January 1967

Brief 67-10001

NASA TECH BRIEF

Program Computes Single-Point Failures in Critical System Designs

The problem:

To devise a method for analyzing the designs of critical systems that will either prove the design is free of single-point failures or detect each member of the population of single-point failures inherent in a system design. A single-point failure is defined as that component failure which can produce an unwanted system output.

The solution:

A computer program which, given the logical model of a system and the model's response in an unfailed mode, will compute the totality of failure effects on a number of selected elements in the system. The computations are performed element by element, system state by system state, until the history of the system has been exhausted. All single-point failures are printed out.

How it's done:

This program detects the single-point failures of a logical system that is representable as variable state constraints within the set of Boolean equations that model the system. A Boolean simulation of the system and the computation of its history in an unfailed operating mode must have been previously computed and recorded on tape. These data are essential input to the program and are generated by a Boolean simulation program which provides the history as a sequence of state vectors.

In order for the recorded history to be compatible with the single-point failure program, the elements of the logical model must be ordered for Boolean simulation such that the elements to be failed are dense on the "tail", or right end, of the state vector. This ordering is accomplished by an equation translator program. This program takes, as input, an ordered list of names of the logical elements together with the Boolean model in terms of the names, and produces, as output, the set of equations in terms of element numbers. This output and the taped history are compatible for processing by the single-point failure program.

The single-point failure program will test a maximum of 24 elements in the system. The unfailed history of the logical system represents "nominal data" to which the computed state vectors are compared in determining the consequences of component failure.

The computer fails an element by changing and freezing its logical state, computing its effect on the rest of the sample elements (checking to see whether the failed element causes any of the other sample elements to change state), and then printing out the differences between the computed state vector and the state vector of the unfailed system. After each computation, the state of the system is reset to the unfailed state so that the next failure computation can be made. This process proceeds, element by element, system state by system state, until the history of the system has been exhausted.

Notes:

1. This program should find application in the checkout of redundant circuits and digital systems.
2. The system size is restricted to 999 elements.
3. If more than 24 sampling elements must be monitored for state alteration, the sampling elements may be grouped in lots of less than or equal to 24 and the program executed for each group.
4. Inquiries concerning this innovation may be directed to:

   COSMIC
   Computer Center
   University of Georgia
   Athens, Georgia 30601
   Reference: B67-10001

(continued overleaf)

Patent status:

No patent action is contemplated by NASA.

Source: W. R. Brown of
North American Aviation, Inc.
under contract to
Manned Spacecraft Center
(MSC-603)