

SPECIFICATION

NRZ/biphase microcomputer serial communication logic

- 5 This invention relates generally to the field of data processing and, more particularly, to the serial I/O communication logic of a single-chip microcomputer. 5
- Microcomputers are sophisticated, general purpose logic devices which can be programmed to perform a wide variety of useful control functions in industrial and communications equipment, large-scale and medium-scale computer peripheral and terminal hardware, automobiles and 10 other transportation media, amusement and educational devices, and the like. 10
- Microcomputers are being used increasingly to control serial data communications between data processing terminals, modems, and processors. Serial data communications between terminals or modems primarily utilizes NRZ (non-return-to-zero) encoding, whereas serial communications between processors primarily utilizes biphase encoding. Particularly, as distributed processing systems become increasingly important, such as in the automotive environment, it also becomes increasingly important to provide a microcomputer with the capability of 15 handling the highly reliable biphase format, which can tolerate a much greater mismatch between transmitter and receiver clocks. 15
- As the unit cost of microcomputers decreases, these devices are becoming more and more attractive for controlling serial data communications of all types. It would be highly desirable to 20 provide a microcomputer with the capability of utilizing both NRZ and biphase protocols. However, in designing a microcomputer which can be sold at a competitive price it is mandatory that the on-chip logic be minimized to the greatest extent possible, since yields are inversely proportional to the chip size. There is a need therefore for a microcomputer serial I/O communication logic which can accommodate both NRZ and biphase encoding, yet which 25 requires a minimum of logic to be implemented. 25
- In a single-chip microcomputer having serial I/O communication logic it is desirable to minimize the number of unnecessary interruptions of the on-board microprocessor or CPU. By reducing unnecessary interrupts of the CPU, the efficiency of the microcomputer and the 30 throughput of the CPU are increased. It is becoming increasingly common to connect several microcomputers together in what is known as a distributed processor or multi-processor configuration, wherein the microcomputers may share a common serial I/O communication line. When serial communications are being conducted between a master CPU and a slave CPU, 35 certain digital information transmitted over the shared serial I/O line may not be of interest to all of the CPUs. It is desirable to provide a means for a CPU to selectively ignore the contents of a message which is not specifically addressed to it and to "wake-up" when the serial I/O line again becomes free, in order that it may receive any future digital communications of interest to it. 35
- In a microcomputer having on-board serial I/O data communication logic, and in fact in any 40 serial data recovery logic such as is commonly used with controllers associated with various magnetic media devices, it is also desirable to have the capability of decoding Manchester-encoded data accurately and at any one of several data rates. The known prior art serial data recovery logic utilizes single-shot multivibrators to determine the "windows" for the transition intervals in Manchester (also known as FM or biphase) data streams. The single-shot multivibrators are subject to manufacturers' tolerances, drift, and other problems. More importantly, they 45 restrict decoding to a single frequency. In order to avoid the tolerance problems associated with single-shot multivibrators, and to provide automatic accommodation of variable data rates, a totally digital system of serial data recovery is needed. 45
- It is an object of the present invention to provide improved serial I/O communication logic in a single-chip microcomputer. 50
- It is another object of the present invention to provide in a single-chip microcomputer serial I/O communication logic capable of handling both NRZ and biphase formats.
- According to the present invention there is provided a microcomputer including a processor, serial I/O communication logic and bus means for coupling said processor to said serial I/O 55 communication logic, said serial I/O communication logic comprising control register means coupled to said processor, for storing transmit mode and receive mode signals provided by said processor, transmitter means coupled to said processor, for receiving and storing data provided periodically by said processor, and for serially transmitting said stored data via a serial output line in either NRZ or biphase format as selected by said transmit mode signal stored in said 60 control register means, and receiver means coupled to said processor, for serially receiving data via a serial input line in either NRZ or biphase format as selected by said receive mode signal stored in said control register means, storing said received data, and periodically providing said stored data to said processor. 60
- The invention will now be described by way of example only with particular reference to the 65 accompanying drawings in which: 65

Figure 1 shows a block diagram of a single-chip microcomputer which embodies the present invention;

Figure 2 illustrates the pin-out arrangement of the single-chip microcomputer which embodies the present invention;

5 Figure 3 shows a general block diagram of the serial I/O communication logic embodying the present invention; 5

Figure 4 shows a detailed block diagram of the serial I/O communication logic of the present invention;

Figure 5 illustrates the software-addressable registers of the serial I/O communication logic;

10 Figure 6 illustrates a multi-processor configuration illustrating one embodiment of the present invention; 10

Figure 7A illustrates NRZ-encoded data;

Figure 7B illustrates biphase-encoded data;

15 Figures 8A-8J constitute a detailed logic diagram for the serial I/O communication logic of the present invention; 15

Figure 9 shows the interconnection of the individual drawings of Figs. 8A-8J;

Figure 10 shows an additional detailed logic diagram for generating signals utilized within the logic circuitry illustrated in Figs. 8A-8J;

20 Figure 11 is a schematic representation of the generation of the $\phi 1$ and $\phi 2$ internal microcomputer clocks from the master E clock input; 20

Figure 12 is a block diagram illustrating the clocking arrangement in a preferred embodiment of the present invention;

Figure 13 illustrates the relationship between the internal clocks $\phi 1$ and $\phi 2$ and the master clock E;

25 Figure 14 represents a flowchart of the reset operation of the serial I/O communication logic; 25

Figure 15 represents a flowchart of the serial I/O communication logic operating in the half-duplex transmission mode;

Figure 16 represents a flowchart of the serial I/O communication logic operating in the half-duplex reception mode and

30 Figure 17-22 represent detailed circuit schematics of the flip-flops and latches utilized in the logic circuitry of Figs. 8A-8J. 30

With reference to Fig. 1, a block diagram of a single-chip microcomputer embodying the present invention is shown. It will be understood that the present invention may be implemented on other than a single-chip microcomputer. A preferred embodiment of the present invention is embodied in the microcomputer shown in Fig. 1, the Motorola MC6801 microcom- 35

puter. The microcomputer shown in Fig. 1 comprises a central processor (CPU) 1, a random access memory (RAM) 2, a read-only-memory (ROM) 3, a timer circuit 4, and a serial I/O portion 5 as major functional blocks. The microcomputer shown in Fig. 1 also comprises a multiplexer (MUX) 40 6, an internal address bus 7, an internal data bus 8, and four I/O ports 11-14. CPU 1 receives as inputs thereto a master clock E over line 20, mode control signals CC1 and CC2 over lines 21 and 22, respectively a RESET signal over line 23, an interrupt request signal (IRQI) over line 24, a non-maskable interrupt signal (NMI) over line 25, a power supply signal (V_{cc}) over line 26, and a ground signal (V_{ss}) over line 27. The RAM 2 receives a V_{cc} standby supply over line 28 to 45 preserve data in the RAM in the event of a power failure. 45

Ports 1, 3, and 4 are 8-bit ports, and port 2 is a 5-bit port. The eight lines, P10-P17, into port 1 are dedicated solely to parallel I/O operations. The lines into ports 2-4 may be configured in different ways depending upon one of three possible modes of operating the microcomputer shown in Fig. 1. Port 2 has five lines P20-P24. Port 3 has eight lines P30-P37 50 and two control lines, SC1 and SC2. Control lines SC1 and SC2 into port 3 serve as input and 50 output strobes. Port 4 has eight lines P40-P47. In the single chip mode all I/O lines of ports 2-4 can be programmed to serve either as inputs or outputs, in accordance with a data direction register associated with each port. In the expanded non-multiplex mode lines P30-P37 of port 3 function as the external data bus (D0-D7). In the expanded non-multiplex mode, lines 55 P40-P47 of port 4 serve as the low-order address lines (A0-A7). If any of the eight address 55 lines are not needed in this mode the remaining lines may be configured as I/O. In the expanded multiplex mode lines P30-P37 of port 3 serve both as the external data bus (D0-D7) and as the low-order address bus (A0-A7). In this mode, lines P40-P47 of port 4 serve as the high-order address lines (A8-A15). If any of the eight high-order address lines are not needed in 60 this mode the remaining lines may be configured as I/O. 60

In all three modes lines P20-P24 of port 2 can be configured either as input or output lines by a data direction register associated with this port. Port 2 also provides access to the serial I/O communications capability and the programmable timer capability of the microcomputer shown in Fig. 1, in the manner to be discussed hereinafter.

