Core Peripherals

Speaker: Tzu-Wei Tseng

Adopted from National Chiao-Tung University
IP Core Design

SOC Consortium Course Material

Goal of This Lab

- Familiarize with ARM Hardware Development Environment
  - ARM Integrator/AP
  - Core Module
  - Logic Module
- How to use Timer/Interrupt

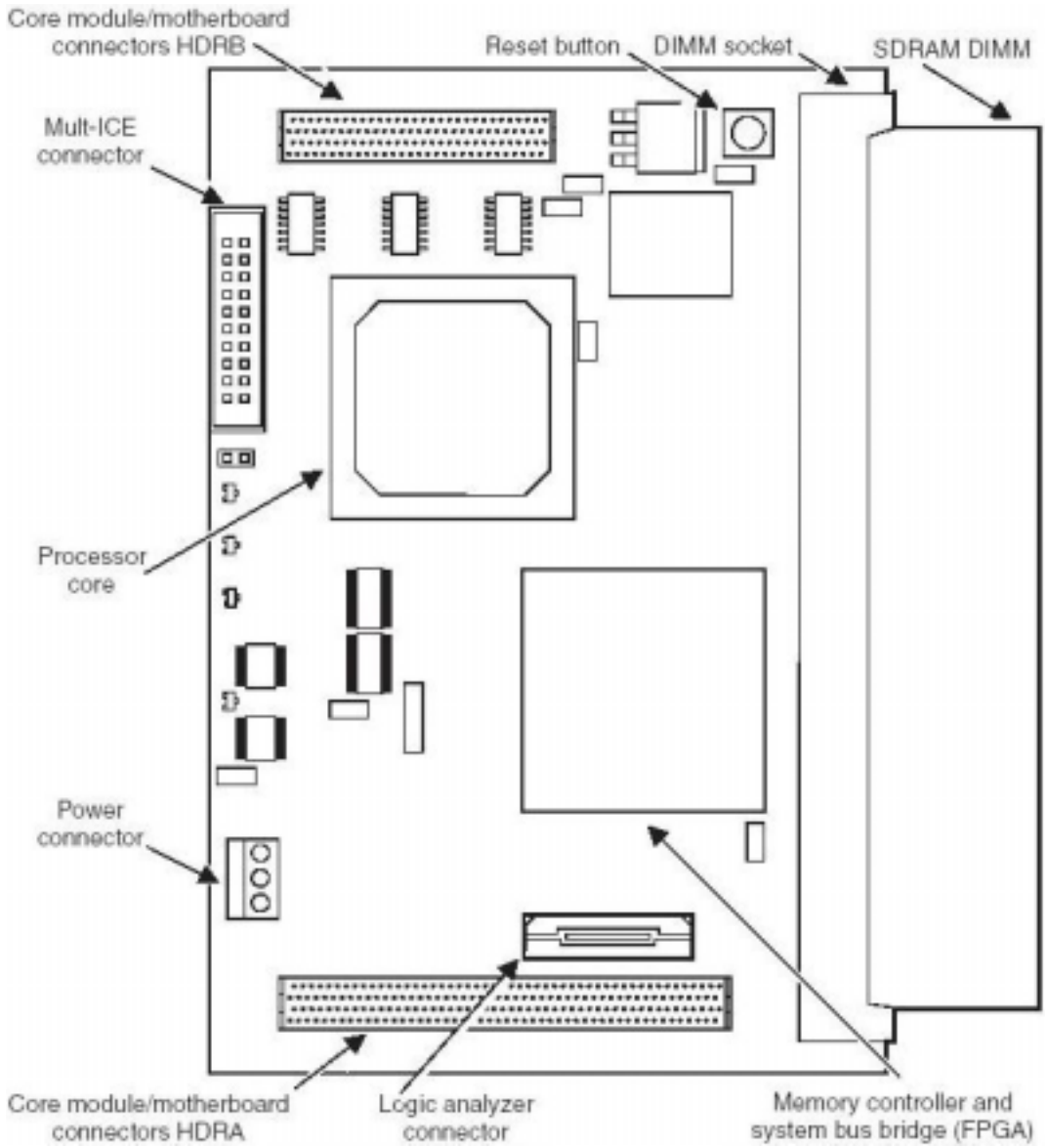
Outline

- ***ARM Integrator Core Module (CM) [1]***
- ARM Integrator Logic Module (LM) [2]
- ARM Integrator ASIC Application Platform (AP) [3]
- System Memory Map [1]
- Lab3 – Core Peripheral

ARM Integrator Core Module/CM

- CM provides **ARM core** personality.
- CM could be used as a standalone development system without AP.
- CM could be mounted onto AP as a system core.
- CM could be integrated into a 3rd-party development or ASIC prototyping system.

