HaL Computer Systems

Performance Evaluation of the Superscalar Speculative Execution

HaL SPARC64 Processor

by

Andrew Essen and Stephen Goldstein
ae@hal.com steven@hal.com

Aug. 95

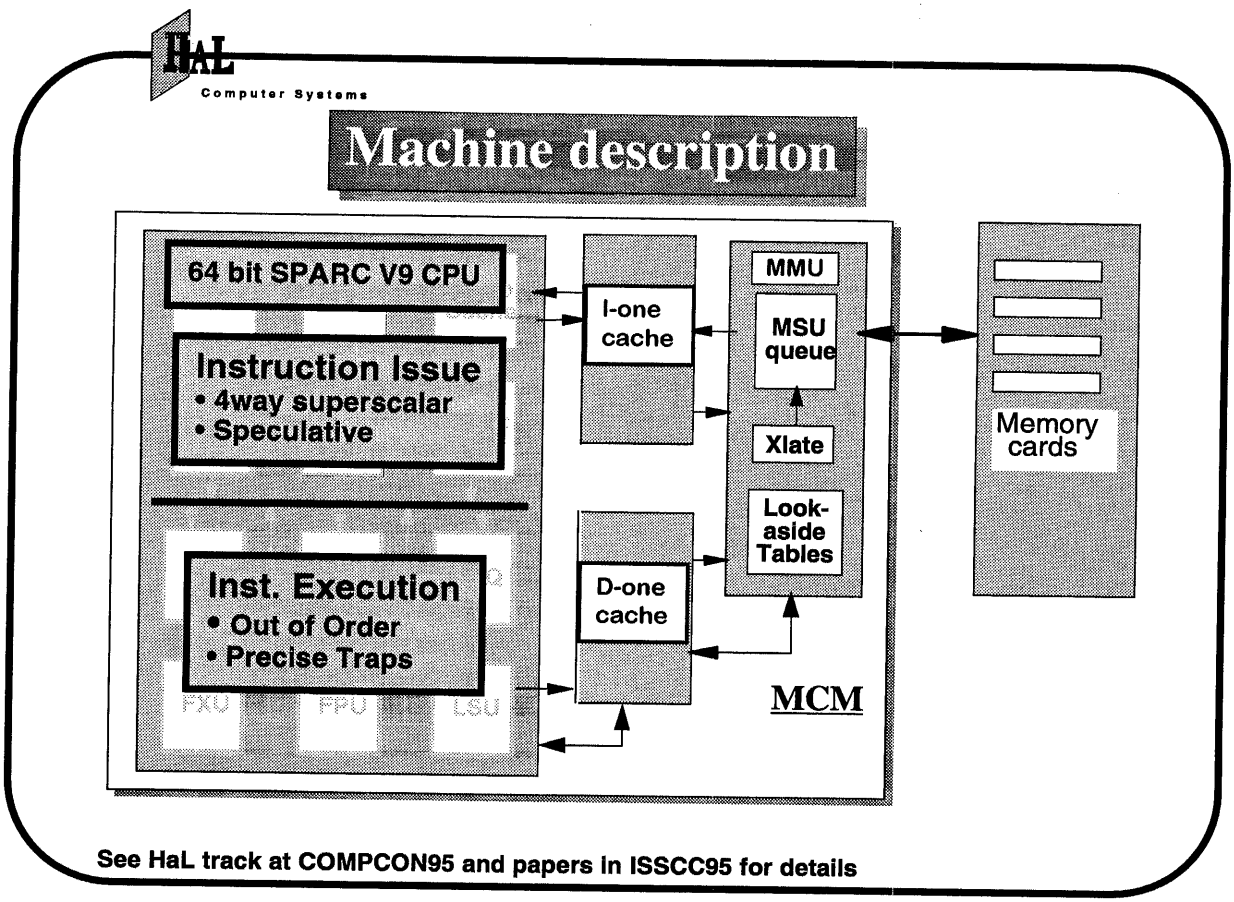
HaL Computer Systems

Outline of Talk

- ⇒ **Machine description**
- ⇒ **Performance Prediction Methods**
- Options Evaluated
- Validation of Predictions
- Measuring Superscalar Execution
- "What-if" Studies of Performance
- A Brief Survey of Superscalar Performance
- Conclusions and Perspective

