Burroughs

# B 6700

# TIMINGS

## FOR SELECTED LANGUAGE CONSTRUCTS

(RELATIVE TO MARK II.6 RELEASE)

$4.00

Printed in U.S. America 2-75 5000854

# Section 1
# CONSTRUCT TIMINGS

Construct timings do not, by themselves, convey a great deal of knowledge about performance. They represent only one set of characteristics out of many that must be considered. Nevertheless, they can be very useful.

They are more informative, for example, than the more primitive indicators such as clock rate or memory speed. and they are reasonably unambiguous. The "add time" of a computer may mean different things to different people. However, "A=B+C" denotes an amount of work done, whether on a particular machine which involves a polish string or a three-address fixed-length instruction.

Construct timings should be useful to the programmer who wants to understand how to write more efficient code. Knowing the relative speeds involved when choices present themselves is of obvious benefit.

The data in this document are actual measured timings. They were obtained by means of programs especially designed for this purpose, taking into account the following:

1. Timing Method

   FORTRAN:

   The TIME (11) function was used to read the time of day register. Since this function in FORTRAN calls an MCP intrinsic, one must consider the time involved to do this (about 80 $\mu$s on a B 6700 5.0/10.0 processor, 1.2 $\mu$s memory). The construct was repeated N3 times in line within a DO loop of N2 iterations with timer readings before and after the DO loop. The same loop was also timed "empty", i.e., with no constructs within it. This loop timing also includes the overhead for reading the timer. The difference in times (i.e., the time for the loop with constructs minus the time for the "empty" loop) divided by N3*N2 represents the time for the construct. N3 is chosen so that the time for N3 constructs is much larger than the

1. FORTRAN (Continued)

empty loop time. N2 is chosen to make the total DO loop time large in relation to the time to read the timer. This maximizes the accuracy of the timings.

ALGOL:

The timing in ALGOL was done in much the same way as in FORTRAN. The construct was repeated N3 times in line between two TIME (11) statements. The relatively small (about 6.4 μs on a B 6700 5.0/10.0 processor) time required to actually read and store the time of day register was later subtracted out. N3 was 100 in all cases. N2 does not apply.

COBOL:

The method is essentially the same as described for FORTRAN, using PERFORM N2 TIMES of N3 constructs. N2 and N3 were usually 100 and 50, respectively.

2. Perturbations

Should an interrupt occur at any time between timer readings, the timing will, of course, be distorted. Therefore, each timing loop was repeated to obtain a number, (N1), of observations. The few times which did not agree with the majority of observations were caused by such perturbations from interrupts and were discarded. N1 was usually 100 in the FORTRAN and ALGOL timings and 25 in the COBOL timings.

3. Code Alignment

For any construct, the generated code may occupy any one of six positions relative to word boundaries -- i.e., the string may start in syllable 0, 1, 2, etc. Some variations in timing will result from this, but they are of little consequence. For uniformity, all timings were done with code beginning in syllable zero.

4. Operands

Many of the constructs timed have a dependency on the actual values of the operands involved. For this reason, a variety of operands were used in the FORTRAN timings for these constructs to measure the extent of that dependency. For those cases, a "range" of times observed is quoted. This range is over those operands used and may or may not include the possible minimum and maximum.

5. Repeatability

All of the programs were run more than once - on different days, different processors, some on different systems - to establish the repeatability of the results.

What is sometimes seen is that there are two repeatable timings. This occurs because the timing is affected slightly by whether the code and operands are in the same memory module. The amount of the difference varies according to the construct involved, but is relatively minor (usually 10% or less).

But there are, in effect, two correct answers for each construct. Because of this, where different language constructs generate the same code, the timing reported may be the same or may vary slightly.

6. Hardware

One 5.0/10.0 processor was used with 1.2 μs memory (B6004-1) in one case and 1.6 μs memory (B6005-1) in the other case.

7. Software

The software release level is noted for each set of timings. For some constructs, the timing could conceivably change from one release to the next, if different code is generated.

A sample FORTRAN timing program follows.

```
      DIMENSION OBSV(1000)
      N1 = 100
      N2 = 14
      N3 = 100
      DO 30 I = 1, N1
      TBEG = TIME(11)
      DO 10 J = 1, N2
      Z = X + Y
C
C   THE CONSTRUCT ABOVE IS REPEATED SO THAT IS APPEARS
C   WITHIN THIS DO-LOOP A TOTAL OF N3 (N3 = 100) TIMES.
C
C   FOR EACH SET OF OPERAND VALUES THE CONSTRUCT
C   IS EXECUTED N1*N2*N3 TIMES.
C
10    CONTINUE
      TEND = TIME(11)
      OBSV(I) = TEND - TBEG
      TBEG = TIME(11)
      DO 20 J = 1, N2
20    CONTINUE
      TEND = TIME(11)
      OBSV(I) = 2.4 * (OBSV(I) - TEND + TBEG) / (N2*N3)
30    CONTINUE
C
C   SORT AND OUTPUT THE OBSERVATIONS (OBSV ARRAY)
C
      STOP
      END
```

# Section 2
# ALGOL CONSTRUCT TIMINGS

The following timings of certain ALGOL constructs were obtained using the MCP and ALGOL compiler from the Mark II.6 systems software release. The timings of each construct for each memory speed (1.2 us and 1.6 us) are displayed on the following pages. An asterisk appearing between the two columns of timings refers to a note on the last page of this section.

The variable names used and their data type definitions are:

```
REAL      XREAL,YREAL,ZREAL;
INTEGER   XINTGR,YINTGR,ZINTGR, I,J,K;
DOUBLE    XDBL,YDBL,ZDBL;
ARRAY     ARAY1[0:9], BRAY1[0:9],
          ARAY2[0:9,0:9], BRAY2[0:9,0:9],
          ARAY3[0:9,0:9,0:9], BRAY3[0:9,0:9,0:9];
```

A range in timing for some of the constructs is due to the different operand values used in the execution of the construct. The pairs of operand values are:

```
(XREAL,YREAL) = (1@0,1@0) and (6517, - 1740)
(XINTGR,YINTGR) = (1,1) and (6517, - 1740)
(XDBL,YDBL) = (1@@0, 1@@0) and (4632@@31, 8623@@27)
(ARAY...,BRAY...) = (1@0,1@0) and (4631527@28, 8623355@24)
I, J and K = 1
```

The constructs containing ABS and DABS had an operand value of -12345.

ALGOL Construct Timings

Proc Speed : 5.00/10.00
MCP : Mark II.6
ALGOL : Mark II.6

## SINGLE PRECISION REAL CONSTRUCTS

| ALGOL Construct | Timings in Microseconds: 1.2 μs Memory | | Timings in Microseconds: 1.6 μs Memory |
|---|---|---|---|
| ZREAL := XREAL + YREAL; | 5.6 - 7.0 | | 10.3 - 11.7 |
| ZREAL := XREAL - YREAL; | 6.0 - 7.2 | | 10.7 - 11.9 |
| ZREAL := XREAL * YREAL; | 7.0 - 8.4 | | 11.7 - 13.1 |
| ZREAL := XREAL / YREAL; | 11.8 - 13.0 | | 16.5 - 17.7 |
| ZREAL := XREAL DIV YREAL; | 6.6 | | |
| ZREAL := 1; | 2.8 | | 4.5 |
| ZREAL := 1.0; | 5.4 | | 8.0 |
| ZREAL := 1@0; | 5.4 | | 8.0 |
| ZREAL := 97; | 3.2 | | 5.1 |
| ZREAL := 9701; | 3.6 | | 5.6 |
| ZREAL := 99999; | 4.6 | | 7.6 |
| ZREAL := 100000; | 5.4 | | 8.0 |
| ZREAL := 100001; | 4.6 | | 7.6 |
| ZREAL := 12345678; | 4.6 | | 7.6 |
| ZREAL := XREAL; | 3.8 | | 6.8 |
| ZREAL := XREAL + XREAL; | 5.0 | | 8.2 |
| ZREAL := XREAL + XREAL + XREAL; | 6.6 | | 11.6 |
| ZREAL := XREAL + XREAL + XREAL + XREAL; | 8.8 | | 14.9 |
| ZREAL := XREAL + XREAL + XREAL + XREAL + XREAL; | 10.0 | | 18.3 |
| YREAL := YREAL + XREAL + XREAL + XREAL + XREAL + XREAL; | 10.8 | | |
| XREAL := XREAL + XREAL + XREAL + XREAL + XREAL + XREAL; | 11.8 | | 21.6 |
| ZREAL := XREAL + XREAL + XREAL + XREAL + XREAL + XREAL + XREAL; | 13.4 | | 24.9 |
| ZREAL := XREAL + XREAL + XREAL + XREAL + XREAL + XREAL + XREAL + XREAL; | 15.2 | | |
| ZREAL := ABS(XREAL); | 5.5 | | 8.8 |
| ZREAL := COSH (XREAL); | 221.7 | * | 337.9 |
| ZREAL := EXP (XREAL); | 166.1 | * | 253.9 |

| | |
|---|---|
| Proc Speed | 5.00/10.00 |
| MCP | Mark II.6 |
| ALGOL | Mark II.6 |

