The TMS320C6x Family: Hardware and Software

Introduction

In this chapter the DSPs of primary focus for the course, the TMS320C6x, will be introduced and explained in terms of hardware, software, and development environments found in the laboratory. The specific C6x family member of most interest is the C6748.

The Family of TI DSP Processors (ordered by price/performance)

<table>
<thead>
<tr>
<th>32-bit real-time MCUs</th>
<th>Ultra Low power DSP</th>
<th>DSP DSP+ARM</th>
<th>Multi-core DSP</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>C2000™ Delfino™ Piccolo™</strong></td>
<td>Up to 300 MHz +Accelerator</td>
<td><strong>DaVinci™ video processors Integra™</strong></td>
<td><strong>C6000™</strong></td>
</tr>
<tr>
<td>40MHz to 300 MHz</td>
<td>Up to 320KB RAM Up to 128KB ROM</td>
<td>300MHz to &gt;1Ghz +Accelerator</td>
<td>24,000 MMACS</td>
</tr>
<tr>
<td>Flash, RAM 16 KB to 512 KB</td>
<td>USB, ADC McBSP, SPI, I²C</td>
<td>Cache RAM, ROM</td>
<td>Cache RAM, ROM</td>
</tr>
<tr>
<td>PWM, ADC, CAN, SPI, I²C</td>
<td>Audio, Voice Medical, Biometrics</td>
<td>USB, ENET, PCIe, SATA, SPI</td>
<td>SRIO, EMAC DMA, PCIe</td>
</tr>
<tr>
<td>Motor Control, Digital Power, Lighting, Ren. Enrgy</td>
<td>$3.00 to $10.00</td>
<td>Floating/Fixed Point Video, Audio, Voice, Security, Conferencing</td>
<td>Telecom test &amp; meas media gateways, base stations</td>
</tr>
<tr>
<td>$1.50 to $20.00</td>
<td></td>
<td>$5.00 to $200.00</td>
<td>$40 to $200.00</td>
</tr>
</tbody>
</table>
DSP Devices Overview

- TI has four classes of DSP processors
  - *C2000 Defino and Piccolo*: Devices are 32-bit microcontrollers with high performance integrated peripherals designed for real-time control applications. Its math-optimized core gives designers the means to improve system efficiency, reliability, and flexibility. Powerful integrated peripherals make C2000 devices the perfect single-chip control solution. C2000’s development tools strategy and software (controlSUITE) create an open platform with the goal of maximizing usability and minimizing development time.
Control Subsystem

- **Features**
  - Precision Control
  - Industry leading computational performance
  - Expanded instruction set
  - Industry’s highest-resolution PWMs

- **Control Modules**
  - 9x PWM Modules: 18x Outputs / 16x HR
  - Fault Trip Zones
  - 6x 32-bit eCAP
  - 3x 32-bit QEP

- **Comms**
  - VCU
  - i2C
  - CAN

- **System**
  - 6Ch DMA

- **Memory**
  - 256-512 KB ECO Flash
  - 20 KB ECO RAM
  - 128-bit Security
  - 16 KB Party RAM
  - 64 KB ROM

Host Subsystem

- **Ecosystem for Developers**
  - ARM Cortex M3
  - System & Clocking
  - Communications

- **Shared**
  - Analog Temp Sense
  - 12b, 10ch, 25K, 3 MSPS
  - 3ch Analog Comparators

- **Memory**
  - 256-512KB ECC Flash
  - 128 KB ECC RAM
  - 2x128-bit Security
  - 16 KB Party RAM
  - 64 KB ROM
  - External Interface

- **Pwr & Clocking**
  - 3.3V / 5V INT OSC
  - 40 MHz / 25 MHz OSEK
  - 3.3V / 5V F elections
  - 12V H-Bridge
  - 2x I2C

- **System & Clocking**
  - 32ChDMA
  - 4 Timers
  - 2 Watchdogs

- **Communications**
  - 10/100 Ethernet NAC
  - USB OTG FS PHY
  - 4x SSI
  - 5x UART
  - 2x I2C
  - 2x CAN

---

**controlSUITE™ Software**

- Comprehensive. Intuitive. Optimized
- Solutions for every design stage
- Unique real-time control IP
- Unparalleled access

Download

---

**Featured Application – Digital Power Conversion**

- Servers/telecom rectifiers
- Hybrid electric vehicles
- Solar energy products
- Street/building lighting
- Wind energy products
– **TMS320C5000™ Power Efficient DSPs**: Very low standby power and advanced power management, for personal and portable products; GPS receivers and medical (a short intro to the VC5505 eZDSP at the end of the chapter)

– **TMS320C6000™ DSPs**: (see below)

**The C6x Families**

- **C64x High Performance DSPs**: Very fast fixed-point processing, with up to 1.2 GHz clock speed (9600 MMACs on C6455-1200, 24000 MMACs on CC6474-1000 with 3 cores)
- **C62x Performance Value DSPs**: High performance and high cost efficiency; optimized for wireless infrastructure, telecom infrastructure, and imaging applications (5760 MMACs on C6412-720)
- **C67x Floating Point DSPs**: The most advanced DSP C compiler and assembly optimizer for efficiency and performance; high performance audio applications (e.g., C6748 375/456-MHz Fixed/Floating point, up to 3648/2746 MIPS/MFLOPS)

**C6000 DSP Platform Roadmap**

![C6000 DSP Platform Roadmap Diagram](image-url)
Binary Compatibility

- Native instructions for IEEE 754, SP & DP
- Advanced VLIW architecture
- Enhanced floating point add capabilities
- Audio-specific and mixed precision instructions
- SPLOOP and 16-bit instructions for smaller code size
- Flexible level one memory architecture
- IDMA for rapid data transfers between local memories
- Best of fixed point and floating point architecture for better system performance and faster time-to-market
- 2x registers
- 100% upward object code compatible with C64x, C64x+, C67x and C67x+

- Four 16-bit or eight 8-bit MACs
- Advanced fixed point instructions
- Two-level cache
- SPLOOP and 16-bit instructions

