November 21, 2022

A History of Early Microcontrollers, Part 2: The Texas instruments TMS1000

by Steven Leibson

As with many first-of-a-kind devices, the Texas Instruments (TI) TMS0100 calculator chip family was a narrowly defi microcontroller, mostly good for making calculators. However, the first chip in the TMS0100 family, originally called t TM1802NC and later renamed the TMS0102, incorporated everything a microcontroller requires to be a microcontrol CPU, RAM, ROM, and I/O. Granted, it was a specialized microcontroller. Its I/O was application-specific and designed attached to a matrix keyboard and a seven-segment display. Nevertheless, the TMS1802NC was a microcontroller.

Originally conceived by TI's Gary Boone and Daniel Baudouin, and then implemented by Boone and Michael Cochran, TI introduced the TMS0100 fa on September 20, 1972, and these chips quickly took over the calculator market. TI suddenly had all-new worlds to conquer. The company quickly realized that it could serve multiple markets with the same programmable silicon if that silicon could be engineered to be sufficiently general. TI ap what it had learned from the family of devices it built from the original programmable calculator chip to produce its first line of general-purpose microcontrollers, the TMS1000 family, which it announced in 1974.

The TMS1000 microcontroller family has a few similarities with the TMS0100 programmable calculator family and many, many differences. Both devices have 4-bit CPUs and Harvard architectures, which have separate address spaces for RAM and ROM. Harvard architectures are common in microcontroller designs because they simplify the designs of the microcontroller's RAM, ROM, address decoders, and data buses. However, in my opinion, Harvard architectures complicate the lives of programmers, who must keep track of two different address spaces and must often devise v to move data from ROM to RAM. (Fortunately, there's no point in moving data from RAM to ROM. The data won't finish that journey. You can't successfully write to a masked ROM.)

Instead of the TMS0102's 3250-bit ROM, organized as 320 11-bit words, the first TMS1000 microcontroller had a 1-Kbyte ROM, organized as 1024 words. So, the TMS0100 and TMS1000 families started with incompatible 11- and 8-bit instruction sets respectively. Similarly, instead of the TMS0 182-bit serial RAM that held three 13-digit numbers in BCD (binary-coded decimal) format and 13 binary flags, the TMS1000 microcontroller had a bit RAM organized as 64 4-bit words.

According to the TMS1000 documentation, the 64 4-bit words stored in RAM "are conveniently grouped into four 16-word files addressed by a two-register." In my experience, when programming a similar microcontroller architecture, to be discussed later in this series, there's nothing convenient about having your tiny RAM address space further segmented into 16-word chunks unless you're programming a 16-entry circular buffer. Like many design compromises that microcontroller design teams made to cram entire CPUs onto early semiconductor die along with RAM, ROM, and I/O cir I'd wager that this specific design choice made the TMS1000 microcontroller's hardware design simpler and smaller, with the usual sacrifice of programmer convenience in exchange for hardware design expediency.

Unlike the TMS0100 calculator chip family with its dedicated I/O design for scanning a keyboard and driving a multi-digit, seven-segment display, th TMS1000 microcontrollers had general-purpose I/O pins, at least nominally. Four input pins (K1, K2, K4, and K8) could be read as a group with one instruction. Output pins were more complex. The original TMS1000 microcontroller had eleven "R" outputs (R0 through R10) and eight "O" outputs through O7). The "R" outputs were individually set and cleared. The "O" outputs were controlled by a mask-programmed PLA and driven by a 5-bit la Four bits in the latch could be set with an instruction that moved data directly from the TMS1000's accumulator to the latch. The fifth output bit car from the ALU's status latch. Using a PLA to expand the 5-bit output latch into eight output pins recalls the TMS1000's heritage as the descendant o calculator chip, which was designed to drive seven-segment displays.

Among his many achievements, Adam Osborne chronicled the early microprocessors and microcontrollers in his book titled *An Introduction to Microcomputers*, published in 1978. In his description of the TMS1000, Osborne seemed to focus more on the microcontroller's limitations:

