C-bus and D-bus are used to read addressed operands during instruction execution. The E-bus is used to write data resulting from instruction execution. Use of these buses is dependent on the operation being performed. Different instructions use different combinations of one or more of these buses. As would be expected, instructions that use more internal buses simultaneously require more current. See the *TMS320LC54x Reference Set, Volume 1*, for more detailed information on how instruction operations utilize the internal bus structure.

4.4.2 Memory Architecture and the External Memory Interface

The TMS320LC54x device is based on a modified Harvard architecture in which program and data occupy completely separate memory spaces. The device also utilizes a third memory space called I/O space to provide flexibility for memory-mapped access to input/output devices such as A/D converters, D/A converters, codecs, and other devices. In the memory map for each of these spaces, some sections of memory are mapped as internal and some are mapped as external. When the CPU activity is limited to internal memory accesses, the external memory interface (composed of the external address, data, and memory control signals) is inactive. When an external memory access occurs, the external interface activates automatically. This feature conserves power since the output drivers for the external memory interface are not operated when it is not necessary. As a result of this feature, less power is required to operate from internal memory than from external memory. The control of the external interface is automatic and does not require any special programming.

4.4.3 Address Visibility Feature

The address visibility feature allows the internal program address to appear at the external address pins even during internal memory accesses. This is accomplished by setting a control bit (the AVIS bit) and allows the internal program address to be traced and interrupt vectors to be decoded in conjunction with $\overline{\text{IACK}}$ (interrupt acknowledge) when the interrupt vectors reside in on-chip memory. This visibility can be a valuable debugging tool. However, this operation of the external address bus consumes power which is not necessary for the execution of a program residing in internal memory. When the debugging is complete, the address visibility mode should be disabled to conserve power.

4.4.4 Power Use Comparison of On-Chip Memory Types

The different types of on-chip memory also exhibit different power use characteristics for the same functions. Several common functions including block transfers from program to data space, code execution from program space, and data read/writes to internal RAM were evaluated.

- **Memory access**

  Of the three internal memory types available, SARAM consumes the most current, followed by DARAM, and then ROM. Memory accesses to DARAM require approximately 4% less current than identical accesses to SARAM. Memory accesses to ROM require approximately 10% less current than identical accesses to SARAM.

- **Code execution**

  Code execution from ROM also requires approximately 10% less CPU current than is required to execute the same code from SARAM.

4.4.5 Effects of Bus Data Patterns

As previously described, data buses have a characteristic capacitance associated with each line as well as intersignal capacitances between the lines. These capacitances cause the current required for a given operation to vary depending on the data patterns occurring on the bus. Measurements of this phenomenon were made by performing repetitive data transfers while varying the data pattern to observe the current differences.

The results, shown in Figure 4, provide a method to scale current use of a particular instruction based on the “data complexity” or the amount of changing data on the data bus. The scale factor is normalized to the worst-case data complexity or the data pattern AAAAh – 5555h. For the TMS320LC54x, the bus data pattern 0000h – FFFFh requires approximately 95% of the worst-case bus current use. No change in bus data requires approximately 70% of the worst-case current use.

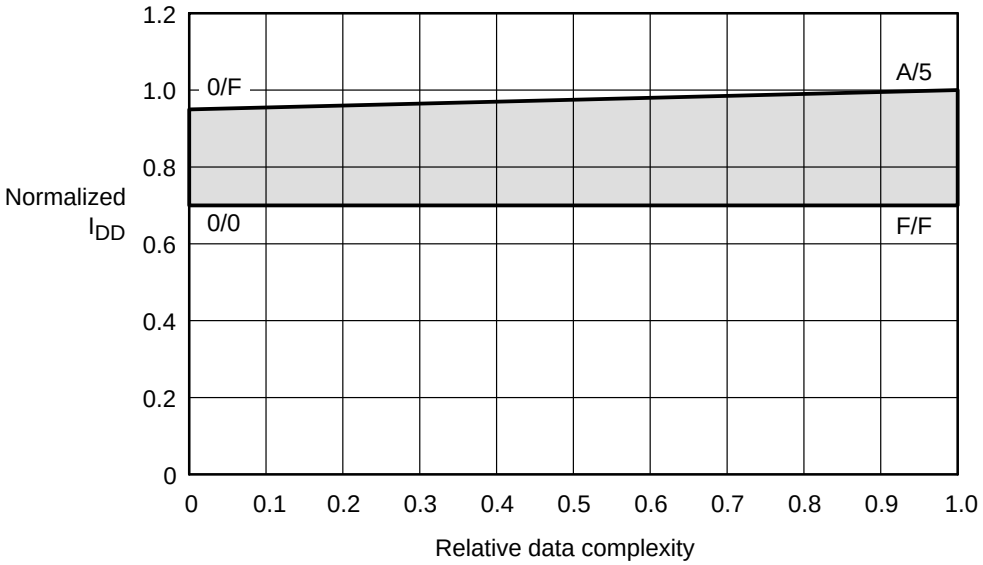

Figure 4. Current Scale Factor for Bus Switching

4.5 Current Due to Peripherals

Although the CPU contributes the majority of the current involved in executing application code, the on-chip peripherals also contribute to the current use of the device. The TMS320LC54x family provides several on-chip peripheral devices for use with the CPU: a timer, a synchronous serial port, a time-division-multiplexed (TDM) serial port, a BSP, and a HPI. Usually the current contribution of the on-chip peripherals is small compared to the current used by the CPU and may be considered negligible. However, in some cases, such as when the CPU is in IDLE mode, the current due to the peripherals may take on greater significance. The current contribution of each of the on-chip peripherals is examined in detail in this section.

4.5.1 Timer

Although the timer runs independently of the CPU, the speed of the timer is still based on CLKOUT. Consequently, the current requirement of the timer changes with processor clock speed just as other functions do. The current use of the timer also depends on its activity. The more frequently the timer reloads, the higher the operating current is. However, in most cases, the timer counts many cycles between reloads so the effect of the reload on the total current required is negligible.

The timer requires approximately:

0.007 mA per MHz of CLKOUT at 3 V

Since this value is relatively low, it may be desirable to neglect the timer in calculation of the current unless increased accuracy is desired or current use in the IDLE mode is being minimized.

4.5.2 Standard Serial Port

The TMS320LC54x standard serial port provides direct communication with serial devices such as codecs and A/D converters. The serial ports may also be used for communication between processors in multiprocessing applications. As a peripheral device to the CPU, the serial ports also use power based on their activity and can be disabled to conserve power when not needed.

The standard serial port can be operated while the CPU is in IDLE1 power-down mode. In this case, the serial port consumes additional current to the IDLE1 current stated previously. The standard serial port does not operate when the CPU is in IDLE2 or IDLE3 power-down mode.

4.5.2.1 External Serial Port Interface

Both the serial port receiver and transmitter are based on three signals: the serial-port transfer clock (CLKR/CLKX), the frame sync (FSR/FSX), and the data line (DR/DX). For detailed information on the operation of the serial port, see the *TMS320LC54x Reference Set, Volume 1*. Since the serial-port clock is always running, it consumes the most current. The serial-port clocks can be generated internally to the DSP or externally. If the serial-port clock is generated internally, more current is required than if it were generated externally since the device consumes additional current due to driving the external CLKX pin. Since the operation of the serial port is synchronized to the serial-port clock only, it is not dependent on the CPU clock unless the serial-port clock is generated internally.

The power used by the port to transfer data on DX and DR is affected by both the serial port operating mode (burst/continuous) and the data pattern being transferred, since both affect the rate of activity on the data signal. The worst case power use of the serial port occurs when transferring alternating bits (i.e., AAAAh) in continuous mode, causing the data signal to toggle at one-half the rate of the serial-port clock.

