HP Archive

Advanced Micro Devices

HEWLETT PACKARD

Developing and Debugging an ISDN Terminal Adapter

Jean Anne Booth

Advanced Micro Devices
5900 East Ben White Blvd., MS 561
Austin, Texas 78741
Tel: (512) 462-5879
Fax: (512) 462-5051

1993
High Speed Digital
Symposium

Advanced Micro Devices

HEWLETT PACKARD

Abstract

The Integrated Services Digital Network digitizes voice signals at the telephone and sends both voice and control signals to the PABX or central office switch digitally. An ISDN terminal adapter interfaces between ISDN and non-ISDN equipment, typically at the subscriber loop level that interfaces between the customer's equipment and the local telephone network office. This presentation introduces ISDN, describes an ISDN terminal adapter, and shows how to implement an ISDN terminal adapter in hardware and software. Using an evaluation board, PC, JTAG emulator, and HP 16500 logic analyzer, the presentation also shows how to develop and debug the ISDN terminal adapter presented here.

Author

Jean Anne Booth

Current Activities:
Jean Anne Booth is a Senior Technical Marketing Engineer at Advanced Micro Devices. She has been with AMD for 6 years, and is currently responsible for technical marketing of current and future high performance 29K™ RISC microprocessors.

Background:
Prior to joining 29K Marketing, she managed the 29K Technical Support Center, providing hardware and software technical and applications support to 29K Family customers. Before joining AMD, Jean Anne was a development engineer involved in the software implementation of real-time control systems. Jean Anne holds a BS in Electrical Engineering and an MS in Computer Engineering.

Slide #1

Developing and Debugging an ISDN Terminal Adapter

Advanced Micro Devices

Slide #2

Developing and Debugging an ISDN Terminal Adapter

- Introduction to ISDN
- An ISDN Terminal Adapter
  - Am79C30A/32A Digital Subscriber Controller (DSC)
  - Am29200™ 32-bit RISC microcontroller
  - Am85C30 Serial Communications Controller (SCC)
- Developing an ISDN Terminal Adapter
  - Hardware
  - Software
- Debugging the ISDN Terminal Adapter
  - PC, JTAG emulator, HP 16500 logic analyzer, SA-29200 evaluation and expansion boards

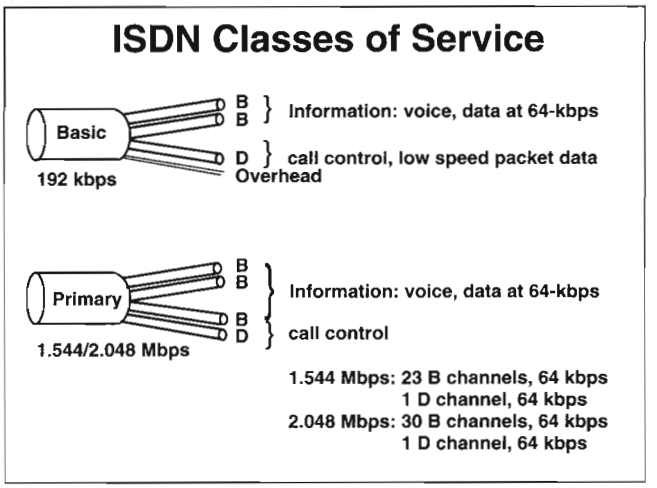
Slide #3

The Integrated Services Digital Network (ISDN)

- All-digital network standard: voice, data, control signals
- Replaces subscriber loop with digital voice and data capability
- Global communications network benefits:
  - increased reliability
  - increased functionality
  - lower cost
  - worldwide standardization

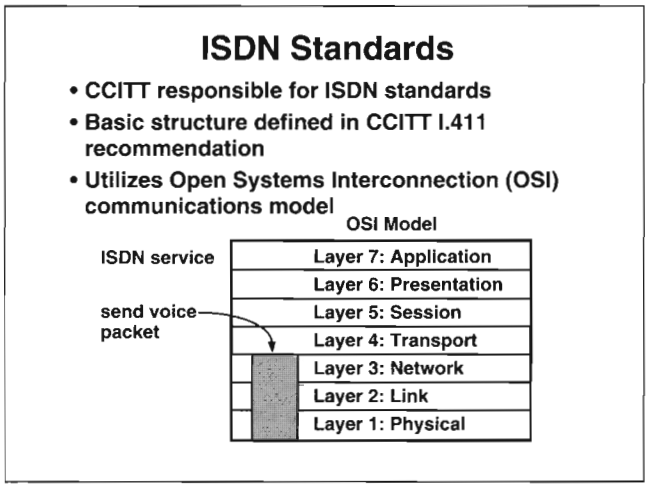
The Integrated Services Digital Network, or ISDN, digitizes voice signals at the telephone and sends both voice and control signals to the PABX or central office switch digitally. Thus, ISDN replaces the last analog component of telephone circuitry, the subscriber loop, with a digital component capable of handling both voice and data information. ISDN brings the benefits of digital technology — increased reliability, new functionality, lower cost, and increased security and privacy — and worldwide standardization to the global communications network.

Slide #4



ISDN service is divided into two classes – *primary rate*, an expensive high-bandwidth connection, and *basic rate*, the type of subscriber connection most commonly used. The focus here is on the basic rate service, which provides three communications channels. The two B (or bearer) channels provide either voice or data service at 64 kbps. The D (or signaling/data) channel provides call control services and low speed packet data transmission (up to 9600 bps). Primary rate service is provided at either 1.544 Mbps (US, Canada, and Japan) or 2.048 Mbps (Europe). The channel structure for the 1.544-Mbps rate is typically 23 B channels and one D channel; the 2.048-Mbps rate is typically composed of 30 B channels and one D channel.

Slide #5



The group overseeing the definition of ISDN is the CCITT, and the basic structure of the ISDN is specified in the CCITT I.411 recommendation. ISDN functions are further subdivided by the OSI (open systems interconnection) seven-layer communication model. The OSI model defines physical and logical services provided by each layer; a vertical "slice" of the model encompassing at least layers 1 through 3 provides one ISDN function, such as a user data transfer.

Slide #6

CCITT Recommendations for ISDN

I Series

- Complete set of recommendations for all standardization aspects of ISDN
- Cross references specifications from other series (Q Series for protocols, and V/X series for non-ISDN terminals)

Slide #7

CCITT Recommendations for ISDN

I Series

I.420/I.421 – Introduction to ISDN concepts and other I Series recommendations

I.430 – Layer 1 interface specification ('S' and 'T' interface recommendations)

I.431 – Layer 1 primary rate interface

I.440/I.441 – Layer 2 protocol specification (LAPD) Cross references Q.920/Q.921

I.450/I.451 – Layer 3 protocol specification. Cross references Q.930/Q.931

The CCITT I Series of recommendations is a complete set of recommendations for all aspects of ISDN. The I Series recommendations cross-reference specifications from other series, like the Q Series for protocols, and the V and X series for non-ISDN terminals. CCITT has defined many more I Series recommendations than the ones listed here.

Slide #8

CCITT Recommendations for ISDN

Q Series

Q.920/Q.921 – Defines the Layer 2 protocol (LAPD) used by ISDN

– Ensures error-free and correctly sequenced data transmission between Layer 3 entities

– Specifies syntax of message format used within HDLC frames at Layer 2

The Q.920/Q.921 recommendation defines the Layer 2 protocol used by ISDN, also known as LAPD. The LAPD protocol ensures error-free and correctly-sequenced data transmission between Layer 3 entities. The services provided by the LAPD protocol include both unacknowledged and acknowledged information transfer on the ISDN D-channel. The Q.920/Q.921 recommendation also specifies the syntax of the message format used within HDLC frames at Layer 2.

Slide #9

CCITT Recommendations for ISDN

Q Series

Q.930/Q.931 – Defines internationally agreed portion of Layer 3 (network layer) protocol for ISDN

– Provides packetizing and blocking of Layer 4 messages for Layer 2 conformance

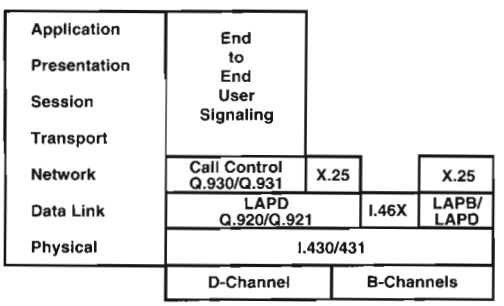
– Does not address supplementary service (defined by national committees)

The Q.930/Q.931 recommendation defines the internationally agreed-upon portion of the Layer 3 protocol used by ISDN. It details packetizing and blocking of Layer 4 messages for Layer 2 conformance, but does not address supplementary services, such as call waiting, call transfer, credit card calling, etc. Supplementary services are defined by national committees.

