World Academy of Science, Engineering and Technology
International Journal of Electronics and Communication Engineering
Vol:1, No:8, 2007

A New Digital Transceiver Circuit for Asynchronous Communication

Aakash Subramanian, Vansh Pal Singh Makh and Abhijit Mitra

Abstract— A new digital transceiver circuit for asynchronous frame detection is proposed where both the transmitter and receiver contain all digital components, thereby avoiding possible use of conventional devices like monostable multivibrators with unstable external components such as resistances and capacitances. The proposed receiver circuit, in particular, uses a combinational logic block yielding an output which changes its state as soon as the start bit of a new frame is detected. This, in turn, helps in generating an efficient receiver sampling clock. A data latching circuit is also used in the receiver to latch the recovered data bits in any new frame. The proposed receiver structure is also extended from 4-bit information to any general n data bits within a frame with a common expression for the output of the combinational logic block. Performance of the proposed hardware design is evaluated in terms of time delay, reliability and robustness in comparison with the standard schemes using monostable multivibrators. It is observed from hardware implementation that the proposed circuit achieves almost 33 percent speed up over any conventional circuit.

Keywords— Asynchronous Communication, Digital Detector, Combinational logic output, Sampling clock generator, Hardware implementation.

I. Introduction

NON-COHERENT or asynchronous communication is a physical layer transmission technique [1] which is most widely used for personal computers providing connectivity to personal appliances and is treated as a core issue in the field of serial communications. The most significant aspect of such communication models is that the transmitter and receiver clocks are independent, or, in other words, the receiver does not have any prior knowledge of the transmitting clock phase [2]. This leads to the problem at the receiving end regarding sampling the incoming data bits at the correct instants. In order to achieve these exact sampling indices, data bits are transmitted in the form of frames rather than individual bits. Each such frame consists of certain start bits, followed by the information bits, and terminated by a few stop bits [3], [4]. The start bits are meant for the receiver information, about the initialization of a new data sequence, sent by the transmitter. The receiver, in turn, processes the same start bits as a stimulus to generate the receiver clock that appropriately samples the incoming data at right indices. Two successive frames, however, have no timing relationship and may be separated by any arbitrary idle period. Note that in idle state, i.e., when the transmitter does not send any new data frame, a constant LOW or HIGH value is received at the receiver and the value (commonly chosen as HIGH) is called the idle state value. From the system design viewpoint for asynchronous transmission, therefore, one should primarily select certain parameters carefully such as frame size, start bits, termination sequence and idle state value. The next task involves the design of a transmitter which properly places the information bits in corresponding frames. Finally, implementing an efficient receiver to recover the transmitted data bits from the frames being received without any knowledge of transmitted clock phase completes the assignment.

In the context of asynchronous communication, various techniques have been proposed [5], [6] with different transmitter designs illustrating several possible ways to encode the transmitted data and different receiver structures for decoding purpose. Mostly, the standard hardware schemes involve the use of monostable multivibrators at the receiver to recover the exact number of data bits following the start bits. The monostable multivibrators commonly rely on variable components such as resistance and capacitance values, as well as they often account for a major part of the propagation delays associated with the receiver. However, to the best of our knowledge, no effort has so far been made to extend this treatment to fully digitized transceiver designs that present more challenging tasks including a digital output feedback. Moreover, use of all digital components in the receivers reduces the propagation delay considerably.

In this paper, we present a digital transceiver circuit for asynchronous frame detection where both the transmitter and receiver contain all digital components, thus avoiding usage of monostable multivibrators in the receiver with unstable external components such as resistances and capacitances. The proposed receiver circuit employs a combinational logic block producing a binary output which changes its state as soon as the start bit of a new frame is detected. Next, an efficient receiver sampling clock is generated employing this output as a feedback. A data latching circuit is also used in the receiver to latch the recovered data bits in any new frame. The proposed treatment is then extended from information length of 4-bit to any general n data bits, targeted for any general 16-bit fixed point digital signal processor (DSP), with a common expression for the output of the combinational logic block. It is verified in hardware that the proposed design achieves better results in comparison with the standard designs in terms of reliability as well as timing efficiency. The detailed aspects of this new design with the extension toward a general receiver circuit are presented in the sequel.

