Ultra-Low Power TMS320C5515 DSP Overview

C5000 Ultra Low Power DSPs
Agenda

• C5515 Device Overview
• C5515 Architecture
• C5515 Peripherals
• C5515 Instructions
• 3<sup>rd</sup> Party Network
## Embedded processing portfolio

<table>
<thead>
<tr>
<th>Microcontrollers (MCUs)</th>
<th>ARM®-Based Processors</th>
<th>Digital Signal Processors (DSPs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>16-bit ultra-low power MCUs</td>
<td>32-bit real-time MCUs</td>
<td>C6000 DSP &amp; DSP+ARM</td>
</tr>
<tr>
<td>MSP430™</td>
<td>C2000™ Delfino™ Piccolo™</td>
<td>Multi core DSP</td>
</tr>
<tr>
<td>Up to 25 MHz</td>
<td>40MHz to 300 MHz</td>
<td>16-bit Ultra Low power DSPs</td>
</tr>
<tr>
<td>Flash 1 KB to 256 KB</td>
<td>Flash, RAM 16 KB to 512 KB</td>
<td></td>
</tr>
<tr>
<td>Analog I/O, ADC LCD, USB, RF</td>
<td>PWM, ADC, CAN, SPI, PC</td>
<td></td>
</tr>
<tr>
<td>$0.25 to $9.00</td>
<td>$1.50 to $20.00</td>
<td></td>
</tr>
</tbody>
</table>

### ARM®-Based Processors

<table>
<thead>
<tr>
<th>Processor</th>
<th>Clock Speed</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSP430™</td>
<td>Up to 25 MHz</td>
<td>Up to 500 MHz</td>
</tr>
<tr>
<td>Flash 1 KB to 256 KB</td>
<td>Flash, RAM 16 KB to 512 KB</td>
<td></td>
</tr>
<tr>
<td>Analog I/O, ADC LCD, USB, RF</td>
<td>PWM, ADC, CAN, SPI, PC</td>
<td></td>
</tr>
<tr>
<td>$0.25 to $9.00</td>
<td>$1.50 to $20.00</td>
<td></td>
</tr>
</tbody>
</table>

### Digital Signal Processors (DSPs)

<table>
<thead>
<tr>
<th>Processor</th>
<th>Clock Speed</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSP430™</td>
<td>Up to 25 MHz</td>
<td>Up to 500 MHz</td>
</tr>
<tr>
<td>Flash 1 KB to 256 KB</td>
<td>Flash, RAM 16 KB to 512 KB</td>
<td></td>
</tr>
<tr>
<td>Analog I/O, ADC LCD, USB, RF</td>
<td>PWM, ADC, CAN, SPI, PC</td>
<td></td>
</tr>
<tr>
<td>$0.25 to $9.00</td>
<td>$1.50 to $20.00</td>
<td></td>
</tr>
</tbody>
</table>

### Software & Dev. Tools

- Texas Instruments...

**MPUs – Microprocessors**
Ultra Low Power C5000 DSP Summary

• C5504/05/14/15 is the industry’s lowest power (< 0.15mW/MHz) DSP family
• Energy efficient audio/voice processing
• Completely C-programmable and no assembly required
• Comprehensive ecosystem to accelerate time to market
  — Fully functional development platforms including specific end equipment System Development Kits
  — Low cost eZdsp tools for simple application development
  — Production-quality software with code examples for specific application designs
  — Large and well established 3rd Party network
Four new C5000™ lowest power DSPs offer longer battery life at reduced system cost
Ultra Low power C5000™ DSP Roadmap
Enable innovation with industry’s lowest power

C55x family:
- 19 devices
- Latest generation industry’s lowest power 16-bit DSP family
- Performance up to 300MHz
- Power consumption <0.15mW/MHz

C54x family:
- 19 devices
- 1st generation low power 16-bit DSP
- Performance up to 160MHz

C5000
- 50 MHz
- USB 2.0

C5002
- 300 MHz

C5502
- 300 MHz

C5402
- 100/160 MHz

C5416
- 120/160 MHz

C5510
- 200 MHz

C5503/C5506/C5507/C5509
- 200 MHz

C5505
- 120 MHz
- USB 2.0
- FFT
- LCD

C5504/05
- 150 MHz

C5514/C5515
- 120 MHz
- USB 2.0
- FFT
- LCD
- LDOs

C55 Next
- Large on-chip memory
- FFT coprocessor
- USB2.0 with PHY integration
- LDO Integration

Value Line
- Very small package (QFP or BGA)

2009 2010 2011
Industry’s Lowest Power DSP

Best combination of standby and active power
Enables longer battery life for portable devices

- Standby mode <0.15mW
- Active mode < 0.15mW/MHz
- Dynamic Voltage & Frequency Scaling
- On-chip FFT hardware acceleration

High integration of peripherals and increased memory

Reduces system cost and enables intuitive user interface

- Up to 320KB memory
- Integrated LDOs, Power Mgmt
- High speed USB 2.0 w/PHY
- LCD controller and ADC
- MMC/SD, I2S, SPI, I2C
- RTC, Watchdog timer

Extensive development resources and pin-to-pin compatible devices

Provide customers with great design support and flexibility

C5515 Evaluation Module $395
eZdsp USB Stick $49
C5504/05/14/15 Value Proposition
C5504/05/14/5 Block Diagram and Deltas

**C5514/5 – additional features to C5504/5**

- **Core**
  - Dual MAC, C55x CPU with JTAG disable option
  - 1.05V @ 60/75MHz, 1.3V @ 100/120MHz, 1.4V @ 150MHz (C5504/05 only)
  - Dynamic Voltage and Frequency Scaling

- **Memory**
  - 256-KB On-Chip Memory: 64-KB DARAM, 192-KB SARAM
  - 128-KB ROM

- **Peripherals**
  - Low Power FFT HWA
  - Four serial busses offering combinations of I²S, UART, SPI, MMC/SD, and GPIO
  - High speed USB2.0
  - Multi-master and Slave I²C with 7 or 10-bit addressing modes
  - Three 32-bit timers with watchdog functionality
  - Four 4-Channel DMAs
  - Low power PLL (0.7mA) with 32KHz crystal oscillator
  - 16-bit EMIF with asynchronous SRAM, NAND (with 4-bit ECC) and SDRAM
  - Real-time clock with 32-KHz crystal input, separate power
  - Integrated LDOs for CPU and USB analog

**C5515 – additional Features to C5514**

- 320-KB On-Chip Memory: 64-KB DARAM, 256-KB SARAM
- 1024-point FFT Coprocessor
- Asynchronous LCD interface supporting LCD displays with memory-mapped interface - MPU68, MPU80, MPUXX, and Hitachi HD44780U
- 4ch 10-bit SAR ADC: Conversion rate 32 clock cycles @ 2MHz

Package: 196-pin 10x10mm BGA with 0.65mm pitch
- Separate I/O supplies for EMIF and serial interface

I/O: 1.8V, 2.5V, 2.8V, 3.3V

Texas Instruments
## C5504/05/14/15 DSP Feature Comparison

<table>
<thead>
<tr>
<th>Software Compatibility</th>
<th>Integrated Peripherals</th>
<th>Memory</th>
<th>CoProcessor</th>
<th>Integrated Power Mgmt</th>
</tr>
</thead>
<tbody>
<tr>
<td>VC5504</td>
<td>C55x DSP</td>
<td>256KB</td>
<td>FFT CoP</td>
<td>1 ANA LDO</td>
</tr>
<tr>
<td>VC5505</td>
<td>C55x DSP</td>
<td>320 KB</td>
<td>FFT CoP</td>
<td>1 ANA LDO</td>
</tr>
<tr>
<td>C5504A</td>
<td>C55x DSP</td>
<td>256KB</td>
<td>FFT CoP</td>
<td>1 ANA LDO</td>
</tr>
<tr>
<td>C5505A</td>
<td>C55x DSP</td>
<td>320 KB</td>
<td>FFT CoP</td>
<td>1 ANA LDO</td>
</tr>
<tr>
<td>C5514A</td>
<td>C55x DSP</td>
<td>256KB</td>
<td>FFT CoP</td>
<td>3 LDOs Power Mgmt</td>
</tr>
<tr>
<td>C5515A</td>
<td>C55x DSP</td>
<td>320KB</td>
<td>FFT CoP</td>
<td>3 LDOs Power Mgmt</td>
</tr>
</tbody>
</table>

- **Pin to Pin Compatible**
- **Rev “A”**
- **In Production**

Security Options available with secure ROM and secure bootloader

**Pin to Pin Compatibility:**
- VC5504 to VC5505
- C5504A to C5505A
- C5514A to C5515A

**Not recommended for new designs:**
- VC5504

**In Production:**
- VC5504, VC5505, C5504A, C5505A, C5514A, C5515A

**Integrated Power Mgmt:**
- 1 ANA LDO
- 3 LDOs Power Mgmt
C5515 Device Nomenclature

- **PREFIX**
  - TMX = Experimental device
  - TMS = Qualified device

- **DEVICE FAMILY**
  - 320 = TMS320™ DSP family

- **TECHNOLOGY**
  - C = Dual-supply CMOS

- **DEVICE**
  - C55x™ DSP: 5515, 5514

### Device Information

A. For actual device part numbers (P/Ns) and ordering information, see the TI website [http://www.ti.com](http://www.ti.com)

