struction field provided for addresses. But this is not our primary goal. A hardwired VLSI implementation would be faster, occupy less real estate, and would perhaps be cheaper in quantity. This decoder was only presented as an example of a single, easily understood BD application. As stated previously, this BD decoder algorithm is a typical routine which might commonly appear as part of, say, a larger control program. There is enough program memory to implement a reasonably elaborate procedure without resorting to any extraneous hardware.

When we started, the issues seemed to be quite clear:

1. how to overcome the inconvenience of using 8-bit microprocessors to implement sequence controllers whose inputs and outputs cannot be naturally associated to form numbers,
2. how to preserve the flexibility of microprocessor-like instruction sequences, and
3. how to compensate, at least partially, for the inherent slowness of examining binary inputs one by one and, similarly, of generating the binary outputs.

We still maintain that the programmable BD machine with parallel outputs is a good compromise. It can execute any programmable procedure. It can be made to look like any m-input, n-output device. For simple procedures, it's much faster than a conventional microprocessor. At no time did we claim that it was faster than a well-tailored, multibit finite state machine. We do believe, however, that it is much easier to program. Apparently our implicit assumption that the designer had a choice only between boolean and BD design techniques was overly simplistic. Maybe the situation is better illustrated by a spectrum with microprocessors at one end and hardwired logic at the other, while BD and MBFSA are somewhere in the middle.

This is getting to be a bit like the five blind, wise men of Hindustan who went to see the elephant. Each gropingly identified a specific attribute of the beast and loudly proclaimed his limited viewpoint to be complete and uniquely correct!

It's somewhat unfortunate that we chose to demonstrate BD effectiveness with a single example—the biphase encoder—but that's the only application device which was working when we wrote the article. We led with our collective chins. Although it isn't fair to mention unpublished and largely untried BD application, a short "notebook" of generally useful BD programs and their trees has been prepared. Included are things such as adders, comparators, first- and second-differencers and short, straight and curved trajectory generators. We feel that such macros are the beginnings of a high-level BD language. Prospects for similar evolution of MBFSA software may exist, but we haven't studied them nor do we know anyone who has.

P. J. Zsombor-Murray
L. J. Vroomen
R. Hudson
DATAC Computer Lab
McGill University
Montreal, Que. H3A 2K6

Reader Interest Survey

Indicate your interest in this department by circling the appropriate number on the Reader Interest Card.

High 173  Medium 174  Low 175