- Flexible level one memory architecture
- IDMA for rapid data transfers between local memories
- Best of fixed point and floating point architecture for better system performance and faster time-to-market
- 2x registers
- 100% upward object code compatible with C64x, C64x+, C67x and C67x+

- Four 16-bit or eight 8-bit MACs
- Advanced fixed point instructions
- Two-level cache
- SPLOOP and 16-bit instructions

- Flexible level one memory architecture
- IDMA for rapid data transfers between local memories
- Best of fixed point and floating point architecture for better system performance and faster time-to-market
- 2x registers
- 100% upward object code compatible with C64x, C64x+, C67x and C67x+
Peripheral Options

- Power protection systems
- Industrial automation
- Barcode scanner
- Test & measurement
- SDR
- Audio conf systems
- Portable data term
- Intelligent occupancy sensors
- Bar code scanner
- Audio effects
- Audio mixers
- Software & Pin -for- Pin Compatible Across Family

- OMAP-L138
- C6748
- C6746
- C6742
- ARM9
- C674x DSP 200 MHz
- C674x DSP 300 MHz
- 2 x 150 MHz CPU
- 488 KB
- 320 KB
- 128 KB
- 448 KB
- 384 KB
- 320 KB
- 2 x 150 MHz CPU
- 2 x 150 MHz CPU
- 128 KB
- 2 x 150 MHz CPU
- 488 KB
- 320 KB
- ARM9

Target applications:
- SDR
- Audio conf systems
- Portable data term
- Intelligent occupancy sensors
- Bar code scanner
- Audio effects
- Audio mixers
- Power protection systems
- Industrial automation
- Barcode scanner
- Test & measurement

Software & Pin -for- Pin Compatible Across Family

- DSP
- UART
- McASP
- SPI
- I2C
- Key
- Memory
- PRU
- Peripheral

Price:
- $6.70@1Ku
- $5.95@10Ku
C6x Family Feature Overview

- To appreciate the C6x family consider a little history first
- The first of TI’s high performance floating point processors was the C3x
- In 1988 the first of the TMS320C3x’s began shipping at a cost of $1,300 each
- At the present time development of the C3x family has slowed, but a core of users still exists in the market place
- In 2000 TI introduced the C33 which is capable of 150 MFLOPS
  - This new processor featured two 1k and two 16k dual access RAM blocks
  - Consumes 0.2 W and costs $5-8 in 100KU
  - The pilot offering of this course, in 1998, used the C31 which comes in low cost 60 ns, lowers cost 74 ns, highest speed 40 ns (used in the C31 DSK), and other single-cycle execution time versions
- Today the C6x family, first announced in 1Q97, continues the high performance traditions of the C3x family, but offers much more in terms of both hardware and software
- Features of the C6x family include:
  - Code compatible fixed- and floating-point
  - Widely adopted by broad-band infrastructure vendors
- Highly parallel VelociTI™ advanced very long instruction word (VLIW) architecture
- RISC-like instructions
- Claim industry’s most efficient C compiler to ease high level language (HLL) development
- Low price points ~ C6738-300 (300 MHz) is $15.75 in 1ku
## Comparison Matrix

<table>
<thead>
<tr>
<th>Product Attributes</th>
<th>C5674x</th>
<th>C5671x</th>
<th>C5670x</th>
<th>C56410</th>
<th>C6714s</th>
<th>C6717</th>
<th>C6714c</th>
<th>C6714h</th>
</tr>
</thead>
<tbody>
<tr>
<td>Floating Point</td>
<td>2.1Gc</td>
<td>1.1Gc</td>
<td>2.1Gc</td>
<td>2.1Gc</td>
<td>1.1Gc</td>
<td>2.1Gc</td>
<td>1.1Gc</td>
<td>2.1Gc</td>
</tr>
<tr>
<td>Fixed Point</td>
<td>2.1Gc</td>
<td>1.1Gc</td>
<td>2.1Gc</td>
<td>2.1Gc</td>
<td>1.1Gc</td>
<td>2.1Gc</td>
<td>1.1Gc</td>
<td>2.1Gc</td>
</tr>
<tr>
<td>Core Clock (MHz)</td>
<td>200</td>
<td>200</td>
<td>200</td>
<td>200</td>
<td>200</td>
<td>200</td>
<td>200</td>
<td>200</td>
</tr>
<tr>
<td>Memory (KB)</td>
<td>128</td>
<td>128</td>
<td>128</td>
<td>128</td>
<td>128</td>
<td>128</td>
<td>128</td>
<td>128</td>
</tr>
<tr>
<td>External Memory</td>
<td>128</td>
<td>128</td>
<td>128</td>
<td>128</td>
<td>128</td>
<td>128</td>
<td>128</td>
<td>128</td>
</tr>
<tr>
<td>Internal Memory</td>
<td>512</td>
<td>512</td>
<td>512</td>
<td>512</td>
<td>512</td>
<td>512</td>
<td>512</td>
<td>512</td>
</tr>
<tr>
<td>I/O Ports</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>USB Port</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Ethernet Port</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Power Consumption</td>
<td>1W</td>
<td>1W</td>
<td>1W</td>
<td>1W</td>
<td>1W</td>
<td>1W</td>
<td>1W</td>
<td>1W</td>
</tr>
<tr>
<td>Price (USD)</td>
<td>240</td>
<td>240</td>
<td>240</td>
<td>240</td>
<td>240</td>
<td>240</td>
<td>240</td>
<td>240</td>
</tr>
</tbody>
</table>
C6x Architecture Overview

- **CPU Cores**
  - ARM926EJ-S™ (MPU) 300MHz+
  - C674x DSP Core 300MHz+
  - 2 PRU Cores upto 150 MHz each

- **Peripherals (1.8/ 3.3V IOs)**
  - 10/100 Ethernet MAC
  - EMIFA - SDRAM/NAND Flash
  - EMIFB – DDR (mDDR/DDR2)
  - Video Port I/F, SATA, uPP, LCDC

