

DECsystem-10

FORTRAN-10
COMPILER
OPTIMIZATION
BENCHMARK

15 January 1973

 *
 * FORTRAN COMPILER OPTIMIZATION BENCHMARK *
 *
 *
 *
 *
 *

THIS REPORT ATTEMPTS TO DEMONSTRATE THE LEVEL OF OPTIMIZATION PERFORMED BY THE FORTRAN COMPILERS OF THREE COMPUTER MANUFACTURERS: DEC, XDS, AND IBM. THE COMPILERS USED DURING THE TEST WERE THE NEW DECSYSTEM-10 FORTRAN, THE XDS FORTRAN-IV COMPILER, AND THE IBM FORTRAN-H WITH FULL OPTIMIZATION.

THIS REPORT CONTAINS A SUMMARY OF THE EXECUTION TIMES AND THE AMOUNT OF CODE PRODUCED BY EACH COMPILER, AND AN ANALYSIS OF THE OBJECT CODE CREATED BY EACH COMPILER.

	DECSYSTEM-10 1070	DECSYSTEM-10 1050	XDS SIGMA-7	XDS SIGMA-9	IBM 360/65
EXECUTION TIME	25 SEC	41 SEC	85 SEC	63 SEC	40 SEC
# OF INSTRUCTIONS GENERATED	66	66	71	71	81
# OF INSTRUCTIONS IN INNER LOOP	19	19	47	47	33

```

IMPLICIT INTEGER (A-Z)
DIMENSION A(100,100),B(100,100)
DO 100 IJK=1,100
SUM1=0
SUM2=0
DO 100 J=1,100
DO 100 I=1,100
K1=I+J
IF (K1.LT.500.OR.K1.GT.1500) K1=0
A(I,J)=K1
K2=I+J
IF (K2.EQ.100.OR.K2.EQ.200.OR.K2.EQ.300)K2=K2+1
B(I,J)=K2
SUM1=SUM1+K1
SUM2=SUM2+K2
CONTINUE
100 C
WRITE (6,10)SUM1,SUM2
FORMAT(7H SUM1= ,I9,10H SUM2= ,I9)
END

```

```

IMPLICIT INTEGER (A-Z)
DIMENSION A(100,100),B(100,100)
DO 100 IJK=1,100

```

```

MOVE 2,(777634000001)

```

```

5M:  MOVEM 2,IJK
6M:  HRREM 2,IJK

```

```

LI,4  1 IJK
STW,4 -101
AI,4

```

```

100000: LA 5,1
SI 5,IJK

```

10:

```

100001:

```

```

SUM1=0

```

```

SETZB 2,SUM1

```

```

LI,9  0 SUM1
STW,9

```

```

LA 0,0
SI 0,SUM1

```

```

;THE COMPILER MAKES USE OF SPECIAL
; PURPOSE INSTRUCTIONS TO SHORTEN CODE

```

```

;XDS NEEDS A LOAD AND A STORE.

```

```

;IBM ALSO NEEDS A LOAD AND STORE.

```

```

SUM2=0

```

```

SETZB 2,SUM2

```

```

LI,9  0 SUM2
STW,9

```

```

SI 0,SUM2

```

```

;XDS DOES NOT REMEMBER THAT
; AC9 ALREADY CONTAINS A
; ZERO.
;THERE IS NO CROSS STATEMENT
; OPTIMIZATION.

```

```

;IBM REMEMBERED THAT AC0 WAS 0
; FROM THE PREVIOUS STATEMENT.

```

DO 100 J=1,100

MOVEI 2, D100
MOVEM 2, R0001
MOVE 2, [777634000001]

LI,2 1
STW,2 J
AI,2 -101

0,400
0,006
4,5

;.R0001 CONTAINS J*100 FOR USE IN
; SUBSCRIPT CALCULATIONS
; THIS IS THE INITIALIZATION OF THE
; REDUCTION IN STRENGTH OPTIMIZATION
; OF J*100

; LIKEWISE, :006 CONTAINS J*400
; FOR USE IN SUBSCRIPT CALCULATIONS.

7M: MOVEM 2, S0001
HRREM 2, J

1000021

2G:

DO 100 I=1,100

8M: MOVE 2, J
MOVEM 2, R0000
MOVE 2, R0001
MOVEM 2, 00000
MOVE 15, K2
MOVE 14, R0000
MOVE 13, K1
MOVE 12, J
MOVE 11, SUM2
MOVE 10, SUM1
MOVE 7, 00001
MOVE 2, [777634000001]
MOVEM 2, S0002
HRREM 2, I

LI,6 1
STW,6 I
AI,6 -101

0,=4
0,006
0,005
6,5
7,005
8,4
9,SUM1
10,SUM2

IXDS DOES NOT MOVE ANYTHING
; OUTSIDE THE INNER DO LOOP.

IBM ALSO ASSIGNS AC'S AS TEMPORARY
; LOCATIONS FOR I, J, K2, SUM1,
; AND SUM2. K1 IS NOT IN AN AC DUE
; TO A LACK OF FREE AC'S.

