CIRRUS LOGIC®

CL-CD1864

Product Bulletin

Intelligent Eight-Channel Communications Controller

FEATURES

- **Eight full-duplex asynchronous channels**
- **Sophisticated interrupt schemes**
  - — Vectored interrupts
  - — *Fair Share* ™ interrupts
  - — *Good Data* ™ interrupts for improved throughput
  - — Simultaneous interrupt requests for three classes of interrupts: Rx, Tx, and modem state changes
- **Independent baud-rate generators for each transmitter and receiver to support data rates of up to 64 kbps**
- **Improved host/controller software interface**
- **Generation and detection of special characters**
- **On-chip flow control**
  - — In-band (Xon, Xoff generation and detection)
  - — Out-of-band (DTR/DSR or RTS/CTS)
- **On-chip FIFO — 8 bytes each for Rx, Tx, and status FIFOs**
- **Line break detection and generation**
- **Multiple-chip daisy-chain cascading feature**
- **Odd, even, forced, or no parity**
- **Five modem/general-purpose I/O signals per channel**
- **System clock up to 25 MHz (x2), 12.5 MHz (x1)**
- **Low-power CMOS technology in a 100-pin PQFP**

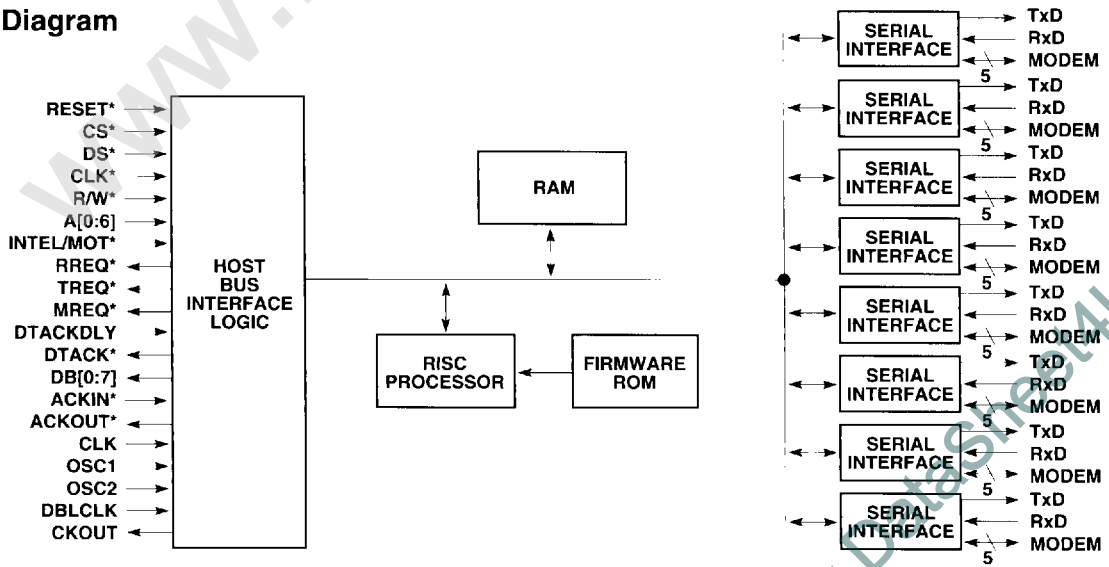
OVERVIEW

The CL-CD1864 is an I/O coprocessor capable of controlling eight full-duplex channels transferring data at rates up to 64 kbps. The advantage of the CL-CD1864 lies in its ability to move data efficiently from the serial channels to the host. This results in an order of magnitude improvement in system-level throughput and a reduction in overhead on the host CPU.

To increase the overall data throughput of the system, the chip relies on a combination of features. Most important are the buffers for transmit and receive data. Each serial channel has three 8-byte FIFOs — one each for transmit, receive, and receive exception status. The receive FIFOs have programmable thresholds to minimize interrupt latency requirements.