- **Power (1.0-1.2V Core, 1.8/3.3V IOs)**
  - Total Power < 440 mW @ 300MHz, 1.2V, 25C
    - For DSP at 70% loading, ARM at 50% loading; mDDR 50% active at 135MHz
  - Standby Power
    - < 9mW @ 1.2V/ 25C

- **Package**
  - 13 x13mm nFBGA (0.65mm), 16x16mm BGA (0.8mm)
  - Pin to pin compatible with C6748/6/2, AM1808/6

- **Applications**
  - Power Protection Systems, Test & Measurement, SDR, Bar Code Scanners, Portable Communications, Portable Medical, Portable Audio
Connectivity

Note: Not all 'C64x+ devices have all the various peripherals shown above. Please refer to the specific device’s datasheet.
High Level Architecture

Performance & Memory

- Up to 300MHz
- 256K L2 (up to 64K cache)
- 32K L1P & L1D Cache/SRAM
- 32-bit DDR2-266
- 16-bit EMIF (NAND Flash)

Communications

- 64-Channel EDMA 3.0
- 10/100 EMAC
- USB 1.1 & 2.0
- SATA
C6000 Core Architecture

- While the dual-MAC speeds math intensive algorithms, the flexibility of 8 independent functional units allows the compiler to quickly perform other types of processing
- Can dispatch up to 8 32-bit instructions every cycle
- All instructions are conditional – allowing efficient hardware pipelining
- The core contains 64 32-bit general purpose registers with few restrictions (aids compiler in generating more efficient code)
- Can perform up to EIGHT 16x16 multiplies/ACC per clk cycle
• MAU is 8 bits for program/data
• Compiler excels at natural C
• Data types: char = 8 bits, short = 16 bits, int = 32 bits, long = 40 bits, long long = 64 bits

Older C6713 Versus the Newer C674 & C64 (not C64+):
## Functional Unit Operations

<table>
<thead>
<tr>
<th>Functional Unit</th>
<th>Fixed–Point Operations</th>
<th>Floating–Point Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>.L unit (.L1, .L2)</td>
<td>32/40-bit arithmetic and compare operations 32-bit logical operations Leftmost 1 or 0 counting for 32 bits Normalization count for 32 and 40 bits Byte shifts Data packing/unpacking 5-bit constant generation Dual 16-bit arithmetic operations Quad 8-bit arithmetic operations Dual 16-bit min/max operations Quad 8-bit min/max operations</td>
<td>Arithmetic operations DP → SP, INT → DP, INT → SP conversion operations</td>
</tr>
<tr>
<td>.S unit (.S1, .S2)</td>
<td>32-bit arithmetic operations 32/40-bit shifts and 32-bit bit-field operations 32-bit logical operations Branches Constant generation Register transfers to/from control register file (.S2 only) Byte shifts Data packing/unpacking Dual 16-bit compare operations Quad 8-bit compare operations Dual 16-bit shift operations Dual 16-bit saturated arithmetic operations Quad 8-bit saturated arithmetic operations</td>
<td>Compare Reciprocal and reciprocal square–root operations Absolute value operations SP → DP conversion operations</td>
</tr>
<tr>
<td>.M unit (.M1, .M2)</td>
<td>16 x 16 multiply operations 16 x 32 multiply operations Quad 8 x 8 multiply operations Dual 16 x 16 multiply operations Dual 16 x 16 multiply with add/subtract operations Quad 8 x 8 multiply with add operation Bit expansion Bit interleaving/de-interleaving Variable shift operations Rotation Galois Field Multiply</td>
<td>32 X 32–bit fixed–point multiply operations Floating–point multiply operations</td>
</tr>
<tr>
<td>.D unit (.D1, .D2)</td>
<td>32-bit add, subtract, linear and circular address calculation Loads and stores with 5-bit constant offset Loads and stores with 15-bit constant offset (.D2 only) Load and store double words with 5-bit constant Load and store non-aligned words and double words 5-bit constant generation 32-bit logical operations</td>
<td>Load doubleword with 5–bit constant offset</td>
</tr>
</tbody>
</table>

**Note:** Fixed-point operations are available on all three devices. Floating-point operations and 32 x 32-bit fixed-point multiply are available only on the ‘C67x. Additional ‘C64x functions are shown in bold.
C6748 Data Paths

Data path A

Data path B

A. On .M unit, dst2 is 32 MSB.
B. On .M unit, dst1 is 32 LSB.
C. On C64x CPU .M unit, src2 is 32 bits; on C64x+ CPU .M unit, src2 is 64 bits.
D. On .L and .S units, odd dst connects to odd register files and even dst connects to even register files.
• Each C64x+ .M unit can perform one of the following each clock cycle:
  – one 32 x 32 bit multiply, one 16 x 32 bit multiply, two 16 x 16 bit multiplies, two 16 x 32 bit multiplies, two 16 x 16 bit multiplies with add/subtract capabilities, four 8 x 8 bit multiplies, four 8 x 8 bit multiplies with add operations, and four 16 x 16 multiplies with add/subtract capabilities (including a complex multiply)
  – There is also support for Galois field multiplication for 8-bit and 32-bit data
  – The complex multiply (CMPY) instruction takes for 16-bit inputs and produces a 32-bit real and a 32-bit imaginary output
  – There are also complex multiplies with rounding capability that produces one 32-bit packed output that contain 16-bit real and 16-bit imaginary values
  – The 32 x 32 bit multiply instructions provide the extended precision necessary for audio and other high-precision algorithms on a variety of signed and unsigned 32-bit data types
• The .L or (Arithmetic Logic Unit) now incorporates the ability to do parallel add/subtract operations on a pair of common inputs
  – Versions of this instruction exist to work on 32-bit data or on pairs of 16-bit data performing dual 16-bit add and sub-
tracts in parallel
- There are also saturated forms of these instructions

