

Fault Assisted Circuits for  
Electronics Training (FACET)

## Courseware Sample

31946-F0







FAULT ASSISTED CIRCUITS FOR  
ELECTRONICS TRAINING (FACET)

COURSEWARE SAMPLE

by  
the Staff  
of  
Lab-Volt (Quebec) Ltd

**Copyright © 2001 Lab-Volt Ltd**

All rights reserved. No part of this publication may be reproduced, in any form or by any means, without the prior written permission of Lab-Volt Quebec Ltd.

# Table of Contents

## **Courseware Outline**

Digital Signal Processor .....	V
--------------------------------	---

## **Sample Exercise from Digital Signal Processor**

Ex. 2-2 Memory Space .....	3
----------------------------	---

## **Other samples extracted from Digital Signal Processor**

Unit Test .....	23
-----------------	----

## **Instructor's Guide Sample Extract from Digital Signal Processor**

Ex. 3-1 The Program Controller .....	27
--------------------------------------	----



# Courseware Outline

## **DIGITAL SIGNAL PROCESSOR**

### **Unit 1 DSP Trainer Familiarization**

Ex. 1-1 Introduction to the DSP Circuit Board

Ex. 1-2 The Assembler and Debugger

Ex. 1-3 Processor Arithmetic

### **Unit 2 CPU Architecture**

Ex. 2-1 The Central Arithmetic Logic Unit

Ex. 2-2 Memory Space

Ex. 2-3 Addressing

### **Unit 3 Program Execution**

Ex. 3-1 The Program Controller

Ex. 3-2 The Pipeline

### **Unit 4 Basic I/O**

Ex. 4-1 DSP Peripherals

Ex. 4-2 Digital Signal Processing: The FIR Filter

### **Appendix A Help Pages**

### **Appendix B New Terms and Words**

### **We Value Your Opinion**



Sample Exercise  
from  
Digital Signal Processor



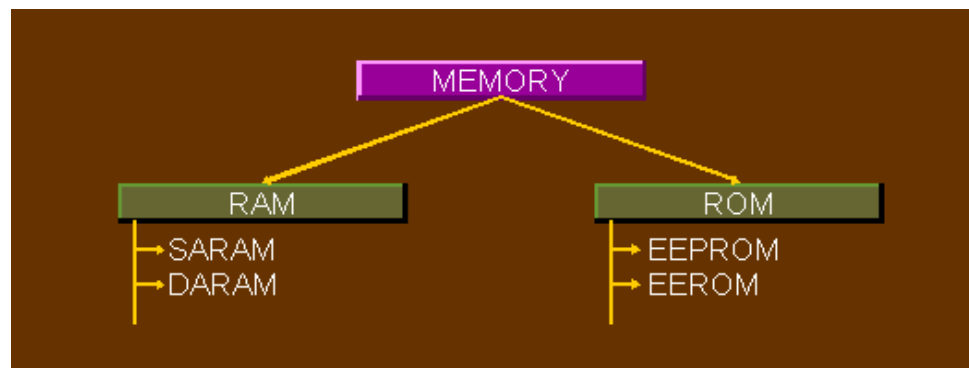
## Memory Space

### EXERCISE OBJECTIVES

Upon completion of this exercise, you will be familiar with the basic characteristics of the modified Harvard architecture, as used by DSPs.

### DISCUSSION

Memory is an important part of any microcomputer or microprocessor. In computers like the one you are using, **memory** is used to store program information such as the program code for the C5x VDE and it is also used to store data information.

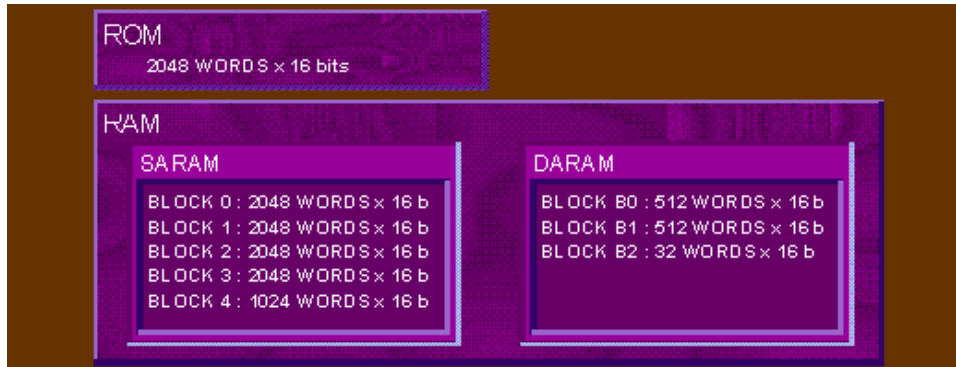


The DSP contains on-chip memory and is also able to **access** off-chip memory through its external address and data buses. On-chip memory is usually of two types:

- ROM (Read Only Memory) is used to store program code during the manufacturing process. ROM is a **non-volatile** memory because it retains its data after the processor has shut down.
- RAM (Random Access Memory) is used to store temporary program information. RAM is a **volatile** memory because when power is removed the stored information is lost.

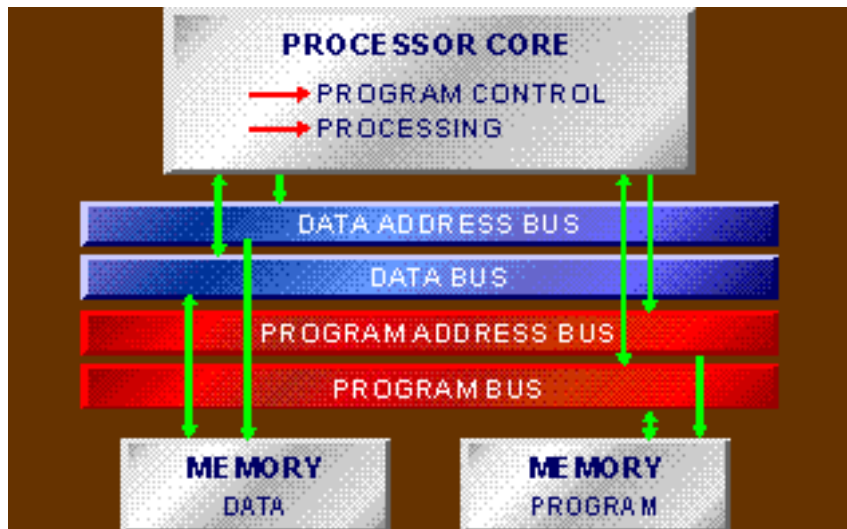
Both categories of memory (ROM and RAM) are found on-chip (inside the DSP). The **allocation in memory space** of these types of memory is able to be configured in various ways.

# Memory Space



The TMS320C50 DSP uses two types of on-chip RAM:

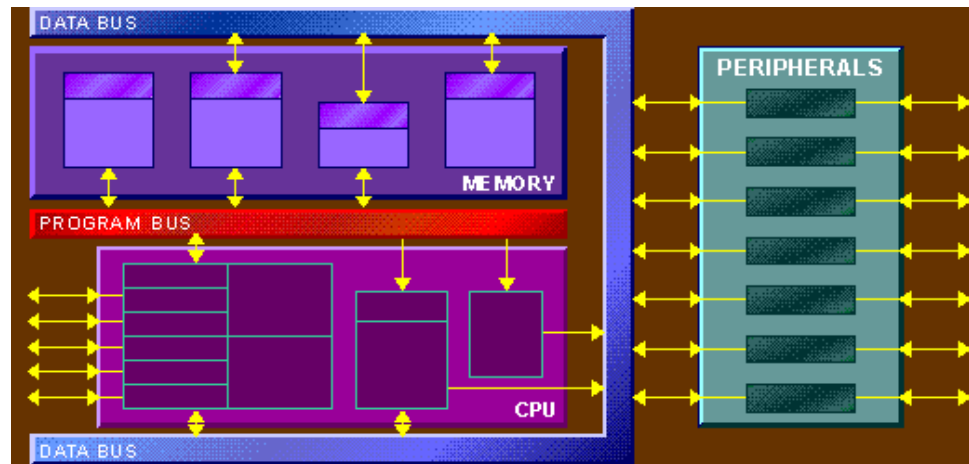
- SARAM (Single-Access RAM) - An SARAM **memory block** can be written to or read from once within one instruction cycle.
- DARAM (Dual-Access RAM) - A DARAM block can be read from and written to in the same instruction cycle.



The **Harvard architecture** has two parallel buses.

One bus (the PB) is dedicated to addressing and transport of programming information and the other (the DB) is dedicated to addressing and transport of data. Two parallel buses allow program and data memory to be accessed simultaneously.

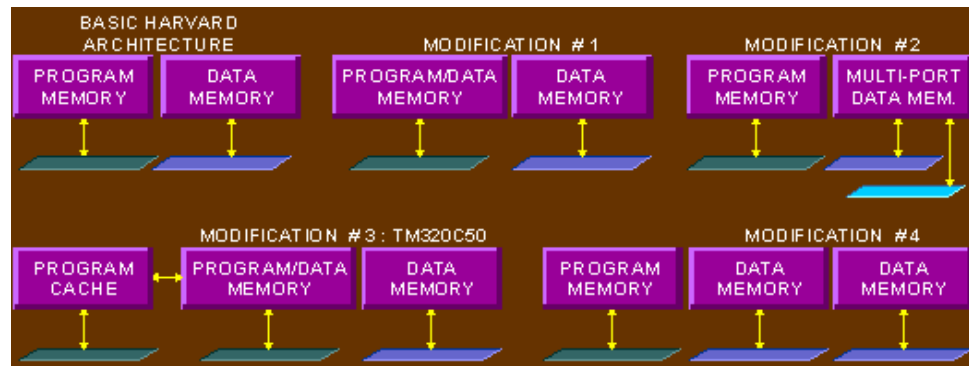
# Memory Space



Each of the parallel busses of a 16-bit fixed-point DSP can allocate  $2^{16}$  addresses to on-chip memory and peripherals.

If each address of a 16-bit data bus was allocated to on-chip 16-bit/word memory, how many bits of storage could be used by the DSP?

- 65536 bits
- less than 1 million bits
- more than 1 billion bits
- more than 1 million bits



Most DSPs today use a **modified Harvard architecture** to increase their **memory bandwidth**. The specific modifications present in the TMS320C5x that have been added to the traditional Harvard structure are:

- a **program/data memory**, a memory that can be addressed by both the DB and the PB;

# Memory Space

- an **instruction cache** that supplements program/data memory.

