

# Contents

- Slide 1-1 Some DSP Chip History
- Slide 1-3 Other DSP Manufacturers
- Slide 1-4 DSP Applications
- Slide 1-5 TMS320C6713 DSP Starter Kit (DSK)
- Slide 1-6 TMS320C6713 DSK Features
- Slide 1-7 TMS320C6713 Architecture
- Slide 1-8 Main 'C6713 Features
- Slide 1-9 'C6713 Features (cont. 1)
- Slide 1-10 'C6713 Features (cont. 2)
- Slide 1-11 Instructions Common to C62x and C67x
- Slide 1-12 Extra Instructions for the C67x
- Slide 1-13 Addressing Modes
- Slide 1-14 Indirect Addresses (cont.)
- Slide 1-15 TMS320C6713DSK Memory Map
- Slide 1-16 Parallel Operations
- Slide 1-17 TMS320C6x Pipeline Phases
- Slide 1-18 Pipeline Operation
- Slide 1-19 TI Software Tools
- Slide 1-20 Building Programs
- Slide 1-21 Other Software
- Slide 1-22 Hardware and Software References
- Slide 1-23 C6713 DSK References
- Slide 1-24 First Lab Session
- Slide 1-25 First Lab Session (cont. 1)

**Slide 1-26** First Lab Session (cont. 2)  
**Slide 1-27** First Lab Session (cont. 3)  
**Slide 1-28** First Lab Session (cont. 4)

## Some DSP Chip History

### First Commercial DSP's

- 1982 – NEC  $\mu$ PD7720
- 1982 – Texas Instruments TMS 32010

These chips initially cost around \$600. DSP's now cost from \$2 to \$200.

### Texas Instruments (TI) DSP Families

- C2000 series  
32-bit fixed-point DSP's  
real-time control applications including digital motor control and power conversion
- C5000 Ultra Low Power series  
16-bit fixed-point DSP's  
Portable devices in audio, voice, communications, medical, security and industrial applications. Used cell phones.
- C6000 High Performance DSP's  
32-bit fixed and floating-point DSP's  
Audio, video, and imaging applications

- C6000 High Performance Multicore DSP's  
Combine both fixed and floating-point capabilities  
Medical imaging, test and automation, video infrastructure, and high-end imaging.
- DaVinci Digital Video Processors  
Optimized for digital video systems. Digital audio, video, imaging, and vision applications. Includes a general purpose processor, video accelerators, an optional DSP, and related peripherals.
- Digital Signal Processor and Arm Microprocessor Platforms  
Include DSP's from the TI families and ARM MPU processors in one chip.
- Custom high performance DSP's  
Designed for special customers like manufacturers of 3G and 4G base stations. Include peripherals like FFT units and Turbo Code decoders.

## **Some Other DSP Manufacturers**

LSI (Lucent, Agere), Freescale Semiconductor (Motorola), Analog Devices, Zilog

## **Fixed vs. Floating-Point DSP's**

- Fixed-point DSP's are cheaper and use less power but care must be taken with scaling to avoid over and underflow.
- Floating-point DSP's are easier to program. Numbers are automatically scaled. They are more complicated and expensive.

