INTERFACING THE NXP SC28L92 DUART

This document is one of several papers addressing the adaptation of an NXP® Impact™ universal asynchronous receiver/transmitter to a computer system powered by a Western Design Center 65C02 or 65C816 microprocessor. The information herein is meant to supplement, not replace, the information in the applicable NXP data sheet.

Reasonable assumptions have been made about your ability to understand electrical schematics and common computer hardware terminology. If there is something you don't understand please seek help from a knowledgeable friend, instructor, co-worker, etc. A recommended on-line resource for 65xx family hardware and programming is 6502.org. If you are a 6502.org member you can post queries in the forum on subjects you don't understand and will likely receive expert help.

NXP's Impact line of UARTs has been targeted to industrial and automotive environments, and thus has design characteristics intended to promote high performance and reliable operation under all reasonable conditions. As the title indicates, this paper focuses on NXP's SC28L92 ("28L92") dual-channel universal asynchronous receiver/transmitter (DUART). The 28L92 is a high performance DUART intended for use in TIA-232, TIA-422 and TIA-485 serial data interfaces. The 28L92 supports Intel® x86 and Motorola® 68000 bus interfaces, and is readily adapted to a 65C02 or 65C816 system with common logic gates.

Key features of the 28L92 include:

- Two independent communications channels.
- FIFO-equipped receivers and transmitters.
- Wide range of data rates and formats.
- High speed communication capability.
- Configurable hardware handshaking.
- General purpose inputs and outputs.
- Precision counter/timer with sub-microsecond resolution.
- Intelligent interrupt subsystem.
- Fine-grained control of all device features.
- 3.3 or 5 volt operation.
- Available in PLCC44 and QFP44 packages.

Throughout this paper, the word "microprocessor" will specifically refer to both the 65C02 and 65C816.[1] Also, references will be made to the "Ø2 clock." Ø2 is the system clock signal used to drive the microprocessor—it is not to be confused with the clock that drives the 28L92.

[1] Use of NXP UARTs with the NMOS 65xx family is not recommended.

Here is some general information about the 28L92's features.

- The 28L92 has two identical high speed communications channels that are independently programmable in all respects.

- The 28L92's FIFOs ("first-in, first-out" data structure) assist in maintaining system throughput and efficiency during sustained high speed data transfers. Both receivers and transmitters are equipped with a FIFO, which may be configured to be either "8-deep" or "16-deep," the numbers referring to the number of datums[2] that each FIFO can temporarily hold.

  A receiver FIFO offer two benefits to the programmer that are not available with a UART that has no FIFO, such as the 65C51 that is often found in 65C02 and 65C816 systems.

  - Decreased potential for an overrun error. In any UART lacking a receiver FIFO, a newly-arrived datum must be read from the receiver and stored somewhere before the arrival of another datum. If system processing loads are such that it isn't possible to immediately service the UART's receiver and another datum arrives, that datum will overwrite the unread datum, resulting in corruption of the data stream. This problem is largely avoided in the 28L92, since it is likely that there will be enough room in the FIFO to hold another datum until the receiver can be serviced.

  - Fewer interrupts during sustained data reception. In a UART with no receiver FIFO, the arrival of a datum usually results in the generation of an interrupt, which must be promptly serviced to avoid the overrun problem described above. In contrast, up to 16 datums can be accepted by each of the 28L92's receivers before it an interrupt will be generated. Furthermore, all 16 datums may be read and stored from both receivers during the servicing of the interrupt. Overall efficiency is improved and more processor time is available to execute the foreground task.

  The transmitter FIFO assists in reducing the number of interrupts that will occur during sustained output. Depending upon configuration, up to 16 datums may be written to the transmitter's FIFO during the servicing of a single interrupt. Once the FIFO has been filled the microprocessor can go on to something else, and later on when all data in the FIFO has been transmitted, the 28L92 will again interrupt the microprocessor. At that time, the processor can load another 16 datums into the transmitter FIFO and repeat the process.

- The 28L92 supports standard data rates from 50 bits per second (bps) to 230.4 kilobits per second (Kbps), and non-standard rates from nearly zero to a maximum of 921.6 Kbps. As well as the ubiquitous 8N1 data format, numerous other formats are supported, with and without parity. Receiver and transmitter data rates and formats are independently configurable.

---

[2]The term "datum" refers to the individual pieces of information flowing on the serial interface. This term is used instead of "byte" because the data format could be something other than eight bits. If the data format is eight bits, no parity and one stop bit (8N1) then a datum and a byte are one and the same. "Datums" is an obscure alternate way of pluralizing "datum"—"data" is the common plural form.

- The 28L92 may be configured to use hardware flow control, which is usually a necessity when operating at speeds beyond 9600 bits per second (bps). The request-to-send (RTS) receiver handshake signal may be configured to automatically de-assert when the receiver FIFO becomes full, which removes the burden of managing incoming data flow control from the microprocessor.

- The general-purpose inputs and outputs (referred to in the text as GPI and GPO, respectively) may be used for a variety of interfacing tasks, such as hardware flow control, receipt of a carrier detect signal from a modem, data terminal ready (DTR) signal generation, or other user-defined functions. Collectively, the GPIs and GPOs are a form of out-of-band signaling that may be used to manage the serial interface. Change-of-state detection on some GPIs may be used to generate an interrupt when input state changes.

- The counter/timer (C/T) may be used in timer mode to generate a periodic interrupt or to generate a square wave output on one of the GPOs for other purposes. Other modes allow the C/T to act as a time delay device or as an interval or pulse counter. The C/T has a minimum resolution of approximately 543 nanoseconds (ns) when the 28L92 is driven by the recommended 3.6864 megahertz (MHZ) clock signal.