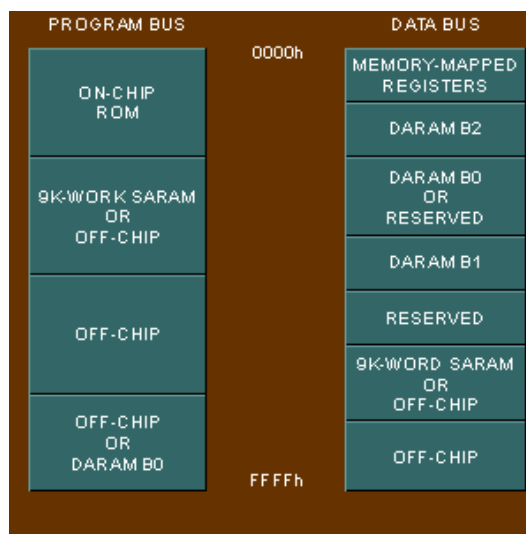
## HARVARD ARCHITECTURE MODIFICATION 1

In the case of the TMS320C50 DSP, certain SARAM blocks can be configured as program/data memory. This implies that each memory element within the SARAM block has been allocated a data bus address and a program bus address.

Program memory is addressed by the program bus. Operands can only be stored in or read from program memory using the program bus.

Data memory is addressed by the data bus. Operands can only be stored in or read from data memory using the data bus.

Program/data memory is addressed by both the program bus and the data bus. Operands can be stored in or read from program/data memory by either using the program bus or the data bus.



The C50 has four **memory configuration bits** that select how data and program bus addresses are allocated among the different on-chip memories, I/O ports, internal memory-mapped registers and external memory-mapped peripherals.

The memory configuration bits are:

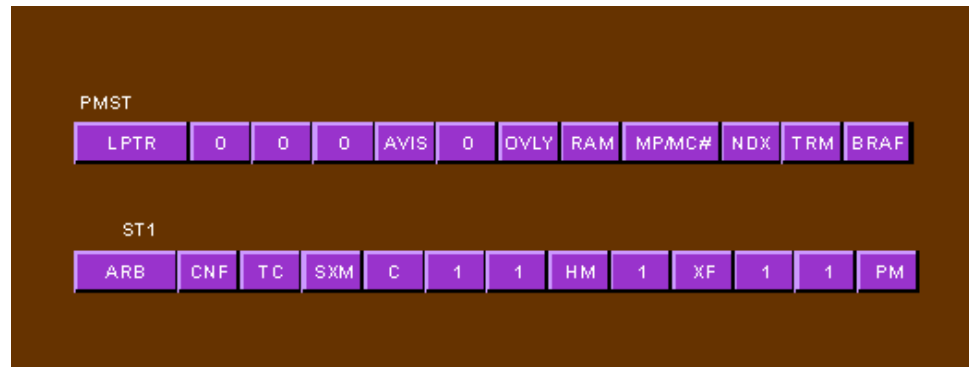
CNF: Enables on-chip DARAM B0 to be addressed by the PB or the DB.

RAM: Enables/disables SARAM from being addressed by the PB.

OVLY: Enables/disables SARAM from being addressed by the DB.

MP/MC#: Enables/disables on-chip ROM from being addressed by the PB.

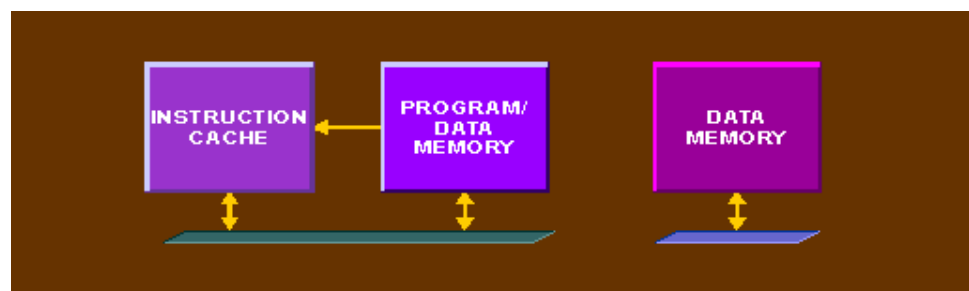
# Memory Space



The memory configuration bits should be initialized (set or cleared) at the beginning of a DSP program and then they should no longer be changed.

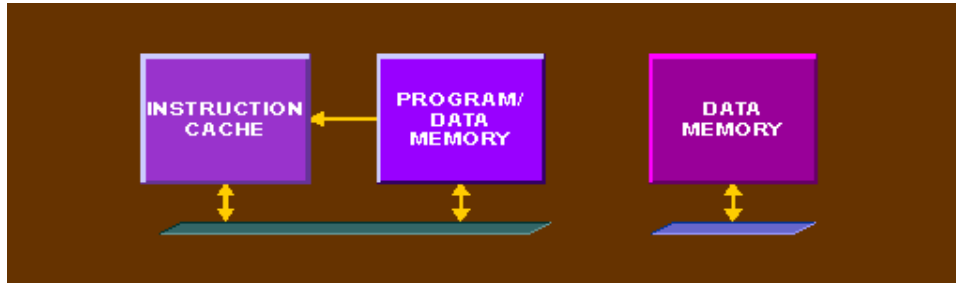
By altering the value of one of the bits, memory elements either become mapped to other addresses (sometimes addresses on a different bus) or become no longer address mapped at all.

## HARVARD ARCHITECTURE MODIFICATION 2



In the case of the 'C50, the register named the Program Counter (PC) acts as an instruction cache. It can store one instruction word (16 bits in width). The instruction once loaded into the PC can be repeated the number of times is specified by the RePeaT Counter register (RPTC).

# Memory Space



During a repeat loop, the program bus does not have to be used to read the next instruction in the program. The DSP simply fetches the next instruction from the instruction cache.

When the instruction from the cache is being executed, a Program Bus (PB) access is freed. The PB is no longer required to fetch the next program instruction.

The freed memory access can be used to read another operand from program/data memory. Specialized instructions like the MAC (Multiply and ACcumulate) when repeated, use the freed Program Bus access to fetch a total of two operands during a single clock cycle.

When programming a DSP the only memory space initializations that should be of a concern to the programmer are:

- The CNF, MP/MC#, RAM and OVLY bit initializations. These select the proper memory configuration.
- The use of the DSK directives describing the memory locations where program and data are stored: `.entry`, `.ps`, `.text`, `.word`, `.byte`, `.data` or `.ds`, `.set`.

## PROCEDURE

**IMPORTANT:** At DSP power up, the memory configuration bits for the TMS320C50 DSP are set to default values. The default values for some of the configuration bits are:

```
MP/MC# = 0
OVLY   = 1
RAM    = 1
```

The RAM bit may not be modified. The program code for the C5x VDE application executes from internal program memory, the C5x VDE application would not function if this bit were changed.

# Memory Space

## Address Allocation of the Data and Program Buses

In this procedure section, you will familiarize yourself with the possible memory configurations of the TMS320C50 DSP.

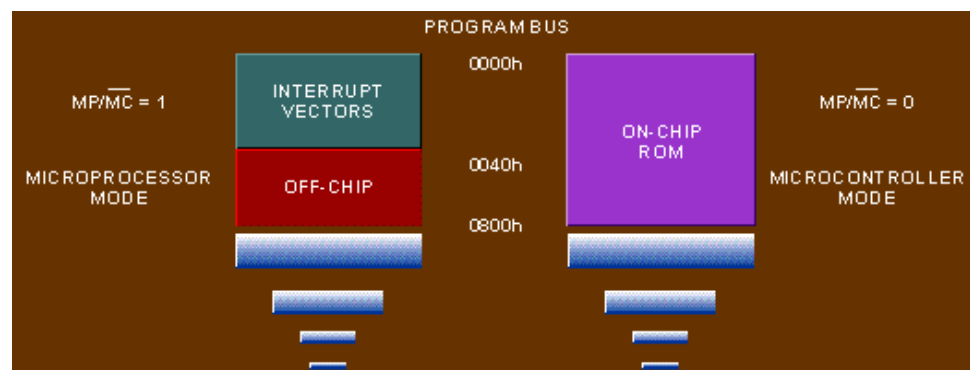
**Note:** Before using the C5x VDE please make certain the circuit board power source is turned ON, and that the serial connection is present between the host computer and the DIGITAL SIGNAL PROCESSOR circuit block labeled SERIAL PORT.

- 1. Open the C5x VDE.
- 2. Open a memory display window with the following options:

Address: 0x0000  
Type: Program Memory  
Display Format: Hex

- 3. Note in the C5x Registers window that the MP/MC# bit is cleared. This is the default value at DSP power up. The DSP is now operating in microcomputer mode and program memory addresses 0h to 800h are allocated to on-chip ROM.

The contents of the program memory addresses correspond to the microcode instructions for the **kernel** used to establish communication between the C5x VDE and the DSP.



- 4. Set the MP/MC# bit (i.e., make MP/MC# = 1). Highlight the memory display window and refresh it (toolbar/Window/Refresh).

# Memory Space

- 5. What value have the addresses between 0h and 800h been initialized to after editing the MP/MC# bit?
  - a. 0x0000
  - b. 980h
  - c. B882h
  - d. 12103d

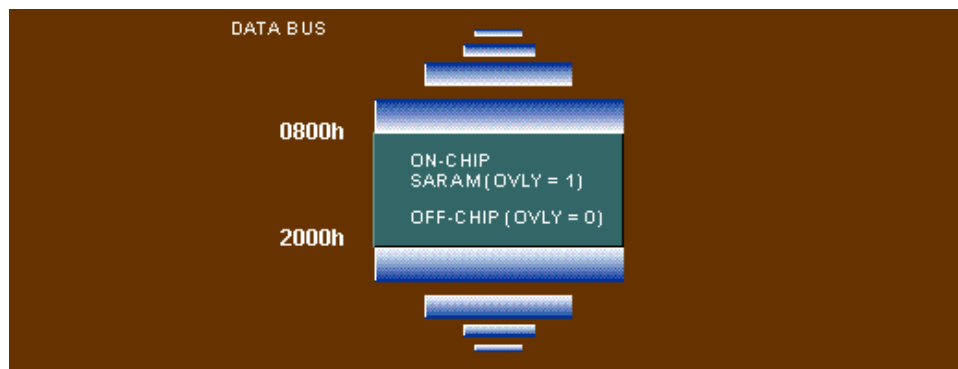
- 6. Open another memory display window with the following options selected:
  - Address: 800h
  - Type: Data Memory
  - Display Format: Hex

- 7. Change the address for the first Program Memory window to 800h, using the window Options menu.

Note that the contents of program and data memory for addresses 0800h to 2C00h are the same.

- 8. Edit a data memory address (between 800h and 2C00h).