A. Subramanian, V. P. S. Makh and A. Mitra are with the Department of Electronics and Communication Engineering, Indian Institute of Technology (IIT) Guwahati, North-Guwahati - 781039, India (e-mail: akashs@iitg.ernet.in, vanash@iitg.ernet.in and a.mitra@iitg.ernet.in).

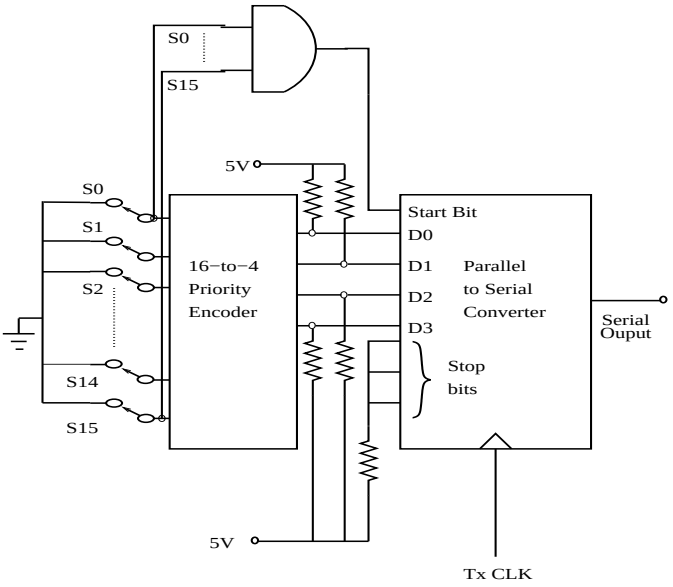

Fig. 1. The transmitter structure.

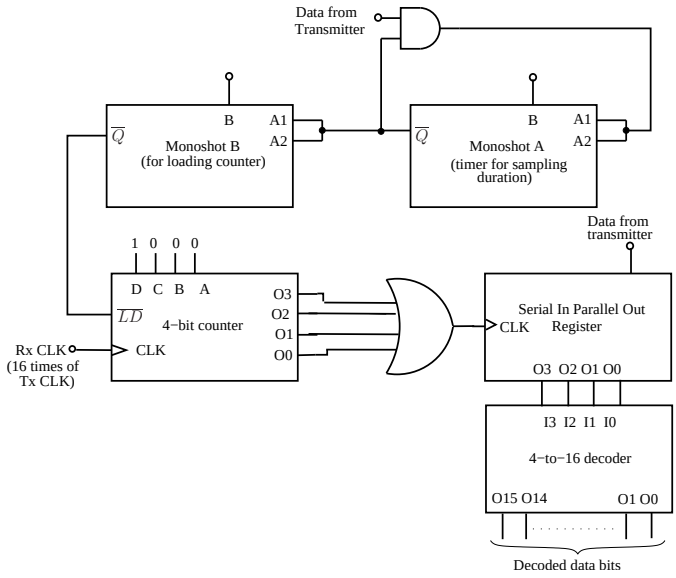

Fig. 2. The conventional receiver structure.

