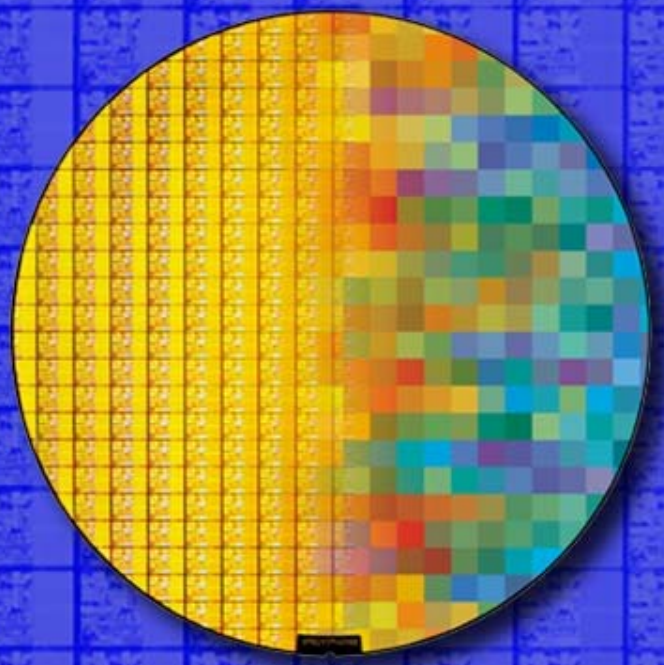


CE-ATA Integration & Design Considerations

Knut Grimsrud
Sr. Principal Engineer

Amber Huffman
Sr. Staff Architect

Legal Disclaimer

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL® PRODUCTS. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER, AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY RELATING TO SALE AND/OR USE OF INTEL PRODUCTS, INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT, OR OTHER INTELLECTUAL PROPERTY RIGHT.

Intel products are not intended for use in medical, life saving, life sustaining, critical control or safety systems, or in nuclear facility applications.

Intel Corporation may have patents or pending patent applications, trademarks, copyrights, or other intellectual property rights that relate to the presented subject matter. The furnishing of documents and other materials and information does not provide any license, express or implied, by estoppel or otherwise, to any such patents, trademarks, copyrights, or other intellectual property rights.

Intel may make changes to specifications, product descriptions, and plans at any time, without notice.

The Pentium 4, Pentium D, Cedar Mill, Presler processors, the 945 Express (Lakeport), the 955X Express (Glenwood) chipsets, or the ICH7 components may contain design defects or errors known as errata, which may cause the product to deviate from published specifications. Current characterized errata are available upon request.

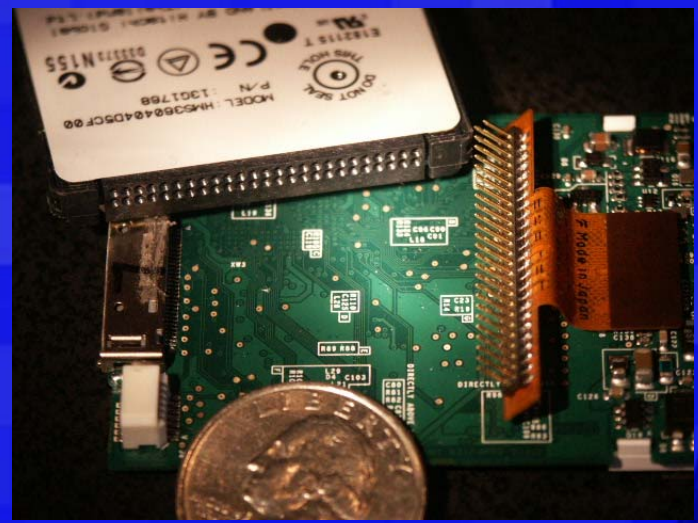
All dates provided are subject to change without notice. All dates specified are target dates, are provided for planning purposes only and are subject to change.

Agenda

- CE-ATA Initiative Update
- Connector & Interconnect Review
- Design Tricks for TTM Implementations

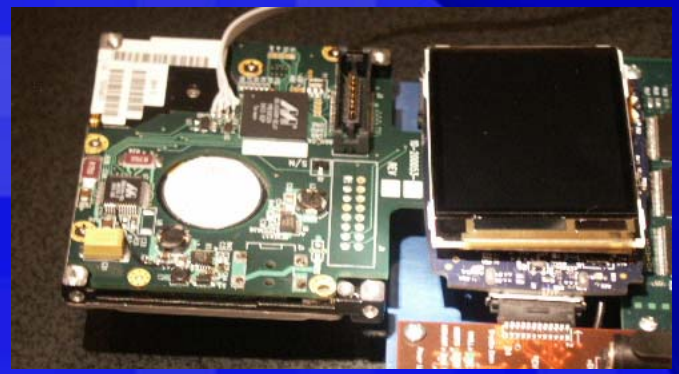
CE-ATA Background/Refresher

- The CE gadget segment is seeing explosive growth
- The current small form factor disk interface is CF+ (Compact Flash + *Parallel ATA*)
  - 50 pin connector with 5V signaling
- CE segment needs highly efficient integration
  - PATA already being displaced in desktop due to integration issues (high pincount, 5V tolerance, …)
  - SATA is not ideal for tiny handheld gadgets with modest requirements



CE-ATA Status

- CE-ATA protocol spec 1.0 ratified and announced at last IDF
- Technology demonstrations have verified technical viability



Intel Press Release

Intel, Other Industry Leaders Announce Final Ce-ATA 1.0 Spec

Debut Nears for Storage Interface for Next-Gen Handhelds and Consumer Electronics

INTEL DEVELOPER FORUM, San Francisco, March 3, 2005 - Just six months after CE-ATA was launched, a final 1.0 specification has been published for the storage interface for next-generation handhelds and portable consumer electronics devices.

"In a word, remarkable. Delivery of the CE-ATA protocol specification is even ahead of the aggressive schedule committed to when we launched the initiative at IDF Fall 2004," said Knut Grimsrud, chairman of the CE-ATA working group and Intel Corporation senior principal engineer. "CE-ATA development is at a feverish pace. The first end products supporting the new technology could be available by the end of this year."

CE-ATA Digital Protocol

Revision 1.0
2-March-2005

Apple Computer, Inc.
Hitachi Global Storage Technologies, Inc.
Intel Corporation
Marvell Semiconductor, Inc.
Nokia Corporation
Seagate Technology LLC
Toshiba America Information Systems, Inc.

Broad Industry Support

Promoters

TOSHIBA

intel

NOKIA

HITACHI
Inspire the Next

MARVELL

Seagate
We turn on ideas

Members

ACARD*
Arcadia Design*
ChipIdea*
Cypress*
Denali Software*
FCI*
Fujitsu*
Hirose*
JAE*
JMicron Tech.*
LSI*
Mentor Graphics*
NucoreTech*
Philips*
QuickLogic*
Silicon Image*
STMicroelectronics*
TransDimension*
ULink*

Agere*
Atmel*
Comax*
Dell*
DSP Group*
Freescale Semiconductor*
Genesys Logic*
Infineon*
Jess-Link Intl*
JST*
M-Systems*
Molex*
NVidia*
Portal Player*
RioSpring*
SMSC*
Synopsis*
Tyco*
Western Digital*

Arasan Chip Systems*
Broadcom*
Cornice*
Delphi Connection Sys.*
EmbWise Tech.*
Fujikura America*
GS Magic*
Intelliprop*
Jinvani Systech*
KnowledgeTek*
Maxtor*
Neomagic*
Oxford Semiconductor*
Quest Electronics*
Samsung*
Square One*
Toorane Tech*
ULI*
Xyratex*