65 The foregoing description of the microcomputer of Fig. 1, including the various operational 65

modes, has been given as general back ground only, and the operation of the serial I/O features of the present invention may be completely understood independently of such description. The serial I/O operations of the present invention are conducted through pins P22, P23, and P24 only, and the function of these pins is unaffected by the particular mode of microcomputer operation as described above.

Fig. 2 provides a schematic representation 15 of the pin-out arrangement of the microcomputer of Fig. 1.

The serial I/O portion of the microcomputer of Fig. 1 is capable of conducting serial communications in full duplex or half-duplex at a variety of clock rates. In addition, the serial I/O logic is capable of handling serial operations in either of two formats: (1) standard mark/space (NRZ) which is typically used between terminals or modems, and (2) self-clocking biphasic which is primarily used for communications between processors. The NRZ format is illustrated in Fig. 7A, and the biphasic format is illustrated in Fig. 7B. Both formats begin with a START bit (always 0) and end with a STOP bit (always 1). The NRZ format provides a signal level corresponding to the bit value at each bit time. The level is then sampled at the middle of the bit time on decoding. The example shown in Fig. 7A represents the NRZ-encoded binary number 01001101. It will be noted that the number is encoded beginning with the least significant bit (LSB). At bit time zero the signal level is high, representing a binary "1". At bit time "1", the signal level is low, representing a binary "0", and so forth until the STOP bit signal indicates the conclusion of one byte. The NRZ format can tolerate only approximately a 3.75% mismatch between the transmitter and receiver clocks for correct operation.

Fig. 7B illustrates the encoding of the binary number 01001101 in the biphasic format. The biphasic format provides a transition in signal level at every bit time and a transition at the centre of every bit with the value 1. The biphasic format, also known as biphasic-M, FM, F/2F, and Manchester format, can tolerate a difference of approximately 25% between the transmitter and receiver clocks for correct operation. It should be noted that an idle line in NRZ format is represented by a constant mark (1) on the line. In biphasic format, an idle line will be toggling at every half-bit time.

It is particularly advantageous to provide the capability of transmitting and receiving communications in biphasic format on a microcomputer. Since the biphasic format can tolerate a much greater clocking mismatch in processor-to-processor communications, it is particularly useful in multi-processor arrangements, such as occur, for example, in automotive environments.

With reference to Fig. 3, a general block diagram of the serial I/O communication logic embodying the present invention is shown. The serial I/O circuitry comprises a transmitter portion including Transmit Data Register (TDR) 37, Transmit Shift Register (TDS) 38, a flip-flop TDST 39, TB Register 40, and TC Counter And Control Logic 41. The serial I/O logic also includes a receiver section including flip-flop RSD 33, Receive Shift Register (RDS) 34, flip-flop RDST 35, RB Register 31, RC Counter And Control Logic 32, and Receive Data Register (RDR) 36. Both the transmitter and receiver sections communicate with the microcomputer via the Peripheral Data Bus 30 and with external devices via pins P22, P23, and P24 of port 2.

In data transmission, an 8-bit data word to be transmitted is written to the Transmitter Data Register TDR 37 from Peripheral Data Bus 30. The eight bits are then transferred in parallel from TDR 37 to the Transmit Shift Register TDS 38, which shifts the bits out to Pin 4 of Port 2 via flip-flop TDST 39. TDST 39 formats the data stream and adds the start and stop bits to each transmitted word.

In data reception, an in-coming data stream is input through Pin 3 of Port 2 and passes through flip-flop RDS 33 to the Receive Shift Register 34 where it is shifted until the start bit is in flip-flop RDST 35, the eight data bits are in the Receive Shift Register 34, and the stop bit is in flip-flop RSD 33. If no framing error or overrun condition exists, the eight data bits are transferred in parallel from the Receive Shift Register 34 to the Receive Data Register RDR 36. The word is then made available to the microcomputer by writing RDR 36 to Peripheral Data Bus 30.

The serial I/O logic of the present invention is programmable with respect to several important features. The data communications format may be programmed to utilize either NRZ or biphasic encoding. The clock may be programmed to utilize either internal or external clock signals. The baud rate is programmable to one of four per given internal or external clock frequency. The wake-up capability may be enabled or disabled. Interrupt request may either be enabled or masked individually regarding the Transmit Data Register 37 and Receive Data Register 36. The clock output to pin 2 of port 2 may either be enabled or disabled. Finally, pins 3 and 4 of port 2 may either be dedicated or not dedicated to serial I/O operations individually with the transmitter and receiver portions.

Table I gives the four possible data communication rates for each of the microprocessor ϕ 2 clock frequencies or the external clock frequencies. The various data rates and clock frequencies will be discussed in further detail within the Detailed Description.

The data link used with the serial I/O logic of the present invention may be either half-duplex

or full duplex and may either use a separate clock or not. Both biphasic and NRZ formats may be used without a separate clock line, but only the NRZ format may be used with a separate clock line (either input or output). Also it is possible to transmit only a clock to a remote device. The various data link configurations are summarized in Table II.

The present invention provides a wake-up capability in multi-processor configurations utilizing the microcomputer shown in Fig. 1. The wake-up capability is intended to increase processor throughput by permitting non-interested processors on a common line to ignore the remainder of a message if the destination address is different from that of the particular processor. If the processor determines that the remainder of the message is not intended for it, it sets a wake-up bit (WU) in the Control and Status Register. With the wake-up bit set, the receiver portion continues to monitor messages without being interrupted. When eleven consecutive "1's" are received by the receiver portion, the receiver portion clears the wake-up bit WU to "wake-up" the interrupt processing for the next message. The eleven consecutive "1's" indicate the idle state on the transmission line. The wake-up capability will be described in greater detail below.

With reference now to Fig. 4, a detailed block diagram of the serial I/O logic of the present invention is shown. Data may be taken off Peripheral Data Bus 30 via an 8-bit bus 47 to Transmit Data Register 37, from which it is entered into Transmit Shift Register 38. From Transmit Shift Register 38 the data is shifted out serially through a gate 42 enabled by the Transmit Enable (TE) pulse. Pin 4 of port 2 is the serial transmit line.

Serial data is received over pin 3 of port 2 through gate 43 enabled by Receiver Enable (RE) signal. The serial input data is shifted into Receive Shift Register 34, and is then transferred in parallel into Receive Data Register 36. The contents of Receive Data Register 36 is sent over 8-bit bus 44 to the Peripheral Data Bus 30, from which it may be utilized by other portions of the microcomputer.

The serial I/O logic of the present invention utilizes four software-addressable registers, which are shown in Fig. 5 in the form of Control And Status Register 46, Rate And Mode Control Register 45, Receive Data Register 36, and Transmit Data Register 37.

Control and Status Register 46 consists of an 8-bit register of which all 8 bits may be read while only bits 0-4 may be written. The register is initialized to

S20

on RESET. The bits in the register are defined as follows:

7	6	5	4	3	2	1	0
---	---	---	---	---	---	---	---

RDRF	ORFE	TDRE	RIE	RE	TIE	TE	WU	ADDR: S0011
------	------	------	-----	----	-----	----	----	-------------

Bit 0 WU "Wake-up" on next message. When set, this bit enables the wake-up function; it is cleared by hardware on receipt of eleven consecutive 1's. WU will not set if the line is idle.

Bit 1 TE Transmit Enable. When set, this bit produces a preamble of nine consecutive 1's and enables gating of the transmitter output to port 2, pin 4; it changes the DDR value for port 2, pin 4 to a "1".

