the Accu-Mill

a keyboard interface for the Accu-Keyer

With this circuit and an ASCII-encoded keyboard connected to an Accu-Keyer you can send perfect Morse at 25 words per minute or better

More than six thousand Accu-Keyers have been built worldwide since the circuit was introduced in the August, 1973, issue of *QST*.[1] The Accu-Mill connects to the Accu-Keyer or to an Accu-Keyer with memory using simple circuitry. With it you can key in your message from a typewriter keyboard, then start sending with your paddle without throwing switches – great for contest operating. The basic Accu-Mill circuit requires a computer-type ASCII-encoded keyboard (many are beginning to show up in the surplus market). You'll also need an Accu-Keyer with an extra-large power supply and a negative 12-volt supply. The following features have been designed into the Accu-Mill:

1. Sixty-four or 128 characters of buffering.
2. Usable with the Accu-Keyer or Accu-Keyer with memory, including provision for external paddles.
3. "Buffer Full" light.
4. Nonbuffered operation with repeating keys and non-buffered operation with non-repeating keys, both switch selected.
5. Speed control by the Accu-Keyer.
6. Easy interface with keyboards having standard or inverted outputs.
7. Only one circuit board.

The Accu-Mill was also designed so that other functions can be added such as:

1. RTTY
2. Radio ASCII when approved.
3. Counter with digital display to let you know how much buffer has been used.
4. A back space function to allow correction of keystroke errors.
5. Three programmable "vectored" message memories that work with the keyboard buffers to allow automatic insertions in a programmed contest report, so that you can type variables such as callsign, RST and number, for example, and the machine will make insertions while you update the log.

All these extras are now on paper and should be available soon. The memory and backspace features are particularly exciting, because they allow a 10-wpm typist who makes lots of mistakes (like me) to operate in a contest at 25-wpm or better.

logic description

The basic Morse board is presented here. The board uses easy-to-obtain 7400-series TTL devices except for the buffers, which are Fairchild OptiMOS devices; and the read-only memories (ROMs), which usually have the 8200-series TTL numbering (8223, 82S23, and 82S123 tri-state), but also have the 7400 designation of 74188.

Most of the devices have standard totem-pole outputs, which provide logic levels of 0 and 1. Some have open-collector outputs, which provide logic levels of 0 and not-0. These special types are used to connect to the outside world in places where either another output or another device (such as paddles) will operate an *input*. For proper operation we provide the logic 1 level with an external pull-up resistor. This resistor provides 5 volts at low current when the device is in the not-0 state. When the device is in the 0 state, the open-collector output shunts the voltage from the resistor, and a 0 logic level occurs.

The buffers are 3341 types: first-in first-out (FIFO) shift registers, the same as used in every popular buffered Morse or RTTY keyboard. Here's how they work: A short *shift-in* pulse commands data to enter. Data goes into the input and "falls through" to the last unoccupied slot of 64 positions. Data is output by a *shift-out* pulse of any length. By analogy, imagine a long, tilted gutter with a man at the top dumping in tennis balls as fast as he can and another man at the bottom, who takes them out when he needs them. As long as the man at the top is faster than the man at the bottom, there will always be a supply of tennis balls in the gutter. That's how

By William E. Smith III, WN9OVY, 1880 Bonnie Lane, Apartment 201, Hoffman-Estates. Illinois 60194

Fairchild explains it. If you can type fast enough you'll get ahead of the output and your sending will be very smooth, or you can load up the registers and go out for coffee while the machine works your contact.

The read-only memories are of the field programmable type. You can easily program them to your code with a very simple circuit. They are used here as look-up tables. Each time a binary address is placed on their inputs, the ROMs "look up" the proper code associated with that binary address and present that code to their output lines. The 74151s are data selectors or multiplexers; they act something like single-pole, 8-throw switches, and scan the ROM outputs.

functional description

Buffer. The Accu-Mill logic diagram is shown in **fig. 1**. U1 is used only for ASCII keyboards with inverted outputs. U2 and U3 condition the signal from the keyboard that says a key has been pressed (KP). U2 and U3 provide the very short pulse needed for the buffer *shift-in* command and the control needed to disable the buffer for non-buffered operation.

U10-U13 are the 3341 buffers. Each buffer contains a channel four bits wide. We need a six-bit data channel, so the buffer is configured as two channels of data buffering, each four bits wide (two unused bits). It is probable that each channel will operate at a different natural speed, which would result in bad data output, so U14 synchronizes the two channels.

