

↳

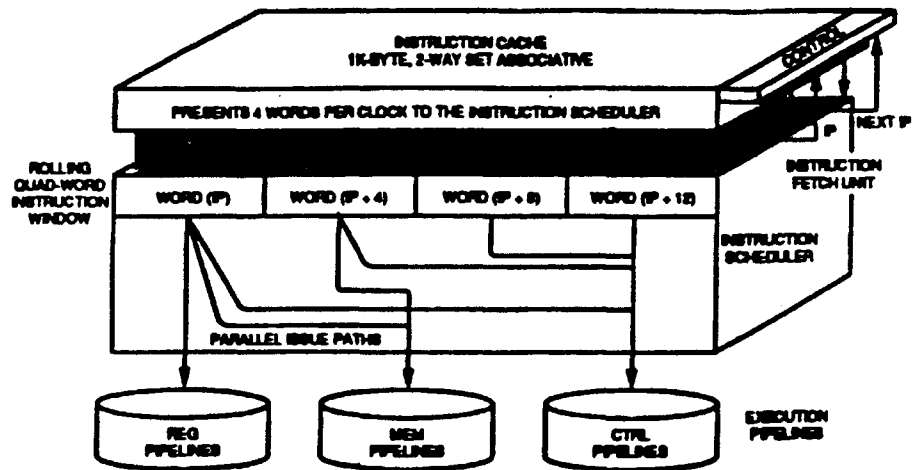
Performance Characteristics of the i960 CA SuperScalar Microprocessor

**S. McGeady
Intel Corporation
Embedded Microprocessor Focus Group**

i960 CA — History

- A highly-integrated microprocessor for embedded control
- Overall performance from 20 - 30 VUPS @ 33 MHz
- Second-generation 960 silicon
- Introduced September 1989
- Working silicon since June 1989
- First microprocessor to implement superscalar execution

Parallel Instruction Dispatch



i960CA Micro-Architecture

- **6-ported register file with Wide on-chip buses**
 - Buses provide aggregate bandwidth of over 2 Gigabytes/sec
- **Parallel Instruction Dispatching, with branch prediction**
- **Independent Execution Units for Integer, Multiply/Divide, Bus, ...**
- **Second ALU in Address Generation Unit**
- **On-Chip Instruction Cache**
- **On-Chip SRAM and Local Register Cache**

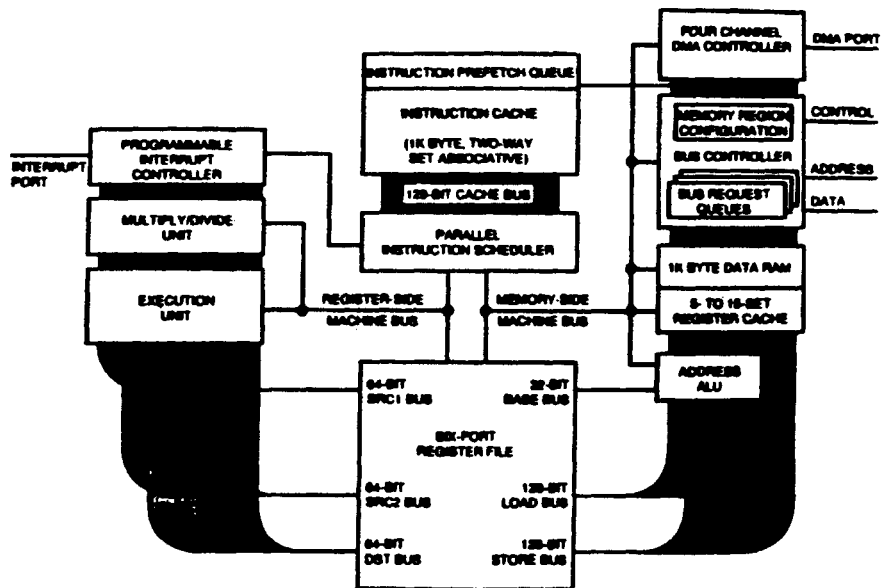
Benefits of Superscalar Execution

- Theoretical Performance Improvement
- i960CA Micro-Architecture Review
- Measured i960CA Performance Improvement
- Silicon Cost of Superscalar Execution
- Performance Enhancements over other RISCs
- Realizing Superscalar's Full Potential
- Compiler Issues