Bit 2 TIE Transmit Interrupt Enable. When set, it permits an $\overline{\text{IRQ2}}$ interrupt to occur if TDRE is set. When clear, the TDRE value is masked from the bus.

Bit 3 RE Receiver Enable. When set, it gates port 2, pin 3 to the input of the receiver. It changes the DDR value for port 2, pin 3 to a zero.

Bit 4 RIE Receiver Interrupt Enable. When set, it permits an $\overline{\text{IRQ2}}$ interrupt to occur when either RDRF or ORFE is set. When clear, the interrupt is masked.

Bit 5 TDRE Transmit Data Register Empty. It is set by hardware when a transfer is made from the Transmit Data Register to the Transmit Shift Register. This transfer is synchronized with the bit rate clock. The TDRE bit is cleared by reading the Status Register, then writing a new byte into the Transmit Data Register. No data will be transmitted if TDRE is not cleared. TDRE is initialized to 1 by RESET.

Bit 6 ORFE Over-Run-Framing Error. It is set by hardware when an overrun or framing error occurs (receiver only). An overrun is defined as a new byte transferred to the Receive Data Register with the RDRF flag set. A framing error has occurred when the byte boundaries in the bit stream are not synchronized to the bit counter. An overrun may be distinguished from a framing error by the corresponding value of RDRF. If $\text{RDRF} = \text{ORFE} = 1$, then an overrun has occurred. If $\text{RDRF} = 0$ and $\text{ORFE} = 1$, a framing error has been detected. The ORFE bit is

cleared by first reading the Status Register, then reading the Receive Data Register, or by a RESET signal.

Bit 7 RDRF Receiver Data Register Full. It is set by hardware when a transfer from the Receive Shift Register to the Receive Data Register is made. The RDRF bit is cleared by first reading the Status Register, then reading the Receive Data Register, or by a RESET signal.

Rate and Mode Control Register 45, controls the following serial I/O variables: baud rate, format, clocking source, and Port 2 Pin 2 configuration.

The register consists of four bits all of which are write-only and are cleared on RESET. The four bits in the register may be considered as a pair of 2-bit fields. The two low-order bits control the bit rate for internal clocking, and the remaining two bits control the format and clock select logic. The register definition is as follows:

7	6	5	4	3	2	1	0	
X	X	X	X	CC1	CC0	S1	S0	ADDR: S0010

Bit 0)
Bit 1) S1, S0 Speed Select. These bits select the baud rate for the internal clock. The four rates which may be selected are a function of the CPU $\phi 2$ clock frequency. The following table lists the available baud rates. If external clocking is selected ($CC1 = CC\phi = 1$), the speed select bits are ignored.

TABLE 1

S1, S0	XTAL $\phi 2 \div$	4.0 MHZ 1.0	4.9152 MHZ 1.2288	2.5476 MHZ 0.6144
00	16	62.5 KBaud	76.8 KBaud	38.4 KBaud
01	128	7,812.5 Baud	9,600 Baud	4,800 Baud
10	1024	976.6 Baud	1,200 Baud	600 Baud
11	4096	244.1 Baud	300 Baud	150 Baud

Bit 2)
Bit 3) CC1, CC0 Clock Control and Format Select. This 2-bit field controls the clock and format select logic. The following table defines the bit field. If $CC1 = 0$, the DDR value for port 2 pin 2 is not affected. If $CC1 = 1$, the DDR value for port 2 pin 2 is changed to the complement of $CC\phi$.

CC1,	FORMAT	CLOCK	PORT 2		
CC0		SOURCE	BIT 2	BIT 3 **	BIT 4 **
00	BI-PHASE	INTERNAL	NOT USED	SERIAL INPUT	SERIAL OUTPUT
01	NRZ	INTERNAL	NOT USED	SERIAL INPUT	SERIAL OUTPUT
10	NRZ	INTERNAL	OUTPUT*	SERIAL INPUT	SERIAL OUTPUT
11	NRZ	EXTERNAL	INPUT	SERIAL INPUT	SERIAL OUTPUT

* Clock output is available regardless of values for bits RE and TE.

** Bit 3 is used for serial input if $RE = "1"$ in Control and Status Register 46. Bit 4 is used for serial output if $TE = "1"$ in Control and Status Register 46.

RB Register 31 is an 8-bit delay line which is used to extract the imbedded clock from the input data stream and to establish receiver synchronization. Reference to the appropriate Boolean equations below and the detailed logic diagram of Figs. 8A-8J shows that in the NRZ format the zero start bit is clocked into RB Register 31 and that thereafter the input toggles at the bit rate so long as RE remains set. Approximately two RT clock cycles are required (setting RE turns on the RB input) before RB Register 31 can respond to a "zero" start bit. As the "zero" propagates along, RB synchronization is established when RSE sets at the end of the third RT clock. The first RSD clock ($RBD \oplus RBE$) is also generated at the end of the third RT clock and the first R θ clock is generated at the end of the fifth RT clock. The RSD ϕ and R θ

clocks continue to be generated at $3/8$ bit time and $5/8$ bit time so long as RB register 31 operates.

The bit counter RC goes to "one" at the end of $R\theta$ and increments on each succeeding $R\theta$. The start bit is clocked into RSD by $RBD + RBE$. There is normal loss of synchronization after data transfer, with synchronization regained during the following start bit.

The effect of resetting RE is that RB stops toggling; there is a loss of synchronization, $R\theta$, and $RBD + RBE$ signals; the bit counter RC resets; and RSD, the RDS shift register 34, and RDST all stop. In the biphase format the entire data stream is clocked into the register.

After the microcomputer of Fig. 1 comes out of reset and before RE is set the RB Register begins filling up with "one", and hence no bit-rate clocks are generated, and the receiver is not functioning. There are three things necessary to establish reception: (a) RE must be set in order for the serial input data to be input to the RB Register, (b) the serial input data must be the idle line condition, namely all "ones", in order for the RB Register to begin generating the bit rate clocks, and (c) the first start bit in the serial input data stream cannot occur until the line has been idle for at least 1 bit time (otherwise synchronization cannot be established.) Therefore, the minimum time the line must remain idle after RE is set is one bit time.

Following the setting of RE, synchronization is established when RSE is set, which occurs when the first "zero" propagates along the RB Register, the first $R\theta$ is generated by the leading edge of RSH at $1/4$ bit time, and RSE is set by the leading edge of RLG at $3/4$ bit time. The separation flip-flop RSD is driven by RSH and RLG and generates a signal SEP which is zero for "zero" input data and 5 volts for $1/2$ bit time if data input is "one". The timing of SEP is such that as it is clocked into the Receive Shift Register 34 by $R\theta$ it is finally totally converted to the NRZ format.

Setting RSE starts the bit counter RC. Since RSE sets $1/2$ bit time later and $R\theta$ occurs $1/8$ bit time later for biphase than for NRZ, the location of each bit counter state for the biphase format lags the corresponding bit counter state for the NRZ format by $1-1/8$ bit times. Thus the data transfer from the Receive Shift Register RDS 34 to the Receive Data Register RDR 36 which occurs at the end of the RC 9-count in both formats, occurs $1-1/8$ bit times later in biphase than in NRZ, i.e. near the end of the stop bit for NRZ and near the end of the following start bit for biphase.

The effect of resetting RE is to cause RB Register 31 to stop toggling (causing loss of synchronization and bit clock $R\theta$), the resetting of bit counter RC, and stopping of RDS 34 together with RSD and RDST. In biphase reception all internal functions are clocked by signals derived from the incoming data stream. Consequently, as long as RSH and RLG are both generated the separation flip-flop SEP 101 (Fig. 8J) will operate properly and generate $R\theta$.

Biphase synchronization is established when RSE sets on the first start-bit following at least one idle bit. Eleven idle bits will reset RSE, but synchronization is not lost so long as data words are separated by no more than ten idle bits.

