VLIW Architectures for DSP: A Two-Part Lecture

Berkeley Design Technology, Inc.

www.BDTI.com

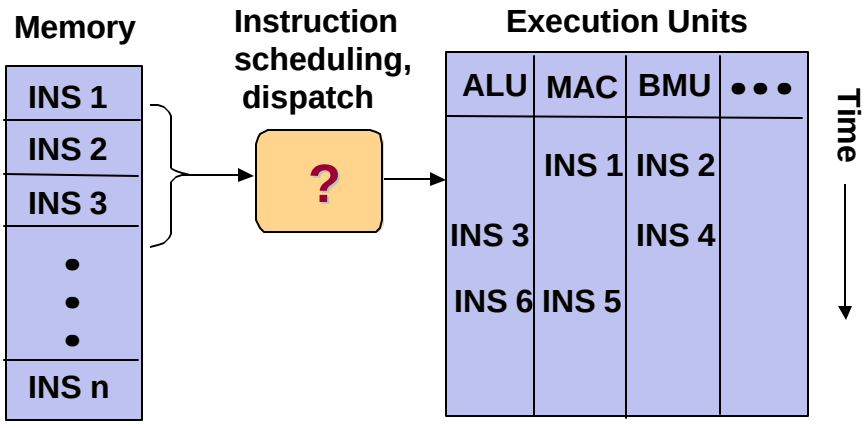
Outline

- **Part I: VLIW basics and a case study**
  - **What's VLIW?**
  - **Why VLIW?**
  - **The TMS320C62xx**
  - **Advantages, disadvantages of VLIW**
- Part II: Other VLIW DSP architectures
  - StarCore SC140
  - ADI TigerSHARC
  - Infineon Carmel

Why VLIW?

- Until ~1997, most DSP processors were very similar
  - Specialized execution units
  - Specialized instruction sets
    - Difficult to program in assembly
    - Unfriendly compiler targets
  - One instruction per instruction cycle
- VLIW architectures execute multiple instructions/cycle and use simple, regular instruction sets
  - More parallelism, higher performance
  - Better compiler targets

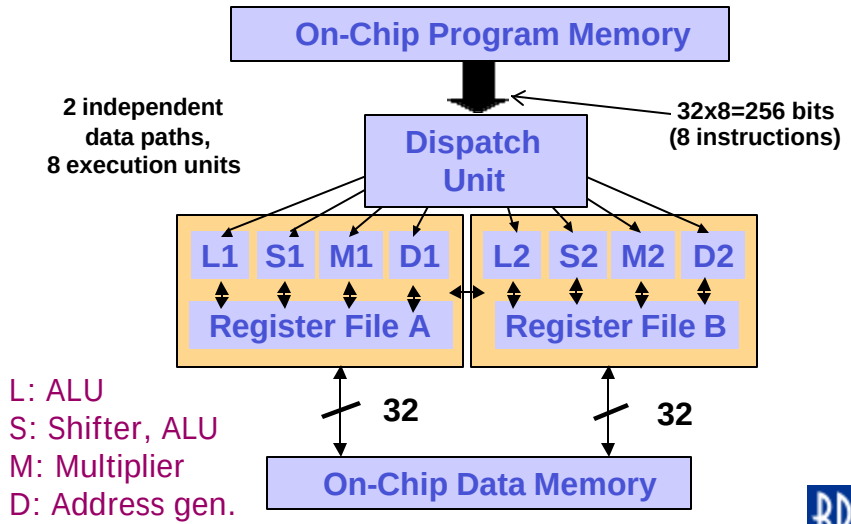
VLIW vs Superscalar



Characteristics of VLIW Processors

- Multiple independent instructions per cycle, packed into single large "instruction word" or "packet"
  - Instructions may be positional, or may include routing information within each sub-instruction
- Large complement of independent execution units
- More regular, orthogonal, RISC-like instructions
  - Usually wider than typical DSP instructions
  - Usually simpler than typical DSP instructions
- Large, uniform register sets
- Wide program and data buses

Example VLIW DSP: The TI TMS320C62xx



L: ALU
S: Shifter, ALU
M: Multiplier
D: Address gen.