4.5.2.2 Serial Port Measurements

The serial port receiver and transmitter were evaluated separately (since they can be operated separately). The measurements were made in continuous mode with continuous data transfer occurring. This represents the highest current used by the serial port. The user may scale these values to determine the actual current use based on the frequency of data transfer occurring in an application. The measurements that follow were made with the serial port interrupts masked to prevent contribution to the current from the CPU response to the interrupts. For the receiver measurements, CLKR was generated externally. For the transmitter measurements, CLKX was generated internally. The low value of the measured current represents the data word FFFFh being transferred (data signal not toggling). The highest current value represents the data word AAAAh being transferred (data signal toggling). The current measurements are represented in terms of mA per MHz of the serial port clock (not CLKOUT). For the transmitter, the current required to operate only the serial-port clock is also included. This value represents the current used by the serial port if no data transfer is occurring and only the serial port clock is running. This measurement represents the current used to drive an unloaded CLKX signal.

serial port receiver:

0.020–0.035	mA per MHz of serial-port clock (CLKR) at 3 V

serial port transmitter:

0.012–0.070	mA per MHz of serial-port clock (CLKX) at 3 V
0.012	mA per MHz is required to operate CLKX only at 3 V

4.5.3 Buffered Serial Port

The TMS320LC54x buffered serial port (BSP) is a standard serial port interface with autobuffering capability that can receive or transmit blocks of data to or from internal DARAM without intervention from the CPU. The buffered serial port uses more current than the standard serial port to receive or transmit data because of the additional internal memory reading/writing and address generation involved. However, since the BSP does not require interrupting the CPU and requesting service for each word transmitted/received, the CPU can remain in an IDLE mode (or perform some other function) until an entire block of data has been transmitted. Consequently, all of the CPU activity (and power consumption) related to servicing serial port interrupts (as on the standard serial port) has been saved.

The buffered serial port can be operated while the CPU is in each of the IDLE modes. In these cases, the BSP consumes additional current besides the Idle-mode currents stated previously. The BSP is fully functional in IDLE1 and IDLE2 modes. The BSP can be operated while the CPU is in IDLE3 mode, but only if external BSP clocks and frame sync signals are provided. Since the internal device clock is stopped in IDLE3 mode, it is not available to generate internal clocks or frame sync signals for the BSP. For more information on the BSP and its operation during low-power modes, consult the *TMS320LC54x Reference Set, Volume 1*.

4.5.3.1 Buffered Serial Port Measurements

The external interface of the BSP and the power issues related to it are identical to the standard serial port described above. The BSP receiver and transmitter were also evaluated separately. The measurements were made under the same conditions as described for the standard serial port and with autobuffering enabled. This represents the highest current used by the BSP.

Buffered serial port receiver:

0.100–0.120 mA per MHz of serial port clock (CLKR) at 3 V

Buffered serial port transmitter:

0.300–0.350 mA per MHz of serial port clock (CLKX) at 3 V

0.300 mA per MHz in required to operate CLKX only at 3 V

4.5.4 Host-Port Interface

This capability allows a host device to read/write to TMS320LC54x internal memory without interruption of the CPU. The HPI can access a single memory location or an entire memory block before interrupting the CPU. This capability allows the CPU to perform other functions or stay in a low power mode while data is transferred. Since the HPI is an 8-bit interface, there are two accesses involving the external HPI interface for each 16-bit access to the internal memory.

When the HPI is not in use, the input pins on the external HPI interface do not need to be pulled high. The device has internal pullup and bus keeper circuitry that maintains all of the inputs at proper logic states, and consequently, there is no current leakage due to floating inputs. The internal pullups for the input pins are enabled and the bus keepers are set when the HPI is disabled by leaving the HPIENA pin open or connecting it to ground.

The HPI can also operate while the CPU is in any of the IDLE modes. The HPI is fully functional when the CPU is in IDLE1 or IDLE2 modes. The HPI can be operated while the CPU is in IDLE3 mode only when it is configured in Host-only Mode (HOM). For more detailed information on the HPI and its operation during power-down modes, consult the *TMS320LC54x Reference Set, Volume 1*.

4.5.4.1 HPI Measurements

The HPI was tested by using another DSP to act as a host to read/write data to the interface. Data transfer to the TMS320LC54x internal memory was performed with the CPU in IDLE1 mode or running NOPs. The HPI was configured in shared-access mode (SAM) and the load on the output pins connected to the host DSP was approximately 15 pF per pin. The HPI was configured to auto-increment the memory address and the data pattern was AAAAh/5555h.

The current required by the HPI is independent of the speed of CLKOUT, but does depend on the word-transfer rate across the interface. The current measurements below are given as a function of word-transfer rate (in millions of 8-bit word transfers per second). These values should be scaled, based on the actual transfer rate.

Host writes (to 'LC54x memory):

0.192 mA per million 8-bit transfers per second

Host reads (from 'LC54x memory):

1.325 mA per million 8-bit transfers per second

5 Current Due to Outputs

Now that the contribution of the internal components is known, current due to outputs can be considered. Outputs are any of the external signals that are driven by the processor.

5.1 Categories of Outputs

Device outputs are categorized into three groups according to their relative current use. These categories are the data bus, the address bus, and the control outputs.

5.1.1 Data Bus

The address and data bus require an amount of current that is proportional to the overall parallel-bus-switching rate or memory cycle time. As discussed previously with respect to the internal buses, the worst case switching current for the bus occurs when all of the lines are switching and each line is always in the opposite logic state to its neighbors (i.e., AAAAh, 5555h, AAAAh...). The current required to toggle all lines where adjacent lines are in the same logic state is approximately 95% of the worst case value. The currents given below represent the worst case current required to drive the unloaded external data bus.

The unloaded external data bus requires approximately:
0.09 mA per MHz of data bus switching frequency per bit at 3 V

So the worst case switching current for the entire external data bus is:
1.44 mA per MHz of data bus switching frequency at 3 V

In addition to the current shown above, the current due to external capacitive load on the data bus can be calculated as:

$$I_{load} = C_{load} \times V_{DD} \times F \quad (6)$$

where

I_{load} is the additional current required per line due to the capacitive load

C_{load} is the external load capacitance per line

V_{DD} is the supply voltage

F is the switching frequency of the line

The maximum specified external capacitive load per pin for the TMS320LC54x family is 80 pF.

5.1.2 Address Bus

The address bus uses a similar amount of current as the data bus if all of the lines are switching; however, the more common operation is when the address lines are incrementing. Each of the bus lines has a characteristic capacitance. For simplicity, it is assumed that these are all equal and, therefore, the total current required to drive the address bus is as follows:

$$I = CVF_0 + CVF_1 + ,\ldots, + CVF_{15} = CV(F_0 + F_1 + F_2 + ,\ldots, = F_{15}) \quad (7)$$

where

F_0 is the switching frequency of A_0

F_1 is the switching frequency of A_1, and so forth

C is the load capacitance, and

V is the voltage of V_{DD}.