### Technical Details

- **DEVICE MAXIMUM OPERATING FREQUENCY**
  - 10 = 60 MHz at 1.05 V, 100 MHz at 1.3 V
  - 12 = 75 MHz at 1.05 V, 120 MHz at 1.3 V

- **TEMPERATURE RANGE**
  - Blank = −10°C to 70°C, Commercial Temperature
  - A = −40°C to 85°C, Industrial Temperature

- **PACKAGE TYPE**
  - ZCH = 196-pin plastic BGA, with Pb-Free soldered balls [Green]

- **SILICON REVISION**
  - Revision 2.0
C5504/C5505/14/15 Extensive Power Modes
Industry’s Lowest Power 16-bit DSP

• Active Power Mode
  – 75% DMAC + 25% ADD typical data
    • 0.15mW/MHz @ 1.05V, 0.22mW/MHz @1.3V

• Standby Power Modes
  – With SARAM retention mode
    • 0.15mW @ 1.05V, 0.28mW @ 1.3V
  – With DARAM retention mode
    • 0.23mW @ 1.05V, 0.40mW @ 1.3V
  – With SARAM and DARAM active mode
    • 0.26mW @ 1.05V, 0.44mW @ 1.3V

• “RTC only” Low Power mode
  – New in C5504/05/14/15 (Rev A silicon)
TMS320C55xx power numbers comparison
Based on 75% DMAC + 25% ADD Test Case

- **Device**
  - Production
  - Sampling
  - In Development
  - Future

- **Power Consumption**
  - C5509A, C5507, C5506, C5503
    - 200 MHz @ 1.6V
    - 0.96mW/MHz
  - C5509A, C5507, C5503
    - 144 MHz @ 1.35V
    - 0.69mW/MHz
  - C5510A
    - 200 MHz @ 1.6V
    - 0.9mW/MHz
  - C5501/02
    - 300 MHz @ 1.26V
    - 1mW/MHz

- **Performance**
  - C55X4/5
    - 120 MHz @ 1.3V
    - 0.22mW/MHz
  - C55X4/5
    - 60 MHz @ 1.05V
    - 0.15mW/MHz

- **Future**
  - Device Production
  - Sampling
  - In Development
  - Future

- **~4x improvement in power consumption**

- *Based on DSP PLL at 100 MHz; DSP at 1.3 CVdd; Room Temp (25 °C), CLOCK_OUT disabled, 75% DMAC + 25% ADD (high data switching)
## Energy efficiency of FFT HWA

<table>
<thead>
<tr>
<th>Complex FFT</th>
<th>FFT with HWA</th>
<th>CPU (Scale)</th>
<th>HWA VS. CPU</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FFT + BR Cycles</td>
<td>Energy/FFT (nJ/FFT)</td>
<td>FFT + BR Cycles</td>
</tr>
<tr>
<td>8 pt</td>
<td>92+38 =130</td>
<td>23.6</td>
<td>196+95 =291</td>
</tr>
<tr>
<td>16 pt</td>
<td>115+55 =170</td>
<td>32.1</td>
<td>344+117=461</td>
</tr>
<tr>
<td>32 pt</td>
<td>234 +87 =321</td>
<td>69.5</td>
<td>609+139=748</td>
</tr>
<tr>
<td>64 pt</td>
<td>285+151 =436</td>
<td>98.5</td>
<td>1194+211=1405</td>
</tr>
<tr>
<td>128 pt</td>
<td>633+279 =912</td>
<td>219.2</td>
<td>2499+299=2798</td>
</tr>
<tr>
<td>256 pt</td>
<td>1133+535=1668</td>
<td>407.2</td>
<td>5404+543=5947</td>
</tr>
<tr>
<td>512 pt</td>
<td>2693+1047=3740</td>
<td>939.7</td>
<td>11829+907=12736</td>
</tr>
<tr>
<td>1024 pt</td>
<td>5244+2071=7315</td>
<td>1836.2</td>
<td>25934+1783=27717</td>
</tr>
</tbody>
</table>

**Note:**
1. BR = Bit-Reverse Operation
2. Power measurement Condition: at room temp only, all peripherals are clock gated, measured at Vddc

**FFT HWA is 4 ~ 6x more energy efficient and 2.2 ~ 3.8x faster**
C5504A/05A improvements from VC5504/05 – 1/2

• Power up sequencing (Core first and then I/O) is not required in the C5504A/C5505A.

• SAR reset bit (bit2) of the PRCR register(0x1C05) has been added.

• “Divide-by-4” added to the PLL output divider.

• mSDRAM Support
  – Available in C5504A/C5505A, was not available in VC5504/VC5505.

• DMA Double Buffering Capability
  – Not possible to support double buffering (ping-pong buffer) in VC5504/VC5505. DMA needs to be configured by CPU at every new DMA. This could cause data loss at high transfer data rate.
  – A ping-pong buffer has been added in C5504A/C5505A.
C5504A/05A improvements from VC5504/05 – 2/2

• Word/Byte swap issue (Advisory 1.4.1)
  – DPORT word swap removed
  – Endianess of EMIF changed to big endian (hardcoded)
  – Endianess of MMC/SD changed to big endian (software controllable)
  – Endianess of USB DMA changed to big endian (hardcoded)

• RTC positive compensation did not work for compensation values that are multiples of 10 (Advisory 1.4.2). This has been fixed in C5504A/C5505A.

• Invalid I2S OUERRFL Error Report at First Frame has been fixed (Advisory 1.4.3).

• DMA in H/W Sync Mode, Auto Reload Bit Overrides Enable Bit has been fixed (Advisory 1.4.6).

• DMA: Hardware Event can trigger DMA data transfer in S/W control mode has been fixed (Advisory 1.4.7).

Note: The advisories mentioned above are with reference to the VC5505/04 Errata (SPRZ281A).
Tools, Software and Support
Get to Market Fast with Best-in-Class Tools and Development Platforms

**Code Composer Studio™ IDE**
- Allows designers of all experience levels to move quickly through application development
  - Design
  - Code and build
  - Debug
  - Analyze
  - Tune
  - Eclipse based

**Evaluation Modules**
- **Simple Application Development Board**
  - eZDSP5505 $49
  - eZDSP5515 $79
- **Fully Featured EVM**
  - C5515 EVM $395

**System Development Kits**
- **Free CCS v4 with XDS100 emulator**
- Solving the Problem from the Sensors
  - Medical Development Kit
    - ECG
    - Digital Stethoscope
    - Pulse Oximeter
  - Fingerprint Biometrics Development Kit $79

Extensive Support: Wiki, E2E Forums, On-line community
- [http://e2e.ti.com/support/dsp/tms320c5000_power-efficient_dsps/default.aspx](http://e2e.ti.com/support/dsp/tms320c5000_power-efficient_dsps/default.aspx)
Begin development today with C5515 EVM

**Hardware**
- TMS320C5515 fixed point low power DSP
- TLV320AIC3204 Stereo codec
- Integrated NAND/NOR/mSDRAM
- Real-time power monitoring circuit
- OLED color LCD display (128x128 pixels)
- I2C and SPI EEPROMs
- 10 user defined push button switches
- Embedded JTAG emulation via mini USB interface and external JTAG emulation interface
- Battery Holder (For 2 AAA, not included)

**Software**
- Chip Support Library
- Code Composer Studio IDE™ Rev 3.3

**Documentation**
- Quick Start Installation Guide
- Schematics

**Connectivity**
- Analog front end connectors
- USB 2.0 slave port high speed
- SD/MMC, I2C, SPI, I2S, UART
- Two expansion connectors
- Stereo line in (2) /out (1), headphone out (1) and microphone in (L/R)
- CE-ATA connector
- External oscillator socket
- BT/Chipcon connector

**Evaluation capability of C55X4/5 low power DSP**
- Several memory options
- **Embedded emulation on board though A-mini B USB cable to save cost**
- Common platform for ECG, Digital Stethoscope & Pulse Oximeter Medical Development Kits

**Order Information**
- Part Number: TMDXEV5515
- Price: $395
- Available now

Blue: Improvements in TMDXEV5515 compared to TMDXEV5505
eZdsp5505 USB stick development tool

- TMS320VC5505 DSP
- On-board emulation
- Integrated audio codec
- Audio line out/line in connectors
- Extension connector (UART/SPI/I2S/I2C/GPIO)
- Simple form factor plugs into any USB host port
- Simplifies development tools setup by eliminating power and interface cables
Begin development today with eZdsp tool

**Hardware**
- TMS320VC5505 fixed point low power DSP
- TLV320AIC3204 stereo codec
- I\(^2\)C EEPROM (64kB)
- Embedded XDS100 JTAG emulation
- Expansion connector

**Software**
- Chip support library
- Code Composer Studio IDE™ rev 4.0

**Documentation**
- Quick Start Installation Guide
- Schematics

**Evaluation capability of VC5504/05 low power DSP**
- Embedded XDS100 emulation
- Online community

**Community**
- Online community
- Code examples with software
- Promotions for example contribution
  - code.google.com/p/c5505-ezdsp

**Sample demos**
- Audio tone gen
- Digital Filtering
- Music/Audio special Efx

**Order information**
- Part Number: TMDX5505ezdsp
- Price: $49
- Order entry open: NOW
C5515 eZdsp USB Stick Development Tool

