Lecture 10a:
Digital Signal Processors:
A TI Architectural History

Collated by: Professor Kurt Keutzer
Computer Science 252, Spring 2000
With contributions from:
Dr. Brock Barton, Clark Hise TI;
Dr. Surendar S. Magar, Berkeley
Concept Research Corporation
DSP ARCHITECTURE EVOLUTION

Application Examples

Video/Imaging
W-CDMA
Radars
Digital Radios
High-End Control
Modems
Voice Coding
Instruments
Low-End Modems
Industrial Control

Multipliers (MUL) → Multiprocessors (MP)


Multi-Processing
DSP Building Blocks & Bit Slice Processors (MUL, etc.)
μC and Analog

Function/Application Specific (MP)
DSP μP and RISC (MP)
μP
## DSP ARCHITECTURE

### Enabling Technologies

<table>
<thead>
<tr>
<th>Time Frame</th>
<th>Approach</th>
<th>Primary Application</th>
<th>Enabling Technologies</th>
</tr>
</thead>
<tbody>
<tr>
<td>Early 1970’s</td>
<td>Discrete logic</td>
<td>Non-real time processing</td>
<td>Bipolar SSI, MSI</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Simulation</td>
<td>FFT algorithm</td>
</tr>
<tr>
<td>Late 1970’s</td>
<td>Building block</td>
<td>Military radars</td>
<td>Single chip bipolar multiplier</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Digital Comm.</td>
<td>Flash A/D</td>
</tr>
<tr>
<td>Early 1980’s</td>
<td>Single Chip DSP μP</td>
<td>Telecom</td>
<td>μP architectures</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Control</td>
<td>NMOS/CMOS</td>
</tr>
<tr>
<td>Late 1980’s</td>
<td>Function/Application</td>
<td>Computers</td>
<td>Vector processing</td>
</tr>
<tr>
<td></td>
<td>specific chips</td>
<td>Communication</td>
<td>Parallel processing</td>
</tr>
<tr>
<td>Early 1990’s</td>
<td>Multiprocessing</td>
<td>Video/Image Processing</td>
<td>Advanced multiprocessing</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>VLIW, MIMD, etc.</td>
</tr>
<tr>
<td>Late 1990’s</td>
<td>Single-chip</td>
<td>Wireless telephony</td>
<td>Low power single-chip DSP</td>
</tr>
<tr>
<td></td>
<td>multiprocessor</td>
<td>Internet related</td>
<td>Multiprocessing</td>
</tr>
</tbody>
</table>
# Texas Instruments TMS320 Family

## Multiple DSP µP Generations

<table>
<thead>
<tr>
<th>First Sample</th>
<th>Bit Size</th>
<th>Clock speed (MHz)</th>
<th>Instruction Throughput (MIPS)</th>
<th>MAC execution (ns)</th>
<th>MOPS</th>
<th>Device density (# of transistors)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Uniprocessor Based (Harvard Architecture)</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TMS32010</td>
<td>16 integer</td>
<td>20</td>
<td>5</td>
<td>400</td>
<td>5</td>
<td>58,000 (3µ)</td>
</tr>
<tr>
<td>TMS320C25</td>
<td>16 integer</td>
<td>40</td>
<td>10</td>
<td>100</td>
<td>20</td>
<td>160,000 (2µ)</td>
</tr>
<tr>
<td>TMS320C30</td>
<td>32 flt.pt.</td>
<td>33</td>
<td>17</td>
<td>60</td>
<td>33</td>
<td>695,000 (1µ)</td>
</tr>
<tr>
<td>TMS320C50</td>
<td>16 integer</td>
<td>57</td>
<td>29</td>
<td>35</td>
<td>60</td>
<td>1,000,000 (0.5µ)</td>
</tr>
<tr>
<td>TMS320C2XXX</td>
<td>16 integer</td>
<td>40</td>
<td>25</td>
<td>20</td>
<td>80</td>
<td></td>
</tr>
<tr>
<td><strong>Multiprocessor Based</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TMS320C80</td>
<td>32 integer/flt.</td>
<td></td>
<td>2 GOPS</td>
<td>MIMD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TMS320C62XX</td>
<td>16 integer</td>
<td>1600</td>
<td>5</td>
<td></td>
<td>20</td>
<td>VLIW</td>
</tr>
<tr>
<td>TMS310C67XX</td>
<td>32 flt. pt.</td>
<td></td>
<td>5</td>
<td></td>
<td>1</td>
<td>VLIW</td>
</tr>
</tbody>
</table>
First Generation DSP μP Case Study
TMS32010 (Texas Instruments) - 1982

