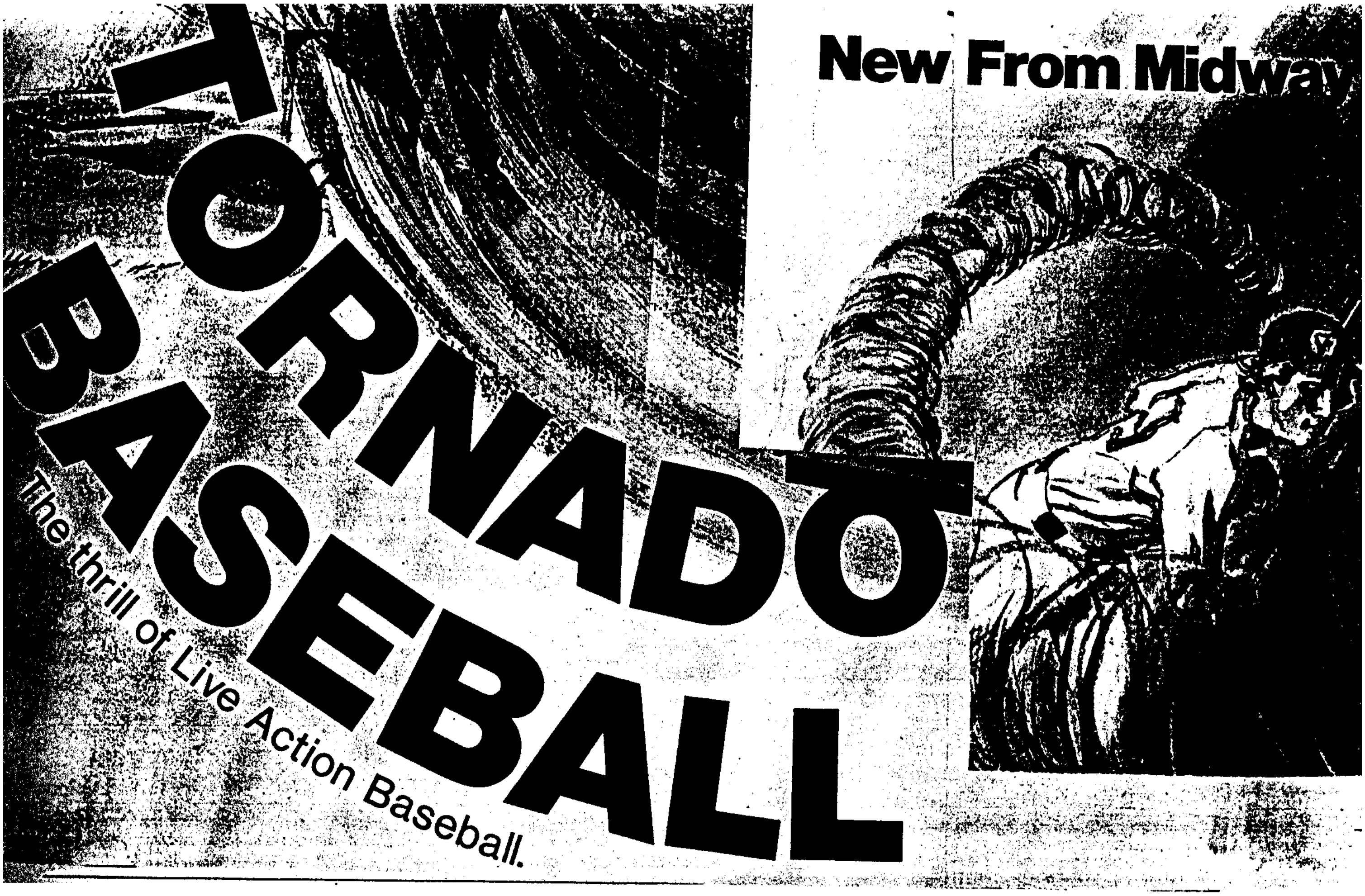


More complex video games keep player interest high

by T. George Blahuta
Midway Manufacturing Co., Franklin Park, Ill.

The microprocessor, busy bee of the electronics world, is now engaging in fun and games as well as drudgery. Although the earliest video games in arcades were built with transistor-transistor logic, they were so simple they could soon become boring to many players. And the manufacturers, lacking the capability to design their own TTL logic, had to buy the designs and sometimes even the logic packages as well.

Designing games around the microprocessor has increased their complexity by two to five times, and allows the manufacturer to design his programs right in his own shop. What's more, changing to the microprocessor has cut development time by 75%.

