My file

IN CONFIDENCE

SCIENCE AND ENGINEERING RESEARCH COUNCIL
RUTHERFORD APPLETON LABORATORY

COMPUTING DIVISION

DISTRIBUTED INTERACTIVE COMPUTING NOTE 775

ICL PERQ v APOLLO DOMAIN - ARITHMETIC BENCHMARKS

issued by
I D Benest
4th January 1983

Copies to: K Robinson
C Prosser
J Collis
L Ford
R Witty
I D Benest
DIC Notes file
Apollo/General

Circulate to: F R A Hopgood
All Section

This note documents the results obtained from running the arithmetic benchmarks on the ICL PERQ and APOLLO DOMAIN Computers

1 INTRODUCTION

A series of benchmarks have been carried out to compare the computation speed of a number of computer systems sited at the Rutherford Appleton Laboratory. This note documents those results obtained from tests, run on the ICL PERQ and APOLLO DOMAIN machines, which highlight their relative arithmetic operating speeds.

The overall operational speed of a machine depends very much on the relative mix of tasks it has to perform and while one machine may provide impressive results for simple arithmetic tests, it may, for example, perform poorly on disk I/O, causing a very large numerical program making heavy demands on virtual memory processing, to produce poor results on the faster arithmetic machine. Thus the results obtained from these benchmarks should only be regarded as a rough guide to the arithmetic processing abilities of each machine.

2 COMPUTER DESCRIPTIONS

2.1 ICL Perq

The ICL PERQ used for these tests had the following characteristics:

1. 512kbytes random access memory
2. 24Mbyte Winchester hard disk
3. General Purpose Interface Bus
4. RS232C interface port
5. 768x1024 black and white portrait oriented screen
6. 1Mbyte floppy disk drive
7. Keyboard and tablet
8. PERQ Operating System (POS) D.70
9. PASCAL V6.0
10. FORTRAN V1.3
11. LINK V4.4
12. Microcoded floating point

2.2 Apollo Domain

The APOLLO DOMAIN used for these tests had the following characteristics:

1. Motorola 68000 8MHz CPU
2. 512kbytes random access memory
3. 33Mbyte Winchester hard disk
4. Intel multibus
5. Three RS232C interface ports
6. 800x1024 black and white portrait oriented screen
7. 1Mbyte floppy disk drive
8. Keyboard with touch pad
9. AEGIS operating System 4.0
10. PASCAL 3.00
11. FORTRAN 4.00
12. Software floating point

3 BENCHMARKS

3.1 Introduction

These benchmarks essentially test the basic arithmetic power of the machine under investigation. All tests were timed using a stop watch from the moment the carriage return was pressed invoking the relevant command, until the operating system prompt re-appeared. Although all compilations, loadings and program runs were timed, only the results of the program runs are included in this note. The tests were performed in the order: compile, load, run, compile, load, run, compile, etc., and completed in June 1982.

3.2 Arithmetic Tests

One million operations in single precision, double precision, integer, real, complex, add, subtract, multiply and divide written in FORTRAN and PASCAL were timed. The 21 tests were as follows:

dpia ... one million double precision integer additions
dpid ... one million double precision integer divisions
dpim ... one million double precision integer multiplications
dpis ... one million double precision integer subtractions

dpra ... one million double precision real additions
dprd ... one million double precision real divisions
dprm ... one million double precision real multiplications
dprs ... one million double precision real subtractions

spca ... one million single precision complex additions
spcd ... one million single precision complex divisions
spcm ... one million single precision complex multiplications
spcs ... one million single precision complex subtractions

spia ... one million single precision integer additions
spid ... one million single precision integer divisions
spim ... one million single precision integer multiplications
spis ... one million single precision integer subtractions

spra ... one million single precision real additions
sprd ... one million single precision real divisions
sprm ... one million single precision real multiplications
sprs ... one million single precision real subtractions

noop ... one million no-operations

The basic outline of the FORTRAN and PASCAL arithmetic benchmarks are given in Appendices 1 and 2. The equivalent variable declarations in all four compilers are given in Table 9 in Appendix 3. This Table indicates the necessary changes needed to transfer the tests from one machine to the other.