The buffers retain stable data on the output pins after *shift-out* returns low. If stable data were to remain on the input to the next stage, the keyer would lock up on one character. U15, U16 are NAND gates. When *shift-out* returns low, the gates stop data, and the next stage gets an acceptable signal (binary 111111), which prevents lockup.

So far we've looked at the first stage of the Accu-Mill. In this stage we input data from the keyboard and subject it to buffering so that we can get ahead of the transmitted output and type at our own speed. The next stage contains the magic that makes Morse code.

Morse-converter. This stage includes seven devices. U17-U20 are the read-only memories, which contain the ASCII-to-Morse conversion logic. U21, U22 are data selectors, which read the Morse code. U23 is the control counter, which drives the data selectors.

Data at the buffer output is presented to the memory address lines. Since each key has a different binary number, each key addresses a different place in memory, and the memories output a different Morse code bit for each key pressed. The Morse information appears at the memory outputs in parallel form: all dots and dashes appear at once. U21, U22 are data selectors, which put the Morse information in serial form. The data selectors start by looking at Morse bit one and sequentially scan all output bits until the character is complete. The memories are wired so that U17, U18 are dot memories, and U19, U20 are dash memories.

Accu-Keyer interface. This stage includes U24, U25. These gates provide control signals and interface to the Accu-Keyer. U24A, U24B replace the paddles. The changes in the Accu-Keyer called for in the wiring of the paddle jack (**fig. 1**) will greatly increase rf noise immunity in the Accu-Keyer. The use of 74132 Schmitt triggers in place of U1, U2 in the Accu-Keyer should cure even the most severe rf problems. Similarly, a 7414 hex Schmitt trigger could be used in high-noise environments in place of U1 in the Accu-Mill.

data flow

The letter A. The letter A is a good example to use in describing data flow through the Accu-Mill, because it has just one dot and one dash. So press A. The keyboard generates 1000001, a seven-bit binary number. But a six-bit number is sufficient to describe all 64 possible characters in memory, so ignore bit seven and leave that wire disconnected.

The six-bit code 000001 is presented to the buffer stage. KP generates *shift-in*. 000001 is entered into the buffer and falls through to the output. U14 senses valid output data and opens gates U15, U16, which invert the data. Now the data is 111110, a six-bit code that is presented to the memory section.

Each memory has only five address lines, so it would appear that there's another unused bit. The extra bit, bit six, is used to select which 32 of the 64 characters is being addressed, letters or numbers. Bit six, which is high (1), is fed to, but does not enable, memories U17, U19, which are programmed for numbers and punctuation. Bit six is also inverted by U25A, and fed to U18, U20. This low (0) signal enables U18, U20, and the memory is ready to make a letter.

The remaining five bits, 11110, address U18 (dot) and U20 (dash) to the places where Morse information necessary to make A is stored. Dot memory U18 outputs 10000000; dash memory U20 outputs 01000000. Data selectors U20, U22 are looking at position one (far left). Dot data selector U21 sees a high bit in position one and causes inverter U24A to ground the dot input of the Accu-Keyer, and a dot is sent. On the dot falling edge, control counter U23 is clocked up by a signal fed back from the Accu-Keyer output.

The output count of U23 changes and drives both data selectors up to the next output lines in the memory. Now dash data selector U22 sees a high bit in position two and causes inverter U24B to ground the dash input of the Accu-Keyer, and a dash is sent. Again control counter U23 is clocked up, and the data selectors are driven up to position three. Neither sees a high bit, so neither sends a dot or a dash. The Accu-Keyer assumes end-of-character and sends a character space.

Spacing considerations. A special condition exists for the seven-baud word space. The keyboard has a space bar, but the Accu-Keyer is capable of sending character-spaces only – not word spaces, so the Accu-Keyer needs to be tricked. A word space always follows a character which has a character space. The character space will be the first three of the seven-baud word space. The space bar will be programmed to send the letter E through the Accu-Keyer. The letter E is one baud, which totals seven baud with the previous character space.