- The C64x+ core enhances the .S unit in several ways
  - In the C64x core, dual 16-bit MIN2 and MAX2 comparisons were only available on the .L units
  - On the C64x+ core they are also available on the .S unit which increases the performance of algorithms that do searching and sorting
  - Finally, to increase data packing and unpacking throughput, the .S unit allows sustained high performance for the quad 8-bit/16-bit and dual 16-bit instructions
  - Unpack instructions prepare 8-bit data for parallel 16-bit operations
  - Pack instructions return parallel results to output precision including saturation support
The Internal Bus Structure: SCR & Megamodule

C55x Low Power DSP Quick Compare
• Although not immediately obvious from the above figures, a distinctive feature of the C6x and C55 over conventional microprocessors, is the Harvard architecture, that is separate buses for program and data
  – Instructions can be fetched while data is being accessed

**C6748 Internal Memory**

- Level 1 Memory (32KB each)
  - Cache or RAM
  - L1P (prog), L1D (data)
- Level 2 Memory (256KB)
  - RAM (prog or data)
  - Up to 256KB can be cache
- Level 3 Memory (128KB)
  - Shared RAM

**Level 1 Memory**
- Single-cycle access
- L2 accessed on miss
- L1P: direct mapped
- L1D: 2-way set associative

**Level 2 Memory**
- Unified: Prog or Data
- 4-way cache support

**L1/L2 Shared Features**
- Configure each memory as cache or addressable RAM (or combination)
- Cache Freeze
C6748 External Memory

- EMIFA has four ranges (8MB each):
  - Program or Data
  - Named: ACE2, ACE3, ACE4, ACE5
- DDR2 is 512MB

Remaining memory is unused
Host Port

- A dedicated bus for connection to a micro or external host
- Bootloading can occur via HPI
System Architecture – SCR/EDMA

- SCR – Switched Central Resource
- Masters initiate accesses to/from slaves via the SCR
- Most Masters (requestors) and Slaves (resources) have their own port to the SCR
- Lower bandwidth masters (HPI, PCI66, etc) share a port
- There is a default priority (0 to 7) to SCR resources that can be modified.

Note: this picture is the “general idea”. Every device has a different scheme for SCRs and peripheral muxing. In other words “check your data sheet”.

"Masters" | "Slaves"
---|---
ARM | C64 Mem
DSP | DDR2
EDMA3 | EMIF64
TC0 | TCP
TC1 | VCP
TC2 | "Slaves"
CC | "Slaves"
PCI | PCI
HPI | McBSP
EMAC | Utopia
SCR | SCR
What is Pin Multiplexing?

- How many pins is on your device?
- How many pins would all your peripheral require?
- Pin Multiplexing is the answer – only so many peripherals can be used at the same time … in other words, to reduce costs, peripherals must share available pins
- Which ones can you use simultaneously?
  - Designers examine app use cases when deciding best muxing layout
  - Read datasheet for final authority on how pins are muxed
  - Graphical utility can assist with figuring out pin-muxing…

Laboratory Hardware Targets

- There are many C6x development systems or hardware targets available from third parties
- TI itself supplies:
– The OMAL-L138 eXperimenters Board $495
– The C6713-225 based DSK $395, with bundled DSK specific software, and USB host interface; 2M x 32 on board SDRAM, 512K bytes on board flash
– The C6416-600 based DSK $395, with bundled DSK specific software, and USB host interface
– The VC5505-100 based eZDSP USB Stick $49, with bundled CCS4 for XDS100 class JTAG emulators

• Full versions of the software tools are running in the lab
  – Full Code Composer Studio Platinum version 4.2
  – Full Code Composer Studio Platinum version 5.1
The OMAP-L138 (C6748) eXperimenters Board

The System on Module (SOM) Board

- Product-ready System on Module with a TI OMAP-L138 processor or TMS320C6748 DSP running at 375 MHz
Chapter 2 • The TMS320C6x Family: Hardware and Software

- The Card (Top View)