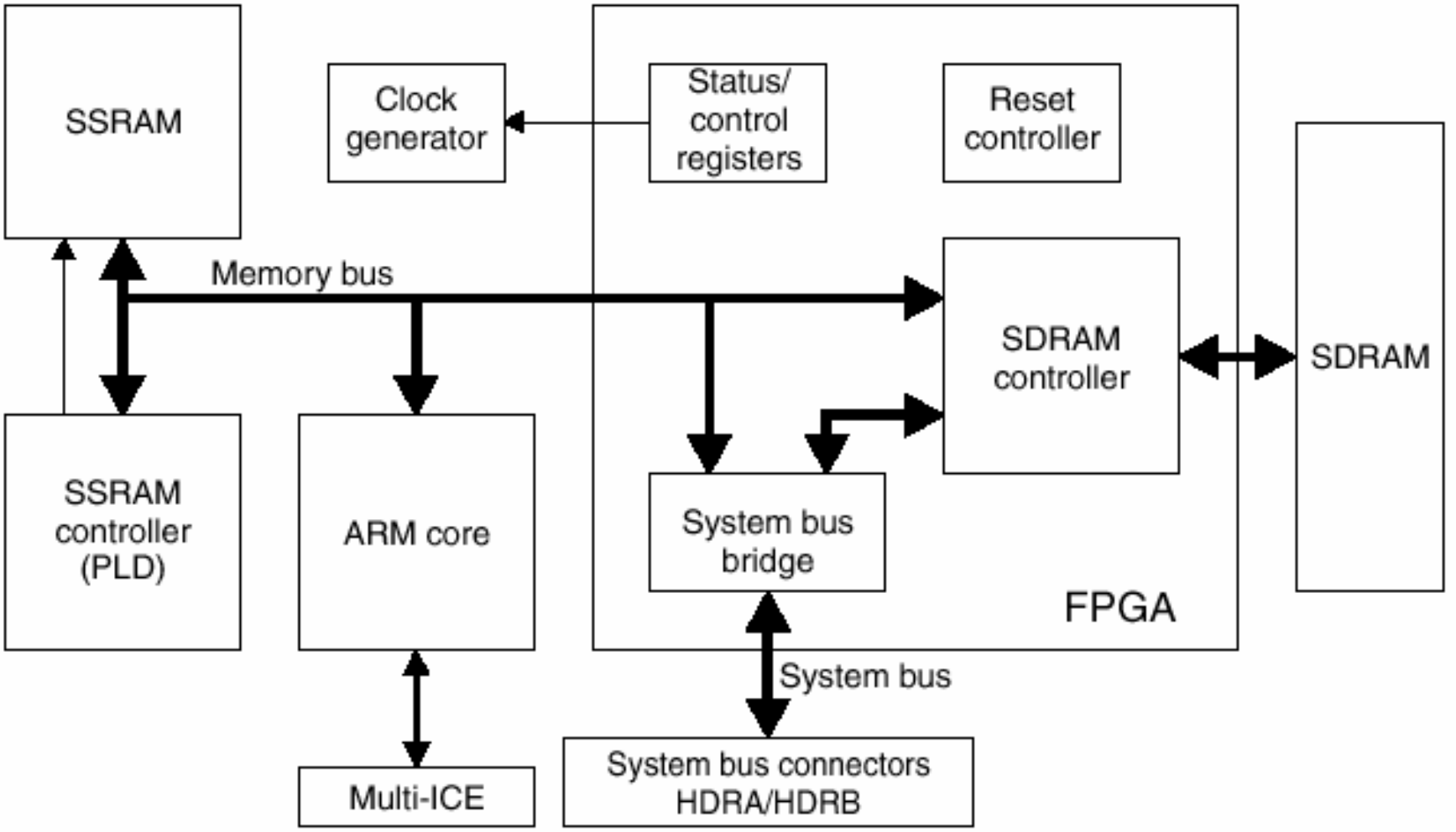
Core module


Core module/motherboard connectors HDRB
Reset button
DIMM socket
SDRAM DIMM
Mult-ICE connector
Processor core
Power connector
Core module/motherboard connectors HDRA
Logic analyzer connector
Memory controller and system bus bridge (FPGA)

ARM Integrator/CM Feature (CM9TDMI)

- ARM9TDMI microprocessor core
  - ARM940T/ARM920T
- Core module controller FPGA :
  - SDRAM controller
  - System bus bridge
  - Reset controller
  - Interrupt controller
- Supports 16MB~256MB PC66/PC100 168pin SDRAM
- Supports 256/512 KB SSRAM
- Multi-ICE, logic analyzer, and optional trace connectors.