- The 28L92's interrupt subsystem is able to report up to eight events, any of which may be set up to interrupt the microprocessor when the corresponding event occurs. It is also possible for the 28L92 to be configured so it reports receiver and transmitter interrupts as discrete hardware outputs, which feature is described at §1.2.6.5.

- Unlike the 65C51, the 28L92 offers fine-grained control of all features, making it possible, for example, to decouple transmitter behavior from the state of the clear-to-send (CTS) handshake input.

- The 28L92 is compatible with 3.3 volt systems, as well as 5 volt systems, with some changes in timing when operated on 3.3 volts—consult the NXP data sheet AC performance specifications for details.[3] PLCC44 packaging makes the device usable on standard 100×100 mil prototyping board in conjunction with any PLCC44 socket that has a 100×100 mil mounting pin grid.

[3] Circuit examples presented later on will generally ignore operating voltage.

1 PINOUT

This section discusses the 28L92's electrical connections. The below pinout chart is for the 28L92 in PLCC44 packaging. Pin descriptions assume the 28L92 is operating in Intel bus mode, which is further discussed at §1.1.1. Pins designated as No connection must not be connected to anything—device behavior is undefined if a No connection pin is connected to any part of the circuit.

Pin Number	Pin Name	Pin Function
1	—	No connection
2	A0	Address bus bit 0
3	IP3	General purpose input 3
4	A1	Address bus bit 1
5	IP1	General purpose input 1
6	A2	Address bus bit 2
7	A3	Address bus bit 3
8	IP0	General purpose input 0
9	WRN	Write data enable
10	RDN	Read data enable
11	RxDB	Channel B receiver serial data input
12	I/M	Operating mode select
13	TxDB	Channel B transmitter serial data output
14	OP1	General purpose output 1
15	OP3	General purpose output 3
16	OP5	General purpose output 5
17	OP7	General purpose output 7
18	D1	Data bus bit 1
19	D3	Data bus bit 3
20	D5	Data bus bit 5
21	D7	Data bus bit 7
22	V_{SS}	Logic ground

Pin Number	Pin Name	Pin Function
23	—	No connection
24	INTRN	Interrupt request
25	D6	Data bus bit 6
26	D4	Data bus bit 4
27	D2	Data bus bit 2
28	D0	Data bus bit 0
29	OP6	General purpose output 6
30	OP4	General purpose output 4
31	OP2	General purpose output 2
32	OP0	General purpose output 0
33	TxDA	Channel A transmitter serial data output
34	—	No connection
35	RxDA	Channel A receiver serial data input
36	X1/CLK	3.6864 MHz clock input
37	X2	3.6864 MHz clock input
38	RESET	Device reset
39	CEN	Chip enabled, active-low
40	IP2	General purpose input 2
41	IP6	General purpose input 6
42	IP5	General purpose input 5
43	IP4	General purpose input 4
44	V_{CC}	Logic power, 3.3 or 5 volts

The next two subsections will elucidate on pin functions. In the descriptions, the word “asserted” means to make an input active. If the input is an active-low type, asserting it means driving it low. Similarly, asserting an active-high input means driving it high. The term “de-assert” is the opposite of “assert.”

1.1 Microprocessor Interface

In this subsection, the connections that interface the 28L92 to the microprocessor and its control circuits will be described.

1.1.1 Pin 12 – I/M: Mode Select

The manner in which the I/M pin is connected determines if the 28L92 will operate in Intel x86 or Motorola 68000 bus mode. If this pin is floated or connected to V_{CC}, operation will be in Intel mode. If connected to ground, operation will be in Motorola mode. We recommend that the 28L92 be operated in Intel mode in a 65C02 or 65C816 system.

1.1.2 Pins 2, 4, 6, 7 – A0-A3: Address Bus

These pins are the 28L92's address bus inputs: A0, A1, A2 and A3, respectively. They should be directly connected to the respective microprocessor address bus lines. When the 28L92 is selected by asserting the CEN input (chip enable, described at §1.1.4), the bit pattern on these inputs, combined with the state of RDN (read data enable, described at §1.1.5) and WRN (write data enable, described at §1.1.6) will determine which of the device's registers will be accessible to the microprocessor.

1.1.3 Pins 28, 18, 27, 19, 26, 20, 25, 21 – D0-D7: Data Bus

These pins are the 28L92's data bus connections: D0, D1, D2, D3, D4, D5, D6 and D7, respectively. They should be directly connected to the respective microprocessor data bus lines. When the 28L92 is not selected these connections will be in the high impedance state, also referred to as the high-Z state.

1.1.4 Pin 39 – CEN: Chip Enable

CEN is an active-low input and is used to select or deselect the 28L92. When CEN is de-asserted, the 28L92 is idle and its data bus connections will be in the high-Z state. When CEN is asserted, the 28L92 is active and one of its registers, determined by the state of the A0-A3 inputs, will be accessible through the data bus. In a 65C816 system, CEN ***must not be asserted when the processor's*** VDA ***and*** VPA ***outputs are simultaneously low***, as the address bus will be in an indeterminate state at that time.

1.1.5 Pin 10 – RDN: Read Data Enable

RDN is an active-low input and is asserted to read data from the 28L92 when CEN is also asserted. In a 65C816 system, this input must be qualified by the Ø2 clock so it is only asserted when Ø2 is high. Best performance is achieved if CEN is asserted before RDN. RDN and WRN (described next) should never be simultaneously asserted, as doing so may cause undefined operation.

1.1.6 Pin 9 – WRN: Write Data Enable

WRN is an active-low input and is asserted to write data to the 28L92 when CEN is also asserted. This input must be qualified by the Ø2 clock so it is only asserted when Ø2 is high. Best performance is achieved if CEN is asserted before WRN. WRN and RDN should never be simultaneously asserted, as doing so may cause undefined operation.

