A separately packaged 16-bit bipolar barrel shifter that handles complex data faster than conventional shifters using software routines can be cascaded for 32-bit operation systems.

Barrel-shifter IC manipulates up to 32 bits

With bit manipulation and shifting operations assuming greater importance in microcomputer systems, designers sometimes need a broader range of capabilities than a conventional shift register offers. On these occasions a fast element called a barrel shifter comes in handy.

All current barrel shifters are part of microprocessor architectures, where they are imbedded in the ALU and therefore unavailable to system designers for non-ALU bit manipulations. This situation will change soon with the introduction of the first stand-alone 16-bit bipolar barrel shifter. A single-chip device, it is faster than a conventional software-controlled shift register and requires much less program memory. It is intended for 16-bit microcomputer systems, but its internal architecture allows it to be cascaded for operation in 32-bit systems (Electronic Design, May 26, 1983, p. 139).

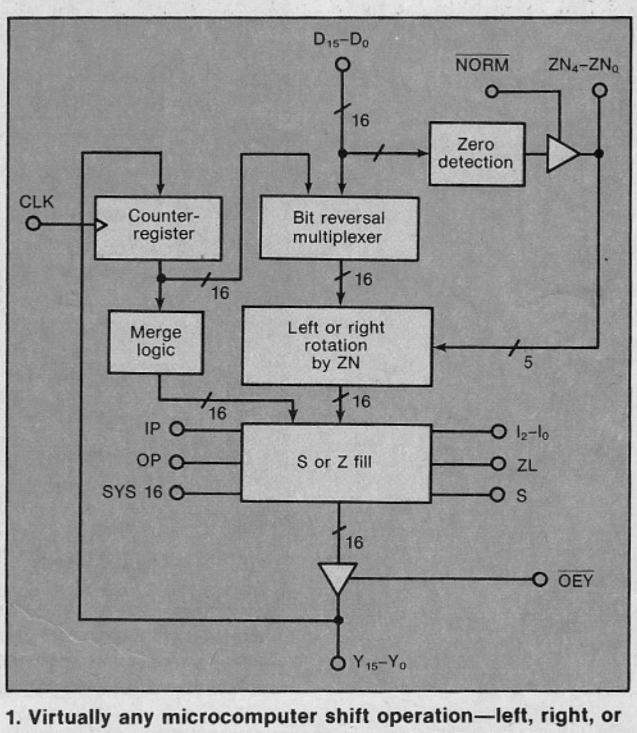
As a stand-alone device, the SN74AS897 will offer a much broader range of shifting capabilities than a conventional shift register. Whereas the normal shift register moves its entire contents one bit position in a clock cycle, a barrel shifter can perform a high-speed, or "flash," operation, shifting its entire contents any number of bit positions in one clock cycle, and it can do that without a clock input if the designer desires.

A barrel shifter also can perform basic circular shift operations—that is, bits can be shifted from, say, the output back around to the input in a wheel- or barrel-like movement. In a conventional shift register, shifting operations usually force the number of bits shifted out of the register.

Bob Bailey, Business Development Engineer
Bipolar Digital Logic Products
Jeff Niehaus, Special Projects Manager
New Products and Technology Department
Texas Instruments Inc.
PO Box 225012, Mail Station 30
Dallas, Texas 75265

In addition, the AS897 packs such features as leading-zero detection, bit setting/resetting, bit reversal, start/stop bit insertion, and parallel-to-serial and serial-to-parallel conversion. With these, the device can be used for normalization in formatting for floating-point computations, for generating bit-reversal fast Fourier transform addresses, for inserting stop/start bits in asynchronous data communications, and for buffering instructions and addresses. It offers dynamic multiplexing capabilities that are vital when a number of devices are connected in parallel (as with 32-bit words).

The AS897 is slated to operate typically at a fre-



1. Virtually any microcomputer shift operation—left, right, or circular—can be performed by the 16-bit SN74AS897 barrel-shifter IC. Using its counter-register, merger logic and S or Z fill circuitry, the chip can alter input data by programming logic 1s or 0s into any bit position of a word.

quency of 20 MHz, and power dissipation will be less than 1 W, a respectable figure for that speed. It is housed in a 52-pin plastic dual in-line package and is optionally available in a 52-lead chip carrier for systems that require greater packaging density than is possible with DIPs.