Slide #10

CCITT Recommendations for ISDN

Applicability of I/Q Series recommendations



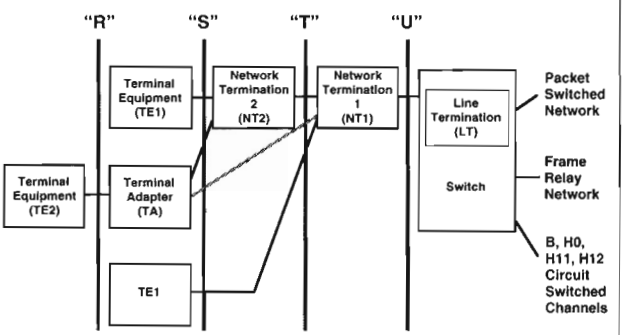
This is another look at the relationship between OSI and ISDN. As a network, ISDN is primarily unconcerned with layers 4 through 7, which you employ for exchanging information. Layer 1, defined in I.430 and I.431, specifies the physical interface for both basic and primary rate access. Because both the B and D channels share the physical interface, these standards apply to both types of channels. Above this layer, the protocol structure differs for the two types of channels.

For the D channel, the LAPD protocol defined in Q.920 and Q.921 is employed in the data link layer. For the B channel, the I.46X series of recommendations defines alternative protocols for interfacing existing equipment to the ISDN. Because the B channel can also be packet-switched, the LAPD protocol can also be used, in addition to using the more common LAPB protocol for data link layer transfers.

At the network layer, Q.930 and Q.931 define call control for the D channel. If the D channel is used to provide packet switching services, the X.25 level 3 protocol is used. For the B channel, the X.25 level 3 protocol provides network layer services for packet switching.

Slide #11

ISDN Network Topography



The network topography from the desktop to the switch is shown here, identifying certain classes of equipment that make up the network. "Reference points" are defined that represent various interfaces with CCITT standards for both hardware and software. In developing a terminal adapter, we will be dealing with the R and S/T interfaces.

Slide #12

ISDN Terminology

- **TE1:** ISDN compatible voice and/or data terminal
- **TE2:** Non-ISDN compatible terminal (such as V.24, X.21, X.25, SNA terminal
- **TA:** Terminal adapter providing physical and/or protocol conversion between a TE2 and the ISDN
- **NT2:** Network termination providing switching and/or concentration (such as PBX) – not present in single line installations

In the ISDN network topography, a TE1 (terminal equipment type 1) refers to devices that support the standard ISDN interface, such as digital telephones, integrated voice/data terminals, or digital facsimile machines.

A TE2 (terminal equipment type 2) refers to any non-ISDN compatible terminal, typically existing equipment. Examples of TE2 are terminals with an RS-232 interface, host computers with an X.25 interface, and SNA terminals. TE2 devices require a TA (terminal adapter) to plug into an ISDN interface. TAs may provide physical conversions, protocol conversions, or both.

All terminal equipment, whether TE1 or TE2, provides protocol handling, maintenance functions, interface functions, and connection functions to other equipment.

An NT2 (network termination 2) is an intelligent device that provides switching and/or concentration functions. Examples of NT2s include digital PBX, terminal controllers, and LANs. An NT2 performs layer 2 and layer 3 protocol handling, layer 2 and layer 3 multiplexing, maintenance functions, and interface termination, in addition to switching and concentration.

Slide #13

ISDN Terminology

- **NT1:** Network termination providing physical and/or protocol conversion between the 'S'/'T' interface and the network-provided 'U' interface
- **LT:** Line termination performing physical and/or protocol conversion between 'U' interface and central office exchange internal highways

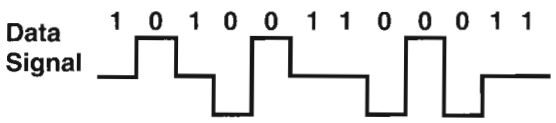
An NT1 (network termination 1) provides physical and electrical termination of the ISDN. It may also provide protocol conversion between the S/T interface and the network's U interface. The NT1 may be controlled by the ISDN provider, and forms a boundary to the network. The functions provided by an NT1 include line transmission termination, line maintenance and performance monitoring, timing, power transfer, layer 1 multiplexing, and interface termination, including multidrop termination employing layer 1 contention resolution.

An LT (line termination) provides physical and/or protocol conversion between the U interface and the provider's network.

Slide #14

Characteristics of 'S' and 'T' Reference Points

- 4 wire interface
- 192 kbps full duplex
  - 48 bit frame each 250 msecs
- Optional remote power feed
- Pseudo-ternary line coding



The S and T reference points contain a 4-wire interface with optional remote power feed. Two 64-kbps B channels and one 16-kbps D channel produce a load of 144 kbps, and they are multiplexed over the 192-kbps S or T interface. The remaining capacity of 48 kbps is used for framing and synchronization overhead.

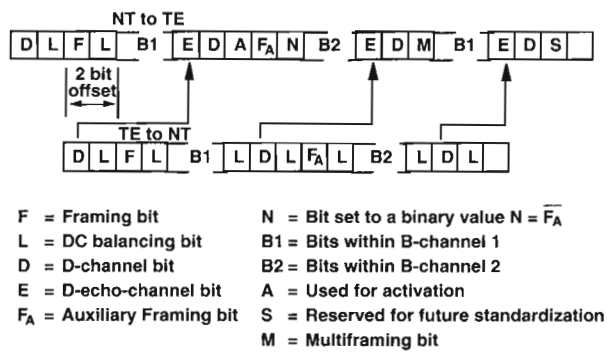
The synchronous time-division multiplexed (TDM) scheme used by the S and T interfaces is composed of 48-bit frames transmitted at a rate of one every 250 microseconds.

To prevent loss of synchronization and signal degradation, pseudo-ternary line encoding is used at the S and T interfaces. In pseudo-ternary line encoding, a binary 1 is represented by no line signal and a binary 0 is represented by a positive or negative pulse. The binary 0 pulses must alternate in polarity to prevent signal degradation.

Slide #15

'S' Interface Frame Structure

48 Bits in 250 Microseconds



F = Framing bit
L = DC balancing bit
D = D-channel bit
E = D-echo-channel bit
F_A = Auxiliary Framing bit
N = Bit set to a binary value N = $\overline{F_A}$
B1 = Bits within B-channel 1
B2 = Bits within B-channel 2
A = Used for activation
S = Reserved for future standardization
M = Multiframing bit

The S interface frame structure, 48 bits repeated at a rate of one frame every 250 microseconds, includes 16 bits from each of the two B channels and 4 bits from the D channel. The upper frame is transmitted from the network (NT1 or NT2) to the terminal equipment (TE); the lower frame is transmitted from the terminal equipment to the network. The frame from a TE to NT follows the frame from NT to TE by 2 bit-times.

Consider the frame from TE to NT first. Each frame begins with a framing bit (F) that is always transmitted as a positive pulse, followed by a dc balancing bit (L) that is a negative pulse to balance the voltage. This F-L pattern synchronizes the receiver on the beginning of the frame. After the synchronization, the first zero bit will be encoded as a zero, and then pseudo-ternary encoding rules are followed for the remaining bits.

The next eight bits (B1) are from the first B channel. This is followed by another dc balancing bit (L). Next is one bit from the D channel and its dc balancing bit. This is followed by the auxiliary framing bit (F_A), which is set to zero unless it is being used in a multiframe structure. Another balancing bit (L) follows, then eight bits from the second B channel (B2), and another balancing bit (L). This entire sequence is duplicated again to transmit another eight bits from the first B channel, a single D channel bit, another eight bits from the second B channel, and another D channel bit, with balancing bits following each group of channel bits.

The frame structure in the NT to TE direction is similar, except that some of the dc balancing bits are replaced by D-channel echo bits (E), which are a retransmission by the NT of the most recently received D bit from the TE. The echo bits are used to provide D-channel contention resolution in a network with multiple TEs in a passive-bus configuration. The activation bit (A) is used to activate a TE (bring it on-line). The N bit is set to one unless it being used in a multiframe structure. The M bit indicates a multiple frame. Multiframing is defined in I.430, and consists of 20 frames as defined here, where the auxiliary framing bit carries the Q multiframe data in the TE to NT direction, and S carries the Q multiframe data in the NT to TE direction.

Slide #16

Characteristics of the 'R' and Reference Point

The 'R' reference point alows non-ISDN terminals to be connected to the ISDN.

Standard	Protocol
V.120	LAPD packet protocol
DMI Mode 2	Bit stuffing
DMI Mode 3	LAPD/X.25
V.110/ECMA 102	Bit stuffing

The R reference point allows non-ISDN terminals to be connected to the ISDN. Some common non-ISDN interfaces and their protocols are listed here.

