To the Editor:  

Your two recent letters on “von Neumann versus microcoding” were rather puzzling.1,2  
My background in the subject is marginal: teaching EE courses on bit-slice (grad) and microcoded hardware (undergrad), assembly-language programming on several machines, interfacing, nine years in the ACM microprogramming SIG, and so on. However, several facets of the argument initially confused me.  
Some examples may serve to illustrate the problem.  
  
First, the IBM 360 was cited as a “classical von Neumann computer” by the “von Neumann cannot be microcoded” authors.2 However, a microcoded APL implementation on a 360 was mentioned in the 1982 second edition of their first reference;3 moreover, only the 360 Model 75 appears to have been built with hardwired control logic, while the other models (20, 25, 30, 40, 50, 60, 85) were microcoded.4  
The Apple seems to be a von Neumann machine if you apply a Turing test (what you can observe from a terminal). If you open the box or read the schematic, it is still von Neumann. However, the silicon of the 6502 CPU chip appears microcoded. Most of the microprocessor chips of this generation are internally microprogrammed.5  
The Commodore 64 also uses the 6502. The disk drive has its own CPU, so technically the 64 is not von Neumann; however, a Turing test cannot distinguish this second CPU from hardwired logic.  
The Data General Nova has different instruction formats, all on a 16-bit word, and most of which may be viewed as having a 1-byte opcode followed by an address.6 However, the arithmetic instruction specifies the two registers and the operation in the first byte, and “reconfiguration” options (shift, skip conditions, and “discard result” option) in the second byte. If “one of the most popular [computers] in the world” were not von Neumann, many more people would have discussed it in the literature.7  
My limited experience in hardware repair of Novas suggests that the earlier models use discrete logic for implementation, rather than microcode. The MicroNova chip is probably microcoded.  
The Data General Eclipse, with substantially the same instruction set as the Nova, is microcoded. An optional writable control store lets you write your own microcode (e.g., a floating-point, 256-long complex FFT in one instruction).  
The problem here and in the two earlier letters seems to be one of definition. A microprogram is defined by the ANSI standard8 and ISO as “a sequence of elementary instructions that correspond to a computer operation, that is maintained in special storage, and whose execution is initiated by the introduction of a computer instruction into an instruction register of a computer.” A von Neumann machine has been described9 as having four characteristics:  
(1) data and instructions share the same memory,  
(2) memory is one dimensional (linear),  
(3) instructions and data are indistinguishable, allowing self-modifying code, and  
(4) data has no inherent meaning.  
The microprogrammed machine by definition has a separate memory for microcode instructions, and therefore cannot be “von Neumann” by ANSI definition.8 However, a Turing test usually cannot reveal the presence of fixed microprogramming.  
To reduce future confusion, I suggest that future writers distinguish more clearly between microprogramming in instruction set (e.g., Nova), architecture (by Turing test), and implementation (the invisible hardware).  
All company and machine names are trademarks.  

James D. Marr
University of Alabama in Huntsville

References

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

High 183 Medium 184 Low 185

June 1986