- TMS320C5515 DSP
- On-board emulation
- Integrated audio codec
- Audio line out/line in connectors
- USB 2.0 Slave Port
- Micro-SD Slot
- 4 MByte NOR Flash
- Extension connectors (UART/SPI/I2S/I2C/GPIO/SD)
- Simple form factor plugs into any USB host port
- Simplifies development tools setup by eliminating power and interface cables
- $79
• Benefits of CSL
  – Peripherals ease of use
  – Shortened development time
  – Portability
  – Hardware abstraction
  – Standardization and compatibility among devices

• Two Layers of Abstraction
  – CSL Register Level
  – CSL Function Level

• Reference for Customer Driver Development
  – All source code for CSL is open to customers
  – Most of CSL is written in C

• CCS 3.3 and CCS 4.0 Compatible
  – Examples are provided for CCS4.x and CCS3.3

C5505/04/15/14 CSL Architecture

USER TASK 1

USER TASK 2

CSL API

<table>
<thead>
<tr>
<th>DMA</th>
<th>RTC</th>
<th>SPI</th>
<th>...</th>
<th>USB</th>
</tr>
</thead>
<tbody>
<tr>
<td>DMAr</td>
<td>RTCr</td>
<td>SPIr</td>
<td>...</td>
<td>USBr</td>
</tr>
</tbody>
</table>

C5505/04/15/14 Hardware

Function Level
Configuration Functions
Csl_file

Register Level
Register & Bit Definitions
Csir_file

TEXAS INSTRUMENTS
What is included in C5505/15 CSL

• Release Notes

• Peripheral Drivers with Source Code
  – DAT, DMA, EMIF, GPIO, GPT, I2C, I2S, INTC, LCD, MMC/SD, NAND, PLL, RTC, SAR, SPI, UART, USB, WDT

• C5505/15 CSL Pre-built library and header files

• APIs documentation

• Example Code for all peripherals including Project Files for CCS 4.x and CCS3.3

• Example Application Code Using CSL
  – USB Mass Storage Class on SD card
  – USB Audio Class
  – FAT16/32 on SD card
Online Community

- Where can I get support?

- C5515 Product Page
  - DSP Datasheets, User’s Guides, App Notes, Simulation Models, Videos
  - http://focus.ti.com/docs/prod/folders/print/tms320c5515.html

- Spectrum Digital C5515 Support Site
  - http://support.spectrumdigital.com

- TI E2E™ Forums
  - Browse, Search, Subscribe to existing posts
  - Free support from engineers and expert users
  - http://e2e.ti.com/support/dsp/tms320c5000_power-efficient_dspds/default.aspx

- TI Embedded Processors Wiki
  - Growing documentation, User contribution
C5505/15 Architecture
C54x vs C55x Architecture

16-bit data busses

B Bus (Coefficient)

C Bus

D Bus

D Registers Interconnect

ACC Buss

MAC-0

MAC-1

40-Bit ALU

16-Bit ALU

Shifter

3 Address Generators

C

Y

X

5 Address Busses

24-bit

Resources on C54x

Added to C55x

BAB

DAB

EAB

FAB

XAR0

AR0

XAR1

AR1

XAR2

AR2

XAR3

AR3

XAR4

AR4

XAR5

AR5

XAR6

AR6

XAR7

AR7

XCDP

CDP

XDP

-
C55x Core Rev 2 Architectural Overview

Data Read busses BB (16-bits), CB (16-bits), DB (16-bits)

Data Read Address busses BAB, CAB, DAB (3x24)

Program Address bus PAB (24)

Data Write Address busses EAB, FAB (2x24)

Data Write busses EB, FB (2x16)
**C55x Core Rev 3 Improvements (C5505/04/15/14)**

Data Read busses BB (32-bits), CB (16-bits), DB (16-bits)

Data Read Address busses BAB, CAB, DAB (3x24)

Program Address bus PAB (24)

**Program Flow**

- Instruction Buffer Queue (128x8 bit)
- Instruction Decoder Controller
  - 1st instruction
  - 2nd instruction
- Program Counter
- Instruction Decoder
- Program Address Gen
- Status Registers
- Program Flow
- Pipeline Protection Unit
- Interrupts
- Prog Address Gen
- RET A
- 64 bits

**Data Flow**

- Auxiliary Registers[0:7]
- Data Registers [0:3]
- Mac
- Coefficient Data Pointer
- Smem/Xmem 23-bit
- Ymem 23-bit
- Cmem 23-bit
- ALU 16-bit
- Transition Regs.
- Bit Operations
- 40 bit ALU
- shifter

Data Write Address busses EAB, FAB (2x24)

Data Write busses FB, FB (2x16)
C55x Rev3 Instruction Buffer Unit

- **Improved Code Density**
  - Automatically scalable instruction word - 8, 16, 24, 32, 40, and 64-bit instruction formats
  - Predicated instructions
  - Program memory byte addressable
  - Protected pipeline

- **Lower Power**
  - Explicit & "Soft-Dual" 48, 56, and 64-bit parallel instructions
  - Program fetched in packets of 32-bits from memory
  - Inner loops buffered, pre-decode
  - Support global and local repeat block

- **Higher Performance**
  - Independent 4 byte packet fetched every cycle (if zero wait states)
  - Determines parallelism and dispatches execution to three processing units
  - Two instructions in parallel can be executed per clock cycle
  - Predictive branching intelligence enables faster processing

---

Program Read Bus (PB) (32)

Instruction Buffer Queue (128x8 bit)

Instruction Decoder Controller

1st instruction

2nd instruction

PU AU DU

Texas Instruments
**C55x Program Flow Unit**

- **Increased Efficiency**
  - Program memory byte addressable
  - Interruptible block and single repeats (2 block, 1 single)
  - Conditional execution instructions (except goto, switch, trap, call, repeat, intr, return, blockrepeat, and reset.)
  - Reduction in Programming Complexity
  - Protected pipeline

- **Higher Performance**
  - 1 x 32-bit, 4 x 16-bit data busses (3R, 2W)
  - 6 x 24 bit address busses
  - 1 x 32 bit program bus

---

- **Data-read data buses CB, DB (16 bits of data each)**
- **Program-read address bus PAB (24 bit address each)**
- **Program Counter**
- **LCRPC**
- **Prog Address Gen**
- **Status Registers**
- **Program Flow**
- **Pipeline Protection Unit**
- **Interrupts**

- **Data-write data buses EB, FB (16 bits of data each)**
C55x Address Data Flow Unit

- **Data-read address buses (3)** (23-bit address each)
- **Data-read data buses (3)** (16-bits of data each)
- **Data-write address buses (2)** (23-bit address each)
- **Data-write data buses** EB, FB (16 bits of data each)

- **8 23-bit Address Registers**
- **4 16-bit Data Registers**
- **23-bit Coefficient Data Pointer**
- **Smem/Xmem 23-bit**
- **Ymem 23-bit**
- **Cmem 23-bit**
- **ALU 16-bit**

**Lower Power**
- One generic 16-bit ALU with shifting capabilities allows simpler arithmetic operations to be performed outside the multiply unit.

**Increased Efficiency**
- No alignment constraint on data page and circular buffers
- 8 23-bit address registers (AR[7:0])
- 4 generic temporary registers (T[3:0]) dedicated address generation arithmetic units
- 3 independent circular buffers
- Interruptible block and single repeats (2 block, 1 single)
- Conditional execution instructions
C55x Data Computation Unit

Data-read data buses or operand from data registers

16-bit → 32-bit → 16-bit

MAC

AC0  AC1  AC2  AC3

40-bit ALU

shifter

Transition Regs.

Bit Operations

Data-write data buses EB, FB (16 bits of data)

- **Lower Power**
  - Parallelism minimizes cycle count per task
  - Program memory byte addressable
  - Protected pipeline

- **Higher Performance**
  - 32-bit B Bus provides fourth independent value for dual MAC and other instructions
  - Four 40-bit generic (Accumulator) Registers
  - Single cycle 17x17 dual MAC hardware
  - 40-bit ALU, splitable to 2 x “16-bit” ALU
  - Rounding option for multiply and store instructions (2 rounding modes)
  - Barrel shifter (extended range -32/+31)
  - Saturation range control

Texas Instruments
## C55x Pipeline

<table>
<thead>
<tr>
<th>Pab</th>
<th>Ram</th>
<th>pb</th>
<th>Ftc</th>
<th>Dec</th>
<th>Adr</th>
<th>Ac1</th>
<th>Ac2</th>
<th>Read</th>
<th>Ex</th>
<th>Write</th>
</tr>
</thead>
</table>

**Pipeline Stage description:**

- **Fetch**: Read four bytes from program memory via PB, and load the four bytes into the instruction buffer queue.
  
  This prefetch is an independent operation from the execution pipeline.

- **1: Decode**: Read up to six bytes from the instruction buffer queue.
  
  Decode an instruction pair or a single instruction.
  
  Dispatch instructions to the program flow unit (P unit),
  
  the address-data flow unit (A unit), and the data computation unit (D unit).

- **2: Address**: Compute data-space address(es) in the data-address generation unit (DAGEN).
  
  Modify pointers and repeat counters as required.
  
  Compute the program-space addresses for PC-relative branching instructions.

- **3: Access 1**: Send addresses for read operands on BAB, CAB, and DAB.

- **4: Access 2**: One cycle for memory access

- **5: Read**: Transfer an operand or operands to the CPU via CB (Ymem operand),
  
  DB (Smem or Xmem operand), CB and DB (Lmem operand), and BB (Cmem operand).
  
  Generate address(es) for operand write(s) and send them on EAB and FAB.
  