Slide #17

Terminal Adaption

- **Bit stuffing – ECMA102/V.110, DMI Mode 2**
  - **Low cost**
  - **No error detection/re-transmission**
- **Packetizing – V.120, DMI Mode 3**
  - **Error detection/re-transmission**
  - **Statistical multiplexing**
  - **Higher throughput**

An ISDN terminal adapter is the interface between ISDN and non-ISDN equipment, typically at the subscriber loop level that interfaces between the customer's equipment and the local telephone network office.

An ISDN data-only terminal adapter will interface either a bit-stuffing protocol or a packetizing protocol to the ISDN. A terminal adapter for bit-stuffing terminals has the advantage of low cost, but also has no error detection or retransmission. Terminal adapters for packetizing protocols are more expensive but also more reliable, with error detection and retransmission included in the protocol conversion. In addition, packetizing systems have a higher throughput.

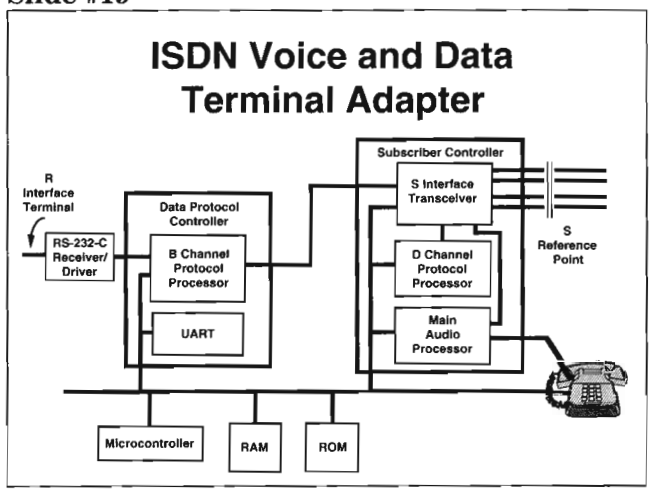
Slide #18

Terminal Characteristics

- High volume products
- Severe cost constraints
  - Low component cost
- Power constraints for voice products
  - Efficient CMOS devices
- High feature content
  - Complex software, powerful MPU
- Switch specific versions
  - Multiple software variants

An ISDN terminal adapter is a high-volume product because the installed base of non-ISDN compatible equipment is nearly the size of the entire installed terminal market. Like all high-volume products, they operate under severe cost constraints, requiring a low component count to maintain competitiveness with other vendors. Voice terminal adapters also have a strict power budget, so solutions with power-efficient CMOS devices are required. Because interfacing a non-ISDN terminal to the ISDN usually involves adding ISDN features that aren't a part of the existing analog solution, terminal adapters require powerful processors to implement these new features in software. To broaden the market potential for an adapter, manufacturers prefer to create specific versions of the same basic adapter that differ only in software to handle different protocols or implement required ISDN features not present in the non-ISDN terminal.

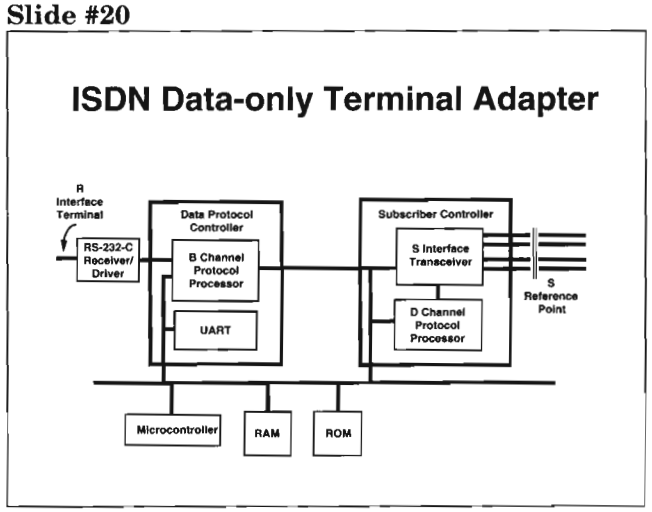
Slide #19



This shows a block diagram of an ISDN terminal adapter.

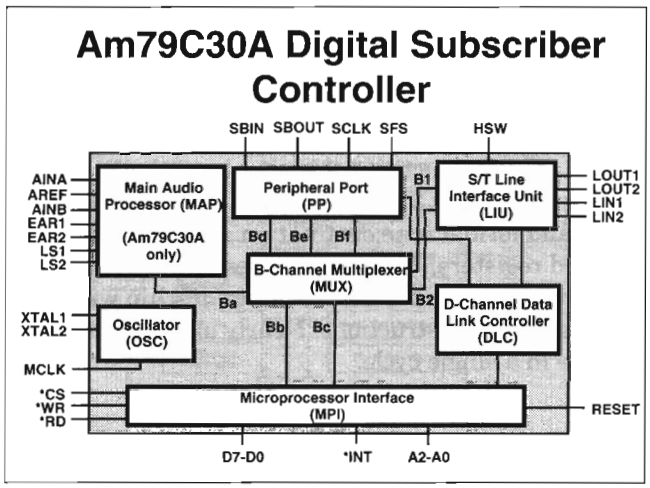
Our case study will build a terminal adapter using an Am79C30A Digital Subscriber Controller (DSC) for basic ISDN services and an Am29200™ RISC microcontroller for control of the DSC and ISDN protocol. An Am85C30 Serial Communications Controller (SCC) provides the serial interface to the R reference point. Using an Am79C30A allows ISDN-compatible transmission of both voice and data.

Slide #20



An ISDN data-only terminal adapter could be built using an Am79C32A ISDN Data Controller (IDC) instead of the Am79C30A, as shown here.

Slide #21



The Am79C30A/32A provides a 192-kbps full duplex digital path over four wires between the TE (terminal equipment) located on the subscriber's premises and the NT (network terminal) or PABX linecard. All physical Layer functions and procedures are implemented, including framing, synchronization, maintenance, and multiple terminal contention. The Am79C30A/32A processes the ISDN basic rate bit stream. The B channels are routed to and from different portions of the DSC under software control. The D channel is partially processed and then passed to the microcontroller for further processing.

This is a block diagram of the Am79C30A DSC. The main audio processor, or MAP, is the only portion of the Am79C30A that is not present on the Am79C32A. The MAP uses DSP to implement a high performance codec/filter function. The MAP supports a loudspeaker, an earpiece, and two separate audio inputs. Gain, frequency response, and tone generation are programmable.

The S/T line interface unit, or LIU, provides the interface to an ISDN S or T reference point. It contains a hook-switch input and differential subscriber line inputs and outputs. The LIU monitors the S interface and hook switch during power down, allowing the microprocessor to be shut down to conserve power during idle periods.

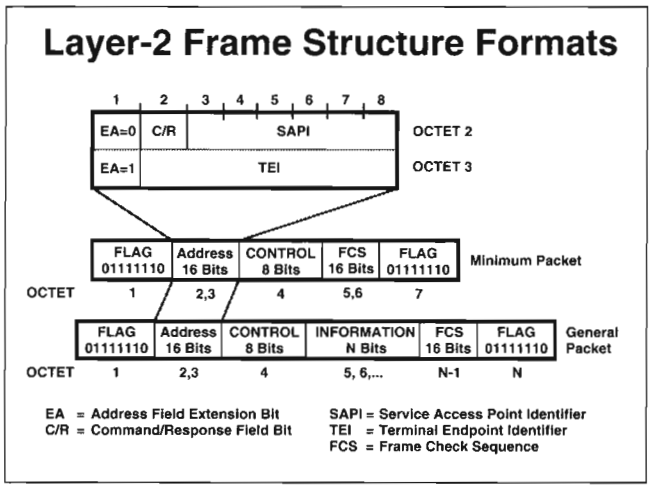
The microprocessor interface unit (MPI) communicates with the processor controlling the terminal adapter. The address line inputs select source and destination registers for read and write operations on the data bus. The data bus is used to exchange information with the controlling processor. An interrupt input informs the processor that the DSC needs service, and chip select and read/write signals are provided for system interfacing.

The B-channel multiplexer routes the 64-kbps full-duplex B channels between the LIU, MAP, MPI, and peripheral port. Routing control is programmed by the microcontroller.

The 16-kbps D channel is time multiplexed within the frame structure of the S interface. The data carried by the D channel is encoded using the Link Access Protocol D-channel (LAPD) format shown here. The LIU controls the multiplexing and demultiplexing of the D-channel data between the S interface and the D-channel data link controller (DLC).