If the address bus is incrementing, each line is toggling at a rate which is one half of the next least significant line as:

$$\text{Since } F_{15} = 1/2\ F_{14}\ , F_{14} = 1/2\ F_{13}, \ldots, F_2 = 1/2\ F_1,\ F_1 = 1/2\ F_0 \quad (8)$$

then,

$$I = CV\left(F_0 + \frac{F_0}{2} + \frac{F_0}{4} + \frac{F_0}{8} + \frac{F_0}{16} + \frac{F_0}{32} + \frac{F_0}{64} + \frac{F_0}{35536}\right) \approx 2CVF_0 \quad (9)$$

Since each address bit is switching at a rate one half of the next least significant bit below it, the entire 16-bit bus requires an amount of current approximately equal to twice the current required to toggle one line only. This estimate assumes that the capacitive load on each of the address lines is the same. If branches occur where many address lines change, the instantaneous current increases accordingly.

The unloaded external address bus requires approximately:
0.09 mA per MHz of switching frequency per bit at 3 V

So the worst-case switching current for the entire unloaded external address bus is:
1.44 mA per MHz of switching frequency at 3 V

The current required for incrementing the address bus is approximately:
0.18 mA per MHz of switching frequency at 3 V

The additional current due to external capacitive load on the address bus can also be calculated as shown above for the data bus.

The measurements given above were taken under worst case data switching conditions, namely, switching between AAAAh and 5555h. If fewer lines are changing or if more adjacent lines share the same logic level, less current is required. The scale factor as shown in Figure 4 may also be applied to calculation of the current due to the outputs to comprehend more realistic conditions of data patterns on the external bus.

5.1.3 Control Outputs

CLKOUT is the primary control output of concern in terms of power consumption because it is switching most rapidly. As discussed previously, if an external CLKOUT is not necessary, it can be disabled using the CLKOFF bit in the PMST register. Other control outputs, such as R/$\overline{W}$, $\overline{PS}$, $\overline{DS}$, $\overline{IS}$, $\overline{MSTRB}$, $\overline{IOSTRB}$, $\overline{MSC}$, $\overline{IAQ}$, CLKX, FSX, DX, BCLKX, BFSX, BDX, TOUT, and XF contribute to the overall power use depending on the system configuration. These outputs consume less current because they switch less frequently, but they may still be important to consider, especially, when designing for low power conditions.

The current required for driving each unloaded control output is approximately:

0.07 mA per MHz of switching frequency per output at 3 V

The additional current for external capacitive loads on these outputs should also be added where necessary and can be calculated as shown for the data bus.

The overall current contribution of the outputs is the sum of the individual groups as defined in the following equation:

$$I_{ext} = I_{add} + I_{data} + I_{ctrl} \quad (10)$$

where

I_{ext} is the total current due to the outputs

I_{add} is the current component due to address bus activity

I_{data} is the current component due to data bus activity

I_{ctrl} is the current component due to control output activity

5.2 Considerations of TTL and Other DC Loads

If any device outputs experience TTL or other predominantly DC loads, consideration must be made for these effects in the overall power calculation. The net result of DC loading on an output is that the current required to drive that output is increased by the magnitude of the average dc-source-current loading on the output.

As an example, consider a DC loading of 300 μA of source current per bit on the address bus outputs when driving alternating 1's and 0's with a 50% duty cycle. Here, the increase in current is calculated as:

$$0.50 \times 300\ \mu A \times 16 = 2.4\ mA$$

In this case, an extra 2.4 mA of current is added to the current value required to drive the capacitive portion of the address bus output loading calculated as described above.

6 Total Power Dissipation

The previous sections have discussed power components used by several different sections of the TMS320LC54x. These sections have been discussed separately to illustrate the contributions of each section. The total power consumption of the device is the sum of the individual components. This total current value is determined as the total current supplied to the device through all of the V_{DD} inputs.

Multiple V_{DD} and V_{SS} pins are present on the device and may be routed to separate internal components. Consequently, all of the V_{DD} connections or all of the V_{SS} connections may not be common internally. All V_{SS} pins can be connected together externally. On single voltage supply devices, all CV_{DD} and DV_{DD} pins can be connected together externally. On dual power supply devices, all CV_{DD} pins should be connected together and all DV_{DD} pins should be connected together, but the two groups should be connected to their respective power planes separately. In each case, the V_{DD} and V_{SS} pins should be connected to the power and ground planes through as low an impedance as possible.

6.1 Calculation of Total Supply Current

Once all of the current components are calculated for unique periods of device activity, calculation of the total device power is straightforward. The separate components can be added to determine the overall current. Note that the CPU current measurements given in Table 3 and Appendix B include the current necessary for the clock-generation circuitry running in PLL multiply-by-one mode. If a different clock-generation mode is being used, these values can be adjusted by adding or subtracting the difference in current used by different clock modes (See Table 2).

To calculate the overall device current, a set of steps may be followed which examine the power issues discussed previously. These steps provide an approach to estimating the actual device power use. Some additional design considerations, which are discussed later in this report, can be employed to reduce power consumption.

6.2 Steps to Calculate Overall Device Power Consumption

1. Algorithm partitioning

   The algorithm under consideration should be broken into sections of unique device activity and the power requirements for these sections calculated separately. The sections can then be time-averaged to determine the overall device current requirement.

2. CPU activity

   a. The current demand due to CPU activity can be determined by examining the code and determining the time-averaged current for each algorithm partition from data in Appendix B. Note that the data provided in Appendix B was measured with the TMS320LC54x running in PLL multiply-by-one clock mode and the current required by the clock generation circuitry is included in the measurements. If a different clock mode is used, these values may need to be adjusted. Table 2 shows the relative current use of each clock mode. Also, bear in mind that most measurements in this report are based on mA per MHz of CLKOUT, not CLKIN. If the clock mode used is changed, the CLKIN speed also must be changed to generate the same CLKOUT frequency.

   b. When considering CPU current use, remember to consider the effects of the RPT instruction in the algorithm. RPT usually lowers the current required to execute a given instruction except in the case of multiply instructions, which increase current use due to automatic address generation when in repeated mode.

3. Memory Usage

   Scale the current calculated in step 2 based on memory usage.

   a. Use of on-chip memory requires less current than off-chip memory (because of the additional current due to the external memory interface).

   b. Running code from internal ROM requires less current than running from internal RAM.

4. Peripherals

   Consider the additional current required by use of the timer, standard serial port, buffered serial port, and host-port interface.

5. Current due to outputs

   a. Consider the current required by the algorithm to operate the external address and data buses. Scale these values to include the effects due to capacitive loading on the address and data buses.

   b. Include the current necessary to operate other fast switching outputs (CLKOUT, $\overline{\text{IAQ}}$, R/$\overline{\text{W}}$). Remember to consider output pins the

peripherals may be driving (i.e., TOUT from the timer, or CLKX/FSX/DX from the serial port). The capacitive loads on these individual outputs should be considered and the current requirements scaled accordingly.

c. Include the current necessary to operate the slow switching outputs ($\overline{\text{HOLDA}}$, $\overline{\text{IACK}}$). The capacitive loads on these individual outputs should also be considered and the current requirements scaled accordingly. Since these outputs are switching more slowly than other outputs, they require less current and may have little effect on the overall current calculation.

d. Include the current requirements of TTL and other DC loads on any outputs.

6.3 Calculation of Average Current

If power supply current is observed over the full duration of device activity, different segments of activity exhibit different current levels for different lengths of time. For example, a program may spend 80% of its time performing internal operations and drawing a current of 35 mA, but spend the remaining 20% of its time performing full speed writes to an external device while drawing 80 mA.

While identifying peak current levels is important in order to establish power supply requirements, determining average current is often more important. This is particularly significant if periods of high peak current are short in duration. Average current can be obtained by performing a weighted summation of the current due to various independent program segments over time. In the example just mentioned, the average current can be calculated as follows:

$$I = (0.8 \times 35\ \text{mA}) + (0.2 \times 80\ \text{mA}) = 44\ \text{mA} \quad (11)$$

6.4 Effects of Temperature and Supply Voltage on Device Operating Current

Two system factors, temperature and supply voltage, affect all of the current components equally. The effects of these factors can be included after the total device current has been calculated. Supply voltage and operating temperature should always be maintained within the required device specifications.

