Processor Design
–
DSP Processors

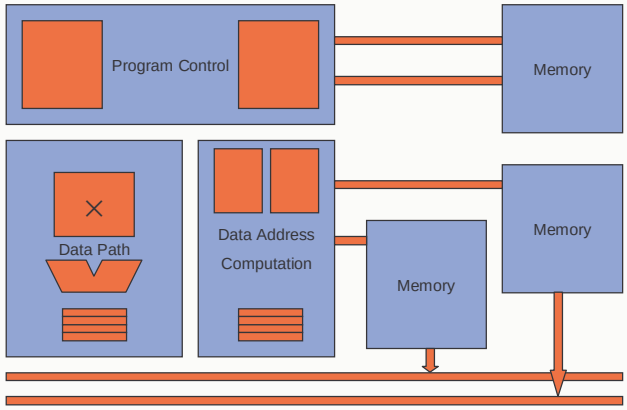
Professor Jari Nurmi
Institute of Digital and Computer Systems
Tampere University of Technology, Finland
email jari.nurmi@tut.fi

Digital Signal Processors

- DSPs optimized for digital filtering and analysis
- Basic operation in digital filters is multiply-accumulate (MAC)
- Highly constrained, non-orthogonal architectures
- Complex, ”compound” instructions encoding many operations
- 16-, 24-, 32-bit instructions, single-cycle execution
- Word length of 16 to 24 bits fixed-point (fractional)
  - 16 bits equals about 96 dB dynamic range of signals
- Dedicated addressing hardware with specialized addressing modes
  - modulo addressing to implement circular buffers for filters
  - bit-reversal addressing for FFT addressing
- Multiple-access on-chip memory architecture (modified Harvard architecture)
- Dedicated hardware for loops and other execution control
- Low cost, low power, low memory usage

Digital Signal Processors (cont'd)

- MAC architecture
- multiple access
- dedicated address HW
- dedicated loop HW
- non-orthogonality
- complex instructions



Enhanced DSPs

- More parallelism
  - e.g., 2nd multiplier, adder (dual MAC)
  - limited SIMD operations
- Highly specialized hardware in core
  - application-oriented data path operations
- Co-processors
  - Viterbi decoding
  - FIR filtering
  - etc.

What do you say to someone with a PhD in signal processing?

What do you say to someone with a PhD in signal processing?

”Two big macs please.”

Enhanced DSPs (cont'd)

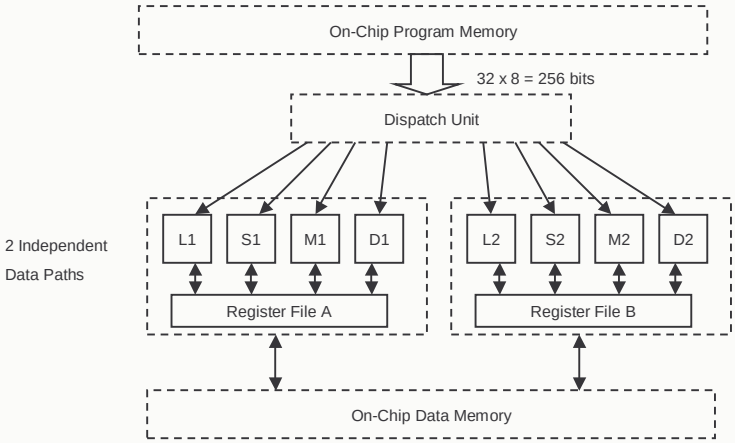
- Advantages
  - Allows incremental performance increases while maintaining competitive cost, power, code density
  - Compatibility is possible; similarity is likely
- Disadvantages
  - Increasingly complex, hard-to-program architectures
  - Poor compiler targets
  - How much further can we get with this approach? (scalability)

VLIW in DSP

- VLIW coming to DSP era in high-performance applications
  - TI TMS320C6xxx, Infineon Carmel, ADI TigerSHARC, Philips TriMedia
- Characteristics
  - Multiple independent operations per cycle, packed into single large ”instruction” or ”packet”
  - More regular, orthogonal, RISC-like operations
  - Large, uniform register sets

VLIW in DSP (cont'd)

- Texas Instruments C6xxx architecture



VLIW in DSP (cont'd)