- OMAP-L138 SOM-M1 Block Diagram
• Audio Interface

OMAP L138 eXperimenters Board

• Communication Interfaces
• Emulation Interfaces

• Video and LCD Interfaces
• Memory Interfaces

• Boot and Power Management
## The C6748 Memory Map

### Table 2-4. C6748 Top Level Memory Map

<table>
<thead>
<tr>
<th>Start Address</th>
<th>End Address</th>
<th>Size</th>
<th>DSP Mem Map</th>
<th>EDMA Mem Map</th>
<th>PRUSS Mem Map</th>
<th>Master Peripheral Mem Map</th>
<th>LCDC Mem Map</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x0000 0000</td>
<td>0x0000 0FFF</td>
<td>4K</td>
<td></td>
<td></td>
<td>PRUSS Local Address Space</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x0070 0000</td>
<td>0x007F FFFF</td>
<td>1024K</td>
<td>DSP L2 ROM (1)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x0080 0000</td>
<td>0x0083 FFFF</td>
<td>256K</td>
<td>DSP L2 RAM</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x0090 0000</td>
<td>0x009F FFFF</td>
<td>512K</td>
<td>DSP L3 RAM</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x00A0 0000</td>
<td>0x00A3 FFFF</td>
<td>256K</td>
<td>DSP L1P RAM</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x00B0 0000</td>
<td>0x00B3 FFFF</td>
<td>256K</td>
<td>DSP L1D RAM</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x0180 0000</td>
<td>0x0180 FFFF</td>
<td>64K</td>
<td>DSP Interrupt Controller</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x0181 0000</td>
<td>0x0181 0FFF</td>
<td>4K</td>
<td>DSP Powerdown Controller</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x0181 1000</td>
<td>0x0181 1FFF</td>
<td>4K</td>
<td>DSP Security ID</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x0181 2000</td>
<td>0x0181 2FFF</td>
<td>4K</td>
<td>DSP Revision ID</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x0181 3000</td>
<td>0x0181 3FFF</td>
<td>52K</td>
<td>DSP Memory System</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x0184 0000</td>
<td>0x0184 FFFF</td>
<td>64K</td>
<td>DSP Memory System</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x0185 0000</td>
<td>0x0185 FFFF</td>
<td>64K</td>
<td>DSP Memory System</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x01C0 0000</td>
<td>0x01C0 7FFF</td>
<td>32K</td>
<td>EDMA3 CC</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x01C0 8000</td>
<td>0x01C0 83FF</td>
<td>1K</td>
<td>EDMA3 TC0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x01C0 8400</td>
<td>0x01C0 87FF</td>
<td>1K</td>
<td>EDMA3 TC1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x01C0 8800</td>
<td>0x01C0 FFFF</td>
<td>256K</td>
<td>DSP Memory System</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x01C1 0000</td>
<td>0x01C1 0FFF</td>
<td>4K</td>
<td>PSC 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x01C1 1000</td>
<td>0x01C1 1FFF</td>
<td>4K</td>
<td>PLL Controller 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x01C1 2000</td>
<td>0x01C1 2FFF</td>
<td>4K</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x01C1 4000</td>
<td>0x01C1 4FFF</td>
<td>4K</td>
<td>SYSCFG0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x01C1 5000</td>
<td>0x01C1 FFFF</td>
<td>4K</td>
<td>Timer0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x01C2 0000</td>
<td>0x01C2 0FFF</td>
<td>4K</td>
<td>Timer1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x01C2 2000</td>
<td>0x01C2 2FFF</td>
<td>4K</td>
<td>i2C 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x01C2 3000</td>
<td>0x01C2 3FFF</td>
<td>4K</td>
<td>RTC</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x01C2 4000</td>
<td>0x01C3 FFFF</td>
<td>4K</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x01C4 0000</td>
<td>0x01C4 0FFF</td>
<td>4K</td>
<td>MMC/SD 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x01C4 1000</td>
<td>0x01C4 1FFF</td>
<td>4K</td>
<td>SPI 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x01C4 2000</td>
<td>0x01C4 2FFF</td>
<td>4K</td>
<td>UART 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x01C4 3000</td>
<td>0x01C4 FFFF</td>
<td>4K</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x01D0 0000</td>
<td>0x01D0 0FFF</td>
<td>4K</td>
<td>McASP 0 Control</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x01D0 1000</td>
<td>0x01D0 1FFF</td>
<td>4K</td>
<td>McASP 0 AFIFO Ctrl</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x01D0 2000</td>
<td>0x01D0 2FFF</td>
<td>4K</td>
<td>McASP 0 Data</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x01D0 3000</td>
<td>0x01D0 BFFF</td>
<td>8K</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(1) The DSP L2 ROM is used for boot purposes and cannot be programmed with application code.
### Table 2-4. C6748 Top Level Memory Map (continued)

<table>
<thead>
<tr>
<th>Start Address</th>
<th>End Address</th>
<th>Size</th>
<th>DSP Mem Map</th>
<th>EDMA Mem Map</th>
<th>PRUSS Mem Map</th>
<th>Master Peripheral Mem Map</th>
<th>LCDC Mem Map</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x01D0 C000</td>
<td>0x01D0 CFFF</td>
<td>4K</td>
<td>UART 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x01D0 D000</td>
<td>0x01D0 DFFF</td>
<td>4K</td>
<td>UART 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x01D1 0000</td>
<td>0x01D1 07FF</td>
<td>2K</td>
<td>McBSP0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x01D1 0800</td>
<td>0x01D1 0FFF</td>
<td>2K</td>
<td>McBSP0 FIFO Ctrl</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x01D1 1000</td>
<td>0x01D1 17FF</td>
<td>2K</td>
<td>McBSP1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x01D1 1800</td>
<td>0x01D1 1FFF</td>
<td>2K</td>
<td>McBSP1 FIFO Ctrl</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x01D1 2000</td>
<td>0x01D1 FFFF</td>
<td>8K</td>
<td>SATA</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x01E0 0000</td>
<td>0x01E0 0FFF</td>
<td>4K</td>
<td>UPP</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x01E0 1000</td>
<td>0x01E0 0000</td>
<td>64K</td>
<td>USB0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x01E1 0000</td>
<td>0x01E1 2FF</td>
<td>8K</td>
<td>LCD Controller</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x01E1 3000</td>
<td>0x01E1 3FFF</td>
<td>4K</td>
<td>EMAC Control Module RAM</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x01E1 4000</td>
<td>0x01E1 4FFF</td>
<td>4K</td>
<td>Memory Protection Unit 1 (MPU 1)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x01E1 5000</td>
<td>0x01E1 5FFF</td>
<td>4K</td>
<td>Memory Protection Unit 2 (MPU 2)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x01E1 6000</td>
<td>0x01E1 6FFF</td>
<td>4K</td>
<td>VPIF</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x01E1 8000</td>
<td>0x01E1 9FFF</td>
<td>4K</td>
<td>SATA</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x01E1 A000</td>
<td>0x01E1 AFFF</td>
<td>4K</td>
<td>PLL Controller 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x01E1 B000</td>
<td>0x01E1 BFFF</td>
<td>4K</td>
<td>MMCSD1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x01E1 C000</td>
<td>0x01E1 CFFF</td>
<td>4K</td>
<td>EMAC Control Module Registers</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x01E1 D000</td>
<td>0x01E1 DFFF</td>
<td>4K</td>
<td>EMAC Control Registers</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x01E1 E000</td>
<td>0x01E1 EFFF</td>
<td>4K</td>
<td>EMAC MDIO port</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x01E2 0000</td>
<td>0x01E2 1FFF</td>
<td>8K</td>
<td>USB1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x01E2 2000</td>
<td>0x01E2 2FFF</td>
<td>4K</td>
<td>GPIO</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x01E2 3000</td>
<td>0x01E2 3FFF</td>
<td>4K</td>
<td>PSC 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x01E2 4000</td>
<td>0x01E2 4FFF</td>
<td>4K</td>
<td>I2C 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x01E2 6000</td>
<td>0x01E2 6FFF</td>
<td>4K</td>
<td>USB1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x01E2 7000</td>
<td>0x01E2 7FFF</td>
<td>4K</td>
<td>GPIO</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x01E2 8000</td>
<td>0x01E2 8FFF</td>
<td>4K</td>
<td>PSC 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x01E2 9000</td>
<td>0x01E2 BFFF</td>
<td>4K</td>
<td>SYSCFG1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x01E2 C000</td>
<td>0x01E2 CFFF</td>
<td>4K</td>
<td>SYSCFG1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x01E3 0000</td>
<td>0x01E3 7FFF</td>
<td>32K</td>
<td>EDMA3 CC1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x01E3 8000</td>
<td>0x01E3 8FFF</td>
<td>1K</td>
<td>EDMA3 TC2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x01E3 8400</td>
<td>0x01E3 FFFF</td>
<td>4K</td>
<td>eHRPWM 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x01F0 0000</td>
<td>0x01F0 0FFF</td>
<td>4K</td>
<td>HRPWM 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x01F0 1000</td>
<td>0x01F0 1FFF</td>
<td>4K</td>
<td>eHRPWM 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x01F0 2000</td>
<td>0x01F0 2FFF</td>
<td>4K</td>
<td>HRPWM 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x01F0 3000</td>
<td>0x01F0 3FFF</td>
<td>4K</td>
<td>ECAP 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x01F0 4000</td>
<td>0x01F0 5FFF</td>
<td>4K</td>
<td>ECAP 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x01F0 6000</td>
<td>0x01F0 6FFF</td>
<td>4K</td>
<td>ECAP 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x01F0 7000</td>
<td>0x01F0 7FFF</td>
<td>4K</td>
<td>Timer2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x01F0 8000</td>
<td>0x01F0 8FFF</td>
<td>4K</td>
<td>Timer3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x01F0 E000</td>
<td>0x01F0 EFFF</td>
<td>4K</td>
<td>SPI1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Table 2-4. C6748 Top Level Memory Map (continued)