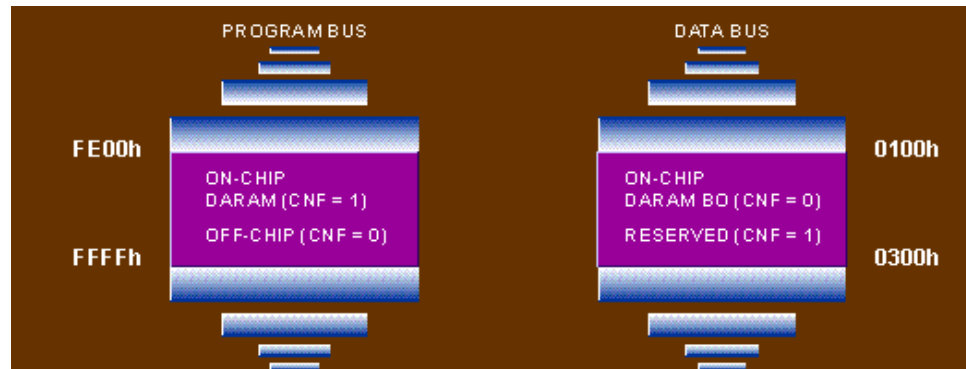
When DSP memory is dually addressed, programmers must be vigilant not to overwrite microcode instructions when writing values through the data bus.



- 9. Clear the OVLY bit and note that data addresses 0800h-2C00h become zero. They are now allocated for off-chip access.

# Memory Space

- 10. Change the address within the Window Options for Program Memory to FE00h. Change the address within the Window Options for Data Memory to 0100h. Clear the CNF bit (make CNF=0).



- 11. Observe that the content of data memory addresses 0100h to 0300h are not equal to zero and that the contents of program memory addresses FE00h to FFFF are equal to zero.
- 12. Set the CNF bit (make CNF = 1) and note the changes that take place in the memory displays.

**See HELP Unit 02 shelp14**

- 13. What occurred after editing the CNF memory configuration bit?
  - The values in data memory were copied to program memory.
  - Memory allocated to the data bus was wiped clear of all information.
  - DARAM B0 that was address mapped by the data bus became address mapped by the program bus.
  - Nothing occurred after editing the CNF memory configuration bit.

## The Recorder

In this procedure section, you will use a Playback/Recorder program to familiarize yourself with the memory architecture of the TMS320C50.

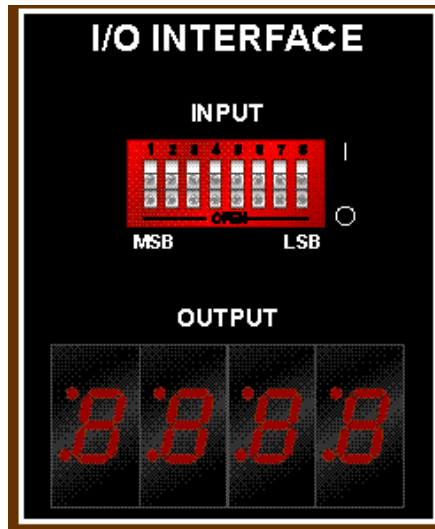
- 14. Open the ex2\_2.asm assembler source file within an ASCII text editor.

This file is the assembler source code for a DSP program that makes the DIGITAL SIGNAL PROCESSOR circuit board become a Playback/Recorder. Refer to this source file at anytime during the procedure.

# Memory Space

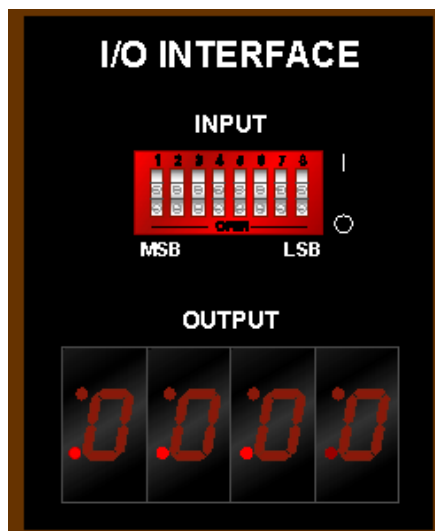
- 15. Carefully read the description of the ex2\_2.asm source file.

Important points:



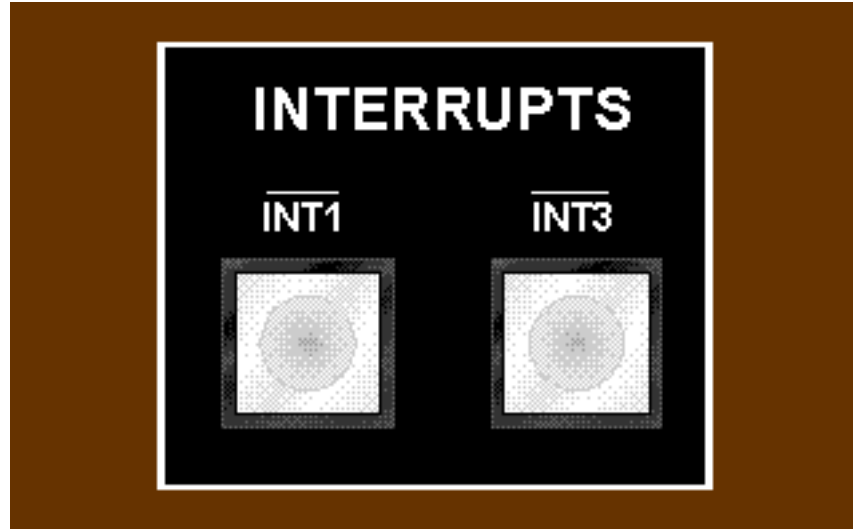
The DIP switch is used to choose a recording mode.

(The modes are differentiated by their data compression methods.)

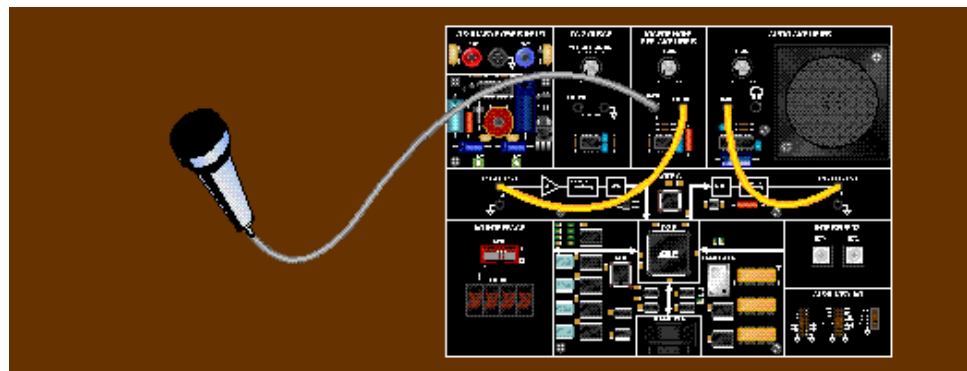


The signal input level of the microphone (proportional to the number of dots) is output to the I/O INTERFACE display.

# Memory Space



Pressing INT1# begins recording and pressing INT3# begins playback.



- 16. Make the following connections on the DIGITAL SIGNAL PROCESSOR circuit board:
  - Connect a microphone to the INPUT of the MICROPHONE PRE-AMP.
  - Connect the OUTPUT of the MIC. PRE-AMP. to the ANALOG INPUT of the CODEC.
  - Connect the ANALOG OUTPUT of the CODEC to the INPUT of the AUDIO AMPLIFIER.
  
- 17. Position to zero the DIP switch on the I/O INTERFACE of the circuit board.

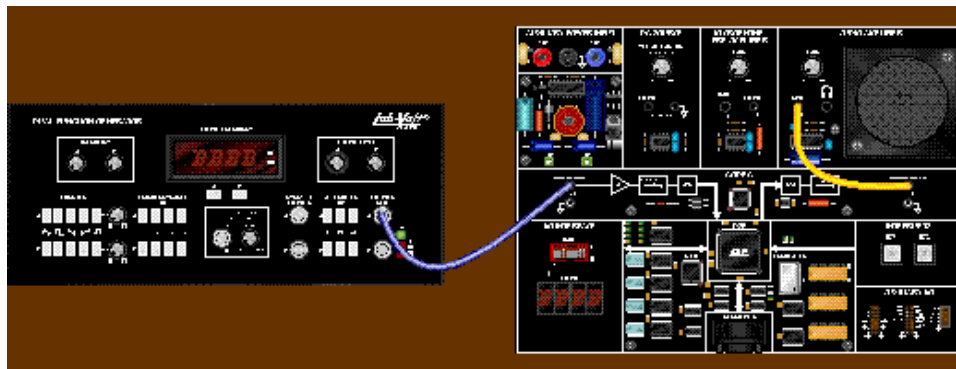
# Memory Space

- 18. Using the C5x VDE, load the ex2\_2.dsk program into the DSP. Press the Run command.
- 19. With the INT1# push button begin recording your voice. Play it back with the INT3# push button when done recording.

Adjust the level of the potentiometers in the MICROPHONE PRE-AMPLIFIER and the AUDIO AMPLIFIER circuit blocks to make the audio level during playback (INT3#) to be sufficient.

- 20. Repeat step 20 with the other two modes of recording.

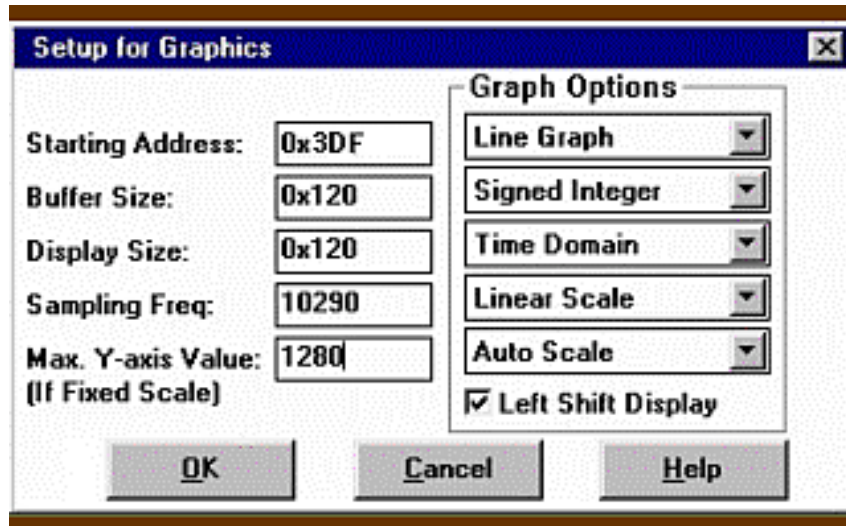
DIP SWITCH VALUE	AUDIO RECORDING MODE
0	16-bit sampling
1	8-bit $\mu$ -law compression
2	16-bit samples truncated to 8-bits



- 21. Make the following connections on the DIGITAL SIGNAL PROCESSOR circuit board:
  - Disconnect the ANALOG INPUT of the CODEC circuit block from the OUTPUT of the MICROPHONE PRE-AMPLIFIER.
  - Connect the ANALOG INPUT to the OUTPUT of a function generator.
- 22. Adjust the function generator to output a  $\sim 300$  Hz sinusoidal signal, at  $\sim 1.00$  Vpp.