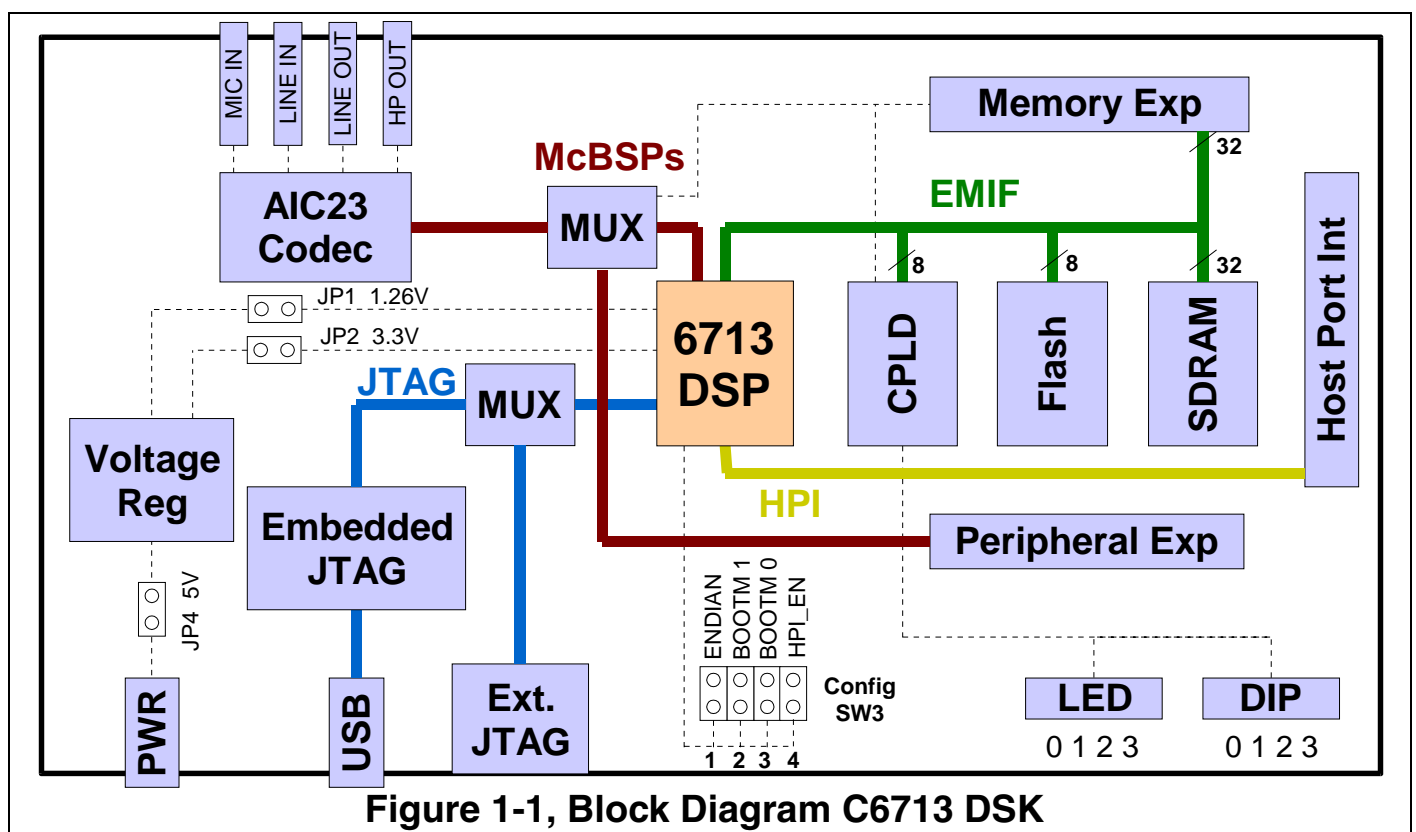
## **Advantages of DSP's over Analog Circuits**

- Can implement complex linear or nonlinear algorithms.
- Can modify easily by changing software.
- Reduced parts count makes fabrication easier.
- High reliability

## DSP Applications

- **Telecommunications:** telephone line modems, FAX, cellular telephones, wireless networks, speaker phones, answering machines
- **Voice/Speech:** speech digitization and compression, voice mail, speaker verification, and speech synthesis
- **Automotive:** engine control, antilock brakes, active suspension, airbag control, and system diagnosis
- **Control Systems:** head positioning servo systems in disk drives, laser printer control, robot control, engine and motor control, and numerical control of automatic machine tools
- **Military:** radar and sonar signal processing, navigation systems, missile guidance, HF radio frequency modems, secure spread spectrum radios, and secure voice
- **Medical:** hearing aids, MRI imaging, ultrasound imaging, and patient monitoring
- **Instrumentation:** spectrum analysis, transient analysis, signal generators
- **Image Processing:** HDTV, image enhancement, image compression and transmission, 3-D rotation, and animation