  Evaluate conditional operators

- **6: Execute**: Execute data processing instructions that are executed in the A unit and the D unit.
  
  Store result of computation into registers.

- **7: Write**: Data sent to memory / memory mapped accesses
### Memory Map

<table>
<thead>
<tr>
<th>Start Byte Address (Hex)</th>
<th>Start Word Address (Hex)</th>
<th>Memory Bank</th>
<th>Size in Bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>000000h</td>
<td>000000h</td>
<td>DARAM0</td>
<td>192 bytes</td>
</tr>
<tr>
<td>000020h</td>
<td>000060h</td>
<td>DARAM1</td>
<td>8K bytes</td>
</tr>
<tr>
<td>000040h</td>
<td>0000A0h</td>
<td>DARAM2</td>
<td>8K bytes</td>
</tr>
<tr>
<td>000060h</td>
<td>0000E0h</td>
<td>DARAM3</td>
<td>8K bytes</td>
</tr>
<tr>
<td>000080h</td>
<td>000120h</td>
<td>DARAM4</td>
<td>8K bytes</td>
</tr>
<tr>
<td>0000A0h</td>
<td>000160h</td>
<td>DARAM5</td>
<td>8K bytes</td>
</tr>
<tr>
<td>0000C0h</td>
<td>0001A0h</td>
<td>DARAM6</td>
<td>8K bytes</td>
</tr>
<tr>
<td>0000E0h</td>
<td>000200h</td>
<td>DARAM7</td>
<td>8K bytes</td>
</tr>
<tr>
<td>000200h</td>
<td>000240h</td>
<td>SARAM0</td>
<td>8K bytes</td>
</tr>
<tr>
<td>000220h</td>
<td>000260h</td>
<td>SARAM1</td>
<td>8K bytes</td>
</tr>
<tr>
<td>000240h</td>
<td>000280h</td>
<td>SARAM2</td>
<td>8K bytes</td>
</tr>
<tr>
<td>000260h</td>
<td>0002A0h</td>
<td>SARAM3</td>
<td>8K bytes</td>
</tr>
<tr>
<td>000280h</td>
<td>0002C0h</td>
<td>SARAM4</td>
<td>8K bytes</td>
</tr>
<tr>
<td>0002A0h</td>
<td>000300h</td>
<td>SARAM5</td>
<td>8K bytes</td>
</tr>
<tr>
<td>0002C0h</td>
<td>000340h</td>
<td>SARAM6</td>
<td>8K bytes</td>
</tr>
<tr>
<td>000300h</td>
<td>000380h</td>
<td>SARAM7</td>
<td>8K bytes</td>
</tr>
<tr>
<td>000320h</td>
<td>0003C0h</td>
<td>SARAM8</td>
<td>8K bytes</td>
</tr>
<tr>
<td>000340h</td>
<td>000400h</td>
<td>SARAM9</td>
<td>8K bytes</td>
</tr>
<tr>
<td>0003C0h</td>
<td>000440h</td>
<td>SARAM10</td>
<td>8K bytes</td>
</tr>
<tr>
<td>000400h</td>
<td>000480h</td>
<td>SARAM11</td>
<td>8K bytes</td>
</tr>
<tr>
<td>000420h</td>
<td>0004A0h</td>
<td>SARAM12</td>
<td>8K bytes</td>
</tr>
<tr>
<td>000440h</td>
<td>0004E0h</td>
<td>SARAM13</td>
<td>8K bytes</td>
</tr>
<tr>
<td>000460h</td>
<td>000500h</td>
<td>SARAM14</td>
<td>8K bytes</td>
</tr>
<tr>
<td>000480h</td>
<td>000520h</td>
<td>SARAM15</td>
<td>8K bytes</td>
</tr>
<tr>
<td>000500h</td>
<td>000540h</td>
<td>SARAM16</td>
<td>8K bytes</td>
</tr>
<tr>
<td>000520h</td>
<td>000560h</td>
<td>SARAM17</td>
<td>8K bytes</td>
</tr>
<tr>
<td>000540h</td>
<td>000580h</td>
<td>SARAM18</td>
<td>8K bytes</td>
</tr>
<tr>
<td>000560h</td>
<td>0005C0h</td>
<td>SARAM19</td>
<td>8K bytes</td>
</tr>
<tr>
<td>000580h</td>
<td>000600h</td>
<td>SARAM20</td>
<td>8K bytes</td>
</tr>
<tr>
<td>0005C0h</td>
<td>000640h</td>
<td>SARAM21</td>
<td>8K bytes</td>
</tr>
<tr>
<td>000600h</td>
<td>000680h</td>
<td>SARAM22</td>
<td>8K bytes</td>
</tr>
<tr>
<td>000640h</td>
<td>0006C0h</td>
<td>SARAM23</td>
<td>8K bytes</td>
</tr>
<tr>
<td>000680h</td>
<td>000700h</td>
<td>SARAM24</td>
<td>8K bytes</td>
</tr>
<tr>
<td>0006C0h</td>
<td>000780h</td>
<td>SARAM25</td>
<td>8K bytes</td>
</tr>
<tr>
<td>000700h</td>
<td>0007C0h</td>
<td>SARAM26</td>
<td>8K bytes</td>
</tr>
<tr>
<td>000720h</td>
<td>000800h</td>
<td>SARAM27</td>
<td>8K bytes</td>
</tr>
<tr>
<td>000780h</td>
<td>000840h</td>
<td>SARAM28</td>
<td>8K bytes</td>
</tr>
<tr>
<td>0007C0h</td>
<td>000880h</td>
<td>SARAM29</td>
<td>8K bytes</td>
</tr>
<tr>
<td>000800h</td>
<td>0008C0h</td>
<td>SARAM30</td>
<td>8K bytes</td>
</tr>
<tr>
<td>000840h</td>
<td>000900h</td>
<td>SARAM31</td>
<td>8K bytes</td>
</tr>
</tbody>
</table>

### External SDRAM Memory

- **CS0**: 050000h - 050008h, **7.6875M bytes**
- **CS2**: 080000h - 080007h, **4M bytes**
- **CS3**: 0C0000h - 0C0007h, **2M bytes**
- **CS4**: E00000h - E00007h, **1M bytes**
- **CS5**: F00000h - F00007h, **896K bytes**

### External Asynchronous Memory

- **CS2**: FE0000h - FE0007h, **32K bytes**
- **CS3**: FE8000h - FE8007h, **32K bytes**
- **CS4**: FF0000h - FF0007h, **32K bytes**
- **CS5**: FF8000h - FF8007h, **32K bytes**
Parallel Execution: Operators

• Add the following instruction pairs to add more computational power and flexibility with the D-Unit.
  - ALU || MAC
  - SHIFT || MAC
  - ALU || SHIFT
  - MAC || MAC

• Relax soft dual parallelism rules in the DAGEN. Allow:
  - Smem_R || Lmem_W
  - Lmem_R || Smem_W
  - Smem_RW || Smem_R
  - Smem_RW || Smem_W

<table>
<thead>
<tr>
<th></th>
<th>A-unit ALU</th>
<th>A-unit Swap</th>
<th>A-unit Load</th>
<th>A-unit Store</th>
<th>D-unit ALU</th>
<th>D-unit Shifter</th>
<th>D-unit MAC</th>
<th>D-unit Load</th>
<th>D-unit Store</th>
<th>D-unit Shift, Store</th>
<th>D-unit Swap</th>
<th>P-unit Control</th>
<th>P-unit Load</th>
<th>P-unit Store</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Add the following instruction pairs to add more computational power and flexibility with the D-Unit.
- Relax soft dual parallelism rules in the DAGEN. Allow:
Parallel Execution: Busses

Bus connections to registers/operators in three main units

CPU Operators and Buses

Note: C55x Rev. 3 has KAB2 & KDB2 buses for transport of constants to A & D from I ⇒ KAB1∥KAB2 & KDB1∥KDB2
## C55x Rev 3.0 Speed-up in Typical Algorithms

<table>
<thead>
<tr>
<th></th>
<th>32-bit B bus</th>
<th>parallel store</th>
<th>64-bit dispatch</th>
<th>C55x Benchmark rev 2.0</th>
<th>C55x Benchmark rev 3.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>16-bit complex vector mpy</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>3N</td>
<td>2N</td>
</tr>
<tr>
<td>32-bit extended precision vector mpy</td>
<td></td>
<td>✓</td>
<td></td>
<td>4N</td>
<td>3N</td>
</tr>
<tr>
<td>16-bit real vector mpy</td>
<td></td>
<td>✓</td>
<td></td>
<td>1.5N</td>
<td>1N</td>
</tr>
<tr>
<td>16-bit real vector dot product</td>
<td>✓</td>
<td></td>
<td></td>
<td>1N</td>
<td>0.5N</td>
</tr>
</tbody>
</table>
C5505/15 Peripherals
Power Management

- **3 LDOs:**
  - Core LDO (1.05 or 1.3V) with min shutdown current of 180mA
  - POR and SAR LDO (1.3V) with min shutdown current of 4mA
  - USB Analog (1.3V) LDO with min shutdown current of 18mA

- DSP Core LDO can be shutdown by software.