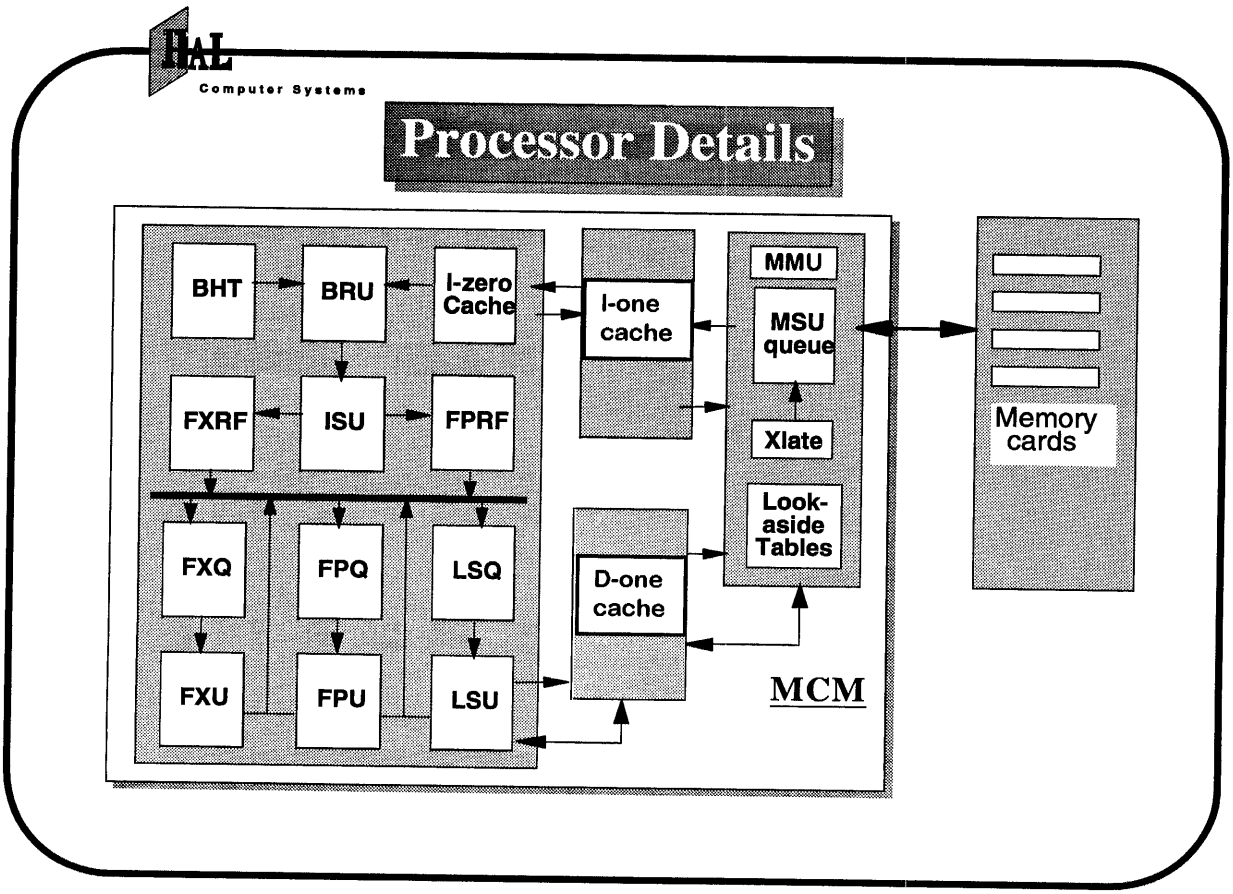
Aug. 95

3.1-03



Aug. 95

3.1-04



Aug. 95

HAL Computer Systems

Performance Estimates

Elapsed Time = CPI x Cycle time x Inst. count

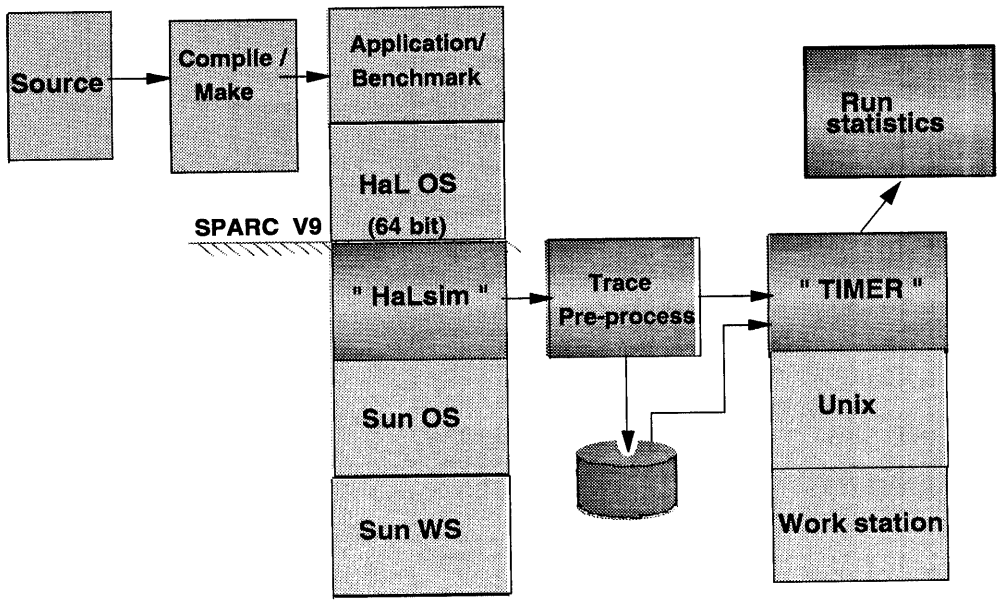
- **Cycle per Inst*:** **Predicted by the Timer software model**
  Input from HaLsim traces
  Statistical sampling techniques were used
- **Cycle time:** **Determined by technology and logic design**
- **Inst. count:** **Measured with applications run on HaLsim**

*Sometimes it's more convenient to use Instructions per Cycle: IPC = 1 / CPI

Aug. 95

HAL Computer Systems

Simulation Process



Aug. 95

HAL Computer Systems

Outline of Talk

- Machine description
- Performance Prediction Methods
- ⇒ **Options Evaluated**
- ⇒ **Validation of Predictions**
- Measuring Superscalar Execution
- "What-if" Studies of Performance
- A Brief Survey of Superscalar Performance
- Conclusions and Perspective