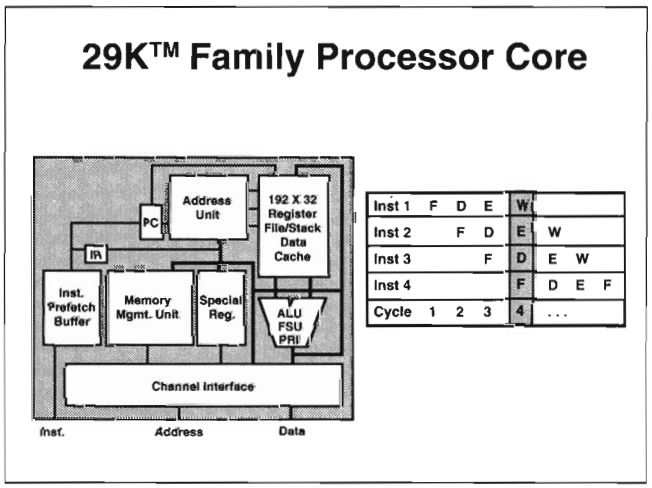
The Am79C30A/32A will generate a maximum of one interrupt every 125µs. Once asserted, the interrupt will remain asserted until the microcontroller reads the DSC's interrupt register. Events that generate interrupts include DLC receive FIFO full, DLC transmit FIFO empty, LIU change of state (on hook/off hook), packet errors, and packet status (last byte, etc.).

Slide #22



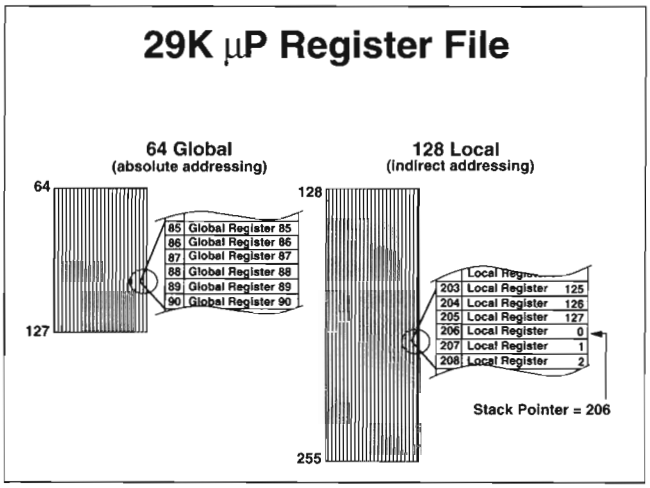
The DLC performs processing of Layer-1 and partial Layer-2 LAPD protocol, which includes flag detection and generation, zero deletion and insertion, and Frame Check Sequence (FCS) processing for error detection. Higher-level protocol processing is done by the external microcontroller. The DLC contains two 8-byte data FIFOs for receive and transmit data, and three 2-byte status FIFOs that make it possible to receive two back-to-back data packets.

Slide #23



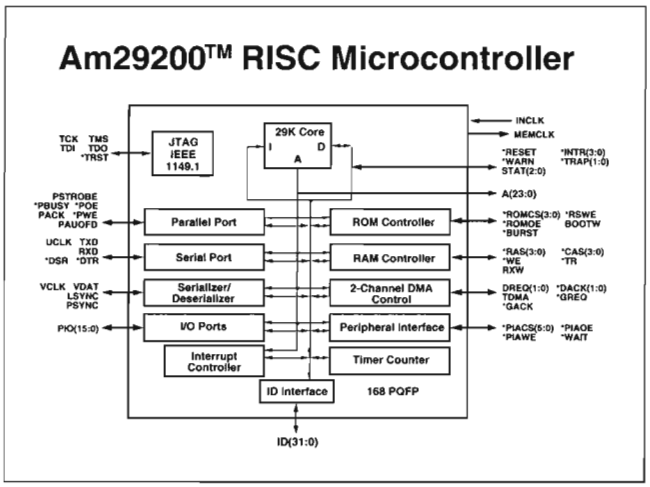
The Am29200 32-bit RISC microcontroller contains a 29K™ Family 32-bit RISC processor core, memory controllers, and integrated peripherals. The 29K processor core is shown here. The processor core contains a 4-stage instruction/execution pipeline, separate 32-bit instruction, address, and data buses, 192 32-bit general purpose registers, a 56-bit timer/counter, and special registers for processor control. The instruction set is simple, with all instructions being 32-bits in length and using a 3-operand format (one destination register and two operand registers). The only addressing mode is register-direct; data is moved on- or off-chip with LOAD and STORE instructions. Nearly all instructions execute in a single cycle.

Slide #24



The 192 general-purpose registers are split into two groups, 64 global registers and 128 local registers. The local registers are organized internally as a circular queue. The register pointed to by global register 1 (**gr1**) is local register 0 (**lr0**); the register below **lr0** is **lr1**, and the register above is **lr127**. The local registers are used by compilers to implement dynamic register windowing, and are controlled by software convention.

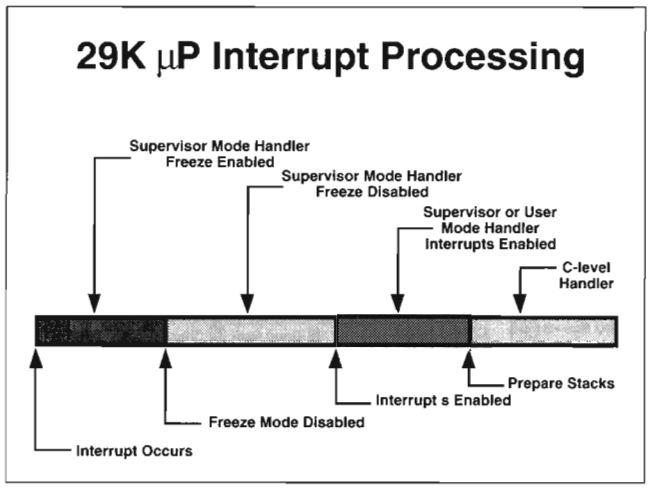
Slide #25



This is a block diagram of the Am29200 microcontroller. The ROM controller and DRAM controller implement a glueless interface to memory. The DMA controller provides two channels for transfer of data between the DRAM and internal or external peripherals. The peripheral interface adapter (PIA) implements a glueless interface to up to six generic peripherals, and will be used to connect to the DSC and SCC. The I/O port provides 16 programmable signals that can be outputs, inputs, or interrupt triggers. The 5-pin JTAG port is used for test and debug control of the processor.

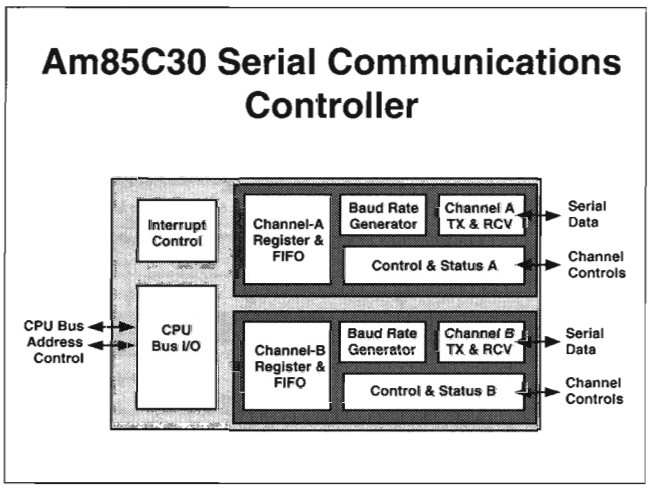
Internal peripherals in the Am29200 microcontroller are addressed with LOAD and STORE instructions using a pre-defined memory mapped address. In addition, special registers are programmable for control and status for each peripheral.

Slide #26



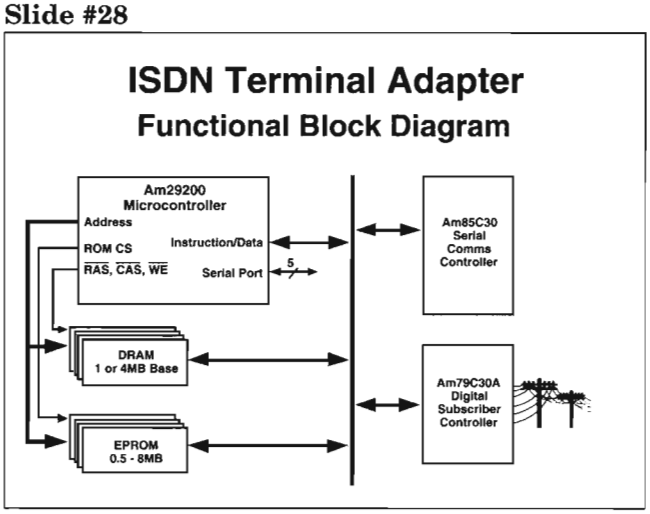
Interrupts in the Am29200 microcontroller are triggered by external interrupt pins, programmed pins in the I/O port, and internal sources, such as the DMA controller or software traps. When an interrupt occurs, the processor completes or cancels current bus activity, saves the Current Processor Status (**cps**) register into the Old Processor Status (**ops**) register, sets itself in Supervisor mode and freezes other processor status (so as not to corrupt the interrupted application), decodes the interrupt, and fetches the first instruction of the applicable interrupt handler. In keeping with the RISC philosophy of machine simplicity, no state or status other than the **cps** is saved. At the software engineer's option, the interrupt handler may execute in Freeze mode, utilizing only registers set aside for this purpose, or it may save more state and utilize more of the processor's resources.