A lack of certain data types meant that the following tests could not be completed:

1. ICL PERQ FORTRAN - Single Precision Integer (4 tests)
2. ICL PERQ FORTRAN - Double Precision Real (4 tests)
3. ICL PERQ PASCAL - Double Precision Real (4 tests)
4. ICL PERQ PASCAL - Single Precision Complex (4 tests)
5. APOLLO DOMAIN PASCAL - Single Precision Complex (4 tests)

4 RESULTS

A no-operation was timed and the results of subtracting that time from those obtained from the arithmetic function tests produced the results given in Table 1. Thus Table 1 gives the time taken, in seconds, to compute one million operations with no additional overheads (except a single x and y initialisation).

During the investigations, it was noted that varying the actual numbers used by the arithmetic functions affected the speed with which the computation progressed; in some cases by quite a substantial amount. To indicate this variation, six x and y operands were chosen for double precision integer multiply (DPIM) and single precision real multiply (SPRM) in FORTRAN and PASCAL. The results are given in Table 2.

1 ICL PERQ v APOLLO DOMAIN Stopwatch Timing (One million operations)				
	ICL PERQ		APOLLO DOMAIN	
TEST	FORTRAN (secs)	PASCAL (secs)	FORTRAN (secs)	PASCAL (secs)
SPIA	NA	5.2	9.8	7.4
SPIS	NA	5.4	8.6	6.4
SPIM	NA	10.4	14.8	12.1
SPID	NA	16.6	27.4	26.6
SPRA	19.6	19.4	97.4	97.0
SPRS	20.2	20.6	109.4	105.4
SPRM	83.6	84.8	154.2	151.0
SPRD	256.4	256.8	187.2	182.8
DPIA	12.4	12.0	14.2	11.2
DPIS	12.4	12.4	14.2	10.2
DPIM	32.2	32.8	75.0	73.2
DPID	58.0	57.4	108.4	106.6
DPRA	NA	NA	124.6	121.4
DPRS	NA	NA	128.8	126.8
DPRM	NA	NA	294.0	290.5
DPRD	NA	NA	737.4	735.6
SPCA	156.2	NA	244.4	NA
SPCS	153.3	NA	246.6	NA
SPCM	526.6	NA	693.4	NA
SPCD	1220.2	NA	1278.6	NA

Table 1: Perq versus Domain - Arithmetic Tests
NA - Not Available; SP - Single Precision;
DP - Double Precision; I - Integer; R - Real;
C - Complex

2 ICL PERQ v APOLLO DOMAIN
Stopwatch Timing
(One million operations)

	OPERANDS		ICL PERQ		APOLLO DOMAIN	
	X	Y	FORTRAN (secs)	PASCAL (secs)	FORTRAN (secs)	PASCAL (secs)
DPIM	1	2	16.8	16.6	65.4	62.8
DPIM	123456	-290	32.2	32.8	75.0	73.2
DPIM	1	-2	21.0	19.6	73.8	71.2
DPIM	123456	-123456	45.8	45.6	72.2	68.6
DPIM	3	5	18.6	18.8	68.0	64.8
DPIM	3	-5	22.8	21.4	76.0	71.8
SPRM	1.7634	8.4923	93.2	93.0	159.4	152.8
SPRM	1.7634E4	8.4923E16	83.6	84.8	154.2	151.0
SPRM	-1.7634	8.4923	92.4	92.4	161.6	157.4
SPRM	-1.7634	8.4923E-16	92.4	91.8	165.4	160.0
SPRM	3.1	5.1	72.4	73.2	155.4	151.4
SPRM	512.9763	2732.568	81.2	81.4	159.2	152.9

Table 2: Computation Time Variation with Number Size
DPIM - Double Precision Integer Multiply.
SPRM - Single Precision Real Multiply.

5 ANALYSIS

By normalising the ICL PERQ results given in Table 1 and using the normalising factors on the Apollo Domain results, the figures in Table 3 were produced. This Table indicates how much longer the Apollo Domain takes to compute the same number of arithmetic operations. The average column under the Apollo Domain heading gives the average of the FORTRAN and PASCAL results and the overall average of 1.87 might be a useful reference figure to gauge the overall arithmetic speed of the Apollo Domain compared with that of the ICL PERQ.