Theoretical Performance Improvements

- "order 3" superscalar shows theoretical 1.5x - 2.5x over base case
 - 960CA is less than an order-3 machine
- resource conflicts and data dependencies reduce observed benefit
 - actual and compiler-induced dependencies
- long memory latency reduces observed benefit
 - instruction fetch
 - other artificial resource conflicts

i960CA Micro-Architecture



Measured i960CA Performance Improvement

- i960 CA measured in *No Imprecise Faults* (NIF) mode
- Guarantees precise faults, disables parallel instruction dispatch
- Load/store scoreboard, other pipelining continues

Measured i960CA Performance Improvement

	Superscalar	Scalar	speedup
sieve1	1460	1645	1.12
matrix mul	1873	2969	1.58
blit	6572	8057	1.22
ocr	2351	2935	1.24
goem mean			1.25

- 33 MHz processor delivers 40MHz worth of performance

Silicon Cost of Superscalar Execution

- instruction cache control
 - more complex for multiword access
- instruction decoder
 - parallel instruction decode & issue
- function units
 - AGU mostly redundant with ALU
- register file ports
 - maybe would have had 3 instead of 6 - 1/3 bigger
- wider buses
 - reduction in reg file ports would shrink on-chip bus

Per-Unit Estimate of Superscalar Cost

Unit	% of Die	% Affected	% Die Impact
IFU/ICache	5.0%	10%	0.5%
RF	7.0	33	2.3
ID/PSeq	7.5	25	1.9
On-chip Buses	15.0	33	5.0
SRAM	5.0	-	-
ROM	3.5	-	-
MDU	7.5	-	-
ALU	5.0	-	-
BCL/DMA	25.0	-	-
AGU	3.0	90	2.7
Fault/Debug	7.0	-	-
Intr	2.5	10	0.2
Misc	10.0	-	-
Total			12.6

- 25% performance improvement for 13% cost

Performance v. First Generation (960KA)

	33 Mhz CA	33MHz KA	KA v. CA
sieve1	1460	4561	2.12x
matrix mul	1873	4927	1.63x
blit	6572	22706	1.67x
ocr	2351	11704	2.77x

Benchmark Notes

- i960CA benchmarks run on 33Mhz, Ows internal demo board
- i960KA benchmarks run on 20Mhz, Ows QT960 board, scaled
- All benchmarks compiled with gcc960, V1.2 of 7/90, -O3 optimization
- Benchmarks:
 - sieve1 - sieve of erosthanes
 - matrix mul - from stanford benchmarks
 - blit - customer blit benchmark, character rendering; sum of 5 to 75 point character blits at 5-point intervals (code size ~ 1.2Kb)
 - ocr - customer OCR benchmark (code ~ 2Kb, data ~ 10Kb)

Other i960CA Performance Improvements

Instruction Latency Reductions	30-100%
Additional Bypass Paths	10-30%
Larger Instruction Cache	0-100%
Improved Bus Bandwidth	10-30%
Branch prediction	3-10%
Stack Frame Cache size increase	0-30%
On-chip SRAM	0-50%
Clock Increase (25Mhz - 33Mhz)	32%
Performance Improvement	75% - 300%

Realizing SuperScalar's Full Potential

- unrealized potential opportunities for superscalar dispatch
 - up to 50% of all cycles
 - ~ 50/50 instruction fetch/resource conflict
- decrease instruction fetch stalls by:
 - widening bus
 - additional buses
 - increasing size of i-cache
 - additional caches
- decrease resource conflict stalls by:
 - increase function of AGU
 - add register renaming to instruction scheduler
 - unroll loops and add software pipelining

Compiler Issues

- Compilers are improving ...

matrix	gcc 0.9 of 9/89	2689	
matrix	gcc 1.2 of 7/90	1873	43.5%

- But they still have room to improve

matrix	handcoded	1163	61.0%
--------	-----------	------	-------

— handcoded algorithm is unrolled and hand-scheduled

- Current compiler issues
 - artificial resource conflicts
 - short basic blocks
 - calling convention constraints
 - insensitivity to memory latency & caching
- Superscalar-specific improvements still in development

Summary

- Superscalar i960 CA achieves up to 50% of expected improvement of an order-3 superscalar machine
- Other 960 CA improvements increase speed up to 2.5x over first-generation RISC
- Other hardware improvements will increase performance 50% - 100%
- Compiler improvements can be expected to contribute 50% performance increase