- Advantages
  - Increased performance
  - More regular architectures
  - Scalable (?)
- Disadvantages
  - New kinds of programmer/compiler complexity
    - Programmer (or code generation tool) must keep track of instruction scheduling
    - Deep pipeline, long latencies can be confusing and may make peak performance elusive
  - Code sized bloat
    - High program memory bandwidth requirements
  - High power consumption

Superscalar DSP

- Superscalar architectures for DSP applications
  - LSI Logic (ZSP), Infineon TriCore
- Characteristics
  - Borrow techniques from high-end CPUs
  - Multiple (usually 2-4) instructions issued per instruction cycle
  - RISC-like instruction set
  - Lots of parallelism

Superscalar DSP (cont'd)

- Advantages
  - Large jump in performance
  - More regular architectures (better compiler targets)
  - Programmer (or code generation tool) doesn't have to worry about instruction scheduling
  - Code size not increased significantly
- Disadvantages
  - Energy consumption a major challenge
  - Dynamic behavior complicates software development
    - Execution-time variability can be a hazard
    - Code optimization is challenging

Embedded Processors for DSP

- General purpose embedded processors start to address DSP needs
  - e.g., Hitachi SH-DSP, ARM Piccolo, Infineon TriCore
- Various approaches
  - Integrate fixed-point DSP data path and related resources with an existing uC core (SH-DSP)
  - Add a DSP co-processor to existing uP core (Piccolo)
  - Creare an all-new hybrid architecture (TriCore)

Embedded Processors for DSP (cont'd)

- Advantages
  - Reasonable DSP performance
  - Cost/performance competitive with fixed-point DSPs
  - Upgrade path from existing embedded cores
  - Potential benefits in size and cost
- Disadvantages
  - Compromizes show
    - Programming complexity
    - Performance penalties
  - Starting with very limited DSP infrastructure (code, tools, ...)

DSP Design Choices

- Three major (interacting) domains of choices
  - Datapath
  - Data address generation
  - Control
- Final targets include
  - Small die size
  - Short cycle time
  - Power consumption
  - Good code density (as in embedded)
  - Powerful instructions
  - Development support (tools)
  - (As) easy (as possible) to program
  - Good compiler target

DSP Design Choices (cont'd)

- Datapath
  - word length
  - integer/fraction
  - signed/unsigned
  - saturating/non-saturating
  - rounding modes
  - register file/accumulator(s)
  - number of registers/accumulators
  - MAC pipelining?
  - barrel shifter?
  - special ALU operations?
  - multiple units?

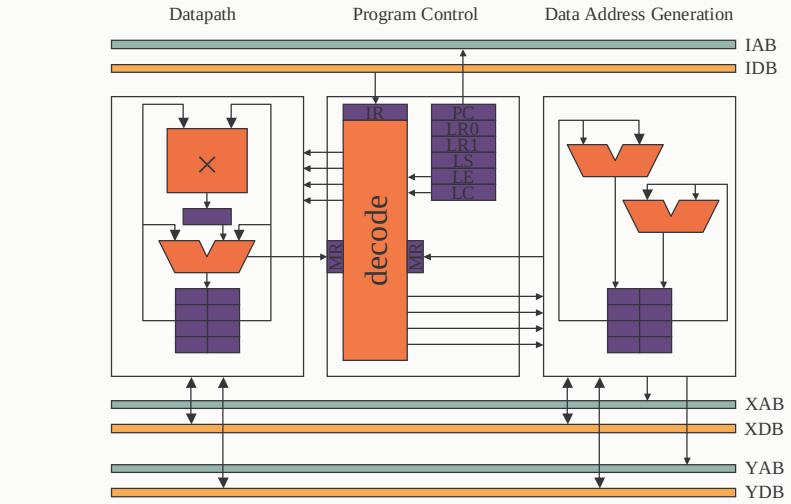
DSP Design Choices (cont'd)

- Data address generation
  - address space
  - addressing modes supported
  - number of data buses (and computation units)
  - post-modify mechanisms
  - address register file size
  - unified/banked register file
  - register-register moves?