In NRZ reception internal operation is initiated by the start bit and terminated by the stop bit of each data word. All operations in between are clocked at the independent internal clock rate. Operation begins when RSE 102 sets at the middle of the "zero" start bit following an idle condition of at least two bit times. RSE 102 then resets immediately following data transfer from the RDS register 34 to the Receiver Data Register 36.

For the case of a data rate faster than $R\theta$, the start bit is clocked into RB Register 31 which then toggles for the next eight bit times thus generating $R\theta$ at the internal clock rate of the serial I/O controller. When the data rate is faster than $R\theta$ the stop bit and the following start bit are successfully clocked into RB Register 31 earlier than nominal. The corresponding periods of the $R\theta$ and $RBD + RBE$ clocks are consequently shortened by one or two RT times. At this highest data rate (for proper operation) RSE resets and data transfers one RT time sooner and synchronization during the following start bit is established 2 RT times sooner.

For the case of a data rate slower than $R\theta$, operation for the start bit and first seven data bits is identical to operation at high data rates. Operation during the eighth data bit and stop bit differs according to whether the last data bit is a "zero" or a "one".

If the last data bit is "zero", the generation of $R\theta$, the resetting of RSE, and the transfer of data are all delayed until the middle of the stop bit. The following start bit then re-initiates the sequence of operations necessary to receive one data word.

If the last data bit is "one", the generation of $R\theta$, the resetting of RSE, and the data transfer all occur at their nominal positions and nothing happens during the stop bit. The following start bit then re-initiates the receive cycle. Thus for slow data a "one" in the last data bit position acts similar to a stop bit, and everything holds during the actual stop bit until the start bit re-initiates the cycle.

TB Register 40 is a four-bit counter that counts continuously from 1 to 8 at the RT rate beginning at the end of RESET. The external clock TEX is the 50% duty cycle waveform available at TBD and hence is in phase with the transmitted data stream. TSH is a 2X clock occurring at the end of each bit. The only difference between the biphase and NRZ formats is in

the operation of the formatting flip-flop TDST.

With regard to transmitter operation note that the idle line condition is established after the setting of TE and that data transfer occurs at the middle of the start bit. When TDE is not serviced before the middle of the stop bit it remains set and causes the TC counter to hold at the 9-count, thus inhibiting the data transfer. When TDE is reset, transmission resumes with the next start bit.

TSH.T θ clocks in data from TSH ϕ to TDST at each bit boundary for either format, and, during biphase only, TSH.T θ clocks in a toggle for TDS ϕ = 1, or if the line is idle, or during each stop bit.

- 10 In either the biphase or the NRZ format the transmitter works in the NRZ format except for the formatting flip-flop, TDST, which is essentially the transmitter output stage. TDST is clocked by RSH (which occurs at twice the R θ rate) and every other RSH coincides with R θ .

- The transmitter outputs ones as long as TSE is not set. TSE sets when the TC counter reaches the 10-count and remains set as long as TE = 1. Accordingly a preamble of ten ones is transmitted when TE is first turned on. At any time after the initial preamble of ten ones if a word has not been written into the Transmit Data Register 37, the TC counter holds at the 9-count, and TDST continually outputs ones.

- The serial I/O controller communicates with the CPU via the IRQ2 interrupts. If receiver interrupts have been enabled by RIE and by resetting WU, an IRQ2 interrupt is generated whenever an overflow or a framing error sets the OR bit or whenever transfer of a valid word from the Receive Shift Register 34 to the Receiver Data Register 36 sets the RDF bit. If transmitter interrupts have been enabled by setting TIE, an IRQ2 interrupt is generated whenever data transfer from the Transmitter Data Register 37 to the Transmit Shift Register 38 sets the TDE bit.

- 25 Use of the WU, the "wake-up" bit, is determined by the data link protocol. If after examination of the first part of a message the CPU ascertains that it has no further interest in the remainder of the message and sets WU, all further interrupts are inhibited until the line goes idle. The WU bit is reset by hardware whenever eleven contiguous "ones" are received, or it can be reset by software.

- 30 Each of the three interrupt bits RDF, OR, and TDE has a follower bit that sets whenever the Control and Status Register 46 is read while its associated bit is also set. The RDF and OR interrupt bits and their followers are reset whenever the Receive Data Register 36 is read while its associated follower bit is also set. The TDE interrupt bit and its follower are reset whenever the Transmit Data Register 37 is written while its follower is set.

- 35 Regarding the control of port 2, at the end of RESET the serial I/O controller overrides CPU control of from one to three pins of port 2 depending upon the operating mode defined by control bits RE, TE, CC1, and CC ϕ . RE and TE determine the duplex configuration while CC1 and CC ϕ determine the external clock configuration.

- Pin 2 of port 2 may be used either to input an externally generated clock or to output a clock generated by the I/O controller. When CC1 = "0" this pin is controlled by the CPU but that when CC1 = "1" this pin is used to either input an external clock or output an internal clock.

- When CC1 = "0" the I/O controller does not use pin 2 and accordingly the CC1.DDR2 coupler allows the CPU control signal WP2.DDR2 to determine the state of the pin 2 data direction latch comprising transmission gates 86 and 87, inverter 88, and NOR gates 85 and 89. When CC1 = "1" the CC ϕ control bit determines the state of the data direction latch.

- The pin data output register comprising NOR gates 90 and 91, inverters 92-94 and transmission gate 95 is handling CPU data when CC1 = "0", and is outputting TEX, the internally generated clock, when CC1.CC ϕ = "1". When CC1.CC ϕ = "1" pin 2 is inputting an externally generated clock and the data output register is in a "don't care" state since the output drive is tri-stated (i.e. in a very high impedance state).

The data input buffer 96 inputs the external signal on pin 2 under all operating conditions. However, it is only passed by the mode control logic when it is actually an external clock, i.e., when CC1.CC ϕ = "1".

- Pin 3 of port 2 is used to input serial data when the I/O controller is in the full-duplex or half-duplex receive modes. As long as the control bit RE = "1" the pin 3 data direction latch DDR23 comprising NOR gate 98, inverters 99 and 106, and transmission gates 97 and 100 holds the output drive in the tri-state mode. The CPU controls DDR23 when RE = "0". The data input buffer 104 inputs the external signal at pin 3 under all conditions.

- Pin 4 of port 2 is used to output serial data when the I/O controller is in the full-duplex or the half-duplex transmit modes. As long as TE = "1" the pin 4 data direction latch comprising inverter 107, NOR gates 108 and transmission gate 105 is held in the output state and the serial data, TDST, is input to the pin 4 data output register comprising NOR gate 110, inverters 113 and 114, and transmission gates 111, 112 and 115. The CPU is in control when TE = "0".

- 65 The overall clocking management is depicted in Fig. 12. The four bits in the Rate And Mode

Control Register 45 (Figs. 4 and 5) determine the source and frequency of the high-speed clock RT (generated over line 110 in Fig. 8A). RT is a 50% duty cycle clock at exactly 8 times the transmitted data rate and approximately 8 times the received data rate.

The transmitter clocks are:

5 TSH—a pulse train at 2 times the data rate which is used to clock the formatting flip-flop TDST. 5

T θ —the bit rate clock which drives the bit counter TC.

TS θ —the Transmit Shift Register clock which is the same as T θ with every 10th pulse missing.

10 Tout—a 50% duty cycle clock at the data bit rate which is available for transmission to remote devices. 10

The receiver clocks are:

RSH—two pulses which are generated for every biphasic "one" in the received data stream. This clock does not exist in the NRZ format.

15 RLG—one pulse which is generated for every biphasic "zero" in the received data stream. In the NRZ format RLG is identical to R θ . 15

R θ —an embedded clock which is extracted from the received data stream and drives RC counter 32, Receive Shift Register 34, and the start-bit flip-flop RDST 35.

RBD $\oplus$ —The clock, at the same rate as R θ but

20 RBE nearer mid-bit, which drives the separation (or deformatting) flip-flop RDS 33. 20

Of the five high-speed clocks provided, four are tapped off of the microcomputer timer 4 and the fifth can be input from an external source. All of the clocks have a 50% duty cycle and are at 8 times the data rate. The four timer clocks are designated T ϕ , T3, T6, and T8 and correspond to ϕ 2 (the CPU clock) divided by 2, 16, 128, and 512 respectively.