Slide #27



The Am85C30 Serial Communications Controller (SCC) is a dual channel multi-protocol data communications peripheral, handling both asynchronous and synchronous formats including SDLC/HDLC and BiSYNC. It contains two channels, and each channel has an independent oscillator, baud-rate generator, and digital-phase locked loop for clock recovery. The device is controlled by internal registers read and written by a microprocessor through the 8-bit data bus. The SCC interrupts its controlling processor for transmit complete, receive complete, and error conditions.

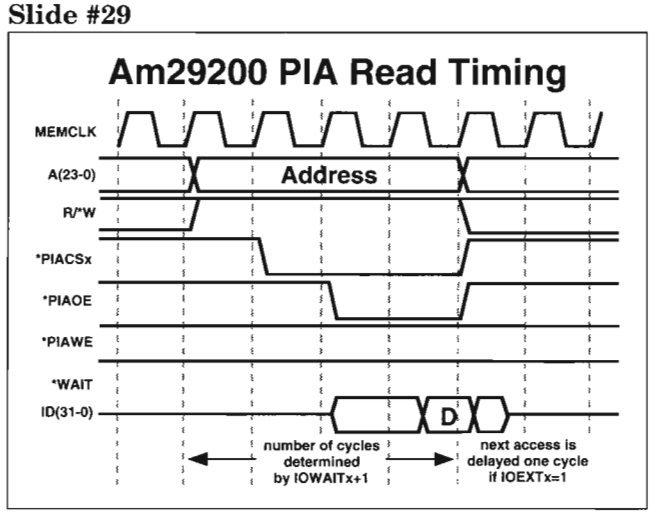
Slide #28



This block overview of the ISDN terminal adapter shows the SCC connecting to the ISDN R interface, the DSC connecting to the ISDN S/T interface, and the Am29200 RISC microcontroller providing control and error processing for both the SCC and DSC. Both the SCC and DSC will be connected to the Am29200 microcontroller through the microcontroller's PIAs.

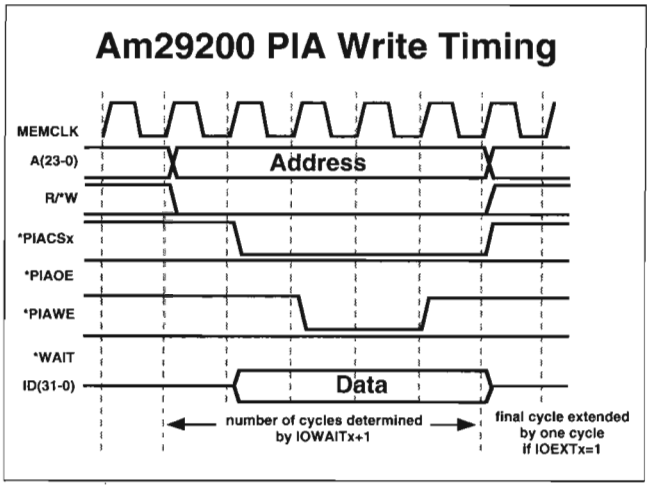
The DSC provides all voice and physical layer S interface functions and partial Layer 2 D-channel handling. The remainder of Layer 2 and all Layer 3 functions are provided by the microcontroller.

Slide #29



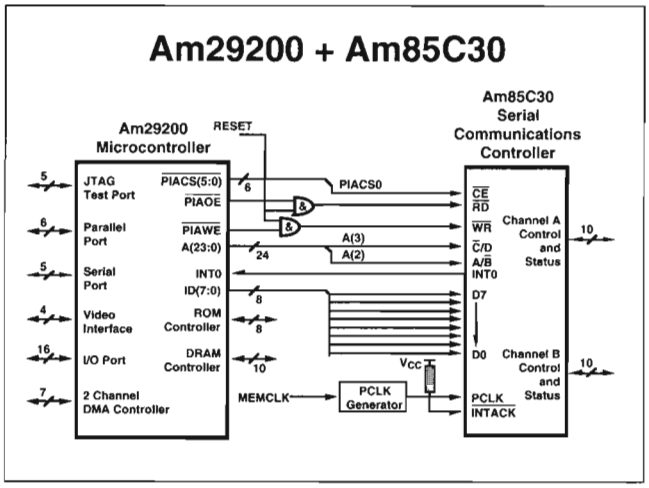
This is the timing of a PIA read cycle. The number of cycles until the PIA Chip Select (*PIACSX) and PIA Output Enable (*PIAOE) are deasserted is dependent upon the value of Input/Output Wait States (IOWAITX) field of the PIA control register in the microcontroller. The minimum access time for a PIA read is 3 cycles (2 wait states). If the Input/Output Extend (IOEXTX) field of the PIA control register is set, the next PIA access will be delayed one cycle for an additional cycle of output disable time.

Slide #30



This is the timing of a PIA write cycle. The number of cycles until the PIA Chip Select (*PIACSX) and PIA Write Enable (*PIAWE) are deasserted is dependent upon the value of Input/Output Wait States (IOWAITX) field of the PIA control register in the microcontroller. The minimum access time for a PIA write is 4 cycles (3 wait states). If the Input/Output Extend (IOEXTX) field of the PIA control register is set, the next PIA access will be delayed one cycle for an additional cycle of data hold time.

Slide #31



This is the interface between the Am29200 microcontroller and the Am85C30 SCC. Note that the SCC doesn't have an explicit RESET signal; to achieve a reset, the device expects *RD and *WR to be asserted at the same time. This interface shows an interrupt-driven communication mechanism between the microcontroller and the SCC. Assuming a 16-MHz microcontroller and an 8-MHz SCC, the SCC's PCLK is generated from the microcontroller's MEMCLK signal with a divide-by-2 circuit. A complicating factor is the access recovery time of the SCC. The minimum time from the leading edge of one command to the leading edge of the next command is defined as 3.5 times the PCLK cycle time. The access recovery time can be generated in software by inserting delay instructions, or in hardware by adding an external PAL with delay states or by using the microcontroller's WAIT signal. This design assumes that software assures the minimum access recovery time for the SCC.

Slide #32

Code Example for Am85C30

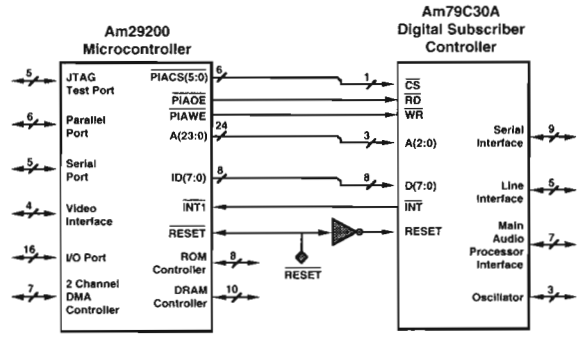
```
SerialSend:
        Save 6 Registers (gr96-gr101);
        Control = 0x90000007
        Data = 0x9000000F
        End = Start + Length -1;
        Current = Start;
$3:     store   0,1,gr98,Control        ; TxRdy?
        Delay_Macro
        load    0,1,gr100,Control
        Delay_Macro
        sll     gr100,gr100,31-2
        jmpf    gr100,$3
        nop
        load    0,1,gr100,Current       ; Send data
        store   0,1,gr100,Data
        Delay_Macro
        add     Current,Current,1
        cpleu   gr100,Current,End
        jmpt    gr100,$3                ; More?
        nop
        Restore 6 Registers (gr96-gr101)
        jmpi    lr0
        nop
```

```
.macro  Delay_Macro
        mfsr    gr98,lru
        mtsr    lru,gr98
        const   gr98,5
$1:     jmpfdec gr98,$1
        nop
.endm
```

When the SCC interrupts the microcontroller for service by asserting *INT0, the processor interrogates the SCC's status register RR2 to determine the source of the interrupt. This pseudo-code shows how serial information is sent using the SCC. Six microcontroller registers are saved before being used in the _SerialSend routine. The microcontroller sets the SCC control register for transmit mode, and then checks the SCC's response. When the SCC is ready to transmit, the microcontroller gets the address of the send buffer, and sends bytes until the message is complete. Delay_Macro consumes 6 cycles to meet the SCC's minimum access recovery time. Because the Am29200 microcontroller can overlap loads and stores in the pipeline for better performance, the delay macro includes a serializing instruction (MTSR, or move to special register), which won't allow loads and stores to overlap. The special register used in the serializing instruction doesn't matter; here special register **lru** (least recently used indicator) for the memory management unit is used.

Slide #33

Am29200 + Am79C30A



This is the interface between the Am29200 microcontroller and the Am79C30A DSC device. Note that the DSC RESET is an active-high signal, while the Am29200 microcontroller *RESET is an active-low signal.