Aug. 95

HAL Computer

Timer is a Flexible Tool

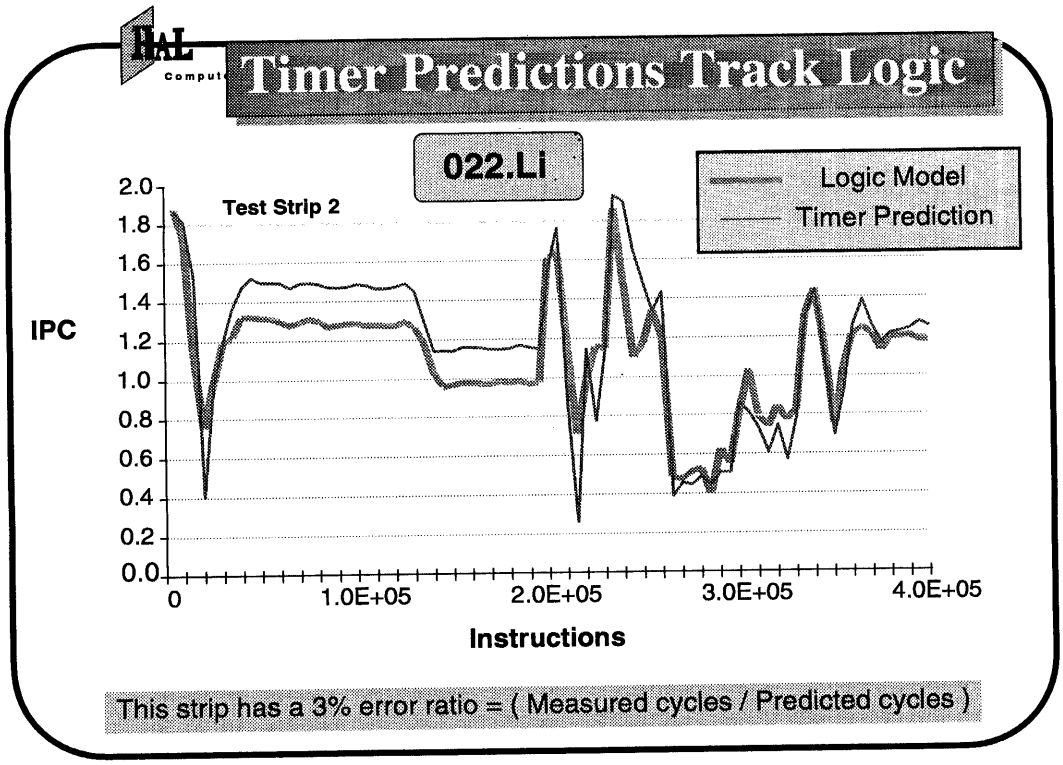
200+ Independent Options Handled as Switches in Timer

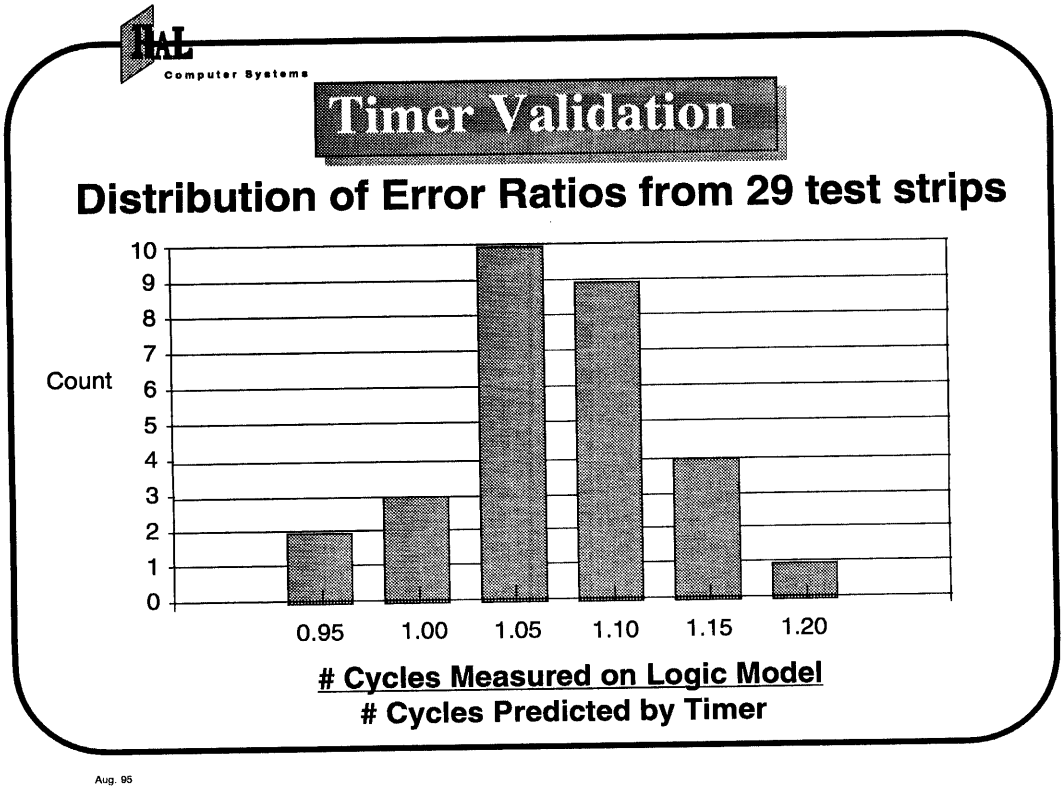
- → **Functional unit Latency, Number of, and Queue depth**
- → **Cache sizes, Organizations and Levels**
- → **Branch prediction, Fetch policies**
- → **Issue rules, Peak Issue rate, and Instruction window size**
- → **Translation and Memory latency and bandwidth (incl. TLB miss modeling)**
- → **Magic Options (e.g. perfect caches, perfect branch predict, . . .)**
- → **. . .**

Options Not Easily Handled in Timer

- → **Speculative side effects**
- → **I/O -- contention from DMA**
- → **Major changes in pipeline stages**

Aug. 95


Timer Predictions Track Logic
022.Li
Test Strip 2
Logic Model
Timer Prediction
IPC
2.0
1.8
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
0
1.0E+05
2.0E+05
3.0E+05
4.0E+05
Instructions
This strip has a 3% error ratio = (Measured cycles / Predicted cycles)
Aug. 95


Computer Systems
Timer Validation
Distribution of Error Ratios from 29 test strips
Count
10
9
8
7
6
5
4
3
2
1
0
0.95
1.00
1.05
1.10
1.15
1.20
Cycles Measured on Logic Model
Cycles Predicted by Timer
Aug. 95

HAL Computer Systems

Outline of Talk

- Machine description
- Performance Prediction Methods
- Options Evaluated
- Validation of Predictions