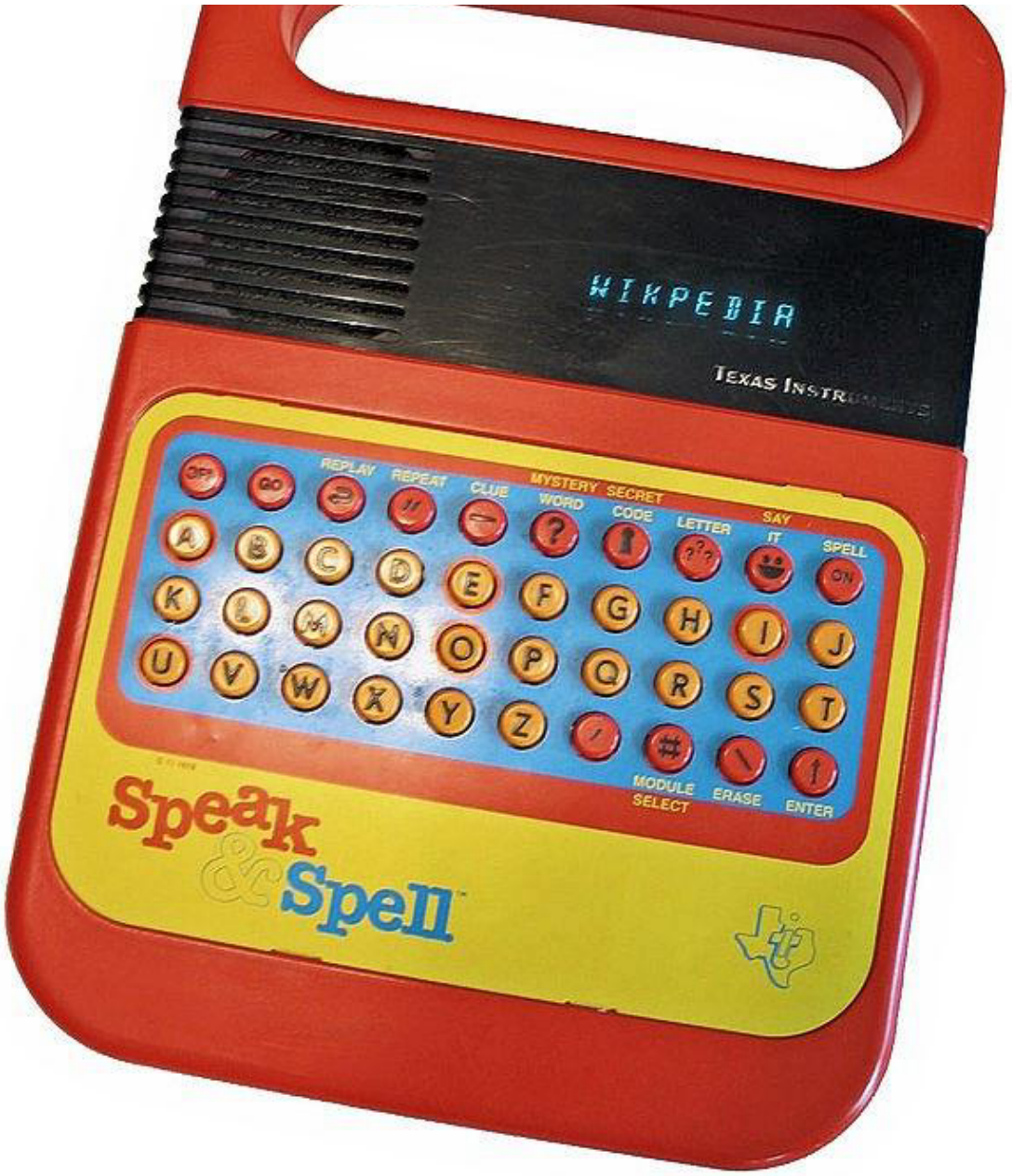
"The fact that the TMS1000 series microcomputers are single-chip devices has a number of secondary, non-obvious implications. Most important there are no such things as support devices. The 1024 or 2048 bytes of ROM [the TMS1200 microcontroller has 2Kbytes of ROM] represent the exa amount of program memory that will be present; there can be neither more nor less. Similarly, the 64 or 128 nibbles of RAM [a nibble is a 4-bit word cannot be expanded. Direct memory access logic is not present – and its presence would make very little sense anyway; with the small total RAM a ROM available, there simply is not the opportunity to transfer blocks of data long enough to warrant bypassing the CPU.

"Interrupts, similarly, would be of marginal value in a TMS1000 microcomputer. Given the small amount of program memory available and the very cost of the package, it would be hard to justify the complexities of interrupt logic simply to have the microcomputer perform more than one task."

