The CLIPPER® C4 Chipset

A Superpipelined, Superscalar Processor

Howard Sachs
Intergraph APD

Hot Chips Symposium
August 21, 1990

Agenda

- Definitions/Issues
- Objectives
- Architecture
- Technology
- Analysis of Superpipelining and Superscalar Operation on:
  - Integer Codes
  - Scalar Floating-Point Codes
  - Vector Floating-Point Codes
- Summary

Definitions/Issues

Definitions

- Superscalar Machines
  - Issue More than One Instruction per Cycle
- Superpipelined Machines
  - Issue One Instruction per Cycle
  - Cycle Times Shorter than the Latency of Any Functional Unit

Definitions/Issues (cont.)

Issues

- Instruction Level Parallelism
- Superscalar and Superpipelined Are Equal, BUT:
  - Hard to Build a General Superscalar Machine, so Both Are not Equal
  - In Fact, They Are Orthogonal
- Integer Problems Have Little Instruction Level Parallelism; Superpipelining Is No Help
- Scalar and Vector Codes Do Have Instruction Level Parallelism;
  Superpipelining and Superscalar Operations Are Very Helpful for Both Codes

Objectives

- High Performance
  - Single-Cycle Integer Execution
  - Superpipelined Floating-Point Function Units
  - Superscalar Floating-Point and Integer Operations
  - 50 MHz Clock Rate
- Scalable Architecture
  - 50 MHz at 1 μ
  - Introduce at 40 MHz
  - Scalable to 80 MHz at 0.5 μ

Objectives (cont.)

- Low Cost
  - Two Chips at 1 μ
    CPU and FPU < $1,000
  - One Chip at 0.5 μ
    Includes CPU, FPU, and Cache
- Provide Binary-Compatible Upgrade Path for Existing CLIPPER Applications
- Samples 4Q90

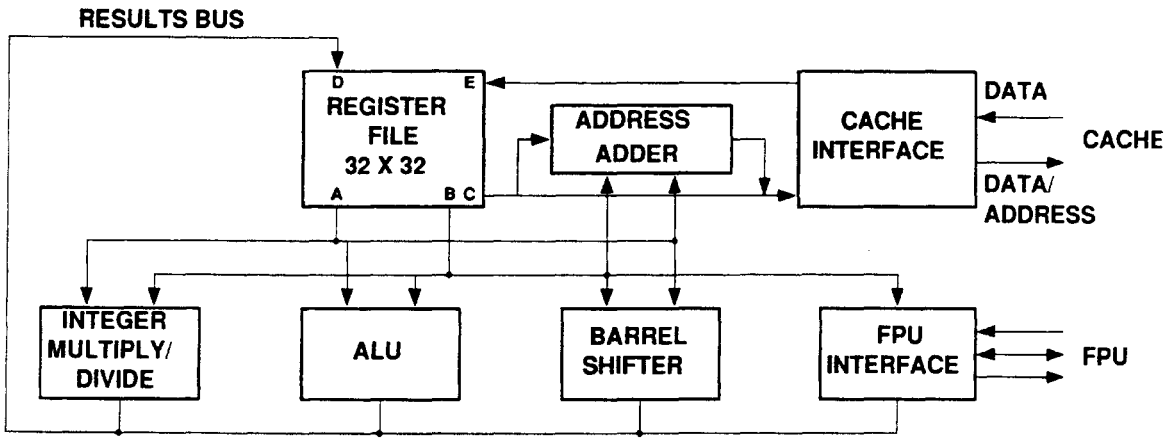
Integer Unit

- Most Instructions Execute in One Cycle
- CPU Issues Integer and Floating Instructions in Same Clock Cycle (Superscalar Operation)
- Five-Port Integer Register File
- Dedicated Address Adder
- Pipelined Loads/Stores
- Delayed-Branch Instruction with Static Branch Prediction