⇒ **Measuring Superscalar Execution**

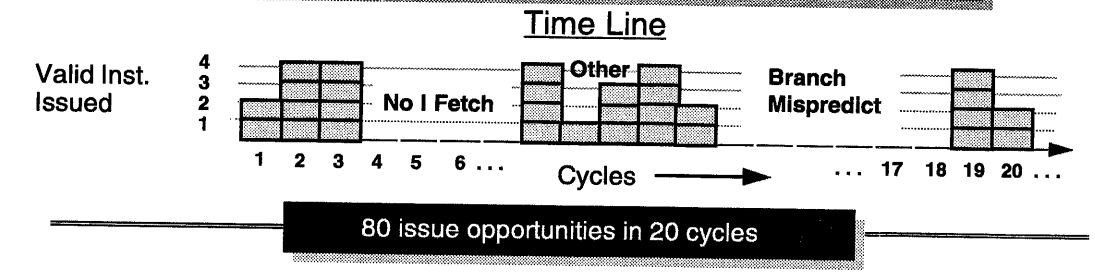
⇒ **"What-if" Studies of Performance**

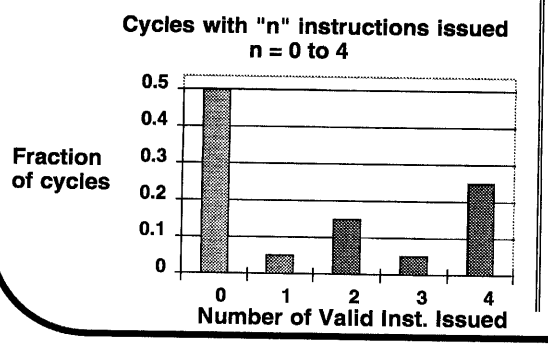
- A Brief Survey of Superscalar Performance
- Conclusions and Perspective

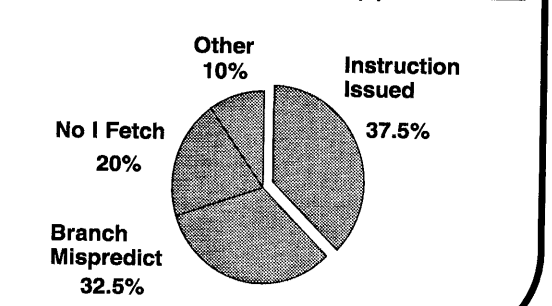
Aug. 95

HAL Computer

Time Line, Cycle, and Issue Three Views -- Same Data



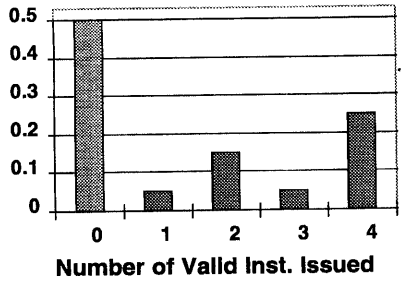




Aug. 95

HAL Computer Systems

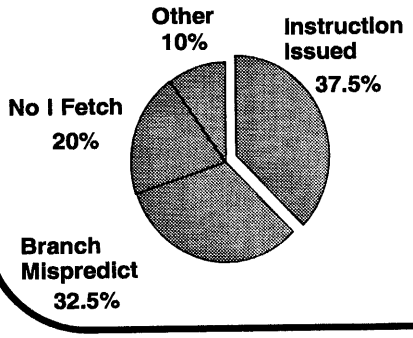
IPC Calculations



$$IPC = \sum_{n=0}^{4} P(n\text{-issue cycles}) \times n$$

For this example:

IPC =.5 (0) + .05(1) + .15(2) + .05(3) + .25(4)
IPC = 1.50



IPC = % inst. issued x opportunities/cycle

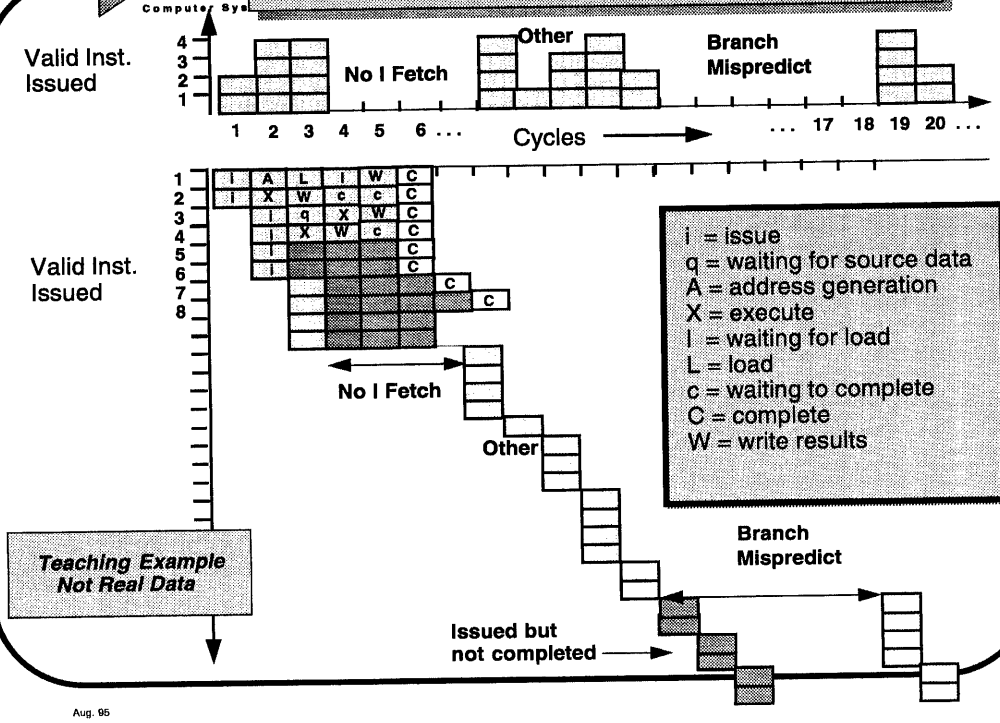
For this example:

