

## SIMPLE PROGRAM MEASUREMENT SYSTEM

The SIMPLE program measurement system allows evaluation of the time spent in the various functions of a program. In order to use the measurement facility, the program must first be recompiled globally in measurement mode.

Example:

```
□SCOM
$SP PROG
$MM 1 (or MM -1)
$SI
$CP
```

Any number of the user functions of a program may be specified to be measured. This may be done by appending MEASURE to the end of the FUNCTION statement for a function to be measured and recompiled. Alternatively, a user may specify that all user functions be measured by recompiling the programs globally after doing an MM -1 command.

Example:

```
FUNCTION F(X,Y,) RETURNING VECOTRS MEASURE
```

Three values are maintained about each measured function:

- 1) the accumulated elapsed real time spent in the function in microseconds.  
This time is measured from the time the function is called until it returns or until some other measured function (perhaps a recursive call on itself) is called. When this function returns real time is once again accumulated. That is, time is accumulated for the topmost measured function on the call stack.
- 2) the CPU compute time spent in the function in milliseconds. It is measured in the same way as real time.

3) the number of times the measured function is called.

To begin measuring a program, the program must execute a call on runtime function INITMEAS . It takes a single Boolean valued argument which specifies if CPU elapsed time measurements are to be taken.

Typically, INITMEAS will be called from the beginning of the program's driver function.

Example:

```
INITMEAS(TRUE)
```

A program may suspend measurement of functions by calling runtime function STOPMEAS and may resume measurement with STARTMEAS .

Example:

```
STOPMEAS()  
STARTMEAS()
```

The accumulated statistics may be printed out by calling runtime function PRINTMEAS .

Example:

```
[ ] SED  
:SP PROG  
:RS  
:DO PRINTMEAS
```