# TMS320C6713 DSP Starter Kit (DSK) Block Diagram



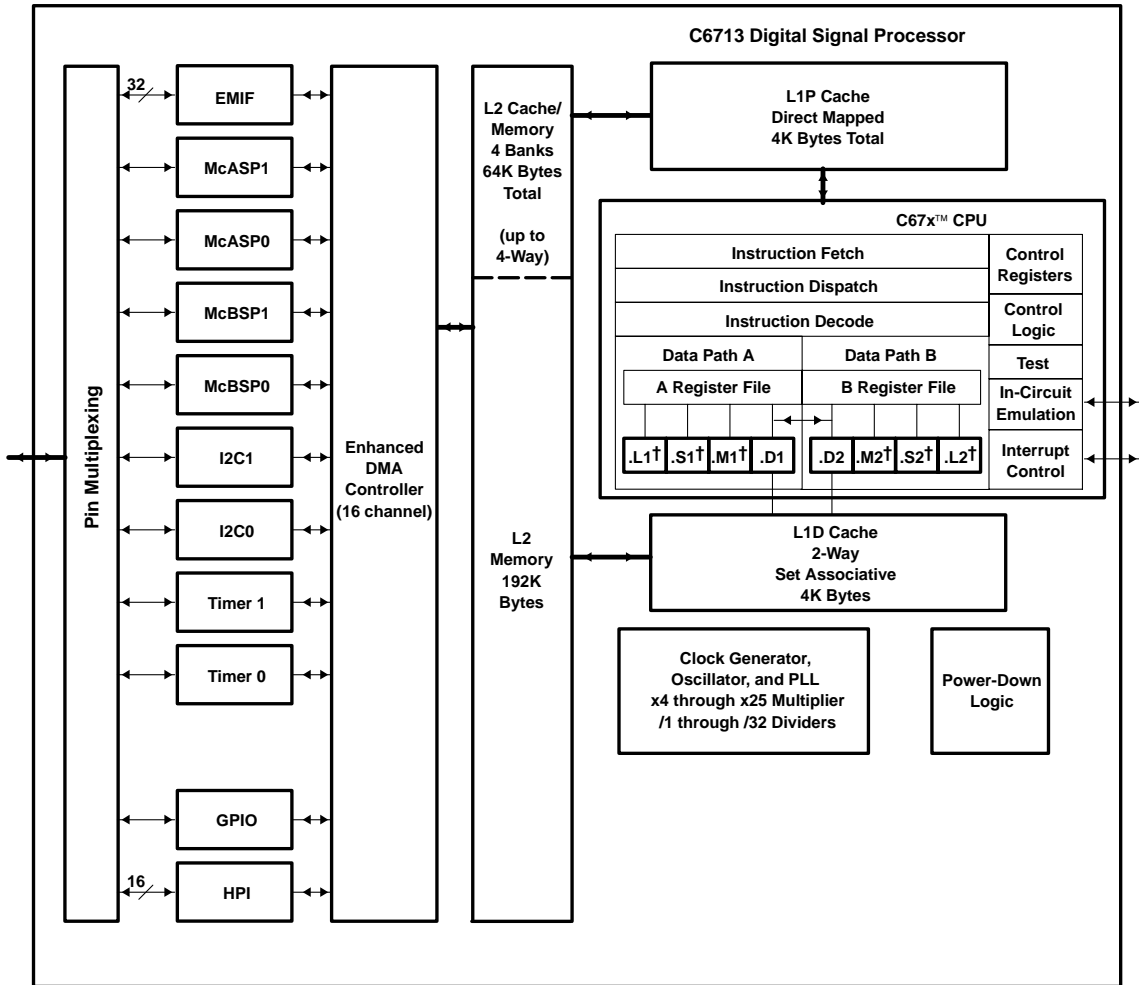
## **TMS320C6713 DSK Features**

- A TMS320C6713 DSP operating at 225 MHz.
- An AIC23 stereo codec with Line In, Line Out, MIC, and headphone stereo jacks
- 16 Mbytes of synchronous DRAM
- 512 Kbytes of non-volatile Flash memory (256 Kbytes usable in default configuration)
- 4 user accessible LEDs and DIP switches
- Software board configuration through registers implemented in CPLD
- Configurable boot options
- Expansion connectors for daughter cards
- JTAG emulation through on-board JTAG emulator with USB host interface or external emulator



# TMS320C6713 Architecture

functional block and CPU (DSP core) diagram



† In addition to fixed-point instructions, these functional units execute floating-point instructions.