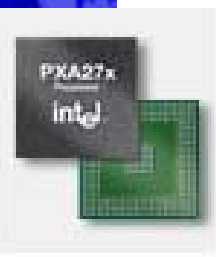
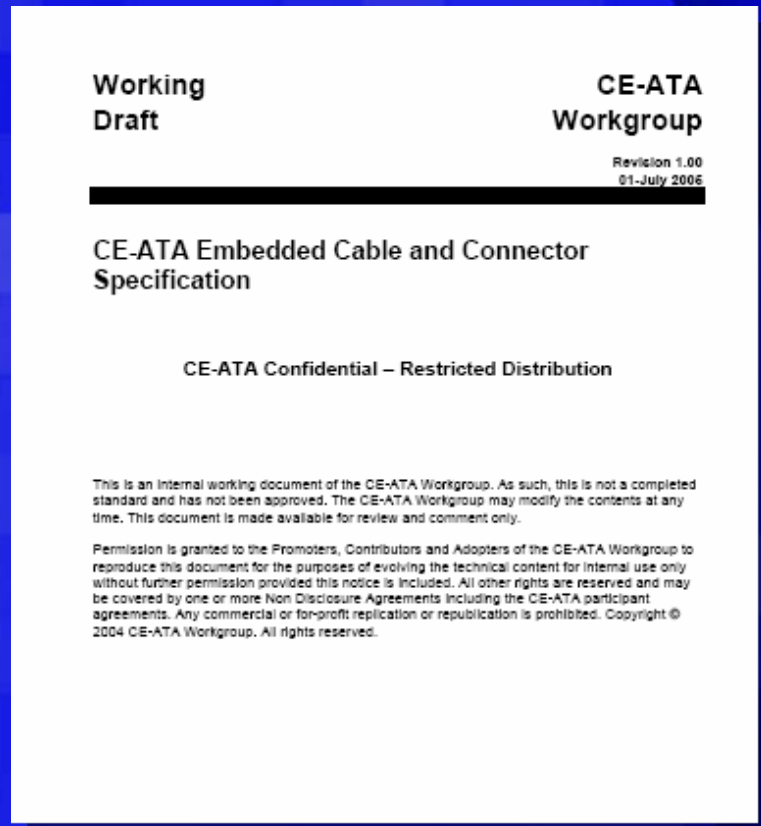


*Image courtesy of Hitachi

*Image courtesy of Seagate

Announcements

- CE-ATA 4-bit connector and interconnect definition is complete
  - Will be available soon for download at www.ce-ata.org
  - Previously ratified 1.0 CE-ATA spec included all protocol definition but not the connector definition



Working Draft

CE-ATA Workgroup

Revision 1.00
01-July 2006

CE-ATA Embedded Cable and Connector Specification

CE-ATA Confidential – Restricted Distribution

This is an internal working document of the CE-ATA Workgroup. As such, this is not a completed standard and has not been approved. The CE-ATA Workgroup may modify the contents at any time. This document is made available for review and comment only.

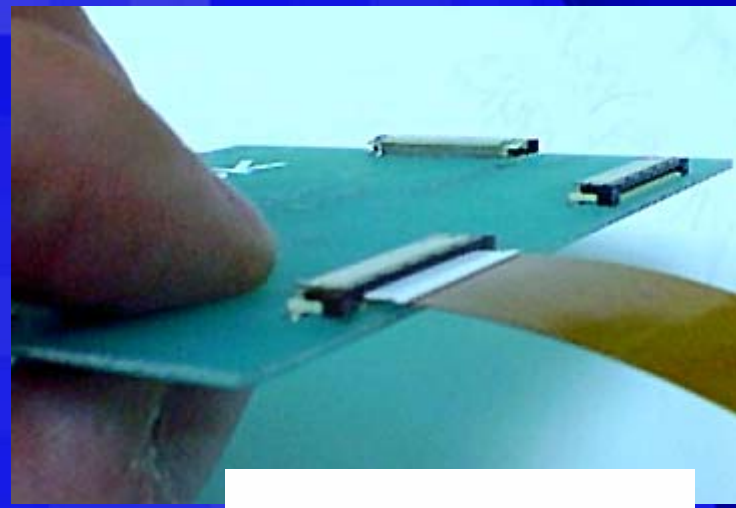
Permission is granted to the Promoters, Contributors and Adopters of the CE-ATA Workgroup to reproduce this document for the purposes of evolving the technical content for internal use only without further permission provided this notice is included. All other rights are reserved and may be covered by one or more Non Disclosure Agreements including the CE-ATA participant agreements. Any commercial or for-profit replication or republication is prohibited. Copyright © 2004 CE-ATA Workgroup. All rights reserved.

CE-ATA technology is stable and complete with strong industry support

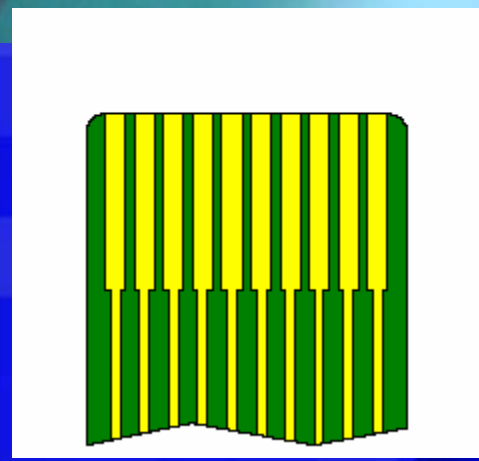
Agenda

- CE-ATA Initiative Update
- Connector & Interconnect Review
- Design Tricks for TTM Implementations

CE-ATA Connector & Interconnect



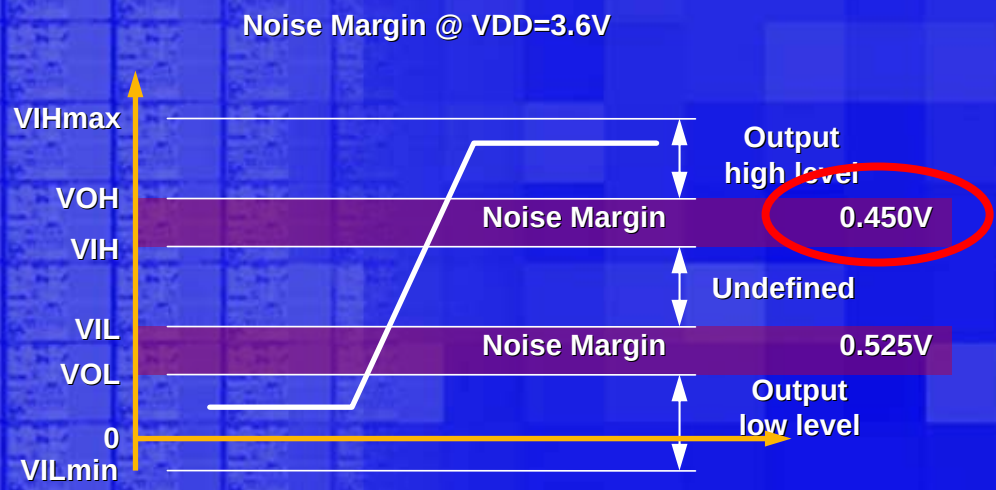
- Connector/interconnect solution must be adequate for highest defined data rates
  - Must provide good longevity
  - Highest MMC frequency is 2x higher than PATA/100
- Connector must also be tiny
  - CF+ (Type 1) is 42.8 x 36.4 x 3.3 mm
  - CE-ATA (x4) is 8.1 x 3.4 x 0.9 mm



CE-ATA connector is 1/200th the volume of CF+

Noise Margin & Integrity Requirements

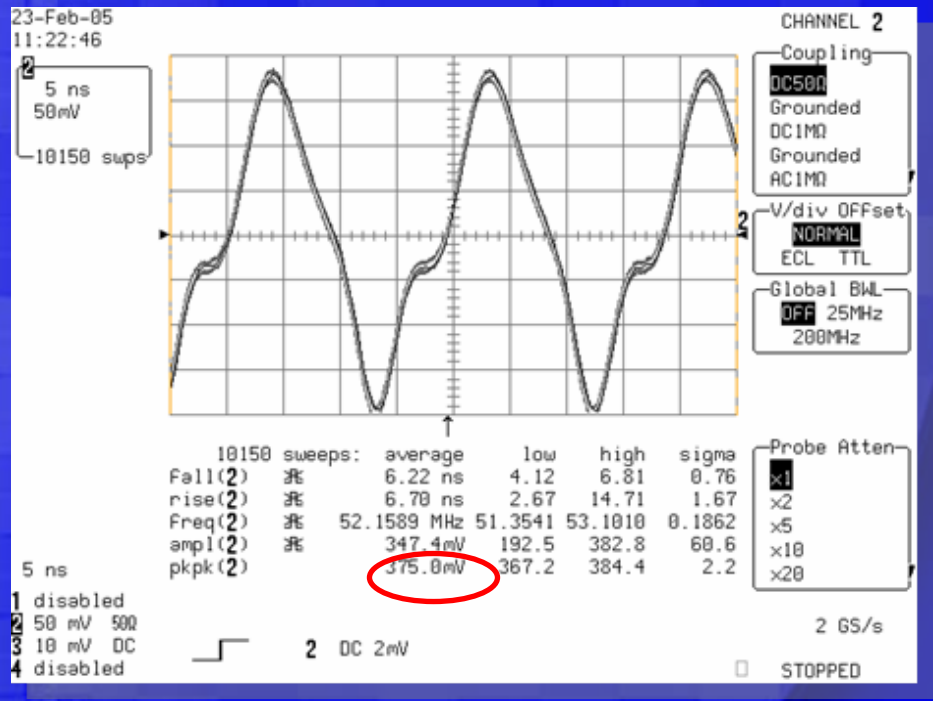
- MMC interface engineered around direct-connect configuration
  - MMC cards have no cable between device and host