IPC = 37.5% x 4
IPC = 1.50

Aug. 95

HAL Computer Systems

Timer Output Shows Details



Aug. 95

3.1-15

HAL Computer Systems

Timer Details from GCC

```
Starting at cycle: 119911
                                       |
                                       111111111122222222223333333333444444444455555555556
                                       123456789012345678901234567890123456789012345678901234567890
dd        %i1, %g1, %g1        1a    FXU    iqqqqqqqXWcC     # 28
add        %g1, %g3, %i1        1b    FXU     iqqqqqqqXWcC     # 29
ubcc      %i1, 0x34, %g0       1c    FXU     iqqqqqqqqXWcC     # 30
          bg          0x44     1d    ISU     iWccccccccccC     # 31
......................mispredict stall:       <<<<<<<<<<<<
dc0          add        %g0, %g2, %i2       1e    FXU    iXWcC    # 32
............................................i0 stall:    @@@
..10e00          or          %g0, 0x0, %i0       1f    FXU    iXWcC    # 33
..10e04        jmpl        [%i7+0x8], %g0        20    ISU    iXWcC    # 34
....10e08          restore     %i0, %g0, %o0       21    ISU    iXWcC    # 35
.....114e4          add         %g0, %o0, %i3      22    FXU    iXWcC    # 36
.....114e8          subcc       %i3, 0x0, %g0      23    FXU    iqXWcC    # 37
.....114ec                    be,a       0xdc      24    ISU    iWcccC    # 38
.....114f0        subcc      %i0, 0x102, %g0       25    FXU    iXWccC    # 39
..................................................i0 stall:    @@@@@@@@
...............115c8               bne        0x1510        26    ISU    iWcC    #40
...............115cc                              nop       27    ISU    iWcC    #41
................115d0              call       0x14b30        28    ISU    iXWcC    #42
.................115d4        ld          [%l2+0x0], %o0(34)  29   AGEN    iALlWc    #43
```

IPC = 0.50

Aug. 95

3.1-16

HAL Computer Systems

Timer Details from GCC

```
                        Starting at cycle: 121771                                  121800
                                                  |                                   |
                                                  77777777788888888889999999999000000
                                                  12345678901234567890123456789012345
       sth         %i5, [%i2+0x2](293)    22    AGEN    iAlILlWccC    # 2020
        add         %i2, 0x2, %i2         23    FXU     iXWccccccC    # 2021
    ld         [%fp+0x1b38], %g1(277)     24    AGEN     iALlWcccC    # 2022
          subcc       %i2, %g1, %g0       25    FXU      iqqqXWccC    # 2023
                  bcs,a      0x128        26    ISU      iWccccccC    # 2024
      ld           [%l4+0x0], %g2(189)    27    AGEN     iAlILlWccC    # 2025
          subcc       %g2, 0x0, %g0        28    FXU      iqqqqXWcC    # 2026
                  be          0x28         29    ISU      iWcccccccC    # 2027
 sethi      %hi(0xffff8000), %l2           2a    FXU      iXWccccccC    # 2028
      sethi       %hi(0xd2400), %g3         2b    FXU      iXWccccC    # 2029
          sll         %i5, 0x1, %l3         2c    FXU      iXWccccC    # 2030
       add         %g3, 0x1f0, %g3          2d    FXU      iqXWcccC    # 2031
        ldsh        [%l3+%g3], %i0(208)     2e    AGEN     iaaALlWccC    # 2032
            subcc       %i0, %l2, %g0        2f    FXU      iqqqqXWcC    # 2033
                  be,a        0x21c          30    ISU      iWcccccccC    # 2034
8       sethi      %hi(0xf4800), %g3          31    FXU      iXWccccC    # 2035
c       ld          [%g3+0x128], %l3(293)     32    AGEN     iaALlWccC    # 2036
0        subcc       %i3, 0x1ffe, %g0         33    FXU      iqqqqXWcC    # 2037
4                  bne         0x48           34    ISU      iWcccccccC    # 2038
f8          ld          [%l4+0x0], %g2(189)    35    AGEN     iAlLlWccC    # 2039
03c           subcc       %i3, 0x0, %g0        36    FXU      iqqXWccC    # 2040
3040                  bg,a         0x44        37    ISU      iWccccC    # 2041
3044          subcc      %i3, 0x128, %g0       38    FXU      iqqXWcC    # 2042
13084                  bgu,a        0x1c       39    ISU      iWcccC    # 2043
.1308c        sethi       %hi(0xd2c00), %g1    3a    FXU      iXWcC    # 2044
.13090          add         %g1, 0x1b0, %g1    3b    FXU      iqXWcC    # 2045
.13094                  ba          0xc        3c    ISU      iWcccC    # 2046
.13098           ldsb        [%i3+%g1], %l7(443)  3d   AGEN    iaaALlWccC    # 2047
..130a0             subcc      %g2, 0x0, %g0    3e    FXU      iXWccccccC    # 2048
..130a4                  be,a        0x3c       3f    ISU      iWcccccccC    # 2049
..130a8             addcc      %i0, %l7, %i0    0     FXU      iqqqqXWcC    # 2050
...130e0                  bl,a        0x11c     1     ISU      iWccccccC    # 2051
....130e8           subcc      %i0, 0x634, %g0  2     FXU      iqqqXWcC    # 2052
....130ec                  bg,a        0x110    3     ISU      iWccccccC    # 2053
```

IPC = 2.43

Aug. 95

HAL Computer Systems

SPEC92 Integer Benchmarks

6 benchmarks

• 008.espresso	Logic Design	Gens and optimizes programmable arrays
• 022.li	Interpreters	Lisp interpreter solves 9 queens problem
• 023.eqntott	Logic Design	Translates logic to Boolean truth tables
• 026.compress	Data Compress	Compresses using Lempel Ziv coding
• 072.sc	Spreadsheet	Calculates budgets etc.
• 085.gcc	Compiler	Translates C source into Sun assembly