EMIF interfaces to:  
 -SDRAM  
 -SBSRAM  
 -SRAM,  
 -ROM/Flash, and  
 -I/O devices

McBSPs interface to:  
 -SPI Control Port  
 -High-Speed TDM Codecs  
 -AC97 Codecs  
 -Serial EEPROM

McASPs interface to:  
 -I2S Multichannel ADC, DAC, Codec, DIR  
 -DIT: Multiple Outputs

*TMS320C6713, TMS320C6713B Floating-Point Digital Signal Processor,*

SPRS186I, p. 12.

## Main 'C6713 Features

- VelociTI Very Long Instruction Word (VLIW) CPU Core
  - Fetches eight 32-bit instructions at once
    - Eight Independent functional units
      - \* Four ALUs (fixed and floating-point)
      - \* Two ALUs (fixed-point)
      - \* Two multipliers (fixed and floating-point)
        - $32 \times 32$  bit integer multiply with 32 or 64-bit result
    - Load-store architecture with 32 32-bit general purpose registers
- Instruction Set Features
  - Hardware support for IEEE single and double precision floating-point operations
  - 8, 16, and 32-bit addressable
  - 8-bit overflow protection and saturation
  - Bit-field extract, set, clear; bit-counting; normalization

## 'C6713 Features (cont. 1)

- L1/L2 Memory Architecture
  - 4K-Byte L1P Program Cache (Direct-Mapped)
  - 4K-Byte L1D Data Cache (2-Way)
  - 256K-Byte L2 Memory Total; 64K-Byte L2 Unified Cache/Mapped RAM and 192K-Byte Additional L2 Mapped RAM
- Device Configuration
  - Boot Mode: HPI, 8-, 16-, 32-Bit ROM Boot
  - Little Endian and Big Endian
- 32-bit External Memory Interface (EMIF)
  - Glueless interface to SDRAM, Flash, SBSRAM, SRAM, and EPROM
  - 512M-byte Total Addressable External Memory Space

## 'C6713 Features (cont. 2)

- Enhanced Direct-Memory-Access (EDMA) Controller (16 Independent Channels)
- 16-Bit Host-Port Interface (HPI)
- Two Inter-Integrated Circuit Bus (I<sup>2</sup>C Bus) Multi-Master and Slave Interfaces
- Two Multichannel Audio Serial Ports (McASPs)
- Two Multichannel Buffered Serial Ports (McBSPs)
- Two 32-Bit General Purpose Timers
- Dedicated GPIO Module with 16 pins
- Flexible Phase-Locked-Loop (PLL) Based Clock Generator Module
- IEEE-1149.1 JTAG Boundary Scan

## Instructions Common to C62x and C67x

.L unit	.M Unit	.S Unit		.D Unit	
ABS	MPY	ADD	SET	ADD	STB (15-bit offset) <sup>2</sup>
ADD	MPYU	ADDK	SHL	ADDAB	STH (15-bit offset) <sup>2</sup>
ADDU	MPYUS	ADD2	SHR	ADDAH	STW (15-bit offset) <sup>2</sup>
AND	MPYSU	AND	SHRU	ADDAW	SUB
CMPEQ	MPYH	B disp	SSHL	LDB	SUBAB
CMPGT	MPYHU	B IRP <sup>1</sup>	SUB	LDBU	SUBAH
CMPGTU	MPYHUS	B NRP <sup>1</sup>	SUBU	LDH	SUBAW
CMPLT	MPYHSU	B reg	SUB2	LDHU	ZERO
CMPLTU	MPYHL	CLR	XOR	LDW	
LMBD	MPLHU	EXT	ZERO	LDB (15-bit offset) <sup>2</sup>	
MV	MPYHULS	EXTU		LDBU (15-bit offset) <sup>2</sup>	
NEG	MPYHSLU	MV		LDH (15-bit offset) <sup>2</sup>	
NORM	MPYLH	MVC <sup>1</sup>		LDHU (15-bit offset) <sup>2</sup>	
NOT	MPYLHU	MVK		LDW (15-bit offset) <sup>2</sup>	
OR	MPYLUHS	MVKH		MV	
SADD	MPYLSHU	MVKLH		STB	
SAT	SMPY	NEG		STH	
SSUB	SMPYHL	NOT		STW	
SUB	SMPYLH	OR			
SUBU	SMPYH				
SUBC					
XOR					
ZERO					