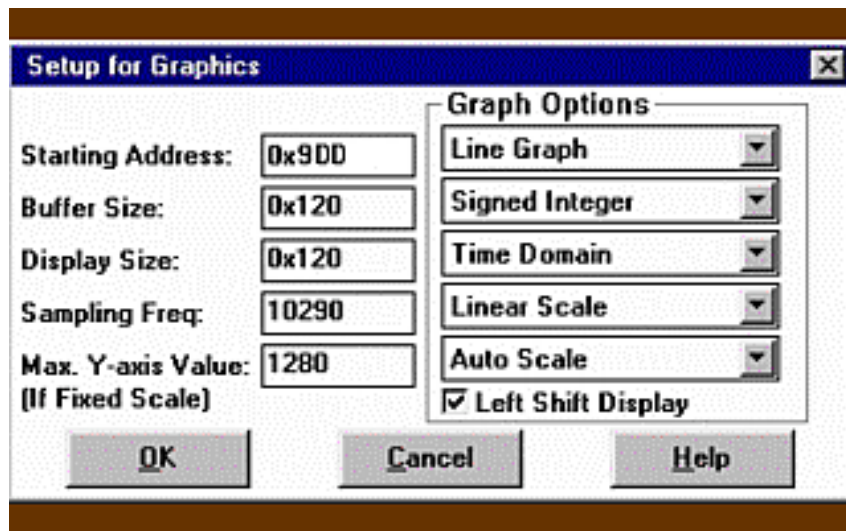
# Memory Space

- 23. Set the position of the DIP switch to zero and record the generated signal.



- 24. Using the C5x VDE, Halt the DSP program. Open a Graphic Display with the settings shown in the figure.

To maximize recording time, the recorded signal samples are stored in two parts. One part is stored in DARAM B1.



- 25. Using the C5x VDE, open a second Graphic Display with the settings shown in the figure.

The second part of the recorded signal samples are stored in SARAM.

# Memory Space

Recorded Signal Sample Ranges

DARAM B1	03DF h - 04FF h
SARAM	09DD h - 2BFF h

If the Playback/Recorder program stores one recorded signal sample per data memory address then how many samples can be stored if the recorded signal samples are stored in the address ranges shown above?

975 = \_\_\_\_\_ samples

- 26. Close the text editor displaying the ex2\_2.asm source file. End the C5x VDE session.

## The Instruction Cache

In this procedure section, you will observe the difference in the execution time of a DSP algorithm that uses the instruction cache and one that does not.

- 27. Open a new C5x VDE session.
- 28. Load the ex2\_2b.dsk program into the DSP.
- 29. Open three Data Memory displays, at the following addresses:
  - Address: TIMER1

Within this data memory window there are three constants.

  - Address: SARAM2

The constants named SARAM1 and SARAM2 are located in different SARAM memory blocks.

  - Address: DARAM
- 30. Set a breakpoint at the program memory address labeled SLOW (by double-clicking on the Dis-Assembly window instruction line).
- 31. Execute the code, corresponding to the DSP initialization sequence, located between the execution line in the Dis-Assembly window and the

# Memory Space

breakpoint labeled SLOW. Do this by executing the RUN command within the C5x VDE.

- 32. Within the Dis-Assembly window of the C5x VDE, set a breakpoint at the program memory address labeled FAST.
  
- 33. Execute the SLOW algorithm by executing the C5x VDE RUN command once.

When the RUN command was pressed the SLOW algorithm was executed. The SLOW algorithm is all of the code located between the instruction lines labeled SLOW and FAST.

The SLOW algorithm executed ten consecutive MAC instructions. The MAC instructions are read consecutively from memory, they do not use the instruction cache to repeat the instruction.

What is the value of the following operation (TIMER1 - TIMER2) converted to decimal?

1014 \_\_\_\_\_ clock cycles

What is a better definition of the meaning of the value (TIMER1-TIMER2)?

- a. The value of the ACCumulator register after the execution of 10 multiply and accumulate (MAC) instruction.
  - b. The number of memory accesses made during the execution of the SLOW algorithm.
  - c. The speed of the DSP during the calculation of the SLOW algorithm.
  - d. The relative number of instruction cycles taken to execute the code located between the SMMR instructions.
- 
- 34. Execute the FAST algorithm by pressing the RUN command again.

When the RUN command was pressed the FAST algorithm was executed. The FAST algorithm is all of the code located between the instruction line labeled FAST and the instruction:

B MAIN,\*

When the FAST algorithm executed, ten consecutive MAC instruction were executed within a repeat loop. The program bus was freed because each MAC instruction (except the first) was fetched directly from the program cache.

What relative number of instruction cycles was taken to execute the fast algorithm (TIMER1-TIMER2)?

1027 \_\_\_\_\_ clock cycles

# Memory Space

- 35. Compare the code of each algorithm and the amount of time it took to execute each.

Which of the following statements relating to the instruction cache is true?

- a. It is used to differentiate between memory that stores data words and memory that stores program words.
- b. It is used to free up the data bus of an access.
- c. It is a small memory within the processor core that is used for storing program instructions.
- d. It is present to increase the calculation speed of the DSP.

## **Measuring the Relative Memory Access Rates Provided by SARAM and DARAM.**

In this procedure section, you will observe the difference between SARAM and DARAM access rates.

- 36. Open the ex2\_2b.asm file inside of an ASCII text editor (such as Notepad).
  
- 37. Familiarize yourself with the source code.

The SLOW and the FAST algorithms are clearly identified. At the top of the source file are the data constants that are or that can be used by the program. The source code belonging to the initialization sequence is clearly identified.

- 38. Edit the fast algorithm, within the ex2\_2b.asm source code file, so that the multiply and accumulate instruction(MAC) calls two constants located in different SARAM blocks as so:

```
MAC SARAM2,SARAM1
```

- 39. Save the modified source file to your personal student folder as:

```
ex2_2bv2.asm
```

- 40. Assemble the file from within your student folder. Execute from within your student folder the following command at a DOS prompt:

```
c:\lv91027\bin\dsk5a.exe ex2_2bv2.asm -l
```

- 41. Answer the following question. The answer can be found by loading the ex2\_2bv2.dsk file into the DSP and executing the necessary part of code.

# Memory Space

Block the code off with breakpoints and press the RUN command inside of the C5x VDE.

What are the number of cycles (TIMER1-TIMER2) taken to execute the FAST algorithm when the operands called by the MAC instruction are not located in the same SARAM memory block?

1055 \_\_\_\_\_ clock cycles

42. Close the text editor open with the ex2\_2bv2.asm source file. End the C5x VDE session.

## CONCLUSION

- SARAM and DARAM are divided up into different sized memory blocks.
- An SARAM memory block can be written to or read from once within one instruction cycle.
- A DARAM memory block can be read from and written to inside of the same instruction cycle.
- A program cache, when used, frees the PB of one instruction read which can then be used in data/program memory as a data read.
- Memory configuration bits (such as the four found in the 'C50) are used to control the configuration of the PB and DB memory map.

## REVIEW QUESTIONS

1. Which of the following modifications to the basic Harvard architecture are used in some DSPs to increase their number of available memory accesses?
  - a. An instruction cache and a data/program (dual) addressed memory.
  - b. An instruction cache and a parallel bus structure.
  - c. A parallel bus structure and a data/program (dual) addressed memory.
  - d. none of the above.
2. How many addresses can each of the parallel buses of a 16-bit Harvard architecture DSP allocate to on-chip memory and to peripherals?
  - a.  $2^{15}$  addresses
  - b.  $2^{16}$  addresses
  - c. 32768 addresses
  - d. 32767 addresses

# Memory Space

3. Instruction words cannot be read from which of the following types of memory?
  - a. data memory
  - b. data/program memory
  - c. program cache
  - d. program memory
  
4. Why can a dually addressed SARAM memory block be of importance to a DSP?
  - a. Two values (a data operand and a microcode instruction) can be stored per memory element.
  - b. The memory storage capacity of the SARAM memory block is doubled.
  - c. An additional SARAM memory access is gained if the program bus is not required to fetch a microcode instruction.
  - d. None of the above.
  
5. What is the instruction cache used to store?
  - a. Program/data memory contents.
  - b. Instructions
  - c. Program addresses
  - d. Operands

Other Samples Extracted  
from  
Digital Signal Processor



# Unit Test

1. Why is it important to incorporate an Address Generation Unit within a DSP?
  - a. It allows AGU arithmetic to be independent of the CALU.
  - b. It allows the DSP to process digital signals.
  - c. It allows all address calculations to proceed in series with instruction execution.
  - d. Data buffers can be managed.
  
2. What is the basic processor memory architecture most often implemented in digital signal processors?
  - a. The Von Neumann architecture.
  - b. The Harvard architecture.
  - c. The modified Von Neumann architecture.
  - d. The DSP architecture.
  
3. Which of the following elements is not part of a typical programmable DSP?
  - a. CODEC
  - b. Central Processing Unit (CPU)
  - c. Bus structure
  - d. Peripherals
  
4. In which of the following situations does an instruction require the use of an addressing mode?
  - a. When the instruction needs to read an operand from memory.
  - b. When the instruction needs to write an operand to memory.
  - c. When the instruction needs to fetch the value held within the accumulator.
  - d. All of the above.
  
5. Which among the following choices does not differentiate a DSP from a general-purpose processor?
  - a. addressing modes
  - b. memory architecture(bus structure)
  - c. execution time of the CALU
  - d. processor native word width
  
6. Which of the following statements is true?
  - a. Most fixed-point DSP processor multipliers produce a result that is the width of the input operands.
  - b. CALU refers to the entire arithmetic processing path excluding the multiplier.
  - c. The CALU uses post-scalers for address calculations.
  - d. ALU refers to the combination adder/subtractor/logical function unit.

## Unit Test (cont'd)

7. Which of the following DSP characteristics permit the Multiplier and Arithmetic Logic Unit to keep a constant arithmetic precision during computation?
  - a. Presence of the accumulator Guard Bits.
  - b. Option of enabling sign-extension mode.
  - c. Option of enabling overflow saturation mode.
  - d. The Multiplier and ALU operate with twice the internal bus bit-width.
  