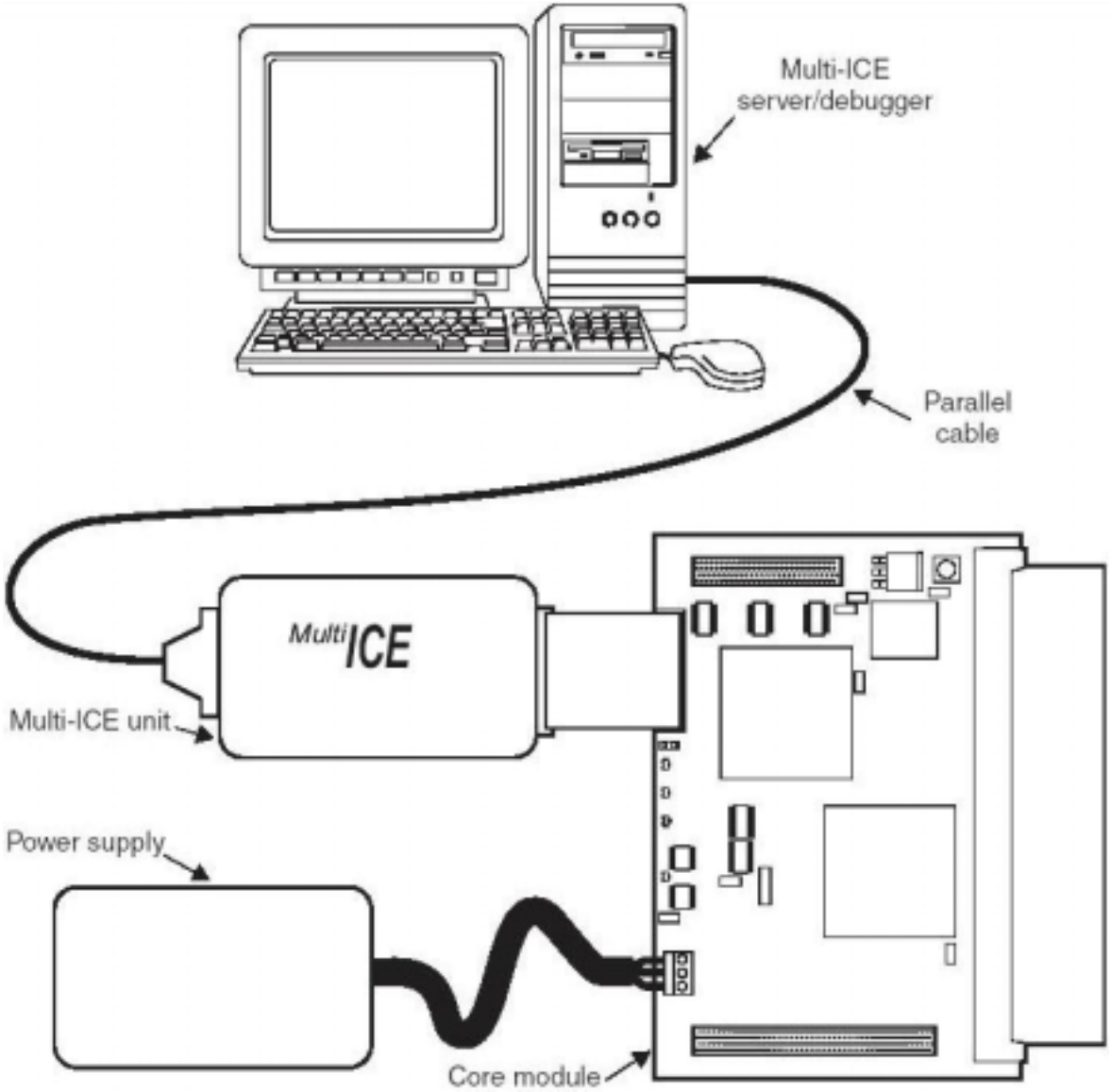
FPGA functional diagram



ARM Integrator Core Module FPGA

- SDRAM controller
  - Supports for DIMMs from 16MB to 256MB.
- Reset controller
  - Initializes the core.
  - Process resets from different sources.
- Status and configuration space
  - Provides processor information.
  - CM oscillator setup.
  - Interrupt control for the processor debug communications channel.
- System bus bridge
  - Provides Interface between the memory bus on the CM and the system bus on the AP.

Connecting Multi-ICE with CM



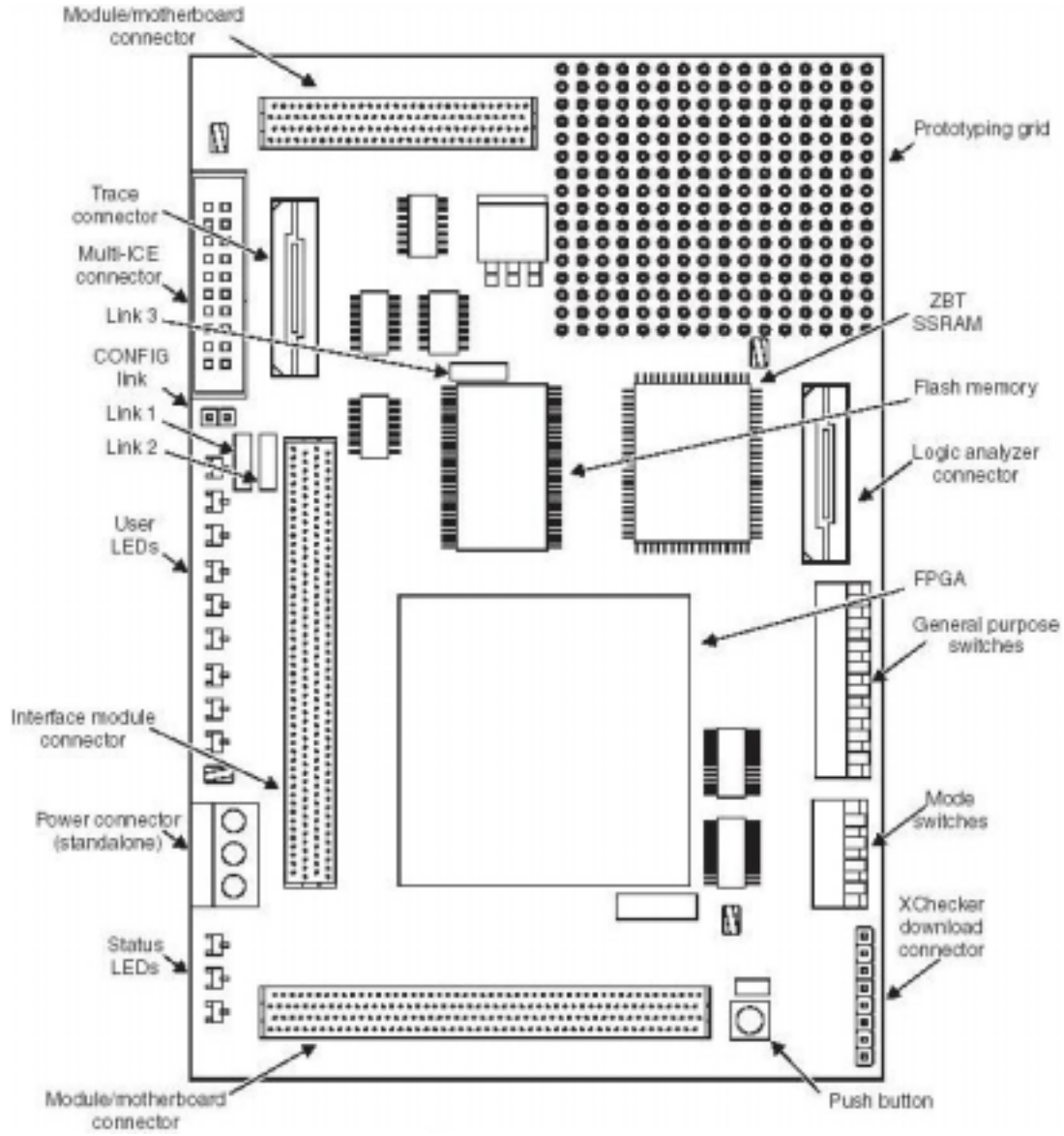
Outline

- ARM Integrator Core Module (CM) [1]
- ***ARM Integrator Logic Module (LM) [2]***
- ARM Integrator ASIC Application Platform (AP) [3]
- System Memory Map [1]
- Lab3 – Core Peripheral

ARM Integrator/LM Logic Module

- LM is designed as a platform for development **AHB/ASB/APB** peripherals for use with ARM cores.
- LM could be used as a standalone system.
- LM could be mounted with an Integrator/CM, and an Integrator/AP motherboard.
- LM could be used as a CM with Integrator/AP if a synthesized ARM core, such as ARM9TDMI-S, is programmed into the FPGA.