## SINGLE PRECISION REAL CONSTRUCTS (Cont)

| ALGOL Construct | Timings in Microseconds 1.2 $\mu$s Memory | | 1.6 $\mu$s Memory |
|---|---|---|---|
| ZREAL := SIN (XREAL); | 145.1 | * | 221.5 |
| ZREAL := SQRT (XREAL); | 171.5 | * | 252.3 |
| ZREAL := XREAL + SQRT (XREAL); | 174.4 | * | 257.8 |
| ZREAL := SQRT (ZREAL); | 171.5 | * | 252.3 |
| YREAL := YREAL + SQRT (YREAL); | 169.6 | * | 252.3 |

## INTEGER CONSTRUCTS

| ALGOL Construct | Timings in Microseconds 1.2 $\mu$s Memory | 1.6 $\mu$s Memory |
|---|---|---|
| ZINTGR := XINTGR + YINTGR; | 6.1 - 6.5 | 10.9 - 11.3 |
| ZINTGR := XINTGR - YINTGR; | 6.1 - 6.5 | 10.9 - 11.3 |
| ZINTGR := XINTGR * YINTGR; | 7.5 - 7.9 | 12.3 - 12.7 |
| ZINTGR := XINTGR / YINTGR; | 14.9 - 15.5 | 19.7 - 20.3 |
| ZINTGR := 1; | 2.8 | 4.5 |
| ZINTGR := 97; | 3.2 | 5.1 |
| ZINTGR := XINTGR; | 4.5 | 7.8 |
| ZINTGR := ZINTGR; | 3.0 | 4.8 |

ALGOL Construct Timings

Proc Speed : 5.00/10.00
MCP : Mark II.6
ALGOL : Mark II.6

## DOUBLE PRECISION REAL CONSTRUCTS

| ALGOL Construct | Timings in Microseconds: 1.2 $\mu$s Memory | | Timings in Microseconds: 1.6 $\mu$s Memory |
|---|---|---|---|
| ZDBL := XDBL + YDBL; | 10.0 - 11.6 | | 17.6 - 19.2 |
| ZDBL := XDBL - YDBL; | 10.0 - 11.6 | | 17.6 - 19.2 |
| ZDBL := XDBL * YDBL; | 16.6 | | 24.2 |
| ZDBL := XDBL / YDBL; | 38.6 - 51.0 | | 48.2 - 60.6 |
| ZDBL := 1@@0; | 6.6 | | 10.6 |
| ZDBL := 8@@0; | 6.6 | | 10.6 |
| ZDBL := XDBL; | 6.4 | | 11.4 |
| ZDBL := DABS (XDBL); | 8.2 | | 13.4 |
| ZDBL := DEXP (XDBL); | 571.7 | * | 836.3 |
| ZDBL := DSIN (XDBL); | 382.9 | * | 576.7 |
| ZDBL := DSQRT (XDBL); | 237.3 | * | 344.5 |

Proc Speed : 5.00/10.00
MCP : Mark II.6
ALGOL : Mark II.6

## ARRAY CONSTRUCTS

| ALGOL Construct | Timings in Microseconds: 1.2 µs Memory | 1.6 µs Memory |
|---|---|---|
| ARAY1 [1] := XREAL; | 5.6 | 9.9 |
| ARAY1 [1] := BRAY1 [1]; | 7.6 | 13.0 |
| ARAY1 [I] := XREAL; | 6.7 | 12.3 |
| ARAY1 [I] := BRAY1 [I]; | 9.5 | 17.6 |
| ZREAL := ARAY1 [I] + BRAY1 [I]; | 11.2 - 12.6 | 20.6 - 22.0 |
| ZREAL := ARAY1 [I] - BRAY1 [I]; | 11.6 - 12.8 | 21.0 - 22.2 |
| ZREAL := ARAY1 [I] * BRAY1 [I]; | 14.0 | 23.4 |
| ZREAL := ZRZY1 [I] / BRAY1 [I]; | 17.4 | 26.8 |
| ARAY2 [1,1] := XREAL; | 8.7 | 14.3 |
| ARAY2 [1,1] := BRAY2 [1,1]; | 14.7 | 24.6 |
| ARAY2 [I,J] := XREAL; | 10.5 | 18.7 |
| ARAY2 [I,J] := BRAY2 [I,J]; | 18.4 | 33.4 |
| ZREAL := ARAY2 [I,J] + BRAY2 [I,J]; | 18.9 - 20.3 | 35.3 - 36.7 |
| ARAY3 [1,1,1] := XREAL; | 11.9 | 18.9 |
| ARAY3 [1,1,1] := BRAY3 [1,1,1]; | 23.7 | 38.4 |
| ARAY3 [I,J,K] := XREAL; | 14.4 | 25.3 |
| ARAY3 [I,J,K] := BRAY3 [I,J,K]; | 29.6 | 52.1 |
| ZREAL = ARAY3 [I,J,K] + BRAY3 [I,J,K]; | 33.5 - 35.1 | 59.2 - 60.6 |

NOTE

These constructs call upon intrinsic routines of which each must be linked from the D1 stack prior to its first execution within a program. The first timing which includes intrinsic linkage will vary between 2 and 160 milliseconds, depending upon the presence or absence of certain MCP segments in main memory. The above timings were observed after the corresponding intrinsic was called once and linkage was thus established.

# Section 3
# COBOL CONSTRUCT TIMINGS

The Mark II.6 MCP and COBOL compiler were used in obtaining the following construct timings. Both 1.2 µs and 1.6 µs memories were used, and the resultant timings for both memory speeds are listed beside each construct on the following pages.

The constructs have been grouped into twenty different "types" representing common data names and thus common data representations (pictures).

Two timings are given for each construct within each memory speed in the first ten construct types. The first timing corresponds with the operand values:

"X-..." = +1 and "Y-..." = +1.

The second timing resulted from using the operand values:

"X-..." = +6517 and "Y-..." = -1740.

The data names and their corresponding pictures and usages for the constructs timed are listed by type in the following WORKING-STORAGE SECTION.

## DATA DIVISION.
## WORKING-STORAGE SECTION.

### TYPES 1 AND 11

```
77  X-S7-CMP1          PIC S9(7)        COMP-1.
77  Y-S7-CMP1          PIC S9(7)        COMP-1.
77  Z-S7-CMP1          PIC S9(7)        COMP-1.
```

### TYPE 2

```
77  X-S5V2-CMP1        PIC S9(5)V99     COMP-1.
77  Y-S5V2-CMP1        PIC S9(5)V99     COMP-1.
77  Z-S5V2-CMP1        PIC S9(5)V99     COMP-1.
```

### TYPE 3

```
77  X-J7               PIC J9(7).
77  Y-J7               PIC J9(7).
77  Z-J7               PIC J9(7).
```

### TYPE 4

```
77  X-J5V2             PIC J9(5)V99.
77  Y-J5V2             PIC J9(5)V99.
77  Z-J5V2             PIC J9(5)V99.
```

### TYPE 15 (PART 1)

```
77  X-S5-CMP1          PIC S9(5)        COMP-1
77  Y-J5               PIC J9(5).
```

### TYPE 16 (PART 1)

```
77  X-S10-CMP1         PIC S9(10)       COMP-1.
77  Y-J8               PIC J9(8).
```

### TYPE 17

```
77  X-3-CMP1           PIC 999          COMP-1.
77  Y-6-CMP1           PIC 9(6)         COMP-1.
77  Z-6-CMP2           PIC 9(6)         COMP-2.
77  W-J6-CMP2          PIC J9(6)        COMP-2.
```