To avoid letting the E get to the transmitter, the

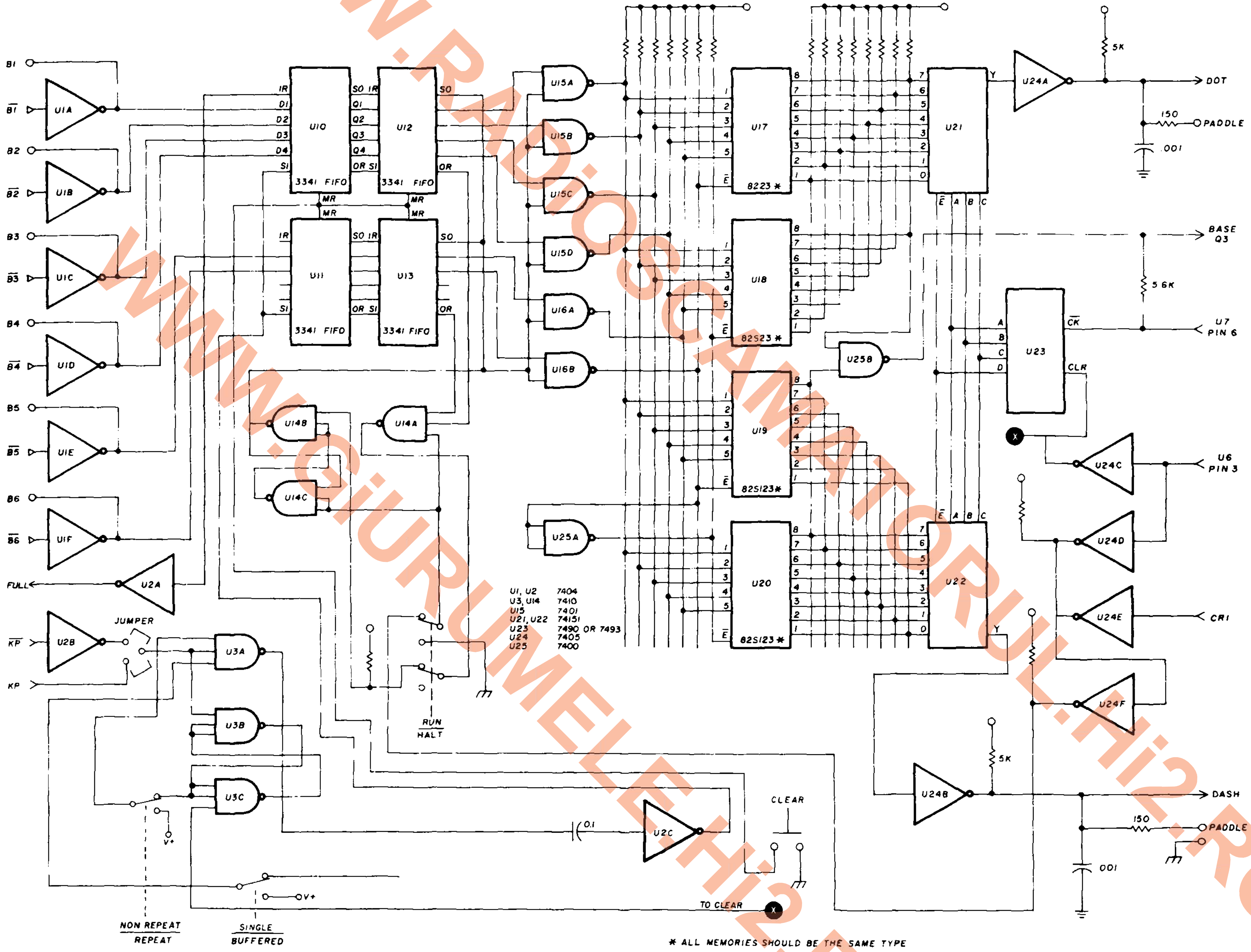

fig. 1. Logic diagram for the Accu-Mill. Shown are methods for connecting three different PROM chips, though all should be of the same type. A complete memory programming chart is supplied with the circuit board, or 82S123s are available preprogrammed. In this circuit all resistors are 2200 to 3300 ohms, ¼ watt, terminated with V+ unless indicated otherwise. All power-supply connections are assumed, as is proper bypassing of V+. Numbered wires are for logical reference, not pin numbers. Keyboard connections are on the left, Accu-Keyer connections are on the right.

memory is programmed differently for this special E to make it seem transparent at the transmitter. The dot memory will be 10000001; dash memory will be 00000001. Note that the first bit (left) in the dot memory will send a dot; the second bit is empty, which signals end of character. A character space is sent, the counter is cleared, and the next character begins. Note also that the data selectors never get to position eight to read the extra ones — this is the trick. These two extra bits are fed to NAND gate U25B. Its output goes low and swamps the keying signal before it gets to output transistor Q3 or Q5. As far as the Accu-Keyer is concerned the letter E was sent, but it never gets to the transmitter.

Using the same logic, a 36-baud pause can also be generated by programming a dot, 00000001, and a dash, 11111111. The keyer sends --- --- --- --- --- --- - ---. Including the character spaces, that's 36 baud of data, but it never gets to the transmitter. This feature can be put on the # key. The other upper case keys are used for signals such as $\overline{AR}$, $\overline{SK}$, $\overline{KN}$. The only key that cannot be used is the @ key. This space must be programmed in memory with all zeros since this space has the address 111111, which is the same as the resting code put out by U15, U16 between characters. If this space is programmed, the keyer will lock up. The @ will halt the keyboard when pressed. This may be cleared by the halt switch.