DSP Design Choices (cont'd)

- Control
  - Instruction set
  - Instruction word length
    - instruction coding style
    - parallelism supported
    - single-issue/superscalar/VLIW?
  - Instruction address space
  - Conditions
  - Loop hardware
    - Existence
    - Levels of hardware-supported loops
    - Mechanisms to extend from HW to SW loop control
  - Branch/link mechanism
  - Pipeline stages
    - Delay slots, load delay, condition evaluation delay
    - Cycle time

DSP Example: VS_DSP



VS_DSP Instruction Coding

- Four types of instructions:
  - immediate loading

opcode	immediate	target register

  - control

opcode	control code

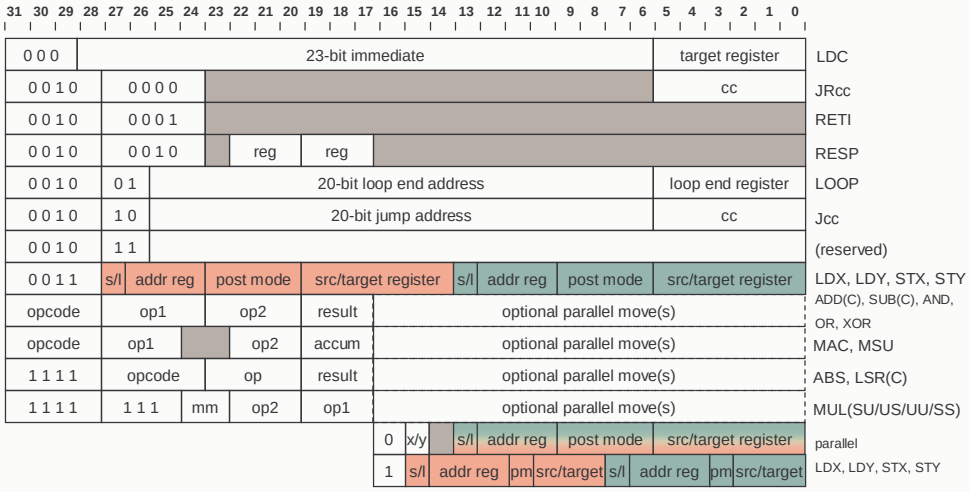
  - moves

opcode	X full move	Y full move

  - arithmetic with 2 parallel moves

opcode	arithmetic operands	parallel moves

VS_DSP Instruction Coding (Cont'd)



VS_DSP Datapath Register Coding

Binary code	register	composition
0000	A0	S:A0:0000
0001	A1	S:A1:0000
0010	B0	S:B0:0000
0011	B1	S:B1:0000
0100	C0	S:C0:0000
0101	C1	S:C1:0000
0110	D0	S:D0:0000
0111	D1	S:D1:0000
1000	NULL	0:0000:0000
1001	ONES	F:FFFF:FFFF
1010	(reserved)	(reserved)
1011	P	S:P1:P0
1100	A	A2:A1:A0
1101	B	B2:B1:B0
1110	C	C2:C1:C0
1111	D	D2:D1:D0

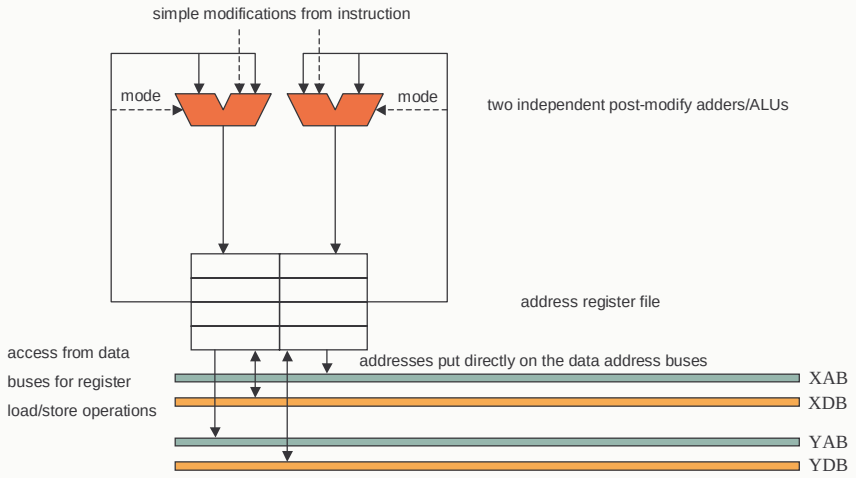
Externally visible
(e.g. in data moves)

Only internally available
for ALU operations

VS_DSP Data Address Calculation

- Addressing modes
  - register indirect
  - register indirect with linear post-modification
  - register indirect with modulo post-modification
  - (register indirect with bit-reversal post-modification)
- AM indicated in instruction
- Simple linear post-inc/dec within ±7 from instruction
- More complex addressing based on register usage (code ”–8”)
  - address – modifier register pairs
  - modifier register declares addressing mode
    - linear
    - modulo
    - bit-reversal
  - and the number to be added to the address register in post-modification