(cont. next page)

Functional Block Diagram



CIRRUS LOGIC

CL-CD1864

Eight-Channel I/O Coprocessor

OVERVIEW (cont.)

The CL-CD1864 is based on a high-performance proprietary RISC processor architecture developed by Cirrus Logic specifically for data communication applications. This processor executes all instructions in one clock cycle, and uses a register window architecture to ensure zero-overhead context switch for each type of internal interrupt.

The CL-CD1864 is fabricated in an advanced CMOS process. The chip's high throughput, low power consumption, and high level of integration permit system designs with minimum parts count, maximum performance, and maximum reliability.

Theory of Operation

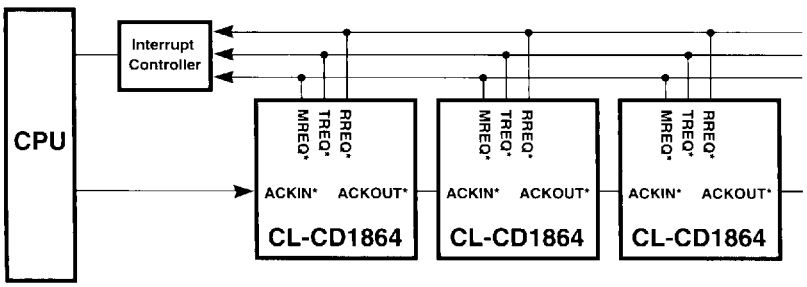
The CL-CD1864 custom RISC processor is assisted in its task by specialized peripheral logic. Serial data transmission and reception is handled by 'bit engines'. Each channel has a bit engine for transmit and another for receive. While each engine handles all bit-level timing, bit-to-character assembly is done in firmware. Bits are passed to the processor by internal interrupts, over a special bus dedicated to this purpose. To reduce internal interrupts to zero, special interrupt context hardware points to the correct register window for every possible context. A unique Global Index register eliminates address calculations by always pointing to the current channel.

The processor assembles bits into characters, checks parity and other formatting parameters, and stores the data in the FIFOs as required. FIFOs are maintained as RAM-based structures. Both the local processor and the host access them via Pointer registers, in effect an Indexed Addressing mode.

The CL-CD1864 communicates with the host via service requests and service acknowledgments. Service requests can be detected via interrupt lines or by on-chip registers. Regardless of the method used, the CL-CD1864 has features to minimize both the number of requests to be serviced and the time required to service them. FIFOs help reduce the number of service requests to one every eight characters. To reduce the time required per request, the CL-CD1864 supplies separate vectors for four different types of service requests. This reduces the time required by the processor to effect the proper operation. For instance, there is a unique vector for 'good data', so that the host wastes no time checking status bits or error conditions. If there is an error condition, the CL-CD1864 supplies a unique vector pointing to the error-handling routine. Other vectors report transmit status and modem signal change.

Interrupts can be acknowledged either by an Interrupt-Acknowledge pin, or by reading an on-chip register. This allows host software maximum flexibility and speed in handling service requests.

Because the CL-CD1864 CPU is processing every character sent or received, features such as automatic flow control and special character recognition are easily implemented. This further reduces the processing burden on the host system. Both In-Band (Xon, Xoff) and Out-of-Band (RTS/CTS, DTR) Flow-Control modes are supported. For in-band flow control, the CL-CD1864 automatically starts and stops its transmitter when the remote unit sends flow-control characters. The CL-CD1864 also makes it easy for the local host to flow-control the remote, via the 'send special character' commands. For out-of-band flow control, the transmitter will optionally assert RTS and monitor CTS for permission to send; and assert/negate DTR when the Receive FIFO reaches a user-