II. The Proposed Transceiver Circuit

The design of the transmitter used in the proposed scheme is shown in Fig. 1. The input to the transmitter is a series of switches, each equivalently representing a specific function. A frame is formed by encoded data bits, derived from these functional switches, along with start and tail bits. This frame is fed to a *parallel to serial converter(P/S)* which converts the parallel set of data to serial data at the rate of the transmitter clock fed to the converter. The P/S converter is connected to a serial link which transmits the data over a physical medium. In this scheme, a set of 16 switches is used which are encoded to four bits using a 16-4 priority encoder. One start bit is used to indicate the start of the frame. The start bit is set as the output of the AND operation on all the 16 switches. Until a switch is pressed, the output of the encoder is a set of four ones and the start bit is set to high. Thus, the transmitted data is simply a continuous sequence of 1's representing the idle frameless state. Whenever a switch is pressed or goes low, the four bit binary pattern corresponding to the switch pressed appears on the output lines of the encoder. The start bit, being the output of the AND operation of the switches, now goes low. This indicates the start of the frame.

A. The standard receiver circuit

The conventional circuit for implementing the receiver is shown in Fig. 2. As stated before, it can be seen that the structure consists of two monostable multivibrators. Monoshot A is used to sample the incoming data from the transmitter uptil the MSB of the information bits following the start bit encapsulated in the frame. The output of monoshot A is high for a duration of $(n+1) \times T$ where n is the number of data bits and T is the bit period. Sampling operation is done only when the output of monoshot A is high, ensuring that only the start bit and the following n data bits are sampled. Taking $n = 4$ as used in the experiment, the range of pulse width for the monoshot for correct operation is

$$4.5T < t_w < 5.5T \tag{1}$$

where T is the bit period and t_w is the pulse width of the monoshot.

Monoshot B is used to load the counter with 8 $(1000)_b$. The complemented output $\overline{Q}$ of monoshot A is connected to the two inputs of monoshot B. Whenever the start bit arrives, the counter is loaded with eight and the sampling operation starts till the output of monoshot A goes low.

The counter is used for generating the sampling clock at the receiver from an oscillator clock which operates at 16 times the transmitter clock. The counter is initially loaded with the value 8 $(1000)_b$. When counting starts, the state of the Mod-16 counter changes as follows, $1000 \rightarrow 1111 \rightarrow 0000 \rightarrow 1000$, and this cycle is repeated till monoshot A's output is high. By ORing the four outputs of the counter, a periodic clock is generated with it's rate equal to the data rate and whose falling edge falls approximately at the center of the bit interval. This falling edge is used as the sampling instant.

The serial data enters the *serial to parallel converter (S/P)* at the sampling edges and appears as parallel output. The parallel set of data is fed to a decoder, a 4-16 decoder in this case, and the outputs are connected to the LED's which are grounded through resistors of 330Ω.

B. The proposed receiver circuit

The proposed structure of the receiver is shown in Fig. 3. In this discussion, it is assumed that the frame size is 8 bits with 1 start bit, 4 encoded data bits and 3 stop bits. The structure can be divided into three parts according to functionality :

1) Combinational logic block
2) Sampling clock generator
3) Data latching circuit