1.1.7 Pin 24 – INTRN: Interrupt Request

INTRN is an active-low, open-drain output that is driven low by the 28L92 when an event that has been configured to cause an interrupt takes place. INTRN should be connected to the microprocessor's IRQB input, or an interrupt controller input (if used), and also must be pulled up to V_{CC} through a resistor, with a minimum recommended value in a 5 volt system of 2200 ohms and a maximum recommended value of 3300 ohms.

1.1.8 Pin 38 – RESET: Reset

RESET is an active-high input when the 28L92 is operating in Intel mode and must be de-asserted for normal device operation. When RESET is asserted, the 28L92 will go through a reset sequence identical to that caused by power cycling, the effects of which are enumerated in the data sheet. As the reset circuit in a 65C02/65C816 system is active-low, RESET must be driven by an inverter from the microprocessor's reset circuit in order to have the 28L92 initiate its reset sequence when the rest of the system is reset.

1.1.9 Pin 36 – X1/CLK: DUART Clock

X1/CLK is one of the 28L92's two clock inputs. The clock signal, referred to as the "X1 clock," is the DUART's time base and is responsible for driving the baud rate generator and C/T, as well as sequencing internal functions that govern the device's general operation. Hence the accuracy and stability of the X1 clock is fundamental to achieving predictable performance.

The "standard" (and recommended) clock frequency for the 28L92 is 3.6864 MHZ. At that frequency, the standard data rates listed in the data sheet will apply. The minimum permissible clock rate is 100 kilohertz[4] and the maximum permissible rate is 8 MHZ. As the X1 clock frequency is increased the 28L92 will tend to respond more quickly to external events, but will also consume more power. Conversely, decreasing the X1 clock frequency will cause slower response to some events, but will also reduce power consumption.

If it is desired to operate the 28L92 at a higher than standard frequency it is recommended that 7.3728 MHZ be used, as the standard data rates will exactly double, making it easier to configure the device. Clocking the 28L92 at arbitrary frequencies will complicate setting up data rates that are compatible with other serial devices.

---

[4] Operation of the 28L92 below 3.6864 MHz is not recommended in most systems.

The X1 clock may be generated by a crystal and external components, or by a self-contained ("can") oscillator. If you decide to use a crystal to generate the X1 clock, one side of your circuit will be connected to the X1/CLK pin and the other side will be connected to the X2 pin (§1.1.10)—refer to the circuit illustrated in the data sheet.

We strongly recommend the use of an oscillator. Achieving consistent performance with a crystal circuit can be tricky. Consider that oscillator manufacturers have already solve the performance problems for you. Furthermore, the printed circuit board (PCB) footprint of the crystal circuit will be about as large as that of a "half-can" oscillator, and the parts cost difference between the two will be negligible.

If used, an oscillator's output must be connected to the X1/CLK pin and the X2 pin must be left unconnected. The oscillator's output must be CMOS-compatible, with a minimum high output voltage equal to $0.8 \times V_{CC}$, and a maximum low output voltage not to exceed 0.4 volts. Most HCMOS oscillators will readily meet these requirements. We recommend the use of a "half-can" oscillator to reduce the device's PCB footprint.

1.1.10 Pin 37 – X2: DUART Clock

X2 is the 28L92's other clock input and is only used in conjunction with X1 clock generation by a crystal circuit. In such a case, your crystal circuit must be connected to this pin, as well as to the X1/CLK pin (§1.1.9). If an oscillator is employed to generate the X1 clock this pin must be left unconnected.

1.1.11 Pin 44 – V_{CC}: Logic Power

V_{CC} must be connected to logic power in your system, either 3.3 or 5 volts. Pin 44 must be bypassed to ground with a suitable capacitor. Logic power connections should be as physically short as practicable.

1.1.12 Pin 22 – V_{SS}: Logic Ground

V_{SS} must be connected to logic ground in your system, using the shortest physical connection that is practicable.

1.2 External Interface

In this subsection, the connections that interface the 28L92 to the "outside world" will be described. These connections operate at CMOS logic levels and hence must be attached to transceivers or line drivers and receivers in order to interface with serial devices such as terminals, printers and modems.

1.2.1 Pin 35 – RxDA: Channel A Receiver Serial Data In

Channel A of the 28L92 receives serial data on the RxDA pin, generally from the output of a transceiver or line receiver. If the device with which the 28L92 is communicating operates at CMOS levels then no transceiver or line receiver is required.

1.2.2 Pin 11 – RxDB: Channel B Receiver Serial Data In

Channel B of the 28L92 receives serial data on the RxDB pin, generally from the output of a transceiver or line receiver. If the device with which the 28L92 is communicating operates at CMOS levels then no transceiver or line receiver is required.

1.2.3 Pin 33 – TxDA: Channel A Transmitter Serial Data Out

Channel A of the 28L92 transmits serial data on the TxDA pin, generally to the input of a transceiver or a line driver. If the device with which the 28L92 is communicating operates at CMOS levels then no transceiver or line driver is required.

1.2.4 Pin 13 – TxDB: Channel B Transmitter Serial Data Out

Channel B of the 28L92 transmits serial data on the TxDB pin, generally to the input of a transceiver or a line driver. If the device with which the 28L92 is communicating operates at CMOS levels then no transceiver or line driver is required.

1.2.5 Pins 8, 5, 40, 3, 43, 42, 41 – IP0-IP6: General Purpose Inputs

The 28L92's seven general purpose inputs, IP0, IP1, IP2, IP3, IP4, IP5 and IP6, respectively, may be interfaced to the CMOS-compatible outputs of other devices. It is acceptable to float these inputs if they are not needed in your system.

IP0, IP1, IP2 and IP3 have change-of-state detectors, which makes it possible to configure the 28L92 to generate an interrupt when the logic level at any one of these inputs changes to the opposite state, for example, from logic 1 to logic 0.