Integer Unit Instruction Timing

FUNCTIONAL UNIT	ISSUE (SAME FU)	ISSUE (DIFF FU)	EXECUTION LATENCY
ADD/SUBTRACT, LOGICAL, SHIFT	1	1	1 - 2
LOAD	1	1	2
STORE	1	1	1
BRANCH	3	1	3
MULTIPLY	4 - 6	1	4 - 6

Integer Unit



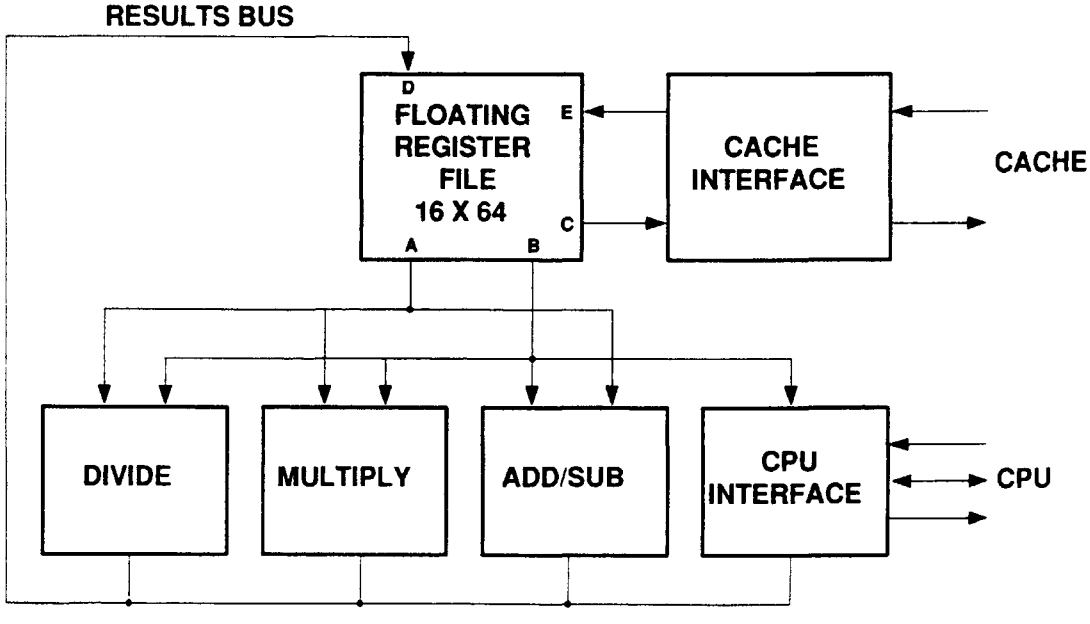
IEEE Floating-Point Unit

- Integer and Floating-Point Operations Issued in Same Clock (Superscalar Operation)
- Five-Port Floating-Point Register File
- 64-Bit Paths On and Off Chip
- Superpipelined Function Units for Add/Subtract and Multiply