Slide #34

Code Example
Signal Interrupt Handler

```
        .global _DSCInt

_DSCInt:
        ; Process signal
        const   l0,SIGUSR1      ; User defined signal number
        sub     msp,msp,4
        store   0,0,l0,msp      ; pass signal number
        sub     msp,msp,4
        store   0,0,gr1,msp     ; save rsp
        sub     msp,msp,4
        store   0,0,rab,msp     ; save reb
        sub     msp,msp,4
        const   l0,128*4
        sub     rab,rfb,l0      ; fix  rab
        pushsr  PC0,PC1,PC2,CHA,CHD,CHC,ALU,OPS   ; save specials
;
        mfsr    l0,OPS
        or0,l0,DI ; Ensure ints disabled
        mtsr    OPS,l0
;
        sub     msp,msp,4
        store   0,0,tav,msp     ; save tav
        mtsrim  chc,0           ; ensure loadm/storem does not restart
        mfsr    pc1, SignalRegister
        add     l2,SignalRegister,4
        mfsr    pc0,l2
        iret                    ; Goto sigcode and C handler, ints disabled
```

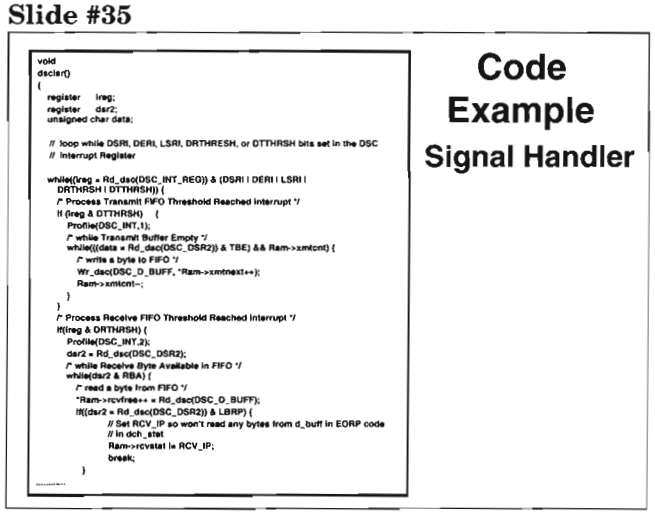
Ensures that Handler is executed with interrupts disabled.

The easiest method to handle the DSC's interrupts is to utilize C signals() and C interrupt handlers. This minimizes the amount of code that must be written by hand in assembly language and allows the developer to take advantage of TsLink3™ software from TeleSoft International, Inc. Written in

ANSI C, the TsLink3™ software provides the developer with a proven efficient solution through Layer 3 of the OSI model, and is compliant with ISDN guidelines for Q.391/X.25 protocols at Layer 3 and the Q.921 LAPD/LAPB protocols at Layer 2. TsLink3™ also includes the V.110 and V.120 rate adaption protocols with a command interpreter for the popular AT command set. Kits are available for the world's major switch specifications, including US National ISDN-1, AT&T 5ESS, Northern Telecom DMS-100, European ETSI NET3, French VN2 (with VN3 coming soon), German 1TR6, and Japanese NTT INSnet64.

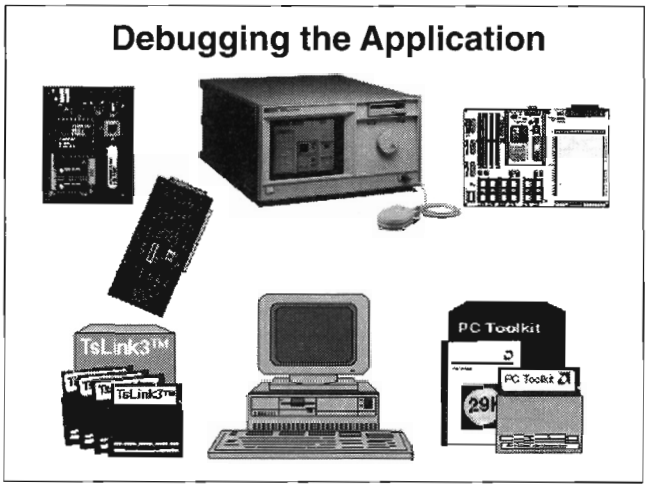
This signals-based interrupt handler for *INT0 sets up the signal frame, saves necessary microcontroller special registers defining the current environment, and then passes control to the C-based signal interrupt handler. The interrupt handler executes with interrupts disabled.

Slide #35



Now in the C language signal handler, the actual interrupt is processed. First, the DSC interrupt register is read to determine what caused the interrupt, and then the condition (transmit FIFO threshold reached, receive FIFO threshold reached, etc.) is handled. Two C macros are used to communicate with the DSC: Rd_dsc() to read an interrupt register (thus clearing the interrupt), and Wr_dsc() to write a register in the DSC. This code is taken directly from the TsLink3 software.

Slide #36

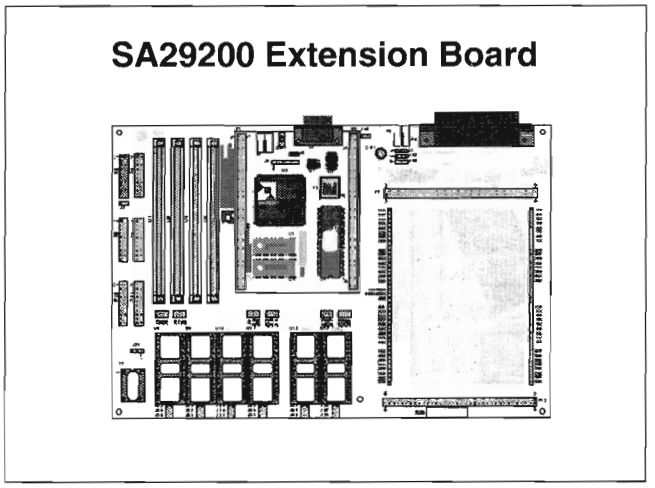


The ISDN terminal adapter can be designed using an IBM-compatible PC (Am386(r) microprocessor-class or better), an SA29200 Evaluation Board and SA29200 Expansion Board, a Corelis Am29200 microcontroller JTAG-based emulator, an HP 16500 logic analyzer, and the TsLink3 software.

Slide #37

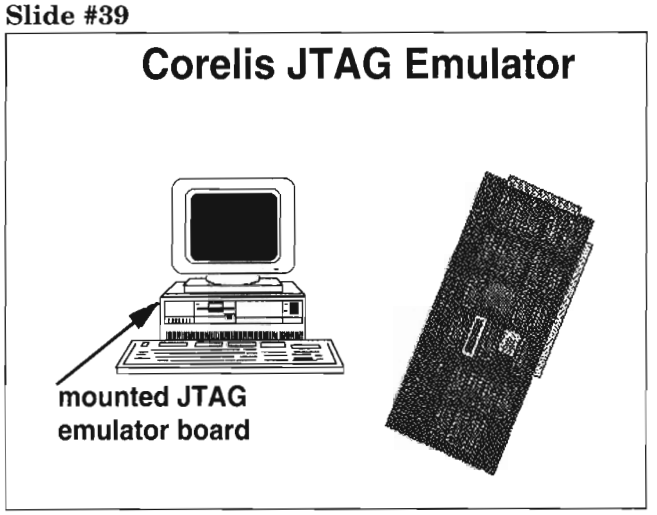
The SA29200 Evaluation Board is a small form-factor board containing an Am29200 microcontroller, DRAM, an EPROM with the MiniMON29K™ debug monitor, and a serial port connector.

Slide #38



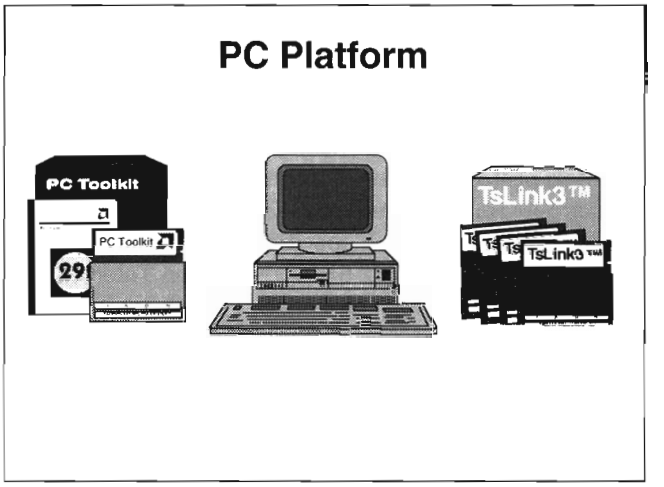
The SA29200 Evaluation Board plugs into an SA29200 Expansion Board which contains header sockets for extra memory, a parallel port connector, and a wire-wrap area in which the DSC and SCC can be wired.