### TYPE 18

```
77  NUM-3-CMP1         PIC 999          COMP-1.
```

### TYPE 19 (PART 1)

```
77  SUB-2-CMP1         PIC 99           COMP-1.
77  INX1               INDEX.
77  X-X5               PIC X(5).
```

```
     TYPE 5

01   WS-OP-1.
     03   X-S7-CMP                  PIC S9(7)           COMP.
     03   Y-S7-CMP                  PIC S9(7)           COMP.
     03   Z-S7-CMP                  PIC S9(7)           COMP.

     TYPE 6

01   WS-OP-2.
     03   X-S5V2-CMP                PIC S9(5)V99        COMP.
     03   Y-S5V2-CMP                PIC S9(5)V99        COMP.
     03   Z-S5V2-CMP                PIC S9(5)V99        COMP.

     TYPE 7

01   WS-OP-3.
     03   X-03-J7                   PIC J9(7).
     03   Y-03-J7                   PIC J9(7).
     03   Z-03-J7                   PIC J9(7).

     TYPE 8

01   WS-OP-4.
     03   X-03-J5V2                 PIC J9(5)V99.
     03   Y-03-J5V2                 PIC J9(5)V99.
     03   Z-03-J5V2                 PIC J9(5)V99.

     TYPE 9

01   WS-OP-5.
     03   X-S7-CMP2                 PIC S9(7)           COMP-2.
     03   Y-S7-CMP2                 PIC S9(7)           COMP-2.
     03   Z-S7-CMP2                 PIC S9(7)           COMP-2.

     TYPE 10

01   WS-OP-6.
     03   X-J7-CMP2                 PIC J9(7)           COMP-2.
     03   Y-J7-CMP2                 PIC J9(7)           COMP-2.
     03   Z-J7-CMP2                 PIC J9(7)           COMP-2.

     TYPE 12

01   WS-OP-7.
     03   X1                        PIC X.
     03   X6                        PIC X(6).
     03   X25                       PIC X(25).
     03   X132                      PIC X(132).
```

COBOL Construct Timings

**TYPE 13**

```
01  WS-OP-8.
    03  X-7-CMP                       PIC 9(7)           COMP.
    03  Y-5V2-CMP                     PIC 9(5)V99        COMP.
    03  Z-Z7                          PIC Z(7).
    03  W-Z5P2                        PIC Z(5).99.
```

**TYPE 14**

```
01  WS-OP-9.
    03  X-S7-COMP                     PIC S9(7)          COMP.
    03  Y-S7-COMP                     PIC S9(7)          COMP.
    03  Z-S7-COMP                     PIC S9(7)          COMP.
    03  A-J7                          PIC J9(7).
    03  B-J7                          PIC J9(7).
    03  C-J7                          PIC J9(7).
```

**TYPE 15 (PART 2)**

```
01  WS-OP-10.
    03  S-S5-CMP                      PIC S9(5)          COMP.
    03  T-S5                          PIC S9(5).
    03  U-J5                          PIC J9(5).
    03  V-S5-CMP2                     PIC S9(5)          COMP-2.
    03  W-J5-CMP2                     PIC J9(5)          COMP-2.
```

**TYPE 16 (PART 2)**

```
01  WS-OP-11.
    03  U-S6-CMP                      PIC S9(6)          COMP.
    03  V-J4                          PIC J9(4).
    03  W-J9-CMP2                     PIC J9(9)          COMP-2.
    03  A-X8                          PIC X(8).
    03  B-X4                          PIC X(4).
     88  B88    VALUE "YES"
01  WS-OP-12  DISPLAY-1.
    03  C-X4-DISP1                    PIC X(4) DISPLAY-1.
```

**TYPE 19 (PART 2)**

```
01  WS-OP-13.
    03  SUB-X5-OC-10                  PIC X(5) OCCURS 10 TIMES.
01  WS-OP-14.
    03  IDX-X5-OC-10                  PIC X(5) OCCURS 10 TIMES
                                               INDEXED BY INX2.
```

**TYPE 20**

```
01  WS-OP-15.
    03  XX-X25                        PIC X   OCCURS 25 TIMES.
```

Proc Speed : 5.00/10.00
MCP : Mark II.6
COBOL : Mark II.6

## TYPE 1 CONSTRUCTS

| COBOL Construct | Timings in Microseconds 1.2 μs Memory | 1.6 μs Memory |
|---|---|---|
| ADD X-S7-CMP1, Y-S7-CMP1 GIVING Z-S7-CMP1. | 5.6 - 6.0 | 10.3 - 10.7 |
| COMPUTE Z-S7-CMP1 = X-S7-CMP1 + Y-S7-CMP1. | 8.0 - 8.4 | 14.7 - 15.1 |
| SUBTRACT Y-S7-CMP1 FROM X-S7-CMP1 GIVING Z-S7-CMP1. | 6.3 - 6.7 | 11.2 - 11.6 |
| COMPUTE Z-S7-CMP1 = X-S7-CMP1 - Y-S7-CMP1. | 8.0 - 8.4 | 14.7 - 15.1 |
| MULTIPLY X-S7-CMP1 BY Y-S7-CMP1 GIVING Z-S7-CMP1. | 10.5 - 10.9 | 17.5 - 17.9 |
| COMPUTE Z-S7-CMP1 = X-S7-CMP1 * Y-S7-CMP1. | 10.0 - 10.4 | 16.9 - 17.3 |
| DIVIDE X-S7-CMP1 BY Y-S7-CMP1 GIVING Z-S7-CMP1. | 17.5 - 17.9 | 24.5 - 24.9 |
| COMPUTE Z-S7-CMP1 = X-S7-CMP1 / Y-S7-CMP1. | 17.0 - 17.4 | 23.9 - 24.3 |

## TYPE 2 CONSTRUCTS

| COBOL Construct | Timings in Microseconds 1.2 μs Memory | 1.6 μs Memory |
|---|---|---|
| ADD X-S5V2-CMP1, Y-S5V2-CMP1 GIVING Z-S5V2-CMP1. | 5.6 - 6.0 | 10.3 - 10.7 |
| COMPUTE Z-S5V2-CMP1 = X-S5V2-CMP1 + Y-S5V2-CMP1. | 8.0 - 8.4 | 14.7 - 15.1 |
| SUBTRACT Y-S5V2-CMP1 FROM X-S5V2-CMP1 GIVING Z-S5V2-CMP1. | 6.3 - 6.7 | 11.2 - 11.6 |
| COMPUTE Z-S5V2-CMP1 = X-S5V2-CMP1 - Y-S5V2-CMP1. | 8.0 - 8.4 | 14.7 - 15.1 |
| MULTIPLY X-S5V2-CMP1 BY Y-S5V2-CMP1 GIVING Z-S5V2-CMP1. | 20.6 - 21.0 | 28.0 - 28.4 |
| COMPUTE Z-S5V2-CMP1 = X-S5V2-CMP1 * Y-S5V2-CMP1. | 20.1 - 20.5 | 27.2 - 27.6 |
| DIVIDE X-S5V2-CMP1 BY Y-S5V2-CMP1 GIVING Z-S5V2-CMP1. | 21.1 - 21.5 | 28.7 - 29.1 |
| COMPUTE Z-S5V2-CMP1 = X-S5V2-CMP1 / Y-S5V2-CMP1. | 20.8 - 21.2 | 28.2 - 28.6 |

COBOL Construct Timings

Proc Speed : 5.00/10.00
MCP : Mark II.6
COBOL : Mark II.6

## TYPE 3 CONSTRUCTS

| COBOL Construct | Timings in Microseconds: 1.2 µs Memory | 1.6 µs Memory |
|---|---|---|
| ADD X-J7, Y-J7 GIVING Z-J7. | 169.0 - 169.4 | 235.4 - 235.8 |
| COMPUTE Z-J7 = X-J7 + Y-J7. | 166.2 - 166.7 | 230.6 - 231.1 |
| SUBTRACT Y-J7 FROM X-J7 GIVING Z-J7. | 167.2 - 169.2 | 233.4 - 235.4 |
| COMPUTE Z-J7 = X-J7 - Y-J7. | 164.2 - 166.2 | 228.6 - 230.6 |
| MULTIPLY X-J7 BY Y-J7 GIVING Z-J7. | 172.9 - 173.3 | 241.4 - 241.7 |
| COMPUTE Z-J7 = X-J7 * Y-J7. | 167.6 - 168.0 | 232.1 - 232.4 |
| DIVIDE X-J7 BY Y-J7 GIVING Z-J7. | 178.0 - 178.4 | 244.8 - 245.2 |
| COMPUTE Z-J7 = X-J7 / Y-J7. | 175.1 - 175.5 | 239.6 - 240.0 |