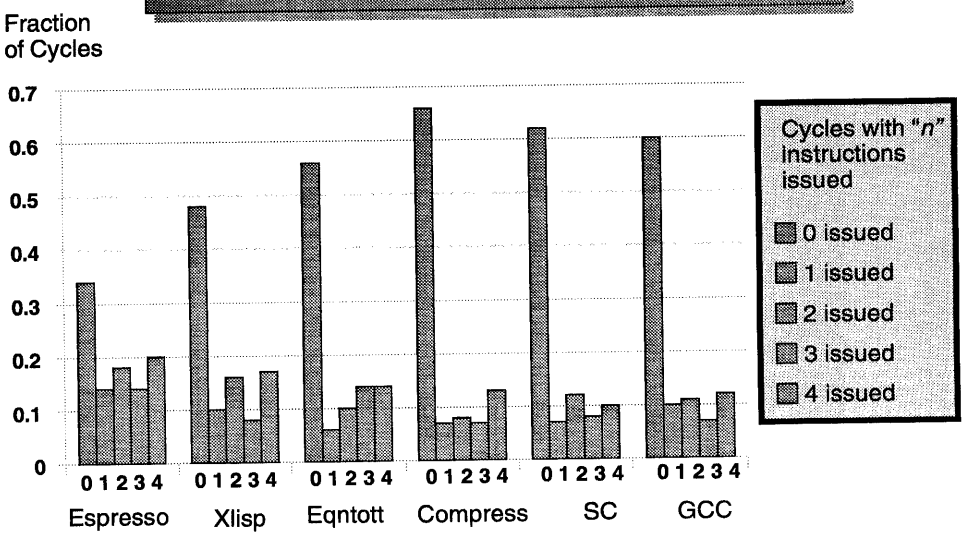
Typically each benchmark executes 10^9 to 10^{10} instructions.

SPEC™ of Standard Performance Evaluation Corporation

Aug. 95

HAL Computer Systems

HaL SPARC64 Design Point

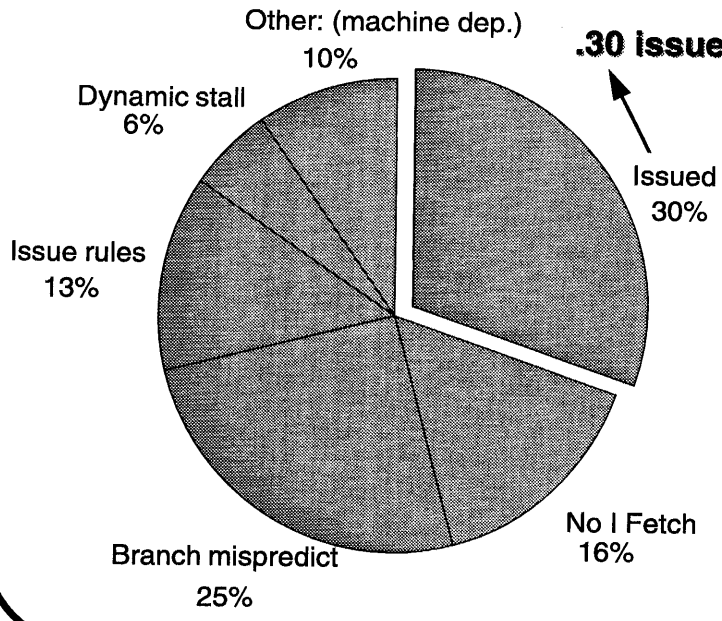


Benchmark results estimated with Timer

Aug. 95

HAL Computer Systems

HaL SPARC64 Design Point



.30 issued/slot x 4 slot/cyc = 1.2 IPC

Estimate for SPECint92
- Avg. of 6 benchmarks
- Each benchmark equal wt.

Aug. 95

HAL Computer Systems

"Back of the Envelope" Sanity Check

$$IPC^{-1} = CPI = .25 + 8 \times MR(Br) + 4 \times MR(I_0) + 28 \times MR(L_1)$$

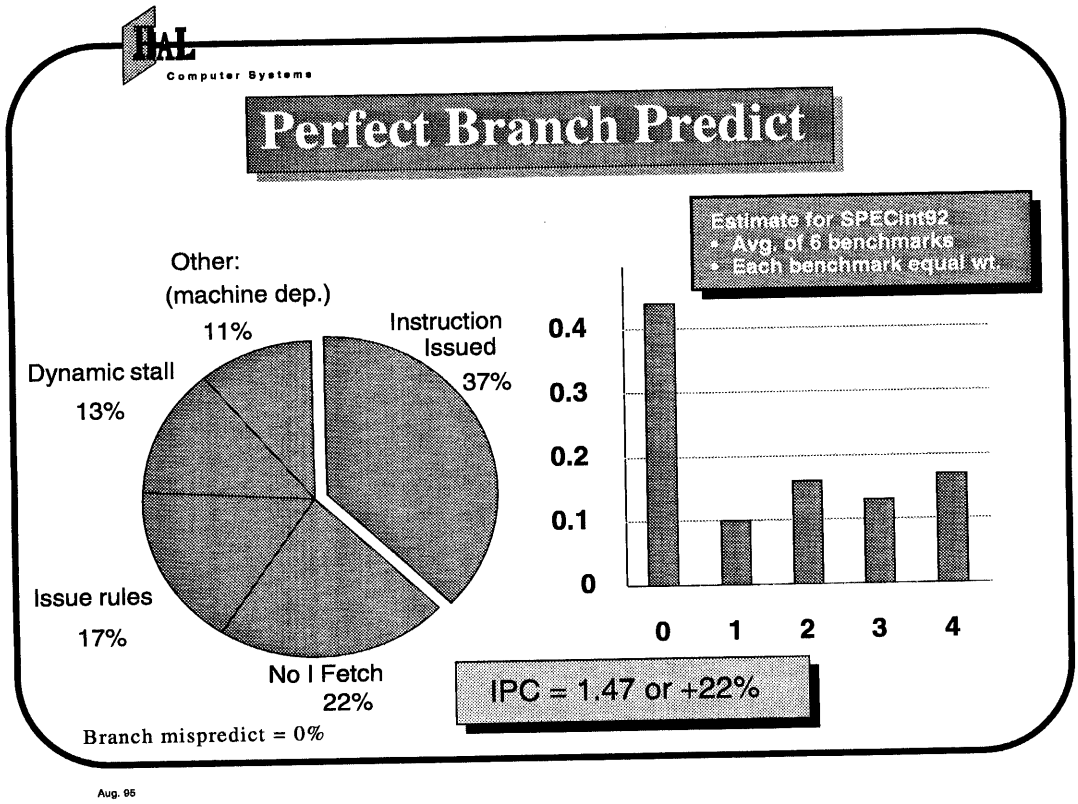
4 issue/cyc	8 cyc/miss .025 miss/inst	4 cyc/ I-zero miss .015 miss/inst	28 cyc to DRAM .0056 miss/inst