8. Why do most DSPs use modified Harvard memory architectures (as opposed to Harvard memory architectures)?
  - a. To differentiate operands from instructions.
  - b. To increase their memory bandwidth.
  - c. To increase their calculation rate.
  - d. To increase the number of specialized addressing modes.
  
9. Which of the following statements is true? The instruction cache when used:
  - a. frees a data memory access.
  - b. enables/disables on-chip program/data memory.
  - c. frees a program memory access.
  - d. None of the above.
  
10. Which of the following choices is true? A dually addressed program/data memory block can:
  - a. hold operands only and is accessed by the PB.
  - b. hold instructions and operands accessed by the PB and DB.
  - c. hold instructions only and is accessed by the DB.
  - d. hold operands only and is addressed by the instruction cache alone.

Instructor's Guide Sample  
Extract from  
Digital Signal Processor



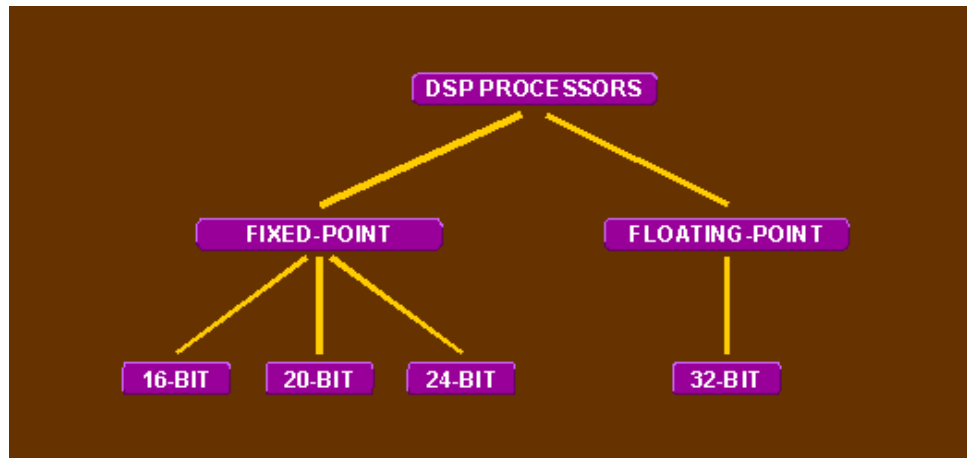
# Exercise 1-3

## Processor Arithmetic

### EXERCISE OBJECTIVES

Upon completion of this exercise, you will be familiar with the numerical formats and representations used within DSPs.

### DISCUSSION

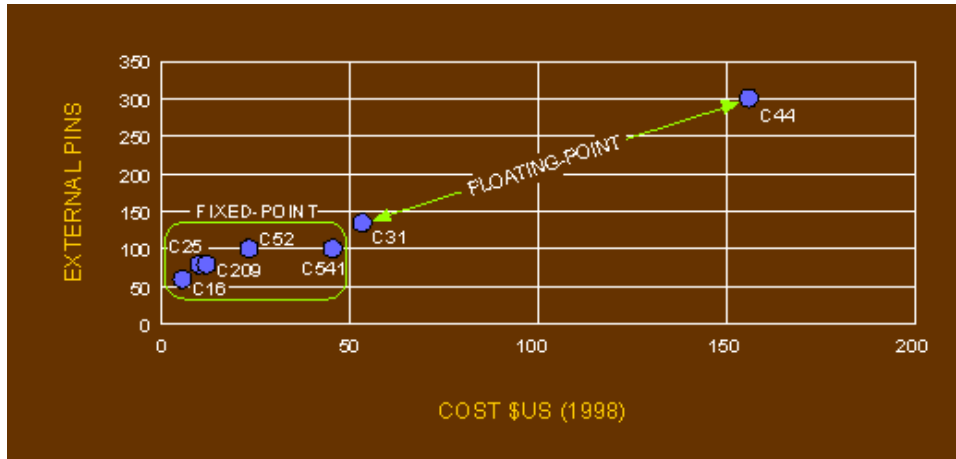


Digital Signal Processors are categorized by the way that their arithmetic is performed. A DSP can either be:

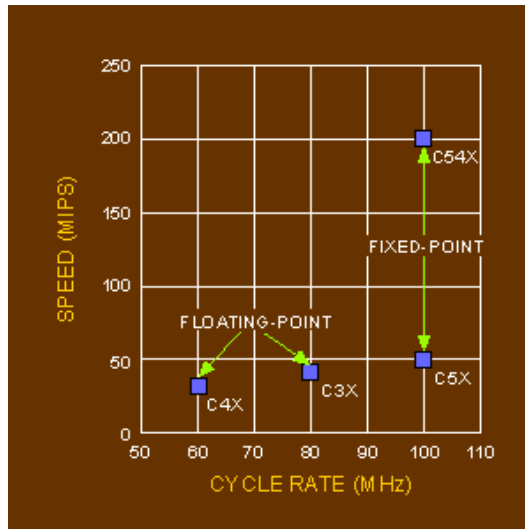
a **fixed-point** DSP, or,  
a **floating-point** DSP

The type of DSP chosen for a specific application depends on the suitability of its arithmetic for the task. The TMS320C50 is a fixed-point DSP.

# Processor Arithmetic



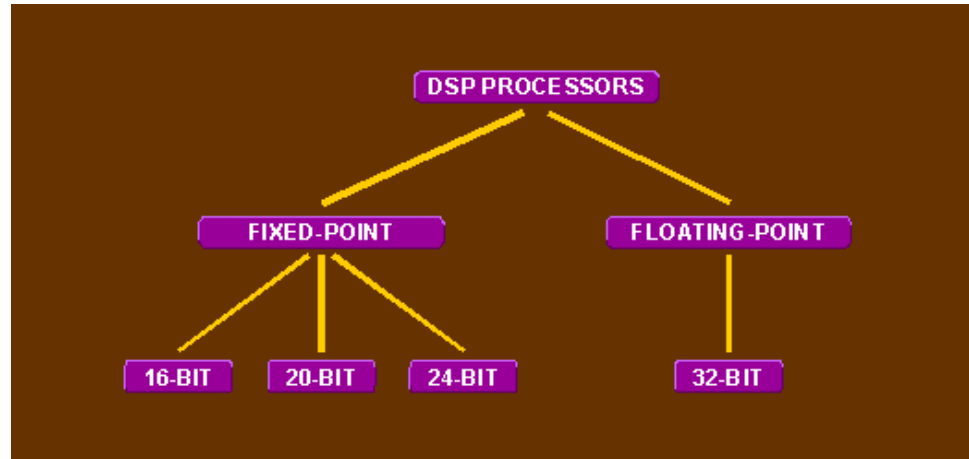
Fixed-point DSPs are usually cheaper than their floating-point counterparts because they contain less silicon and have less external pins.



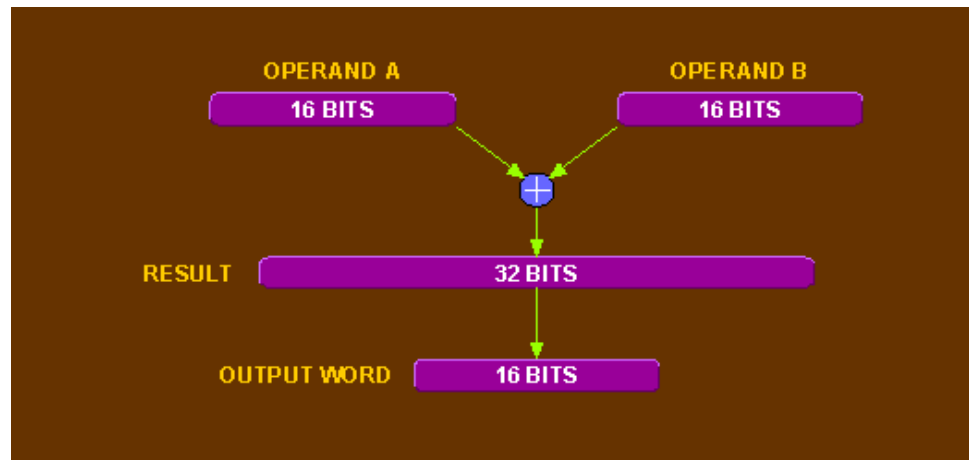
Fixed-point devices generally have faster clock cycle rates.

In 1998, these clock cycles were as small as 10 ns, corresponding to a processor cycle rate of 100 MHz.

# Processor Arithmetic



Floating-point devices are usually more flexible because their arithmetic system has access to a wider **dynamic range** and in many cases these systems are more precise.

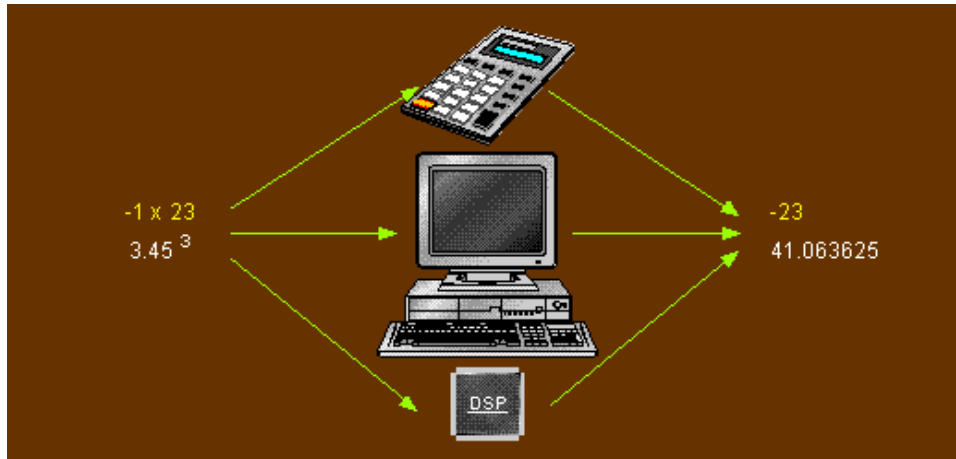


A typical 16-bit fixed-point processor stores coefficients and data values with 16-bit precision.

However, within the **internal arithmetic unit** of the DSP, intermediate values are kept at 32 bits of precision.

By so doing, the cumulative rounding error made during calculations is minimized.