In addition to their general purpose nature, all GPIs may be used for specialized functions:

1.2.5.1 IP0, pin 8, may be configured to act as channel A's clear-to-send (CTS) input. Typically, CTS would be connected to the other station's RTS output, thus implementing transmission hardware flow control.

1.2.5.2 IP1, pin 5, may be configured to act as channel B's CTS input. Typically, CTS would be connected to the other station's RTS output, thus implementing transmission hardware flow control.

1.2.5.3 IP2, pin 40, may be configured to act as an external clock input to the counter/timer. In such an arrangement, the counter/timer will be running independently of the X1 clock.

1.2.5.4 IP3, pin 3, may be configured to act as an external clock input to channel A's transmitter. In such an arrangement, the transmitter would be running independently of the 28L92's baud rate generator.

1.2.5.5 IP4, pin 43, may be configured to act as an external clock input to channel A's receiver. In such an arrangement, the receiver would be running independently of the 28L92's baud rate generator.

1.2.5.6 IP5, pin 42, may be configured to act as an external clock input to channel B's transmitter. In such an arrangement, the transmitter would be running independently of the 28L92's baud rate generator.

1.2.5.7 IP6, pin 41, may be configured to act as an external clock input to channel B's receiver. In such an arrangement, the receiver would be running independently of the 28L92's baud rate generator.

1.2.6 Pins 32, 14, 31, 15, 30, 16, 29, 17 – OP0-OP7: General Purpose Outputs

The 28L92's eight general purpose outputs, OP0, OP1, OP2, OP3, OP4, OP5, OP6 and OP7, respectively, may be interfaced to the CMOS- or TTL-compatible inputs of other devices. In addition to their general purpose nature, all GPOs may be used for specialized functions:

1.2.6.1 OP0, pin 32, may be configured to act as channel A's RTS output. Typically, RTS would be connected to the other station's CTS input, thus implementing reception hardware flow control.

1.2.6.2 OP1, pin 14, may be configured to act as channel B's RTS output. Typically, RTS would be connected to the other station's CTS input, thus implementing reception hardware flow control.

1.2.6.3 OP2, pin 31, may be configured to output either channel A's receiver's or transmitter's clock to other devices for synchronization purposes.

1.2.6.4 OP3, pin 15, may be configured so it outputs a square wave whose frequency is determined by the C/T if the latter is running in timer mode. Also, OP3 may be configured to output either channel B's receiver's or transmitter's clock to other devices for synchronization purposes.

1.2.6.5 OP4, OP5, OP6 and OP7, pins 30, 16, 29 and 17, respectively, may be configured to act as open-drain interrupt request outputs that reflect receiver and transmitter activity. If used in this fashion, these pins must be connected to V_{CC} through separate resistors, with a minimum recommended value in a 5 volt system of 2200 ohms and a maximum recommended value of 3300 ohms. Enabling this feature does not affect the behavior of the 28L92's interrupt request output (INTRN, §1.1.7).

In a typical application, these pins would be connected to the inputs of a priority encoder or other logic. Early in the interrupt service routine (ISR), the encoder would be queried to determine the source of the interrupt and a simple function would route execution in the ISR according to the source.

The microprocessor would not have to poll all likely interrupt sources in the 28L92. As receiver and transmitter activity generates most of the interrupts in any UART, a substantial gain in ISR performance would be realized.

1.2.6.5.1 OP4 may be configured to act as channel A's receiver interrupt request output. If a channel A receiver interrupt occurs the 28L92 will drive OP4 low. OP4 will return to the open-drain state once the interrupt has been cleared, usually by reading data from the channel A receiver.

1.2.6.5.2 OP5 may be configured to act as channel B's receiver interrupt request output. If a channel B receiver interrupt occurs the 28L92 will drive OP5 low. OP5 will return to the open-drain state once the interrupt has been cleared, usually by reading data from the channel B receiver.

1.2.6.5.3 OP6 may be configured to act as channel A's transmitter interrupt request output. If a channel A transmitter interrupt occurs the 28L92 will drive OP6 low. OP6 will return to the open-drain state once the interrupt has been cleared, usually by writing data to the transmitter.

1.2.6.5.4 OP7 may be configured to act as channel B's transmitter interrupt request output. If a channel B transmitter interrupt occurs the 28L92 will drive OP7 low. OP7 will return to the open-drain state once the interrupt has been cleared, usually by writing data to the transmitter.