- DSP Core LDO can be enabled by RTC alarm or external WAKEUP pin.
C5505 Parallel Interface

C5505 External Interface:
- **Asynchronous:**
  - 21-bit address with A[20:13] individually selectable as GPIO. Each GPIO supports interrupt capabilities and pull-downs
  - 16-bit data
  - 2 Chip Selects
  - Programmable cycle timing for each chip select
- **NAND:**
  - 8 and 16-bit data bus width
  - 1-bit ECC for 8-bit NAND Flash
  - 4-bit ECC for 8 and 16-bit NAND Flash
  - 4 NAND Chip Selects
  - Programmable cycle timing for each chip select
- **Mobile SDRAM (mSDR):**
  - Supports 2 and 3 CAS Latencies
  - Supports 1, 2, and 4 internal banks
  - Supports 256, 512, 1024, and 2048-word page sizes
  - Supports 4 and 8 burst lengths
C5505 Serial Busses

- **Two Serial Interface Busses:**
  - **Serial 0:**
    - 6 pins of MMC/SD
    - 4 pins of I2S and 2 GPIO
    - 6 pins GPIO
  - **Serial 1:**
    - 6 pins of MMC/SD
    - 4 pins of I2S and 2 GPIO
    - 6 pins GPIO

- **I2S**
  - Bidirectional clock and frame sync. One output and one input data lines.
  - 8, 10, 12, 16, 20, 24, or 32-bit MSB-first, left justified transfers
  - Programmable clock and frame sync polarities
  - Overrun and underrun error detection
  - Loopback mode
  - Data packing modes into a 32-bit word

- **MMC/SD**
  - Supports a Multimedia Card (MMC), a Secure Digital Memory Card (SD), or a Secure Digital I/O Card (SDIO)
  - The capability to use either the MMC/SD protocol or the SPI protocol
  - Programmable frequency for the operation of the MMC/SD controller
  - Programmable frequency for the clock that controls the timing of transfers between the MMC/SD controller and the memory card.
C5505 Serial Busses (cont.)

- **LCD Bridge:**
  - Supports single panel mode display with size from 16 to 1024 pixels, total video frame up to 1024 x 1024
  - Supports passive (STN) with 16, 256, or 3375 colors or 15 grayscale level for monochrome screen
  - Supports active (TFT) with up to 4096 colors active
  - Supports up to 16-bits per pixel
  - If less than 8-bits per pixel, then the other 8-bits are available for UART, I2S, SPI, and GPIO.

- **SPI:**
  - Programmable data size: 1-32 bits
  - Programmable frame length: 1-4096 words
  - Programmable CS to data delay: 0-3 clocks
  - Programmable clock and CS polarities
  - Frame and/or word interrupts
  - Loopback mode

- **UART:**
  - 16-bit FIFO, 1, 4, 8, or 16 byte selectable receiver FIFO trigger level for autow flow control and DMA
  - Programmable auto-rts and auto-cts for autow flow control
  - Frequency pre-scale values from 0 to 65,535 to generate appropriate baud rates
  - 5, 6, 7, or 8-bit characters
  - Even, odd, or no parity bit generation and detection
  - 1, 1.5, or 2 stop bit generation
  - False start bit detection
  - Line break generation and detection
  - Loopback controls for communications link fault isolation
  - Break, parity, overrun, and framing error simulation
  - Modem control functions (CTS, RTS, DSR, DTR, RI, DCD)
DMA Controller

- 4 independent DMAs with 4 Channels each
  - 2 channels actively transferring at a given time
  - Round robin arbitration scheme between all channels of a DMA controller
  - Auto increment source and destination
  - Peripheral generate DMA events that dictate when the DMA can transfer
  - 1, 2, 4, 8, 16, or 32 FIFO depth per channel
  - Ping-Pong Buffering
10-bit SAR A/D

- Features:
  - 4 channel 10-bit Successive Approximation Converter
  - Measure battery voltage, internal analog voltage, and volume control by measuring across a potentiometer.
  - 4-Wire Resistive Touch Screen coordinate pair measurement.
  - General purpose outputs (drive high or drive low (except for GPAIN0 which only drives low)
  - General purpose voltage measurement
  - Continuous and single cycle conversion modes
  - Interrupt driven or polling conversion
  - Internal configurable bandgap reference voltages of 1V or 0.8V
  - Software controlled power down
  - Individually configurable general purpose outputs that drive high or low
  - Conversion rate 32 clock cycles with a 2MHz clock
C55x FFT Hardware Coprocessor

- Support 8 to 1024-point real- and complex-valued FFTs
- Internal twiddle factor generator for optimal use of memory bandwidth and more efficient programming.
- Basic and software-driven auto-scaling feature provides good precision vs. cycle count trade-off.
- Single-stage and double-stage modes enable to compute one or two stages in one pass, and thus to better handle odd power of two FFT widths.
- A programmable scaled or not-scaled butterfly operation (default is scaled).
- The capability to do FFT or Inverse FFT (default is FFT).
Watchdog Timer

- The purpose of Watchdog timer is to prevent system from locking up if the software becomes trapped in loops with no controlled exit.

- The watchdog timer requires a special service sequence to be executed periodically. Without this periodic servicing, the watchdog timer counter reaches zero and times out.

- The watchdog output can be selected to be connected to the local hardware reset, NMI (non-maskable interrupt), timer interrupt, or not connected. This allows maximum flexibility for utilizing the watchdog as required by the particular application.

- The watchdog timer consists of a 16 bit counter and a 16-bit pre-scalar, and supports up to 32 bit dynamic range. Out of reset, the watchdog is disabled by default, this allows a flexible period of time for code to be loaded into the on-chip memory.

- Once the watchdog is enabled, it can not be disabled by software, but can be disabled by watchdog time-out and hardware reset. A special key sequence is provided to prevent watchdog from being accidentally serviced while the software is trapped in a dead loop or in some other software failures.
Real Time Clock (RTC)

- Allows implementation of clock and calendar functions
  - 100 year calendar up to 2099
  - Count milliseconds, seconds, minutes, hours, days, day of the week, date month, and year with leap year compensation
  - BCD representation of time calendar and alarm
  - 24-hour clock
  - Every millisecond, second, minute, hour, or day alarm interrupt
  - Alarm interrupt: precise time of day

- Support external 32K oscillator
- Time compensation registers
- Millisecond time correction/round-up
- Separate power supply, allows DSP core to be completely powered down while RTC keeps track of time.
- Wakes up DSP LDO upon fixed alarm, periodic alarm, or external interrupt
I2S

- Up to 4 2-ch I2S ports with the following features:
- Full-duplex (transmit and receive) dual-channel communication for each I2S module
- Double buffered data registers that allow for continuous data stream
- Most significant bit (MSB) - first data transfers
- I2S/Left-justified and DSP data format with a data delay of 1 or 2 bits
- Data word-lengths of 8, 10, 12, 14, 16, 18, 20, 24, or 32 bits
- Ability to sign-extend received data samples for easy use in signal processing algorithms
- Packing mode to transmit/receive multiple samples of data before interrupting CPU or DMA
- Programmable polarity for both frame synchronization and bit clocks
- Digital loopback of data from transmit to receive register(s) for application code debug
- Stereo (in I2S/Left-justified or DSP data formats) or mono (in DSP data format) mode.
- Programmable divider for serial data clock (bit-clock) generation when I2S module is used as the master device
- Programmable divider for frame sync generation when I2S module is used as the master device
- Detection of over-run, under-run, and frame-sync error conditions
I2C

- Compliance of Philips I2C specification (reference to Philips I2C specification v2.1)
- Byte format transfer
- 7-bit or 10-bit addressing modes
- Multi-master transmitter/slave receiver mode
- Multi-master receiver/slave transmitter mode
- Free data format
- I2C data transfer rate of from 10kbps up to 400kbps (Philips I2C rate)
- Has one read and one write DMA event that can be used by the DMA
- Module operates from 6.7MHz to 13.3MHz
MMC/SD

- 2 ports, each one supporting a MultiMediaCard (MMC) or Secure Digital Memory Card (SD card).
- A programmable frequency of the clock that controls the timing of transfers between the MMC/SD Controller and memory card.
- 256 bit Read/Write FIFO to lower system overhead.
- Signaling to support DMA transfers
- 50 MHz maximum clock to SD (spec. V1.1)
LCD Bridge

- LCD Interface Display Driver
  - support a variety of MPU-like interface requirements which are commonly used in low-end LCD character displays.
  - support the signal function and timing requirements for the most prevalent parallel LCD display interface standards – MPU68, MPU80, MPUXX, and Hitachi HD44780U.