Features
◆ 200 ns instruction cycle (5 MIPS)
◆ 144 words (16 bit) on-chip data RAM
◆ 1.5K words (16 bit) on-chip program ROM - TMS32010
◆ External program memory expansion to a total of 4K words at full speed
◆ 16-bit instruction/data word
◆ Single cycle 32-bit ALU/accumulator
◆ Single cycle 16 x 16-bit multiply in 200 ns
◆ Two cycle MAC (5 MOPS)
◆ Zero to 15-bit barrel shifter
◆ Eight input and eight output channels
### TMS32010 Program Memory Maps

#### Microcomputer Mode

<table>
<thead>
<tr>
<th>Address</th>
<th>16-bit word</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Reset 1st Word</td>
</tr>
<tr>
<td>1</td>
<td>Reset 2nd Word</td>
</tr>
<tr>
<td>2</td>
<td>Interrupt</td>
</tr>
<tr>
<td>1525</td>
<td></td>
</tr>
<tr>
<td>1536</td>
<td></td>
</tr>
<tr>
<td>4095</td>
<td></td>
</tr>
</tbody>
</table>

#### Microprocessor Mode

<table>
<thead>
<tr>
<th>Address</th>
<th>16-bit word</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Reset 1st Word</td>
</tr>
<tr>
<td>1</td>
<td>Reset 2nd Word</td>
</tr>
<tr>
<td>2</td>
<td>Interrupt</td>
</tr>
<tr>
<td>4095</td>
<td></td>
</tr>
</tbody>
</table>
Digital FIR Filter Implementation (Uniprocessor-Circular Buffer)

Start each Time here

\[ a_{n-1} \quad a_{n-2} \quad a_1 \quad a_0 \]

\[ a_0 \quad a_{n-1} \]

\[ X_0 \quad X_1 \quad X_2 \quad X_3 \quad X_4 \quad X_5 \quad \cdots \quad X_{n-1} \]

Replace starting value with new value

Start 1st. Cycle

End 2nd. Cycle

Start

End

Acc
TMS32010 FIR FILTER PROGRAM
Indirect Addressing (Smaller Program Space)

\[ Y(n) = x[n-(N-1)] \cdot h(N-1) + x[n-(N-2)] \cdot h(N-2) + \ldots + x(n) \cdot h(0) \]

* THIS SECTION OF CODE IMPLEMENTS THE EQUATION:
* \[ x(n-(N-1))h(N-1) + x(n-(N-2))h(N-2) + \ldots + x(n)h(0) = y(n) \]
* 
* LARP AR0  * AUXILIARY REGISTER POINTER SET TO AR0 *
NXTPT  IN XN,PA2  * PULL IN NEW INPUT FROM PORT PA0 *
* 
LARK AR0,XNMM1  * AR0 POINTS TO X(n-(N-1)) *
LARK AR1,HNMM1  * AR1 POINTS TO H(N-1) *
* 
ZAC  * ZERO THE ACCUMULATOR *
* 
LT *-,AR1  * x(n-(N-1))h(N-1) *
MPY *-,AR0  
* 
LOOP LTD *,AR1  * x(n-(N-1))h(N-1)+x(n-(N-2))h(N-2)+\ldots+x(n)h(0)=y(n)*
MPY *-,AR0  
* 
BANZ LOOP  * IF AR0 DOES NOT EQUAL ZERO, *
* THEN DECREMENT AR0 AND BRANCH TO LOOP *
* 
APAC  * ADD THE P REGISTER TO THE ACCUMULATOR *
* 
SACH YN,1  * STORE THE RESULT IN YN *
* 
OUT YN,PA2  * OUTPUT THE RESPONSE TO PORT PA1 *
* 
B NXTPT  * GO GET THE NEXT INPUT POINT *