2 MICROPROCESSOR INTERFACE

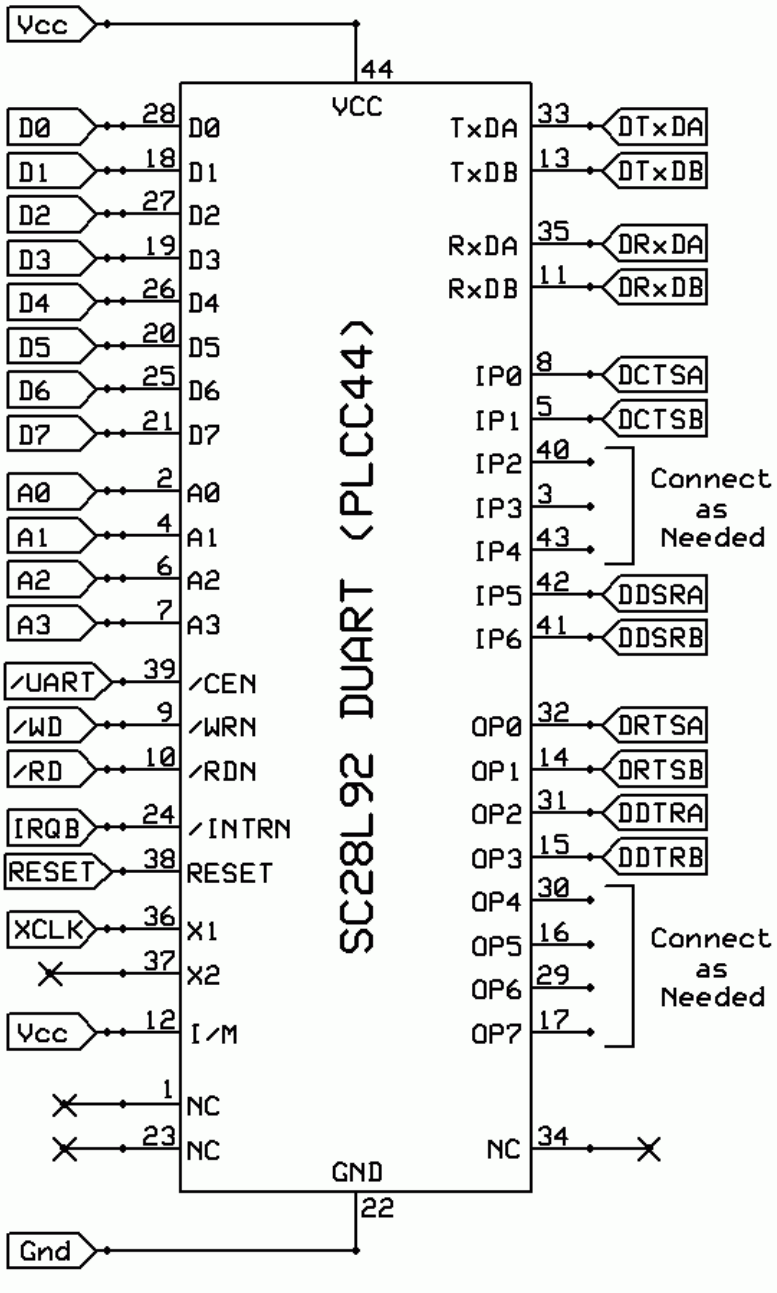
This section presents material describing how to interface the 28L92 to the microprocessor, using illustrations of circuits that were derived from a working system. All examples show the use of 74AC logic devices. You may use any logic family you wish as long as it is compatible with the operating voltage and other devices in the system, and is able to perform at the required Ø2 clock frequency. Generally speaking, 74HC logic is suitable up to 8 MHZ. Beyond that, we recommend 74AC logic. 74AC and 74HC logic may be used in 3.3 volt circuits if a performance penalty can be tolerated. Otherwise, 74LVC logic should be utilized.

In general, the microprocessor interface is the same for the 65C02 and 65C816. However, DUART selection logic that will be illustrated doesn't account for the bank address generated by the 65C816, which can be ignored in a system with no more than 64 kilobytes (KB) of address space. If your 65C816 system's address space exceeds 64KB you will need additional logic to make the 28L92 and other input/output (I/O) hardware appear only in one bank (conventionally, bank `$00`).

For clarity, bypass capacitors associated with the various devices in most of the illustrations have been omitted. Bypassing is an essential part of good circuit design, and therefore should not be neglected. Follow manufacturers' recommendations as necessary.

2.1 28L92 Symbology

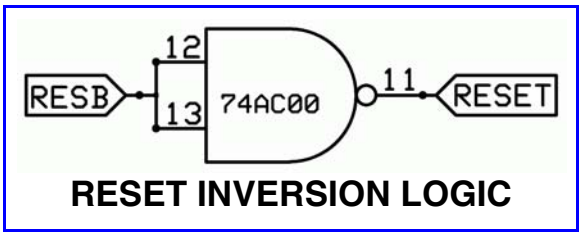
The illustration on the next page is that of the 28L92 with netlist symbols attached to its pins. All subsequent circuit illustrations will refer to these and other netlists. Pins connected to an X symbol are "no-connects" and therefore *must not be connected to anything*.



SC28L92 CIRCUIT SYMBOL

2.2 Reset Generation

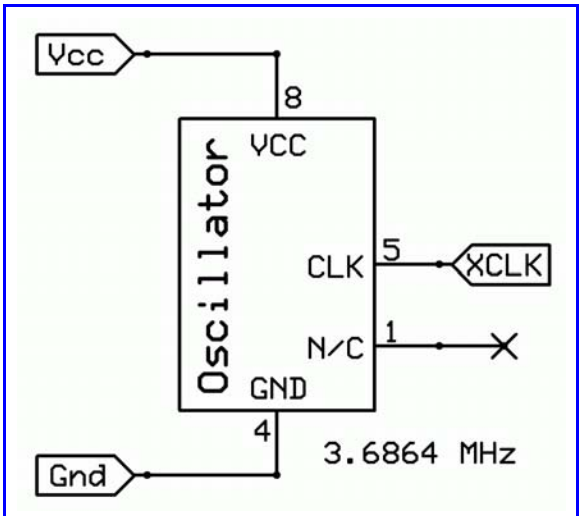
As described at §1.1.8, the 28L92's reset input is active-high, necessitating inversion of the microprocessor's active-low reset circuit. The following illustration shows the use of one section of a quadruple NAND gate to generate the required signal. The remaining sections of the same device will be used in other circuit functions.



RESET INVERSION LOGIC

2.3 X1 Clock Generation

The following illustration shows the use of a self-contained oscillator to generate the X1 clock. The pinout for the oscillator is typical of "half-can" units, which are physically the size of an eight pin, dual inline package.