- LCD Raster Controller
  - Support raster-type LCD displays that require a constant flow of data at a ~70 Hz Frame rate.
  - Displays supported: passive and active color, and passive and active monochrome
  - Passive STN mode allows a total of 3375 possible colors, allowing any 16, 256 or 3375 colors to be displayed in each frame, as well as 15 grayscale levels for monochrome screens
  - Active TFT mode allows 4096 possible colors, in the 16 BPP mode, up to 64K possible colors
  - Screen size from 16x16 up to 1024x1024 pixels
  - Frame, line and pixel clocks, AC-bias drive signal
  - 4, 8, 12 and 16 bit-per-pixel display modes and dithering mode
USB 2.0 Slave

- Supports USB 2.0 peripheral at speeds HS (480 Mb/s) and FS (12 Mb/s)
- Supports all modes of transfers (control, bulk, interrupt)
- Supports 4 Transmit (TX) and 4 Receive (RX) endpoints in addition to endpoint 0
- Supports USB Session Resume (SRP) and Host Negotiation (HNP)
- Includes a 4K endpoint FIFO RAM, and supports programmable FIFO sizes
- Includes a DMA sub-module that supports 4 TX and 4 RX channels of CPPI 3.0 DMAs
- The DMA interfaces to the system bus through a CBA 3.0 VBUSP master port using incrementing addressing mode and up to 64 byte burst size
- Includes RNDIS mode for accelerating RNDIS type protocols using short packet termination over USB
Bootloader
C5505 Bootloader

Supports System Initialization from non-encrypted and encrypted content utilizing a 64-bit device ID and 64-bit manufacturer key from:

- **External Parallel Bus:**
  - NAND
    - non-encrypted
    - encrypted
  - NOR
    - non-encrypted
    - encrypted

- **Serial Bus:**
  - SPI EEPROM
    - non-encrypted
    - encrypted
  - I2C EEPROM
    - non-encrypted
    - encrypted
  - MMC/SD with FAT16/32
    - encrypted
  - UART
    - encrypted
  - USB
    - Encrypted
### C5505 Unsecure Boot Table

<table>
<thead>
<tr>
<th>Word</th>
<th>Content</th>
<th>Valid Data Entries</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Boot Signature (16-bits)</td>
<td>0x09AA, 0x095A</td>
</tr>
<tr>
<td>2:3</td>
<td>Entry Point (32-bits)</td>
<td>Byte address to begin execution</td>
</tr>
<tr>
<td>4</td>
<td>Register Configuration Count (16-bits, (N=\text{count}))</td>
<td>1 to (2^{16}-1)</td>
</tr>
<tr>
<td></td>
<td>First register address followed by address contents or 0xFFF F followed by delay count</td>
<td></td>
</tr>
<tr>
<td>4+2N</td>
<td>Last register address followed by contents or 0xFFF F followed by delay count</td>
<td></td>
</tr>
<tr>
<td>5+2N</td>
<td>Word Count (size) of first data block (16-bits)</td>
<td>1 to (2^{16}-1)</td>
</tr>
<tr>
<td></td>
<td>Size is the number of valid (non-pad) data words in block</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(M=(\text{size}+2)) rounded up to nearest multiple of 4</td>
<td></td>
</tr>
<tr>
<td>6+7 (2N)</td>
<td>Destination address to load the first source block (32-bits)</td>
<td>16-bit word address (0x00006 0 to 0x097 FFF)</td>
</tr>
<tr>
<td>8+2N</td>
<td>First word of first data block (16-bits)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>. . .</td>
<td></td>
</tr>
<tr>
<td>5+2N + M</td>
<td>Last word of first data block, often pad data (pad ded to 8-byte boundary)</td>
<td></td>
</tr>
<tr>
<td>X</td>
<td>Word Count (size) of first data block (16-bits)</td>
<td>1 to (2^{16}-1)</td>
</tr>
<tr>
<td></td>
<td>Size is the number of valid (non-pad) data words in block</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(N=(\text{size}+2)) rounded up to nearest multiple of 4</td>
<td></td>
</tr>
<tr>
<td>X+1</td>
<td>Destination address to load the last source block (32-bits)</td>
<td>16-bit word address (0x00006 0 to 0x097 FFF)</td>
</tr>
<tr>
<td>X+3</td>
<td>First word of last data block</td>
<td></td>
</tr>
<tr>
<td>X+N'</td>
<td>Last word of last data block, usually pad data (pad ded to 8-byte boundary)</td>
<td></td>
</tr>
<tr>
<td>X+N'+1</td>
<td>Zero word. Note that if more than one source block was read, word (X+N') shown above would be the last word of the last source block. Each block would have the format shown in the shaded entries.</td>
<td>0x0000</td>
</tr>
</tbody>
</table>
Secure Bootload vs ROM code

• Secure Bootload
  – Field upgrades through USB Secure bootload.
  – No additional costs.
  – No minimum volume.
  – Encrypted code can be bound to a specific device.
  – Simple JTAG disability by grounding JTAG pins in the board, under the package.
  – Used by customers in MP3 players since 2000.

• ROM Code
  – Fast start-up, no need to boot.
  – Field upgrade possible by patching ROM routines and bootloading encrypted code. Adds extra level of indirection on ROM routines.
  – JTAG must be disabled at factory to protect contents.
Instruction Set
Addressing Modes

- **Absolute:**
  - \( \text{k16: 16-bit unsigned constant } \times \text{abs(#k16)} \)
  - \( \text{k23: 23-bit unsigned constant } \times (#k23) \)
  - \( \text{I/O: 16-bit unsigned constant } \times \text{port(#k16)} \)

- **Direct:** uses DP, SP
  - Paging scheme with base pointer
    - \( \text{XDP} = \#0 \)
    - \( \text{T2} = @(\text{AC0}_L) \)

- **Indirect:**
  - Smem: single word (16-bit) access
  - Lmem: long-word (32-bit) access
  - Xmem and Ymem: two simultaneous single word access using C and D bus
  - Cmem: single word access using B-Bus
Indirect Addressing (single access)

- **Smem addressing options:**
  - *ARn*
  - *ARn+
  - *ARn-
  - *+ARn*
  - *-ARn*
  - *(ARn + AR0)*
  - *(ARn - AR0)*
  - *(ARn + T0), *(ARn + T1)*
  - *(ARn - T0), *(ARn - T1)*
  - *ARn(AR0)*
  - *ARn(T0), *ARn(T1)*
  - *(ARn + AR0B)*
  - *(ARn - AR0B)*
  - *(ARn + T0B)*
  - *(ARn - T0B)*
  - *ARn(#k16)*
  - *+ARn(#k16)*

- **Cmem options:**
  - *CDP*
  - *CDP+
  - *CDP-
  - *CDP(#k16)*
  - *+CDP(#k16)*

  post ARn = ARn + AR0
  post ARn = ARn - AR0
  post ARn = ARn + T0
  post ARn = ARn - T0
  post ARn = ARn + T0 bit reversed
  post ARn = ARn - T0 bit reversed

  post ARn = ARn + 1
  post CDP = CDP + 1
  post CDP = CDP - 1
  post CDP = CDP + k16
  pre CDP = CDP + k16
## Indirect Addressing (dual/triple access)

### Xmem and Ymem options:

<table>
<thead>
<tr>
<th>Expression</th>
<th>Post Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>(AR_n)</td>
<td>(AR_n + 1)</td>
</tr>
<tr>
<td>(AR_n^+)</td>
<td>(AR_n)</td>
</tr>
<tr>
<td>(AR_n^-)</td>
<td>(AR_n - 1)</td>
</tr>
<tr>
<td>((AR_n + AR_0))</td>
<td>(AR_n + AR_0)</td>
</tr>
<tr>
<td>((AR_n - AR_0))</td>
<td>(AR_n - AR_0)</td>
</tr>
<tr>
<td>((AR_n + T_0), (AR_n + T_1))</td>
<td>(AR_n + T_0)</td>
</tr>
<tr>
<td>((AR_n - T_0), (AR_n - T_1))</td>
<td>(AR_n - T_0)</td>
</tr>
</tbody>
</table>

### Cmem options:

<table>
<thead>
<tr>
<th>Expression</th>
<th>Post Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>(CD_P)</td>
<td>(CD_P + 1)</td>
</tr>
<tr>
<td>(CD_P^+)</td>
<td>(CD_P)</td>
</tr>
<tr>
<td>(CD_P^-)</td>
<td>(CD_P - 1)</td>
</tr>
<tr>
<td>((CD_P + AR_0))</td>
<td>(CD_P + AR_0)</td>
</tr>
<tr>
<td>((CD_P + T_0))</td>
<td>(CD_P + T_0)</td>
</tr>
</tbody>
</table>
## Circular Addressing

<table>
<thead>
<tr>
<th>Pointer</th>
<th>Linear/Circular Configuration Bit</th>
<th>Supplier of Main Data Page</th>
<th>Buffer Start Address Register</th>
<th>Buffer Size Register</th>
</tr>
</thead>
<tbody>
<tr>
<td>AR0</td>
<td>ST2_55(0) = AR0LC</td>
<td>AR0H</td>
<td>BSA01</td>
<td>BK03</td>
</tr>
<tr>
<td>AR1</td>
<td>ST2_55(1) = AR1LC</td>
<td>AR1H</td>
<td>BSA01</td>
<td>BK03</td>
</tr>
<tr>
<td>AR2</td>
<td>ST2_55(2) = AR2LC</td>
<td>AR2H</td>
<td>BSA23</td>
<td>BK03</td>
</tr>
<tr>
<td>AR3</td>
<td>ST2_55(3) = AR3LC</td>
<td>AR3H</td>
<td>BSA23</td>
<td>BK03</td>
</tr>
<tr>
<td>AR4</td>
<td>ST2_55(4) = AR4LC</td>
<td>AR4H</td>
<td>BSA45</td>
<td>BK47</td>
</tr>
<tr>
<td>AR5</td>
<td>ST2_55(5) = AR5LC</td>
<td>AR5H</td>
<td>BSA45</td>
<td>BK47</td>
</tr>
<tr>
<td>AR6</td>
<td>ST2_55(6) = AR6LC</td>
<td>AR6H</td>
<td>BSA67</td>
<td>BK47</td>
</tr>
<tr>
<td>AR7</td>
<td>ST2_55(7) = AR7LC</td>
<td>AR7H</td>
<td>BSA67</td>
<td>BK47</td>
</tr>
<tr>
<td>CDP</td>
<td>ST2_55(8) = CDPLC</td>
<td>CDPH</td>
<td>BSAC</td>
<td>BK03</td>
</tr>
</tbody>
</table>