For N=50, Indirect Addressing \( t=42 \) \( \mu s \) (23.8 KHz)
For N=50, Direct Addressing \( t=21.6 \) \( \mu s \) (40.2 KHz)
TMS320C203/LC203 BLOCK DIAGRAM
DSP Core Approach - 1995
Third Generation DSP µP Case Study
TMS320C30 - 1988

TMS320C30 Key Features

◆ 60 ns single-cycle instruction execution time
  ■ 33.3 MFLOPS (million floating-point operations per second)
  ■ 16.7 MIPS (million instructions per second)
◆ One 4K x 32-bit single-cycle dual-access on-chip ROM block
◆ Two 1K x 32-bit single-cycle dual-access on-chip RAM blocks
◆ 64 x 32-bit instruction cache
◆ 32-bit instruction and data words, 24-bit addresses
◆ 40/32-bit floating-point/integer multiplier and ALU
◆ 32-bit barrel shifter
Third Generation DSP μP Case Study
TMS320C30 - 1988

TMS320C30 Key Features (cont.)

◆ Eight extended precision registers (accumulators)
◆ Two address generators with eight auxiliary registers and two auxiliary register arithmetic units
◆ On-chip direct memory Access (DMA) controller for concurrent I/O and CPU operation
◆ Parallel ALU and multiplier instructions
◆ Block repeat capability
◆ Interlocked instructions for multiprocessing support
◆ Two serial ports to support 8/16/32-bit transfers
◆ Two 32-bit timers
◆ 1 µ CDMOS Process
TMS320C30 BLOCK DIAGRAM
TMS320C3x CPU BLOCK DIAGRAM
TMS320C3x MEMORY BLOCK DIAGRAM
## TMS320C30 Memory Organization

### Microprocessor Mode

<table>
<thead>
<tr>
<th>Address Range</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oh</td>
<td>Interrupt locations &amp; reserved (192)</td>
</tr>
<tr>
<td>BFh</td>
<td>External STRB Active</td>
</tr>
<tr>
<td>COh</td>
<td></td>
</tr>
<tr>
<td>7FFFFFFFh 800000h</td>
<td>Expansion BUS MSTRB Active (8K)</td>
</tr>
<tr>
<td>801FFFFh 802000h</td>
<td>Reserved (8K)</td>
</tr>
<tr>
<td>803FFFFh 804000h</td>
<td>Expansion Bus IOSTRB Active (8K)</td>
</tr>
<tr>
<td>805FFFFh 806000h</td>
<td>Reserved (8K)</td>
</tr>
<tr>
<td>807FFFFH 80800h</td>
<td>Peripheral Bus Memory Mapped Registers (Internal) (6K)</td>
</tr>
<tr>
<td>8097FFFFH 809800h</td>
<td>RAM Block 0 (1K) (Internal)</td>
</tr>
<tr>
<td>809BFFFFH 809C00h</td>
<td>RAM Block 1 (1K) (Internal)</td>
</tr>
<tr>
<td>809FFFFH 80A00h</td>
<td></td>
</tr>
<tr>
<td>0FFFFFFFh</td>
<td></td>
</tr>
</tbody>
</table>