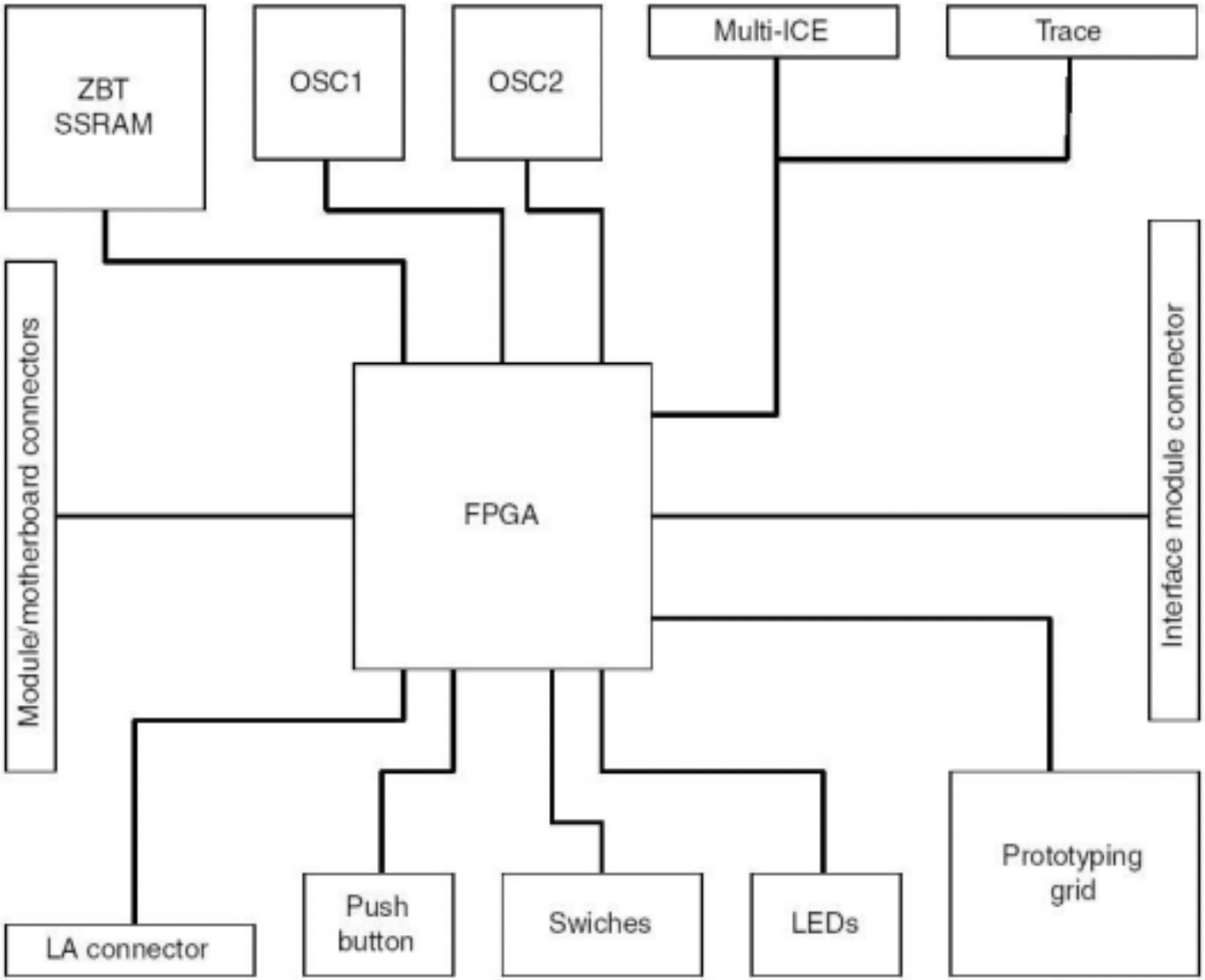
Integrator/LM



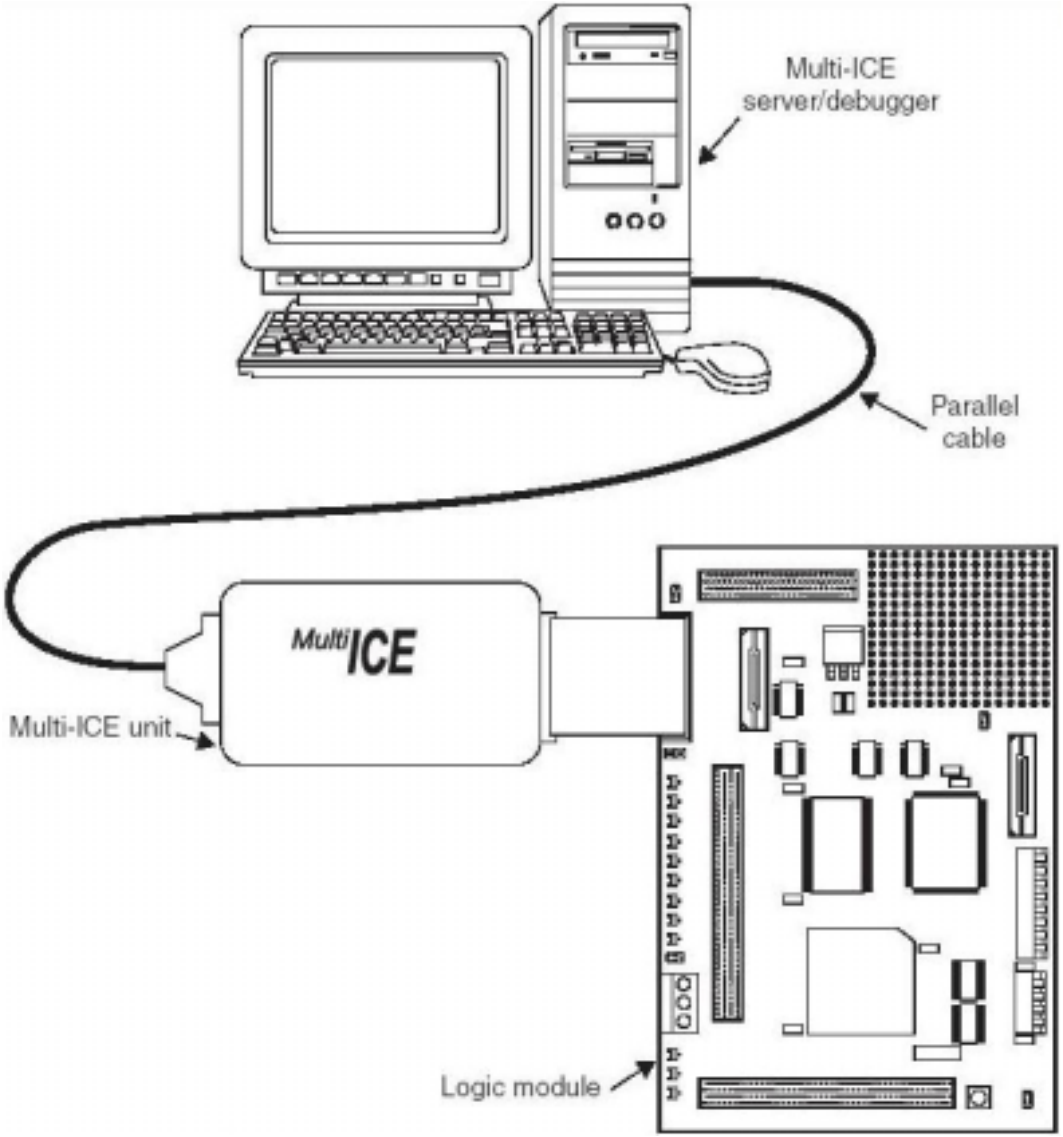
ARM Integrator/LM Feature (XCV-2000E)