Device operating current is proportional to temperature; however, its variation due to temperature is small, and therefore, generally not significant. The device power measurements made in this report were taken at room temperature. If necessary, to account for absolute worst case effects including temperature across the total operating range of the device, power supply current values can be scaled approximately +/– 1% for operation at device temperature extremes above and below room temperature respectively.

Power supply current requirements also vary proportionally to V_{DD} supply voltage levels. Figure 5 shows the variation in device operating current (as a scale factor) with respect to V_{DD} supply voltage. This data can be used to determine the scale factor that is applied to the total current calculated for any given period of device activity.

Effects due to temperature and supply voltage are the final factors to be applied to current values calculated for any given period of device activity. After scaling for these factors, actual current levels are produced, and average current for the entire duration of device operation can then be calculated as described in the following sections.

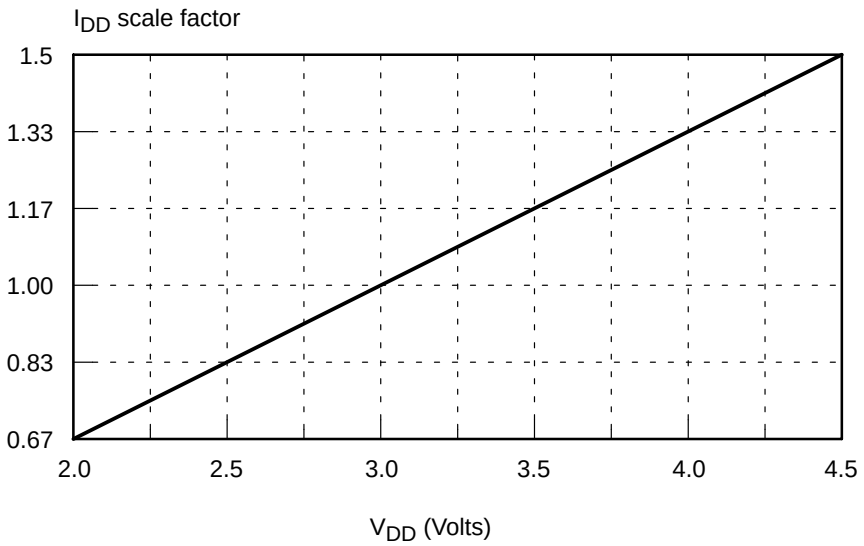

Figure 5. I_{DD} Variation With Respect to V_{DD} Supply Voltage

6.5 Thermal Management Considerations

Heating characteristics of the TMS320LC54x device are dependent upon power dissipation which, in turn, is dependent on power supply current. When making thermal management calculations, several considerations must be made that relate to the manner in which the power supply current contributes to power dissipation and to the TMS320LC54x thermal characteristics' time constant. Depending on sources and destinations of current in the device, some current contributions to I_{DD} do not constitute a component of power dissipation in the TMS320LC54x. Consequently, if the total current flowing into V_{DD} is used to calculate power dissipation at a given value of V_{DD}, excessive values for power dissipation are obtained.

Power dissipation is defined as:

$$P = I \times V$$

where

P is power

I is current, and

V is voltage

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, the current components should be subtracted out of the total supply current value, and their contribution to power dissipation calculated separately and then added to the total power dissipation (See Figure 6). If this is not done, these currents resulting from driving a high logic level into a DC load cause unrealistically high power dissipation values. The error occurs because the resulting currents from driving a logic high appear as a portion of the current used to calculate power dissipation due to V_{DD} at a given value.

Furthermore, external loads draw output current only when outputs are being driven high, because when outputs are in a logic low state, the device is sinking current supplied from an external source. Therefore, the power dissipation due to outputs being driven low does not receive a contribution through I_{DD}, but does contribute to power dissipation with a magnitude of:

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

where

V_{OL} is the low level voltage, and

I_{OL} is the current being sunk by the output as shown in Figure 6.

The power dissipation component due to outputs being driven low should be calculated and added to the total power dissipation.

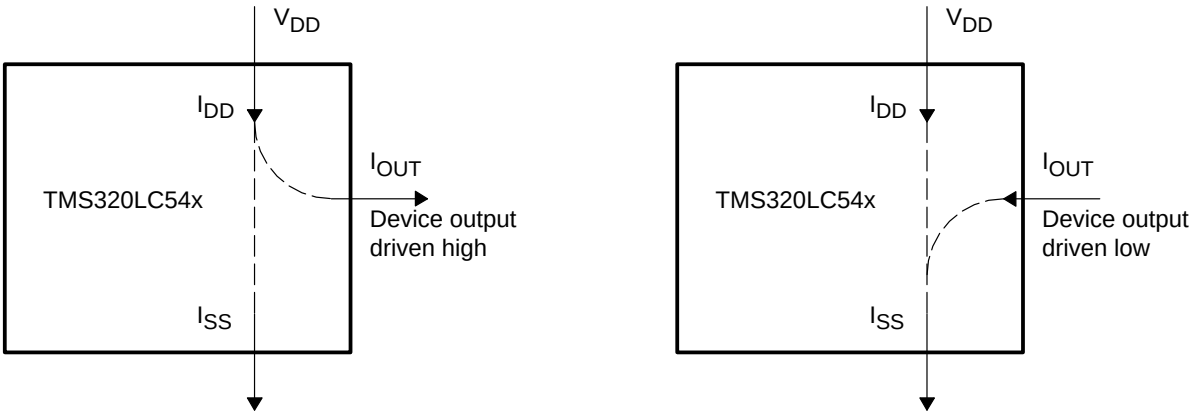

Figure 6. Device I/O Currents

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

When using power dissipation values to determine thermal management, the average power should be used unless the time duration of the individual periods of device activity is long. The thermal characteristics of the TMS320LC54x package are exponential in nature with a time constant on the order of several minutes. Therefore, when the device is subjected to a change in power, the package requires several minutes or more to reach thermal equilibrium. If the time duration of periods of device activity exhibiting high power dissipation values is short (on the order of a few seconds or less) in comparison to the package thermal characteristics' time constant, the average power, calculated in the same manner as the average current described previously, should be used.

Maximum device temperature should be calculated on the basis of actual time duration for the periods of device activity involved. For example, if a particular device activity lasts for twelve 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 period of device activity (including all considerations described above) and performing a time average of these values, rather than simply multiplying the average current by V_{DD}, as described in the previous subsection.

When the average power has been determined, specific device temperature calculations can be made using the thermal impedance characteristics for the type of package being used.

7 System Design Considerations for Minimizing Power Dissipation

There are several issues that can be considered in the design process to reduce power consumption of a particular TMS320LC54x application. Although some of these issues have already been covered in this report, they are included here for convenience.

7.1 System Clock and Switching Rates

The power consumption of the TMS320LC54x device is directly proportional to the system clock (CLKOUT) switching speed. If the clock speed doubles, the current doubles. Obviously, power can be saved by operating the device at the lowest clock speed possible that still meets the specifications for the device and the requirements of the application.