- Available interconnect noise margin only 450mV

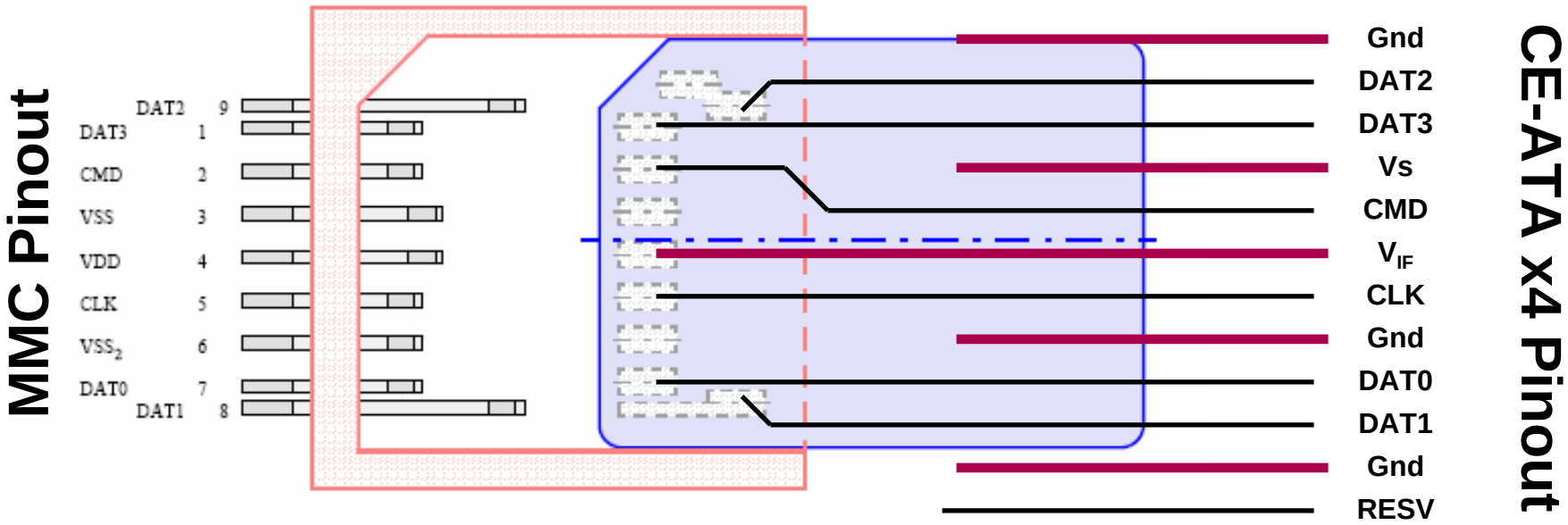
Default Pinout Technical Shortcomings

- For CE-ATA default interface pinout yielded inadequate signal integrity over a flat/flex cable
- Reverse coupling crosstalk between adjacent lines measured 375mV for an applied 3.3V signal at max tolerance (3.9V)
- When superimposed on forward adjacent line crosstalk, 450mV noise margin exceeded



Measurement data and image courtesy of Infineon* ADS S-ATA* and CE-ATA Lab

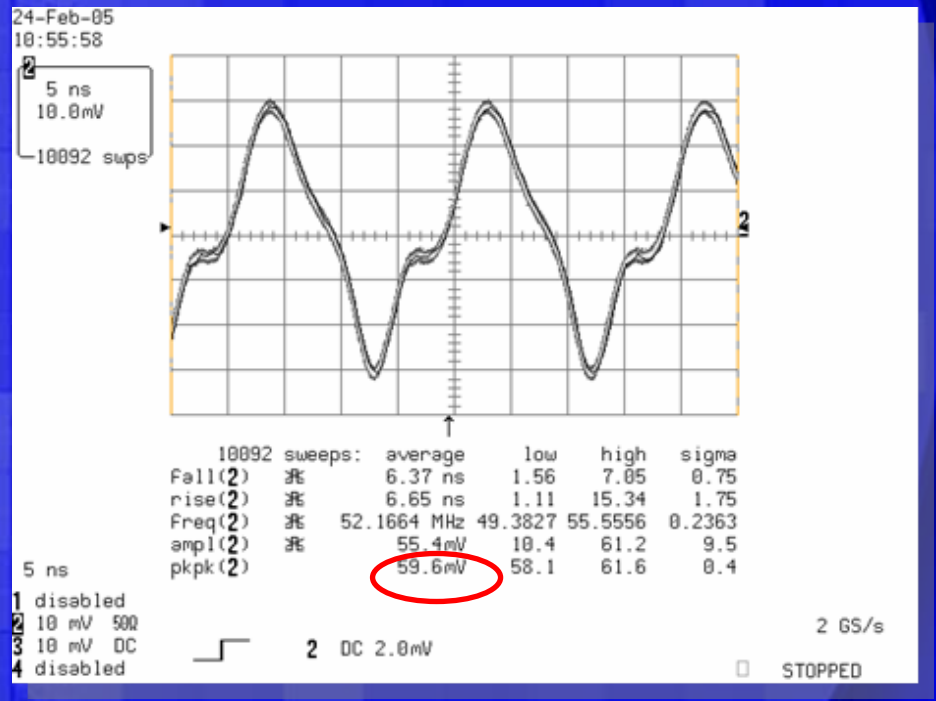
Connector x4 Pinout & MMC Synergy



- Susceptible signals all guarded with DC signals on both sides (either ground or supply)
- All data lines have adjacent DC signal
- Interface signals align with MMC pinout