Slide #39



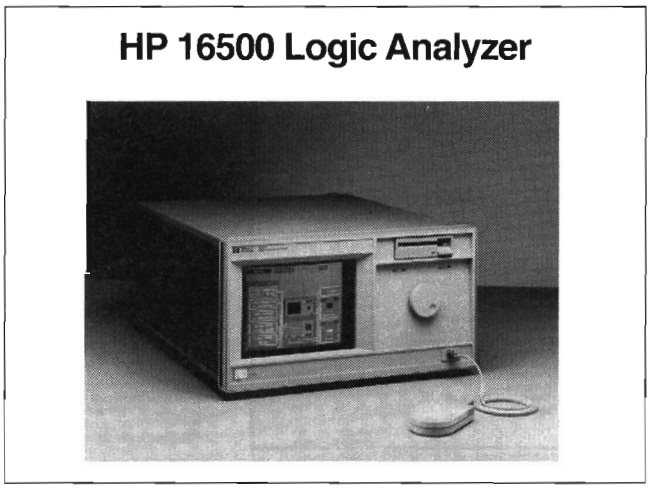
The Corelis JTAG emulator is a low-cost in-circuit emulator that clips onto the Am29200 microcontroller and uses the 5-pin JTAG port to start, stop, and single-step the Am29200 microcontroller, and to request current status information from the microcontroller. It is a board that plugs into the expansion slot of a PC. A special cable connects the emulator board to the system under development.

Slide #40



The PC does triple-duty as a software development platform, the host side of the MiniMON29K debug monitor, and the host platform for the JTAG emulator.

Slide #41



The HP 16500 is used to analyze system activity on a signal/bus level. It works with the JTAG emulator to provide comprehensive information about the system under development.

Slide #42

HP 16500 Analyzer Listing

Label>	ADDR	AM29200 Disassembly	STAT	R/W_	ROMOE/	ROMCS30
Base>	Hex	mnemonics RDP	Hex	Hex	Hex	Hex
5	000AAC	STORE 0,0x00,gr96,gr97	0	7	1	1	E
6	000AAC	ROM Inst. Read: 0x1E	0	5	1	0	E
7	000AAD	Byte #1	0	7	1	1	E
8	000AAD	ROM Inst. Read: 0x00	0	7	1	0	E
9	000AAE	Byte #2	0	7	1	1	E
10	000AAE	ROM Inst. Read: 0x60	0	7	1	0	E
11	000AAF	Byte #3	0	7	1	1	E
12	000AAF	ROM Inst. Read: 0x61	0	7	1	0	E
13	000AB0	CONST gr97,0x008C	0	7	1	1	E
...							
21	000AB4	CONSTH gr97,0x8000	0	7	1	1	E
22	000AB4	ROM Inst. Read: 0x02	0	5	1	0	E
23	000AB5	Byte #1	0	7	1	1	E
24	000AB5	ROM Inst. Read: 0x80	0	7	1	0	E
25	000AB6	Byte #2	0	7	1	1	E
26	000AB6	ROM Inst. Read: 0x61	0	7	1	0	E
27	000AB7	Byte #3	0	7	1	1	E
28	000AB7	ROM Inst. Read: 0x00	0	7	1	0	E
29	000080	Idle Cycle		7	0	1	F
30	000080	Idle Cycle		5	0	1	F
31	000080	Internal data access: Serial Port Control Register		7	0	1	F
32	000AB8	LOAD 0,0x00,gr96,gr97	0	7	1	1	E
33	000AB8	ROM Inst. Read: 0x16	0	6	1	0	E

This HP 16500 display shows a part of the startup code for the Am29200 microcontroller. The startup code is located in a byte-wide EPROM, so the microcontroller does four byte fetches for each 32-bit instruction word. The lines labeled 5 through 12 show bus activity for the fetching of a single instruction (STORE 0,0x00,GR96,GR97) from byte-wide memory; each memory access takes two cycles. The RDP field indicates which DRAM or ROM bank was accessed; in this case, the accesses were all to ROM bank 0.

The microcontroller's three STAT (status) signals show what the microcontroller was doing during the previous cycle. A status of 7 indicates that the microcontroller was idle (data/instruction not valid). A status of 5 indicates that the microcontroller was executing during the previous cycle. A status of 6 indicates that the microcontroller executed an internal data access (to an internal peripheral), in this case to the serial port.

The R/*W signal indicates if a read access is taking place (1) or a write access is taking place (0). The *ROMOE signal is asserted (0) when the output enable to the EPROM is asserted; the analyzer listing shows that the microcontroller's ROM Controller was correctly programmed for two-cycle access memories. The ROMCS30 field shows the ROM chip selects that have been asserted; there is one chip select for each of the four allowed ROM banks. The startup code is all located in ROM bank 0, so this is the only chip select asserted during this listing. Note that during the internal access to the microcontroller's serial port, no ROM banks are enabled.

Slide #43

HP 16500 Analyzer Listing

Label>	ADDR	AM29200 Disassembly	STAT	R/W_	TR/OE	WE/	CAS30/	RAS30/
Base>	Hex	mnemonics RDP	Hex	Hex	Hex	Hex	Hex	Hex
16	000024	CONST lr3,0x40C4	3	4	1	1	1	F	7
17	000024	DRAM Inst. Read from 0x4024	3	7	1	0	1	0	7
18	000024	DRAM Inst. Read from 0x4024	3	7	1	0	1	0	7
19	000028	CONSTH lr3,0x4000	3	7	1	0	1	F	7
20	000028	DRAM Inst. Read from 0x4028	3	5	1	0	1	0	7
21	00004A	Idle cycle		7	0	1	1	F	F
22	00004A	DRAM Write: 0x40000C50 to 0x96E8	3	5	0	1	1	F	7
23	0006E8	DRAM Write: 0x40000C50 to 0x96E8	3	7	0	1	0	0	7
24	0006E8	DRAM Write: 0x40000C50 to 0x96E8	3	7	0	1	0	0	7
25	000020	Idle cycle		7	1	1	1	F	F
26	00002C	CONST lr2,0x0040	3	4	1	1	1	F	7
27	00002C	DRAM Inst. Read from 0x402C	3	7	1	0	1	0	7
28	00002C	DRAM Inst. Read from 0x402C	3	7	1	0	1	0	7
29	000030	HALT	3	7	1	0	1	F	7
30	000030	DRAM Inst. Read from 0x4020	3	5	1	0	1	0	7
31	000034	ASNEQ 69,sp,sp	3	7	1	0	1	F	7
32	000034	DRAM Inst. Read from 0x4024	3	5	1	0	1	0	7
33	000038	CONST lr3,0x40EC	3	7	1	0	1	F	7
34	000038	DRAM Inst. Read from 0x4028	3	5	1	0	1	0	7
35	000038	Idle cycle		7	1	1	1	F	F
36	000038	Halted		5	1	1	1	F	F

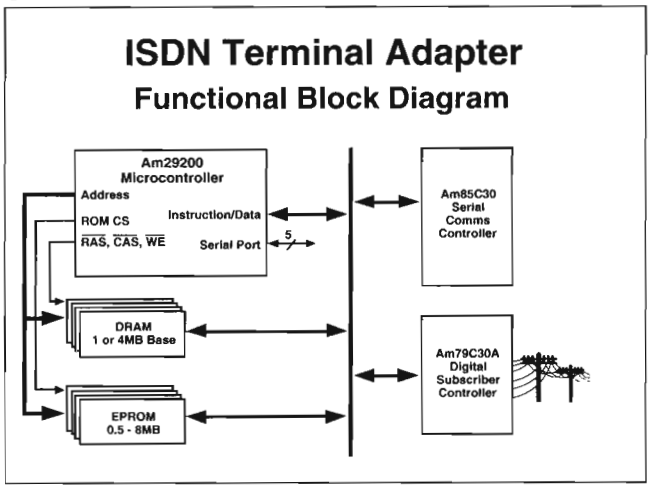
This HP 16500 display shows the execution of a sequence of code that includes a breakpoint set by the JTAG emulator. The code executes out of 3-cycle first access, 2-cycle page mode access DRAM located in DRAM bank 3.

Again the STAT signals show what the microcontroller was doing in the previous cycle; a status of 4 indicates an external data access, in this case to DRAM bank 3. The *TR/*OE signal provides an output enable for read accesses. The *WE signal provides a write enable for write accesses.

The grouped signals *CAS3..0 and *RAS3..0 show the state of RAS and CAS for each of the four DRAM banks.