To me, Osborne's words suggest that many people (possibly including Osborne) did not have a clear picture of the difference between microcontro and microprocessors back in 1978 when he published that book, four years after TI first announced the TMS1000 family. However, many people di understand the difference, because TI reportedly was cranking out tens of millions of TMS1000 parts every year by 1979, and they sold the TMS10 $2 or $3 in high volumes. The low unit cost for the TMS1000 was possible, in part, because TI packaged the device in an inexpensive, 28-pin, plasti

TI offered the low-cost TMS1000 microcontroller in an inexpensive, 28-pin, plastic DIP.
Image credit: Antonio Martí Campoy, Wikimedia Commons

TI ate its own dog food by using members of the TMS1000 microcontroller family in some of its own consumer products, including the legendary T Speak & Spell game and the SR-16 “Electronic Slide Rule” calculator.



TI used its own TMS1000 microcontroller to create the TI Speak & Spell. Image credit: FozzTexx, Wikimedia Commons

Inventor, game designer, and “father of the home video game console” Ralph Baer realized he could create affordable electronic games with microcontrollers and incorporated a TMS1000 into one of the most successful handheld electronic games ever made, Milton Bradley’s Simon, introduced in 1978. These days, everyone plays handheld games on their mobile phones, but back then, these games required dedicated hardware.

Milton Bradley's handheld electronic game Simon was an early product that incorporated TI's TMS1000 microcontroller. Image credit: Shritwod, Wikimedia Commons

Parker Brothers released the handheld Merlin electronic game based on the TMS1000 microcontroller in 1978, and a year later, Milton Bradley usec TMS1000 microcontroller as the programmable brains for its Big Trak, a futuristic, 6-wheeled, tank-like vehicle that could be pre-programmed to fol specific path using a membrane keyboard embedded on the back of the toy. The Big Trak could execute a 16-command sequence punched into its keypad, which seems closely related to the turtle graphics of the Logo programming language developed in 1967 by Wally Feurzeig, Seymour Pape Cynthia Solomon at a Cambridge, Massachusetts research firm named Bolt, Beranek and Newman (BBN).

Milton Bradley used a TMS1000 microcontroller in the Big Trak toy vehicle. Image credit: Martin Ling, Wikimedia Commons

Mattel introduced a tremendously successful electronic Football game in 1977. It was based on a Rockwell calculator chip, but companies around world cloned this game, and a game manufacturer in Hong Kong named Conic appears to have used a TMS1000 microcontroller in its clone instea calculator chip. The open-source game emulator MAME (Multiple Arcade Machine Emulator) can still run the TMS1000 ROM code for Conic's Footl emulation, of course.

In his book "State of the Art," author Stan Augarten notes that the TMS1000 was used in calculators, toys, games, appliances, burglar alarms, photocopying machines, and jukeboxes. Augarten concludes his TMS1000 description by writing, "As much as any IC, the TMS 1000 has helped m the power of modern electronics available to everyone."

I suspect that myriad undocumented applications for the TMS1000 family also exist. That's quite a success story and legacy for the second earlies microcontroller family and a testament to the truly universal nature of the basic single-chip microcontroller concept. After TI introduced the TMS10 1974, new microcontroller introductions from other semiconductor vendors would come at a fast and furious pace, as will be discussed by future articles in this history series.

References

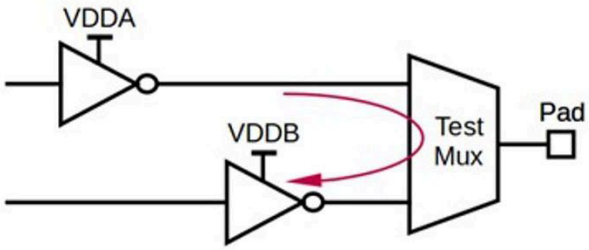
An Introduction to Microcomputers Volume 2: Some Real Microprocessors, Adam Osborne with Gerry Kane, 1978.

State of the Art: A Photographic History of the Integrated Circuit, Stan Augarten, 1983.

One thought on “A History of Early Microcontrollers, Part 2: The Texas instruments TMS1000”

Pingback: The TMS1000 Powered Electronic Boardgames – Troy Press

featured paper



Quickly and accurately identify inter-domain leakage issues in IC designs

Sponsored by Siemens Digital Industries Software

Power domain leakage is a major IC reliability issue, often missed by traditional tools. This white paper describes challenges of identifying leakage, types of false results, and presents Siemens EDA's Insight Analyzer. The tool proactively finds true leakage paths, filters out false positives, and helps circuit designers quickly fix risks—enabling more robust, reliable chip designs. With detailed, context-aware analysis, designers save time and improve silicon quality.

Click to read more

featured chalk talk