25 Figs. 8A–8J constitute individual section of a preferred embodiment of the serial I/O logic of the present invention. The individual figures may be fitted together in the manner shown by Fig. 9 to form the complete serial I/O logic schematic of the Fig. 1 single-chip microcomputer. 25

Given the logic schematic of Figs. 8A–8J, one of ordinary skill in the art is enabled to implement the serial I/O logic of the present invention by using existing circuit technology, such as MOSFET (Metal Oxide Semiconductor Field Effect Transistor) technology. A detailed recitation of the interconnections between the logic and circuit elements shown in Figs. 8A–8J would only encumber the description. However, various of the structures shown and described with regard to the general block diagram of Fig. 3 will be highlighted in the detailed logic schematic of Figs. 8A–8J.

35 Figure 8A shows the 4-bit TB Register comprising flip-flops 201–204. 35

Fig. 8B shows the TC Counter and Control Logic comprising flip-flops 211–215.

Fig. 8C and 8D show the Rate And Mode Control Register comprising flip-flops 221–224.

Also shown in Figs. 8C and 8D is the Control And Status Register comprising flip-flops 231–238.

40 Fig. 8E shows the Receive Data Register comprising flip-flops 241–248 and the Receive Shift Register comprising flip-flops 251–258. 40

Fig. 8F shows the Transmit Data Register comprising flip-flops 261–268 and the Transmit Shift Register comprising flip-flops 271–278.

45 Fig. 8G shows the bonding pad 320 for I/O pin 2 of port 2, as well as V_{DD} pad 303, and V_{SS} pad 304. The V_{DD} and V_{SS} signals will be understood to be distributed throughout the logic shown on Figs. 8A–8J as required. 45

Fig. 8H shows the bonding pad 321 for I/O pin 3 of port 2 and bonding pad 322 for I/O pin 4 of port 2.

50 Fig. 8I shows RC Counter And Control Logic comprising flip-flops 281–285 and shows in addition the RSE flip-flop 102. 50

Fig. 8J shows the RB Register comprising flip-flops 291–298. Fig. 8J also shows the SEP flip-flop 101, the RSD flip-flop 111, and the TDST flip-flop 112.

Fig. 10 shows the RESET bonding pad 323 and associated circuitry for generating the RESET signal over line 300, the MODL signal over line 301 and the VRBIAS signal over line 302. It will be noted that the RESET signal is sometimes designated as the POR signal in the logic diagram of Figs. 8A–8J. 55

The lines of Figs. 8A–8J which connect other portions of the microcomputer (not shown) will now be discussed. In Fig. 8A a TMR signal is transmitted over line 400 and signals T ϕ , T6, T3, and T8 are transmitted over lines 401–404, respectively, to the microcomputer programmable timer 4 (Fig. 1) for purposes not associated with the object of the present invention. In Fig. 8D an interrupt control signal IRQS is transmitted over line 410 to the programmable timer. In Fig. 8F the individual lines of the Peripheral Data Bus 30, which bus transmits data to and from the serial I/O logic, are identified as PDB0–PDB7. In Fig. 8I signal PC2, which is transmitted over line 415, represents one of three program control bits in port 2, which program control bits are

65 used to program the mode of operation of the microcomputer in a manner unassociated with the 65

objects of the present invention. Also shown in Fig. 8I is a signal IRSB on line 416, which signal is merely a control signal generated by the CPU. Further shown in Fig. 8I are some of the internal microcomputer address lines identified as A1-A4 and the Read/Write control line identified as R/W.

5 Fig. 11 illustrates schematically a clock generator 420 for generating clock signals $\phi 2$ and $\phi 2$ 5 used for internal microcomputer operations. Signals $\phi 1$ and $\phi 2$ are generated from master clock E and the relationship between clocks E, $\phi 1$, and $\phi 2$ is shown by Fig. 13.

The serial I/O controller should be initialized by the microcomputer ROM software prior to operation. This sequence will normally consist of (1) writing the desired operation control bits to 10 the Rate And Mode Control Register 45 and (2) writing the desired operational control bits in the Control And Status Register 46. The Transmitter Enable (TE) and Receiver Enable (RE) bits 10 may be left set for dedicated operations.

Reference may be made to Fig. 14 for a flow chart of the reset operation. During reset the Transmit Data Register 37, the Receive Data Register 36, and all of the status and control bits 15 in both the Rate and Mode Control Register 45 and the Control and Status Register 46 are reset with the exception of control bit TDE which is set to indicate that the Transmit Data Register 37 is empty. In addition, the TB Register 40 is reset, the RB Register 31 is stopped (not reset), and the serial I/O pins of port 2 (P22, P23, and P24) are configured as input. Although an external 20 clock may be available from P22 the RB Register 31 is stopped because no clock is available from the Timer 4 during this time. 20

In addition to the possible external clock four more clock speeds (submultiples of $\phi 2$) become available from Timer 4 at the end of reset. Since $MCR = 0$, the highest-speed clock $T\theta$ ($= \phi 2 + 2$) is selected by the mode logic as the high-speed clock RT. RT clocks both the RB and TB Registers which run continuously during RESET. The input to the RB Register is held at 25 "one" until RE is set, while TB counts continuously. TSH and $T\theta$ are derived logically from the TB Register and are used to clock the formatting flip-flop TDST 112 (Fig. 8J) and the transmitter bit counter TC (TA through TCD in Fig. 8B), respectively. Every tenth $T\theta$ clock is suppressed by the transmitter bit counter logic to generate 25 $TS\theta$, the Transmit Shift Register clock.

30 It is thus seen that at the end of reset the serial I/O controller is functioning in a degenerate mode with the transmitter bit counter running at the highest speed. The formatting flip-flop TDST 112 is toggling in the biphase format, with the port 2 pins not properly configured. The receiver RB Register is also being clocked at the highest speed but fills up with "ones" and hence cannot generate the clock for the de-formatting flip-flop RSD 111 (Fig. 8J) which remains 35 in an arbitrary state.

Non-trivial operation begins with the writing of four control bits into the Rate and Mode Control Register 45. These bits (CC1, CC ϕ , S1, S ϕ) select the desired clock source and speed, determine the data format, and configure the clock pin (P22) as required. Next, five control bits are written into the Control and Status Register to complete the mode specification. Two of 40 these bits (TIE and RIE) determine whether or not interrupts are masked. Two other bits (RE and TE) determine whether operation is half-duplex (either transmitting or receiving) or full duplex. The fifth bit (WU) is used for the Wake-up mode which will be explained under Receive Operation.

The transmit operation is enabled by the TE bit in the Control and Status Register 46. This 45 bit, when set, gates the output of the serial Transmit Shift Register 38 to port 2 pin 4 and takes precedence over the Data Direction Register value for port 2. 45

Following a RESET, both the Rate and Mode Control Register 45 and the Control and Status Register 46 should be configured for desired operation. Setting the TE bit during such procedure initiates the serial output by first transmitting a nine-bit preamble string of 1's.

50 Following the preamble, internal synchronization is established and the transmitter section is ready for operation. 50.

At this point one of two situations exist: (1) if the Transmit Data Register 37 is empty (TDRE = 1), a continuous string of ones will be sent indicating an idle line, or (2) if data has been loaded into the Transmit Data Register 37 (TDRE = 0), the word is transferred to the 55 Transmit Shift Register 38 and transmission of the data word will begin. 55

During the transfer itself, the 0 start bit is first transmitted. Then the 8 data bits (beginning with bit 0) followed by the stop bit are transmitted. When the Transmit Data Register 37 has been emptied, the hardware sets the TDRE flag bit.