9M: MOVEI 7, 0(2)
ADD 7, 00000

1000031

3G:

10C: CALCULATE THE SUBSCRIPT (I, J)
; ONLY ONCE. NOTICE THAT A
; MULTIPLICATION IS NOT USED.

```

*****
K1=I+J
-----
MOVE 13,14
JNO MULTIPLY IS NEEDED BECAUSE
; OF THE REDUCTION IN STRENGTH
; OPTIMIZATION OF I+J WHICH IS
; KEPT IN AC14.
*****
LENGTHY MULTIPLY INSTRUCTION
; IS USED.
*****
IF (K1,LT,500,OR,K1,GT,1500) K1=0
-----
CALL 13,D500
CALL 13,D1500
MOVEI 13,0
11M:
; DEC DOES VERY CLEVER NESTING OF
; COMPARE STATEMENTS, ESPECIALLY
; IF THE CONDITIONALLY EXECUTED
; EXPRESSION IS ONLY ONE INSTRUCTION
; AS IT IS HERE AND WOULD BE IN
; THE CASE OF A "GO TO" STATEMENT.
*****
A(I,J)=K1
-----
10M: MOVEM 13,A-10(17)
; AC7 CONTAINS THE SUBSCRIPT
; CALCULATION OF (I,J)
*****
; XDS DOES A MULTIPLY AND AN
; ADD FOR THIS CALCULATION.
*****
; IBM MAKES GOOD USE OF DOUBLE
; INDEXING.
*****
K2=I+J
-----
MOVEI 15,0(2)
ADD 15,12
; SINCE K2, I, AND J ARE IN THE
; AC'S NO MEMORY REFERENCES
; ARE MADE.
*****
; XDS DOES 3 MEMORY REFERENCES,
*****
; IBM DOES NO MEMORY REFERENCES.
*****

```

```

LR 1,4
MR 0,6
SI 0,K1

```

```

C 1,500
BC 4,100004
L 0,K1
C 0,=1500
BC 13,100005
LA 0,0
SI 0,K1

```

```

LI,9 500
CW,9 K1
BC 4G
LI,9 1500
CW,9 K1
BGE 5G
LI,9 0
STW,9 K1

```

```

100005: SI 8,001
LR 11,7
L 0,K1
L 15,ABASE
SI 0,(7,15)

```

```

5G: LW,9 K1
LI,1 100
MW,1 J
AW,1 I
STW,9 A-10(1,1)

```

```

LR 3,4
AR 3,6

```

IF (K2.EQ.100.OR.K2.EQ.200.OR.K2.EQ.300) K2=K2+1

GAIN 15,D100
JRST 0,13M
CAIE 15,D200
GAIN 15,D300
ADDI 15,1

JAGAIN NOTE THE CLEVER NESTING
; OF COMPARE INSTRUCTIONS.
; NOTICE THAT NOT A SINGLE
; MEMORY REFERENCE IS DONE.

6G

LI,9 100
CM,9 K2
BE 6G
LI,9 200
CM,9 K2
BE 6G
LI,9 300
CM,9 K2
BNE 7G
MTH,1 K2

C 3,2100
BC 9,100006
C 3,2200
BC 9,100006
C 3,2300
BC 6,100007
AR 3,5

B(I,J)K2

12M: MOVEM 15,B-101(7)

JAC7 CONTAINS THE COMMON
; SUBSCRIPT CALCULATION USED
; HERE AND ABOVE.

LW,9 K2
LI,1 100
MW,1 J
AW,1 I
STW,9 B-101,1

1000071 L
SI 15,BBASE
3,(11,15)

JXDS DOES THE MULTIPLY AND THE
; ADD FOR THE SECOND TIME.

SUM1=SUM1*K1

ADD 10,13

NO MEMORY REFERENCES NEEDED.

LW,9 K1
AMM,9 SUM1

JXDS DOES AN ADD TO MEMORY
; TO SAVE AN EXTRA STORE
; INSTRUCTION.

SUM2=SUM2+K2

ADD 11,15

JAGAIN NO MEMORY REFERENCES.

LW,9 K2
AMM,9 SUM2

AR 10,3

100 CONTINUE

100P1 ADD 14,12
AOBJN 2,9M

100S LW,4 IJK
AI,4 J -101
LW,2 J -101
AI,2 I -101
LW,6 I -101
AI,6 I
MTW,1 I
BIR,6 3G

100: A 8,=4
AR 7,=4
C 6,5
BC 6,=100
13,100003

UPDATE I*J IN AC14 BY
; ADDING J (AC12) TO THE
; RUNNING TOTAL IN AC14.
; NOTE: THE AOBJN INSTRUCTION
; DOES THE INCREMENTING OF
; I, THE CHECKING FOR THE
; END CONDITION, AND THE
; JUMP BACK ALL IN ONE
; INSTRUCTION.

IXDS RELOADS ALL OF THE DO LOOP
; COUNTERS IN AC'S 4, 2, AND 6,
; EVEN THOUGH NONE OF THOSE AC'S
; WERE OVERWRITTEN.

2M: MOVEM 7,00001
MOVEM 10,SUM1
MOVEM 11,SUM2
MOVEM 12,J
MOVEM 13,K1
MOVEM 14,R0000
MOVEM 15,K2
MOVEI 2,0100
ADDM 2,R0001
MOVE 2,S0001
AOBJN 2,7M

1000001 AR 4,5
L 0,006
A 0,=400
SI 0,006
C 4,=100
BC 13,100002

SI 9,SUM1
SI 10,SUM2

THE TEMPORARY VARIABLES ARE
; PUT BACK INTO CORE
; OUTSIDE THE INNER DO LOOP.

MOVE 2,S0000
AOBJN 2,5M

MTW,1 IJK
BIR,4 1G

1000001 L AR 0,IJK
SI 0,5
C 0,IJK
BC 0,=100
13,100001

IBM ALSO STORES THE TEMPORARY
; VARIABLES BACK INTO CORE
; OUTSIDE THE INNER DO LOOP.

