Embedded digital systems architecture

Edward A. Lee
H. John Reekie

Department of EECS
U. C. Berkeley

Why not just use a RISC chip?

What is RISC?

"Any microprocessor architecture designed after 1986."

- Load/store architecture
- 32, 64b IEEE fp add, multipy, divide, sqrt
- Elaborate exception handling
- MMU for memory hierarchy
- TLB for virtual memory (also protection)
- "Low power" means about 2 watts (vs. 10)

Why use Programmable DSPs?

What is a programmable DSP?

"Any microcomputer still programmed in assembly language"

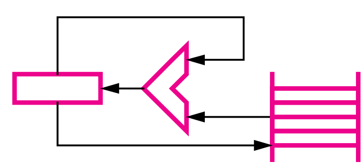
- Can be small, cheap.
- "Marching band" timing.
- Little or no memory hierarchy
- Low power means 50mW
- Low system cost.

Fixed-point vs floating-point

"Generation gap"

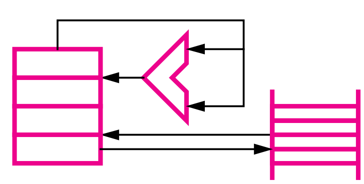
Fixed-point

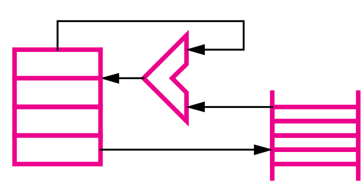
- Accumulator architecture
- Fewer registers
- 16 or 24-bit integer data
- Cheaper



Floating-point

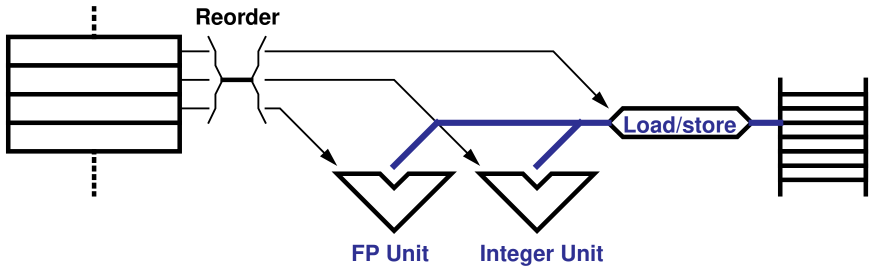
- Load/store or memory-register architecture
- More registers
- 32-bit integer or floating point (IEEE or native) data
- More expensive
- Good compilers (C, C++, Ada)