<table>
<thead>
<tr>
<th>Start Address</th>
<th>End Address</th>
<th>Size</th>
<th>DSP Mem Map</th>
<th>EDMA Mem Map</th>
<th>PRUSS Mem Map</th>
<th>Master Peripheral Mem Map</th>
<th>LCDC Mem Map</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x01F0 F000</td>
<td>0x01F0 FFFF</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x01F1 0000</td>
<td>0x01F1 0FFF</td>
<td>4K</td>
<td>McBSP0 FIFO Data</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x01F1 1000</td>
<td>0x01F1 1FFF</td>
<td>4K</td>
<td>McBSP1 FIFO Data</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x1170 0000</td>
<td>0x117F FFFF</td>
<td>1024K</td>
<td>DSP L2 ROM (2)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x1180 0000</td>
<td>0x1183 FFFF</td>
<td>256K</td>
<td>DSP L2 RAM</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x1184 0000</td>
<td>0x11DF FFFF</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x11E0 0000</td>
<td>0x11E0 7FFF</td>
<td>32K</td>
<td>DSP L1P RAM</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x11E0 8000</td>
<td>0x11E0 7FFF</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x11F0 0000</td>
<td>0x11F0 7FFF</td>
<td>32K</td>
<td>DSP L1D RAM</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x11F0 8000</td>
<td>0x13FFF FFFF</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x4000 0000</td>
<td>0x5FFF FFFF</td>
<td>512M</td>
<td>EMIFA SDRAM data (CS0)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x6000 0000</td>
<td>0x61FF FFFF</td>
<td>32M</td>
<td>EMIFA async data (CS2)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x6200 0000</td>
<td>0x63FF FFFF</td>
<td>32M</td>
<td>EMIFA async data (CS3)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x6400 0000</td>
<td>0x65FF FFFF</td>
<td>32M</td>
<td>EMIFA async data (CS4)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x6600 0000</td>
<td>0x67FF FFFF</td>
<td>32M</td>
<td>EMIFA async data (CS5)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x8000 0000</td>
<td>0x8000 7FFF</td>
<td>32K</td>
<td>EMIFA Control Regs</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x8000 8000</td>
<td>0x77FFF FFFF</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x8000 0000</td>
<td>0x8001 FFFF</td>
<td>128K</td>
<td>On-chip RAM</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x8002 0000</td>
<td>0xAF00 FF00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0xB000 0000</td>
<td>0xB000 7FFF</td>
<td>32K</td>
<td>DDR2 Control Regs</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0xB000 8000</td>
<td>0xBFFF FFFF</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0xC000 0000</td>
<td>0xDFFFF FFFF</td>
<td>512M</td>
<td>DDR2 Data</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0xE000 0000</td>
<td>0x0FFFFFF</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(2) The DSP L2 ROM is used for boot purposes and cannot be programmed with application code.
The C6713 DSK

- The C6713 DSK was introduced summer 2003
- The 6713 is an enhanced version of the 6711, including
  - McBSP1 is used as a bi-directional data channel
  - McBSP0 is used as a unidirectional codec control channel in SPI format (operative normally only when first configuring the codec)
- The AIC codec uses a 12 MHz clock (popular USB clock rate)
- Through division common audio sample rate frequencies available are: 48 KHz, 44.1 KHz, and 8 KHz
C6713 DSK Board Layout

Figure 3-1, TMS320C6713 DSK

<table>
<thead>
<tr>
<th>Connector</th>
<th># Pins</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>J4</td>
<td>80</td>
<td>Memory</td>
</tr>
<tr>
<td>J3</td>
<td>80</td>
<td>Peripheral</td>
</tr>
<tr>
<td>J1</td>
<td>80</td>
<td>HPI</td>
</tr>
<tr>
<td>J301</td>
<td>3</td>
<td>Microphone</td>
</tr>
<tr>
<td>J303</td>
<td>3</td>
<td>Line In</td>
</tr>
<tr>
<td>J304</td>
<td>3</td>
<td>Line Out</td>
</tr>
<tr>
<td>J303</td>
<td>3</td>
<td>Headphone</td>
</tr>
<tr>
<td>J5</td>
<td>2</td>
<td>+5 Volt</td>
</tr>
<tr>
<td>J6 *</td>
<td>4</td>
<td>Optional Power Connector</td>
</tr>
<tr>
<td>J8</td>
<td>14</td>
<td>External JTAG</td>
</tr>
<tr>
<td>J201</td>
<td>5</td>
<td>USB Port</td>
</tr>
<tr>
<td>JP3</td>
<td>10</td>
<td>CPLD Programming</td>
</tr>
<tr>
<td>SW3</td>
<td>8</td>
<td>DSP Configuration Jumper</td>
</tr>
</tbody>
</table>
The C6416 DSK Block Diagram