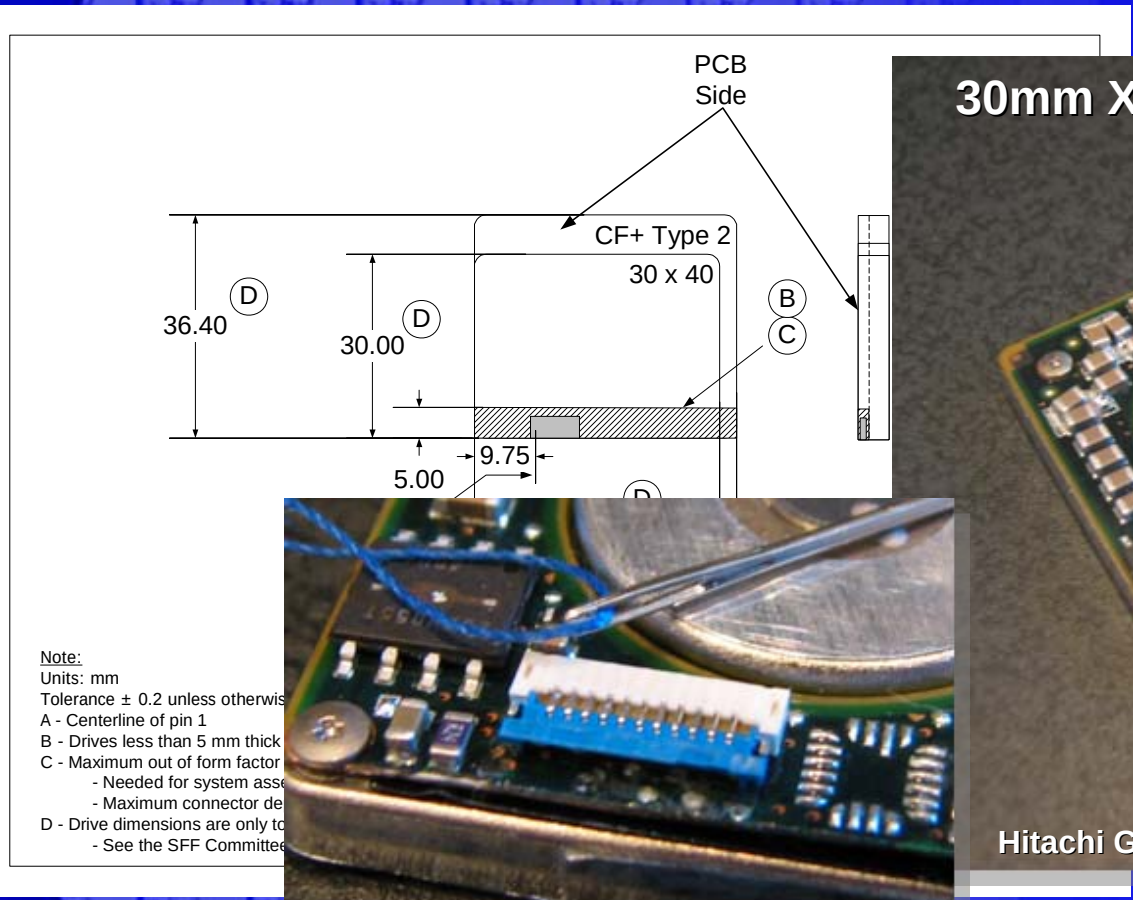
CE-ATA Interconnect Signal Integrity

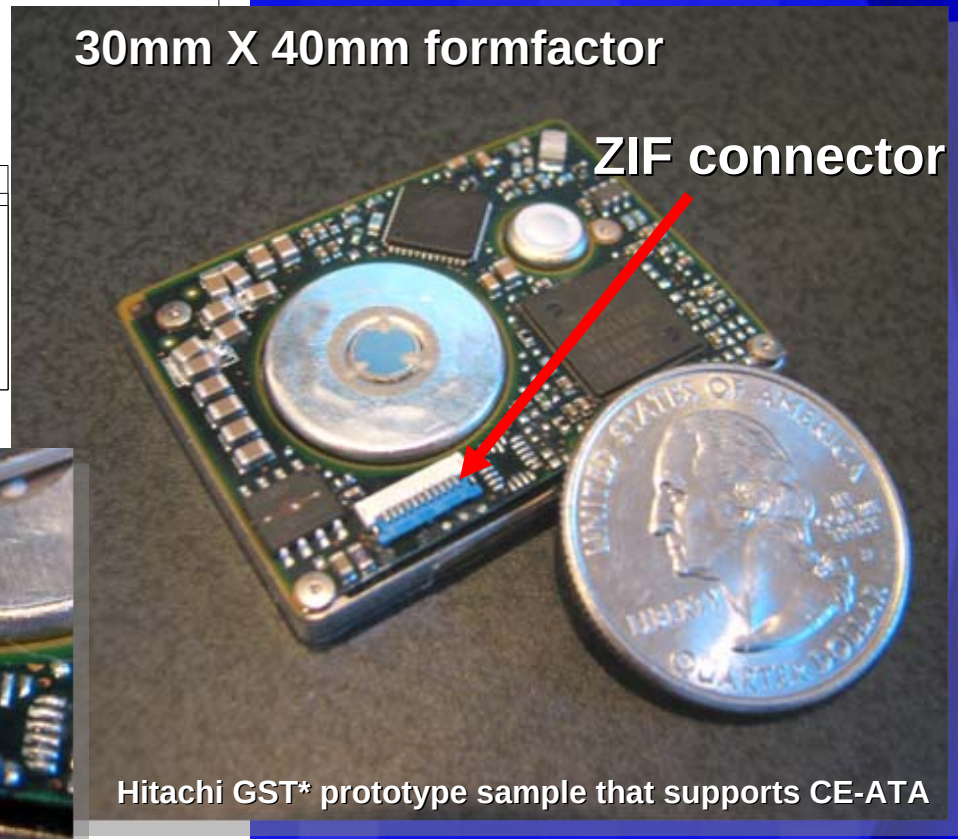
- CE-ATA connector pinout addresses the signal integrity issues
- Worst case reverse coupling crosstalk between adjacent lines measures 60mV for an applied 3.3V signal at max tolerance (3.9V)
- When superimposed on forward crosstalk, still well within noise margin



Measurement data and image courtesy of Infineon* ADS S-ATA* and CE-ATA Lab

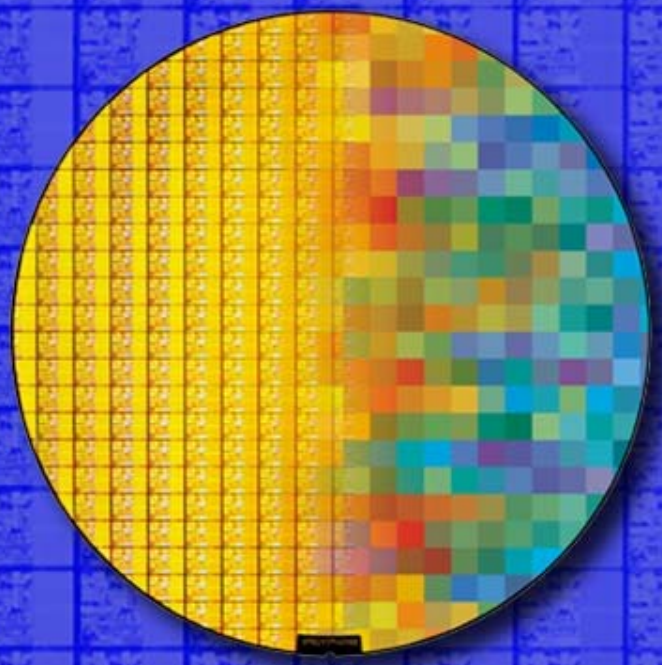
Connector Location Standardization





Hitachi GST* prototype sample that supports CE-ATA

CE-ATA interconnect and connector well suited to handheld applications



Design Tricks for TTM Implementations

Amber Huffman
Sr. Staff Architect

CE-ATA Discovery & Initialization

- There are two levels of device discovery and initialization
- The first level is at the MMC level, including:
  - Hard reset (via CMD0)
  - Setting the block length (via CMD16)
  - Setting the bus width (via CMD6)
  - Selecting ATA mode, details to follow
- The second level of discovery happens at the ATA level by checking for the CE-ATA signature

The ATA Operating Mode in MMC

- MMC specifies card properties in the EXT_CSD register with two segments
  - Properties segment: Information about card capabilities
  - Modes segment: Shows currently selected modes for the card
- There is an ATA mode of operation, and a property bit that specifies whether ATA mode is supported
  - When in ATA mode, using FAST_IO (CMD39) in the MMC TRAN state is allowed. This command is used for ATA Status register reads.
- ATA mode was specified in the MMC specification after the CE-ATA 1.0 protocol specification was released.
  - CE-ATA 1.0 devices may not show support for ATA mode
  - CE-ATA 1.1+ devices should show support for ATA mode

Discovering & Selecting ATA Mode

- To determine whether ATA mode is supported, the EXT_CSD register is read with the SEND_EXT_CSD (CMD8) command
  - Bit 4 of byte 504 (S_CMD_SET) of the EXT_CSD register indicates whether ATA mode is supported
- If ATA mode is supported and MMC initialization is complete, then ATA mode is selected using the SWITCH (CMD6) command
  - Bit 4 of byte 191 (CMD_SET) of the EXT_CSD register when written to ‘1’ selects ATA mode

MMC Command & Response Structure

Background detail

- MMC commands all have the same structure, with a varying command index and argument

	7	6	5	4	3	2	1	0
5	Start 0	Xmit 1	Command Index					
4	Argument[31:24]							
3	Argument[23:16]							
2	Argument[15:8]							
1	Argument[7:0]							
0	CRC7							End 1

- Common responses (R1 and R4) are also 48-bits with a similar structure, except the Xmit bit is ‘0’

