EXAR

DATA COMMUNICATIONS APPLICATION NOTE

DAN170

AUGUST 2004

MIGRATING TO EXAR'S FIFTH GENERATION UARTS: XR16L784/788

Author: BL

1.0 INTRODUCTION

This application note provides sytem design engineers with the information necessary to migrate to Exar's newer and more powerful Fifth Generation UARTs. It describes the hardware and software differences between the classic four channel UARTs (ST16C454, ST16C554, ST16C654, XR16C854 and XR16C864) and the newer fifth generation UARTs (XR16L784 Quad UART and XR16L788 Octal UART) which provide:

- easier hardware interface
- easier software programming
- higher perfomance in interrupt service routine
- register compatibility to the industry standard 16550

2.0 ADVANTAGES OF THE FIFTH GENERATION XR16L784 AND XR16L788

The upgrade path from one or two ST16C454/554/654/854/864 to an XR16L784 or XR16L788 offers the following benefits:

- Simpler hardware interface with Intel/Motorola CPU
- Reduction in board space (when moving from two ST16C454/554/654/854/864's to one XR16L788)
- Ability to interface with both 3.3V and 5V devices (5V Tolerant inputs)
- Higher baud rate via 8X sampling
- Fifth generation Flat Register Set, No shadow/mirror registers
- Global Interrupt Source Registers for easier/faster interrupt handling

It is to be noted that the fifth generation devices continue to offer the same advantages as the classic devices in the following aspects:

- Each channel is independent and has its own TX and RX FIFOs
- Each channel has a 16550-compatible register set
- Each channel has its own baud rate selection
- All channels use XTAL1 clock as the input clock for baud rate generation

3.0 HARDWARE DIFFERENCES

The newer XR16L78x UARTs enable simpler interfacing with a CPU and/or a programmable logic device. Let us consider two common cases:

- migrating from one ST16C454/554/654/854/864 to one XR16L784
- migrating from two ST16C454/554/654/854/864's to one XR16L788

Tables 1 and 2 describe the footprint and hardware differences for both the cases. Figures 1 and 2 show the connections required for a CPU-UART interface for both the cases.

TABLE 1: HARDWARE DIFFERENCES BETWEEN ONE ST16C454/554/654/854/864 AND ONE XR16L784

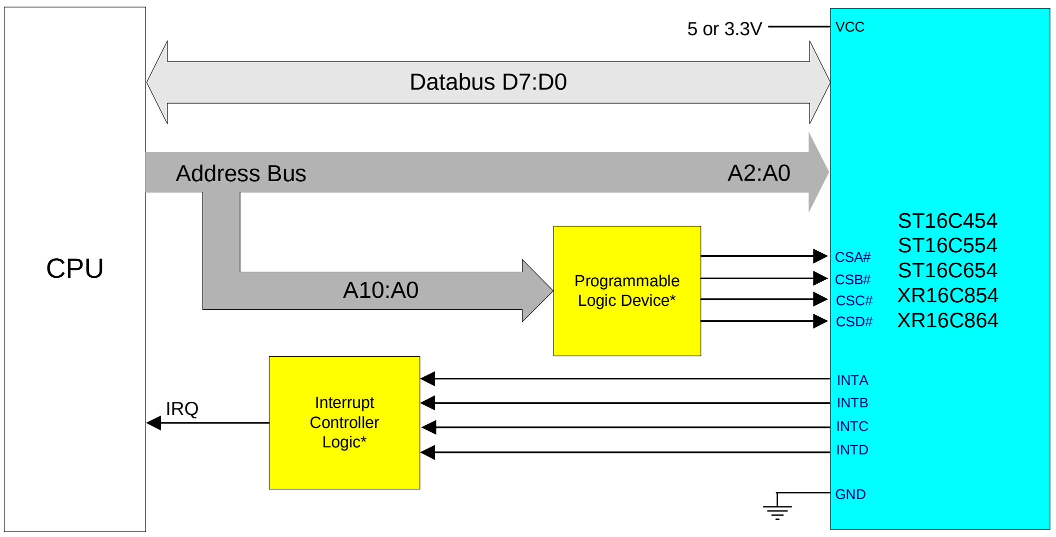
DESCRIPTION	ONE ST16C454/554/654/854/864	ONE XR16L784
Footprint	68-PLCC, 64-TQFP or 100-QFP	64-TQFP
5V Tolerant inputs (when VCC < 5V)	Not Available	Yes
Chipselect	One chipselect per channel* (requires external address decoder logic to generate the 4 chipselects)	One chipselect for the entire chip
Interrupt	One active Interrupt per channel* (cannot be wire-ORed; requires external logic to combine the 4 interrupts into a single IRQ)	One open-drain interrupt for the entire chip (can be wire-ORed)
Address Lines	3 (A2:A0) for 8 registers (and 7 shadow registers)	8 lines (A7:A0) for 16 registers per channel and 15 Device Configuration registers for the entire device)