As the clock speed increases, the current increases proportionally, but the time required to execute the same operation decreases proportionally. If the clock speed doubles, twice the current is required but half of the time is required for the same operation. For example, consider a section of code that runs for 500 clock cycles at 0.8 mA per MHz. At a CLKOUT speed of 10 MHz, the code requires 8 mA for 50 µs (500 x 100-ns cycles). If the clock speed is doubled to 20 MHz, the current required doubles to 16 mA but the time duration required for the operation is cut in half, to 25 µs. Figure 7 illustrates this behavior. So the energy required to complete the operation is the same since it depends only on the number of internal logic state transitions the device makes.

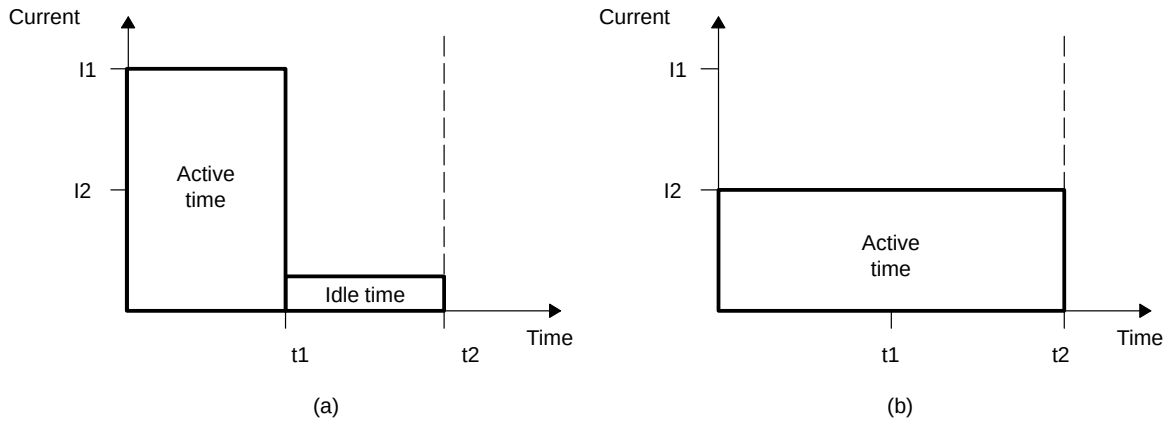

Figure 7. Algorithm Current Use Versus Clock Speed

It would appear that there is no difference between operating the device at the lowest speed possible for the application and at the highest speed supported by the device since the energy required by the algorithm is the same. So why not operate the device at its highest speed even if all of the MIPS capacity is not needed?

Consider the two cases where the algorithm is completed quickly and the remaining time is spent in a low-power mode (IDLE) as shown in Figure 7(a), and the case where the system clock is slowed down so the algorithm requires all of the time available as in Figure 7(b). In both cases, the energy required to perform the algorithm is the same. However, in case (a), power is also consumed for the rest of the time in the IDLE mode. This does not happen in case (b) because there is no time spent in IDLE mode. For this reason, if the application does not require the entire MIPS capability of the device, it is more power efficient to slow down the system clock to minimize the IDLE time.

7.2 CLKOUT Switching

TMS320LC54x devices include a power conservation feature that provides the option to disable external activity on the CLKOUT pin. The CLKOUT pin is provided as a master processor clock reference which can be used by external devices to synchronize with the TMS320LC54x CPU. In a system where there are no external devices that need a clock reference, there is no need to consume power operating the CLKOUT pin driver circuitry. In this case, the CLKOUT pin can be disabled so it no longer consumes power. The internal processor clock remains unaffected.

The CLKOUT pin is controlled by the CLKOFF bit in the PMST register. If this bit is 1, CLKOUT is inactive and no signal reaches the CLKOUT pin. If the bit is 0, the clock signal is available at the pin and it operates normally.

At 3 V, the unloaded CLKOUT pin requires 0.09 mA per MHz (4.5 mA at 50 MIPS). This amount of current can be saved if the CLKOUT output is disabled. See the *TMS320LC54x Reference Set, Volume 1*, for more details on CLKOUT switching.

7.3 Stopping the Internal Processor Clock

Many portable applications, such as pagers and cellular telephones, are battery operated and are consequently very sensitive to power consumption. When these portable devices are not in use, the power being drawn from the batteries can be of critical importance. The IDLE modes mentioned previously can be used to disable part or all of the TMS320LC54x processor and provide a versatile method of managing power use during periods of low device activity. However, when an application is turned off, it may be desirable to stop the processor clock to achieve absolute lowest power consumption. Since the TMS320LC54x family of processors is designed using a fully static CMOS technology, stopping the processor clock is possible without any corruption of internally stored data as long as adherence to the timing specifications in the data sheet is maintained.

Several conditions are necessary to safely stop the input clock (CLKIN) and achieve absolute lowest device power consumption:

- The processor should be in IDLE3 mode.
- Divide-by-two clock mode should be selected.
- CLKIN should be synchronously stopped in a logic high state while meeting the rise/fall time maximum and pulse duration minimum specifications as indicated in the data sheet.
- All unused pins configured as inputs should be driven high or pulled-up using 10k resistors (with the exception of $\overline{\text{TRST}}$ which has an internal pulldown and should be left unconnected).

Under these conditions, the device consumes power in the nanoampere range. The conditions described above must be met for proper operation of the processor. If these conditions are not met, the following may occur:

- If the clock is stopped when the processor is not in IDLE mode, or if the input clock timing fails to meet the data sheet specifications, data corruption and/or erratic operation may result.
- If the clock is stopped in one of the PLL clock modes, the PLL continues to attempt to track the input clock (which is not present) and current use varies as it does so, even up to the milliampere range. The input clock can be safely stopped in PLL clock mode (after IDLE mode has been executed), but this does not result in the lowest power state. The PLL also requires a transitory phase to become stable after the input clock is restarted. The device should not be removed from IDLE mode until the transitory phase has been completed. See the data sheet for more details about the PLL transitory phase.

- Any unused inputs left floating may drift to an intermediate voltage between the power rails and cause internal buffer levels to float in transition regions. This causes current consumption and the device current increases with each input that is not controlled. Device pins that have built-in bus keepers or pullups, as mentioned previously, do not require additional external pullups.

7.4 On-Chip Versus Off-Chip Memory

On-chip memory requires less power because the external memory interface is not driven during internal accesses. Minimizing accesses to external memory space lowers the device current requirement.

7.5 On-Chip ROM Versus On-Chip RAM

Use of internal ROM requires less power than use of internal RAM. Code execution from internal ROM requires about 10% less CPU current than the same code executing from internal SARAM.

7.6 Capacitive Loading of Outputs

Increased capacitive loading on device outputs increases the current required to drive the output pins. Minimizing this loading minimizes the current required to operate these pins.

7.7 Address Visibility

When address visibility is enabled, addresses are passed to the external address bus even during internal memory accesses. This feature is very useful primarily as a development and debugging tool, but it should be disabled when debug is complete to minimize activity on the external memory interface.

7.8 DC Loading of Outputs

DC loading of outputs due to TTL or other sources should be minimized to conserve power.

7.9 Power-Down Mode

When device CPU activity is not necessary, the device should be placed in one of the IDLE modes to conserve power. This is achieved by executing one of the IDLE instructions or by a logic low input on the $\overline{\text{HOLD}}$ pin. In IDLE1 mode, the internal clocks to the CPU are shut off but all peripherals and the PLL remain active. In IDLE2 mode, the CPU, timer, and standard serial ports are disabled and only the BSP, the HPI, and the PLL remain active. In IDLE3, the CPU, all peripherals, and the PLL are disabled, although the BSP and HPI can be operated in IDLE3 mode under special conditions. These states conserve a considerable amount of power compared to normal operation. The IDLE states are easily exited by the occurrence of an internal or external interrupt. See the *TMS320LC54x Reference Set, Volume 1*, for more information.