Buffer must be alinged to a power of two greater than size:

\[ 2^{\text{buffer\_size}} > \text{buffer\_start\_address} \]
Stack

- Two pointers: decrement before push
  - SP: data stack
  - SSP: system stack

- Program Flow Registers:
  - PC: Program counter
  - RETA: Return Address Register, used in Fast-Return
  - CFCT: Control-Flow Context Register, used in Fast-Return

- Stack options:
  - Dual 16-bit stack with Fast-Return:
    - RETA and CFCT used for fast return
    - Independent SP and SSP
      RST: .ivec _MAIN0, USE_RETA
  - Dual 16-bit stack with Slow-Return:
    - RETA and CFCT are not used
    - Independent SP and SSP
  - 32-bit stack with Slow-Return:
    - RETA and CFCT are not used
    - Linked SP and SSP
C55x Rev 3 New Instructions

- **LMS:**
  
  ```
  lms (Xmem, Ymem, ACx, ACy)
  ACy = ACy + Xmem * Ymem
  ACx = rnd(ACx + (Xmem << #16))
  
  lmsf (Xmem, Ymem, ACx, ACy)
  ACx = T3 * Ymem
  ACy = ACy + Xmem * Ymem
  Ymem = HI(rnd(ACx + (Xmem << #16)))
  ```

- **Parallel Multiplies-Accumulates:** (4 operands for Dual-MAC operations)
  
  ```
  ACx = M40(rnd(ACx + (uns(Smem)) * uns(LO(Cmem))),
  ACy = M40(rnd(ACy + (uns(Smem)) * uns(H(LO(Cmem)))
  ```

- **Parallel Stores:** (freed Address Y-unit to perform parallel store)
  
  ```
  ACx = M40(rnd( uns(Xmem)) * uns(LO(Cmem)) ,
  ACy = M40(rnd( uns(Xmem)) * uns(HI(Cmem))
  || uns(*Ymem) = pair(HI(ACz))
  ```
New Instruction: 4-Operand Dual-MAC using 32-bit Bus

\[
\begin{align*}
AC_x &= \text{M}40(\text{rnd}(AC_x + (\text{uns}(\text{HI}(\text{Lmem}))) \times \text{uns}(\text{HI}(\text{Cmem})))), \\
AC_y &= \text{M}40(\text{rnd}(AC_y + (\text{uns}(\text{LO}(\text{Lmem}))) \times \text{uns}(\text{LO}(\text{Cmem})))), \\
\end{align*}
\]

D bus data from address Lmem
C bus data from address Lmem+1
B bus data from addresses Cmem (B[15:0]) and Cmem+1 (B[31:16])
++ modifier increments Lmem and Cmem by 2 words.

Advantage: Y-DAGEN not used. Available to do a parallel store.
New Instruction: 3-Operand Dual-MAC using 32-bit Bus

\[
\begin{align*}
AC_x &= M40(rnd(ACx + (\text{uns}(\text{Smem})) \times \text{uns}(\text{LO}(Cmem)))), \\
AC_y &= M40(rnd(ACy + (\text{uns}(\text{Smem})) \times \text{uns}(\text{HI}(Cmem))))
\end{align*}
\]

D bus data from address Smem
C bus data from address Smem
B bus data from addresses Cmem(B[15:0]) and Cmem+1 (B[31:16])
*+ modifier increments Cmem pointer by 2 words.

Advantage: Y-DAGEN not used. Available to do a parallel store.
New Instruction: 2-Operand UNS MAC using B-bus

\[ ACx = \text{rnd}(ACx + \text{Smem}*\text{uns}(Cmem)) \]

D bus data from address Smem
B bus data from addresses Cmem (B[15:0])
*+ modifier increments XCDP pointer by 2 words.

Advantage: Y-DAGEN not used. Available to do a parallel store.
3rd Party Network
Broad Third Party Network to Enable Faster Time to Market
BACKUP
<table>
<thead>
<tr>
<th>Feature</th>
<th>C5501</th>
<th>C5502</th>
<th>C5503</th>
<th>C5504</th>
<th>C5505</th>
<th>C5514</th>
<th>C5515</th>
<th>C5506</th>
<th>C5507</th>
<th>C5509A</th>
<th>C5510A</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Availability</strong></td>
<td>Now</td>
<td>Now</td>
<td>Now</td>
<td>Now</td>
<td>Now</td>
<td>Now</td>
<td>Now</td>
<td>Now</td>
<td>Now</td>
<td>Now</td>
<td>Now</td>
</tr>
<tr>
<td><strong>CPU</strong></td>
<td>C55x</td>
<td>C55x</td>
<td>C55x</td>
<td>C55x</td>
<td>C55x</td>
<td>C55x</td>
<td>C55x</td>
<td>C55x</td>
<td>C55x</td>
<td>C55x</td>
<td>C55x</td>
</tr>
<tr>
<td>Frequency (MHz)</td>
<td>300</td>
<td>200/300</td>
<td>200</td>
<td>150</td>
<td>150</td>
<td>120</td>
<td>120</td>
<td>108</td>
<td>200</td>
<td>200</td>
<td>160/200</td>
</tr>
<tr>
<td>Peak MMACS</td>
<td>600</td>
<td>400/600</td>
<td>300</td>
<td>300</td>
<td>240</td>
<td>240</td>
<td>216</td>
<td>400</td>
<td>400</td>
<td>320/400</td>
<td></td>
</tr>
<tr>
<td><strong>FFT Cop</strong></td>
<td>-</td>
<td>-</td>
<td>Y</td>
<td>-</td>
<td>Y</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td><strong>RAM</strong></td>
<td>32KB</td>
<td>64KB</td>
<td>64KB</td>
<td>256KB</td>
<td>320KB</td>
<td>256KB</td>
<td>128KB</td>
<td>128KB</td>
<td>128KB</td>
<td>256KB</td>
<td>320KB</td>
</tr>
<tr>
<td><strong>ROM</strong></td>
<td>32KB</td>
<td>32KB</td>
<td>64KB</td>
<td>128KB</td>
<td>128KB</td>
<td>64KB</td>
<td>64KB</td>
<td>64KB</td>
<td>64KB</td>
<td>32KB</td>
<td></td>
</tr>
<tr>
<td><strong>On-chip L1</strong></td>
<td>16KB</td>
<td>16KB</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>24KB</td>
<td></td>
</tr>
<tr>
<td><strong>EMIF</strong></td>
<td>32-bit</td>
<td>32-bit</td>
<td>16-bit</td>
<td>16-bit</td>
<td>16-bit</td>
<td>16-bit</td>
<td>16-bit</td>
<td>16-bit</td>
<td>16-bit</td>
<td>32-bit</td>
<td></td>
</tr>
<tr>
<td><strong>DMA</strong></td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td><strong>USB2.0 FS/HS</strong></td>
<td>-</td>
<td>-</td>
<td>1 HS</td>
<td>1 HS</td>
<td>1 HS</td>
<td>1 HS</td>
<td>1 HS</td>
<td>1 FS</td>
<td>1 FS</td>
<td>1 FS</td>
<td></td>
</tr>
<tr>
<td><strong>MMC/SD</strong></td>
<td>-</td>
<td>-</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>-</td>
<td>-</td>
<td>2</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td><strong>McBSP</strong></td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td><strong>HPI</strong></td>
<td>1 8-b</td>
<td>1 8/16-b</td>
<td>1 8/16-b</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>1 8/16-bit</td>
<td>1 8/16-bit</td>
</tr>
<tr>
<td><strong>I2S/I2C</strong></td>
<td>0/1</td>
<td>0/1</td>
<td>0/1</td>
<td>4/1</td>
<td>4/1</td>
<td>4/1</td>
<td>4/1</td>
<td>0/1</td>
<td>0/1</td>
<td>0/1</td>
<td>0/1</td>
</tr>
<tr>
<td><strong>UART</strong></td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td><strong>SPI</strong></td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>1 w/4CS</td>
<td>1 w/4CS</td>
<td>1 w/4CS</td>
<td>1 w/4CS</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td><strong>ADC</strong></td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td><strong>Timer</strong></td>
<td>3 (1 WD)</td>
<td>3 (1 WD)</td>
<td>4 (1 RTC, 1 WD)</td>
<td>3 (WD, 1 RTC)</td>
<td>3 (WD, 1 RTC)</td>
<td>3 (WD, 1 RTC)</td>
<td>3 (WD, 1 RTC)</td>
<td>4 (1 RTC, 1 WD)</td>
<td>4 (1 RTC, 1 WD)</td>
<td>4 (1 RTC, 1 WD)</td>
<td>2</td>
</tr>
<tr>
<td><strong>LDO</strong></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td><strong>LCD Bridge</strong></td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td><strong>Package</strong></td>
<td>15x15</td>
<td>15x15</td>
<td>12x12</td>
<td>10x10</td>
<td>10x10</td>
<td>10x10</td>
<td>10x10</td>
<td>12x12</td>
<td>12x12</td>
<td>12x12</td>
<td>15x15</td>
</tr>
</tbody>
</table>
## C54x DSP Comparison – 23 Devices