FIR Filtering on the 'C62xx

Can execute up to eight 32-bit instructions in parallel →

```
LOOP:
  ADD    .L1 A0,A3,A0
||ADD    .L2 B1,B7,B1
||MPYHL  .M1X A2,B2,A3
||MPYLH  .M2X A2,B2,B7
||LDW    .D2 *B4++,B2
||LDW    .D1 *A7--,A2
|||[B0] ADD .S2 -1,B0,B0
|||[B0] B .S1 LOOP
```

Compare to a conventional DSP...

```
dotprod:
 MR=MR+MX0*MY0(SS), MX0=DM(I0,M0),MY0=PM(I4,M4);
```

Advantages of VLIW Architectures

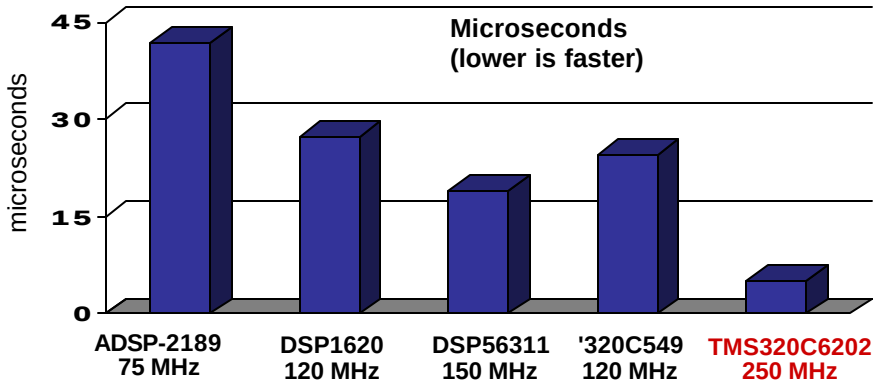
- Increased performance
- Better compiler targets
- Potentially easier to program
- Potentially scalable
  - Can add more execution units, allow more instructions to be packed into the VLIW instruction

Disadvantages of VLIW Architectures

- New kinds of programmer/compiler complexity
  - Programmer (or code-generation tool) must keep track of instruction scheduling
  - Deep pipelines and long latencies can be confusing, may make peak performance elusive
- Increased memory use
  - High program memory bandwidth requirements
- High power consumption
- Misleading MIPS ratings

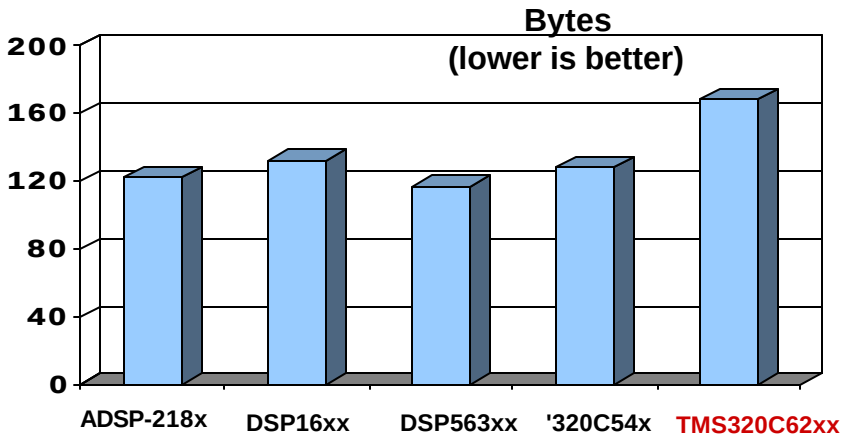
Benchmark Results

Execution Time on Complex Block FIR



Benchmark Results

Memory Usage on FSM Benchmark



For More Information...

Free resources on BDTI's web site,

www.bdti.com

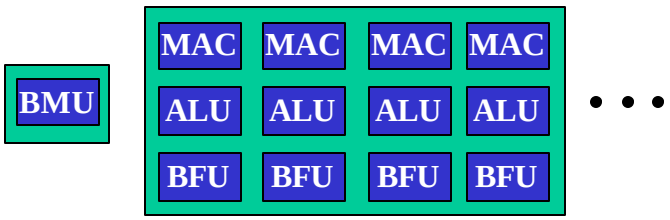
- *DSP Processors Hit the Mainstream* covers DSP architectural basics and new developments. Originally printed in *IEEE Computer* Magazine.
- *Evaluating DSP Processor Performance,* a white paper from BDTI.
- Numerous other BDTI article reprints, slides
- *comp.dsp* FAQ

Outline

- Part I: VLIW basics and a case study
  - What's VLIW?
  - Why VLIW?
  - The TMS320C62xx
  - Advantages, disadvantages of VLIW
- **Part II: Other VLIW DSP architectures**
  - **StarCore SC140**
  - **ADI TigerSHARC**
  - **Infineon Carmel**

StarCore SC140

- 16-bit fixed-point VLIW DSP core from Lucent/Motorola
- StarCore claims it's a scalable architecture
  - First VLIW machine to target low-power apps
- More execution units (13) than 'C62xx (8), but fewer instructions can be issued per cycle
  - Six for SC140 vs eight for 'C62xx