TABLE 2: HARDWARE DIFFERENCES BETWEEN TWO ST16C454/554/654/854/864'S AND ONE XR16L788

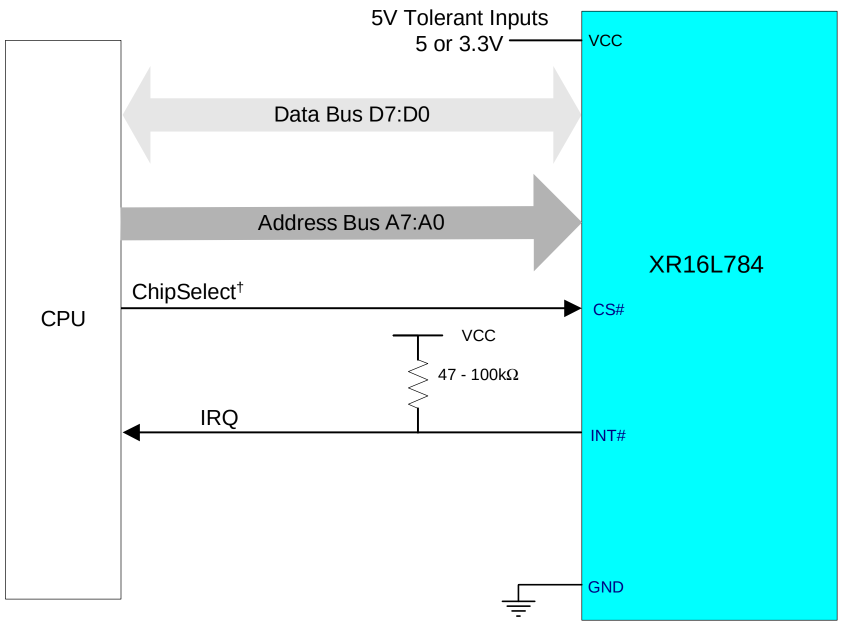
DESCRIPTION	TWO ST16C454/554/654/854/864	ONE XR16L788
Footprint	Two 68-PLCC, 64-TQFP or 100-QFP	One 100-QFP
5V Tolerant inputs (when VCC < 5V)	Not Available	Yes
Chipselect	One chipselect per channel* (requires external address decoder logic to generate the 8 chipselects)	One chipselect for the entire chip
Interrupt	One active Interrupt per channel* (cannot be wire-ORed; requires external logic to combine the 8 interrupts into a single IRQ)	One open-drain interrupt for the entire chip (can be wire-ORed)
Address Lines	3 (A2:A0) for 8 registers (and 7 shadow registers)	8 lines (A7:A0) for 16 registers per channel and 15 Device Configuration registers for the entire device)

*Note that these differences are only when the device operates in Intel mode (16/68# pin is HIGH). In Motorola mode (16/68# pin is LOW), each ST16C454/554/654/854/864 device has only one chipselect (with two additional address lines to select the four channels) and one open-drain interrupt output, similar to the XR16L784. However, when using two of the ST16C454/554/654/854/864's, one extra chipselect is still required, compared to the XR16L788.