If the microcomputer CPU 1 fails to respond to the flag within the proper time (TDRE is still 60 set when the next normal transfer from the Transmit Data Register 37 to the Transmit Shift Register should occur), then a 1 will be sent (instead of a 0) at the "Start" bit time, followed by more 1's until more data is supplied to the Transmit Data Register 37. No 0's will be sent while TDRE remains a 1. 60

The biphase mode operates as described above except that the serial output toggles each bit 65 time, and also on 1/2 bit times when a 1 is sent. 65

Reference may be made to the flow chart of Fig. 15 for a summary of the operation in the half-duplex transmission mode. When TE is set the transmit pin P24 is configured to output by holding DDR24 at zero. As long as TE is set CPU operation of both DDR24 and P24 is inhibited.

- 5 Setting TE also releases the transmitter bit counter (TC) from reset. TC counts continuously during TE = 1 at the T₀ clock rate, but will hold at the 9-count as long as status bit TDE remains set. Transmission synchronization is established when TSE sets during the 9-count. When the interrupt caused by TDE = 1 is serviced, a new data word is loaded into the Transmit Data Register 37 and TDE is reset. Resetting TDE allows TC to resume counting. 5
- 10 Transmission begins at the middle of the TC 10-count when TSH 426 (Fig. 8A) clocks the start bit into the formatting flip-flop TDST 112 (Fig. 8J). The data word is transferred from the Transmit Data Register 37 to the Transmit Shift Register 38 at the end of the 10-count. If the data is to be transmitted in the biphase format the TDST toggle that would normally have occurred at the end of the TC 10-count is inhibited, thus establishing the transmitted data bit boundaries at the mid-points of each TC counter state. In the NRZ format the transmitted bit boundaries are also established at the mid-points of each TC counter state, but here the controlling factor is that TDST can only change state on the trailing edge of the next TSH clock which occurs after TDS₀ changes state, i.e. the TSH clock which occurs between TSD₀ clocks. 10
- 15 The receive operation is enabled by first configuring the Rate and Mode Control Register 45 and then writing a 1 to the RE bit in the Control and Status Register 46. In the NRZ format, the receiver is immediately ready to accept data. In biphase format, however, it is necessary to present an idle line (toggling at half-bit times) to the receiver for at least one bit time to permit synchronization to occur. 15
- 20 The receiver bit interval is divided into 8 sub-intervals for internal synchronization. In the standard, non-biphase mode, the received bit stream is synchronized to the leading edge of the first 0 (space) encountered. 20
- 25 The approximate centre of each bit time is strobed during the next 10 bits. If the tenth bit is not a 1 (stop bit), a framing error is assumed, and bit ORFE is set. Note that RDRF is not set for a framing error. If the tenth bit is a 1, the data is transferred to the Receive Data Register 36, and interrupt flag RDRF is set. If RDRF is still set at the next tenth bit time, ORFE will be set, indicating an over-run has occurred. When the CPU 1 responds to either flag (RDRF or ORFE) by reading the Control and Status Register 46 followed by reading the Receive Data Register 36, RDRF (and ORFE) will be cleared. 25
- 30 In the biphase mode, it must be determined whether a transition interval is more or less than 6 of the 8 sub-intervals. Pairs of short intervals are defined as 1's, while long intervals are defined as 0's. If RE is set during a string of 1's, the receiver cannot distinguish between bit and mid-bit boundaries. Proper bit synchronization occurs the first long (zero) transition interval. Synchronization to byte boundaries occurs when eight 1's followed by a zero are received. 30
- 35 Reference may be made to Fig. 16 for a summary of the operation in the receive half-duplex mode. When RE is set the receive data pin, P23, is configured to input by holding DDR23 at "1". As long as RE remains set CPU operation of both DDR23 and P23 is inhibited. Setting RE also allows the RB Register 31 to start clocking in the data on the input line. The remaining receiver mode description is separated into NRZ operation and biphase operation. 35
- 40 NRZ receiver operation begins when the idle line goes to zero indicating arrival of the start bit. Receiver synchronization is established by setting RSE at approximately the middle of the start bit. R₀ is derived logically from the RB Register 31 which runs continuously as long as RE is set. R₀ clocks the Receive Shift Register 34 and the receiver bit counter RC. In the NRZ format the separation flip-flop RSD 111 (Fig. 8J) acts merely as another bit of the Receive Shift Register 34 to hold the stop bit, and it is clocked by RBD $\oplus$ RBE, which lead R₀ by 1/4 bit time. 40
- 45 In NRZ reception the incoming start bit is input to the RB Register 31 to synchronize one frame by setting RSE 102 (Fig. 8I). For the next nine bits of the frame RB Register 31 toggles to generate R₀ and the RSD clock. The data stream is clocked into RSD 111, the Receive Shift Register 34, and RST 250. If no framing error exists the eight data bits are transferred to the Receive Data Register 36, and the RBF status bit is set to indicate receipt of one correct word. If a framing error has occurred or if a previous RDF has not been serviced and reset the OR status bit is set to indicate an overrun. RSE resets the RC 10-count. 45
- 50 Biphase receiver operation begins with receipt of a "zero" which has been preceded by at least one "one". Receiver synchronization is established by setting RSE 102 at approximately the middle of the start bit. RSH and RLG are derived logically from the RB Register and are used to control the separation flip-flop SEP 101. The separation logic generates the receiver bit clock R₀ which drives the receiver bit counter RC. RSH clocks the RSD flip-flop 111 which transforms the biphase data format to the NRZ format. From this point on the Receive Shift Register 34 and RDST 35 function as in NRZ format mode of operation, with the exception that RSE 102 does not automatically reset during the 10-count and set during the start bit. 50
- 55 A full duplex operation is initiated by setting both RE and TE. The operation of the transmitter 55

is independent of the receiver except for the data format and rate.

The wake-up capability is intended to increase the throughput of the CPU in multi-processor applications by permitting non-interested CPU's on a common line to ignore the remainder of a message being transmitted. A processor wishing to invoke the wake-up feature sets the wake-up bit (WU) in the Control and Status Register 46. With the wake-up bit set, the receiver section continues to process messages, but does not set the RDRF flag bit. This effectively masks the message from the processor. Upon receiving eleven consecutive 1's, the receiver section clears the wake-up bit, thus enabling normal flag operation. Refer to Fig. 6 wherein a master CPU 500 communicates with only Slave CPU 505. Slave CPUs 503 and 504 ignore the message on serial bus 502.

It is important to consider the duties of the transmitter processor during wake-up operation. A "message" is considered to consist of a string of characters transmitted in such a manner that the transmission line does not go idle within the message for a period sufficient for a receiver in wake-up operation to clear its wake-up bit. The wake-up bit is cleared (or reset) by a string of eleven consecutive ones. It is important to realize that both data and the stop bit count in the total of eleven ones. Software servicing the transmitter will store the next character in the Transmit Data Register 37 upon detecting the TDRE is equal to "1". The transmission line will be kept busy shifting out the current character for 10 bit times (1 start bit, 8 data bits, 1 stop bit) at which time the line will go idle if the transmitter has not been serviced. If the last byte transmitted was all ones (SFF), the transmitter has a total of 11 bit times to respond to the TDRE flag before a receiver will see eleven consecutive ones. If the most significant bit of the data was a zero, the transmitter service routine has a total of 13 bit times to respond to the TDRE flag before affecting the wake-up bit. These two cases represent the minimum and maximum response times required by the transmitter service routine in order to keep the receivers in wake-up.

After the last character of a given message has been sent, it is also the duty of the transmitter to idle the line for the required time before beginning the next message. The time required is data-dependent, as are the response times. If the last byte sent was all ones, the transmitter must wait a total of 12 bit times (after servicing the TDRE flag) before beginning the next message. The maximum wait is required if the most significant bit of the last byte transmitted was a zero. With this condition, the transmitter must wait a total of 20 bit times before beginning the next message.