### Microcomputer Mode

<table>
<thead>
<tr>
<th>Address Range</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oh</td>
<td>Interrupt locations &amp; reserved (192)</td>
</tr>
<tr>
<td>BFh</td>
<td></td>
</tr>
<tr>
<td>COh</td>
<td></td>
</tr>
<tr>
<td>1000h</td>
<td>ROM (Internal)</td>
</tr>
<tr>
<td>7FFFFFFFh 800000h</td>
<td>Expansion BUS MSTRB Active (8K)</td>
</tr>
<tr>
<td>801FFFFh 802000h</td>
<td>Reserved (8K)</td>
</tr>
<tr>
<td>803FFFFh 804000h</td>
<td>Expansion Bus IOSTRB Active (8K)</td>
</tr>
<tr>
<td>805FFFFh 806000h</td>
<td>Reserved (8K)</td>
</tr>
<tr>
<td>807FFFFH 80800h</td>
<td>Peripheral Bus Memory Mapped Registers (Internal) (6K)</td>
</tr>
<tr>
<td>8097FFFFH 809800h</td>
<td>RAM Block 0 (1K) (Internal)</td>
</tr>
<tr>
<td>809BFFFFH 809C00h</td>
<td>RAM Block 1 (1K) (Internal)</td>
</tr>
<tr>
<td>809FFFFH 80A00h</td>
<td></td>
</tr>
<tr>
<td>0FFFFFFFh</td>
<td></td>
</tr>
</tbody>
</table>
TMS320C30 FIR FILTER PROGRAM

\[ Y(n) = x[n-(N-1)] \cdot h(N-1) + x[n-(N-2)] \cdot h(N-2) + \ldots + x(n) \cdot h(0) \]

\begin{verbatim}
TOP    LDF    IN, R3 ; Read input sample.
STF    R3, *AR1++% ; Store with other samples, 
                ; and point to top of buffer.
LDF    0, R0 ; Initialize R0.
LDF    0, R2 ; Initialize R2.

* Filter *

RPTS   N-1 ; Repeat next instruction.
MPYF3   *AR0++%, *AR1++%, R0
|| ADDF3  R0, R2, R2 ; Multiply and accumulate.
|| ADDF   R0, R2 ; Last product accumulated.

* STF    R2, Y ; Save result.
B TOP    ; Repeat.
\end{verbatim}

For \( N=50 \), \( t=3.6 \text{ ms} \) (277 KHz)

THE WORLD LEADER IN DSP SOLUTIONS
‘C54x Architecture
Architecture optimized for DSP

#1: CPU designed for efficient DSP processing
- MAC unit, 2 Accumulators, Additional Adder, Barrel Shifter

#2: Multiple busses for efficient data and program flow
- Four busses and large on-chip memory that result in sustained performance near peak

#3: Highly tuned instruction set for powerful DSP computing
- Sophisticated instructions that execute in fewer cycles, with less code and low power demands
Key #1: DSP engine

\[ Y = \sum_{n=1}^{40} a_n \times x_n \]
**Key #1: MAC Unit**

MAC  *AR2+, *AR3+, A

Data  Acc A  Temp  Coeff  Prgm  Data  Acc A

S/U  ---|---  S/U

Fractional Mode Bit

MPY

ADD

acc A
cacc B
**Key #1: Accumulators + Adder**

General-Purpose Math example: \( t = s + e - r \)

- **A Bus**
- **B Bus**
- **MUX**
- **acc A**
- **acc B**
- **ALU**
- **U Bus**
- **Shifter**
- **LD @s, A**
- **ADD @e, A**
- **SUB @r, A**
- **STL A, @t**

**Diagram Elements**: A Bus, B Bus, MUX, acc A, acc B, ALU, U Bus, Shifter, LD @s, ADD @e, SUB @r, STL A, @t
Key #1: Barrel shifter

LD @X, 16, A
STH @B, Y

A B C D

Barrel Shifter (-16+-31)

S Bus

ALU

E Bus
Key #1: Temporary register

LD @x, T
MPY @a, A

For example:
A = xa
Key #2: Efficient data/program flow

#1: CPU designed for efficient DSP processing
- MAC unit, 2 Accumulators, Additional Adder, Barrel Shifter