A versatile shifting architecture

Figure 1 shows the generalized block diagram of the barrel shifter. The device operates in any of eight user-programmable modes, as determined by the logic levels applied to pins I_0-I_2 (Table 1).

The first two modes—0 and 1—are basically the same, except that 0 causes data to be shifted right, whereas mode 1 is used for shifting left. Each of these modes is associated with a so-called S fill. The signal Sign-Fill Data line (S) specifies the logic level that is used to fill the positions left empty by the shift operation. For example, if five bits were shifted either left or right (but not circularly) out of the device, the empty bit positions would be filled (by the S or Z fill block) with the logic level residing on the S pin.

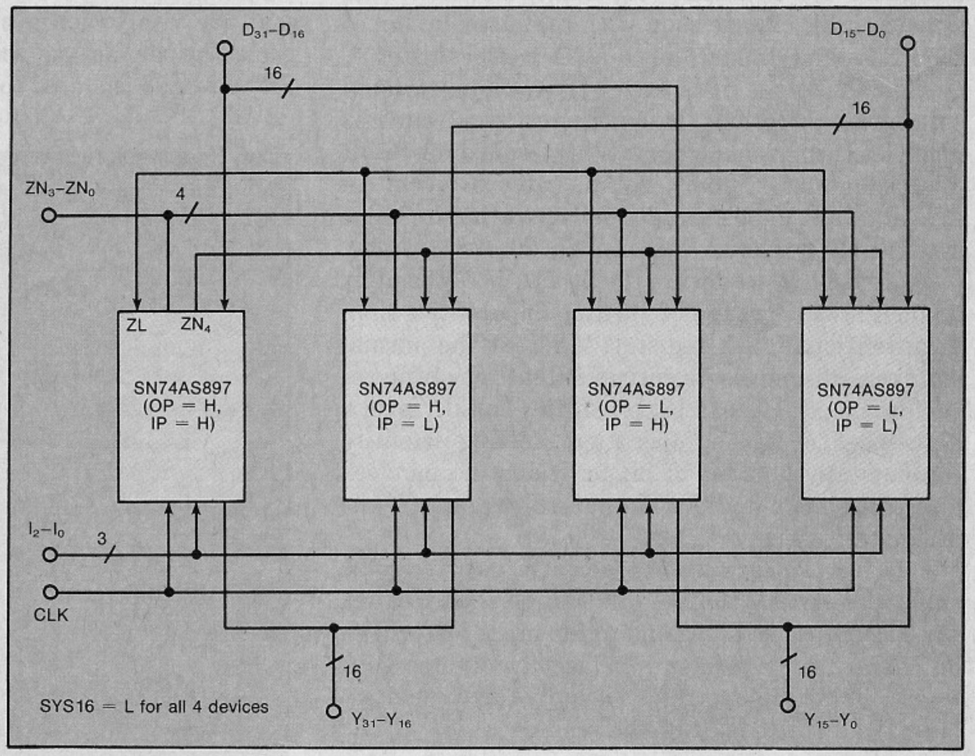
The actual number of bit positions to be shifted can be programmed by the user via the ZN_4-ZN_0 lines. ZN_4 has a dual function, one of its roles being to serve as the most significant bit of ZN_4-ZN_0 when a movement of greater than 15 bit positions is commanded in a 32-bit system.

Modes 2 and 3 permit circular shifting, to the right and left respectively. In a circular shift, the data is neither altered nor removed from the internal register but merely moved to a different bit position. As with modes 0 and 1, the number of positions to be moved is specified by the ZN lines. These rotation operations are carried out in the block labeled "left or right rotation by ZN."

The internal counter-register is 16 bits wide and is loaded when it is clocked with a positive-going transition on the Clock line (CLK).

In modes 4 and 5, input data (D_0-D_{15}, where D_0 is the LSB) is shifted (left and right, respectively) and filled with the logic level on the S line, just as in modes 0 and 1. In addition, the shifted and filled data is ORed (merged) with the contents of the internal 16-bit counter-register.

Whereas data input lines D_0-D_{15} serve only to bring data into the barrel shifter, data output lines Y_0-Y_{15} act as both inputs and outputs. As output ports, they deliver shifted data to the outside world.