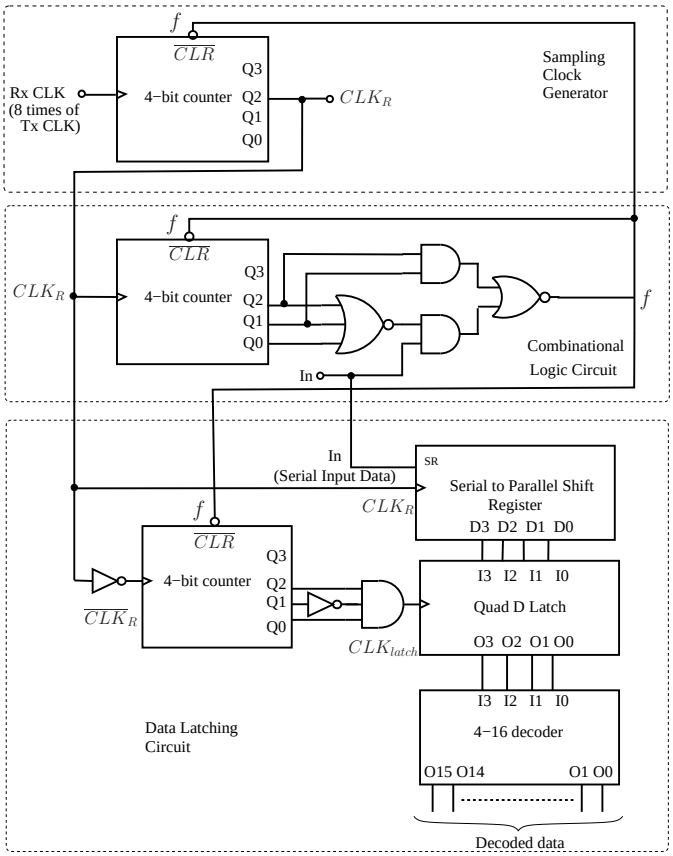

Fig. 3. The proposed receiver structure.

Each of these 3 parts is controlled by a common feedback signal, referred to as f in this paper, which is the output of the combinational logic block.

1) Combinational logic block: The combinational logic block is designed such that in idle state, the output f is always 0 as shown in bold in Table I. As soon as the start bit is detected, the output f goes HIGH and remains HIGH for a duration of $5\frac{1}{2}$ (1 Start bit + 4 Data bits + 1/2 buffer period) bit periods. It is in this duration that the 4 encoded data bits are extracted from the received frame. The truth table for generating the required combinational logic is shown in Table I. Using 4 input *Karnaugh map (K-map)* simplification, the obtained expression for f is,

$$f = \overline{Q}_2 \cdot \left(\overline{In} + Q_1 + Q_0\right) + Q_2 \cdot \overline{Q}_1 \quad (2)$$

in SOP format, and,

$$f = (\overline{In} + Q_2 + Q_1 + Q_0) \cdot (\overline{Q}_2 + \overline{Q}_1) \quad (3)$$

in POS format. The above expression for f is implemented using 3/4 input NOR and AND gates using the POS expression for f in order to get the minimum number of gate stages.

2) Sampling clock generator: The receiver clock is generated using an oscillator at eight times the transmitter clock frequency . The oscillator output is referred to as CLK_8. A Mod-8 counter is used to generate the desired clock, referred to as CLK_R, for sampling the data at the receiver. This is done by taking the output Q_2 of a 4-bit counter as CLK_R. To adjust the phase of CLK_R such that its positive edges

TABLE I
TRUTH TABLE FOR THE COMBINATIONAL LOGIC CIRCUIT

In	Q_2	Q_1	Q_0	f
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0
1	**0**	**0**	**0**	**0**[a]
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	0
1	1	1	1	0

[a]Idle state of f.

occur at the center of each bit, we apply the signal f to the active low clear $(\overline{CLR})$ input of the Mod-8 counter. Whenever the communication system is in idle state, f is LOW and hence the output clock CLK_R LOW. With the arrival of a new frame, the signal f goes HIGH. This makes the clear input to the Mod-8 counter HIGH and it starts counting from $000 \rightarrow 100 \rightarrow 111 \rightarrow 000 \rightarrow 100 \rightarrow 111 \rightarrow \ldots$ until f goes LOW again. After f makes a transition from 0 to 1, it takes 4 periods of CLK_8 for the generated receiver clock CLK_R to go from LOW to HIGH. This transition of state, which serves as the first positive edge of the generated receiver clock CLK_R, occurs exactly at the center of the start bit. Each subsequent positive edge of CLK_R occurs at a spacing of 8 cycles of CLK_8 from the first positive edge. Hence, every subsequent positive edge of CLK_R occurs at the center of a data bit contained in the frame and can therefore be used to sample the received data bits. When f goes back from 1 to 0, the clock CLK_R goes LOW again. Since f goes HIGH for a duration of $5\frac{1}{2}$ bit periods whenever a start bit is detected, we obtain 5 cycles of the sampling clock each time a new frame arrives.