A receiver service routine wishing to invoke the wake-up mode usually makes this decision after determining that it is not an addressee for a particular message. The receiver routine may then effectively ignore the remainder of the message by setting the wake-up bit. It is important to realize that the wake-up bit cannot be set if the transmission line is idle. With the wake-up bit set, the RDRF flag bit will not be set during the remainder of the message. Providing the RDRF flag bit is clear, the ORFE bit will be set upon receipt of the next character of the current message. When the receiver section detects eleven consecutive ones on the transmission line, the wake-up bit will clear and the RDRF flag bit will be set upon receipt of the first character of the next message.

During reset the microcomputer is set to its initial state and the only active signals are the ϕ_1 and ϕ_2 clocks, which run as long as power is applied to the chip. At the end of reset the Timer counter starts operating with the rising edge of the ϕ_2 clock. In general, initiation of serial I/O operation requires execution of two or three write instructions, but since the serial I/O comes out of reset in the biphase format at the highest data rate, it may be possible to initiate operation with only one write instruction. The 16 data modes are possible: four data rates (see Table 1) for each of biphase, NRZ (no clock), NRZ (internal clock) and NRZ (external clock).

If at the end of reset the first desired usage of the serial I/O is to receive biphase data at the highest data rate, this mode can be established with one write instruction. Setting RE will establish the reception mode. Since the RB Register 31 begins filling up with "ones" as soon as the RT clock starts, at least one idle bit must precede the start bit to insure generation of an RLG pulse on the leading edge of the start bit. If the start bit happened to be a high value there would be no leading edge to generate the first RLG and synchronization would not be established.

If a mode other than MCR = 0 is desired or if the current mode is to be changed, two write instructions must be executed, one for the Rate and Mode Control Register 45, and one for the Control and Status Register 46. The format and data rate are determined by the former, and the duplex and control modes are determined by the latter.

The waveforms of MCR = 1, 2, and 3 are the same as those in the case where MCR = 0 with the exception that the setting of RE is delayed 5 more ϕ_2 times, since the Rate and Mode Control Register must be written also. Initiation of NRZ operation (MCR ≥ 4) can occur after reset as soon as RE is set.

In addition to establishing the data, control, and duplex modes it may be necessary to write data to the Transmit Data Register 37. If it is desired to begin transmission as soon as possible,

the data word is written to the Transmit Data Register 37, and then the bits of the Control and Status Register 46 should be configured to TDRE = 0 and TE = 1. Loading the Transmit Data Register before loading the Control and Status Register allows data transfer and data transmission to begin immediately.

- 5 If the transmission is to be interrupt-driven TDE is left set, and the interrupt service routine handles the writing of the Transmit Data Register 37. 5

The serial I/O allows interrupt service routines to monitor its operation. Transmitter interrupts are enabled by setting TIE, and receiver interrupts are enabled by setting RIE.

- 10 If a polling operation is desired, interrupts can be disabled by resetting either TIE or RIE or both. 10

Fig. 17 illustrates the detailed circuit diagram of the D flip-flops shown in the detailed logic diagram of Figs. 8A-8J, as, for example, used to implement flip-flops RBA-RBH (291-298) in Fig. 8J.

- 15 Fig. 18 illustrates a detailed circuit diagram corresponding to the D flip-flops with reset, as used, for example, to implement flip-flops TBA-TBD in Fig. 8A. 15

Fig. 19 illustrates a detailed circuit diagram corresponding to the coupled latches with set and reset, as used, for example, to implement flip-flops TSE 427 (Fig. 8B) and RDF 235 (Fig. 8D).

Fig. 20 illustrates a detailed circuit diagram corresponding to the coupled latches with $\bar{S}$ and $\bar{R}$ inputs, as used, for example, to implement flip-flop RSE 102 in Fig. 8I.

- 20 Fig. 21 illustrates a detailed circuit diagram corresponding to the latch with reset used, for example, to implement flip-flops RB0-RB7 in Fig. 8E. 20

Fig. 22 illustrates a detailed circuit diagram corresponding to the follower latches used, for example, to implement latches RFF, TFF, and ORF in Fig. 8D.

- 25 The operation of the serial I/O communication logic illustrated in Figs. 8A-8J, 10 and 11 may be summarized by the following logic equations: 25

CONTROL SIGNALS

$$BP = \overline{CC1} \cdot \overline{CC\phi}$$

$$30 \quad R81 = \left[\begin{array}{c} 7 \\ | \quad | \\ n=0 \end{array} RDS_n \right] \cdot RSD \cdot RDST \cdot R\theta \quad 30$$

$$35 \quad RSE_S = \overline{BP} \cdot \overline{RBD} \cdot \overline{RBE} \cdot P23 + BP \cdot LG \quad 35$$

$$RSE_R = \overline{RE} + RXFR \cdot (BP + SH) + \overline{RCD} \cdot \overline{RCE} \cdot \overline{RSD} \cdot R\theta$$

$$RSE_C = R\theta$$

$$RSH = \overline{RBA} \cdot \overline{RBF} \cdot \overline{RBG} + \overline{RBA} \cdot \overline{RBF} \cdot RBG$$

$$RLG = \overline{RBA} \cdot \overline{RBF} \cdot \overline{RBG} + \overline{RBA} \cdot \overline{RBF} \cdot RBG$$

$$40 \quad SEP_D = \overline{SEP} \quad 40$$

$$SEP_R = RLG$$

$$SEP_C = RSH$$

$$R\theta = RLG + RSH \cdot \overline{SEP}$$

$$RXFR^* = RSE \cdot \overline{RCD} \cdot \overline{RCE} \cdot \overline{RSD} \cdot R\theta$$

$$45 \quad RXFR = RXFR^* \cdot \phi_1 \quad 45$$

$$XFR = RXFR \cdot \overline{RDF}$$

$$RT = (\overline{CC1} + \overline{CC\phi}) \cdot (\overline{S1} \cdot \overline{S\phi} \cdot T\phi + \overline{S1} \cdot S\phi \cdot T3 + S1 \cdot \overline{S\phi} \cdot T6 + S1 \cdot S\phi \cdot T8) + \overline{CC1} \cdot \overline{CC\phi} \cdot P22$$