D31–D16
D15–D0
16
16
ZN3–ZN0
4
ZL
ZN4
SN74AS897 (OP = H, IP = H)
SN74AS897 (OP = H, IP = L)
SN74AS897 (OP = L, IP = H)
SN74AS897 (OP = L, IP = L)
I2–I0
3
CLK
16
16
SYS16 = L for all 4 devices
Y31–Y16
Y15–Y0

2. An internal expansion capability allows four barrel-shifter ICs to be cascaded to perform shifting and data-changing operations on a 32-bit word. This type of hardware scheme is much faster than using software routines.

As input ports, the lines can be used for bringing in external data to the on-chip register. When data is entered, the Output Enable Y line (OEY) must be brought high to disable the Y output lines.

In the set/reset bit ZN mode (mode 6), the data specified on the ZN lines can be set or reset (altered). This mode also permits setting or resetting the logic level programmed on the S line.

In the normalization mode (mode 7), the input word (D_0-D_{15}) is shifted left until its first logic 1 bit is located in the MSB position of the output (Y_{15}). The number of shift operations required to accomplish this is sent out on the ZN lines, which, like the Y lines, serve as both inputs and outputs. To control the I/O function of the ZN lines, the NORM input serves as an enabling element. When the NORM line is low, the ZN lines can send data to the outside world.

Three control lines—SYS16, IP, and OP—are responsible for configuring the AS897 in system applications. For example, when SYS16 is high, the device is in the 16-bit mode, which means that it acts as the only barrel-shifting element in the system. When SYS16 is low, it is in the 32-bit mode and operates with three other AS897s to perform 32-bit shift operations. Together with IP and OP, SYS16 determines the device's I/O configuration (Table 2).

The shifter as a system element

If the chip is in the 16-bit mode and OP is low, the data to be manipulated or shifted is that on the input lines (D_0-D_{15}). When OP is high, the chip shifts the data in its internal register (Q_0-Q_{15}).

When the barrel shifter is in the 32-bit configuration, interfacing with three identical chips, the IP and OP pins must be programmed as illustrated (Fig. 2) to permit proper shifting. IP is defined as the input position and OP as the output position of the data bits. If both pins are programmed high, the device interprets this to mean that it is the most significant package. Similarly, if both pins are programmed low, the device is the least significant package. Therefore the AS897 at the far left handles the MSBs of the 32-bit word and the chip at the far right is responsible for the LSBs. Of the middle two packages, the one with OP high and IP low handles less significant (lower-order) input bits than the package with OP low and IP high, and it provides the fill bits for the most significant output half.

The 32-bit mode involves the use of the ZL line, which links all four devices. ZL is another line with a dual function—it acts as either an input or an output. If the ZL output of the LSB package goes high, it indicates to the other packages that the least significant 16 bits of the input word are all

Table 1. The functional modes of the SN74AS897 barrel shifter

Mode	I_2	I_1	I_0	Function
0	0	0	0	Shifts right ZN places and fills with S
1	0	0	1	Shifts left ZN places and fills with S
2	0	1	0	Shifts right circularly ZN places
3	0	1	1	Shifts left circularly ZN places
4	1	0	0	Shifts right, fills with S, and merges
5	1	0	1	Shifts left, fills with S, and merges
6	1	1	0	Sets/resets bit ZN
7	1	1	1	Normalizes data and puts out count on ZN