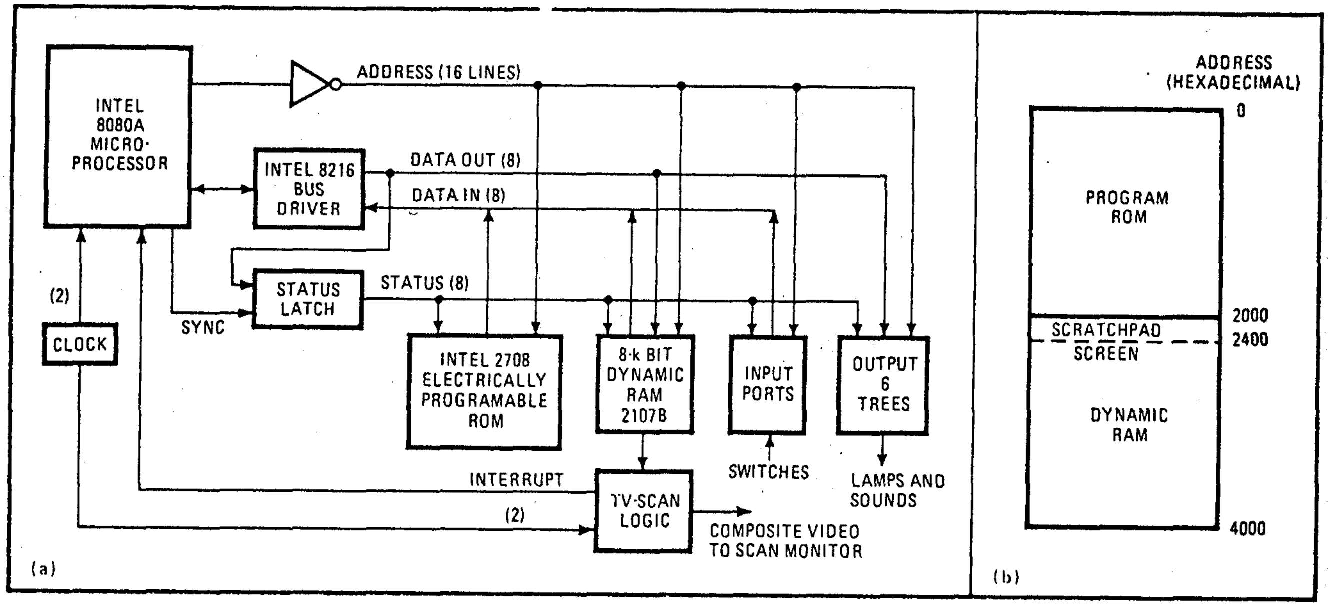
Of course, the transition from TTL was not accomplished without difficulties. In the first place, the data rate required for the faster scan to keep the image on the video screen is far beyond the capabilities of microprocessors. And, although this problem was solved with a random-access memory, the technicians were at first frightened by the microprocessor and did not fully understand the waveforms involved in its operation.

Training quickly increased the proficiency of the technicians, and a read-only memory was programed to isolate problems they had with RAMs. With this help and their increasing expertise, the technicians could repair twice as many defective boards as before, and ironically, the high repair rate created a serious parts shortage as suppliers struggled to keep with demand.

The microprocessor was adapted for use with the video display by storing the picture information in a large RAM, and reading it out on to the screen one bit at a time, while conventional horizontal and vertical deflection circuits generate the raster just as in any home TV set. The pattern on the screen is 224 horizontal lines, each line consisting of 256 dots. The RAM therefore must hold the 224 $\times$ 256 bits of information that determine whether the individual dots are bright or dark.

In this arrangement, the microprocessor is not used for scanning the picture, but merely changes the contents of the RAM at a slow rate—once every frame or even every few frames. The only interaction between the microprocessor and the screen-scanning logic is an interrupt signal at the bottom of the screen that the microprocessor uses for synchronization and timing.

Midway chose the Intel 8080A microprocessor, 16 In-



1. Game circuit. Because microprocessor is too slow to generate raster scan for arcade video game, image is plucked from RAM and impressed on screen during conventional raster scan (a). Microprocessor responds to inputs from switches controlled by players, forms new images according to program stored in electrically alterable ROM. Memory map (b) shows ROM and RAM for typical game.

tel 2107 dynamic RAMs and an Intel 2708 electrically programable read-only memory (Fig. 1). Intel's MDS 800 hardware/software development system [*Electronics*, May 29, 1975, p. 95] was modified to develop and debug programs. The Intel system was chosen for speed, register capacity, availability of parts, and support in developing games and programing.

RAM configured for screen pattern

The RAM contains 57 kilobits to correspond to the screen pattern of 224 by 256 dots. The raster reads out the memory in 1/30 second. The microprocessor uses 1 kilobyte by 8 bits of the RAM as a scratchpad without any significant loss of vertical resolution. Conventional input/output ports are handled by the microprocessor as on/off switches.

A series of games has been written in 3 to 5 kilobytes of instructions (Fig. 2). A good portion of each game consists of patterns. A typical game requires 8 kilobits of read-only memory for the program. An equivalent hard-wired circuit could require 150 TTL packages. The microprocessor writes the image on the screen by moving the two-dimensional pattern changes from the programable ROM to the RAM a byte at a time.

Each byte move shifts the pattern incrementally to simulate motion. The interrupt limits the number of operations within the interrupt routine. Since the interrupt generates the synchronous signals, it provides timing information to the main program. Its accurate clock times an output switch that must be kept open for a certain length of time. Finally, the interrupt checks the state of input ports and flags changes to the memory.

A game program is designed in two parts: the main game-logic pattern and the interrupt. The main program sets an action in motion and checks for its completion. The interrupt moves the pattern and informs the main program an expected condition is met. For example, a pattern might be a car at point A with velocity B traveling in one direction until condition C is met.

The interrupt veers the pattern from point A at velocity B and informs the program when condition C is met. The main program then initiates the next operation, part of which may be sound or light output. The interrupt operates from the blanking pulse when the scan reaches the bottom of the video screen.

The MDS 800 for developing and debugging programs has been expanded by replacing its teletypewriter with a floppy disk, a powerful operating system, a high-speed printer, and a cathode-ray-tube terminal. The floppy disk and printer handle the vital editing and assembly function at 10 times the speed of the unmodified system. The initial debugging is checked on the MDS monitor as the new program is run from the MDS memory, and final debugging is conducted with an in-circuit emulator on an erasable programable read-only memory that is repeatedly blasted by pulses from a programable-ROM programer on the actual game board.

2. Quick on the draw. Basic microprocessor circuit is used for a variety of games by changing instructions stored in read-only memory.