3) Data latching circuit: The data latching circuit is used to latch the recovered data bits at the receiver. The data latching circuit is composed of a 4-bit shift register, a Quad D-latch, and a counter. The incoming serial data at the receiver is fed to the serial input pin SR of the shift register. The shift register is driven by the sampling clock CLK_R while the counter is driven by the inverse of the sampling clock CLK_R. The clock which drives the Quad D-latch is obtained by combining the outputs of the counter as shown in figure 3.

When a new frame arrives, initially, the start bit is applied to the serial input of the Shift Register. The first positive edge of clock CLK_R occurs at the center of the start bit following which the start bit advances to the MSB of the shift register. After one bit duration, the serial input of the

shift register changes to the first encoded data bit which then propagates to the MSB of the Shift register at the next positive edge of CLK_R while the start bit advances one to the right. Continuing in this way, after 5 positive edges of the clock CLK_R, the 4 output lines of the shift register contain the data that was present in the frame. With the next negative edge of CLK_R, the counter advances to the state 101 and the combinational output of the counter makes a transition from 0 to 1. This serves as a clock for the quad D-latch and the encoded data bits present at the output of the shift register are latched to the quad D-latch. Under such constraints, the duration of f for the correct operation of the circuit should lie in the range,

$$nT < t_f < (n+1)T \tag{4}$$

$$5T < t_f < 6T, \ for \ n = 4 \tag{5}$$

where t_f is the duration for which f is HIGH, n is the number of data bits and T is the bit period.

While the system is in idle state, the data latching circuit is inactive and thus the state of the quad D-latch remains the same as the data contained in the last received frame.

III. Generalization of the Scheme for any n Data Bits

The receiver structure described above has been designed assuming that the frame size is 8 bits and that each frame consists of 1 start bit, 4 data bits and 3 stop bits. With 4 data bits per frame, it is possible to execute 16 different tasks. However, for practical purposes, the number of different tasks to be executed may vary depending on the application and hence, the number of data bits to be sent out in each frame may also vary. Taking the number of data bits to be sent in a frame to be n, the truth table which determines the combinational logic block must be modified such that as soon as a start bit is detected by the receiver, the output f goes HIGH for a duration of $n + 1 + \frac{1}{2}$ bit periods. Also, in the data latching part, the states of the counter must be combined in such a way that the latching edge occurs after $n + 1$ cycles of the generated clock CLK_R. For each value of n, therefore, a different combinational logic block has to be implemented, which is, however, not feasible from the design viewpoint.

The main focus here, therefore, is to extend the proposed asynchronous communication model as a general design for any 16-bit fixed point digital signal processors (DSP) so that the size of the frames to be transmitted is known apriori to be of 16 bits with any n data bits. The transmission scheme is modeled such that each transmitted frame is composed of 1 start bit (LOW), n data bits and $16 - n - 1$ stop bits (all HIGH). At the receiver, the number of data bits contained in each frame is assumed to be 12 and the combinational logic block is accordingly designed such that the output signal f goes HIGH for a duration of $12 + 1\frac{1}{2}$ bit periods after the detection of the start bit. With this, the expression for f in POS format changes to,

$$f = (\overline{In} + Q_3 + Q_2 + Q_1 + Q_0) \cdot (\overline{Q}_3 + \overline{Q}_2 + \overline{Q}_1 + \overline{Q}_0) \tag{6}$$

However, in the data latching circuit, we still combine the states of the counter in such a way that the output goes HIGH for a duration of *only* $n + 1$ cycles of the generated clock CLK_R.