See *TMS320C6000 CPU and Instruction Set, Reference Guide*, SPRU189F for complete descriptions of instructions.

## Extra Instructions for the C67x

.L unit	.M Unit	.S Unit	.D Unit
ADDDP	MPYDP	ABSDP	ADDAD
ADDSP	MPYI	ABSSP	LDDW
DPINT	MPYID	CMPEQDP	
DPSP	MPYSP	CMPEQSP	
DPTRUNC		CMPGTDP	
INTDP		CMPGTSP	
INTDPU		CMPLTDP	
INTSP		CMPLTSP	
INTSPU		RCPDP	
SPINT		RCPSP	
SPTRUNC		RSQRDP	
SUBDP		RSQRSP	
SUBSP		SPDP	

See *TMS320C6000 CPU and Instruction Set, Reference Guide*, SPRU189F for complete descriptions of instructions.

## Addressing Modes

- Linear Addressing – with all registers
- Circular Addressing – with registers A4–A7 and B4–B7

### Forms for Indirect Addresses

- Register Indirect

No Modification	*R
Preincrement of	*++R
Predecrement of	*--R
Postincrement of	*R++
Postdecrement of	*R--

- Register Relative

No Modification	*±R[ucst5]
Preincrement of	*++R[ucst5][ucst5]
Predecrement of	*--R[ucst5]
Postincrement of	*R++[ucst5]
Postdecrement of	*R--[ucst5]

## Forms for Indirect Addresses (cont.)

- Register Relative with 15-bit Constant Offset
  - No Modification      $*_{+B14/B15}[ucst15]$
- Base + Index
  - No Modification      $*_{\pm R}[offsetR]$
  - Preincrement of      $*_{++R}[offsetR]$
  - Predecrement of     $*_{--R}[offsetR]$
  - Postincrement of     $*R_{++}[offsetR]$
  - Postdecrement of    $*R_{--}[offsetR]$

Notes:

$ucst5$  = 5-bit unsigned integer constant

$ucst15$  = 15-bit unsigned integer constant

$R$  = base register

$offsetR$  = index register

**Example:** `LDW .D1 *++A4[9], A1`

Load a 32-bit word using functional unit D1 into register A1 from the memory byte address:

*contents of*  $(A4) + 4 \times 9$



## TMS320C6713DSK Memory Map

Address	C67x Family Memory Type	C6713DSK
0x00000000	Internal Memory	Internal Memory
0x00030000	Reserved Space or Peripheral Regs	Reserved or Peripheral
0x80000000	EMIF CE0	SDRAM
0x90000000	EMIF CE1	Flash
0x90080000		CPLD
0xA0000000	EMIF CE2	Daughter Card
0xB0000000	EMIF CE3	

## Parallel Operations

- The instruction word for each functional unit is 32 bits long.
- Instructions are fetched 8 at a time consisting of  $8 \times 32 = 256$  bits. The group is called a *fetch packet*. Fetch packets must start at an address that is a multiple of 8 32-bit words.
- Up to 8 instructions can be executed in parallel. Each must use a different functional unit. Each group of parallel instructions is called an *execute packet*.
- The  $p$ -bit (bit 0) determines if an instruction executes in parallel with another. The instructions are scanned from the lowest address to the highest. If the  $p$ -bit of instruction  $i$  is 1, then instruction  $i + 1$  is executed in parallel with instruction  $i$ . If it is 0, instruction  $i + 1$  is executed one cycle after instruction  $i$ .