Table 2. I/O configuration

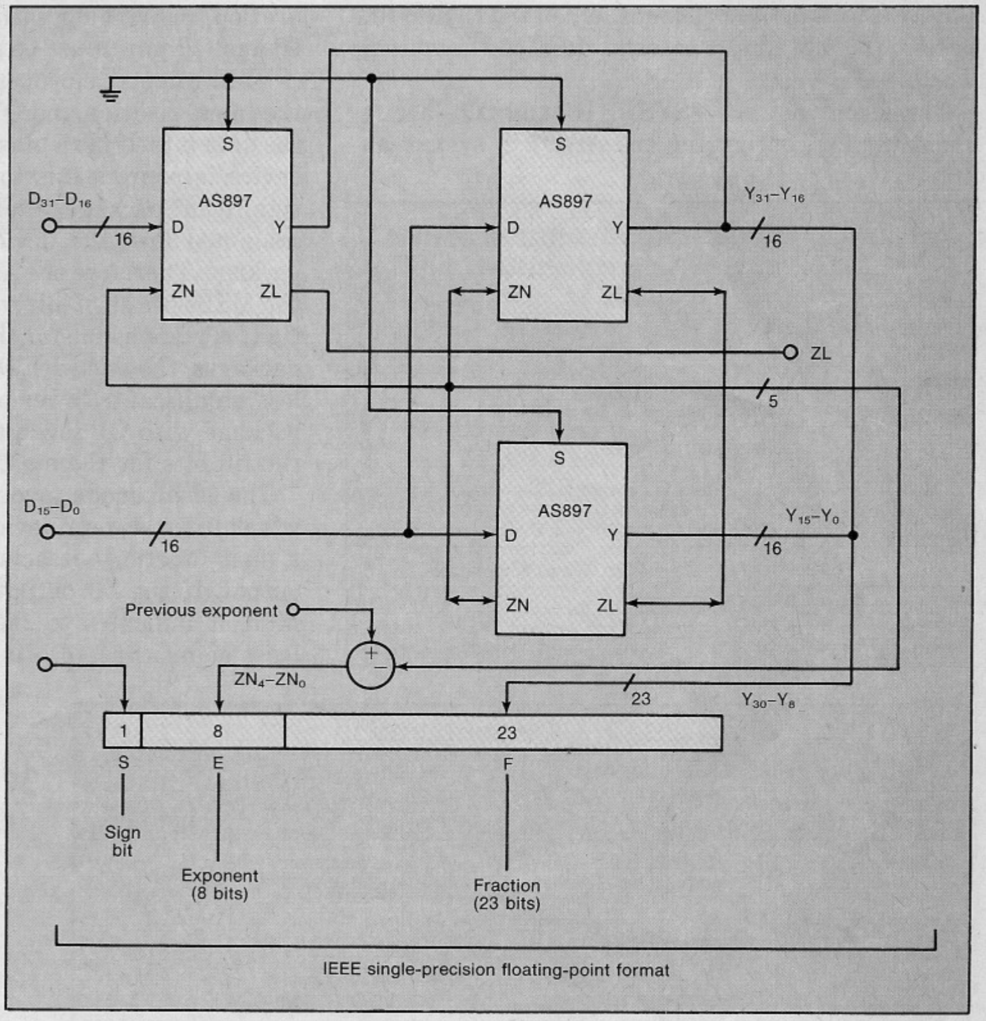
Control lines					
SYS16	IP	OP	Input data	Output data	Configuration
L	L	L	D_0-D_{15} is the least significant input position	Y_0-Y_{15} is the least significant output position	32 bits
L	L	H	D_0-D_{15} is the least significant input position	Y_0-Y_{15} is the most significant output position	
L	H	L	D_0-D_{15} is the most significant input position	Y_0-Y_{15} is the least significant output position	
L	H	H	D_0-D_{15} is the most significant input position	Y_0-Y_{15} is the most significant output position	
H	L	L	The input data is normal	The data to be shifted is D_0-D_{15}	16 bits
H	L	H	The register data is normal	The data to be shifted is the register output (Q_0-Q_{15})	
H	H	L	The input data is bit-reversed	The data to be shifted is D_0-D_{15}	
H	H	H	The register data is bit-reversed	The data to be shifted is the register output (Q_0-Q_{15})	

logic 0s. The ZN_4 output of the MSB device is used in a similar manner—that is, if ZN_4 is high, it indicates to the other devices that the most significant 16 bits of the input word are all logic 0s.

Entire 32-bit word handled

The D_0–D_{31} inputs are cross-connected to the four AS897s to permit the devices responsible for each half of the output word—Y_{31}–Y_{16} and Y_{15}–Y_0—to receive the entire 32-bit input word. To understand the need for this connection, assume the opposite situation: If the two devices responsible for the Y_{15}–Y_0 outputs received only input bits D_{15}–D_0, they obviously could not operate on high-order bits, nor would they be "aware" of the operations, if any, required for those bits. The same argument holds for the devices responsible for the Y_{31}–Y_{16} outputs and the low-order input bits. When the barrel shifters are connected as in Fig. 2, each device pair (most and least significant packages) "sees" the entire 32-bit input word. Therefore each pair is able to perform shifting, filling, and altering operations on any of the 32 bits in the input word. This capability is particularly important in shift operations, since data bits are often shifted between devices, with low-order bits progressing up to the high-order shifters.