IEEE Floating-Point Unit Instruction Timing

FUNCTIONAL UNIT	ISSUE	EXECUTION LATENCY
ADD/SUBTRACT (S/D)	1/1	4/4
MULTIPLY (S/D)	1/2	5/6

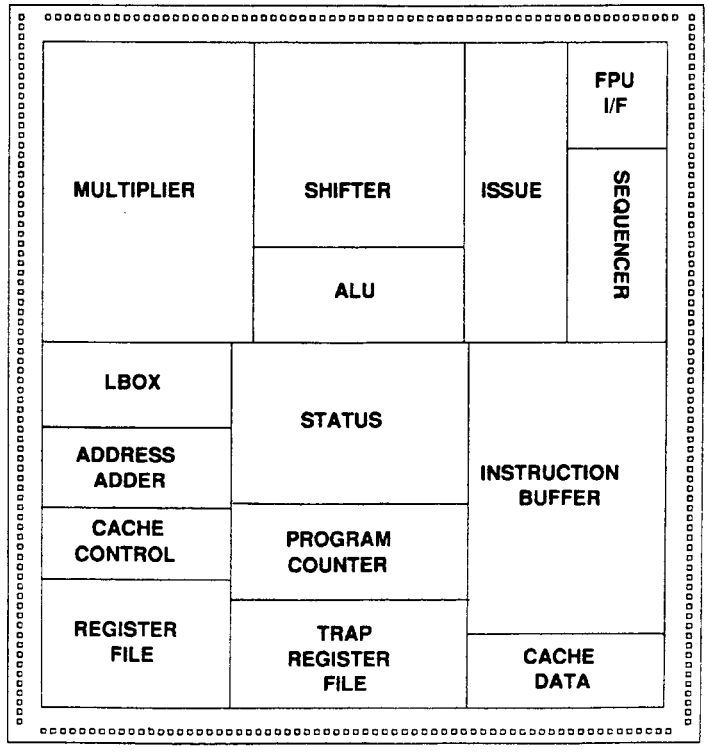
IEEE Floating-Point Unit



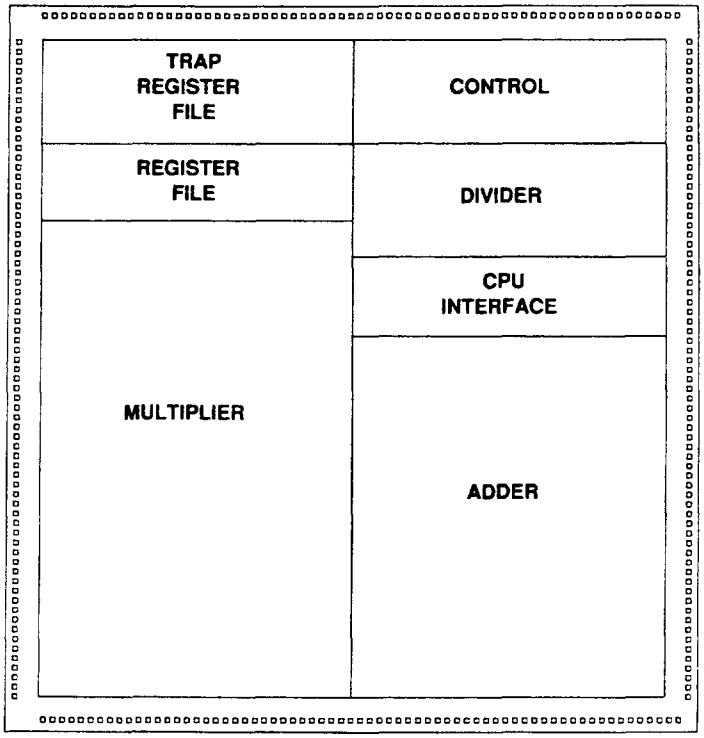
Semiconductor Design Rules

- 1 μ CMOS (5 Volts)
- Single Poly/Double Metal
- 1 μ Poly Width, 2 μ Pitch
- 1 μ Contacts/Vias
- Metal Pitch < 4 μ

Integer Unit Floor Plan



FPU Floor Plan



Chip Characteristics

- CPU
  - Die Size 220K Sq. Mils
  - 160,000 Transistors
  - 299-Pin PGA, 224 Signal Pins
- FPU
  - Die Size 260K Sq. Mils
  - 140,000 Transistors
  - 299-Pin PGA, 176 Signal Pins

Analysis of Superpipelining and Superscalar Operation on:

- Integer Code: Espresso
- Scalar Floating Point: Spice
- Vector Floating Point: Linpack

Examples Are Drawn from Small Segments of SPEC Benchmarks and Are Believed to be Representative of the Classes of Programs Described.