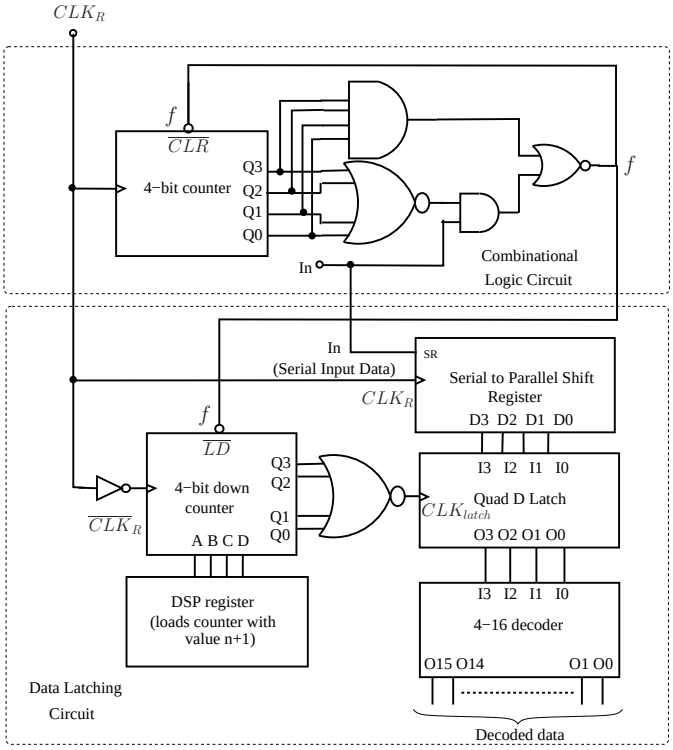

Fig. 4. The generalized receiver structure.

Thus, the whole receiver structure can be divided into a fixed and a variable part. The fixed part, consisting of the sampling clock generator and the combinational logic block, remains unchanged irrespective of the number of data bits n being transmitted in each frame. This is because it assumes that each frame is composed of 1 stop bit, 12 data bits and 3 stop bits. On the other hand, the variable part, which consists of the data latching circuit, changes with the number of data bits n contained in each frame and is instrumental in extracting the exact number of encoded data bits from the received frame.

The generalized receiver structure is shown in Fig.4. Since the proposed scheme is intended for asynchronous communication for a 16-bit fixed point DSP processor, it may be possible to program the variable part through software such that it gets automatically readjusted whenever the number of data bits n contained in a frame changes. One such possible way may be to use a down-counter instead of an up-counter in the data latching circuit and load it with the value $n + 1$ by applying the feedback signal f to the active LOW load input of the counter instead of the clear input. The states of the counter can be combined such that the output goes HIGH when the counter reaches the state 0000, which would happen after $n + 1$ cycles of the generated sampling clock CLK_R.

IV. Results and Discussions

While analyzing the time domain performance of the two circuits, only those circuit components are considered that are different in the two circuits and cause unequal delay.

TABLE II
PROPAGATION DELAY OF CIRCUIT COMPONENTS

Component	Min	Typical	Max	Units
Monoshot	-	47.5	75.0	ns
Counter	-	-	28.5	ns
4 Input NAND Gate	1.95	3.45	4.65	ns
4 Input AND Gate	1.50	5.10	7.10	ns
4 Input NOR Gate	1.15	4.60	7.40	ns
4 Input OR Gate	1.30	5.45	7.90	ns

TABLE III
TOTAL PROPAGATION DELAY OF THE THREE RECEIVERS

Design	Min	Typical	Max	Units
Standard	124.80	128.95	186.40	ns
Proposed	64.60	85.60	100.80	ns
Generalized	63.10	84.10	99.30	ns

The propagation delay for the conventional and the proposed receiver circuits can then be expressed as,

$$t_{pd}^{conv} = 2t_{pd,monoshot} + t_{pd,counter} + t_{pd,OR} \quad (7)$$
$$t_{pd}^{prop} = 2t_{pd,counter} + 2t_{pd,AND} + 4t_{pd,NOR} \quad (8)$$
$$t_{pd}^{gen} = 2t_{pd,counter} + t_{pd,AND} + 4t_{pd,NOR} \quad (9)$$

where, t_{pd}^{conv} represents the propagation delay of the conventional circuit, t_{pd}^{prop} represents the propagation delay of the proposed receiver circuit and t_{pd}^{gen} represents the propagation delay of the generalized receiver circuit.