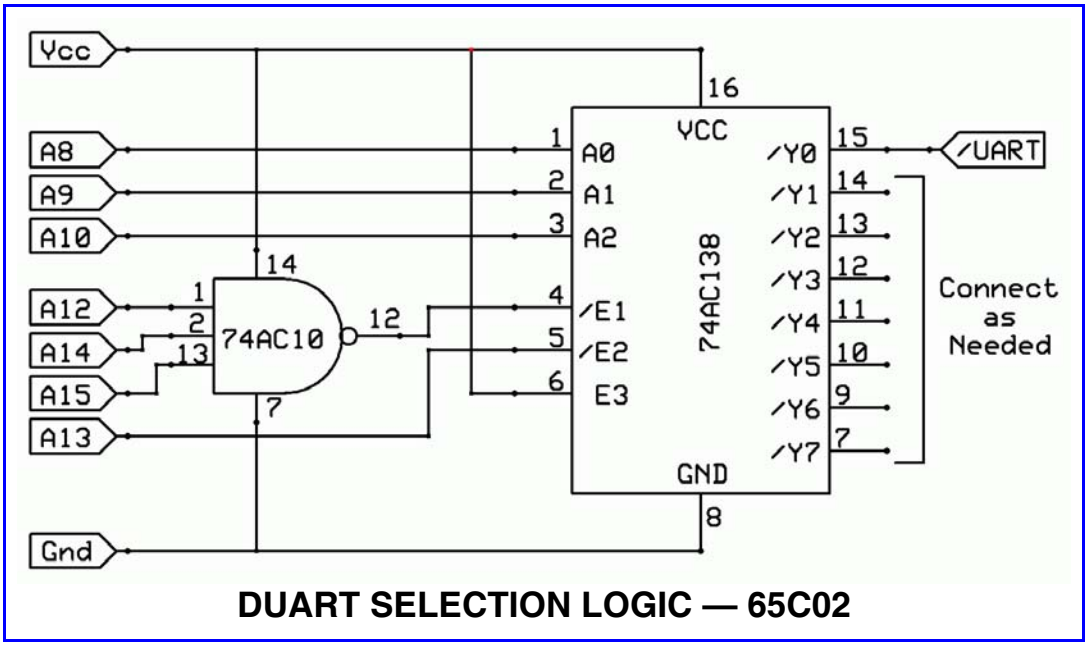
X1 CLOCK OSCILLATOR

2.4 Device Selection

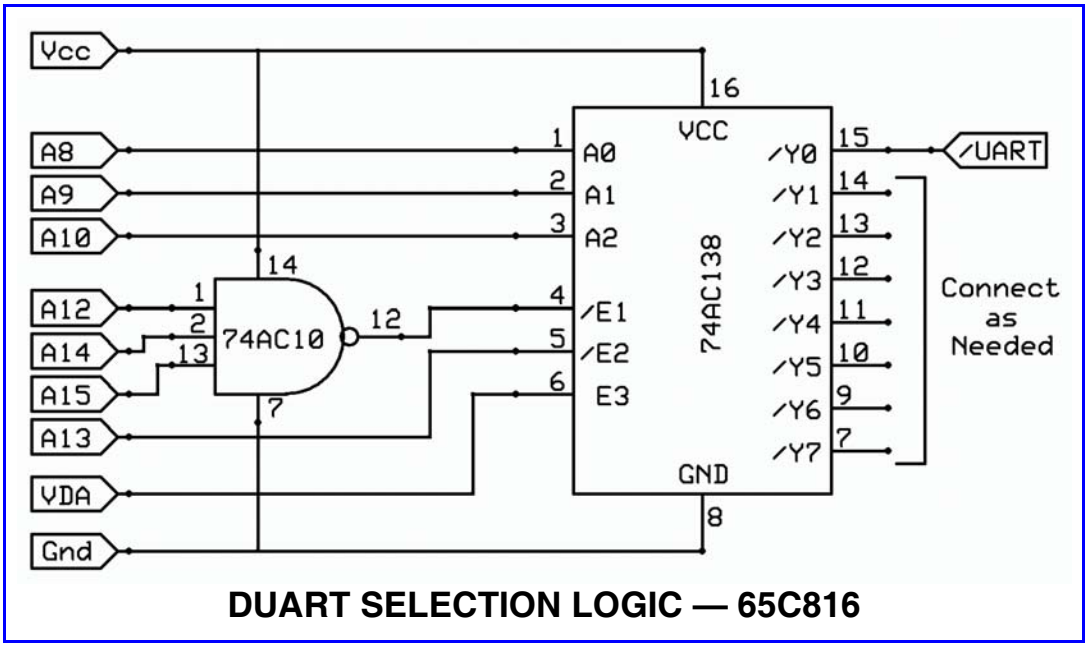
The illustrations on the next page are of DUART selection circuitry, with one version for the 65C02 and another for the 65C816. In both examples, a 74AC138 3-to-8 decoder is used to provide up to eight device select outputs in the address range `$D000-D7FF`,[5] with each device appearing on a different page boundary.

---

[5] As illustrated, decoding will repeat in the range `$D800-$DF00` due to address line A11 not being accounted for in the circuit.



DUART SELECTION LOGIC — 65C02

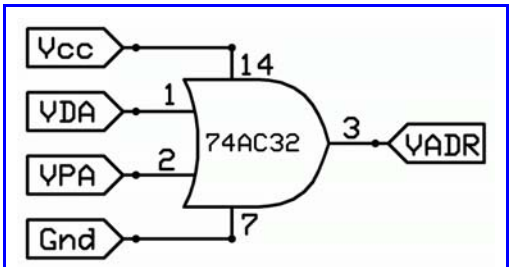


DUART SELECTION LOGIC — 65C816

As it is wired to the decoder's /Y0 output, the 28L92 will be selected any time the address is `$D0xx` (the decoder's /Y1 output would be driven low if the address were `$D1xx`, /Y2 would go low if the address were `$D2xx`, and so on). The 28L92's first register will appear at `$D000`, and its last register at `$D00F`.