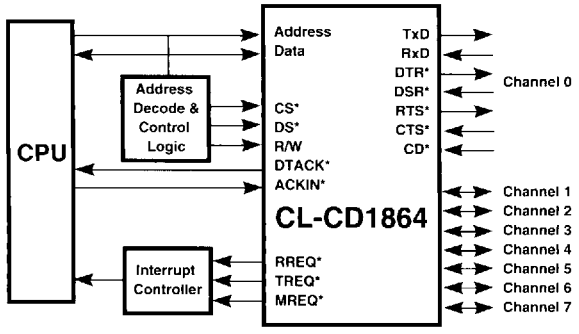


CL-CD1864 in Daisy Chain Scheme

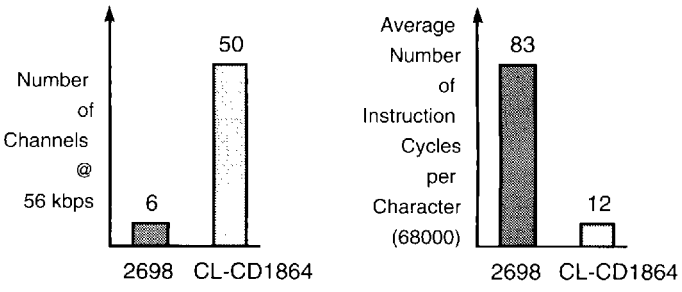
CL-CD1864

Eight-Channel I/O Coprocessor

CIRRUS LOGIC



Typical CL-CD1864 CPU Interface



CL-CD1864 Performance

definable threshold. Together, the in-band and out-of-band features not only allow the data flow to be controlled in real time with minimum or no host intervention, it also prevents loss of data.

As shown above, the CL-CD1864 can interface virtually any CPU, with a minimum of glue logic. Refer to the CL-CD1864 Data Sheet for detailed information on how to interface various microprocessors. Systems with multiple CL-CD1864s are easily implemented, with no external glue, via a daisy-chain scheme. A 'fair share' feature ensures equal access for all service requests, both within one CL-CD1864 and across multiple devices.

FIFO — 24 bytes of FIFO are dedicated to each channel partitioned as 8 bytes for transmitter, 8 bytes for receiver, and 8 bytes for status. The receive FIFO has a user-programmable threshold to optimize system response and latency. The receive FIFO threshold programming range is from 1-8 characters.

Vectored Interrupt Structure — Three interrupt signals ([R, T, M]REQ*) are used. These signals may also be read as an on-chip register. Each REQ* signal represents one of three interrupt groups: receive, transmit, and modem signal state changes. Upon servicing by the host, an interrupt vector will be generated by the CL-CD1864 to define the interrupt group to be serviced and which CL-CD1864 generated the interrupt. This allows the host software to enter directly into the proper interrupt service routine, reducing the amount of interaction between the host and the controller, and determining the nature of the interrupt.

Good Data Interrupt — If data received is all good, the host is advised of the number of good data bytes in the FIFO, allowing the host to read data without further status queries until all good data has been transferred.

Fair Share Interrupt Scheme — To ensure equal service of all channels, a fair share scheme is used for each interrupt group. No channel can interrupt for the same condition until all others have a chance to be serviced for the same interrupt condition.

System design is simplified in the CL-CD1864 by providing a choice of crystal or external clock operation, at 1x- or 2x-rated frequency. The software-based polling and acknowledgment features eliminate external glue logic.

CL-CD1864 Features/Benefits

Unique Features	Benefits
■ ***Three 8-byte FIFOs per channel***	Greatly reduces real-time interrupt response time requirement of the CPU. Simplifies system tasks in a real-time multi-tasking environment.
■ ***User-programmable receive FIFO threshold***	Enables tailoring of interrupt conditions to different system requirements. Speeds software development.
■ ***Data Interrupt for transferring multiple bytes of data***	Reduces host time required to transfer data and significantly improves system performance. 'Frees-up' bandwidth for host to perform higher-level system tasks.
■ ***Interrupt vectoring by device ID and type of service required***	Permits direct jump into proper interrupt service routine, improving overall system performance.
■ ***Automatic flow control***	Real-time control of data flow reduces risk of losing valuable data.