## TYPE 4 CONSTRUCTS

| COBOL Construct | Timings in Microseconds: 1.2 µs Memory | 1.6 µs Memory |
|---|---|---|
| ADD X-J5V2, Y-J5V2 GIVING Z-J5V2. | 169.0 - 169.4 | 235.4 - 235.8 |
| COMPUTE Z-J5V2 = X-J5V2 + Y-J5V2. | 166.2 - 166.6 | 230.6 - 231.0 |
| SUBTRACT Y-J5V2 FROM X-J5V2 GIVING Z-J5V2. | 166.9 - 169.0 | 233.4 - 235.4 |
| COMPUTE Z-J5V2 = X-J5V2 - Y-J5V2. | 164.2 - 166.2 | 228.6 - 230.6 |
| MULTIPLY X-J5V2 BY Y-J5V2 GIVING Z-J5V2. | 183.2 - 183.6 | 252.0 - 252.4 |
| COMPUTE Z-J5V2 = X-J5V2 * Y-J5V2. | 177.7 - 178.1 | 242.5 - 242.8 |
| DIVIDE X-J5V2 BY Y-J5V2 GIVING Z-J5V2. | 181.4 - 181.8 | 248.8 - 249.2 |
| COMPUTE Z-J5V2 = X-J5V2 / Y-J5V2. | 179.0 - 179.5 | 244.1 - 244.5 |

Proc Speed 5.00/10.00
MCP Mark II.6
COBOL Mark II.6

## TYPE 5 CONSTRUCTS

| COBOL Construct | Timings in Microseconds: 1.2 $\mu s$ Memory | Timings in Microseconds: 1.6 $\mu s$ Memory |
|---|---|---|
| ADD X-S7-CMP, Y-S7-CMP GIVING Z-S7-CMP. | 12.3 - 12.7 | 20.7 - 21.1 |
| COMPUTE Z-S7-CMP = X-S7-CMP + Y-S7-CMP. | 13.8 - 14.2 | 24.2 - 24.6 |
| SUBTRACT Y-S7-CMP FROM X-S7-CMP GIVING Z-S7-CMP. | 12.4 - 12.8 | 21.1 - 21.5 |
| COMPUTE Z-S7-CMP = X-S7-CMP - Y-S7-CMP. | 13.8 - 14.2 | 24.2 - 24.6 |
| MULTIPLY X-S7-CMP BY Y-S7-CMP GIVING Z-S7-CMP. | 16.3 - 16.7 | 26.8 - 27.2 |
| COMPUTE Z-S7-CMP = X-S7-CMP * Y-S7-CMP. | 15.8 - 16.2 | 26.3 - 26.7 |
| DIVIDE X-S7-CMP BY Y-S7-CMP GIVING Z-S7-CMP. | 23.3 - 23.7 | 33.8 - 34.2 |
| COMPUTE Z-S7-CMP = X-S7-CMP / Y-S7-CMP. | 22.8 - 23.2 | 33.4 - 33.8 |

## TYPE 6 CONSTRUCTS

| COBOL Construct | Timings in Microseconds: 1.2 $\mu s$ Memory | Timings in Microseconds: 1.6 $\mu s$ Memory |
|---|---|---|
| ADD X-S5V2-CMP, Y-S5V2-CMP GIVING Z-S5V2-CMP. | 12.1 - 12.5 | 20.7 - 21.1 |
| COMPUTE Z-S5V2-CMP = X-S5V2-CMP + Y-S5V2-CMP. | 13.8 - 14.2 | 24.2 - 24.6 |
| SUBTRACT Y-S5V2-CMP FROM Y-S5V2-CMP GIVING Z-S5V2-CMP. | 12.1 - 12.5 | 20.7 - 21.1 |
| COMPUTE Z-S5V2-CMP = X-S5V2-CMP - Y-S5V2-CMP. | 13.8 - 14.2 | 24.2 - 24.6 |
| MULTIPLY X-S5V2-CMP BY Y-S5V2-CMP GIVING Z-S5V2-CMP. | 27.2 - 27.6 | 38.9 - 39.3 |
| COMPUTE Z-S5V2-CMP = X-S5V2-CMP * Y-S5V2-CMP. | 25.9 - 26.3 | 36.7 - 37.1 |
| DIVIDE X-S5V2-CMP BY Y-S5V2-CMP GIVING Z-S5V2-CMP. | 27.2 - 27.6 | 38.9 - 39.3 |
| COMPUTE Z-S5V2-CMP = X-S5V2-CMP / Y-S5V2-CMP. | 26.6 - 27.0 | 37.7 - 38.1 |

COBOL Construct Timings

| | |
|---|---|
| Proc Speed | 5.00/10.00 |
| MCP | Mark II.6 |
| COBOL | Mark II.6 |

## TYPE 7 CONSTRUCTS

| COBOL Construct | Timings in Microseconds: 1.2 µs Memory | 1.6 µs Memory |
|---|---|---|
| ADD X-03-J7, Y-03-J7 GIVING Z-03-J7. | 168.0 - 168.3 | 234.3 - 234.8 |
| COMPUTE Z-03-J7 = X-03-J7 + Y-03-J7. | 165.2 - 165.6 | 229.6 - 230.0 |
| SUBTRACT Y-03-J7 FROM X-03-J7 GIVING Z-03-J7. | 169.3 - 171.3 | 237.1 - 239.1 |
| COMPUTE Z-03-J7 = X-03-J7 - Y-03-J7. | 163.2 - 165.2 | 227.6 - 229.6 |
| MULTIPLY X-03-J7 BY Y-03-J7 GIVING Z-03-J7. | 171.9 - 172.3 | 240.3 - 240.8 |
| COMPUTE Z-03-J7 = X-03-J7 * Y-03-J7. | 166.6 - 167.0 | 231.1 - 231.4 |
| DIVIDE X-03-J7 BY Y-03-J7 GIVING Z-03-J7. | 177.0 - 177.4 | 243.7 - 244.1 |
| COMPUTE Z-03-J7 = X-03-J7 / Y-03-J7. | 174.1 - 174.5 | 238.6 - 239.0 |

## TYPE 8 CONSTRUCTS

| COBOL Construct | Timings in Microseconds: 1.2 µs Memory | 1.6 µs Memory |
|---|---|---|
| ADD X-03-J5V2, Y-03-J5V2 GIVING Z-03-J5V2. | 168.0 - 168.3 | 234.3 - 234.8 |
| COMPUTE Z-03-J5V2 = X-03-J5V2 + Y-03-J5V2. | 165.2 - 165.6 | 229.6 - 230.0 |
| SUBTRACT Y-03-J5V2 FROM X-03-J5V2 GIVING Z-03-J5V2. | 165.9 - 167.9 | 232.4 - 234.3 |
| COMPUTE Z-03-J5V2 = X-03-J5V2 - Y-03-J5V2. | 163.2 - 165.2 | 227.6 - 229.6 |
| MULTIPLY X-03-J5V2 BY Y-03-J5V2 GIVING Z-03-J5V2. | 182.2 - 182.6 | 251.0 - 251.5 |
| COMPUTE Z-03-J5V2 = X-03-J5V2 * Y-03-J5V2. | 176.8 - 177.1 | 241.5 - 241.9 |
| DIVIDE X-03-J5V2 BY Y-03-J5V2 GIVING Z-03-J5V2. | 180.4 - 180.8 | 247.8 - 248.3 |
| COMPUTE Z-03-J5V2 = X-03-J5V2 / Y-03-J5V2. | 178.1 - 178.5 | 243.1 - 243.5 |

Proc Speed : 5.00/10.00
MCP : Mark II.6
COBOL : Mark II.6

## TYPE 9 CONSTRUCTS

| COBOL Construct | Timings in Microseconds 1.2 $\mu s$ Memory | 1.6 $\mu s$ Memory |
|---|---|---|
| ADD X-S7-CMP2, Y-S7-CMP2 GIVING Z-S7-CMP2. | 94.6 - 95.0 | 125.3 - 125.7 |
| COMPUTE Z-S7-CMP2 = X-S7-CMP2 + Y-S7-CMP2. | 91.8 - 92.2 | 120.3 - 120.7 |
| SUBTRACT Y-S7-CMP2 FROM X-S7-CMP2 GIVING Z-S7-CMP2. | 92.6 - 94.6 | 123.3 - 125.3 |
| COMPUTE Z-S7-CMP2 = X-S7-CMP2 - Y-S7-CMP2. | 89.8 - 91.8 | 118.3 - 120.3 |
| MULTIPLY X-S7-CMP2 BY Y-S7-CMP2 GIVING Z-S7-CMP2. | 98.5 - 98.9 | 131.3 - 131.7 |
| COMPUTE Z-S7-CMP2 = X-S7-CMP2 * Y-S7-CMP2. | 93.2 - 93.6 | 121.7 - 122.1 |
| DIVIDE X-S7-CMP2 BY Y-S7-CMP2 GIVING Z-S7-CMP2. | 103.6 - 104.0 | 134.5 - 134.9 |
| COMPUTE Z-S7-CMP2 = X-S7-CMP2 / Y-S7-CMP2. | 100.9 - 101.3 | 129.5 - 130.1 |