- The C6416 DSK is very similar to the 6713 DSK, except the C6416 is a high performance fixed point device having:
  - 600 MHz CPU clock (1.67 ns cycle time yielding 4800 MIPS)
  - Only fixed-point hardware
  - Viterbi decoder co-processor for comm applications
  - Turbo decoder co-processor for comm applications (3GPP)
- The external memory and codec configurations are identical
Memory Mapping for the DSK’s

- Memory map details can be found in the TI documentation
- We are most interested in the memory addresses locations for storing programs and data on DSK
- The *linker command file, *.cmd* is used to handle the differences in memory mapping between the two platform, and in general across all TI DSP processors
Software Development Overview

Code Composer Studio (CCS) is the primary development environment on all of TI’s DSP platforms

- At a high level CCS consists of the following:

![Diagram of Code Composer Studio and Target hardware](image)

- The phases of code development in CCS can be viewed as follows:

<table>
<thead>
<tr>
<th>Design</th>
<th>Code &amp; build</th>
<th>Debug</th>
<th>Analyze</th>
</tr>
</thead>
<tbody>
<tr>
<td>conceptual planning</td>
<td>create project, write source code, configuration file</td>
<td>syntax checking, probe points, logging, etc.</td>
<td>real-time debugging, statistics, tracing</td>
</tr>
</tbody>
</table>

ECE 5655/4655 Real-Time DSP
Irrespective of the development environment employed, command-line or IDE, the generation of machine code in the form of an executable common object file format (coff) file follows a common flow.

**C6x Code Generation Overview**
Code Development Flow Chart

Phase 1
- Write C code
- Compile
- Profile
- Efficient?
  - Yes → Complete
  - No

Phase 2
- Refine C code
- Compile
- Profile
- Efficient?
  - Yes → Complete
  - No
  - More C optimizations?
    - Yes
    - No

Phase 3
- Write linear assembly
- Assembly optimize
- Profile
- Efficient?
  - Yes → Complete
  - No
DSP/BIOS

- Built-in instrumentation capability
  - `printf` stops the DSP to send a string back to the debugger
- BIOS can automatically log events back to CCS using `LOG_printf()`
- The DSP/BIOS API contains user specified functions to send event information back to CCS
- View real-time statistics that are passed back during non-critical times
- This capability comes about via *Real-Time Data Exchange* (RTDX)

**Traditional Start/Stop Data Transfer**

![Traditional Start/Stop Data Transfer Diagram]

**Continuous Run Data Transfer with RTDX**

![Continuous Run Data Transfer Diagram]
• DSP/BIOS can also be used to view system events
• All of this is controlled/defined via the CCS configuration tool, and the associated configuration file, which replaces the .cmd file
• How do we manage/synchronize real-world events?
• Hardware events are driven by hardware interrupts
• With BIOS hardware interrupts are transformed to software interrupts (SWI)
• Each SWI can be managed via prioritization in a scheduler that is part of the DSP/BIOS configuration tool

**DSP/BIOS Summary for Now**
A real-time kernel that consists of:

• real-time scheduling
• real-time capture
• real-time I/O
VC5505 eZDSP USB Stick

- Exposure to the new low-cost DSP platform from the fixed-point c55x family is also planned
- The board

Test points
<table>
<thead>
<tr>
<th>TP #</th>
<th>Schematic Page</th>
<th>Signal Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>TP1</td>
<td>3</td>
<td>CLKOUT, Pin A7, VC5505</td>
</tr>
<tr>
<td>TP2</td>
<td>3</td>
<td>RTC_CLKOUT, Pin D8, VC5505</td>
</tr>
<tr>
<td>TP3</td>
<td>2</td>
<td>GPAIN3, Pin C11, VC5505</td>
</tr>
<tr>
<td>TP4</td>
<td>2</td>
<td>GPAIN2, Pin B11, VC5505</td>
</tr>
<tr>
<td>TP5</td>
<td>2</td>
<td>GPAIN1, Pin A11, VC5505</td>
</tr>
<tr>
<td>TP6</td>
<td>2</td>
<td>GPAIN0, Pin D10, VC5505</td>
</tr>
<tr>
<td>TP7</td>
<td>3</td>
<td>GND</td>
</tr>
<tr>
<td>TP8</td>
<td>7</td>
<td>GND</td>
</tr>
<tr>
<td>TP9</td>
<td>5</td>
<td>GND</td>
</tr>
<tr>
<td>TP10</td>
<td>6</td>
<td>Vcore, VCC_1V3, Pin 7,8 U8</td>
</tr>
<tr>
<td>TP11</td>
<td>6</td>
<td>3V3, VCC_3V3, Pin 7,8 U7</td>
</tr>
<tr>
<td>TP12</td>
<td>4</td>
<td>GPIO22, Pin E2, VC5505</td>
</tr>
<tr>
<td>TP13</td>
<td>4</td>
<td>GPIO23, Pin F2, VC5505</td>
</tr>
<tr>
<td>TP14</td>
<td>4</td>
<td>GPIO24, Pin G2, VC5505</td>
</tr>
<tr>
<td>TP15</td>
<td>4</td>
<td>GPIO25, Pin G4, VC5505</td>
</tr>
<tr>
<td>TP16</td>
<td>4</td>
<td>GPIO21, Pin N1, VC5505</td>
</tr>
</tbody>
</table>