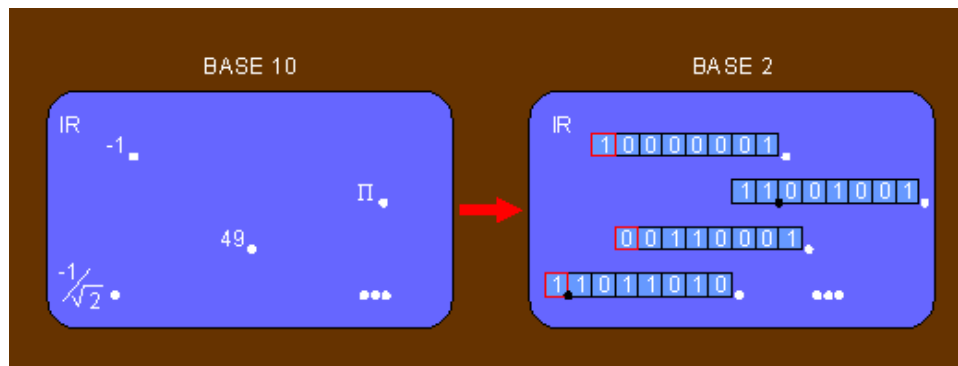
# Processor Arithmetic



When you use your computer or your calculator you can calculate such values as:

$$(-1 \times 23) \text{ or } (3.45^3)$$

A DSP can also provide answers to the same types of questions.

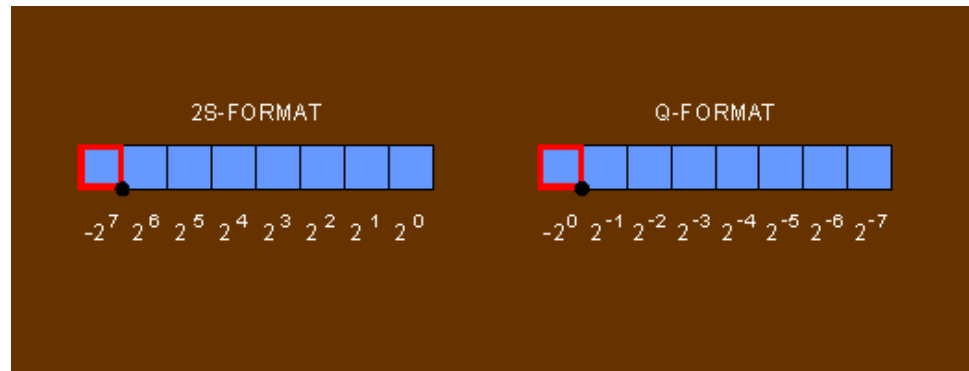


A programmer must use certain **numerical formats** so that every value desired to be used in the DSP has a binary representation associated with it.

This binary value will need at times to represent either a positive or negative, fractional or integer number.

Since a DSP is a processor that specializes in doing rapid calculations, it is essential to understand how the diverse range of numeric values can be expressed.

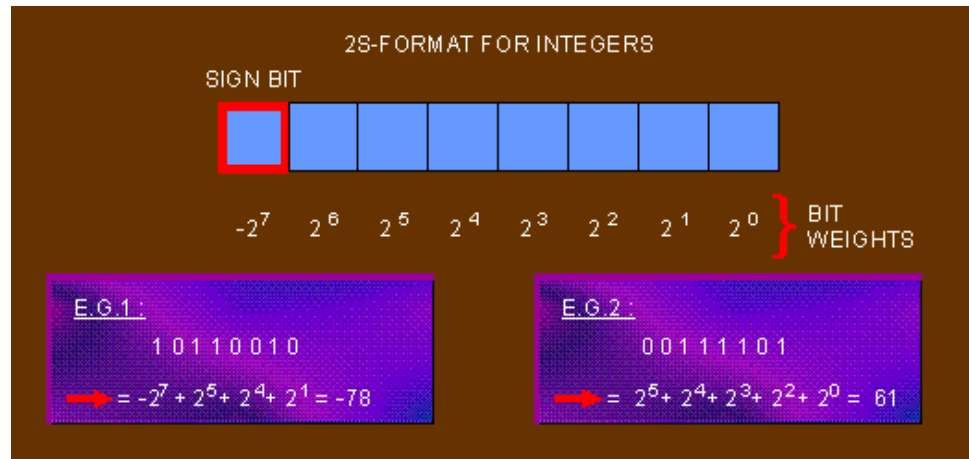
# Processor Arithmetic



Integers, both negative and positive, are represented by the **two's complement** integer format (2s-format).

Fractional numbers, both negative and positive, are represented by the two's complement fractional format (Q-format).

These formats differ only by the associated **weights** that are given to each bit of information.



In two's complement integer notation (2s-format) a negative sign is associated with the most significant bit.

The 2s-format provides a numeric range covering:

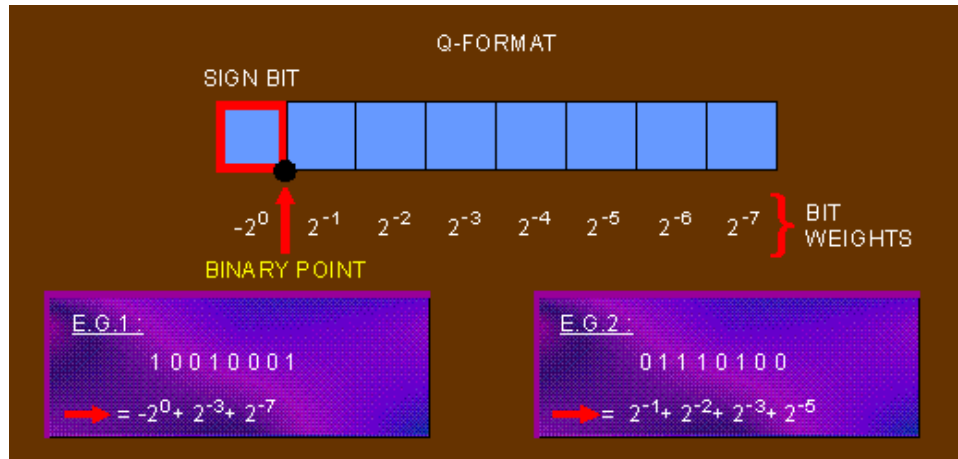
$$-2^{N-1} \text{ to } +(2^{N-1} - 1)$$

where N represents the number of bits in the binary number.

# Processor Arithmetic

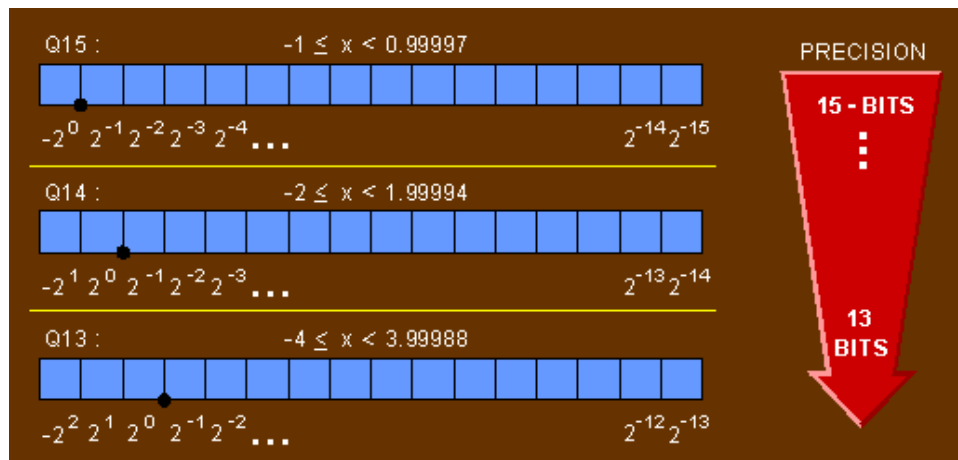
Represent (-6) in a 4-bit 2s-format binary number.

$$-6 = 1010 \pm 0 \%$$



The two's complement fractional format (or Q-format) associates different weights with each bit as well.

The existence of the **binary point** separating the fractional weighted values from the integral weighted values is implied.

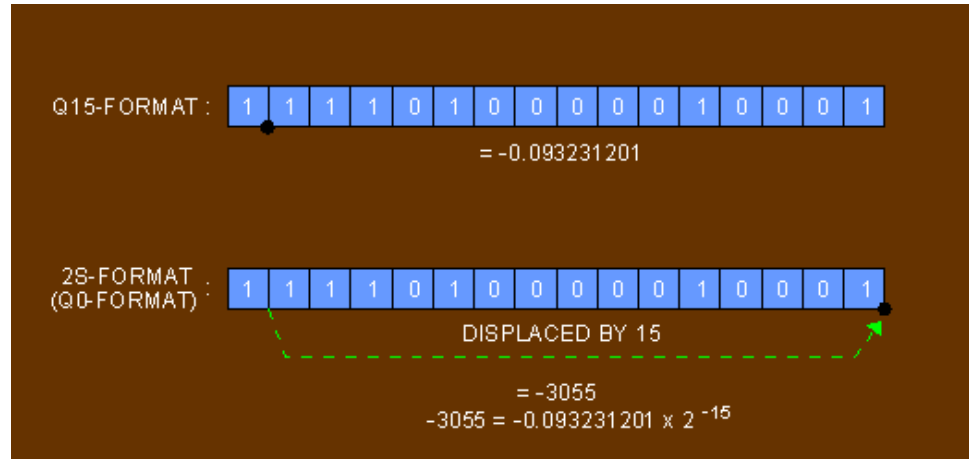


In Q15-format the most significant bit is the sign bit and it is given a weight of  $-2^0$ .

This implies that the binary point is located between the MSB and the 14th bit.

By changing the position of the binary point the weight given to each bit is also changed. Consequently, the dynamic range and the precision of the two's complement fractional format may vary with the type of format being used.

# Processor Arithmetic



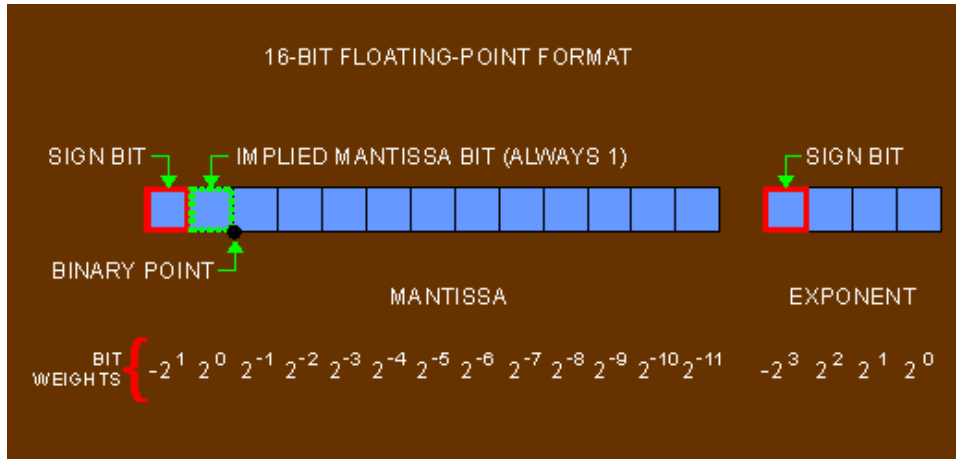
Note that by continuing to move the binary point further and further to the right a handy relationship is uncovered. The 2s-format and the Q15-format decimal representations are proportional by a scaling factor of  $2^{15}$ .