## TYPE 10 CONSTRUCTS

| COBOL Construct | Timings in Microseconds 1.2 $\mu s$ Memory | 1.6 $\mu s$ Memory |
|---|---|---|
| ADD X-J7-CMP2, Y-J7-CMP2 GIVING Z-J7-CMP2. | 138.5 - 139.1 | 187.5 - 188.1 |
| COMPUTE Z-J7-CMP2 = X-J7-CMP2 + Y-J7-CMP2. | 135.7 - 136.3 | 182.6 - 183.2 |
| SUBTRACT Y-J7-CMP2 FROM X-J7-CMP2 GIVING Z-J7-CMP2. | 136.5 - 138.7 | 185.5 - 187.7 |
| COMPUTE Z-J7-CMP2 = X-J7-CMP2 - Y-J7-CMP2. | 133.7 - 135.9 | 180.6 - 182.8 |
| MULTIPLY X-J7-CMP2 BY Y-J7-CMP2 GIVING Z-J7-CMP2. | 144.9 - 145.4 | 197.8 - 198.4 |
| COMPUTE Z-J7-CMP2 = X-J7-CMP2 * Y-J7-CMP2. | 139.5 - 140.1 | 188.4 - 189.0 |
| DIVIDE X-J7-CMP2 BY Y-J7-CMP2 GIVING Z-J7-CMP2. | 147.5 - 148.0 | 196.8 - 197.4 |
| COMPUTE Z-J7-CMP2 = X-J7-CMP2 / Y-J7-CMP2. | 144.7 - 145.2 | 191.7 - 192.3 |

COBOL Construct Timings

Proc Speed : 5.00/10.00
MCP : Mark II.6
COBOL : Mark II.6

## TYPE 11 CONSTRUCTS

| COBOL Construct | Timings in Microseconds: 1.2 $\mu$s Memory | Timings in Microseconds: 1.6 $\mu$s Memory |
|---|---|---|
| ADD 1 TO X-S7-CMP1. | 4.7 | 8.0 |
| COMPUTE X-S7-CMP1 = X-S7-CMP1 + 1. | 6.7 | 12.1 |
| ADD 1, X-S7-CMP1 GIVING Z-S7-CMP1. | 4.5 | 7.8 |
| COMPUTE Z-S7-CMP1 = X-S7-CMP1 + 1. | 6.7 | 12.1 |
| ADD 1, X-S7-CMP1, Y-S7-CMP1 GIVING Z-S7-CMP1. | 6.2 - 6.6 | 11.2 - 11.6 |
| COMPUTE Z-S7-CMP1 = Y-S7-CMP1 + X-S7-CMP1 + 1. | 8.7 - 9.1 | 15.7 - 16.1 |
| MULTIPLY 1 BY X-S7-CMP1. | 9.6 - 10.0 | 15.3 - 15.7 |
| MULTIPLY 1 BY X-S7-CMP1 GIVING Z-S7-CMP1. | 9.6 - 10.0 | 15.3 - 15.7 |
| DIVIDE 1 INTO X-S7-CMP1. | 15.7 | 21.2 |
| DIVIDE 1 INTO X-S7-CMP1 GIVING Z-S7-CMP1. | 16.6 | 22.3 |
| DIVIDE X-S7-CMP1 BY 2 GIVING Z-S7-CMP1 REMAINDER Y-S7-CMP1. | 44.4 - 46.2 | 66.5 - 68.3 |

NOTE

Where there are two timings for a construct within a memory speed above, the first timing represents operand values

X-S7-CMP1 = +1 and Y-S7-CMP1 = +1

and the second timing represents operand values

X-S7-CMP1 = +6517 and Y-S7-CMP1 = -1740

Proc Speed 5.00/10.00
MCP Mark II.6
COBOL Mark II.6

## TYPE 12 CONSTRUCTS

| COBOL Construct | Timings in Microseconds: 1.2 µs Memory | Timings in Microseconds: 1.6 µs Memory |
|---|---|---|
| MOVE SPACE TO X1. | 20.1 | 30.6 |
| MOVE SPACE TO X6. | 25.9 | 37.7 |
| MOVE SPACE TO X25. | 37.7 | 52.7 |
| MOVE SPACE TO X132. | 79.1 | 122.6 |
| MOVE SPACE TO WS-OP-7. | 69.4 | 111.6 |
| MOVE SPACE TO X1, X6, X25, X132. | 162.8 | 243.7 |
| MOVE "A" TO X1. | 18.1 | 24.6 |
| MOVE "A" TO X6. | 35.9 | 49.3 |
| MOVE "A" TO X25. | 48.9 | 69.1 |
| MOVE ALL "A" TO X6. | 27.2 | 39.4 |
| MOVE ALL "A" TO X25. | 38.8 | 54.0 |
| MOVE ALL "A" TO X6, X25. | 66.0 | 93.4 |
| MOVE X6 TO X25. | 62.4 | 90.2 |
| MOVE X25 TO X6. | 34.4 | 49.2 |

## TYPE 13 CONSTRUCTS

| COBOL Construct | Timings in Microseconds: 1.2 µs Memory | Timings in Microseconds: 1.6 µs Memory |
|---|---|---|
| MOVE 0 TO X-7-CMP. | 4.8 | 8.2 |
| MOVE 0 TO Y-5V2-CMP. | 4.6 | |
| MOVE 0.00 TO Y-5V2-CMP. | 4.6 | |
| MOVE 2 TO X-7-CMP. | 5.2 | |
| MOVE 97 TO X-7-CMP. | 5.2 | 8.4 |
| MOVE 9.7 TO Y-5V2-CMP. | 13.0 | 18.6 |
| MOVE 12345 TO X-7-CMP. | 5.5 | 8.9 |
| MOVE 123.45 TO Y-5V2-CMP. | 5.5 | 8.9 |
| MOVE X-7-CMP TO Y-5V2-CMP. | 14.6 | 22.4 |
| MOVE Y-5V2-CMP TO X-7-CMP. | 17.0 | 24.9 |
| MOVE X-7-CMP TO Z-Z7. | 37.4 | 53.4 |
| MOVE Y-5V2-CMP TO W-Z5P2. | 83.2 | 118.5 |

COBOL Construct Timings

Proc Speed 5.00/10.00
MCP Mark II.6
COBOL Mark II.6

## TYPE 14 CONSTRUCTS

| COBOL Construct | Relationship of Operands | Timings in Microseconds: 1.2 us Memory | 1.6 us Memory |
|---|---|---|---|
| P100. GO TO P200. | | 1.7 | 2.8 |
| P200. IF X-S7-COMP = Y-S7-COMP GO TO P300. | EQUAL | 9.9 | 16.7 |
| P200. IF X-S7-COMP = Y-S7-COMP GO TO P300. | NOT EQUAL | 8.7 | 14.5 |
| P300. IF A-J7 = B-J7 GO TO P400. | EQUAL | 109.3 | 150.2 |
| P300. IF A-J7 = B-J7 GO TO P400. | NOT EQUAL | 108.1 | 148.0 |
| IF X-S7-COMP = Y-S7-COMP ADD 1 TO Z-S7-COMP. | EQUAL | 19.1 | 31.7 |
| IF X-S7-COMP = Y-S7-COMP ADD 1 TO Z-S7-COMP. | NOT EQUAL | 9.3 | 15.9 |
| IF X-S7-COMP < Y-S7-COMP ADD 1 TO Z-S7-COMP. | LESS THAN | 18.9 | 31.5 |
| IF X-S7-COMP < Y-S7-COMP ADD 1 TO Z-S7-COMP. | NOT LESS THAN | 9.5 | 16.1 |
| IF X-S7-COMP > Y-S7-COMP ADD 1 TO Z-S7-COMP. | GREATER THAN | 19.1 | 31.7 |
| IF X-S7-COMP > Y-S7-COMP ADD 1 TO Z-S7-COMP. | NOT GREATER THAN | 9.5 | 16.1 |
| IF A-J7 = B-J7 ADD 1 TO C-J7. | EQUAL | 215.6 | 297.7 |
| IF A-J7 = B-J7 ADD 1 TO C-J7 | NOT EQUAL | 108.9 | 149.6 |
| COMPUTE Y-S7-COMP = ABS(X-S7-COMP). | X-S7-COMP = 1 | 11.5 | 19.2 |
| COMPUTE Y-S7-COMP = ABS(X-S7-COMP). | X-S7-COMP = -1 | 11.5 | 19.2 |
| COMPUTE Y-S7-COMP = SIN(X-S7-COMP). | X-S7-COMP = 1 | 149.5 | * 228.4 |
| COMPUTE Y-S7-COMP = SIN(X-S7-COMP). | X-S7-COMP = 4 | 198.5 | * 304.2 |
| COMPUTE Y-S7-COMP = SQRT(X-S7-COMP). | X-S7-COMP = 1 | 175.9 | * 259.2 |
| COMPUTE Y-S7-COMP = SQRT(X-S7-COMP). | X-S7-COMP = 4 | 167.9 | * 251.2 |