FIGURE 1. CPU-UART INTERFACE WHEN USING ONE ST16C454/554/654/854/864 VS. ONE XR16L784



1A) Typical Connections between the CPU and One ST16C454/554/654/854/864

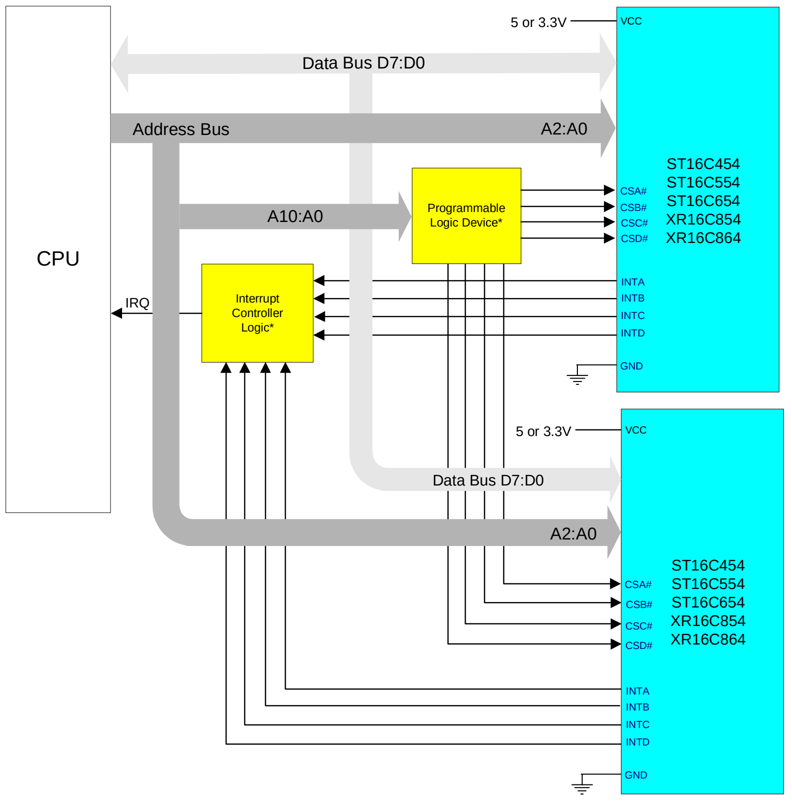


1B) Typical Connections between the CPU and One XR16L784

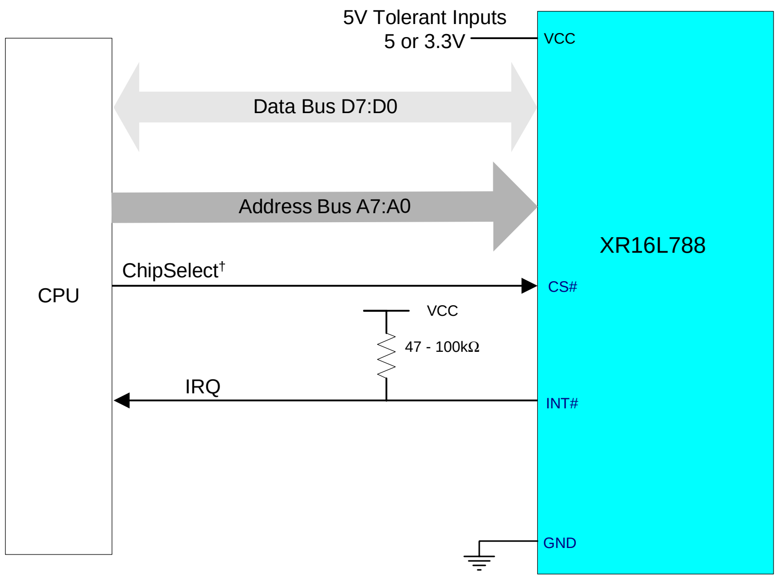
* Since there are four chipselects and four interrupts, external glue logic is essential.

†One of the chipselect outputs from the CPU can usually be used to drive the CS# pin of the XR16L784.

FIGURE 2. CPU-UART INTERFACE WHEN USING TWO ST16C454/554/654/854/864'S VS. ONE XR16L788



2A) Typical Connections between the CPU and Two ST16C454/554/654/854/864's



2B) Typical Connections between the CPU and One XR16L788

* Since there are eight chipselects and eight interrupts, external glue logic is essential.

†One of the chipselect outputs from the CPU can usually be used to drive the CS# pin of the XR16L788.

4.0 SOFTWARE DIFFERENCES

The ST16C454/554/654/854/864 as well as the XR16L784/788 have 16550 compatible registers. In addition, the XR16L784/788 have the following enhancements/additions:

- Flat Register Set: Apart from the baud rate registers DLL and DLM, there are no shadow registers. The enhanced registers in the ST16C654 and XR16C854 are shadow registers and require that LCR = 0xBF before they can be accessed. This is not required for the XR16L784/788.
- Device Configuration Registers: These are newly added registers which provide means to configure or obtain status from all channels. These are:
    - Global Interrupt Registers INT0 - INT3: Provide detailed information on the channel(s) and the type of interrupt
    - TIMER Registers: Configure a general purpose Timer/Counter to issue one-time or periodic interrupts
    - 8X Mode: Double the data rate by using an 8X sampling clock instead of 16X sampling
    - RESET: Software Reset for all the channels
    - SLEEP: Place individual channels in sleep mode to conserve power consumption
    - DREV & DVID Registers: Read the Revision and Device ID of the UART (This feature is available through two shadow registers in the XR16C854/864, but not available in ST16C454/554/654)
    - REGB: Write to control registers of all channels simultaneously thereby simplifying the initialization routine