## TMS320C6x Pipeline Phases

Stage	Phase	Symbol
Program Fetch	Program Address Generation	PG
	Program Address Sent	PS
	Program Wait	PW
	Program Data Receive	PR
Program Decode	Dispatch	DP
	Decode	DC
Execute	Execute 1	E1
	⋮	⋮
	Execute 10	E10

See *TMS320C6000 CPU and Instruction Set Reference Guide*, SPRU189F, Table 7-1, pp. 7-7 to 7-9, for details of pipeline phases.

## Pipeline Operation Asuming One Execute Packet per Fetch Packet

Clock Cycle	Fetch Packet										
	$n$	$n+1$	$n+2$	$n+3$	$n+4$	$n+5$	$n+6$	$n+7$	$n+8$	$n+9$	$n+10$
1	PG										
2	PS	PG									
3	PW	PS	PG								
4	PR	PW	PS	PG							
5	DP	PR	PW	PS	PG						
6	DC	DP	PR	PW	PS	PG					
7	E1	DC	DP	PR	PW	PS	PG				
8	E2	E1	DC	DP	PR	PW	PS	PG			
9	E3	E2	E1	DC	DP	PR	PW	PS	PG		
10	E4	E3	E2	E1	DC	DP	PR	PW	PS	PG	
11	E5	E4	E3	E2	E1	DC	DP	PR	PW	PS	PG
12	E6	E5	E4	E3	E2	E1	DC	DP	PR	PW	PS
13	E7	E6	E5	E4	E3	E2	E1	DC	DP	PR	PW
14	E8	E7	E6	E5	E4	E3	E2	E1	DC	DP	PR
15	E9	E8	E7	E6	E5	E4	E3	E2	E1	DC	DP
16	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	DC
17		E10	E9	E8	E7	E6	E5	E4	E3	E2	E1

### Need for NOP's

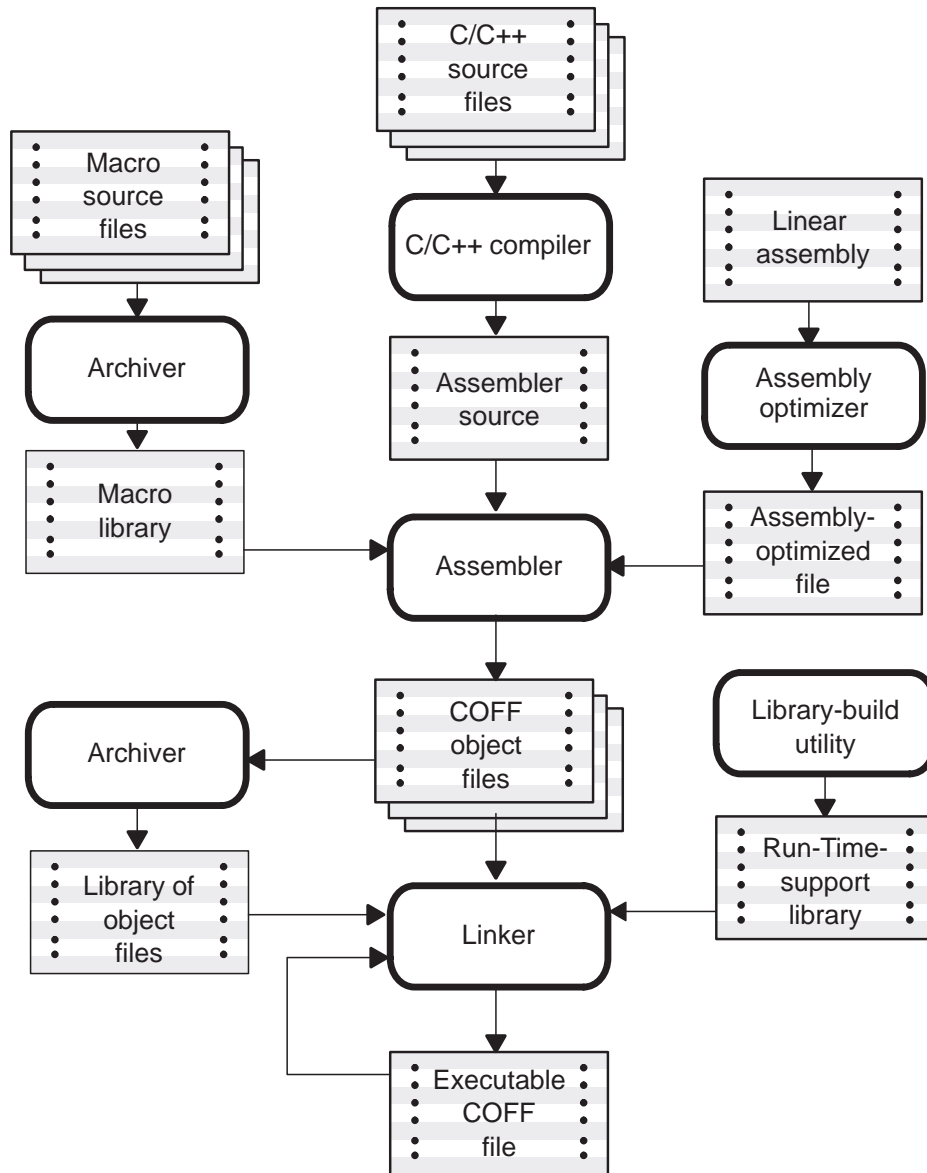
- Different instruction types require from 1 to 10 execution phases. Therefore, NOP instructions must be added to make sure results of one instruction are needed by another.
- NOP's can be added manually in hand coded assembly (hard), in linear assembly by the assembler (easier), or by the C compiler (easiest).