Which of the following choices represents the proportionality constant between the 2s-format and Q13-format, for the 16-bit binary number?

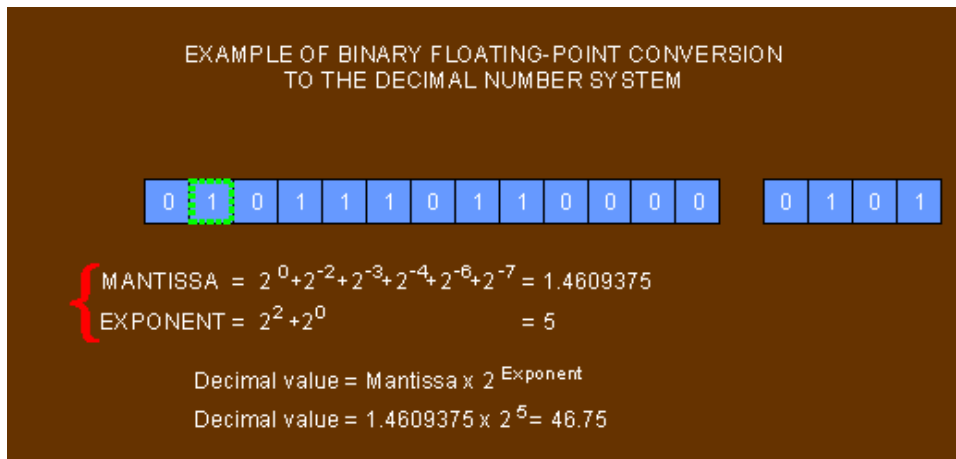
- a. 13
- b.  $2^{15}$
- c.  $-3.0518 \times 10^{-5}$
- d. **8192**

The 2s- and Q-formats can be used by the fixed-point internal arithmetic units of any DSP. These formats are numerical conventions used by programmers. The binary arithmetic done inside of a fixed-point DSP is not affected by the format of the binary number used.

# Processor Arithmetic



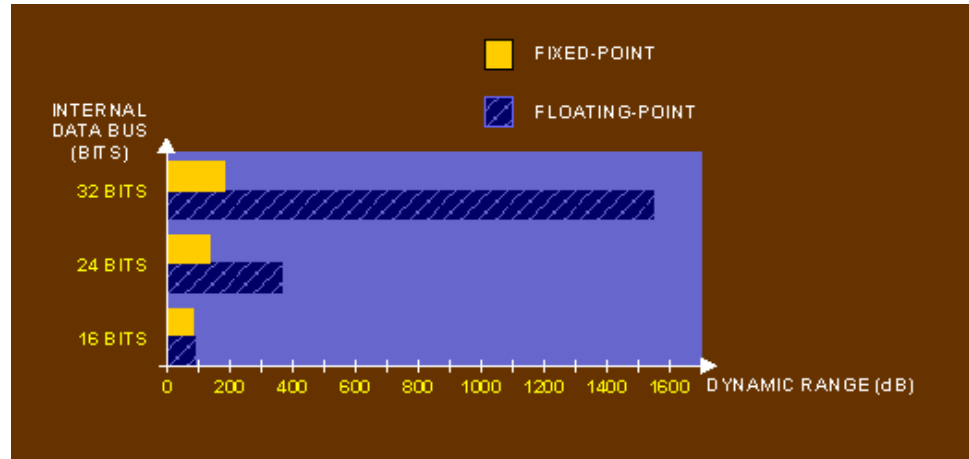
Floating-point DSPs generally use a 32-bit format where the 24 left-most bits represent the mantissa and the 8 remaining bits represent the exponent.



So that a continuous range of values is covered by a 32-bit floating-point number, the mantissa must vary over -1 to 0 and +1 to +2.

This means that the bit weighted by  $2^0$  will always be equal to 1. Therefore, it becomes unnecessary to store it in memory and during calculations it becomes an implied bit.

# Processor Arithmetic



Floating-point processors are usually more precise and have a larger dynamic range.

While in theory the choice between fixed- and floating-point arithmetic is independent of the choice of precision, in practice floating-point processors usually provide higher precision.

This arises because more bits are provided to define the mantissa (24 bits + 1 implied bit) compared to fixed-point DSPs that usually have 16 bits, although 20- and 24-bit fixed-point DSPs exist.

## PROCEDURE

### Converting a Signed Fractional Number to Q14-Format

In this procedure section, you will learn how to convert a signed fractional number to a binary number written in Q-format.

- 1. Follow steps 2 to 5 to convert the following decimal value:

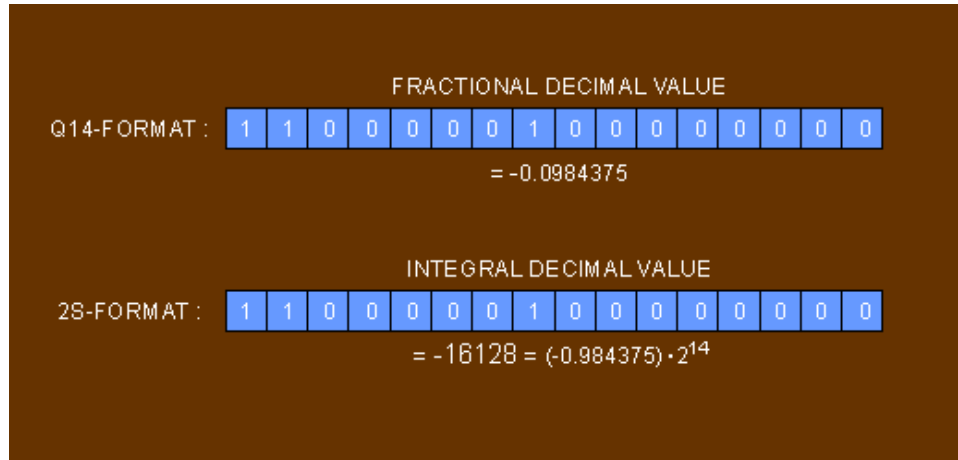
-0.984375

to binary Q14-format.

- 2. Open the Microsoft® Calculator present in your version of Windows.
- 3. Make certain that the Scientific option under the View menu is checked. Checking this option makes the Standard calculator become a Scientific calculator.

# Processor Arithmetic

- 4. Multiply (-0.984375) by  $2^{14}$ .



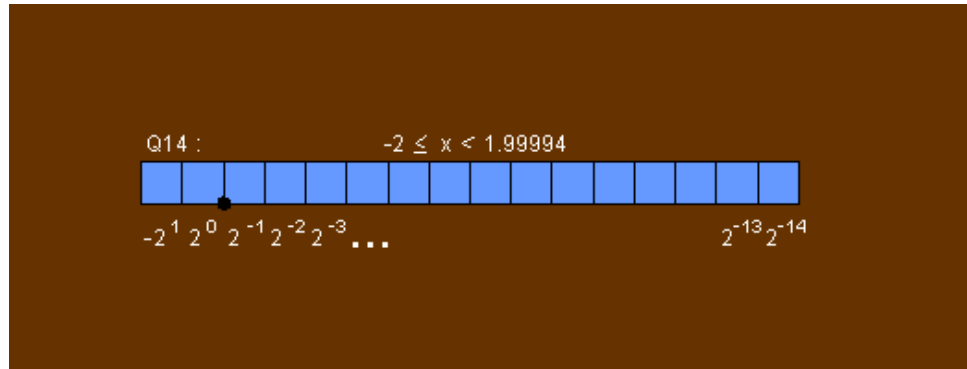
This scales the fractional decimal value to an integral decimal value allowing easy conversion to a binary number.

- 5. Use the Calculator conversion functions (Dec to Bin) to change the value obtained in step 4 to a one Word (16 bits) binary value (make certain the Word check box has been clicked).

What Q14-format binary value representing -0.984375d did you obtain?

- a. **1100 0001 0000 0000**
- b. 1001 0001 0010 0001
- c. 0111 0001 0000 0000

# Processor Arithmetic



6. Verify that the binary number that you calculated (1100 0001 0000 0000) is equal to the decimal value -0.984375 it was converted from. Use the Q14-format bit-weights to calculate the decimal value.

## Converting a Binary Number to a Decimal Value

In this procedure section, you will learn how to convert a binary number to a decimal value.

7. Follow steps 7 through 13 to convert the following number:

$$B093h = 1011\ 0000\ 1001\ 0011b$$

to a decimal value when:

1. The hexadecimal number represents an unsigned integer.
  2. The hexadecimal number is written with the 2s-numerical format.
  3. The hexadecimal number is written in Q15-numerical format.
8. What is the weight given to each bit of a binary word representing unsigned integers?
- a.  $2^0 \ 2^{-1} \ 2^{-2} \ 2^{-3} \ 2^{-4} \ 2^{-5} \ 2^{-6} \ 2^{-7} \ 2^{-8} \ 2^{-9} \ 2^{-10} \ 2^{-11} \ 2^{-12} \ 2^{-13} \ 2^{-14} \ 2^{-15}$
  - b.  $2^{15} \ 2^{14} \ 2^{13} \ 2^{12} \ 2^{11} \ 2^{10} \ 2^9 \ 2^8 \ 2^7 \ 2^6 \ 2^5 \ 2^4 \ 2^3 \ 2^2 \ 2^1 \ 2^0$
  - c.  $-2^{15} \ 2^{14} \ 2^{13} \ 2^{12} \ 2^{11} \ 2^{10} \ 2^9 \ 2^8 \ 2^7 \ 2^6 \ 2^5 \ 2^4 \ 2^3 \ 2^2 \ 2^1 \ 2^0$

# Processor Arithmetic

9. Consider that B093h was written with the above numerical format (unsigned integer). Use the unsigned integer format bit-weights to calculate the corresponding decimal value. What is the value of the calculated decimal number?

$$= 45203 \pm 0 \%$$

10. What is the weight given to each bit of a binary word in 2s-format?

a.  $-2^{15} \ 2^{14} \ 2^{13} \ 2^{12} \ 2^{11} \ 2^{10} \ 2^9 \ 2^8 \ 2^7 \ 2^6 \ 2^5 \ 2^4 \ 2^3 \ 2^2 \ 2^1 \ 2^0$   
b.  $-2^0 \ 2^1 \ 2^2 \ 2^3 \ 2^4 \ 2^5 \ 2^6 \ 2^7 \ 2^8 \ 2^9 \ 2^{10} \ 2^{11} \ 2^{12} \ 2^{13} \ 2^{14} \ 2^{15}$   
c.  $-2^2 \ 2^1 \ 2^0 \ 2^{-1} \ 2^{-2} \ 2^{-3} \ 2^{-4} \ 2^{-5} \ 2^{-6} \ 2^{-7} \ 2^{-8} \ 2^{-9} \ 2^{-10} \ 2^{-11} \ 2^{-12} \ 2^{-13}$

