

## 1 SUMMARY

Provides the Fortran programmer with the CPU time.

**ATTRIBUTES** — **Version:** 1.0.0. **Types:** Real (single, double) **Calls:** CPU\_TIME (ISO Fortran 95), ETIME (generic UNIX), MCLOCK (AIX), LIB\$GETJPI (Vax), CPUTIME (IBM mainframe), SECOND (Cray), CLOCK@ (Salford with Intel). **Original date:** March 2005. **Licence:** A third-party licence for this package is available without charge.

## 2 HOW TO USE THE PACKAGE

### 2.1 The Argument List and Calling Sequence

*The single precision version*

```
T = ZA12A(DUMMY)
```

*The double precision version*

```
T = ZA12AD(DUMMY)
```

T is a REAL (DOUBLE PRECISION in the D version) variable that will be set to the current CPU clock reading in seconds.

DUMMY is a REAL (DOUBLE PRECISION in the D version) dummy argument that is ignored by ZA12A/AD.

## 3 GENERAL INFORMATION

**Workspace:** none.

**Use of common:** none.

**Other routines called directly:** calls the system cpu clock

**Input/output:** none.

## 4 METHOD

The ZA12A/AD functions call the system cpu clock function. The clock reading is normally of no interest on its own and ZA12 is principally used to measure intervals of cpu time by differencing two readings. The standard version calls CPU\_TIME (the ISO standard Fortran 95 timer) but there are alternative versions, e.g. ETIME (generic UNIX), MCLOCK (AIX), LIB\$GETJPI (Vax), CPUTIME (IBM mainframe), SECOND (Cray) and CLOCK (Salford with Intel).

## 5 EXAMPLE OF USE

Suppose it is required to time the execution of a calculation performed using a pair of nested do loops. Use ZA12 to take the clock reading TIME1 before the loops are entered and another reading TIME2 after they have completed then differencing the two readings gives the time taken in seconds, for example,

```
DOUBLE PRECISION TIME1, TIME2, ZA12AD, DUMMY, TIME, A
INTEGER I
EXTERNAL ZA12AD
TIME1=ZA12AD(DUMMY)
A = 1.0D0
DO 20 J = 1,1000
```

```
      DO 10 I = 1, 100000
        A = A * 1.000001D0
10    CONTINUE
        A = 1.0D0 + 1.0D0/A
20    CONTINUE
      TIME2=ZA12AD(DUMMY)
      TIME=TIME2-TIME1
      WRITE(*,*)' TIME TAKEN IS ',TIME
      STOP
      END
```

which produces a result similar to

```
TIME TAKEN IS      0.9300000000000000
```