3 ICL PERQ v APOLLO DOMAIN
Relative Comparison

TEST	ICL PERQ		APOLLO DOMAIN		
	FORTRAN	PASCAL	FORTRAN	PASCAL	AVERAGE
SPIA	NA	1.0	*	1.42	1.42
SPIS	NA	1.0	*	1.19	1.19
SPIM	NA	1.0	*	1.16	1.16
SPID	NA	1.0	*	1.60	1.60
SPRA	1.0	1.0	4.97	5.00	4.99
SPRS	1.0	1.0	5.42	5.12	5.27
SPRM	1.0	1.0	1.84	1.78	1.81
SPRD	1.0	1.0	0.73	0.71	0.72
DPIA	1.0	1.0	1.15	0.93	1.04
DPIS	1.0	1.0	1.15	0.82	0.99
DPIM	1.0	1.0	2.33	2.23	2.28
DPID	1.0	1.0	1.87	1.86	1.87
DPRA	NA	NA	*	*	*
DPRS	NA	NA	*	*	*
DPRM	NA	NA	*	*	*
DPRD	NA	NA	*	*	*
SPCA	1.0	NA	1.56	NA	1.56
SPCS	1.0	NA	1.61	NA	1.61
SPCM	1.0	NA	1.32	NA	1.32
SPCD	1.0	NA	1.05	NA	1.05

Table 3: Relative Comparison

Average: 1.87

Tables 4,5,6,7 and 8 were produced from the average results obtained from the FORTRAN and PASCAL tests. Table 4 shows the increase in compute time between multiplication and division. For an ICL PERQ it takes 2.2 times as long to do a division compared with a multiplication. The comparable figure on the Apollo Domain is 1.8.

TEST	ICL PERQ POS D.70	APOLLO DOMAIN AEGIS SR 4.0
SPID/SPIM	1.60	2.00
SPRD/SPRM	3.05	1.21
DPID/DPIM	1.78	1.45
DPRD/DPRM	NA	2.52
SPCD/SPCM	2.30	1.84
Average	2.20	1.80

Table 4: Speed Comparison between multiplication and division

Table 5 provides a comparison between single precision and double precision integers. On the ICL PERQ, integer computations in double precision are 2.8 times slower than single precision. The comparable figure for the Apollo Domain is 3.2.

TEST	ICL PERQ POS D.70	APOLLO DOMAIN AEGIS SR 4.0
DPIA/SPIA	2.35	1.48
DPIS/SPIS	2.30	1.63
DPIM/SPIM	3.13	5.49
DPID/SPID	3.48	3.98
Average	2.82	3.15

Table 5: Speed Comparison between double precision integer and single precision integer

Table 6 provides a comparison between single precision real and double precision real. On the Apollo Domain, double precision real takes about twice as long as single precision.

TEST	ICL PERQ POS D.70	APOLLO DOMAIN AEGIS SR 4.0
DPRA/SPRA	NA	1.27
DPRS/SPRS	NA	1.19
DPRM/SPRM	NA	1.92
DPRD/SPRD	NA	3.98
Average	NA	2.09

Table 6: Speed Comparison between double precision real and single precision real

Table 7 provides a comparison between double precision integer (32 bits) and single precision real (32 bits) On the ICL PERQ, single precision real takes about two and a half times as long as double precision integer. The comparable figure on the Apollo Domain is approximately five times as long.

TEST	ICL PERQ POS D.70	APOLLO DOMAIN AEGIS SR 4.0
SPRA/DPIA	1.60	7.65
SPRS/DPIS	1.65	8.80
SPRM/DPIM	2.59	2.06
SPRD/DPID	4.45	1.72
Average	2.57	5.06

Table 7: Speed Comparison between single precision real and double precision integer

Table 8 provides a comparison between single precision real (32 bits) and single precision complex (64 bits). On the ICL PERQ, single precision complex takes six and a half times as long as single precision real. The comparable figure on the Apollo Domain is four times as long.

TEST	ICL PERQ POS D.70	APOLLO DOMAIN AEGIS SR 4.0
SPCA/SPRA	8.01	2.51
SPCS/SPRS	7.51	2.30
SPCM/SPRM	6.25	4.54
SPCD/SPRD	4.76	6.91
Average	6.63	4.07

Table 8: Speed Comparison between single precision complex and single precision real

6 CONCLUSIONS

From these results the following conclusions may be drawn:

1. There is a consistent difference between FORTRAN and PASCAL implementations of each arithmetic function on the Domain. The evidence suggests that there is an overhead in the procedure call from FORTRAN which does not occur in PASCAL. No such difference exists on the ICL PERQ.