#2: Multiple busses for efficient data and program flow
- Four busses and large on-chip memory that result in sustained performance near peak

#3: Highly tuned instruction set for powerful DSP computing
- Sophisticated instructions that execute in fewer cycles, with less code and low power demands
Key #2: Multiple busses

MAC *AR2+, *AR3+, A
Key #2: Pipeline

Prefetch Fetch Decode Access Read Execute

- Prefetch: Calculate address of instruction
- Fetch: Collect instruction
- Decode: Interpret instruction
- Access: Collect address of operand
- Read: Collect operand
- Execute: Perform operation
Key #2: Bus usage
Key #2: Pipeline performance

Fully loaded pipeline
Key #3: Powerful instructions

#1: CPU designed for efficient DSP processing
- MAC Unit, 2 Accumulators, Additional Adder, Barrel Shifter

#2: Multiple busses for efficient data and program flow
- Four busses and large on-chip memory that result in sustained performance near peak

#3: Highly tuned instruction set for powerful DSP computing
- Sophisticated instructions that execute in fewer cycles, with less code and low power demands
Key #3: Advanced applications

Symmetric FIR filter: FIRS
Adaptive filtering: LMS
Polynomial evaluation: POLY
Code book search: STRCD

Viterbi: SACCD

The World Leader in DSP Solutions
C62x Architecture
TMS320C6201 Revision 2

Program Cache / Program Memory
32-bit address, 256-Bit data
512K Bits RAM

C6201 CPU Megamodule

Program Fetch
Instruction Dispatch
Instruction Decode

Data Path 1
A Register File
L1 S1 M1 D1

Data Path 2
B Register File
D2 M2 S2 L2

Control Registers
Control Logic
Test
Emulation
Interrupts

Ext. Memory Interface

Host Port Interface

4-DMA

Pwr Dwn

Data Memory
32-Bit address, 8-, 16-, 32-Bit data
512K Bits RAM

2 Timers
2 Multi-channel buffered serial ports (T1/E1)

Program Cache / Program Memory
32-bit address, 256-Bit data
512K Bits RAM
C6201 Internal Memory Architecture

◆ Separate Internal Program and Data Spaces
◆ Program
  ■ 16K 32-bit instructions (2K Fetch Packets)
  ■ 256-bit Fetch Width
  ■ Configurable as either
    ♦ Direct Mapped Cache, Memory Mapped Program Memory
◆ Data
  ■ 32K x 16
  ■ Single Ported Accessible by Both CPU Data Buses
  ■ 4 x 8K 16-bit Banks
    ♦ 2 Possible Simultaneous Memory Accesses (4 Banks)
    ♦ 4-Way Interleave, Banks and Interleave Minimize Access Conflicts
C62x Interrupts

- 12 Maskable Interrupts, Non-Maskable Interrupt (NMI)
- Interrupt Return Pointers (IRP, NRP)
- Fast Interrupt Handling
  - Branches Directly to 8-Instruction Service Fetch Packet
  - Can Branch out with no overhead for longer service
  - 7 Cycle Overhead: Time When No Code is Running
  - 12 Cycle Latency: Interrupt Response Time
- Interrupt Acknowledge (IACK) and Number (INUM) Signals
- Branch Delay Slots Protected From Interrupts
- Edge Triggered
C62x Datapaths

- Registers A0 - A15
- Registers B0 - B15

- DDATA_I1 (load data)
- DDATA_O1 (store data)
- DADDR1 (address)
- DADDR2 (address)
- DDATA_O2 (store data)

- Cross Paths
- 40-bit Write Paths (8 MSBs)
- 40-bit Read Paths/Store Paths
Functional Units

◆ L-Unit (L1, L2)
  ■ 40-bit Integer ALU, Comparisons
  ■ Bit Counting, Normalization
◆ S-Unit (S1, S2)
  ■ 32-bit ALU, 40-bit Shifter
  ■ Bitfield Operations, Branching
◆ M-Unit (M1, M2)
  ■ 16 x 16 -> 32
◆ D-Unit (D1, D2)
  ■ 32-bit Add/Subtract
  ■ Address Calculations
C62x Datapaths