Integer Code: Espresso

- Code Evaluates Many Boolean and Bit-Wise Expressions
- No Floating Point
- Few Opportunities for "Instruction-Level Parallelism"
- Requires 23 Instructions
- Executes in 23 Clocks
- CPI = 1.0
- Superpipelining and Superscalar Issue Provide No Assistance

Scalar Code: Spice

- Code Contains Many Independent Sub-Expressions
- Some Opportunity for Instruction Level Parallelism
  - 42% of Operations Are Load/Stores
  - 42% of Operations Are Adds
- 27% of Instructions Can Issue Simultaneously
  - Load/Store and ADDF
- Code Structures Easily Pipelined

Scalar Code: Spice (cont.)

SUPER-PIPELINE	SUPERSCALAR	INSTRUCTION COUNT	# CLOCKS	CPI	MFLOPS @ 50MHz	% SPEED-UP
NO	NO	21	24	1.14	16.7	0
NO	YES	21	23	1.10	17.4	4
YES	NO	21	21	1.00	19.0	14
YES	YES	21	19	0.90	21.1	26

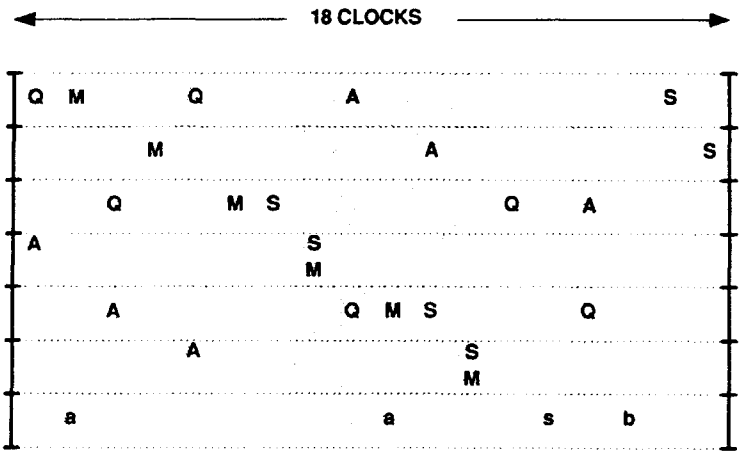
Vector Code: Linpack (Daxpy)

- Loads and Stores Comprise 60% of Daxpy Inner Loop (L - L - M - A - S)
- Many Opportunities for Instruction Level Parallelism
- Superscalar Operation Allows Load/Store Issue Concurrently with Multiply or Add
- `LDD2` Key to Increased Issue Bandwidth

Vector Code: Linpack (Daxpy) (cont.)

SUPER-PIPELINE	SUPERSCALAR	INSTRUCTION COUNT	# CLOCKS	CPI	MFLOPS @ 50 MHz	% SPEED-UP
NO	NO	34	36	1.05	16.7	0
NO	YES	34	36	1.05	16.7	0
YES	NO	34	28	0.82	21.4	28
YES	YES	34	18	0.53	33.3	100

Impact of Superscalar on "Daxpy"



LOOP UNROLLED 6 TIMES
34 INSTRUCTIONS
12 FLOPS
18 CLOCKS

a	REGISTER ADD
s	REGISTER SUBTRACT
S	STORE
M	FLOATING-POINT MULTIPLY
A	FLOATING-POINT ADD
Q	LDD2 INSTRUCTION
b	BRANCH

Performance Summary

CLASS	CPI*	MFLOPS @ 50MHz	STRATEGY
INTEGER CODES	1.00	N/A	HIGH PERFORMANCE THROUGH SINGLE-CYCLE EXECUTION RESULTING IN LOW CPI. SUPERPIPELINING AND SUPERSCALAR NO HELP.
SCALAR FLOATING POINT	0.90	21.1	HIGH PERFORMANCE THROUGH SUPERSCALAR AND SUPERPIPELINING. 26% IMPROVEMENT.
VECTOR FLOATING POINT	0.53	33.3	HIGH PERFORMANCE THROUGH SUPERSCALAR AND SUPERPIPELINING. 100% IMPROVEMENT.

* FROM SMALL KERNELS PREVIOUSLY SHOWN