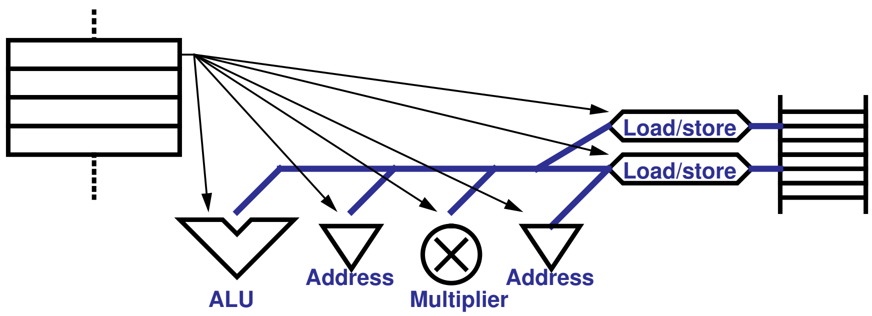


RISC vs DSP: Instruction encoding

RISC: Super-scalar

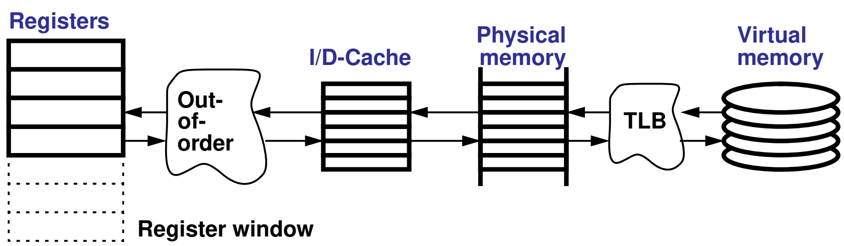


DSP: Horizontal microcode

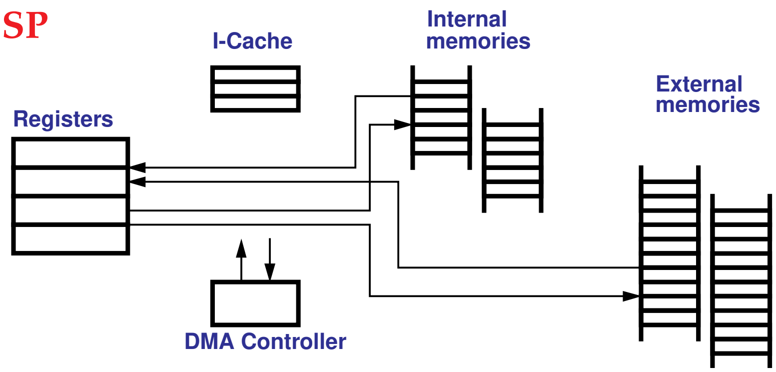


RISC vs DSP: Memory hierarchy

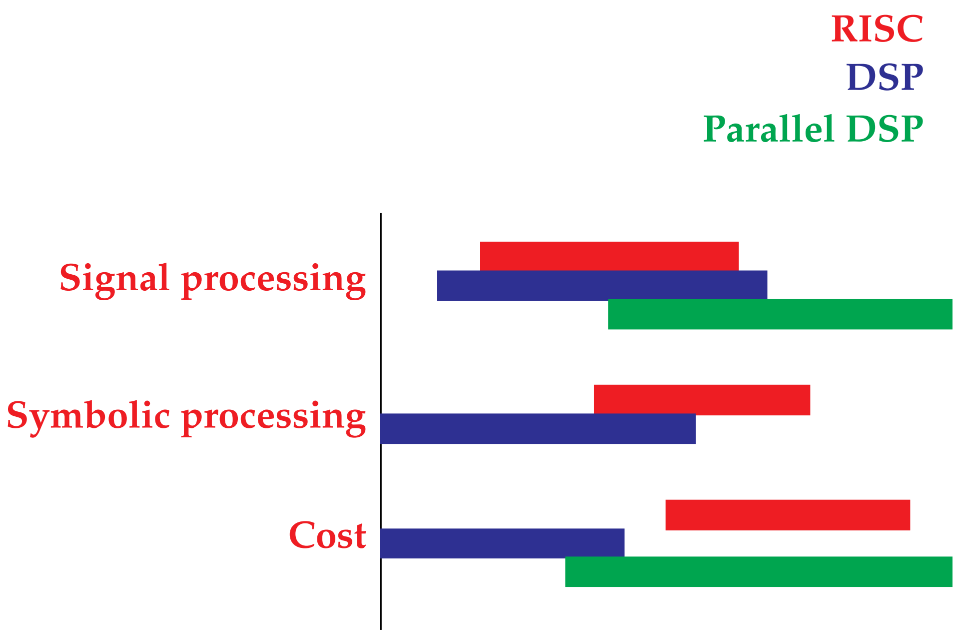
RISC



DSP



RISC vs DSP: Performance and cost



Addressing Modes

	TMS320C50	TMS320C30
• immediate		
• The operand is part of the instruction	ADD #0FFh	addi 32,r0
• register		
• The operand is in a specified register	(implied)	addi r1,r0
• direct		
• The address of the operand is part of the instruction (added to implied memory page)	ADD 010h	addi @count,r0
• indirect		
• The address of the operand is stored in a register	ADD *	addi *+ar2(7),r0
• Often used with indexing or pre-/post-decrement		addi *ar2++,r0

Fixed-point arithmetic

Decimal:

$$\begin{array}{r} 99.999 \\ +\ 9.99 \\ \hline =109.989 \end{array} \qquad \begin{array}{r} 99.999 \\ \times\ 9.99 \\ \hline =998.99001 \end{array}$$

Binary:

$$\begin{array}{r} 0.111 \\ +\ 0.111 \\ \hline =1.110 \end{array} \qquad \begin{array}{rl} 0.111 & 7/8 \\ \times\ 0.111 & 7/8 \\ \hline =00.110001 & 49/64 \end{array}$$

$$\begin{array}{rl} 0.111 & 7/8 \\ \times\ 1.000 & -8/8 \\ \hline =11.111000 & -56/64 \end{array}$$

$$\begin{array}{rl} 1.000 & -8/8 \\ \times\ 1.000 & -8/8 \\ \hline =01.000000 & 64/64 \end{array}$$

Except for one special case, the sign and integer bit are always the same, so a fix-point multiply means:

- **Multiply two N-bit words**
- **Shift 2N-bit product left one bit**
- **Take high-order N bits**

Some examples of products with prog. DSPs

Company	Product	Description	DSP
Digicom	Various Modems	V.32, V.32bis, V.22bis, Fax	ADSP-21xx
Ericsson	Hotline GH197	GSM digital cellular phone	ADSP-2102
E-mu	Proteus MPS	Music effects	ADSP-2105
Intel	ActionMedia II	DVI product series	ADSP-2105
Peavy Electr.	Spectrum Bass Tone Module	Sound synthesis	ADSP-2105
Peavy Electr.	IDL 1000	Digital delay for audio	ADSP-2105
Sharp	JY-7500 MOD Drive	Magneto optical disk drive (servo loop)	ADSP-2101
Siemens	NNSR	Neural net speech recognizer	ADSP-2111
Xing	VT-Compress	JPEG/MPEG image compression	ADSP-2105