Real-Time State Machine Implementation Programming Techniques

Randy M. Dumse
New Micros, Inc.
Grand Prairie, TX

ABSTRACT

Experienced system designers readily admit that state diagrams are more useful in describing real-time automina than flowcharts. This is particularly true during the system definitions, when the actions of the machine are described in terms of the inputs and outputs and time conditioned responses. These same designers would be hard pressed to name even one source describing any method for generate programs from state diagrams. State diagrams have historically been the tools of the hardware designer, which may explain why published materials on conversion methods from state diagrams to computer programs (flow chart, pseudo-code or high level language routine) are not widely available. This paper will attempt to identify why high level language programmers have lined up behind the flow chart (or pseudo-code equivalent) and left state diagrams to hardware engineers and a few stubborn assembly language programmers. It will also explain at least one version of the rules used to design using state diagrams, and how to program from these drawings.

*This paper has been submitted for publication to the **Journal of Forth Application and Research.**