8 Power Calculation Example

To illustrate the techniques described earlier in this report, this section presents an FIR filter as a simple example of a power dissipation calculation. The program reads in input samples from an external data memory location at a rate of 8 kHz. After each input sample is read, the program performs an FIR filter and then writes all of the current data samples in the filter out to external data memory. When this operation is complete, the processor goes into IDLE mode while waiting for the next timer interrupt, which indicates it is time to read another input sample. The entire process then repeats. A listing of the program is provided in Appendix A.

8.1 System Environment

The TMS320LC545 processor is running at 20 MIPS (20 MHz CLKOUT) in PLL multiply-by-one clock mode. The V_{DD} supply voltage is 3.0 volts. The external address bus, data bus, and relevant control signal outputs are unloaded. Address visibility mode is disabled and CLKOUT is disabled. Zero wait-states are assumed for all external memory accesses.

8.2 Algorithm Partitioning

The example application is separated into three sections and the power dissipation calculated for each section separately. These sections are chosen based on program activity. In the first section, the device reads in a single input-data word and runs the FIR filter. In the second section, the processor writes a table of the current data sample values in the filter to an external memory location. In the third section, the processor enters IDLE mode and waits for the next timer interrupt. When the interrupt occurs, the process is repeated with a new data sample. For each of these three sections, the current requirements due to the internal CPU activity, the internal peripheral activity, the external memory interface activity, and the external loads is calculated. After the current requirements of each of the sections is calculated, these sections are time-averaged to determine the overall current use by the application.

8.3 Timer Configuration and Activity

Since the application reads input-data samples at a rate of 8 kHz, the timer is configured to generate a timer interrupt every 125 microseconds. To generate this time period, the timer is loaded with a count of 2500 clock cycles as the period (20 MHz/8 kHz = 2500). The timer interrupt is unmasked so it can be used to exit IDLE mode, but interrupts are disabled (INTM = 1) so there is no need to generate a timer interrupt service routine.

The timer runs during all of the algorithm partitions mentioned above so its contribution to each of the sections is equal. The estimated current use for the timer is 0.007 mA per MHz of CLKOUT.

Since no other on-chip peripherals are used in this algorithm, they are disabled and their current requirement is zero.

Timer current use:
0.007 mA per MHz of CLKOUT for 100% of the execution time

8.4 Filter Section

The first section occurs after the timer interrupt has indicated that it is time to read an input-data sample. A single input-data word is read from external data memory at address 2000h and copied into a working buffer at internal data memory address 0300h. Then the FIR filter routine runs. The filter is 256 taps and the coefficients are assumed to already have been stored in a table called "COEFF". When the filter routine completes, the result is stored at data memory variable "result". Reading the input samples and storing the results requires very few clock cycles (5 cycles) compared to the FIR filter calculation (259 cycles), so the current use for this section is primarily due to the filter calculation; therefore, the other instructions can be considered negligible.

When the MACD instruction is repeated, it executes in single cycles unless limited by the memory configuration. In this case, the coefficients and data points are stored in on-chip DARAM. This memory configuration allows the CPU to complete one MACD instruction per CLKOUT cycle. If a different memory configuration was used which caused wait states to occur during the MACD execution, the current value would have to be scaled according to the speed of actual execution. Table B–1 indicates that a repeated, single-cycle, MACD at 3 V requires approximately 0.8 – 1.1 milliamps per MHz of CLKOUT. For this example, the worst case value of 1.1 is considered.

Internal CPU contribution:
1.1 mA per MHz of CLKOUT

Since both the coefficients and the data samples have been stored on-chip, the external memory interface is not active during this section and does not consume power. So the following components contribute to the current required during the filter section routine:

External interface contribution:
0

External loads contribution:
0

8.5 External Table Write Section

In the second section, the processor writes a table of data samples (from the working buffer in DARAM) to external data memory at address 3000h. This section requires only a repeated MVDD instruction (block move from data memory to data memory). Since this instruction is repeated 256 times, it dominates the current use in this section and the current use of the other instructions can be considered negligible.

Since all memory in this example requires zero wait-states, the MVDD instruction requires two cycles of each execution because external data writes require a minimum of two cycles. Details of the timing requirements and the behavior of an external memory interface write cycle are shown in the data sheet.

First, the internal CPU contribution to the operating current requirements is calculated. From Table B–1, a repeated MVDD instruction requires approximately 0.8 milliamps per MHz of CLKOUT. This value, however, was measured for single-cycle (on-chip to on-chip) transfer. Since the destination for the data in this example is in external memory, the instruction requires two cycles per execution. Since the execution speed of this instruction is cut in half, the current required to perform this instruction is also cut in half to 0.4 mA per MHz of CLKOUT.

Internal CPU contribution:
0.04 mA per MHz of CLKOUT

During the repeated MVDD instruction, the external address bus is not switching since all writes go to address 3000h. Consequently, the current consumed by the external address bus is negligible. The current required to drive the external data bus varies, depending on the data pattern being written. If the data is unchanging from cycle to cycle, no current is required because no nodes are changing state. If the worst-case data happened (AAAAh, 5555h, AAAAh, ... for example), the current required would be 1.44 mA per MHz of switching frequency, as determined previously. Since the external writes each take two cycles, they occur at a rate that is one-half of the speed of CLKOUT. Also, the data bus requires two writes to switch data and switch back, so the data bus switches at one-half the rate of the writes, or one-fourth the rate of CLKOUT; therefore, the worst case current required to operate the external data bus is calculated as:

$$(1.44) \times (1/4) = 0.36 \text{ mA per MHz of CLKOUT (worst case)} \quad (12)$$

Since it is unlikely that all data lines change on every write, an average value can be chosen between best case (no lines changing) and worst case (16 lines changing). This average value assumes that 8 lines change on each word. In this case, the current required to operate the external data bus would be:

$$8 \times (0.09) \times (1/4 \text{ of CLKOUT}) = 0.18 \text{ mA per MHz of CLKOUT (average case)} \quad (13)$$

Since the external data bus is unloaded in this example, there is no current contribution due to driving loads. In an actual application, this additional current requirement should be considered.

The only control pin operating during the repeated external writes is $\overline{\text{MSTRB}}$. Although other control pins are active before and after the repeated MVDD execution, they do not change during its execution. See the Memory Interface External Write Timing diagram in the TMS320LC54x data sheet for more detailed information. $\overline{\text{MSTRB}}$ switches at a rate one-half of the CLKOUT so the current required to operate the pin can be calculated as:

$$(0.09) \times (1/4 \times \text{CLKOUT}) = 0.045 \text{ mA per MHz of CLKOUT} \quad (14)$$

Since the external control signals are unloaded in this example, there is no current contribution due to driving loads. In an actual application, this additional current requirement should be considered.

The sum of these calculations indicates the overall supply current required during the external table write section:

Internal CPU requirement:	0.4	mA per MHz of CLKOUT
External address bus requirement:	0	mA per MHz of CLKOUT
External data bus requirement:	0.18	mA per MHz of CLKOUT
External control pins requirement:	0.045	mA per MHz of CLKOUT
Total output pin requirement:	0.225	mA per MHz of CLKOUT

8.6 Idle Section

During the Idle section, the processor enters IDLE1 mode and waits for the timer interrupt. When the timer interrupt occurs, the processor continues program execution with the filter section. It does not perform a timer interrupt service routine because the interrupts were disabled. In IDLE1 mode, the processor requires 0.12 mA per MHz of CLKOUT at 3 V when running in PLL multiply-by-one mode.