* These constructs call upon intrinsic routines of which each must be linked from the D1 stack prior to its first execution within a program. The first timing, which includes intrinsic linkage, will vary between 2 and 160 milliseconds, depending upon the presence or absence of certain MCP segments in main memory. The timings shown for these constructs were observed after the corresponding intrinsic was called once and linkage was thus established.

Proc Speed 5.00/10.00
MCP Mark II.6
COBOL Mark II.6

## TYPE 15 CONSTRUCTS

NOTE: All conditions below are satisfied (equal)

| | COBOL Construct | Timings in Microseconds 1.2 μs Memory | 1.6 μs Memory |
|---|---|---|---|
| P110. | IF X-S5-CMP1 = Y-J5 GO TO P210. | 56.2 | 78.9 |
| P210. | IF X-S5-CMP1 = S-S5-CMP GO TO P310. | 7.6 | 12.9 |
| P310. | IF X-S5-CMP1 = T-S5 GO TO P410. | 34.1 | 47.7 |
| P410. | IF X-S5-CMP1 = U-J5 GO TO P510. | 56.5 | 79.5 |
| P510. | IF X-S5-CMP1 = V-S5-CMP2 GO TO P610. | 36.2 | 50.8 |
| P610. | IF X-S5-CMP1 = W-J5-CMP2 GO TO P710. | 51.6 | 71.3 |
| P710. | IF Y-J5 = S-S5-CMP GO TO P810. | 55.7 | 77.2 |
| P810. | IF Y-J5 = T-S5 GO TO P910. | 81.5 | 110.8 |
| P910. | IF Y-J5 = U-J5 GO TO Q110. | 105.7 | 145.2 |
| Q110. | IF Y-J5 = V-S5-CMP2 GO TO Q210. | 83.6 | 113.9 |
| Q210. | IF Y-J5 = W-J5-CMP2 GO TO Q310. | 99.4 | 135.0 |
| Q310. | IF S-S5-CMP = T-S5 GO TO Q410. | 36.0 | 50.8 |
| Q410. | IF S-S5-CMP = U-J5 GO TO Q510. | 59.6 | 84.3 |
| Q510. | IF S-S5-CMP = V-S5-CMP2 GO TO Q610. | 38.0 | 53.8 |
| Q610. | IF S-S5-CMP = W-J5-CMP2 GO TO Q710. | 53.6 | 74.5 |
| Q710. | IF T-S5 = U-J5 GO TO Q810. | 83.8 | 114.4 |
| Q810. | IF T-S5 = V-S5-CMP2 GO TO Q910. | 61.6 | 82.7 |
| Q910. | IF T-S5 = W-J5-CMP2 GO TO R110. | 77.3 | 103.8 |
| R110. | IF U-J5 = V-S5-CMP2 GO TO R210. | 85.7 | 117.3 |
| R210. | IF U-J5 = W-J5-CMP2 GO TO R310. | 101.4 | 138.3 |
| R310. | IF V-S5-CMP2 = W-J5-CMP2 GO TO R410. | 79.3 | 106.8 |
| R410. | IF X-S5-CMP1 = +1 GO TO R510. | 5.3 | 8.6 |
| R510. | | | |

COBOL Construct Timings

Proc Speed : 5.00/10.00
MCP : Mark II.6
COBOL : Mark II.6

## TYPE 16 CONSTRUCTS

NOTE: All conditions below are satisfied (equal)

| | COBOL Construct | Timings in Microseconds 1.2 $\mu$s Memory | 1.6 $\mu$s Memory |
|---|---|---|---|
| P120. | IF X-S10-CMP1 = Y-J8 GO TO P220. | 56.0 | 79.6 |
| P220. | IF X-S10-CMP1 = U-S6-CMP GO TO P320. | 7.7 | 13.0 |
| P320. | IF X-S10-CMP1 = V-J4 GO TO P420. | 55.0 | 77.4 |
| P420. | IF X-S10-CMP1 = W-J9-CMP2 GO TO P520. | 55.0 | 76.0 |
| P520. | IF Y-J8 = U-S6-CMP GO TO P620. | 55.6 | 78.2 |
| P620. | IF Y-J8 = V-J4 GO TO P720. | 102.8 | 142.4 |
| P720. | IF Y-J8 = W-J9-CMP2 GO TO P820. | 102.1 | 139.7 |
| P820. | IF U-S6-CMP = V-J4 GO TO P920. | 56.8 | 80.5 |
| P920. | IF U-S6-CMP = W-J9-CMP2 GO TO Q120. | 56.6 | 78.5 |
| Q120. | IF V-J4 = W-J9-CMP2 GO TO Q220. | 101.5 | 138.3 |
| Q220. | IF A-X8 = B-X4 GO TO Q320. | 78.4 | 116.4 |
| Q320. | IF A-X8 = "YES" GO TO Q420. | 35.3 | 51.2 |
| Q420. | IF A-X8 = "YES " GO TO Q520. | 35.3 | 51.2 |
| Q520. | IF B-X4 = "YES" GO TO Q620. | 27.7 | 37.8 |
| Q620. | IF B88 GO TO Q720. | 27.7 | 37.8 |
| Q720. | IF A-X8 = "XXX" OR A-X8 = "YES" GO TO Q820. | 69.5 | 100.2 |
| Q820. | IF A-X8 = "XXX" OR "YES" GO TO Q920. | 69.5 | 100.2 |
| Q920. | IF A-X8 = "XXX" OR B-X4 = "YES" GO TO R120. | 60.8 | 84.6 |
| R120. | IF A-X8 = "XXX" OR B88 GO TO R220. | 60.8 | 84.6 |
| R220. | IF A-X8 = "YES" AND B-X4 = "YES" GO TO R320. | 61.0 | 84.8 |
| R320. | IF A-X8 = "YES" AND B88 GO TO R420. | 61.0 | 84.8 |
| R420. | IF A-X8 = "XXX" GO TO P120 ELSE IF A-X8 = "YES" GO TO R520. | 69.5 | 100.2 |
| R520. | IF B-X4 = C-X4-DISP1 GO TO R620. | 65.9 | 97.7 |
| R620. | | | |

Proc Speed 5.00/10.00
MCP Mark II.6
COBOL Mark II.6

## TYPE 17 CONSTRUCTS

| | COBOL Construct | Timings in Microseconds: 1.2 us Memory | 1.6 us Memory |
|---|---|---|---|
| P130. | IF X-3-CMP1 = 0 GO TO P230. | 4.8 | 8.0 |
| P230. | IF X-3-CMP1 EQUAL 0 GO TO P330. | 4.9 | 8.2 |
| P330. | IF X-3-CMP1 = 1 GO TO P430. | 5.3 | 8.6 |
| P430. | IF Y-6-CMP1 = 1 GO TO P530. | 5.3 | 8.6 |
| P530. | IF Z-6-CMP2 = 1 GO TO P630. | 32.1 | 42.1 |
| P630. | IF W-J6-CMP2 = 1 GO TO P730. | 49.2 | 65.8 |
| P730. | IF X-3-CMP1 = 2 GO TO P830. | 5.5 | 8.8 |
| P830. | IF X-3-CMP1 = 3 GO TO P930. | 5.5 | 8.8 |
| P930. | IF X-3-CMP1 = 10 GO TO Q130. | 5.5 | 8.8 |
| Q130. | IF X-3-CMP1 = 97 GO TO Q230. | 5.4 | 8.7 |
| Q230. | IF X-3-CMP1 = 100 GO TO Q330. | 5.4 | 8.7 |
| Q330. | IF Y-6-CMP1 = 9701 GO TO Q430. | 6.2 | 10.1 |
| Q430. | IF Y-6-CMP1 = 99999 GO TO Q530. | 7.7 | 13.2 |
| Q530. | IF Y-6-CMP1 = 100000 GO TO Q630. | 7.7 | 13.2 |
| Q630. | IF Y-6-CMP1 = 100001 GO TO Q730. | 7.7 | 13.2 |
| Q730. | IF X-3-CMP1 = 0 OR X-3-CMP1 = 1 GO TO Q830. | 9.1 | 14.6 |
| Q830. | IF X-3-CMP1 = 0 OR 1 GO TO Q930. | 8.9 | 14.4 |
| Q930. | IF X-3-CMP1 = 1 AND Y-6-CMP1 = 1 GO TO R130. | 9.5 | 15.0 |
| R130. | IF X-3-CMP1 = 0 GO TO P130 ELSE IF Y-6-CMP1 = 1 GO TO R230. | 8.9 | 14.3 |
| R230 | | | |