<table>
<thead>
<tr>
<th>Feature</th>
<th>C549</th>
<th>C5401</th>
<th>C5402/A</th>
<th>C5404</th>
<th>C5407</th>
<th>C5409/A</th>
<th>C5410/A</th>
<th>C5416</th>
<th>C5420</th>
<th>C5421</th>
<th>C5441</th>
<th>C5470</th>
<th>C5471</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>Now</td>
<td>Now</td>
<td>Now</td>
<td>Now</td>
<td>Now</td>
<td>Now</td>
<td>Now</td>
<td>Now</td>
<td>Now</td>
<td>Now</td>
<td>Now</td>
<td>Now</td>
<td>Now</td>
</tr>
<tr>
<td>CPU</td>
<td>C54x</td>
<td>C54x</td>
<td>C54x</td>
<td>C54x</td>
<td>C54x</td>
<td>C54x</td>
<td>C54x</td>
<td>2 C54x</td>
<td>2 C54x</td>
<td>2 C54x</td>
<td>4 C54x</td>
<td>C54x + ARM7</td>
<td>C54x + ARM7</td>
</tr>
<tr>
<td>Peak MMACS</td>
<td>100</td>
<td>120</td>
<td>50</td>
<td>80/100</td>
<td>120</td>
<td>120</td>
<td>80/100</td>
<td>120/160</td>
<td>120/160</td>
<td>200 (2x100)</td>
<td>200 (2x100)</td>
<td>532 (4x133)</td>
<td>100 + 47.5</td>
</tr>
<tr>
<td>RAM</td>
<td>64KB</td>
<td>16KB</td>
<td>32KB</td>
<td>32KB</td>
<td>80KB</td>
<td>64KB</td>
<td>128KB</td>
<td>256KB</td>
<td>384KB</td>
<td>512KB</td>
<td>1280KB</td>
<td>160KB</td>
<td>160KB</td>
</tr>
<tr>
<td>ROM</td>
<td>32KB</td>
<td>8KB</td>
<td>8/32KB</td>
<td>128KB</td>
<td>256KB</td>
<td>32KB</td>
<td>32KB</td>
<td>32KB</td>
<td>8KB</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EMIF (Async)</td>
<td>1 16-bit</td>
<td>1 16-bit</td>
<td>1 16-bit</td>
<td>1 16-bit</td>
<td>1 16-bit</td>
<td>1 16-bit</td>
<td>1 16-bit</td>
<td>1 16-bit +1 32-bit</td>
<td>1 16-bit +1 32-bit</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DMA</td>
<td>-</td>
<td>-</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>2x6</td>
<td>4x6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>McBSP</td>
<td>-</td>
<td>-</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>6</td>
<td>12</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HPI</td>
<td>1 18-bit</td>
<td>18/16-bit</td>
<td>1 18/16-bit</td>
<td>1 18/16-bit</td>
<td>1 18/16-bit</td>
<td>-</td>
<td>1 18/16-bit</td>
<td>-</td>
<td>-</td>
<td>1 16-bit</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>I2C</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>SPI</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10/100 EMAC</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>1</td>
</tr>
<tr>
<td>ADC</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>UART</td>
<td>-</td>
<td>-</td>
<td>1</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Timer</td>
<td>1</td>
<td>2</td>
<td>2/2/1</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>4 (1 WD)</td>
<td>4 (1 WD)</td>
</tr>
<tr>
<td>Core Voltage</td>
<td>1.8</td>
<td>1.8/1.6</td>
<td>1.5</td>
<td>1.5</td>
<td>1.8</td>
<td>1.5/1.6</td>
<td>2.5/3.3</td>
<td>1.6/1.5</td>
<td>1.5/1.6</td>
<td>1.8</td>
<td>1.6</td>
<td>1.8</td>
<td>1.8</td>
</tr>
<tr>
<td>IO Voltage</td>
<td>2.5/3.3</td>
<td>3.3</td>
<td>1.8-3.3</td>
<td>3.3</td>
<td>3.3</td>
<td>1.8-3.3</td>
<td>3.3</td>
<td>3.3</td>
<td>3.3</td>
<td>3.3</td>
<td>/3.3</td>
<td>/3.3</td>
<td>3.3</td>
</tr>
<tr>
<td>Pin No.</td>
<td>144</td>
<td>144</td>
<td>144</td>
<td>144</td>
<td>144</td>
<td>144</td>
<td>144</td>
<td>144</td>
<td>144</td>
<td>144</td>
<td>169/176</td>
<td>257</td>
<td>257</td>
</tr>
</tbody>
</table>
TMS320VC5502 DSP

**Features**

- **Core**
  - Dual MAC, C55x Rev 2 CPU
  - 1.26V @ 200MHz, 300 MHz

- **Memory**
  - 80-KB On-Chip Memory
    - 64-KB DARAM + 16-KB I-Cache
  - 32-KB ROM

- **Peripherals**
  - 2 McBSPs
  - Multi-master and Slave I²C with 7 or 10-bit addressing modes
  - UART muxed with a McBSP
  - 2 64-bit timers; 1 Watchdog
  - 76 GPIO, 8 dedicated
  - 6 Channel DMA
  - Analog PLL
  - 32-bit EMIF with asynchronous SRAM, SDRAM, SBRAM
  - 8-bit/16-bit Host Port Interface

- **I/O**
  - 3.0V - 3.3V

- Package: 144-Pin LQFP, 201-Pin µ*BGA
- In Production Today
TMS320VC5509A DSP

**Portable Media/Comm/Audio Device**

- **Core**
  - Dual MAC, C55x Rev 2 CPU
  - 1.2V @ 108MHz, 1.35V @ 144MHz, 1.6V @ 200 MHz

- **Memory**
  - 256-KB On-Chip Memory
    - 64-KB DARAM + 192-KB SARAM
  - 64-KB ROM

- **Peripherals**
  - USB 2.0 full speed (12Mbps)
  - 2 MultiMedia Card/Secure Digital (MMC/SD) serial ports
  - 4ch/2ch 10-bit SAR, 500-us ADC 3 McBSPs
  - Multi-master and Slave I²C with 7 or 10-bit addressing modes
  - 2 20-bit timers; 1 Watchdog
  - 36/35 GPIO, 8/7 dedicated
  - 6 Channel DMA
  - PLL
  - 16-bit EMIF with asynchronous SRAM, SDRAM, and mixed with HPI
  - Real-time clock with 32-KHz crystal input, separate power

- **I/O**
  - 2.7V - 3.6V

*Package: 144-Pin LQFP, 179-Pin μ*BGA
*In Production Today
TMS320VC5507 DSP

Features

- **Core**
  - Dual MAC, C55x Rev 2 CPU
  - 1.2V @ 108MHz, 1.35V @ 144MHz, 1.6V @ 200 MHz

- **Memory**
  - 128-KB On-Chip Memory:
    - 64-KB DARAM + 64-KB SARAM
  - 64-KB ROM

- **Peripherals**
  - USB 2.0 full speed (12Mbps)
  - 2ch/4ch 10-bit SAR, 500-us ADC for keypad, button and battery monitoring functions
  - 3 McBSPs
  - Multi-master and Slave IC with 7 or 10-bit addressing modes
  - 2 20-bit timers; 1 Watchdog
  - 36/35 GPIO, 8/7 dedicated
  - 6 Channel DMA
  - PLL
  - 16-bit EMIF with asynchronous SRAM, SDRAM, and muxed with HPI
  - Real-time clock with 32-KHz crystal input, separate power

- **I/O**
  - 2.7V - 3.6V

- **Package:** 144-Pin LQFP, 179-Pin µ*BGA
- **In Production Today**
TMS320VC5506 DSP

Features

- Core
  - Dual MAC, C55x Rev 2 CPU
  - 1.2V @ 108MHz

- Memory
  - 128-KB On-Chip Memory:
    - 64-KB DARAM + 64-KB SARAM
  - Boot ROM

- Peripherals
  - USB 2.0 full speed (12Mbps)
  - 2ch/4ch 10-bit, 500-us ADC for keypad, button and battery monitoring functions
  - 3 McBSPs
  - Multi-master and Slave PC with 7 or 10-bit addressing modes
  - 2 20-bit timers; 1 Watchdog
  - 36/35 GPIO, 8/7 dedicated
  - 6 Channel DMA
  - PLL
  - 16-bit EMIF with asynchronous SRAM, SDRAM, and muxed with HPI
  - Real-time clock with 32-KHz crystal input, separate power

- I/O
  - 2.7V - 3.6V

- Package: 144-Pin LQFP, 179-Pin μ*BGA
- In Production Today
TMS320VC5503 DSP

Features

- **Core**
  - Dual MAC, C55x Rev 2 CPU
  - 1.2V @ 108MHz, 1.35V @ 144MHz, 1.6V @ 200 MHz

- **Memory**
  - 64-KB On-Chip Memory:
    - 64-KB DARAM
  - 64-KB ROM

- **Peripherals**
  - 2ch/4ch 10-bit SAR, 500-us ADC for keypad, button and battery monitoring functions
  - 3 McBSPs
  - Multi-master and Slave PC with 7 or 10-bit addressing modes
  - 2 20-bit timers; 1 Watchdog
  - 36/35 GPIO, 87 dedicated
  - 6 Channel DMA
  - PLL
  - 16-bit EMIF with asynchronous SRAM, SDRAM, and muxed with HPI
  - Real-time clock with 32-KHz crystal input, separate power

- **I/O**
  - 2.7V - 3.6V

- **Package:** 144-Pin LQFP, 179-Pin μ*BGA
- **In Production Today**