A ready made PC board is available for the basic circuit of the Accu-Mill.* It's double-sided and measures 3 x 8 inches (76x203mm), so it should fit in just about any enclosure. A parts-placement guide and step-by-step instructions are supplied with each board. Buffers and memories are also available. This project should not be attempted on vector or perf board.

Several companies sell ASCII-encoded keyboards. These keyboards are usually used or surplus items, but one company, at least, sells inexpensive new kits. New or used, as long as the keyboard works and looks good you're in fine shape. Many keyboards have square keys with no spaces between them, which makes the enclosure easy to build since you cut only one large hole instead of many small ones. Rf has never been a problem for me, but it makes good sense to keep the wiring clean.

Interface to the Accu-Keyer is easy and requires only about six wires. The Accu-Keyer controls the keyboard speed, so you may want to move the Accu-Keyer into the keyboard enclosure and mount your paddle jack on the side of the enclosure. If your Accu-Keyer has a keying monitor, remove the RC network from pin 4 of the 555 timer and connect pin 4 directly to Q3 or Q5 base. C4, C5, R11, and R12 on the Accu-Keyer inputs should also be removed and placed on the paddle jack, as indicated on the schematic.

U1 of the Accu-Mill is used only for ASCII keyboards with inverted outputs. To test for this condition just power-up the keyboard and press A. Test the outputs. Standard boards have bits 7 and 1 high (more than 2.4 volts); in this case don't use U1. Inverted boards have bits 7 and 1 low and everything else high; in this case use U1.

A note of caution: keyboards are available with codes such as BCDIC, EBCDIC, and Hollerith. These keyboards can be used but the programming in the memories is much different. A keyboard with separate contacts for each key is suitable only if an ASCII encoder is added to it.

The 3341 buffers are MOS devices and may be sensitive to static charges. Install these last and handle them carefully. Use two 3341s for 64 characters of buffering, or all four for 128 characters of buffering. It's possible to build either a nonbuffered board by using jumpers or a board with an outrageous amount of buffering (like 512 characters or more), but neither serves much useful purpose.

The memories can be the obsolete 8223 or the newer 82S23. Both require pull-up resistors (¼ watt, 2.2k). The very new 82S123 tri-state devices don't require resistors. **Fig. 1** shows how to handle each type. A complete memory programming chart is supplied with the circuit board, or 82S123s are available preprogrammed.

IC sockets cannot be used since this is a double-sided board without plated-through holes. This type of board requires some soldering on the top side, and the plastic body of the IC socket will be in the way. Use Molex pins or solder the ICs directly.

In some places this circuit board has conductors running between adjacent pins, which means that the circuit pads are very small. A very fine-tipped soldering pencil is mandatory. Try grinding a spare tip down to a 1/32 x 1/2 inch (0.8x13mm) taper. Use very small-diameter solder. A soldering gun will ruin the board.

The 5-volt power supply should be capable of at least 2 amps. Use a 12.6-volt transformer, a bridge rectifier, and a zener diode with pass transistor and heatsink. Filter it well and use a transformer with about a 3 to 4 amp secondary rating.

The minus 12-volt supply should be capable of supplying 200 mA. Use a transformer with a 600-mA secondary rating, a bridge rectifier, and a zener with a current-limiting resistor. Filter the supply with 2500 μF at 25 Vdc ahead of the resistor and a 1 μF tantalum and 0.001 μF disc ceramic after the resistor.

The enclosure can be made of aluminum, wood, or even plexiglass. A flat piece of plexiglass with switches and keyboard mounted on top of a plexiglass or wooden box shaped like a typewriter looks good.

To operate the Accu-Mill, turn it on, select buffered or nonbuffered operation, and start typing. To use the paddle, just plug it in and wait for the Accu-Mill to finish, then start sending. You can change over without throwing any switches. The paddle may remain connected if desired. The HALT switch may be used at any time and will stop the machine at the end of the present character.

*An etched and drilled, double-sided, glass-epoxy circuit board is available for $15 from W.E. Smith, Post Office Box 544, Hoffman Estates, Illinois 60194. Prices for 3341 buffers and 82S123 memories (preprogrammed) are available upon request.

reference

1. James M. Garrett, WB4VVF, "The WB4VVF Accu-Keyer," *QST*, August, 1973, page 19.

ham radio