$$TSE_S = \overline{TCD} \cdot \overline{TCE} \cdot T\theta$$

$$TSE_R = \overline{TE}$$

$$50 \quad TSE_C = T\theta \quad 50$$

$$TSH = \overline{TBC} \cdot \overline{TBD} + \overline{TBC} \cdot TBD$$

$$T\theta = \overline{TBC} \cdot TBD$$

$$T3 = T\theta \cdot (TCA + TCE)$$

$$TXFR^* = TSE \cdot \overline{TCA} \cdot \overline{TCE} \cdot T\theta$$

$$55 \quad TXFR = TXFR^* \cdot \phi_1 \quad 55$$

CONTROL AND STATUS REGISTER

$$WU_{WR} = \overline{PDB\phi} \cdot \overline{WSR} \cdot \overline{Reset}$$

$$WU_C = \overline{WSR} \cdot R\theta$$

$$60 \quad WU_R = R81 + \overline{RE} \quad 60$$

$$RDF_{WR} = RXFR \cdot \overline{Reset} \cdot \overline{WU}$$

$$RDF_R = \overline{RFF} \cdot \overline{RRD}$$

$$RDF_C = \overline{RXFR}$$

$$RFF_{WR} = \overline{RDF} \cdot \overline{RSR}$$

$$65 \quad RFF_R = \overline{RDF} \quad 65$$

	$TDE_{WR} = TXFR + \text{Reset}$	
	$TDE_R = TFF.WTD$	
	$TFF_{WR} = TDE.RSR$	
	$TFF_R = TDE + \text{Reset}$	
5	$OR_{WR} = (RDF.RXFR + \overline{RCD.RCE.RSD.WU.R\theta}).\text{Reset}$	5
	$OR_R = ORF.RRD$	
	$OR_C = RDF.RXFR + \overline{RCD.RCE.RSD.WU.R\theta}$	
	$ORF_{WR} = OR.RSR$	
10	$ORF_R = \overline{OR}$	10
	$SCR_{nWR} = PDB_n.WSR.\text{Reset} \quad 1 \leq n \leq 4$	
	$TE = SCR1$	
	$TIE = SCR2$	
	$RE = SCR3$	
15	$RIE = SCR4$	15
	RATE AND MODE CONTROL REGISTER	
	$MCR_{nWR} = PDB_n.WMR.\text{Reset} \quad 0 \leq n \leq 3$	
	$S\phi = MCR\phi$	
20	$S1 = MVR1$	20
	$CC\phi = MCR2$	
	$CC1 = MCR3$	
	$MCR_{nC} = \overline{WMR}$	
	RECEIVE DATA REGISTER	
25	$RDR_{nWR} = XFR.RDS_n \quad 0 \leq n \leq 7$	25
	$RDR_{nR} = \text{Reset} \quad 0 \leq n \leq 7$	
	TRANSMIT DATA REGISTER	
30	$TDR_{nWR} = PDB_n.WTD \quad 0 \leq n \leq 7$	30
	$TDR_{nR} = \text{Reset} \quad 0 \leq n \leq 7$	
	RB REGISTER	
	$RBA_D = \overline{RE} + BP.P23 + \overline{BP}.\overline{RBH}.(RCA + RCD) + \overline{RCA}.\overline{RCD}.(WU + RBH).P23$	
35	$RBB_D = RBA$	35
	$RBD_D = RBB$	
	$RBD_D = RBC$	
	$RBE_E = RBD$	
	$RBF_D = RBE$	
40	$RBG_D = RBF$	40
	$RBH_D = RBG$	
	$RBA_C = RBB_C = RBC_C = RBD_C = RBE_C = RBF_C = RBG_C = RBH_C = RT$	
	TB REGISTER	
45	$TBA_D = \overline{TBD}$	45
	$TBB_D = TBA$	
	$TBC_D = TBB$	
	$TBD_D = TBC$	
	$TBA_R = TBB_R = RBC_R = TDB_R = \text{Reset}$	
50	$TBA_C = TBB_C = TBC_C = TBD_C = RT$	50
	RC COUNTER	
	$RCA_D = RSE.RCE$	
	$RCB_D = RCA$	
55	$RCC_D = RCB$	55
	$RCD_D = RCC$	
	$RCE_D = RCD$	
	$RCA_R = RCB_R = RCC_R = RCD_R = RCE_R = \overline{RSE}$	
	$RCA_C = RCB_C = RCC_C = RCD_C = RCE_C = R\theta$	
60		60
	TC COUNTER	
	$TCA_D = \overline{TCE}$	
	$TCB_D = TCA$	
	$TCC_D = TCB$	
65	$TCD_D = TCC$	65

TABLE II
DATA LINK CONFIGURATIONS

5	Operation Mode	External Clock	No. Pins Used	Port 2 Pin Configuration			5
				P22	P23	P24	
	BP . Receive	no	1	not used	input	not used	
10	BP . Transmit	no	1	not used	not used	output	10
	BP . Full-Duplex	no	2	not used	input	output	
	NRZ . Receive	no	1	not used	input	not used	
	NRZ . Transmit	no	1	not used	not used	output	
	NRZ . Full-Duplex	no	2	not used	input	output	
15	NRZ . Receive	yes	2	input	input	not used	15
				or			
				output			
	NRZ . Transmit	yes	2	input	not used	output	
				or			
20				output			20
	NRZ . Full-Duplex	input	3	input	input	output	
	NRZ . Full-Duplex	output	3	output	input	output	
	Clock Only	output	1	output	not used	not used	

25 25

CLAIMS

1. A microcomputer including a processor, serial I/O communication logic and bus means for coupling said processor to said serial I/O communication logic, said serial I/O communication logic comprising control register means coupled to said processor, for storing transmit mode and receive mode signals provided by said processor, transmitter means coupled to said processor, for receiving and storing data provided periodically by said processor, and for serially transmitting said stored data via a serial output line in either NRZ or biphasic format as selected by said transmit mode signal stored in said control register means, and receiver means coupled to said processor, for serially receiving data via a serial input line in either NRZ or biphasic format as selected by said receive mode signal stored in said control register means, storing said received data, and periodically providing said stored data to said processor.
2. A microcomputer as claimed in claim 1, wherein said control register also stores transmit enable and receive enable signals provided by said processor, wherein said transmitter means are operative only in response to said transmit enable signal stored in said control register means, and wherein said receiver means are operative only in response to said receive enable signal stored in said control register means.
3. A microcomputer as claimed in claim 1, wherein said control register also stores transmission rate and reception rate signals provided by said processor, wherein said transmitter means transmit at a rate selected by said transmission rate signal stored in said control register means, and wherein said receiver means receive at a rate selected by said reception rate signal stored in said control register means.
4. A microcomputer as claimed in claim 1, wherein said serial I/O communication logic operates in synchronization with a clock signal, wherein said microcomputer further comprises an internal clock source for generating an internal clock signal and a terminal for receiving an external clock signal from an external clock source, and wherein said control register also stores a clock source signal provided by said processor, said serial I/O communication logic further comprising means coupled to said control register for operating said serial I/O communication logic in synchronization with said internal clock signal or said external clock signal as selected by said clock source signal stored in said control register.
5. A microcomputer having serial I/O communication logic and substantially as hereinbefore described and as shown in the accompanying drawings.



TABLE II
DATA LINK CONFIGURATIONS

5	Operation Mode	External Clock	No. Pins Used	Port 2 Pin Configuration			5
				P22	P23	P24	
	BP . Receive	no	1	not used	input	not used	
10	BP . Transmit	no	1	not used	not used	output	10
	BP . Full-Duplex	no	2	not used	input	output	
	NRZ . Receive	no	1	not used	input	not used	
	NRZ . Transmit	no	1	not used	not used	output	
	NRZ . Full-Duplex	no	2	not used	input	output	
15	NRZ . Receive	yes	2	input	input	not used	15
				or			
	NRZ . Transmit	yes	2	output			
				input	not used	output	
				or			
20	NRZ . Full-Duplex	input	3	input	input	output	20
	NRZ . Full-Duplex	output	3	output	input	output	
	Clock Only	output	1	output	not used	not used	

25 CLAIMS 25

1. A microcomputer including a processor, serial I/O communication logic and bus means for coupling said processor to said serial I/O communication logic, said serial I/O communication logic comprising control register means coupled to said processor, for storing transmit mode and receive mode signals provided by said processor, transmitter means coupled to said processor, for receiving and storing data provided periodically by said processor, and for serially transmitting said stored data via a serial output line in either NRZ or biphase format as selected by said transmit mode signal stored in said control register means, and receiver means coupled to said processor, for serially receiving data via a serial input line in either NRZ or biphase format as selected by said receive mode signal stored in said control register means, storing said received data, and periodically providing said stored data to said processor. 30 35

2. A microcomputer as claimed in claim 1, wherein said control register also stores transmit enable and receive enable signals provided by said processor, wherein said transmitter means are operative only in response to said transmit enable signal stored in said control register means, and wherein said receiver means are operative only in response to said receive enable signal stored in said control register means. 40

3. A microcomputer as claimed in claim 1, wherein said control register also stores transmission rate and reception rate signals provided by said processor, wherein said transmitter means transmit at a rate selected by said transmission rate signal stored in said control register means, and wherein said receiver means receive at a rate selected by said reception rate signal stored in said control register means. 45

4. A microcomputer as claimed in claim 1, wherein said serial I/O communication logic operates in synchronization with a clock signal, wherein said microcomputer further comprises an internal clock source for generating an internal clock signal and a terminal for receiving an external clock signal from an external clock source, and wherein said control register also stores a clock source signal provided by said processor, said serial I/O communication logic further comprising means coupled to said control register for operating said serial I/O communication logic in synchronization with said internal clock signal or said external clock signal as selected by said clock source signal stored in said control register. 50

5. A microcomputer having serial I/O communication logic and substantially as hereinbefore described and as shown in the accompanying drawings. 55