- Altera or Xilinx FPGA
- Configuration PLD and flash memory for storing FPGA configurations
- 1MB SSRAM
- Clock generators and reset resources
- Switches
- LEDs
- Prototyping grid
- JTAG, Trace, and logic analyzer connectors
- System bus connectors to a motherboard or other modules

LM Architecture



Using Multi-ICE with LM



Outline

- ARM Integrator Core Module (CM) [1]
- ARM Integrator Logic Module (LM) [2]
- ***ARM Integrator ASIC Application Platform (AP) [3]***
- System Memory Map [1]
- Lab3 – Core Peripheral

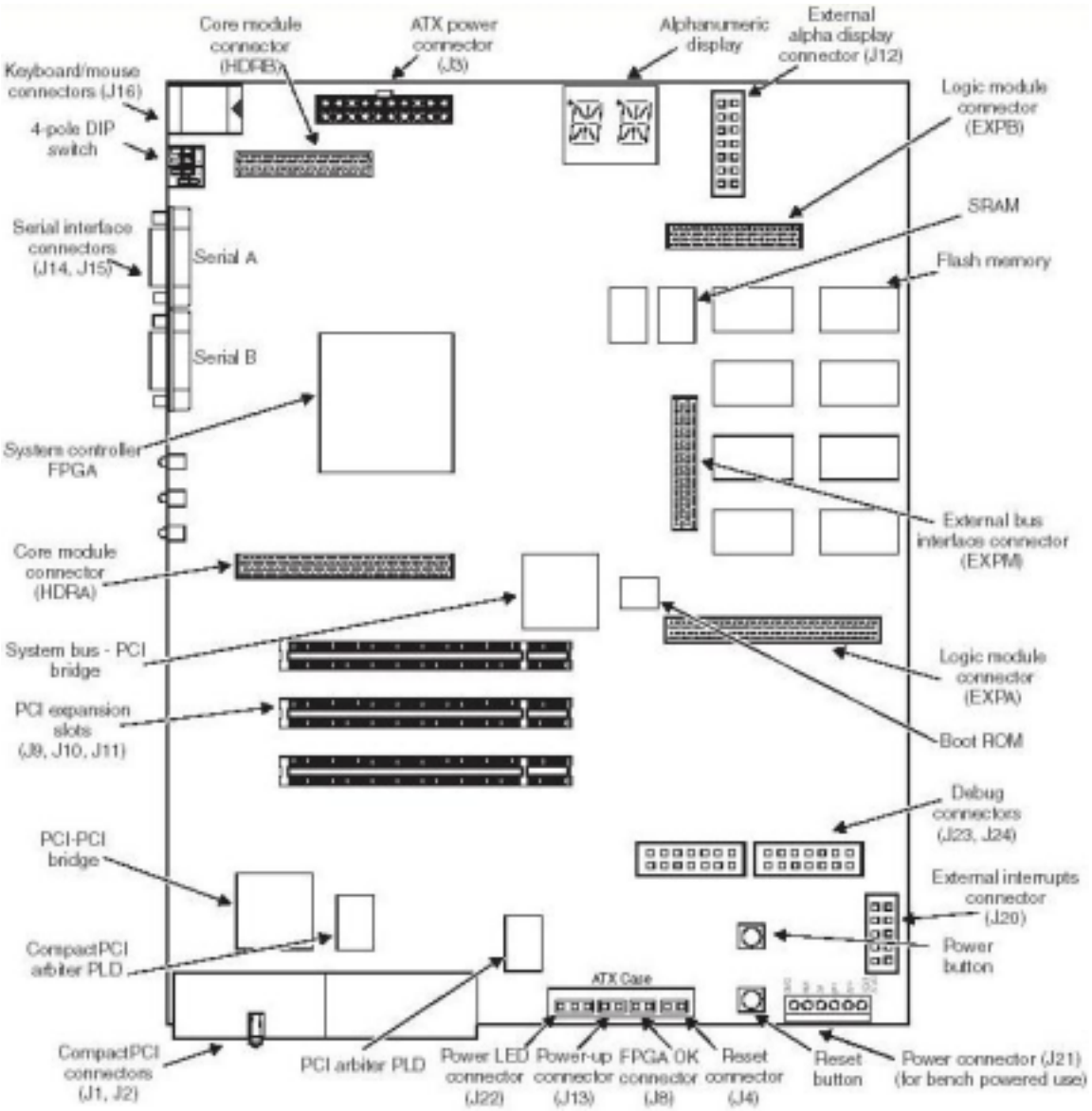
About ARM Integrator/AP

- An ATX motherboard which can be used to support the development of applications and hardware with ARM processor.
- Platform board provides the ***AMBA*** backbone and system infrastructure required.
- *Core Modules* (CM) *& Logic Modules* (LM) could be attached to ASIC Platform.