# TI Software Tools

## Code Composer Studio Version 6

- Built on Eclipse platform
- Create and edit source code
- Compile (cl6x.exe), assemble (asm6x.exe), and link (lnk6x.exe) programs using project “.pjt” files. (Actually, cl6x.exe is a shell program that can compile, assemble and link.)
- Build libraries with ar6x.exe
- Include a real-time operating system, DSP/BIOS, in the DSP code with real-time data transfer (RTDX) between the PC and DSP
- Load programs into DSP, run programs, single step, break points, read memory and registers, profile running programs, etc.

# Building Programs



*TMS320C6000 Optimizing Compiler User's Guide (SPRU187I, April 2001, Figure 1-1, p. 1-2)*

## Other Software

- **Microsoft Visual C++**
- **MATLAB**
- **Digital Filter Design Programs**
  - **Old Console Versions**
    - \* WINDOW.EXE
    - \* REMEZ.EXE
    - \* IIR.EXE
    - \* RASCOS.EXE
    - \* SQRTRACO.EXE
  - **Windows Forms Class Versions**
    - \* FirWindowFunct.exe
    - \* IIRWFC.exe
    - \* remezWFC.exe
    - \* RaisedCosine.exe
    - \* SquareRootRaisedCosine.exe
- **Standard MS Windows Programs like MS Word and Excel**
- **SSH Terminal Program (PUTTY) and SSH File Transfer Program (WINS CP)**

## Hardware and Software References

TI documents for the hardware and software tools are all available online at [www.ti.com](http://www.ti.com). Use the TI search engine to find the particular part or document. Enter a document number like “SPRU189” shown in the first item below in the TI search box. You will probably find more up-to-date versions of the documents than the ones listed below. In particular, the following documents will be very useful and are also available locally in the folder

C:\c6713dsk\docs\C6713 DSP User Guides:

1. *TMS320C6000 CPU and Instruction Set Reference Guide*, SPRU189F, October 2000.
2. *TMS320C6000 Peripherals Reference Guide*, SPRU190D, March 2001.
3. *TMS320C6000 Chip Support Library API Reference Guide*, SPRU401b, April 2001.
4. *TMS320C6000 Optimizing Compiler User's Guide*, SPRU187I, April 2001