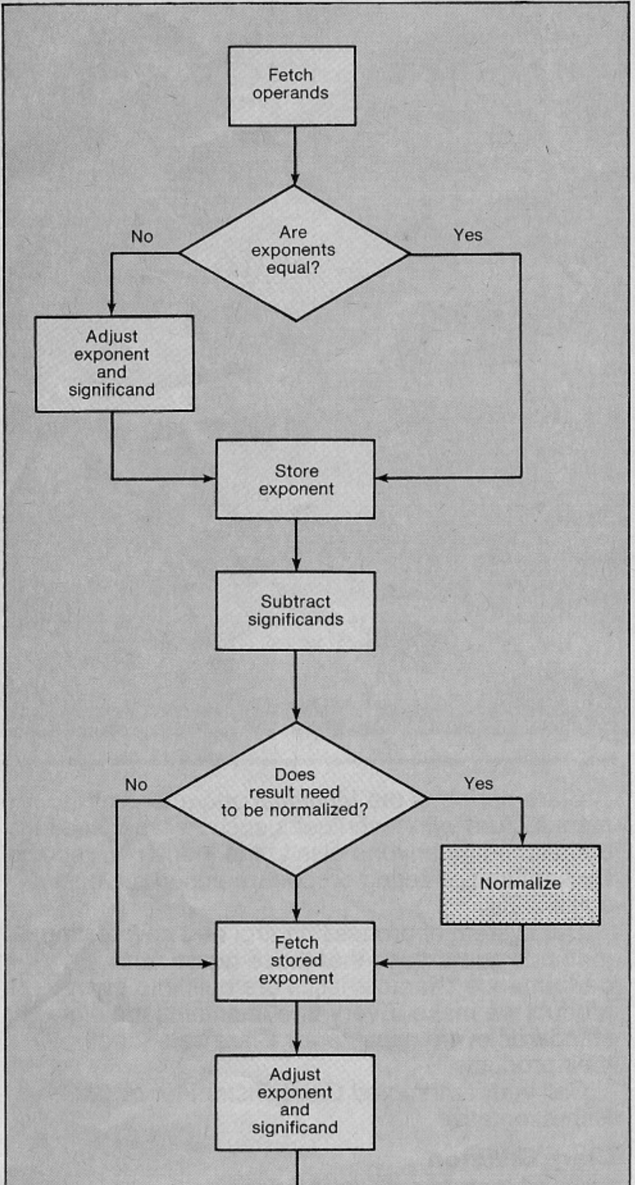
Because many current microcomputer systems



3. Normalizing the result of a 32-bit floating-point computation requires a circuit with three barrel shifters. When normalizing a small fractional binary number, the data word is shifted to the left to eliminate all the leading zeros.

handle numbers in a floating-point format, they must be able to normalize these numbers when required. (A floating-point number is said to be normalized when it is represented as a mantissa [significand] with a minimum exponent.) Three AS897 barrel shifters are needed to perform normalization of 32-bit numbers according to the IEEE single-precision floating-point format (Fig. 3).

Normalization is necessary in a microcomputer architecture to preserve the system resolution after a subtraction or other algorithm yields a fractional result that differs from the original data by orders of magnitude. The resulting fraction is too small to be represented in the standard data format of the computer and must be stored as an exponent and mantissa. To normalize a fractional binary number, the leading zeros must be stripped away, the number shifted to the left, and the number of shifts required must be identified (Fig. 4).



4. This flow chart illustrates a method for subtracting two floating-point numbers and normalizing the result. Although the barrel shifter does not perform the subtraction, it alone handles the normalization operation.

The ZN ports determine the number of shifts needed to put the most significant logic 1 input bit in the most significant output position (Y_{31}). The MSB of the floating-point significand is an implicit bit; it is always a logic 1, but it is dropped to conform to the IEEE format. It must be added before additional arithmetic operations are performed. The only limitation of the AS897s in this application is that the sign and exponent bits must be handled with external hardware.