COBOL Construct Timings

| | |
|---|---|
| Proc Speed | 5.00/10.00 |
| MCP | Mark II.6 |
| COBOL | Mark II.6 |

**TYPE 18 CONSTRUCTS**

| COBOL Construct | NUM VALUE | Timings in Microseconds: 1.2 us Memory | 1.6 us Memory |
|---|---|---|---|
| ALTER PARG-GOTO TO PROCEED TO PARG-NOP. | | 3.2 | |
| PERFORM PARG-NOP. | | 21.1 | 35.0 |
| PERFORM PARG-NOP NUM-3-CMP1 TIMES. | 10 | 276.0 | 457.2 |
| PERFORM PARG-NOP NUM-3-CMP1 TIMES. | 100 | 2652.0 | 4417.2 |
| PERFORM PARG-NOP VARYING NUM-3-CMP1 FROM 1 BY 1 UNTIL NUM-3-CMP1 > 10. | | 311.4 | 512.6 |
| PERFORM PARG-NOP VARYING NUM-3-CMP1 FROM 1 BY 1 UNTIL NUM-3-CMP1 > 100. | | 3029.4 | 4994.6 |
| P140. GO TO P141, P142, P143, P144 DEPENDING ON NUM-3-CMP1. | 0 | 15.8 | |
| P140. GO TO P141, P142, P143, P144 DEPENDING ON NUM-3-CMP1. | 1 | 21.6 | 34.4 |
| P140. GO TO P141, P142, P143, P144 DEPENDING ON NUM-3-CMP1. | 2 | 21.6 | 34.4 |
| P140. GO TO P141, P142, P143, P144 DEPENDING ON NUM-3-CMP1. | 3 | 21.6 | |
| P140. GO TO P141, P142, P143, P144 DEPENDING ON NUM-3-CMP1. | 4 | 21.8 | 34.6 |
| P140. GO TO P141, P142, P143, P144 DEPENDING ON NUM-3-CMP1. | 5 | 16.7 | 27.2 |
| GO TO<br>NOTE: This GO TO has been altered | | 6.5 | 11.3 |

Proc Speed : 5.00/10.00
MCP : Mark II.6
COBOL : Mark II.6

## TYPE 19 CONSTRUCTS

| COBOL Construct | Timings in Microseconds: 1.2 us Memory | 1.6 us Memory |
|---|---|---|
| MOVE 5 TO SUB-2-CMP1. | 3.2 | 5.0 |
| MOVE X-X5 TO SUB-X5-OC-10 (SUB-2-CMP1). | 40.6 | 57.0 |
| SET INX1 TO INX2. | 3.8 | 6.8 |
| SET INX2 TO INX1. | 3.8 | 6.8 |
| SET INX2 TO 5. | 3.2 | 5.0 |
| MOVE X-X5 TO IDX-X5-OC-10(INX2). | 35.2 | 49.6 |
| SET INX2 UP BY 1. | 5.1 | 8.6 |
| SET INX2 TO 1. | 2.8 | 4.5 |
| SET INX2 TO 1. SEARCH IDX-X5-OC-10; WHEN IDX-X5-OC-10(INX2) = "H*I*T" NEXT SENTENCE. NOTE: IDX-X5-OC-10(1) = "H*I*T". | 30.8 | 44.8 |
| SET INX2 TO 1. SEARCH IDX-X5-OC-10; WHEN IDX-X5-OC-10(INX2) = "H*I*T" NEXT SENTENCE. NOTE: IDX-X5-OC-10(5) = "H*I*T". | 173.0 | 248.6 |
| SET INX2 TO 1. SEARCH IDX-X5-OC-10; WHEN IDX-X5-OC-10(INX2) = "H*I*T" NEXT SENTENCE. NOTE: IDX-X5-OC-10(10) = "H*I*T". | 348.2 | 498.9 |
| SET INX2 TO 1. SEARCH IDX-X5-OC-10; WHEN IDX-X5-OC-10(INX2) = "H*I*T" NEXT SENTENCE. NOTE: IDX-X5-OC-10(INX2) NOT = "H*I*T". | 354.4 | 508.2 |

COBOL Construct Timings

Proc Speed 5.00/10.00
MCP Mark II.6
COBOL Mark II.6

## TYPE 20 CONSTRUCTS

| COBOL Construct | XX VALUE | Timings in Microseconds 1.2 us Memory | 1.6 us Memory |
|---|---|---|---|
| EXAMINE WS-OP-15 TALLYING UNTIL FIRST "X". | 1st POS="X' | 33.4 | 49.1 |
| EXAMINE WS-OP-15 TALLYING UNTIL FIRST "X". | 2nd POS="X" | 33.8 | 49.1 |
| EXAMINE WS-OP-15 TALLYING UNTIL FIRST "X". | 5th POS="X" | 35.0 | 49.7 |
| EXAMINE WS-OP-15 TALLYING UNTIL FIRST "X". | 10th POS="X' | 37.4 | 52.1 |
| EXAMINE WS-OP-15 TALLYING UNTIL FIRST "X". | 25th POS="X" | 44.6 | 60.3 |
| EXAMINE WS-OP-15 TALLYING ALL "X". | NO X" | 42.9 | 57.8 |
| EXAMINE WS-OP-15 TALLYING ALL "X". | 1 "X" | 82.5 | 119.1 |
| EXAMINE WS-OP-15 TALLYING ALL "X". | 5 "X" | 241.4 | 365.9 |
| EXAMINE WS-OP-15 TALLYING ALL "X". | 10 "X" | 439.6 | 673.6 |
| EXAMINE WS-OP-15 TALLYING ALL "X". | 25 X" | 1033.8 | 1594.6 |
| EXAMINE WS-OP-15 REPLACING ALL "X" BY "Y". | NO "X" | 40.0 | 53.1 |
| EXAMINE WS-OP-15 REPLACING ALL "X" BY "Y". | 1 "X" | 82.6 | 118.7 |
| EXAMINE WS-OP-15 REPLACING ALL "X" BY "Y". | 5 "X" | 253.5 | 382.7 |
| EXAMINE WS-OP-15 REPLACING ALL "X" BY "Y". | 10 X" | 465.7 | 709.9 |
| EXAMINE WS-OP-15 REPLACING ALL "X" BY "Y". | 25 X' | 1102.1 | 1689.3 |

# Section 4
# FORTRAN CONSTRUCT TIMINGS

The following measurements were performed on various FORTRAN constructs using the Burroughs Mark II.6 release FORTRAN Compiler. Both 1.2 $\mu$s and 1.6 $\mu$s memories were utilized, and the timings of a particular construct for each memory are shown side-by-side on the following pages. The asterisk(s) between the timing results refer to notes which are located on page 4-6.

Ranges in times for a given construct are simply differences due to the operand values used in the execution of the construct.

Certain variables are used exclusively for each general type of construct. These are:

| | |
|---|---|
| Single Precision Real | X, Y, Z |
| Integer | L, M, N |
| Double Precision Real | D, E, F |
| Complex | A, B, C |

The permissible combinations of operand values (limited by overflow, underflow, division by zero, etc.) with which the timings were taken are given in the following table. Constructs which contain both variables were timed using all possible combinations of the operand values given. Thus, the construct "Z = X + Y" was timed using 25 different pairs of operand values for X and Y. Constructs containing only one variable were timed using those operand values beneath that variable in the table. Hence, the construct "Z = ABS (Y)" was timed using 5 different operand values for Y.