ARM Integrator/AP Features

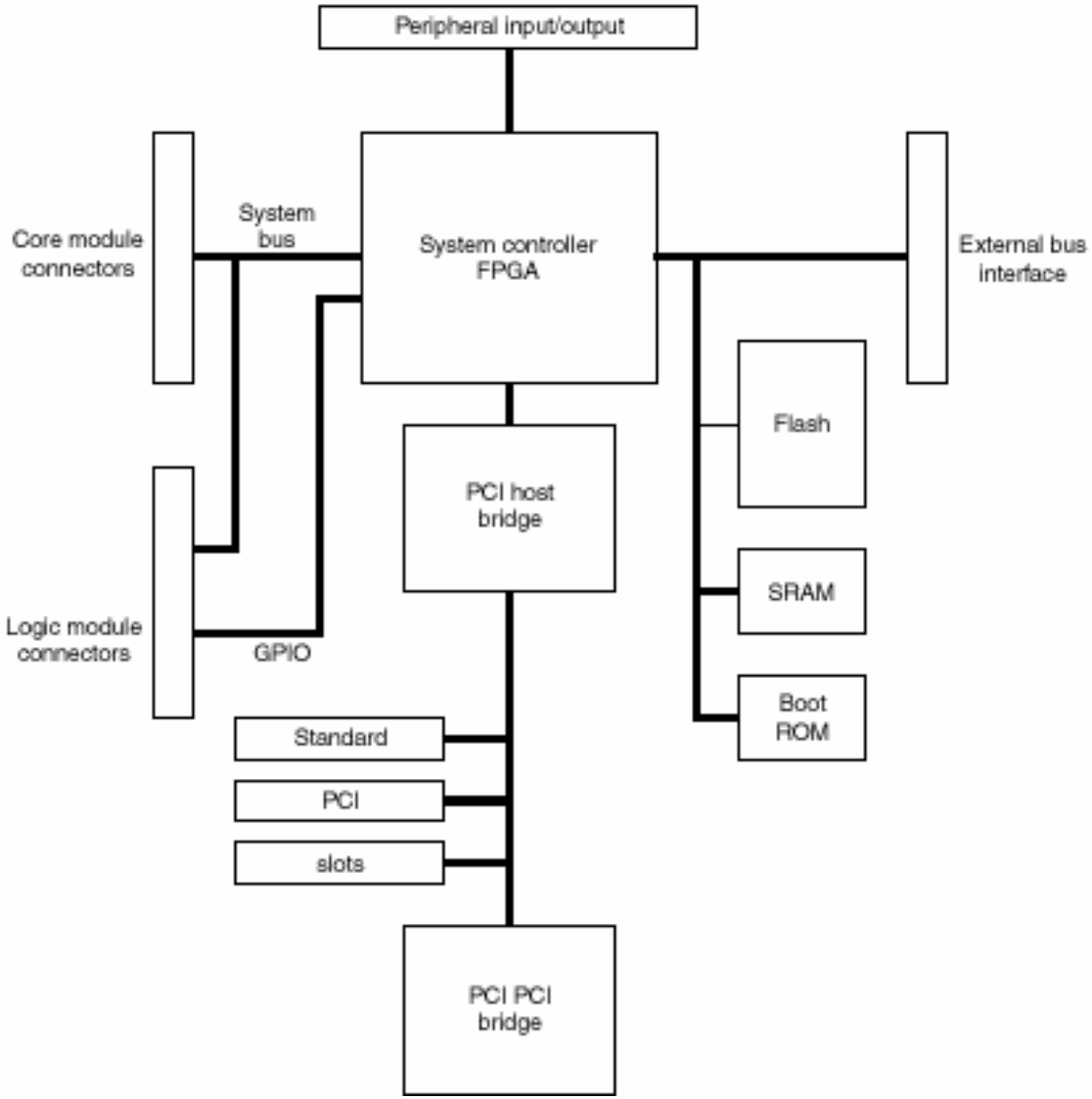
- System controller FPGA.
  - System bus to CMs and LMs
  - System bus arbiter
  - Interrupt controller
  - Peripheral I/O controller
  - 3 counter/timers
  - Reset controller
  - System status and control registers
- Clock Generator
- Two serial ports (RS232 DTE)
- PCI bus interface supporting onboard expansion.
- *External Bus Interface* (EBI) supporting external memory expansion.
- 256KB boot ROM
- 32MB flash memory.
- 512K SSRAM.