• Block diagram

---

**VC5505 eZDSP USB Stick**

ECE 5655/4655 Real-Time DSP

2–45
• Features

1.1 TMS320VC5505 Features

• High-Performance, Low-Power, TMS320C55x™ Fixed-Point Digital Signal Processor
  – 16.67-, 10-ns Instruction Cycle Time
  – 60-, 100-MHz Clock Rate
  – One/Two Instruction(s) Executed per Cycle
• Dual Multipliers [Up to 200 Million Multiply-Accumulates per Second (MMACS)]
• Two Arithmetic/Logic Units (ALUs)
• Three Internal Data/Operand Read Buses and Two Internal Data/Operand Write Buses
• Fully Software-Compatible With C55x Devices
• Tightly-Coupled FFT Hardware Accelerator (MMACS)
• 10-Bit 4-Input Successive Approximation (SAR) ADC
• LCD Bridge With Asynchronous Interface
• 320K Bytes Zero-Wait State On-Chip RAM, Composed of:
  – 64K Bytes of Dual-Access RAM (DARAM), 8 Blocks of 4K x 16-Bit
  – 256K Bytes of Single-Access RAM (SARAM), 32 Blocks of 4K x 16-Bit
• 128K Bytes of Zero Wait-State On-Chip ROM (4 Blocks of 16K x 16-Bit)
• 16-/8-Bit External Memory Interface (EMIF) with Glueless Interface to:
  – 8-/16-Bit NAND Flash, 1- and 4-Bit ECC
  – 8-/16-Bit NOR Flash
  – Asynchronous Static RAM (SRAM)
• Direct Memory Access (DMA) Controller
  – Four DMA With 4 Channels Each (16-Channels Total)
• Three 32-Bit General-Purpose Timers
  – One Selectable as a Watchdog and/or GP
• Two MultiMedia Card/Secure Digital (MMC/SD) Interfaces
• Universal Asynchronous Receiver/Transmitter (UART)
• Serial-Port Interface (SPI) With Four Chip-Selects
• Master/Slave Inter-Integrated Circuit (I²C Bus™)

• Four Inter-IC Sound (I²S Bus™) for Data Transport
• Device USB Port With Integrated 2.0 High-Speed PHY that Supports:
  – USB 2.0 Full- and High-Speed Device
• LCD Bridge With Asynchronous Interface
• Tightly-Coupled FFT Hardware Accelerator
• 10-Bit 4-Input Successive Approximation (SAR) ADC
• Real-Time Clock (RTC) With Crystal Input, With Separate Clock Domain, Separate Power Supply
• Four Core Isolated Power Supply Domains: Analog, RTC, CPU and Peripherals, and USB
• Four I/O Isolated Power Supply Domains: RTC I/O, EMIF I/O, USB PHY, and DVDDIO
• Low-Power S/W Programmable Phase-Locked Loop (PLL) Clock Generator
• On-Chip ROM Bootloader (RBL) to Boot From NAND Flash, NOR Flash, SPI EEPROM, or I²C EEPROM
• IEEE-1149.1 (JTAG™) Boundary-Scan-Compatible
• Up to 26 General-Purpose I/O (GPIO) Pins (Multiplexed With Other Device Functions)
• 196-Terminal Pb-Free Plastic BGA (Ball Grid Array) (ZCH Suffix)
• 1.05-V Core (60 MHz), 1.8-V, 2.5-V, 2.8-V, or 3.3-V I/Os
• 1.3-V Core (100 MHz), 1.8-V, 2.5-V, 2.8-V, or 3.3-V I/Os
• Applications:
  – Wireless Audio Devices (e.g., Headsets, Microphones, Speakerphones, etc.)
  – Echo Cancellation Headphones
  – Portable Medical Devices
  – Voice Applications
  – Industrial Controls
  – Fingerprint Biometrics
  – Software Defined Radio
### Memory Map

<table>
<thead>
<tr>
<th>CPU BYTE ADDRESS(A)</th>
<th>DMA/USB/LCD BYTE ADDRESS(A)</th>
<th>MEMORY BLOCKS</th>
<th>BLOCK SIZE</th>
</tr>
</thead>
<tbody>
<tr>
<td>000000h</td>
<td>0001 0000h</td>
<td>MMR (Reserved)(B)</td>
<td></td>
</tr>
<tr>
<td>0000C0h</td>
<td>0001 00C0h</td>
<td>DARAM(D)</td>
<td>64K Minus 192 Bytes</td>
</tr>
<tr>
<td>010000h</td>
<td>0009 0000h</td>
<td>SARAM</td>
<td>256K Bytes</td>
</tr>
<tr>
<td>050000h</td>
<td>0100 0000h</td>
<td>Reserved</td>
<td>8M Minus 320K Bytes</td>
</tr>
<tr>
<td>800000h</td>
<td>0200 0000h</td>
<td>External-CS2 Space(C)</td>
<td>4M Bytes Asynchronous</td>
</tr>
<tr>
<td>C00000h</td>
<td>0300 0000h</td>
<td>External-CS3 Space(C)</td>
<td>2M Bytes Asynchronous</td>
</tr>
<tr>
<td>E00000h</td>
<td>0400 0000h</td>
<td>External-CS4 Space(C)</td>
<td>1M Bytes Asynchronous</td>
</tr>
<tr>
<td>F00000h</td>
<td>0500 0000h</td>
<td>External-CS5 Space(C)</td>
<td>1M Minus 128K Bytes Asynchronous</td>
</tr>
<tr>
<td>FE0000h</td>
<td>050E 0000h</td>
<td>ROM</td>
<td>128K Bytes Asynchronous (if MPNMC=0)</td>
</tr>
<tr>
<td>FFFFFFFh</td>
<td>05FF FFFFh</td>
<td>External-CS5 Space(C) (if MPNMC=1)</td>
<td>128K Bytes ROM (if MPNMC=0)</td>
</tr>
</tbody>
</table>

A. Address shown represents the first byte address in each block.
B. The first 192 bytes are reserved for memory-mapped registers (MMRs).
C. Out of the four DMA controllers, only DMA controller 3 has access to the external memory space.
D. The USB and LCD controllers do not have access to DARAM.

- For more details on the C55 architecture in general see Section 4.4 of the Kuo text
- Since the C55 follows in the line of the C54, Section 4.3 of the Kuo text covers this family of processors