IDLE requirement:

0.12 mA per MHz of CLKOUT

Table 4 shows a summary of the current activity during the algorithm in mA per MHz of CLKOUT.

Table 4. Example Algorithm Current Activity (in mA per MHz of CLKOUT)

ALGORITHM PARTITIONS	CPU CURRENT	ON-CHIP PERIPHERAL CURRENT	OUTPUT CURRENT	EXTERNAL LOADS CURRENT	TOTAL DEVICE CURRENT
Filter section	1.1	0.007	0	0	1.107
Write output section	0.4	0.007	0.225	0	0.632
IDLE section	0.12	0.007	0	0	0.127

8.7 Determining the Time-Averaged Current

To determine the total current required by the device, the current required by each section of the algorithm is time-averaged. For this calculation, the time required to execute each section of the code must be determined. The Filter section requires 264 cycles to execute. At 20 MHz CLKOUT, execution of this section takes 13.2 μs. The external table write section requires 519 cycles to execute. Execution of this section takes 25.95 μs. The entire cycle repeats every 125 μs, so the remainder of this time (85.85 μs) is spent in the IDLE section. The contribution of the timer current is present 100% of the time.

The execution times can be expressed as a percentage of the total time as:

Filter section: (13.2 μs/125 μs) = 10.6% of total time

Table write section: (25.95 μs/125 μs) = 20.8% of total time

IDLE section: (85.85 μs/125 μs) = 68.6% of total time

The current required for each section is determined by multiplying the “milliamps per MHz” current by the CLKOUT speed and adding the frequency-independent component of the PLL current:

Filter section:	$(1.107 \times 20) + 0.8 = 22.94$ mA
Table write section:	$(0.635 \times 20) + 0.8 = 13.44$ mA
IDLE section:	$(0.127 \times 20) + 0.8 = 3.34$ mA

Finally, the time-averaged current is determined by multiplying the current required for each of the sections by the fractional time the section is executing, and then summing each of those contributions.

$$I_{total} = I_1 t_1 + I_2 t_2 + I_3 t_3 + I_4 t_4 \quad (15)$$

$$= (22.94)\,(0.106) + (13.44)\,(0.208) + (3.34)\,(0.686) = 7.52 \text{ mA}$$

8.8 Experimental Results

In order to confirm the values for current calculated in the example, the actual power supply current for this sample program was measured using the test setup previously described in this report. The actual measured current values are included below:

Filter section:	22.18 mA
Write output section	12.04 mA
IDLE section:	3.22 mA
Overall section:	7.06 mA

A listing of the sample program and a photograph of the actual current waveforms observed are included in Appendix A.

9 Summary and Conclusion

The power supply current requirements for the TMS320LC54x DSPs cannot be expressed simply in terms of operating frequency, supply voltage, and output capacitance. A more complete specification, one based on device activity, must be used to determine an accurate power supply current requirement. This application report has presented the information necessary to accurately analyze power supply current requirements. These requirements are based on the knowledge of various periods of device activity and their operation of the TMS320LC54x in terms of internal and external activity.

The power supply current requirements for the TMS320LC54x DSPs depend on system parameters as well as device activity. Dependencies related to system parameters include operating frequency, supply voltage, operating temperature, and output capacitance. The components related to device activity include CPU activity, peripheral activity, and external bus operations.

Taking into account the current effects and dependencies involved in analysis of device power dissipation, system design may be performed proactively to minimize device and system power dissipation. With the combination of high processing speed and low power dissipation design, the TMS320LC54x family of DSPs is an ideal solution for high-performance power-sensitive applications.

Appendix A Example Program Listing

```
          .mmregs
          .version 545
          .global INIT, result
          .bss    result,1
*********************************
*     Initialization
*********************************
      .text
INIT
      ssbx intm                 ;disable interrupts
      stm  #00000h,IMR          ;mask all interrupts
      stm  #0ffffh,IFR          ;clear all pending interrupts
      stm  #01940h,ST1          ;configure ST1
      stm  #0ffe4h,PMST         ;configure PMST
      stm  #00000h,SWWSR        ;zero wait states in all spaces
      stm  #0F802h,BSCR         ;configure bank switching
                                register
      stm  #00030h,TCR          ;stop timer
      stm  #009c4h,PRD          ;timer period = 2500
      stm  #0028h,@0022h        ;BSPC – buffered serial port in
                                reset
      stm  #0028h,@0032h        ;SPC – standard serial port in
                                reset
      stm  #00008h,IMR          ;enable timer interrupt
      stm  #00020h,TCR          ;start timer
      stm  #02000h,ar2          ;set AR2 to input data memory
                                address
      stm  #00300h,ar3          ;set AR3 to buffer data memory
                                address
      stm  #03000h,ar4          ;set AR4 to output data memory
                                address
      ld   #0,DP                ; initialize data page to 0
      stm  #0ffffh,IFR          ;clear all pending interrupts
************************
IDLE Section
************************
begin
      stm  #0300h,ar3           ;reset buffer starting address
      idle 1                    ;wait for timer interrupt
************************
Filter Section
************************
      mvdd *ar2,*ar3            ;copy input data word from
                                external
                                ; memory address 2000h to buffer
      stm  #0ffffh,IFR          ;clear timer interrupt flag
                                filter
      stm  #003ffh,ar3          ;point to the end of the buffer
      ld   #0,a                 ;clear accumulator A
      rpt  #255                 ;run 256-tap FIR filter
           macd   *ar3–,coeff,a
```

```
           sth    a,result       ;store result
*****************************
External Table Write Section
*****************************
write
       stm  #00300h,ar3          ;point to top of buffer
       rpt  #255                 ;write entire data buffer to
       mvdd *ar3+,*ar4           ; external memory address 3000h
       b    BEGIN

*****************************************************************
```

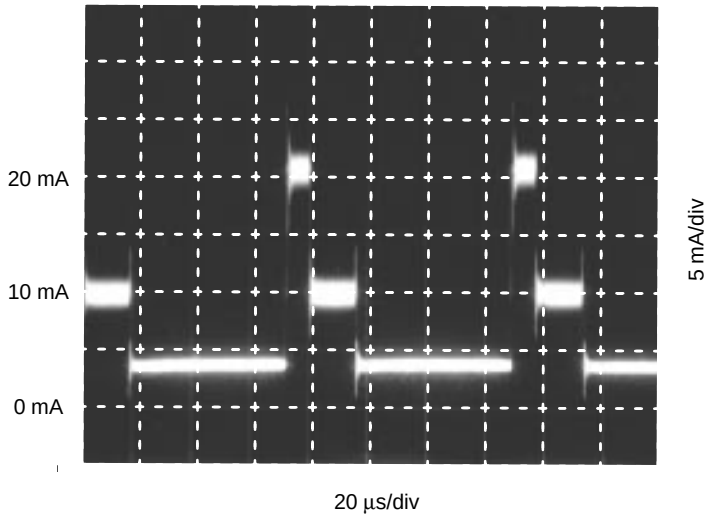

Figure A–1. Actual Measured Current

Appendix B TMS320LC54x Instruction Set Power Characteristics

The measurements included below are intended to provide a general characteristic of the supply current requirements of the TMS320LC54x CPU. Current use of any instruction may vary due to source or destination memory type, wait states, pipeline conflicts, and/or data patterns. The measurements given were made at V_{DD} = 3.0 Volts and a temperature of 25°C. PLL multiply-by-one clock mode was used unless otherwise specified. Where a range of values is shown, the low value represents the current use of the instruction with low-data complexity and the high value represents the current use at high-data complexity. The '~' symbol indicates the measurement was not performed. Units are mA per MHz of CLKOUT.