The 65C816 circuit uses the microprocessor's VDA output to prevent decoder selection when the address bus is in an indeterminate state—VDA will be low during that time.

The illustrated arrangement is the most basic possible form of 65C816 address qualification, as it requires no additional gates. However, it doesn't account for all possible combinations of VDA and VPA. Should you desire to make it possible to select I/O hardware any time the 65C816 is emitting a valid address you will need to logically OR the VDA and VPA outputs and use the product of that operation to qualify decoder selection. The following circuit is an example.

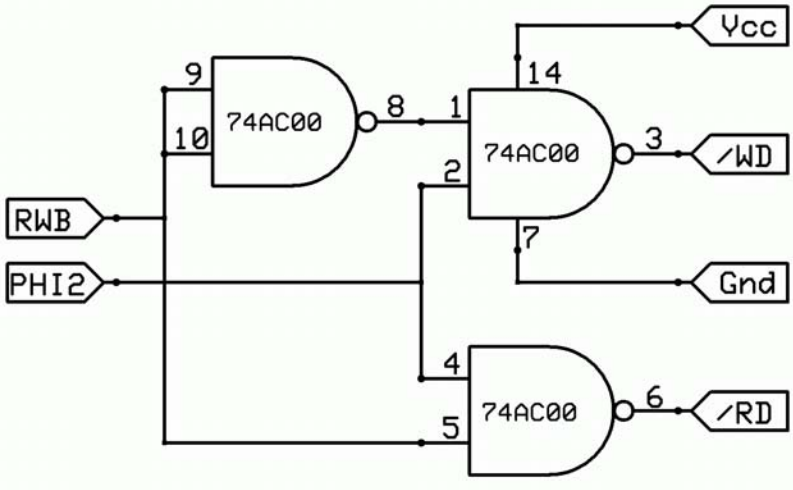


FULL 65C816 ADDRESS QUALIFICATION

If you use the above circuit, connect VADR to E3 (pin 6) of the 74AC138 decoder in place of VDA. VADR could also be used to qualify selection of other hardware in your system as needed.

2.5 Read/Write Generation

As the 28L92 has individual read and write control inputs (§1.1.5 and §1.1.6, respectively), the RWB output of the microprocessor must be converted to two separate active-low signals, designated /RD (read data) and /WD (write data). The below circuit is suitable for use with the 65C02 and the 65C816. It not only generates the /RD and /WD signals, it qualifies both of them so they are active only when the Ø2 clock is high. In any 6502-based system, /WD must be so qualified because the data bus will not necessarily contain actual data during Ø2 low. In a 65C816 system, both signals must be qualified by Ø2 due to the multiplexing of the bank address on the data bus during Ø2 low.



READ/WRITE LOGIC

Shortly after Ø2 goes low, the 65C816 will form a 24-bit address for the next read or write operation if the current instruction step involves such an access. The address will persist through the entire clock cycle and slightly beyond the next high-to-low Ø2 transition. While Ø2 is low, the 65C816 will drive bits 16-23 of the address onto the data bus—the resulting "bank address" will be emitted until Ø2 goes high. This technically amounts to a write operation and occurs without regard to the state of the 65C816's RWB output.

The generation of an address will result in a device such as the 28L92 being selected, assuming the address is one to which a device is wired. If /RD is also low at that time, meaning not qualified by Ø2, the selected device and the 65C816 will simultaneously drive the data bus, causing contention. It is possible that the bank address will be garbled by the contention, potentially resulting in a fatal hardware error and a system crash. Hence the need for qualifying /RD with Ø2.

The truth table for the above read/write circuit is as follows:

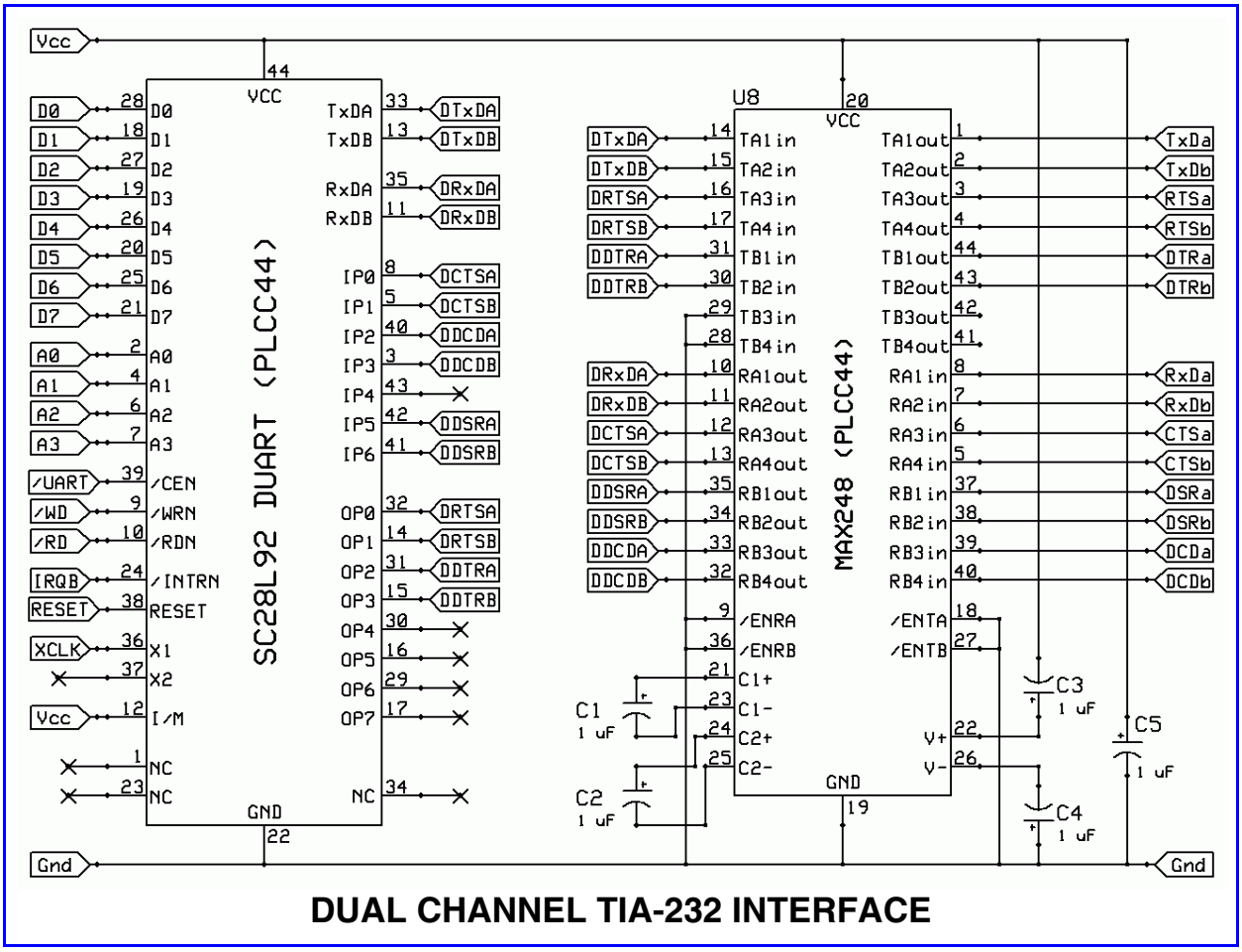
Ø2	RWB	/RD	/WD	Operation
L	X	H	H	——
H	H	L	H	read data
H	L	H	L	write data