VS_DSP Data Address Calculation (Cont’d)



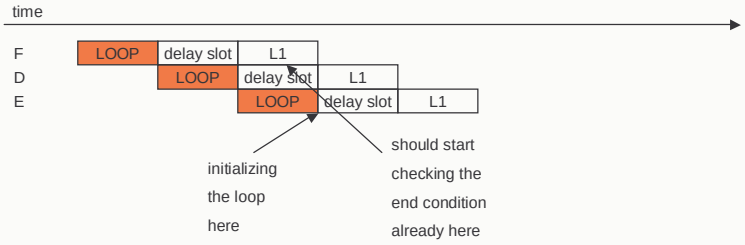
VS_DSP Loop Control

- Loop instruction is of the form

  LOOP *count_reg, end_address*
- The loop hardware includes registers LC, LS, LE
- In loop initialization, the hardware puts
  - *count_reg* → LC
  - *end_address* → LE
  - PC+2 → LS
- Note: a delay slot between LOOP instruction and the loop body
- Always when PC reaches LE
  - LC is decremented and tested for zero
  - LS is loaded to PC next, if the count did not hit zero

VS_DSP Loop Control (Cont'd)

- Looking at the pipeline diagram, one (or two) problem(s) can be noticed
  - loop control of one-instruction loops
  - loop control of loops executed only once (exiting when entering)
  - zero times looping is not allowed for simplicity, always run once



- Solution: steal at least half-a-cycle in loop initialization (tap out the LC, LE and LS already after the master latch of a MS-FF)
- The HW has ½ cycle for decrement, zero-comparison and PC-input-mux control

VS_DSP Interrupts

- In the original (VS_DSP1) version the interrupts are simple
  - only one interrupt line to the core
  - external device takes care of priorization, masking, etc. if required
  - a (memory-mapped) register at the external interrupt controller for resolving the interrupt source
  - interrupts will not occur
    - between a change-of-flow instruction and its delay slot
    - immediately before the loop end address
  - LR1 is used for storing the return address, and the mode/status bits from MR0 are stored in MR1
  - interrupt service must first store LR1 and MR1, if the routine is more than 4 instructions
    - 2 instructions are quaranteed to be executed before next interrupt can occur, to be able to save the state information if needed
    - RETI and its delay slot block the interrupts for another 2 cycles
  - at the end of service, LR1 and MR0 (!) are restored immediately before RETI instruction
  - the interrupts can be nested to any depth

VS_DSP Characteristics

- All single cycle 32-bit instructions
- Data word length 16 bits (nominal), optionally 12-32 bits
- Parameterized features (register counts, modes, looping)
- Extension capabilities
- Configurable basic tools (assembler, linker, ISS)
- C-compiler, emulator board, RTOS for 16-bit core
- Power consumption below 0.1 mW/MIPS @1.8V
- Core size below 2 mm^2 in conventional 0.35 µm CMOS
- Clock rate at 50 MHz in 0.6 µm @5V, approaching 100 MHz in 0.35 µm @3.3V (16-bit basic core)
- Hard core and synthesizable (soft) versions
- Peripherals and on-chip memory available

VS_DSP Later Improvements

- Register-to-register transfers
- Subroutine CALL instruction (previously jump + moving the return address to LR0 in the delay slot, now automated)
  - these two changes imply about 5% saving in program memory
- Vectored interrupts
  - up to 32 vectored interrupts, 3 priorities for each, enable/disable separately for each source or for all
  - interrupt latency decreased to a miminum of 7 instructions (<50% of original)
- 32-bit program addresses by adding a page register
  - affects branch and call mechanisms (near/far addresses)
- 32-bit external data addresses by concatenating two address registers (normally for separate X and Y access)
- More low-power features
  - HALT instruction for freezing the clocking/pipeline
  - input latching, clock gating, static implementation allowing F_c adjustment
  - low-voltage operation

Summary

- Basic architecture of all DSPs similar
  - MAC-based datapath
  - Harvard memory architecture
  - DSP-specific addressing
- Differences start to show in
  - enhanced features added
  - VLIW or superscalar architectures
  - mixed DSP/control processors
- However, a lot of choices in the design of (even a traditional DSP)
  - datapath
  - data address computation
  - control

End of DSP Processors

next we will look at performance measures