11. Consider that B093h was written with the above numerical format (2s-format). Use the 2s-format bit-weights to calculate the corresponding decimal value. What is the value of the calculated decimal number?

$$= -20333 \pm 0 \%$$

12. What is the weight given to each bit of a binary word in Q15-format?

a.  $-2^0 \ 2^{-1} \ 2^{-2} \ 2^{-3} \ 2^{-4} \ 2^{-5} \ 2^{-6} \ 2^{-7} \ 2^{-8} \ 2^{-9} \ 2^{-10} \ 2^{-11} \ 2^{-12} \ 2^{-13} \ 2^{-14} \ 2^{-15}$   
b.  $-2^0 \ 2^1 \ 2^2 \ 2^3 \ 2^4 \ 2^5 \ 2^6 \ 2^7 \ 2^8 \ 2^9 \ 2^{10} \ 2^{11} \ 2^{12} \ 2^{13} \ 2^{14} \ 2^{15}$   
c.  $2^0 \ 2^{-1} \ 2^{-2} \ 2^{-3} \ 2^{-4} \ 2^{-5} \ 2^{-6} \ 2^{-7} \ 2^{-8} \ 2^{-9} \ 2^{-10} \ 2^{-11} \ 2^{-12} \ 2^{-13} \ 2^{-14} \ 2^{-15}$

13. Consider that B093h was written with the above numerical format (Q15-format). Use the Q15-format bit-weights to calculate the corresponding decimal value. What is the value of the calculated decimal number?

$$= -0.62051 \pm 0 \%$$

## Making Numerical Conversions with the C5x VDE

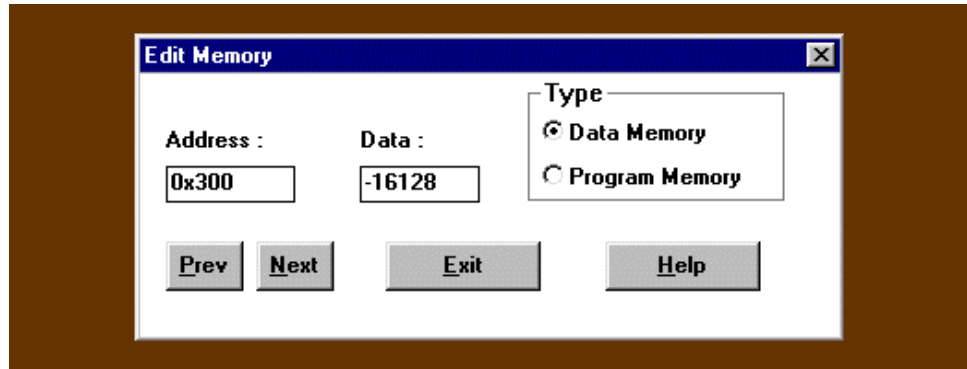
In this procedure section, you will make the same numerical conversions done in the previous section but this time using the C5x VDE.

**Note:** Before using the C5x VDE please make certain the circuit board power source is turned ON, and that the serial connection is present between the host computer and the DIGITAL SIGNAL PROCESSOR circuit block labeled SERIAL PORT.

14. Open the C5x Visual Development Environment (VDE). Close the C5x Registers window.

# Processor Arithmetic

- 15. Open a Memory display window to **dma** 0x300 by executing the Memory command found in the View menu. Use the Signed Integer display format. This window will be used to make numerical conversions.



- 16. Edit data memory address 0x300 to the following value:

address	value
0x300	-16128 (- 0.984375 x 2 <sup>14</sup> )

The value entered into data memory corresponds to the decimal value that you scaled by 2<sup>14</sup> in step 4, and then converted to binary.

- 17. Open the Options window of the Data Memory display, and change the display format to binary.

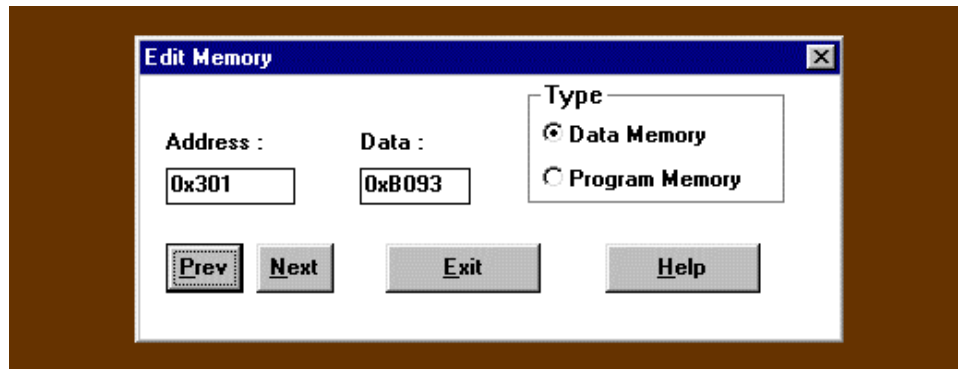
What is the binary value contained in data memory address 0x300?

- a. **1100 0001 0000 0000**
- b. 1011 1001 0000 0001
- c. 0000 0011 0000 0001

In step 5, you had converted -0.984375 to the same binary Q14-format value: 1100 0001 0000 0000

- 18. Open the Options window of the Data Memory display and change the display format to Hex (hexadecimal).

# Processor Arithmetic



- 19. Edit data memory address 0x301 to the following value:

address	value
0x301	0xB093

The hexadecimal value, entered into the data memory address, corresponds to the initial value in step 7 before you converted it into three different decimal values.

- 20. Open the Options window of the Data Memory display, and change its display format to Unsigned Integer.

What is the decimal value contained in data memory address 0x301?

**= 45203 ± 0 %**

In step 9, you had converted B093h to the same unsigned value: 45203

- 21. Open the Options window of the Data Memory display, and change its display format to Signed Integer.

What is the decimal value contained in data memory address 0x301?

**= -20333 ± 0 %**

In step 11, you had converted B093h, written in 2s-format, to the same decimal value: -20333

- 22. Open the Options window of the Data Memory display, and change its display format to Fixed-Point Q15.

# Processor Arithmetic

What is the decimal value contained in data memory address 0x301?

$$= -0.62051 \pm 0 \%$$

In step 13, you had converted B093h, written in Q15-format, to approximately: -0.620513916 d

23. End the C5x VDE session.

## CONCLUSION

- DSPs are categorized by the way that their arithmetic is performed. A DSP can either be a fixed-point DSP, or a floating-point DSP.
- Integers, both negative and positive, are represented by the two's complement integer format (the 2s-format).
- Fractional numbers, both negative and positive, are represented by the two's complement fractional format (the Q-format).
- Numerical formats are a mathematical convention, and they differ only by the weights that are associated with each bit in a binary word.

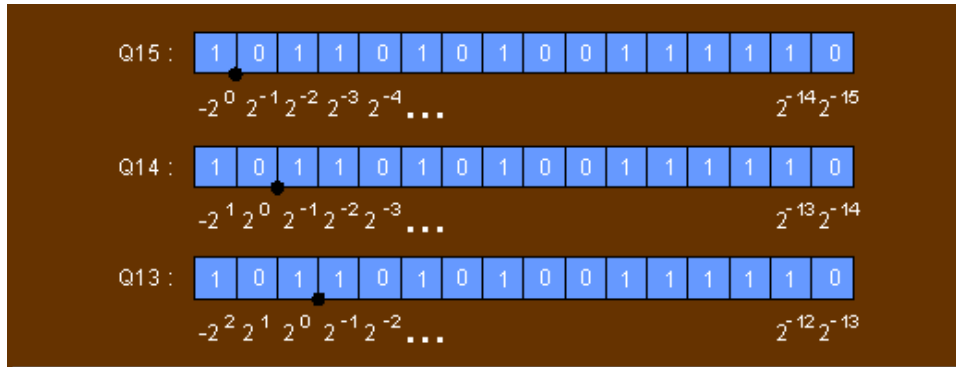
## REVIEW QUESTIONS

1. What is the range of decimal values that a 2s-format 16-bit binary number can represent?
  - a. -65536 to 65535
  - b. 0 to 65535
  - c. -32768 to 32767**
  - d. None of the above.
2. The following sentences make certain statements about floating-point Digital Signal Processors. Which one of the following statements is true?
  - a. Floating-point processors are usually cheaper than their fixed-point counterparts because they contain less silicon and have less external pins.
  - b. Floating-point devices generally have faster clock cycle rates.
  - c. Floating-point processors are usually more precise and have a larger dynamic range compared with fixed-point processors.**
  - d. A typical 24-bit floating-point processor stores coefficients and data values with 24-bit precision.

# Processor Arithmetic

3. Which one of the following choices lists the correct 2s-format bit-weights?

- a.  $2^{15} 2^{14} 2^{13} 2^{12} 2^{11} 2^{10} 2^9 2^8 2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0$
- b.  $-2^{15} 2^{14} 2^{13} 2^{12} 2^{11} 2^{10} 2^9 2^8 2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0$
- c.  $-2^0 2^{-1} 2^{-2} 2^{-3} 2^{-4} 2^{-5} 2^{-6} 2^{-7} 2^{-8} 2^{-9} 2^{-10} 2^{-11} 2^{-12} 2^{-13} 2^{-14} 2^{-15}$
- d.  $-2^0 2^1 2^2 2^3 2^4 2^5 2^6 2^7 2^8 2^9 2^{10} 2^{11} 2^{12} 2^{13} 2^{14} 2^{15}$



4. What is the decimal value of the above binary number in Q15-, Q14- and Q13-format?

- a. -19138, 46398, 13630, respectively
- b. approximately 1.4160, 2.8319, 5.6638 respectively
- c. **approximately -0.58405, -1.1681, -2.3362, respectively**
- d. None of the above.

5. Which of the following statements describes correctly how the decimal representation,  $D_{Q14}$ , of a Q14-format binary number is related to the decimal representation,  $D_{2s}$ , of the same binary number read in 2s-format?

- a.  $D_{Q14} = 2^{-15} \cdot D_{2s}$
- b.  **$D_{Q14} = 2^{-14} \cdot D_{2s}$**
- c.  $D_{Q14} = 2^{15} \cdot D_{2s}$
- d.  $D_{Q14} = 2^{14} \cdot D_{2s}$