The advantages of these registers in the newer XR16L784/788 devices are best clarified using the following software routine examples. These examples show typical receive and transmit interrupt routines for the classic ST16C454/554/654/854/864 vs. the fifth generation XR16L784 or XR16L788:

4.1 INTERRUPT SERVICE ROUTINE FOR ST16C454/554/654/854/864:

```
Interrupt_Routine_Classic_QuadUART () {
    Disable_Interrupts();
    status_channel_A = read (ISR_channel_A);
        switch (status_channel_A)        {
                case RXRDY :
                        Read bytes out of RHR;
                        break;
                case TXRDY :
                        Load bytes into THR;
                        break;
                Other cases :
                ......
                case No_Interrupt :
                        break;              // go to next channel
        }       // end switch statement for channel A

    status_channel_B = read (ISR_channel_B);
        switch (status_channel_B)  {
                case ....
                ........
        }       // end switch statement for channel B
        ......
        ......
        Repeat the above for channels C and D

    Enable_Interrupts();
}       // end of Interrupt_Routine
```

As is clearly seen above, the interrupt routine is inefficient since it has to loop through all channels to determine the cause of the interrupt. Also, only one interrupt for the entire device can be serviced at once. If two of these devices are used, the scheme becomes even more inefficient. This inherent latency is not seen in the interrupt service routine for the XR16L784/788 because of the availability of global interrupt registers paving the way for servicing multiple interrupting conditions per interrupt.

4.2 Interrupt Service Routine for XR16L784/788

```
Interrupt_Routine_788 () {
        Disable_Interrupts();
        source_channel = read (INT0);   // bit-0 = 1 indicates an interrupt pending for channel 0, etc

        /* 3-bit encoding per channel making it 12-bits for the XR16L784 and 24-bits for XR16L788 */
        /* In this encoding, bits 2:0 are for channel 0, bits 5:3 are for channel 1 etc. */
        /* or you can read the same info from ISR register of the channel - but the encoding IS NOT THE SAME.
                Please read the datasheet of the XR16L784 or XR16L788 for more information */

        long interrupt_info = (  ( (read (INT3) << 8) | read (INT2)) << 8) | read (INT1);   // 24 bits for the XR16L788
        if (source_channel & 0x01)  {                   // bit-0 is channel 0, bit-7 is channel 7 etc.

                switch (interrupt_info & 0x7)                   {
                        case RXRDY :
                                Read bytes out of RHR;
                                break;
                        case TXRDY :
                                Load bytes into THR;
                                break;
                        Other cases :
                        ......
                        case No_Interrupt :
                                break;          // go to next channel
                }       // end switch statement for channel 0
        }       // end if (source_channel & 0x01)

        if (source_channel & 0x02)              {
                switch (interrupt_info & 0x38)  {       // bits 5:3 of the 12 or 24-bit value for channel 1
        ......
        .....
        }

        ......          // service all channel interrupts similarly

        Enable_Interrupts();
}       // End of Interrupt_Routine()
```

The above scheme provides the following advantages:

- Global interrupt registers support quicker interrupt source identification.
- Interrupts from multiple channels can be serviced per interrupt, according to the priority assigned to each channel.
- Shorter and deterministic time inside the interrupt service routine due to fewer reads of status registers (a maximum of 4 registers, INT0-INT3 as against a maximum of 8 ISR registers of individual channels).

5.0 SUMMARY

From the above discussion, it is clear that the XR16L784/788 devices have many advantages from both hardware and software point of view. To sum up, these devices deliver higher performance through the following:

- higher integration (one chip 788 vs. two chips 554/654/854's)
  - less board space
- simpler interface with the CPU
  - single chipselect
  - single interrupt to the CPU
- ability to interface with both 3.3V and 5V devices
  - 5V tolerant inputs (all inputs except XTAL1)
- less CPU bandwidth requirement (faster interrupt parsing)
  - Global Interrupt Registers for quicker interrupt service