FORTRAN Construct Timings

### SINGLE PRECISION REAL

| X | Y |
|---|---|
| 0.0000000E+00 | -.1740000E-32 |
| 0.1000000E+01 | 0.0000000E+00 |
| 0.4000000E+01 | 0.1000000E+01 |
| 0.4631527E+35 | 0.4000000E+01 |
| 0.6517000E-33 | 0.8623355E+31 |

### DOUBLE PRECISION REAL

| E | F |
|---|---|
| 0.0000D+00000 | -.1740D-00032 |
| 0.1000D+00001 | 0.1000D+00001 |
| 0.4000D+00001 | 0.3778D+04095 |
| 0.4632D+00035 | 0.4000D+00001 |
| 0.6517D-00033 | 0.8623D+00031 |

### INTEGER

| M | N |
|---|---|
| 0 | 0 |
| 1 | 1 |
| 4 | 4 |
| 6517 | -1740 |
| 4631527 | 8623355 |

### COMPLEX

| A | B |
|---|---|
| (0.00000E+00,0.00000E+00) | (-.46315E+35,-.65170E-13) |
| (0.10000E+01,0.10000E+01) | (-.86234E+17,0.17400E-32) |
| (0.15000E+01,0.15000E+01) | (0.10000E+01,0.10000E+01) |
| (0.65170E+13,0.65170E-13) | (0.15000E+01,0.15000E+01) |
| (0.86234E+17,0.86234E-17) | (0.40000E+01,0.40000E+01) |

Proc Speed 5.00/10.00
MCP Mark II.6
FORTRAN Mark II.6

## SINGLE PRECISION REAL CONSTRUCTS

| FORTRAN Construct | Range of Times in Microseconds 1.2 µs Memory | | 1.6 µs Memory |
|---|---|---|---|
| IF (Y) 10000,10000,20000<br>10000 CONTINUE<br>20000 CONTINUE | 4.0 - 5.1 | * | 6.3 - 8.1 |
| W1(N2) = Y | 8.3 | ** | 13.9 |
| W1(N1,N2) = Y | 21.3 | ** | 30.1 |
| Z = +.6517E-33 | 4.6 | | 7.6 |
| Z = -.1740E-32 | 5.0 | | 8.0 |
| Z = ABS(Y) | 5.7 | | 8.9 |
| Z = ALOG(X) | 128.5 - 173.5 | *** | 211.3 - 262.5 |
| Z = AMAX1(X,Y) | 19.2 - 20.7 | | 31.1 - 33.2 |
| Z = AMIN1(X,Y) | 19.2 - 20.7 | | 31.1 - 33.2 |
| Z = EXP(X) | 125.3 - 165.7 | *** | 210.3 - 248.7 |
| Z = N2 | 4.0 | | 6.8 |
| Z = SIN(Y) | 86.5 - 192.1 | *** | 145.5 - 291.1 |
| Z = SQRT(X) | 58.5 - 170.1 | *** | 97.1 - 246.1 |
| Z = X + Y | 5.6 - 7.0 | | 10.3 - 11.7 |
| Z = X * Y | 5.6 - 8.4 | | 10.3 - 13.1 |
| Z = X - Y | 5.6 - 7.2 | | 10.3 - 11.9 |
| Z = X / Y | 5.9 - 13.7 | | 10.3 - 18.1 |
| Z = 0.0 | 2.9 | | 4.5 |
| Z = 1.0 | 2.9 | | 4.5 |
| Z = 97.01 | 4.6 | | 7.6 |

Proc Speed 5.00/10.00
MCP Mark II.6
FORTRAN Mark II.6

**INTEGER CONSTRUCTS**

| | FORTRAN Construct | Range of Times in Microseconds 1.2 μs Memory | | 1.6 μs Memory |
|---|---|---|---|---|
| | CALL CONT | 33.5 | **** | 54.0 |
| | DO 10001 I=1,M | (9*M)+2.9 | | (15*M)+3.8 |
| 10001 | CONTINUE | | | |
| | GO TO 10002 | 1.6 | | |
| 10002 | CONTINUE | | | |
| | IF(N)10003,10003,20003 | 4.1 - 5.1 | | 6.3 - 8.1 |
| 10003 | CONTINUE | | | |
| 20003 | CONTINUE | | | |
| | L = IABS(N) | 5.7 | | |
| | L = ISIGN(M,N) | 7.6 | | 12.5 |
| | L = M + N | 6.1 - 6.5 | | 11.0 - 11.4 |
| | L = M * N | 6.1 - 7.9 | | 11.0 - 12.8 |
| | L = M - N | 6.5 - 6.9 | | 11.0 - 11.4 |
| | L = M / N | 6.5 - 11.9 | | 11.0 - 16.4 |
| | L = MAX0(M,N) | 19.2 - 20.0 | | 31.2 - 32.0 |
| | L = MIN0(M,N) | 18.8 - 19.8 | | 31.2 - 32.2 |
| | L = N**M | 80.7 - 198.6 | | 135.1 - 318.7 |
| | L = 0 | 2.9 | | |
| | L = 1 | 2.9 | | 4.5 |
| | L = 100 | 3.2 | | 5.1 |
| | L = 2 | 3.2 | | 5.1 |
| | L = 97 | 3.2 | | 5.1 |
| | L1(N2) = M | 7.9 | ** | 13.9 |
| | L2(N1,N2) = M | 21.3 | | 30.1 |

Proc Speed : 5.00/10.00
MCP : Mark II.6
FORTRAN : Mark II.6

## DOUBLE PRECISION REAL CONSTRUCTS

| FORTRAN Construct | Range of Times in Microseconds: 1.2 μs Memory | | 1.6 μs Memory |
|---|---|---|---|
| D = DABS(F) | 8.3 | | 13.4 |
| D = DEXP(E) | 375.6 - 570.6 | *** | 632.6 - 829.6 |
| D = DLOG(F) | 405.2 - 412.0 | *** | 653.4 - 660.2 |
| D = DMAX1(E,F) | 30.2 - 32.1 | | 51.0 - 53.5 |
| D = DMIN1(E,F) | 30.2 - 32.1 | | 51.0 - 53.5 |
| D = DSIGN(E,F) | 11.4 | | 19.2 |
| D = DSIN(E) | 268.8 - 442.0 | *** | 458.8 - 672.0 |
| D = DSQRT(E) | 78.2 - 256.2 | *** | 129.4 - 358.6 |
| D = E + F | 10.3 - 12.7 | | 17.6 - 20.0 |
| D = E * F | 9.7 - 18.7 | | 17.0 - 26.0 |
| D = E - F | 10.3 - 12.9 | | 17.6 - 20.2 |
| D = E / F | 9.7 - 57.9 | | 17.0 - 67.2 |
| D = 1D0 | 5.0 | | 7.8 |
| D = 511D511 | 9.6 | | 15.6 |
| D = 8D0 | 5.2 | | 8.2 |

## COMPLEX CONSTRUCTS

| FORTRAN Construct | Range of Times in Microseconds: 1.2 μs Memory | 1.6 μs Memory |
|---|---|---|
| C = A + B | 27.5 - 30.7 | 45.1 - 48.1 |
| C = A * B | 130.3 - 145.3 | 222.5 - 235.3 |
| C = A - B | 28.3 - 31.1 | 45.8 - 48.6 |
| C = A / B | 159.6 - 190.6 | 271.8 - 300.6 |

## NOTES

* The longer timing if the operand is positive (greater than zero)

** After the array is present in memory.

*** These constructs call upon intrinsic routines of which each must be linked from the D1 stack prior to its first execution within a program. The first timing, which includes intrinsic linkage, will vary between 2 and 160 milliseconds, depending upon the presence or absence of certain MCP segments in main memory. The timings shown for these constructs were observed after the corresponding intrinsic was called once and linkage was thus established.

**** CONT is a subroutine which consists of the following code:

```
      SUBROUTINE CONT
   10 CONTINUE
      RETURN
      END
```

Burroughs Corporation Publications Remarks Form

# B 6700 TIMINGS FOR SELECTED LANGUAGE CONSTRUCTS

Form No. 5000854, February 1975

Comments

From: Date

Name

Title

Company

Address

Fold Along Dashed Line

Fold, Staple, And Mail

Postage Will Be Paid by Addressee

No Postage Stamp Necessary If Mailed in the United States

**BUSINESS REPLY MAIL**
First Class Permit No. 1009; El Monte, CA 91731


**Burroughs Corporation**

P. O. Box 142
El Monte, CA 91734

attn: Publications Department
Technical Information Organization


Fold, Staple, And Mail