The propagation delay of the proposed scheme is calculated when f switches from the idle state of LOW to HIGH upon receiving the start bit. Since the counter of combinational logic block does not add to the delay at this stage, the expression for the propagation delay for the proposed structure contains only the other two counters. The case when f switches from HIGH to LOW indicating the end of frame is not being considered since it does not produce any additional time lag between the incoming data and the latched output. It simply indicates the duration for which the frame is in *ON* state. The only constraint that f should satisfy, given in (5), can now be expressed as,

$$2t_{pd,counter} + 2t_{pd,AND} + 2t_{pd,NOR} < T/2 \quad (10)$$

where the symbol meanings remain same as given earlier.

The propagation delay of the different circuit components is given in Table II and total delay for the two proposed receivers along with the standard one is given in Table III. The proposed and generalized circuits have better delay profile with respect to the standard circuit typically by 43.35 ns and 44.85 ns respectively.

In the context of generalization of the circuit for a 16-bit fixed point DSP, usually the latching of data would occur much before f goes LOW. In the standard circuit, latching can be done by directly using the monoshot's falling edge. The monoshot pulse width can be set equal to the duration specified in (1) for different number of data bits being transferred by choosing the appropriate values of the external capacitor and resistor. However, it might be difficult to procure resistors and capacitances to satisfy (1) for different data rates. Additionally, the pulse width of the monoshot can vary with external conditions such as temperature, etc. and make it unreliable at high data rates when the bound of (1) becomes tighter. On the contrary, the duration of f in the proposed circuit is realized from the receiver clock itself and thus is independent of the data rate. Implementing a similar latching circuit in the conventional receiver would also produce an additional and significant amount of propagation delay.

V. CONCLUSIONS

An efficient design for asynchronous communication has been presented which implements the receiver structure using digital components only. The design has been able to perform an accurate recovery of the incoming data, as expected theoretically. A comparison with the standard receiver circuit has indicated a possible improvement in reliabilty as well as the timing efficiency, at the cost of a very marginal increase in receiver size. It has also been observed that the proposed scheme achieves a considerable speed-up over any conventional circuit with respect to delay profile.

The design presented here applies, in general, to any application using asynchronous communication and is in no way limited to processors alone. In the future, it might be possible to control the variable part of the receiver circuit directly through software, thereby making it possible to tailor the design to the specific application at hand with minimal effort. Active effort by the authors is now underway to carry out the same on a 16-bit fixed point processor with any general n data bits.

REFERENCES

[1] G. Fairhurst (2001, January). Asynchronous Communication [Online]. Available: http://www.erg.abdn.ac.uk/users/gorry/course/phy-pages/async.html.
[2] D. Comer, *Computer Networks and Internets with Internet Applications, 4th ed.* Prentice Hall, Upper Saddle River, NJ, 2004.
[3] IBM technical guide (1998, April). AIX Versions 3.2 and 4 Asynchronous Communications Guide [Online]. Available: http://davinci01.man.ac.uk/aix433/aixasync/asycomgd/toc.htm.
[4] National Instruments Technical Report (2004, July). Serial Port Communication [Online]. Available: http://cnx.rice.edu/content/m12293/latest/.
[5] National Semiconductor Datasheet (1995, June). PC16550D Universal Asynchronous Receiver/Transmitter with FIFO's [Online]. Available: http://www.national.com/ds.cgi/PC/PC16550D.pdf.
[6] Lattice Semiconductor Corporation Datasheet (2001, June). Universal Asynchronous Transmitter/Receiver [Online]. Available: http://www.fpga.com.cn/freeip/uart_an_lattice.pdf.