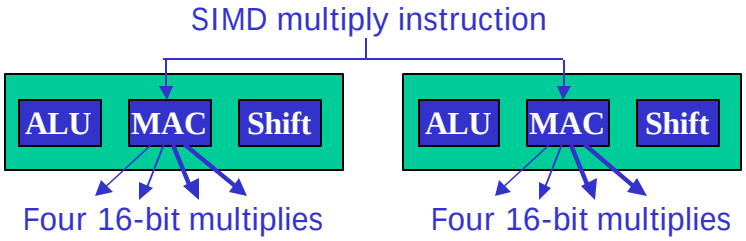


StarCore SC140

- Uses 16-bit instructions with optional 16-bit prefixes
  - Should have pretty good code density, better than 'C62xx ('C62xx uses fixed-width 32-bit instructions)
- Pipeline relatively simple and shallow (5 stages)
- Targeting 198 mW @ 300 MHz, 1.5 V
- Development chip expected late '99
- Lucent and Motorola will each create chips using the SC140 core
  - Motorola's MSC8101 sampling 1H00

ADI TigerSHARC

- 8-, 16-, 32-bit fixed-point *and* 32-bit floating-point
  - Unusual data-type agility
- Combines VLIW with extensive SIMD (single instruction, multiple data) to get massive parallelism
  - Using SIMD, can perform eight 16x16-bit fixed-point multiplications per cycle (4X the 'C62xx)



ADI TigerSHARC

- "Hierarchical" SIMD is unusual
- Requires high on-chip data memory bandwidth
  - Sixteen 16-bit data words/cycle
- Issues and executes up to four instructions per cycle
- Uses 32-bit instructions, like '62xx
  - May have high program memory use
  - Memory use may also be increased by data- and algorithm-rearrangement needed for use of SIMD
- Targeting 250 MHz, expected to begin sampling in late 1999

Infineon Carmel

- 16-bit fixed-point VLIW DSP core from Infineon (Siemens)
  - In silicon at 120 MHz, 0.25 µm (development chip)
- Two data paths, six execution units

- Mixed-width 24/48-bit instruction set
- Can execute in parallel:
  - One 48-bit instruction, or
  - One or two 24-bit instructions, or
  - Up to six instructions as part of a "CLIW"

Infineon Carmel
CLIW (Configurable Long Instruction Word)

General format:

```
cliw name (operand1, ... , operand 4) {
  ALU1 || MAC1 || ALU2 || MAC2 || MOV1 || MOV2
}
```

Example CLIW:

```
cliw fft4(r0+=rn0, r1, r4, r5) {
           a2 = a1l * a0h
       || *ma1 = ff1 + ff2
       || *ma2 = a2 - a1h * a0h
       || a1h = *ma3 - *ma4
       || ff1 = *ma3
       || ff2 = *ma4;
}
```

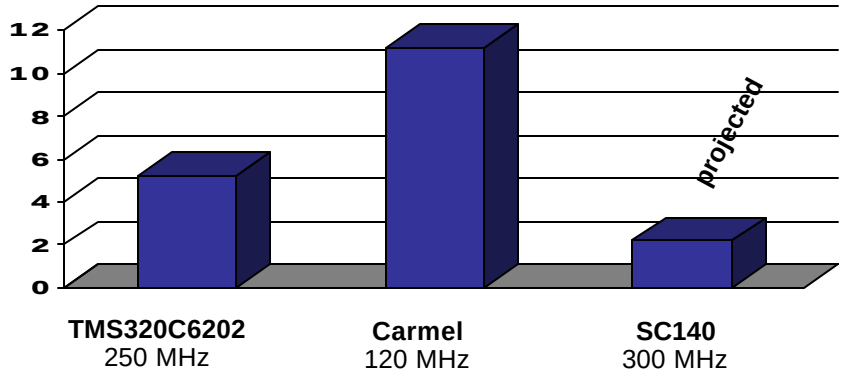

Comparison

Processor	Issue width	Data memory bandwidth (16-bit words)	Instruction size	Clock (MHz)	Pipeline depth	Notable characteristics
TMS320C62xx	8	4 words/cycle	32 bits	250	11	1st VLIW-based DSP processor
SC140	6	8 words/cycle	16 bits w/ 16-bit prefixes	300*	5	Scalable, approach to compact code
TigerSHARC	4	16 words/cycle	32 bits	250*	8	SIMD + VLIW, data type agility
Carmel	2, 6	4 words/cycle	24/48 bits	120	8	CLIW instructions, 4 AGUs

*Projected

Benchmark Results

Execution Time on Complex Block FIR



For More Information...

Free resources on BDTI's web site,

http://www.BDTI.com

- *DSP Processors Hit the Mainstream* covers DSP architectural basics and new developments. Originally printed in *IEEE Computer* Magazine.
- *Evaluating DSP Processor Performance,* a white paper from BDTI.
- Numerous other BDTI article reprints, slides
- *comp.dsp* FAQ