Discovering and Selecting ATA Mode

```
mmcCommandResponseRecv(
        SEND_CSD_OPCODE,          // Command index value = 8
        0x0,                      // No argument for SEND_CSD
        1,                        // Single data block
        R1,                       // Response is of R1 type
        pucDataBuffer);           // Buffer to receive data into

// If bit 4 is set, then ATA mode is supported
if (pucDataBuffer[504] & (1 << 4)) {

   // When selecting a command set, the Access argument is 00b,
   // and the Cmd Set argument specifies the bit of the command set
   // to select.  The Index and Value fields are ignored.
   mmcCommandAndResponse(
          SWITCH_OPCODE,          // Command index value = 6
          0x04,                   // Specify ATA mode be selected
          0,                      // No data block
          R1b);                   // Response is of R1b type

   // Check that there was not an error with the SWITCH command.
   mmcCommandAndResponse(
          SEND_STATUS,            // Command index value = 13
          0x0,                    // Set RCA to 0
          0,                      // No data block
          R1);                    // Response is of R1 type
}
```

Checking for the CE-ATA Signature

- The CE-ATA signature indicates if a device complies with the CE-ATA specification
- The CE-ATA signature should be checked for with the FAST_IO (CMD39) command
  - Until a CE-ATA signature is found, it is unknown whether the device supports the more efficient RW_MULTIPLE_REGISTER (CMD60) command

Checking for the CE-ATA signature

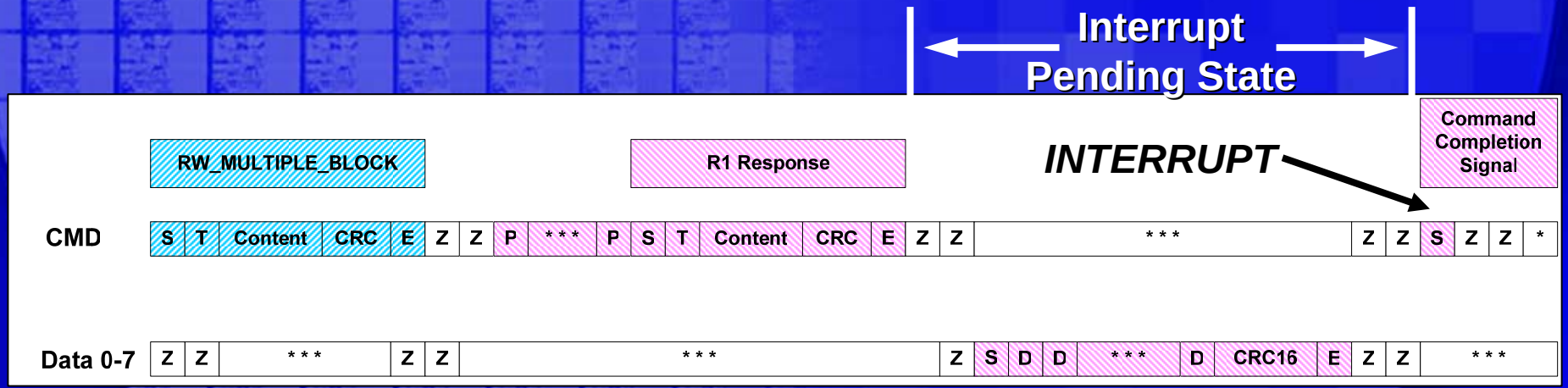
```
mmcCommandResponseArg(
        FAST_IO_OPCODE,          // Command index value = 39
        (0x0C << 8),             // Read LBA Mid register
        0,                       // No data block
        R4,                      // Response is of R4 type
        &respArg);               // Byte to receive register contents

if (respArg == 0xCE) {
   mmcCommandResponseArg(
           FAST_IO_OPCODE,       // Command index value = 39
           (0x0D << 8),          // Read LBA High register
           0,                    // No data block
           R4,                   // Response is of R4 type
           &respArg);            // Byte to receive register contents
   if (respArg == 0xAA) {
        // CE-ATA DEVICE!  YEA!
   }
}
```

	Token	CMD Name	Status	Data	CRC	Time
-84	CMD39	FAST IO(MMC)		6700 000C 0021	10	000,000,671,660 ns
-83	R4			2700 018C CE7B	7D	000,000,180,540 ns
-82	CMD39	FAST IO(MMC)		6700 000D CE81	40	000,047,903,880 ns
-81	R4			2700 018D AA09	04	000,000,180,540 ns
-80	CMD39	FAST IO(MMC)		6700 000F AA49	24	000,001,265,920 ns
-79	R4			2700 018F 403F	5F	000,000,180,540 ns

Trace from Intel PXA 270 platform connected to CE-ATA/CF+ bridge by Intelliprop*

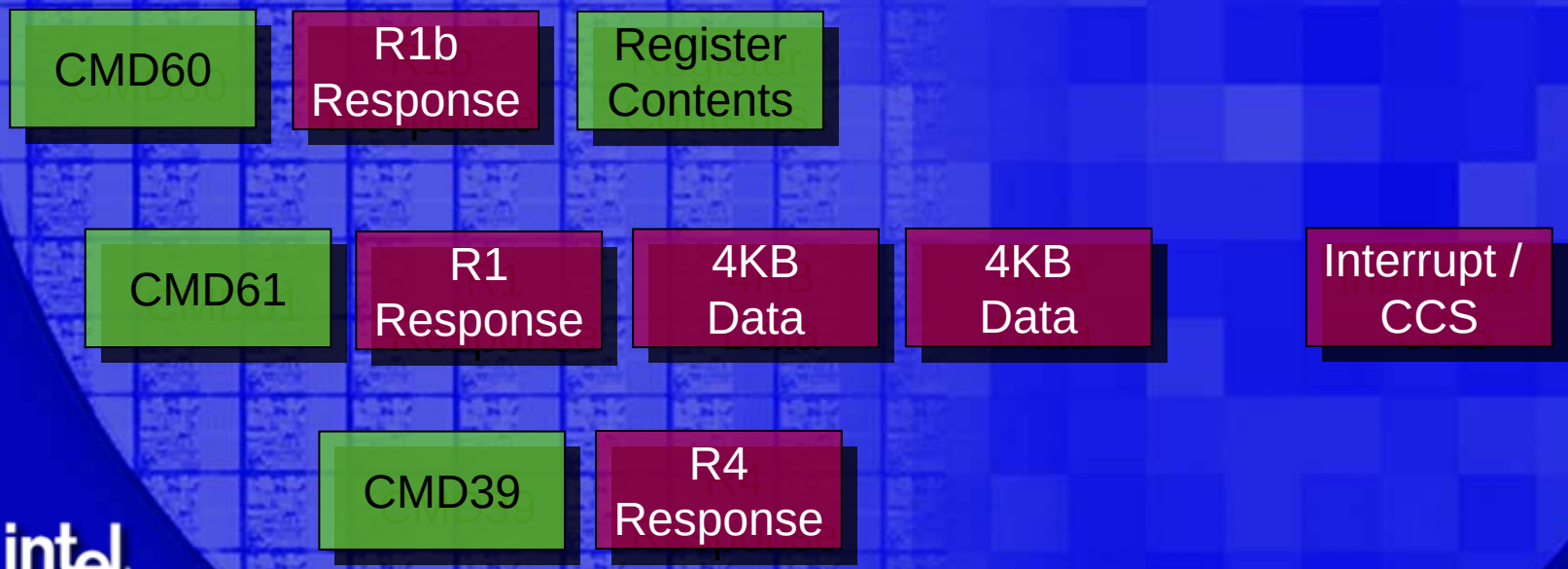
Command Completion Signal

- CE-ATA defines an interrupt mechanism
  - By default it is disabled. Host must explicitly enable by clearing nIEN to zero in ATA Control register in order to activate it.
  - After RW_MULTIPLE_BLOCK is issued and the device response is received, the command line is quiescent and the interrupt pending state is entered
  - In the interrupt pending state, the device can signal the host by clocking a single zero on the CMD line