## **Hardware and Software References for the C6713 DSK**

For detailed information about the c6713 DSK hardware, see the file on the local C drive:

`C:\c6713dsk\docs\6713_dsk_techref.pdf`

For additional information about the c6713 DSK hardware and documentation on how to use the Board Support Library and the functions it includes, double click on the file on the local C drive:

`C:\c6713dsk\docs\help\c6713dsk.hlp`

## First Lab Session

**No lab report is required for Chapter 1.** The software utility you will use to generate and edit source code, build executable DSP programs, and load these programs into the 'C6713 DSK is called *Code Composer Studio*. Do the following introductory tasks for your first lab session to learn about the hardware and software tools.

### **1. Check out the hardware.**

The DSK has been installed inside the PC case to keep it secure and allow you access to the lab outside of regular class hours. The important DSK connectors have been brought out to the side of the PC case.

The DSK is connected to a USB port on the motherboard and the power supply has been brought out to an external plug.

Find the stereo connectors for the A/D and D/A converters on the case. Notice that the connectors are labeled MIC IN, LINE IN, LINE OUT, and HEADPHONE. The MIC IN input is for low voltage signals. For ENEE 428 you should use only the LINE IN and LINE OUT connectors.

## First Lab Session (cont. 1)

### 2. Learn About Code Composer Studio.

Start Code Composer Studio (CCS) by double clicking on its icon that looks like a Rubik's cube.

1. The “Workspace Launcher” window will appear. In the first lab session before shared ECE labs server files are set up, use the default workspace `C:\Users\<<your login name>\workspace_v6_0`. Once your shared file is set up, use it as your default workspace.

**Warning:** The lab PC's are re-imaged when rebooted and any files you store on the C drive will disappear.

2. In a web browser like Google Chrome or Microsoft Edge, go to [www.YouTube.com](http://www.YouTube.com). Search for “Getting Started with Code Composer Studio v6.2” and watch it up until “App Center.”

## First Lab Session (cont. 2)

3. Click on “Help” and then “Help Contents.”
  - (a) Expand “Workbench User Guide” and then “Getting Started.”
    - i. Work through the “Basic Tutorial” following each “Contents” item in order.
    - ii. Skip the rest of the “Workbench User Guide”.
  - (b) Expand “Code Composer Studio Help.”
    - i. Expand “Getting Started Quickly.”
      - Work through “Creating a New CCS Project.”
      - Go through the “Graphs Overview.”
      - Skip the other topics in “Getting Started Quickly.”
    - ii. Expand “Views and Editors.”
      - Carefully work through the “Breakpoints View” topic. You will use breakpoints in some experiments for code debugging and measuring the execution load of parts of your programs.
      - Go through “Graph Views”.
      - Quickly browse through the other topics under “Views and Editors.”

## First Lab Session (cont. 3)

### 3. Do the Classical “Hello World” First C Project

1. Start Code Composer Studio, Click on “Project” and then “New CCS Project”
2. Enter “HelloWorld” for the “Project name:”
3. On the “Target” line leave the box on the left alone. Open the box on the right and select “DSK6713” from the drop-down menu.
4. Open the “Connection:” box menu and select “Spectrum Digital DSK-EVM eZdsp onboard USB Emulator.”
5. Select the TI v7.4.24 compiler.
6. Under “Basic Examples” select “Hello World” and click “Finish.”
7. The C code should appear in the right panel. Replace the line “return(0);” with the line “for(;;);” which is an infinite loop.

## First Lab Session (cont. 4)

8. Build the project by clicking on the green bug icon and run the program. You should see “Hello World!” in the bottom of the CCS window.

**4. Experiment with capturing screen images and channel data from the Agilent oscilloscope to the PC.** See the explanation of how to do this on the class web site after Chapter 1.

**5. Explore the other “Help” topics** if you finish all the above items before the end of your first lab session. You can also browse through the TI manuals for the TMS320C6713 DSP CPU and peripherals. You can find them on the bookshelf and in C:\c6713dsk\docs

**You should stay the entire time for each three hour lab period. If you finish a current experiment go on to the next one.**