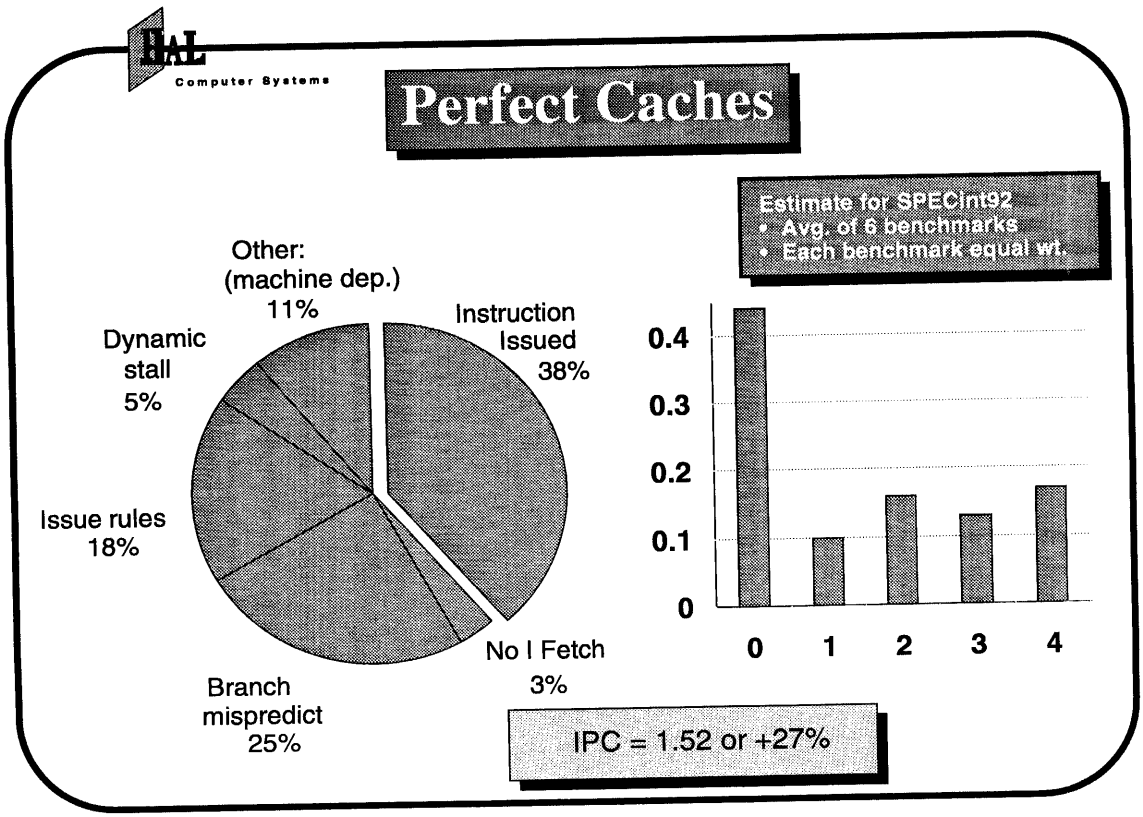
This gives a rough estimate of performance

CPI = .67 or IPC = 1.5

MR is Miss Rate expressed as Misses per Instruction Retired

Aug. 95

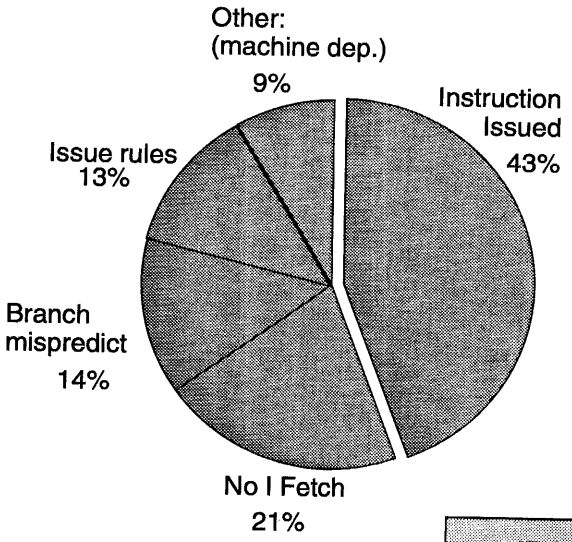

HAL
Computer Systems
Perfect Branch Predict
Estimate for SPECint92
• Avg. of 6 benchmarks
• Each benchmark equal wt.
Other:
(machine dep.)
11%
Instruction
Issued
37%
Dynamic stall
13%
Issue rules
17%
No I Fetch
22%
Branch mispredict = 0%
0.4
0.3
0.2
0.1
0
0 1 2 3 4
IPC = 1.47 or +22%
Aug. 95