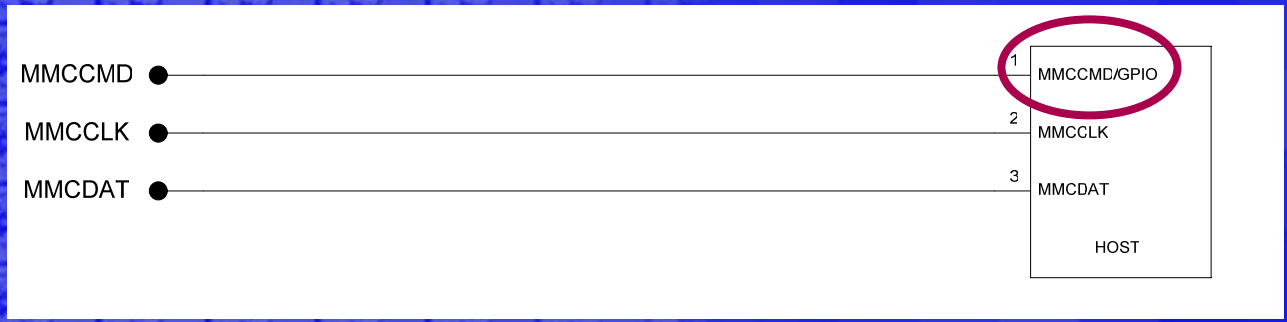
Typical CE-ATA Command Sequence

- Issue READ DMA EXT with RW_MULTIPLE_REGISTER (CMD60) for 8KB of data
- Issue RW_MULTIPLE_BLOCK (CMD61) to transfer data
- Device sends command completion signal (interrupt) when ATA command complete
- Read the ATA Status register to get final status



Interrupts and Current Hosts, Option 1

- Option #1: CMD line reconfigure on the fly
  - When entering interrupt pending state, reconfigure the CMD line to a GPIO with interrupt on change of state
  - Reconfiguration must be completed before interrupt fires to ensure interrupt detected



- Works only with hosts that have MMC CMD line multiplexed with a GPIO signal
- Must watch race condition – reconfiguration must be complete before interrupt arrives
  - May be mitigated for typical case by accounting for latencies and minimum number of clocks for a device to transmit the minimum sized data block

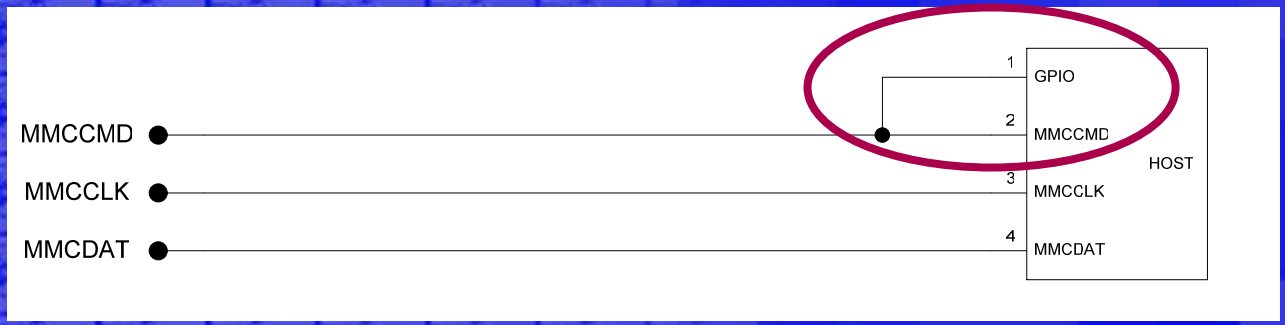
Port Reconfiguration Driver Design

- Enable internal interrupt on R1 response in CMD61 data transfer
- In R1 interrupt routine, reconfigure CMD port to GPIO with interrupt on change of state
- In GPIO interrupt routine, signal command completion and reconfigure port to MMC CMD line

```
Initialize()
{
      ConfigurePort(MMCPORT, MMC_CMD_CONFIG);
}

IssueCMD61(*Args)
{
      *CommandStruct=BuildCommand(CMD61, *Args)
      ControllerInterruptOnR1=TRUE;
      IssueCommand(*CommandStruct);
}

ControllerR1Interrupt()
{
      ConfigurePort(MMCPORT, GPIO_INPUT_CONFIG);
      ControllerInterruptOnChange=TRUE;
}

GPIOInterrupt()
{
      ConfigurePort(MMCPORT,MMC_CMD_CONFIG);
      RetireCompletedCommand();
}
```

Interrupts and Current Hosts, Option 2

- Option #2: Auxiliary GPIO
  - After R1 response to CMD61, enable interrupt on change of state for auxiliary GPIO
  - Enable must be completed before interrupt fires to ensure interrupt detected



- Works with hosts that have an available GPIO with interrupt on change of state, provided pin loading OK
- Race condition from #1 still exists and corresponding precautions must be taken

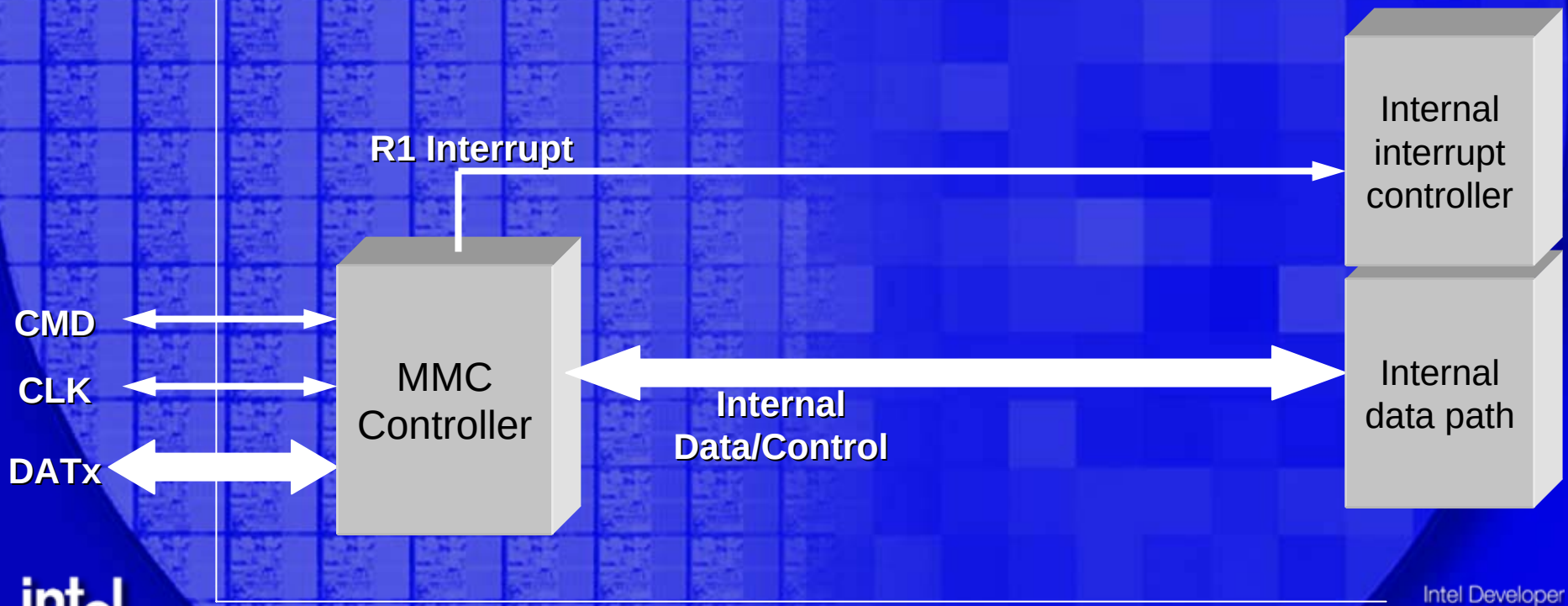
Aux GPIO Driver Design

- Same as port reconfiguration from previous example
- Enable GPIO interrupt in interrupt pending state

```
Initialize()
{
      ConfigurePort(MMCPORT, MMC_CMD_CONFIG);
      ConfigurePort(GPIOPORT,INPUT);
      ControllerInterruptOnGPIO=FALSE;
}

IssueCMD61(*Args)
{
      *CommandStruct=BuildCommand(CMD61, *Args)
      ControllerInterruptOnR1=TRUE;
      IssueCommand(*CommandStruct);
}

ControllerR1Interrupt()
{
      ControllerInterruptOnGPIO=TRUE;
}

GPIOInterrupt()
{
      ControllerInterruptOnGPIO=FALSE;
      RetireCompletedCommand();
}
```