Table B–1. TMS320LC54x Instruction Set Power Characteristics

INSTRUCTIONS	RPT 3 V	INLINE 3 V
GENERAL FUNCTIONS		
NOP	0.3	0.4
IDLE1	~	0.12
IDLE2	~	0.03
IDLE3	~	0
AUXILIARY REGISTER FUNCTIONS		
CMPR	~	0.5
MAR	0.3	0.4
BRANCH AND CONTROL FUNCTIONS		
B	~	0.5
CALL, RET	~	0.6
RSBX	~	0.5
SSBX	~	0.5
LOAD/STORE FUNCTIONS		
LD (load data page with long constant)	~	0.5
LD (load accumulator with long constant)	~	0.7
LD (load accumulator from data memory location)	~	0.9
LDM	~	0.8
LDR	~	0.8
LTD	0.5	0.7
ST	0.4	0.8
STH	0.4	0.8
STL	0.4	0.8
STLM	0.4	0.8

Table B–1. TMS320LC54x Instruction Set Power Characteristics (Continued)

INSTRUCTIONS	RPT 3 V	INLINE 3 V
MULTIPLIER FUNCTIONS		
MAC[R], MACA, MAS, MPY, MPYA, MPYU (s)	0.5 – 0.8	0.7 – 1.0
MAC[R], MACSU, MAS, MPY (xy)	0.6 – 1.0	0.8 – 1.2
MAC[R], MPY (lk)	0.4	0.7
MAC[R], MPY (s,lk)	0.4 – 0.8	0.7 – 0.9
MACD	0.8 – 1.1	0.8 – 0.9
MACP	0.6 – 1.1	0.8 – 0.9
SQUR, SQURA, SQURS (s)	0.4 – 0.8	0.6 – 1.0
SQUR (acc)	0.4	0.5
SPECIAL MULTIPLIER FUNCTIONS		
SQDST	0.6 – 1.1	0.8 – 1.3
FIRS	0.9 – 1.2	~
LMS	0.7 – 1.1	0.9 – 1.3
POLY	0.9	1.1
PARALLEL OPERATION FUNCTIONS		
LD || MAC	0.6 – 1.1	0.8 – 1.3
LD || MAS	0.6 – 1.0	0.8 – 1.2
ST || LD, ST || ADD, ST || SUB	0.5 – 0.8	0.7 – 1.0
ST || MPY, ST || MAC, ST || MAS	0.5 – 0.9	0.7 – 1.1
DOUBLE-PRECISION FUNCTIONS		
DLD, DADD, DADST, DSADT, DRSUB, DSUB, DSUBT	0.5 – 0.9	0.7 – 1.1
DST	~	0.6 – 0.8
I/O AND DATA MEMORY FUNCTIONS		
DELAY	0.8	1.0
MVDD (xy)	0.8	1.0
MVDP, MVPD (s)	0.8	~
MVDK, MVKD	0.8	~
POPD, POPM, PSHD, PSHM (s)	0.8	1.0

NOTES: 1. Accumulator shift values have little effect on the current required to execute the instruction.
2. Type of auxiliary register increment/decrement (single/indexed/bit-reversed) has little effect on the current required to execute the instruction.
3. Current values are shown as a range when the current required to execute the instruction is data-pattern dependent.
4. Instruction syntax type is denoted as follows:
   - (s) Single data-memory operand addressing
   - (xy) Dual data-memory operand addressing
   - (lk) Single operand is a 16-bit constant
   - (s, lk) Dual operands are a data-memory operand and a 16-bit constant
   - (acc) Operand is the accumulator

Table B–1. TMS320LC54x Instruction Set Power Characteristics (Continued)

INSTRUCTIONS	RPT 3 V	INLINE 3 V
I/O AND DATA MEMORY FUNCTIONS (CONTINUED)		
PORTR (s) [CPU component only]	0.4	0.6
PORTW (s) [CPU component only]	0.6	0.9
READA (s) [1 cycle in RPT, 5 cycles inline]	0.7	0.5
WRITA (s) [1 cycle in RPT, 5 cycles inline]	0.8	0.6
ARITHMETIC FUNCTIONS		
ABS	0.3	0.5
ADD (s)	0.4 – 0.7	0.6 – 0.9
ADD (xy)	0.8	1.0
ADD (lc)	0.4	0.7
ADDM	~	0.7
CMPS	0.4	0.6
MIN	0.3	0.5
MAX	0.3	0.5
NEG	0.4	0.6
SAT	0.3	0.5
SFTA	0.3	0.5
SUB (s)	0.4 – 0.7	0.6 – 0.9
SUB (xy)	0.9	1.1
SUB (lk)	0.4	0.7
SUBC	0.5 – 0.7	0.7 – 0.9

NOTES: 1. Accumulator shift values have little effect on the current required to execute the instruction.
2. Type of auxiliary register increment/decrement (single/indexed/bit-reversed) has little effect on the current required to execute the instruction.
3. Current values are shown as a range when the current required to execute the instruction is data-pattern dependent.
4. Instruction syntax type is denoted as follows:
   - (s) Single data-memory operand addressing
   - (xy) Dual data-memory operand addressing
   - (lk) Single operand is a 16-bit constant
   - (s, lk) Dual operands are a data-memory operand and a 16-bit constant
   - (acc) Operand is the accumulator

Table B–1. TMS320LC54x Instruction Set Power Characteristics (Continued)

INSTRUCTIONS	RPT 3 V	INLINE 3 V
LOGIC FUNCTIONS		
AND (s)	0.4	0.6 – 0.8
AND (lk)	0.3	0.7
ANDM	~	0.8
BIT, BITF, BITT	0.6	0.8
CMPL	0.4	0.6
CMPM	0.4	0.6
OR (s)	0.4	0.6 – 0.8
OR (lk)	0.3	0.6
ORM	~	0.8
ROL, ROR	0.4	0.6
ROLTC, SFTL	0.3	0.5
XOR (s)	0.4	0.6 – 0.8
XOR (lk)	0.3	0.7

NOTES: 1. Accumulator shift values have little effect on the current required to execute the instruction.
2. Type of auxiliary register increment/decrement (single/indexed/bit-reversed) has little effect on the current required to execute the instruction.
3. Current values are shown as a range when the current required to execute the instruction is data-pattern dependent.
4. Instruction syntax type is denoted as follows:
   - (s) Single data-memory operand addressing
   - (xy) Dual data-memory operand addressing
   - (lk) Single operand is a 16-bit constant
   - (s, lk) Dual operands are a data-memory operand and a 16-bit constant
   - (acc) Operand is the accumulator

IMPORTANT NOTICE

Texas Instruments (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 the specifications applicable at the time of sale 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.

Certain applications using semiconductor products may involve potential risks of death, personal injury, or severe property or environmental damage ("Critical Applications").

TI SEMICONDUCTOR PRODUCTS ARE NOT DESIGNED, INTENDED, AUTHORIZED, OR WARRANTED TO BE SUITABLE FOR USE IN LIFE-SUPPORT APPLICATIONS, DEVICES OR SYSTEMS OR OTHER CRITICAL APPLICATIONS.

Inclusion of TI products in such applications is understood to be fully at the risk of the customer. Use of TI products in such applications requires the written approval of an appropriate TI officer. Questions concerning potential risk applications should be directed to TI through a local SC sales office.

In order to minimize risks associated with the customer's applications, adequate design and operating safeguards should be provided by the customer to minimize inherent or procedural hazards.

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.

Copyright © 1996, Texas Instruments Incorporated