2. Single precision real divide on the Apollo Domain is about 1.9 times faster than the equivalent on the PERQ. Single precision add and subtract on the Apollo Domain are about five times slower than on the PERQ. Double precision integer multiply on the Apollo Domain is over twice as slow as on the PERQ. By taking simple averages the Apollo Domain takes 1.9 times as long to compute as the ICL PERQ although the variation ranges from 0.71 to 5.42. If the exceptional cases are removed, then the Apollo Domain takes 1.3 times longer than the ICL PERQ to do the same arithmetic computation.

3. The ICL PERQ exhibited a variation in computation time, but it is unclear just what were the causes of the variation. It seemed to be dependent on the number of characters in the operands and not dependent on the actual size of the number. For double precision integer computations, negative numbers also had an adverse affect on the computation time. For single precision real computations, negative numbers seemed to have no adverse affect. On the Apollo Domain, a variation did exist but it was less significant than on the PERQ. The main cause of the variation seemed to be due to the presence of negative numbers rather than the number of characters in the operands. As insufficient results were recorded, a prediction as to the sort of numbers which may be operated on more quickly, is difficult to provide. It can only be concluded that the variation exists and that it is quite substantial on the ICL PERQ. This means that the relative comparisons in Table 3 reflect only the numbers used in the test and that if different numbers had been used the comparisons would also have been different. For example, the double precision integer relative comparison varies from 1.5 to 3.8 and the single precision real relative comparison varies from 1.7 to 2.1.

4. The divide operation on both machines should be avoided.

7 SUMMARY

The foregoing has tabulated:

1. the raw arithmetic speed of both machines.

2. the variation in computation time with operand size.

3. the relative speed of the Apollo Domain compared with the ICL PERQ.

4. the 'price' incurred of using division instead of multiplication.

5. the 'price' incurred of using double, instead of single, precision.

6. the 'price' incurred of using single precision real instead of double precision real.

7. the 'price' incurred of using complex, instead of real, arithmetic.

APPENDIX 1 - ARITHMETIC PROGRAM LAYOUT (FORTRAN)

```
      program name
      integer i,j,count
      integer outmax,loopmx

        declare x,y,z

      outmax=100
      loopmx=100

        assign x and y thus:
                            x = 17          single precision integer
                            y = 84             "         "        "

                            x = 1.7634e4    single precision real
                            y = 8.4923e16      "         "      "

                            x = 3.475d12    double precision real
                            y = 6.0291d-7      "         "      "

                            x = (138.672,7.5238e2) complex real
                            y = (-835.2,1.5739)        "     "

                            x = 123456      double precision integer
                            y = -290           "         "        "

      do 500 count=1,outmax
         do 400 i=1,loopmx
             do 300 j=1,loopmx

               arithmetic assignment or no-operation

  300        continue
  400    continue
  500 continue
      stop
      end
```

APPENDIX 2 - ARITHMETIC PROGRAM LAYOUT (PASCAL)

```
program name;
const   loopmx=100;
        outmax=100;
var     i,j,count : integer;

        x,y,z type declaration

begin
          assign x and y thus:
                            x:=17;         integer - 16 bits
                            y:=84;            "         "

                            x:=1.7634e4;   real - 32 bits
                            y:=8.4923e16;   "         "

                            x:=123456;     integer32 - 32 bits
                            y:=-290;          "          "

                            x:=3.475e12;   double - 64 bits
                            y:=6.0291e-7;    "         "

        for count:=1 to outmax do
        begin
          for i:=1 to loopmx do
           for j:=1 to loopmx do
           begin

             arithmetic declaration or no-operation

           end
        end;
end.
```

APPENDIX 3 VARIABLE DECLARATION EQUIVALENTS

ICL PERQ		APOLLO DOMAIN		SIZE
FORTRAN	PASCAL	FORTRAN	PASCAL	BITS
****	INTEGER	INTEGER*2	INTEGER INTEGER16	16 16
INTEGER	LONG	INTEGER INTEGER*4	INTEGER32	32 32
REAL	REAL	REAL	REAL SINGLE	32 32
DOUBLE PRECISION	****	DOUBLE PRECISION	DOUBLE	64
COMPLEX	****	COMPLEX	****	64

Table 9: Variable Declaration Equivalents