HAL
Computer Systems
Perfect Caches
Estimate for SPECint92
• Avg. of 6 benchmarks
• Each benchmark equal wt.
Other:
(machine dep.)
11%
Instruction
Issued
38%
Dynamic
stall
5%
Issue rules
18%
No I Fetch
3%
Branch
mispredict
25%
0.4
0.3
0.2
0.1
0
0 1 2 3 4
IPC = 1.52 or +27%
Aug. 95

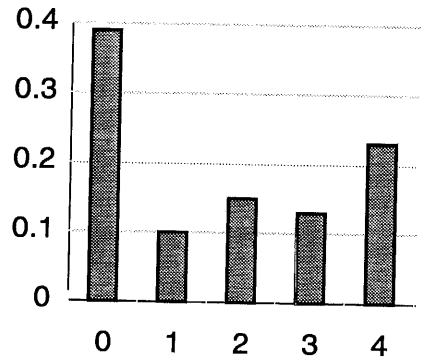
3.1-23

HAL Computer Systems

All execution in one cycle



⇒ Implies perfect D cache
⇒ Minimizes losses due to branch mispredict



Dynamic stall = 0%

IPC = 1.72 or +43%

Aug. 95

3.1-24

HAL Computer Systems

Outline of Talk

- Machine description
- Performance Prediction Methods
- Options Evaluated
- Validation of Predictions
- Measuring Superscalar Execution
- "What-if" Studies of Performance

⇒ **A Brief Survey of Superscalar Performance**

⇒ **Conclusions and Perspective**

Aug. 95

HAL Computer Systems

SPECint92 / MHz

⇒ **SPECint92 and SPECint92 / MHz are good metrics**

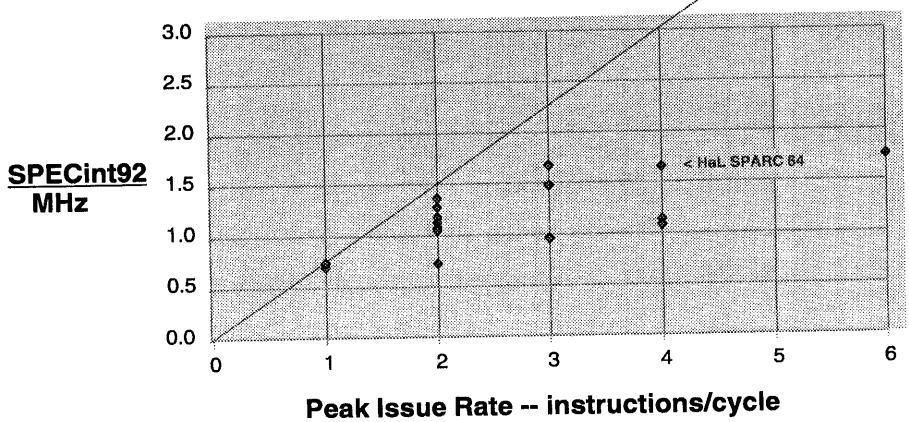
⇒ **Broader comparisons allowed**
- IPC's are seldom published
- SPECint results are publicly available

⇒ **Directly comparing IPC's has "the MIPS problem"**
- Different architectures make IPC comparisons meaningless across vendors
- A machine running the same benchmark with different binaries can give misleading IPC results

Aug. 95

HAL Computer Systems

Diminishing Returns from Multiple Issue



Diminishing Returns from Multiple Issue is a Universal Problem

Other vendor SPECint data from recent SPEC Newsletters

Aug. 95

3.1-27

HAL Computer Systems

Efficient Superscalar Operation is Difficult

Recent Performance Claims of 64 bit RISC Processors

Model	SPECint92@MHz	SPECint / MHz
UltraSPARC	240 @ 167	1.4
R10000	300+@ 200	1.5+
SPARC 64	256 @ 154	1.7
620	225 @ 133	1.7
PA8000	360 @ ~200	1.8

SPECint92 / MHz is a Figure-of-Merit for Micro-Architectures, Most Vendors Claim to be Doing About Equally Well.

Vendor data from recent Microprocessor Reports

Aug. 95

3.1-28

HAL Computer Systems

Conclusions

⇒ **It's hard to sustain high IPC's**
- **Simply increasing peak issue rates runs into Amdahl's law**
- **Parallel out-of-order execution is not a panacea**

⇒ **Impacts of proposed features must be evaluated**
- **Test cases should be realistic**
- **Modeling can identify true bottlenecks and opportunities**
- **Quick analyses plus detailed simulations make a powerful tool**

⇒ **Benchmark execution time is the correct metric**
- **Focusing solely on IPC's is dangerous**

Aug. 95

HAL Computer Systems

Perspective

⇒ **We've come a long way from 1 SPECint92 machines**

The SPEC92 "reference machine" is

- **5 MHz, single issue, CISC architecture**
- **circa 1978**

⇒ **From 0.2 to 1.67 SPECint92/MHz**

- **Over a factor of 8 improvement**

⇒ **5 MHz to 154 MHz**

- **Over a factor of 30 improvement**

These Improvements Multiply giving 250+ Times Faster Uniprocessors

Aug. 95

HAL Computer Systems

Acknowledgment

THE AUTHORS WISH TO THANK and ACKNOWLEDGE THE CONTRIBUTIONS OF:

- THE HAL ENGINEERING
- THE HAL TEST TEAMS
- THE HAL SOFTWARE OS AND TOOLS TEAMS
- THE HAL COMPILER DEVELOPMENT TEAM
- OTHER MEMBERS OF HAL ARCHITECTURE AND PERFORMANCE
  PARTICULARLY:
  VENKAT BUDUMA, MYKE SMITH, AND DR. WINFRIED WILCKE

Aug. 95