Registers A0 - A15

Registers B0 - B15

DDATA_O1 (store data)  DDATA_I1 (load data)  DADR1 (address)

DDATA_I2 (load data)  DDATA_O2 (store data)

Cross Paths
40-bit Write Paths (8 MSBs)
40-bit Read Paths/Store Paths
C62x Instruction Packing
Instruction Packing Advanced VLIW

Example 1

A B C D E F G H

Fetch Packet
- CPU fetches 8 instructions/cycle

Execute Packet
- CPU executes 1 to 8 instructions/cycle
- Fetch packets can contain multiple execute packets

Parallelism determined at compile / assembly time

Examples
- 1) 8 parallel instructions
- 2) 8 serial instructions
- 3) Mixed Serial/Parallel Groups
  - A // B
  - C
  - D
  - E // F // G // H

Reduces Codesize, Number of Program Fetches, Power Consumption

Example 2

A B C D E F G H

Example 3

A B C D E F G H

THE WORLD LEADER IN DSP SOLUTIONS

T HE W O R LD L E A D E R I N D S P S OL U T I O N S
C62x Pipeline Operation
Pipeline Phases

Fetch • Decode • Execute

PG PS PW PR DP DC E1 E2 E3 E4 E5

◆ Decode
◆ Single-Cycle Throughput
◆ Operate in Lock Step
◆ Fetch • Execute
  ▪ PG Program Address Generate
  ▪ PS Program Address Send
  ▪ PW Program Access Ready Wait
  ▪ PR Program Fetch Packet Receive

Execute Packet 1
Execute Packet 2
Execute Packet 3
Execute Packet 4
Execute Packet 5
Execute Packet 6
Execute Packet 7

The World Leader In DSP Solutions
C62x Pipeline Operation
Delay Slots

- Delay Slots: number of extra cycles until result is:
  - written to register file
  - available for use by a subsequent instruction
  - Multi-cycle NOP instruction can fill delay slots while minimizing codesize impact

Most Instructions: E1 No Delay

Integer Multiply: E1 E2 1 Delay Slots

Loads: E1 E2 E3 E4 E5 4 Delay Slots

Branches: E1

Branch Target: P G P S P W P R D P D C E1 5 Delay Slots
C6000 Pipeline Operation Benefits

◆ Cycle Time
  ■ Allows 6 ns cycle time on 67x
  ■ Allows 5 ns cycle time & single cycle execution on C62x

◆ Parallelism
  ■ 8 new instructions can always be dispatched every cycle

◆ High Performance Internal Memory Access
  ■ Pipelined Program and Data Accesses
  ■ Two 32-bit Data Accesses/Cycle (C62x)
  ■ Two 64-bit Data Accesses/Cycle (C67x)
  ■ 256-bit Program Access/Cycle

◆ Good Compiler Target
  ■ Visible: No Variable-Length Pipeline Flow
  ■ Deterministic: Order and Time of Execution
  ■ Orthogonal: Independent Instructions
C6000 Instruction Set Features
Conditional Instructions

◆ All Instructions can be Conditional
  ■ A1, A2, B0, B1, B2 can be used as Conditions
  ■ Based on Zero or Non-Zero Value
  ■ Compare Instructions can allow other Conditions (<, >, etc)

◆ Reduces Branching
◆ Increases Parallelism
C6000 Instruction Set Addressing Features

◆ Load-Store Architecture
◆ Two Addressing Units (D1, D2)
◆ Orthogonal
  ■ Any Register can be used for Addressing or Indexing
◆ Signed/Unsigned Byte, Half-Word, Word, Double-Word Addressable
  ■ Indexes are Scaled by Type
◆ Register or 5-Bit Unsigned Constant Index
C6000 Instruction Set Addressing Features