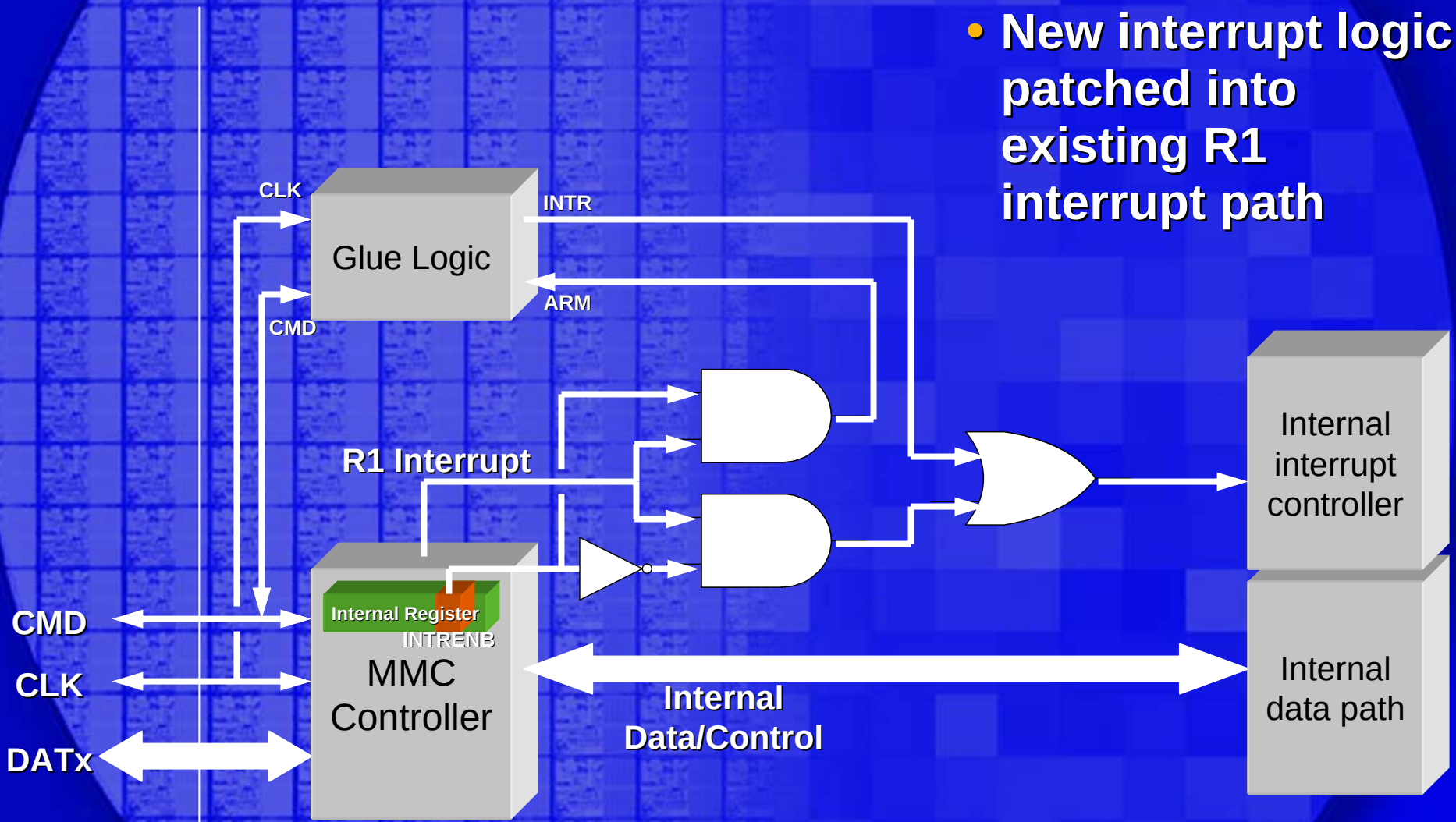
Integrated Hardware Handler

- Generic internals arrangement



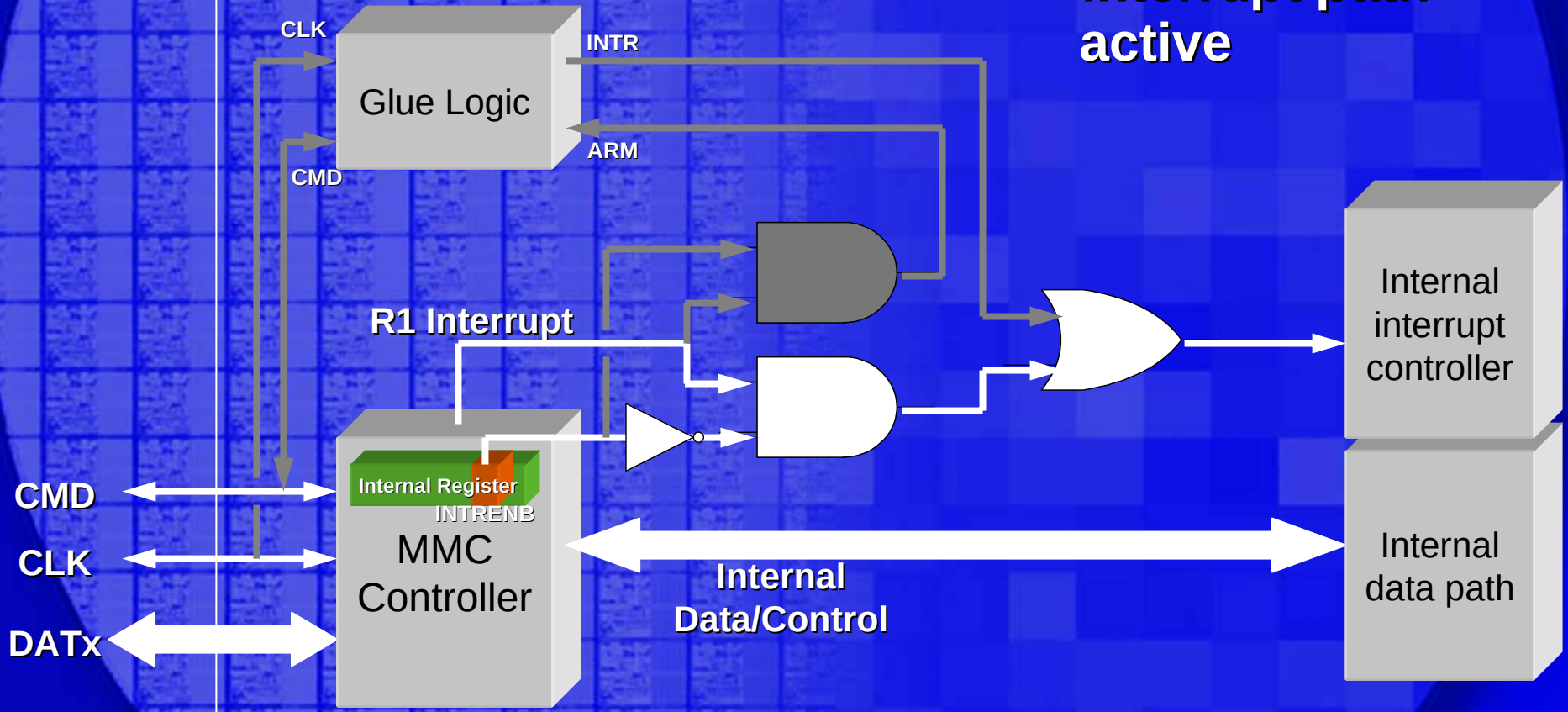
Integrated Hardware Handler

- New interrupt logic patched into existing R1 interrupt path



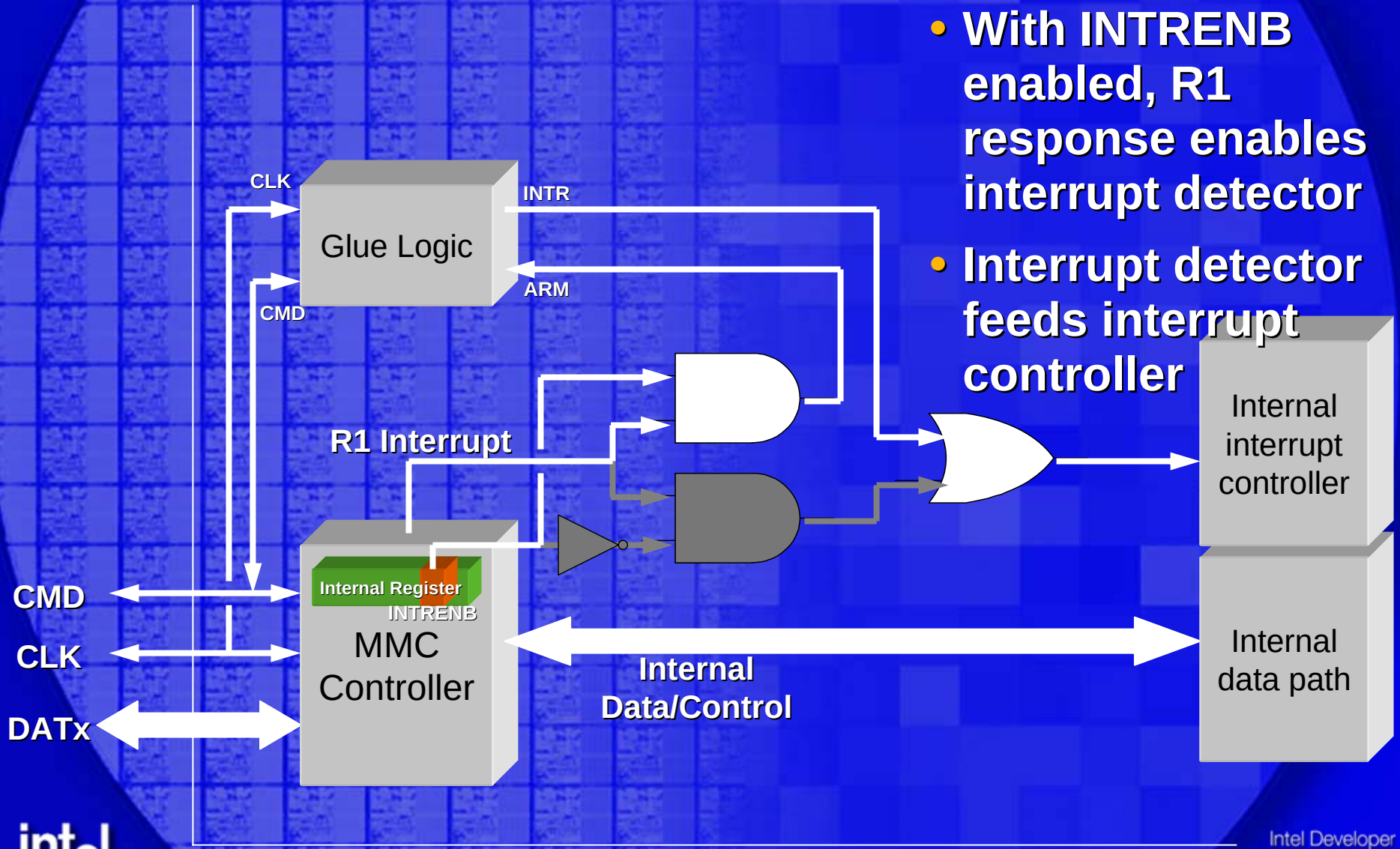
Integrated Hardware Handler

- With INTRENB disabled, normal interrupt path active

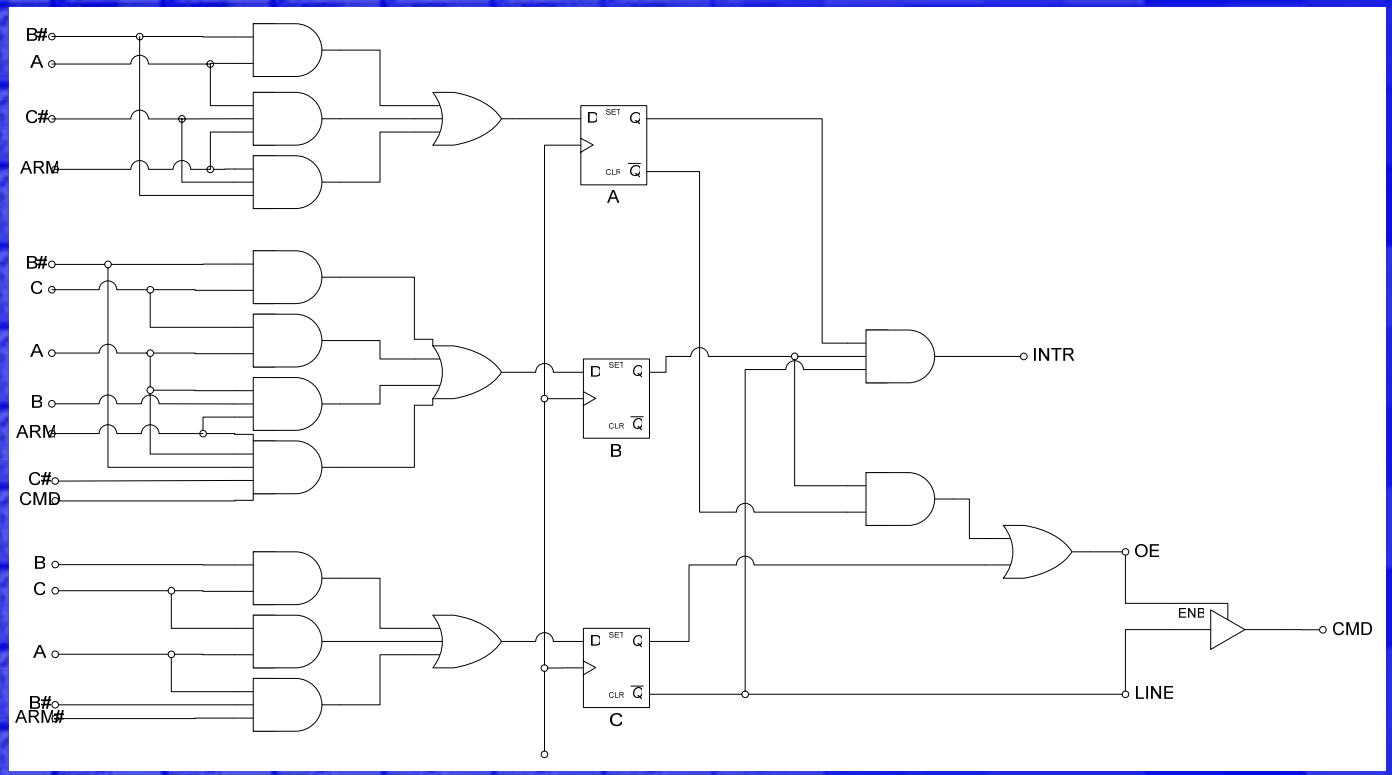


Integrated Hardware Handler

- With INTRENB enabled, R1 response enables interrupt detector
- Interrupt detector feeds interrupt controller



Hardware Interrupt Handler Glue Logic



- Modest gate count (3 flip-flops and combinatorial logic)
- Full-function interrupt handler including transmission of interrupt cancellation signal if disabled
- More details on logic and implementations options available in CE-ATA supplemental design document

Integrated Hardware Driver Design

- The interrupt logic is enabled when CMD61 data transfer issued
- R1 interrupt fires internally only after both the R1 response occurred and the following command completion signal is received

```
Initialize()
{
      ConfigurePort(MMCPORT, MMC_CMD_CONFIG);
      INTR_ENB=0;
}

IssueCMD61(*Args)
{
      *CommandStruct=BuildCommand(CMD61, *Args)
      ControllerInterruptOnR1=TRUE;
      INTR_ENB=1;
      IssueCommand(*CommandStruct);
}

ControllerR1Interrupt()
{
      INTR_ENB=0;
      RetireCompletedCommand();
}
```

Summary

- CE-ATA technology is stable and complete with strong industry support
- CE-ATA interconnect and connector is well suited to handset applications
- A number of TTM host implementation options are available depending on existing platform capabilities

Target of Opportunity

- Embedded interconnect solution is complete and designs can confidently move forward
  - Design your interconnect solution today
  - Build platforms that incorporate the interconnect solution
- Existing host platforms can take advantage of CE-ATA, including the advanced interrupt capability
  - Design CE-ATA into existing host platforms today and the future host platforms of tomorrow

Acknowledgements

- Thanks to Harvey Newman at Infineon* ADS S-ATA* and CE-ATA Lab for providing signal integrity analysis and scope images
- Thanks to Tracy Spitler at Intelliprop* for trace from CE-ATA/CF+ bridge solution

Additional Resources for this Session

- Session presentation can be downloaded from the IDF web site – when prompted enter:
  - Username: idf
  - Password: fall2005
- More web based info including spec & membership info at: www.CE-ATA.org

Please fill out the Session Evaluation Form.

Thank You!

Backup Slides