As an alternative to the IEEE floating-point format, a circuit arrangement very similar to that shown in Fig. 3 can be configured to normalize a 32-bit data word to yield a 32-bit significand and a 5-bit exponent adjustment.

Data field manipulations

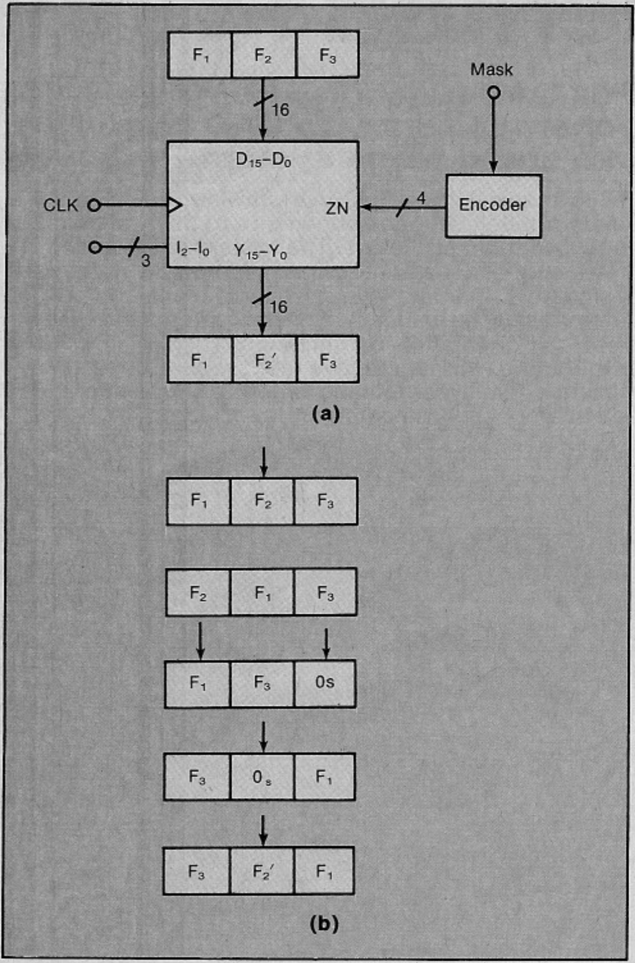
A useful application of the AS897, commonly required to adjust digital-communication word formats, is the extraction or insertion of data within an existing data field (Fig. 5).

Assume that each of the fields F_1, F_2, and F_3 is a 16-bit data field. Also assume that it is necessary to change the data in F_2. A dynamic mask and encoder are used with the barrel shifter to control the number of shifts required for each data field. The mask is variable and enters the number of shift operations required by each data field.

The operation begins when the original data in the three fields is applied to the AS897's D_0–D_{15} input lines (step 1). In step 2, the input data is shifted circularly until the F_2 data is stored in the register with the F_2 MSB in the most significant register location.

The operation in step 3 is called a purge. Data in the internal register is shifted to the left, and all F_2 bit positions are filled with logic 0s. Then the altered word is returned to the internal register.

Step 4 requires that the purged field be shifted circularly to the left to restore it to its correct position—MSB first, LSB last. In step 5, the new data is inserted into the field through a merger with the register containing the cleared F_2 field. The new, merged data can then be sent out onto a bus through the Y_{15}–Y_0 output lines. When the dynamic mask sends in a new code, the appropriate field can



5. Field insertion—changing the information content of a block of data (a)—is a task uniquely suited to a barrel-shifter chip. The device can shift data circularly, altering it only as necessary (b). This capability is important for positioning a data block in the chip's register and operating on varied data blocks in the system.

be extracted and loaded into the AS897. The operation appropriate to the new field—insertion, extraction, and so on—is then carried out.

Each step can be accomplished in a single clock cycle of the barrel shifter. If the chip runs at a 15-MHz clock rate, the five steps each are completed within a 3-MHz period (about 330 ns). In the data communications application, this type of field insertion not only is useful for altering the data, but also can be performed asynchronously with respect to the system clock, often at a higher speed than that of the clock. Moreover, while the field data is inside the barrel shifter, the chip provides an intermediate storage area for the information.□