◆ Indirect Addressing Modes
  - Pre-Increment       *++R[index]
  - Post-Increment     *R++[index]
  - Pre-Decrement      *--R[index]
  - Post-Decrement     *R--[index]
  - Positive Offset   *+R[index]
  - Negative Offset   *-R[index]

◆ 15-bit Positive/Negative Constant Offset from Either B14 or B15
C6000 Instruction Set Addressing

Features

◆ Circular Addressing
  - Fast and Low Cost: Power of 2 Sizes and Alignment
  - Up to 8 Different Pointers/Buffers, Up to 2 Different Buffer Sizes

◆ Dual Endian Support
C67x Architecture
TMS320C6701 DSP Block Diagram

Program Cache/Program Memory
- 32-bit address, 256-Bit data
- 512K Bits RAM

'C67x Floating-Point CPU Core
- Program Fetch
- Instruction Dispatch
- Instruction Decode
- Data Path 1
  - A Register File
  - L1, S1, M1, D1
- Data Path 2
  - B Register File
  - D2, M2, S2, L2

Control Registers
- Control Logic
- Test
- Emulation
- Interrupts

Power Down

Host Port Interface

4 Channel DMA

External Memory Interface

Data Memory
- 32-Bit address
- 8-, 16-, 32-Bit data
- 512K Bits RAM

External Memory Interface

2 Timers

2 Multi-channel buffered serial ports (T1/E1)
TMS320C6701
Advanced VLIW CPU (VelociTI™)

◆ 1 GFLOPS @ 167 MHz
  ■ 6-ns cycle time
  ■ 6 x 32-bit floating-point instructions/cycle
◆ Load store architecture
◆ 3.3-V I/Os, 1.8-V internal
◆ Single- and double-precision IEEE floating-point
◆ Dual data paths
  ■ 6 floating-point units / 8 x 32-bit instructions
TMS320C6701
Memory /Peripherals

◆ Same as ’C6201
◆ External interface supports
  ■ SDRAM, SRAM, SBSRAM
◆ 4-channel bootloading DMA
◆ 16-bit host port interface
◆ 1Mbit on-chip SRAM
◆ 2 multichannel buffered serial ports (T1/E1)
◆ Pin compatible with ’C6201
C67x Interrupts

- 12 Maskable Interrupts
- Non-Maskable Interrupt (NMI)
- Interrupt Return Pointers (IRP, NRP)
- Fast Interrupt Handling
  - Branches Directly to 8-Instruction Service Fetch Packet
  - 7 Cycle Overhead: Time When No Code is Running
  - 12 Cycle Latency: Interrupt Response Time
- Interrupt Acknowledge (IACK) and Number (INUM) Signals
- Branch Delay Slots Protected From Interrupts
- Edge Triggered
C67x New Instructions

**.L Unit**
- ADDSP
- ADDDP
- SUBSP
- SUBDP
- INTSP
- INTDP
- SPINT
- DPRINT
- SPTRUNC
- DPTRUNC
- DSPS

**.M Unit**
- MPYSP
- MPYDP
- MPYI
- MPYID
- MPY24
- MPY24H

**.S Unit**
- ABSSP
- ABSDP
- CMPGTSP
- CMPEQSP
- CMPLTSP
- CMPGTDP
- CMPEQDP
- CMPLTDP
- RCPSP
- RCPDP
- RSQRSP
- RSQRDP
- SPDP
C67x Datapaths

- 2 Data Paths
- 8 Functional Units
  - Orthogonal/Independent
  - 2 Floating Point Multipliers
  - 2 Floating Point Arithmetic
  - 2 Floating Point Auxiliary
- Control
  - Independent
  - Up to 8 32-bit Instructions
- Registers
  - 2 Files
  - 32, 32-bit registers total
- Cross paths (1X, 2X)