In the above, L means a low logic level, H means a high logic level and X means "don't care."

Earlier, the illustration for the reset generation circuit (§2.2) showed the use of one section of a 74AC00 quadruple NAND gate. In the above read/write circuit, the remaining gates in the same device are used to derive /RD and /WD from the microprocessor's RWB signal. The PHI2 netlist symbol is the Ø2 clock signal. For convenience, one of the gates is wired to act as an inverter, a trick that was also used in the reset generation circuit.

3 TIA-232 INTERFACE

There are a number of ways the 28L92 may interfaced to other serial devices. We will only illustrate one method, which utilizes a Maxim MAX248 octal transceiver to create a TIA-232 (a.k.a RS-232) DTE ("data terminal equipment") dual channel interface with full hardware handshaking and data carrier detection, the latter a useful feature if a modem is attached to the system. A DTE usually communicates with a DCE ("data communications equipment"), which device is conventionally a modem. Two DTEs can communicate with each other through a pair of modems or through a "null modem," which is a cable that cross-connects the devices. Details on this aspect of the interface will not be covered—such information is readily available from numerous sources.



DUAL CHANNEL TIA-232 INTERFACE

In the above circuit, local connections to the two TIA-232 devices that are driven by the 28L92 are CTSx, DCDx, DSRx, DTRx, RxDx, RTSx and TxDx (right side of the illustration), where x in the symbol name is **a** for channel A and **b** for channel B.. Minimally, RxDx, TxDx and ground (Gnd) must be connected at both ends of each channel to establish the data link, a basic hook-up referred to as a "3-wire connection."

For a less elaborate interface, the MAX238 may be used in place of the MAX248 and the DCD, DSR and DTR connections eliminated. The MAX238 is in a DIP24 or SOIC24 package, versus the PLCC44 package of the MAX248, hence some printed circuit board space will be saved.

In order to set up hardware flow control, connect CTSx to the other device's RTS output and connect RTSx to the other device's CTS input. In lieu of these connections it will be necessary to implement flow control in software, which is not advisable if the data rate will exceed 9600 bps.[6]

Use of the DTR connections is optional in many cases. The other devices may require that their DSR inputs be asserted before they will go on line. In such a case, DSR on the external device would be connected to DTRx. Control of a modem by raising and lowering DTR is fairly common and a convenient way to cause the modem to drop a call when a connection session has terminated. Connection schemes vary with different devices, so be sure to consult applicable documentation.

Note that the DCDx connections are ultimately routed to the IP2 and IP3 inputs on the 28L92. These two GPIs were chosen for carrier detection because they have change-of-state detection and hence may be used to interrupt the microprocessor when the modem has answered a call and established carrier with the remote station's modem. Similarly, when the remote station breaks the call the change-of-state interrupt that would occur when carrier is lost can be used to terminate the local session that was started when the call was first established.

The DCTSx, DDCDx, DDSRx, DDTRx, DRxDx, DRTSx and DTxDx symbols attached to the MAX248 (right side of device) correspond to the same symbols attached to the 28L92. The 1 μF charge pump capacitors, C1-C4, should be a low-loss part—we use tantalum capacitors due to their small size. MLCCs are also acceptable. The bypass capacitor, C5, may be any low-loss part, including a low ESR electrolytic. All capacitors should be placed in close proximity to the MAX248 to minimize connection length. This is especially important with the bypass capacitor.

Note that the MAX248 requires five volts at its V_{CC} pin for proper operation. Power and ground connections to the MAX248 should be robust.

BDD: 2017/07/10

[6]Software flow control, also known as "XON/XOFF handshaking," requires that the DUART driver respond to <DC3> (XOFF, stop transmission) and <DC1> (XON, start transmission) control datums from the other device, as well as output <DC3> and <DC1> to tell the other device when to stop transmitting and when to resume. At high data rates too much time may elapse after a <DC3> is received before transmission is stopped, possibly causing a receiver overrun error. Such a problem is generally avoided with hardware flow control.