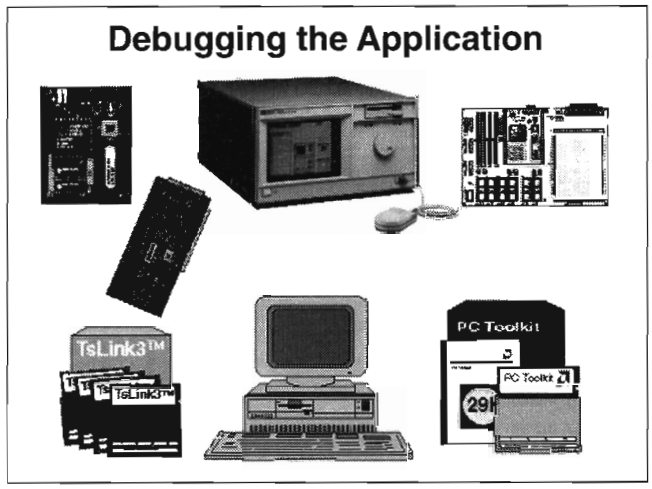
In this listing, a software breakpoint was set at address 0x030 in DRAM bank 3 using the JTAG emulator. The emulator replaced the instruction at address 0x030 with a HALT instruction and asserted the control signals necessary to have the microcontroller report to the emulator when the HALT instruction is encountered. Because the microcontroller is pipelined, the fetch of the HALT instruction occurs at lines 29-30 but isn't executed until line 36.

Slide #44



An ISDN terminal adapter has been developed using an Am29200 microcontroller, an Am85C30 serial communications controller, and an Am79C30A digital subscriber controller. This ISDN terminal adapter interfaces non-ISDN equipment to an ISDN network between the R and S/T ISDN reference points. With an ISDN terminal adapter, the user is able to send both voice and data information digitally, accruing the benefits of increased reliability, new functionality, and lower cost over current analog solutions.

Slide #45



This ISDN terminal adapter can easily be developed in the lab using an IBM-compatible PC (Am386(r) microprocessor-class or better), an SA29200 Evaluation Board and SA29200 Expansion Board, a Corelis Am29200 JTAG-based emulator, an HP 16500 logic analyzer, and the TsLink3 software.

Glossary of Terms and Conditions

Basic rate: The 192-kbps connection between the subscriber and the network. It contains two B-channels and one D-channel.

Bit Stuffing: A type of rate adaption that adds non-data dummy bits to bring the data rate up to 64 kbps. In addition, multiple channels can be multiplexed to bring up the data rate. However, bit stuffing does not support statistical multiplexing or error-checking and retransmission.

BiSYNC: A synchronous character-oriented transmission protocol. BiSYNC is used primarily by IBM. A special "sync" character precedes and ends the data being transmitted. The sync character is chosen such that its bit pattern is significantly different than the other characters being transmitted.

B-channel: A 64-kbps channel that can be used for either data or digitized voice communications.

CCITT: **Consultative Committee on International Telegraphy and Telephony**: The organization responsible for the ISDN standard, among others. The CCITT is part of the ITU (International Telecommunications Union), one of the oldest organizations in the United Nations.

Central office: The lowest level of switching in the public telephone network. A residential telephone or business PABX connects to the public network at a central office.

DLC: **Data Link Controller**: A functional block in the DSC that performs processing of Layer-1 and partial Layer-2 LAPD protocol for the D channel.

DMI: **Digital Multiplexed Interface**: DMI is a freely licensed specification from AT&T that contains four modes, three of which are commonly used in full data rate B channel transmission.

DSC: The Am79C30A Digital Subscriber Controller. An integrated chip that handles basic rate ISDN services for both voice and data.

DSP: **Digital Signal Processing**: A software method of digitally processing analog signals.

D-channel: A 16-kb/s channel provided by the ISDN basic rate interface. The D-channel is primarily used for call-control signaling functions. It can also be used for low priority, low-speed user data at rates up to 9600-bps.

FCS: **Frame Check Sequence**: A method of detecting errors in the LAPD protocol.

HDLC: **High-Level Data Link Control**: A bit-oriented synchronous communications protocol that uses a special bit pattern (the flag) to mark the beginning and end of data transmissions. If the data contains a flag, an extra 0 is inserted into the data stream; this is known as bit stuffing.

IDC: The **Am79C32A ISDN Data Controller**: An integrated chip that handles basic rate ISDN services for data only.

ISDN: **Integrated Digital Services Network**: An international standard for digital voice and data transmission over the switched telephone network.

ISO: **International Standards Organization**: An international organization that sets worldwide standards in telecommunications and other fields.

JTAG: Joint Test Access Group: The name of the group responsible for a 5-pin test and debug interface codified in IEEE 1149.1 specification. JTAG is also used generically as the name for any implementation that meets the IEEE 1149.1 specification.

LAPB: **Link Access Protocol Balanced**: A subset of the HDLC OSI Layer-2 communications protocol. LAPB is the accepted Layer-2 protocol of CCITT's X.25 packet switch specification, and establishes and maintains an error-controlled point-to-point link between a terminal and the packet network.

LAPD: **Link Access Protocol D-channel**: The OSI Layer-2 protocol defined by CCITT for use in ISDN's D-channel. LAPD can be used on the B channels as well (eg, V.120, DMI mode 3) and many times is preferred because only one software package needs to be supported for all channels.

Layer-2 protocol: Refers to Layer 2 (Data Link Layer) of the OSI communications model. Layer-2 converts an unreliable transmission channel into a reliable one; sends frames of data with a checksum; and uses error detection and acknowledgment. Standards that implement Layer-2 protocol include HDLC, SDLC, and BiSYNC.

Layer-3 protocol: Refers to Layer 3 (Network Layer) of the OSI communications model. Layer-3 transmits packets of data through a network. It is responsible for routing and congestion control. Standards that implement Layer-3 protocol include X.25.

LIU: **Line Interface Unit**: The functional block in the DSC that interfaces to an ISDN S or T reference point.

LT: Line Termination: Interfaces to the ISDN U reference point. The LT is located in the telephone company's switch, often at the central office. The LT performs Layer-1 functions for B and D channels, and Layer-2 and Layer-3 functions for D channels.

MAP: Main Audio Processor: The functional block in the DSC that provides a telephone audio interface.

MPI: Microprocessor Interface: The functional block in the DSC that interfaces to an external microprocessor or microcontroller.

NT: **Network Termination**: There are two types of NTs: NT1 and NT2. The NT1 acts as a repeater and performs two- to four-wire conversion (U to S interface). An NT1 deals only with Layer-1 of the OSI model. NT2s are intelligent and actively participate in the call routing/control process. PABXs and line concentrators are examples of NT2 devices. NT2 devices can be connected to multiple types of ISDN lines simultaneously. NT devices often form the boundary between equipment owned by the customer and equipment owned by the telephone company.

OSI: **Open Systems Interconnection**: A seven-layer model of communications services developed by the ISO. This layered model of communications divorces upper layers from changes in technology in lower layers.

PABX: **Private Automatic Branch Exchange**: A telephone exchange on the user's premises. It serves as a private central office and attaches to the public network at a central office on the network.

PIA: **Peripheral Interface Adapter**: The generic peripheral interface defined for the Am29200 microcontroller.

Primary rate: An expensive, high-bandwidth ISDN connection. The primary rate interface is composed of multiple B channels and one 64 kbps D channel.

R Reference point: The R reference point establishes the boundary between non-ISDN equipment and the ISDN network. Terminal adapters are used to convert the protocol used by the non-ISDN terminal to ISDN basic rate or primary rate protocol.

Reference points: CCITT-identified interfaces with established standards for both hardware and software.

S Reference Point: The CCITT designation for the connection between terminal equipment (TE) and the network terminator (NT2), or between TAs and TEs and NT1 if there is no NT2.

SDLC: **Synchronous Data Link Control**: An OSI Layer-2 bit-oriented synchronous communications protocol.

SNA: **Systems Network Architecture**: A structure of data protocols developed by IBM that predates the OSI model.

Subscriber loop: The connection between the central office or PABX and the user's equipment.

T Reference point: The CCITT designation for the connection between NT1 and NT2. If no NT2 is present, there is no T reference point.

TA: Terminal adapter: A device that connects a non-ISDN device between the R and S interface.

TE: Terminal Equipment: An ISDN-compatible device connection to the S/T reference points. A TE can be a computer, telephone, data terminal, etc.

U Reference point: The CCITT designation for the connection between the LT and NT1. Normally, a two-wire basic rate interface or a primary rate line is used, but the four-wire basic rate interface can also be used.

V.110: The CCITT recommendation for interfacing non-ISDN equipment to the ISDN using the bit-stuffing technique. ECMA 102 is the European Computer Manufacturer's Association's version of V.110.

V.120: The CCITT recommendation for interfacing non-ISDN equipment to the ISDN using packetizing techniques. V.120 uses LAPD, provides rate adaption via statistical multiplexing, and supports multiple logic connections.

X.25: The CCITT international packet-oriented protocol used primarily at Layer 3 of the OSI model. channel is primarily used for call-control signaling functions. It can also be used for low-speed user data.