- L-Unit (L1, L2)
  - Floating-Point, 40-bit Integer ALU
  - Bit Counting, Normalization
- S-Unit (S1, S2)
  - Floating Point Auxiliary Unit
  - 32-bit ALU/40-bit shifter
  - Bitfield Operations, Branching
- M-Unit (M1, M2)
  - Multiplier: Integer & Floating-Point
- D-Unit (D1, D2)
  - 32-bit add/subtract Addr Calculations

THE WORLD LEADER IN DSP SOLUTIONS
C67x Instruction Packing
Instruction Packing Enhanced VLIW

Example 1

A B C D E F G H

Example 2

A B C D E F G H

Example 3

A B C D E F G H

- Fetch Packet
  - CPU fetches 8 instructions/cycle
- Execute Packet
  - CPU executes 1 to 8 instructions/cycle
  - Fetch packets can contain multiple execute packets
- Parallelism determined at compile/assembly time
- Examples
  - 1) 8 parallel instructions
  - 2) 8 serial instructions
  - 3) Mixed Serial/Parallel Groups
    - A // B
    - C
    - D
    - E // F // G // H
- Reduces
  - Codesize
  - Number of Program Fetches
  - Power Consumption
C67x Pipeline Operation

Pipeline Phases

Fetch  Decode  Execute

PG  PS  PW  PR  DP  DC  E1  E2  E3  E4  E5  E6  E7  E8  E9  E10

◆ Operate in Lock Step
◆ Fetch
  ■ PG  Program Address Generate
  ■ PS  Program Address Send
  ■ PW  Program Access Ready Wait
  ■ PR  Program Fetch Packet Receive

◆ Decode
  ■ DP  Instruction Dispatch
  ■ DC  Instruction Decode

◆ Execute
  ■ E1 - E5  Execute 1 through Execute 5
  ■ E6 - E10  Double Precision Only
C67x Pipeline Operation

Delay Slots

Delay Slots: number of extra cycles until result is:

- written to register file
- available for use by a subsequent instructions
- Multi-cycle NOP instruction can fill delay slots while minimizing codesize impact

**Most Integer**

- E1: No Delay

**Single-Precision**

- E1 E2 E3 E4: 3 Delay Slots

**Loads**

- E1 E2 E3 E4 E5: 4 Delay Slots

**Branches**

- E1

**Branch Target**

- PG PS PW PR DP DC E1: 5 Delay Slots
’C67x and ’C62x Commonality

- Driving commonality ( ) between ’C67x & ’C62x shortens ’C67x design time.
- Maintaining symmetry between datapaths shortens the ’C67x design time.

<table>
<thead>
<tr>
<th>C62x CPU</th>
<th>’C67x CPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>M-Unit 1 Multiplier Unit</td>
<td>M-Unit 1 Multiplier Unit with Floating Point</td>
</tr>
<tr>
<td>D-Unit 1 Data Load/Store</td>
<td>D-Unit 1 Data Load/Store</td>
</tr>
<tr>
<td>S-Unit 1 Auxiliary Logic Unit</td>
<td>S-Unit 1 Auxiliary Logic Unit with Floating Point</td>
</tr>
<tr>
<td>L-Unit 1 Arithmetic Logic Unit</td>
<td>L-Unit 1 Arithmetic Logic Unit with Floating Point</td>
</tr>
<tr>
<td>Register file</td>
<td>Register file</td>
</tr>
<tr>
<td>Decode</td>
<td>Decode</td>
</tr>
<tr>
<td>Program Fetch &amp; Dispatch</td>
<td>Program Fetch &amp; Dispatch</td>
</tr>
</tbody>
</table>
TMS320C80 MIMD MULTIPROCESSOR
Texas Instruments - 1996

Diagram of the TMS320C80 MIMD Multiprocessor with various components and connections labeled.

- PP (Processor Pair)
- MP/FPU (Multiprocessor/Fixed-Point Unit)
- Crossbar
- Shared RAM
- Instruction cache
- Video controller
- Display/capture
- Transfer controller
- 8/16/32/64

The World Leader in DSP Solutions

Texas Instruments