Integrator/AP

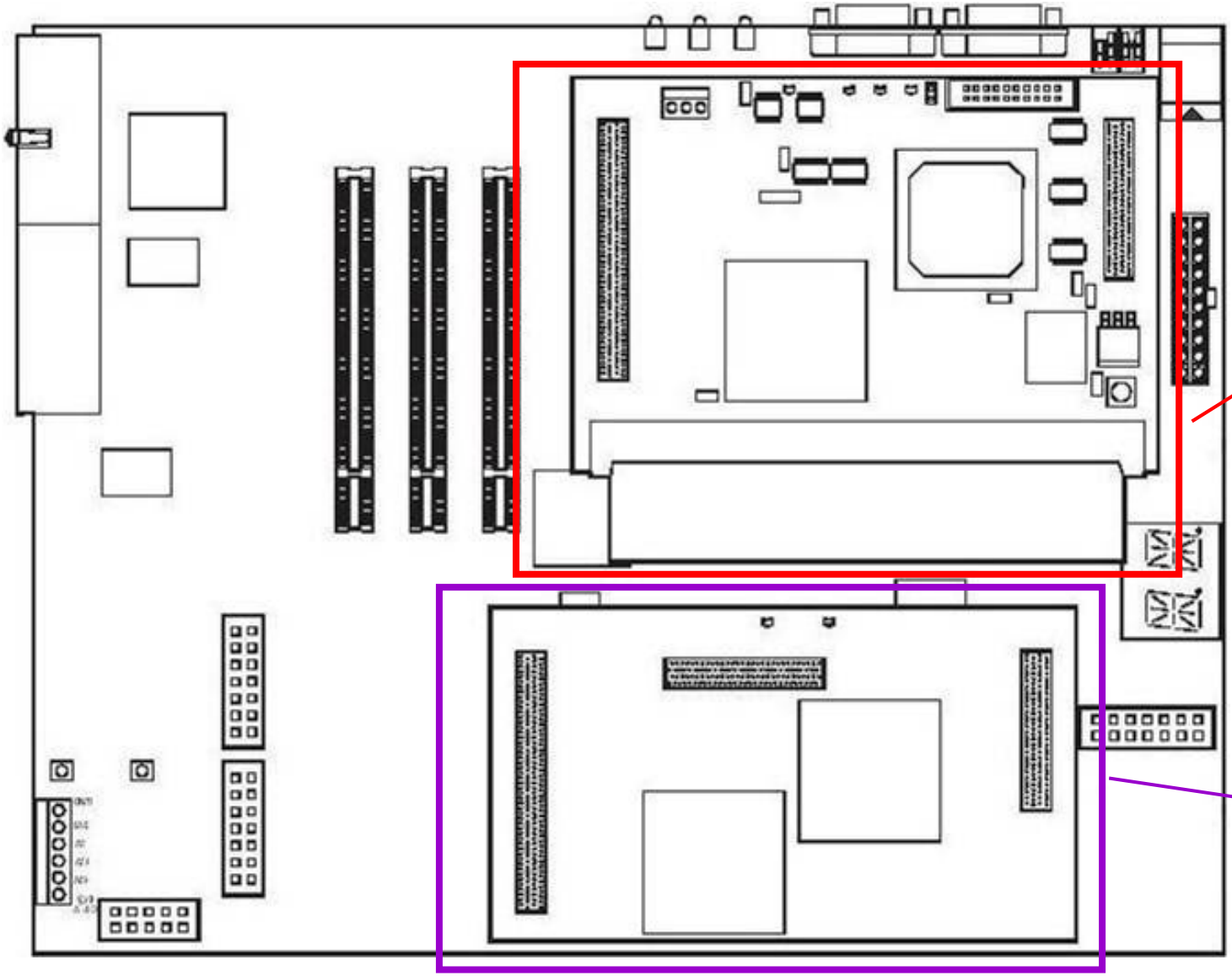


Not to scale

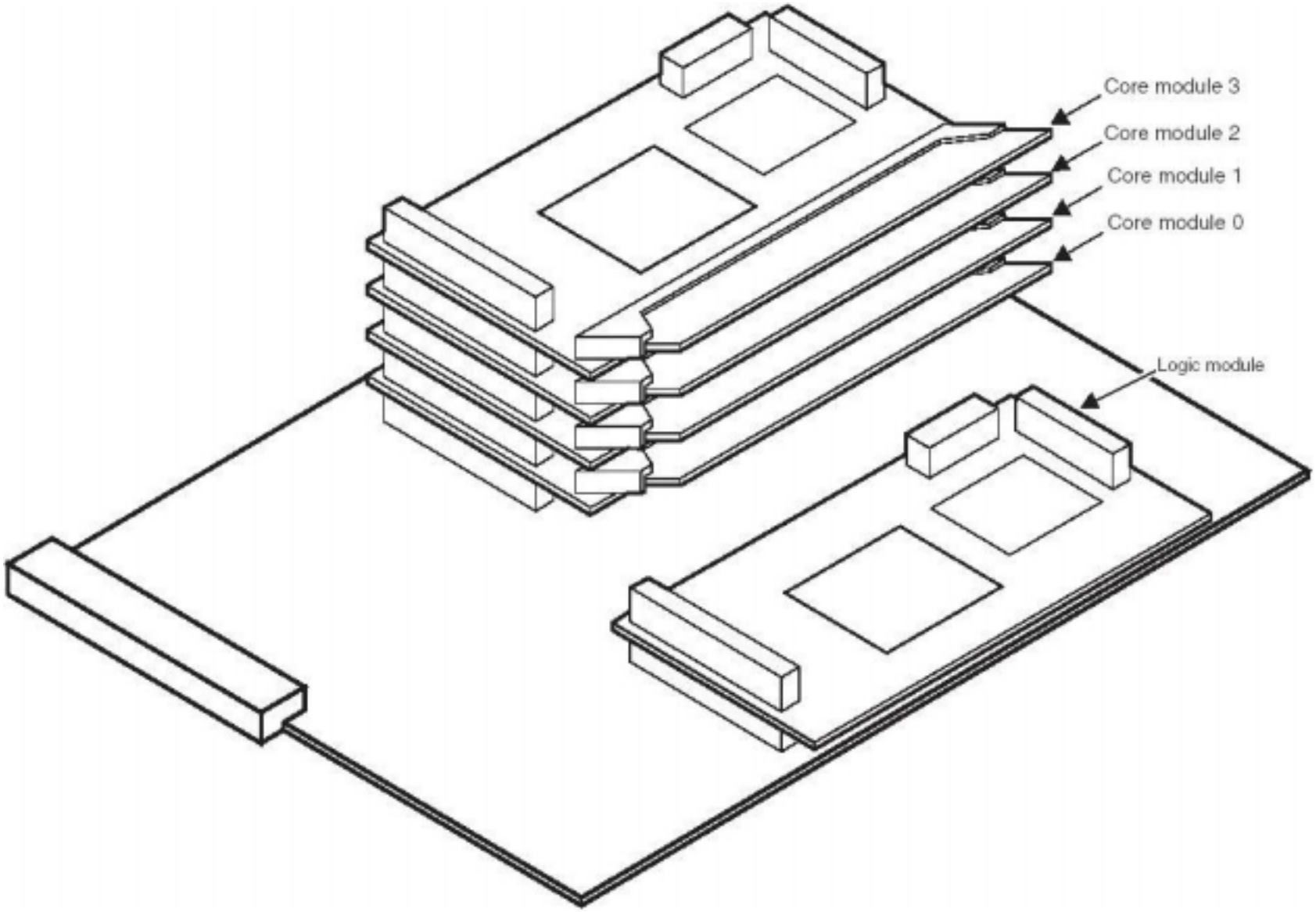
ARM Integrator/AP Block Diagram



Assembled Integrator Development System



Assembled Integrator/AP system



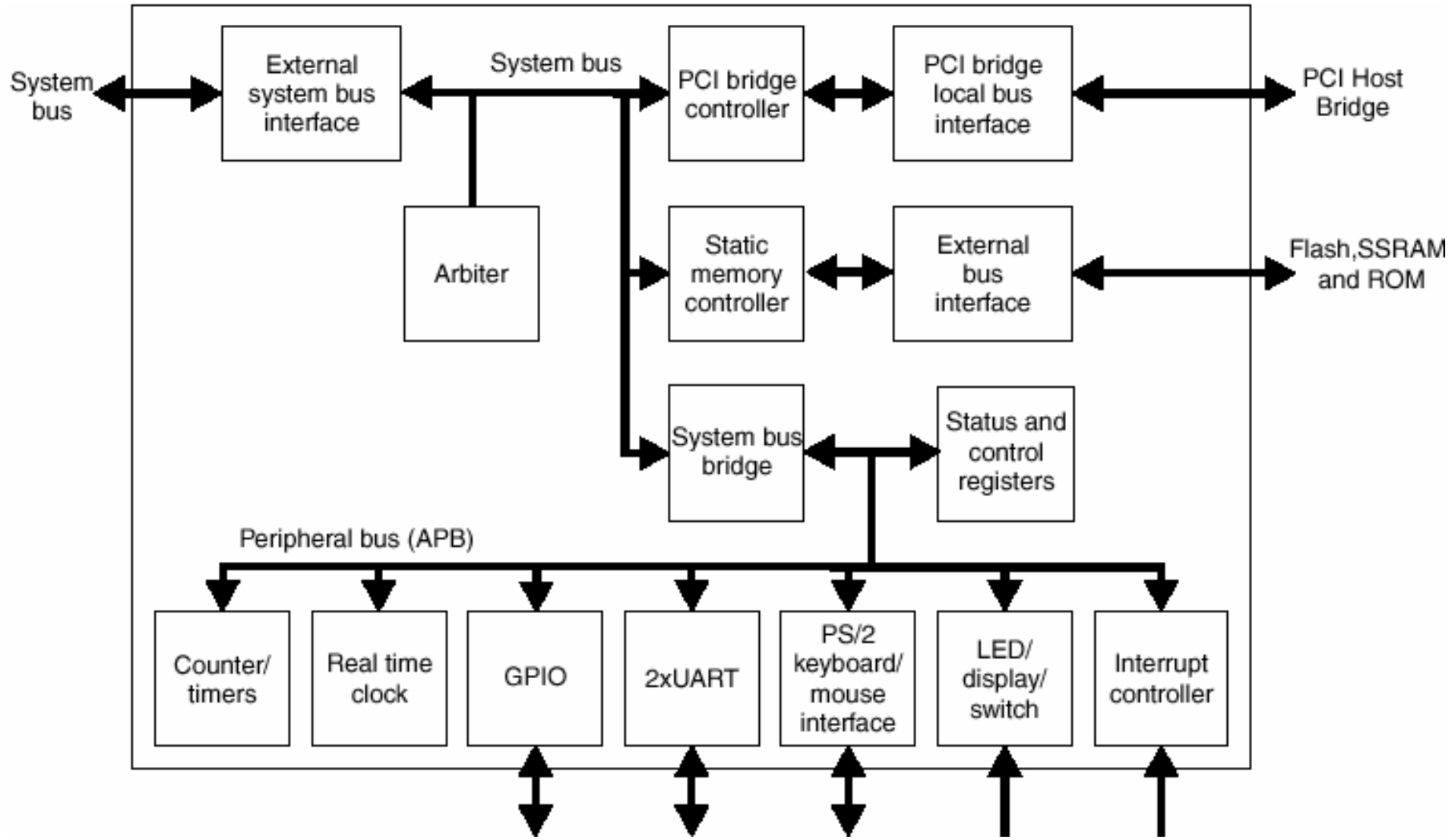
System Controller FPGA (1/2)

- System Bus Interface
  - Supports transfers between system bus and the *Advanced Peripheral Bus* (APB).
  - Supports transfers between system bus and the PCI bus.
  - Supports transfers between system bus and the *External Bus Interface* (EBI).
- System Bus Arbiter
  - Provides arbitration for a total of 6 bus masters.
    - Up to 5 masters on CMs or LMs.
    - PCI bus bridge. (the highest priority)
- Interrupt Controller
  - Handles IRQs and FIQs for up to 4 ARM processors.
  - IRQs and FIQs originate from the peripheral controllers, PCI bus, and other devices on LMs.

System Controller FPGA (2/2)

- Peripheral I/O Controllers
  - 2 ARM PrimeCell UARTs
  - ARM PrimeCell Keyboard & Mouse Interface (KMI)
  - ARM PrimeCell Real Time Clock (RTC)
  - 3 16-bit counter/timers
  - GPIO controller
  - Alphanumeric display and LED control, and switch reader
- Reset Controller
  - Initializes the Integrator/AP when the system is reset
- System Status & Control Register
  - Clock speeds
  - Software reset
  - Flash memory write protection

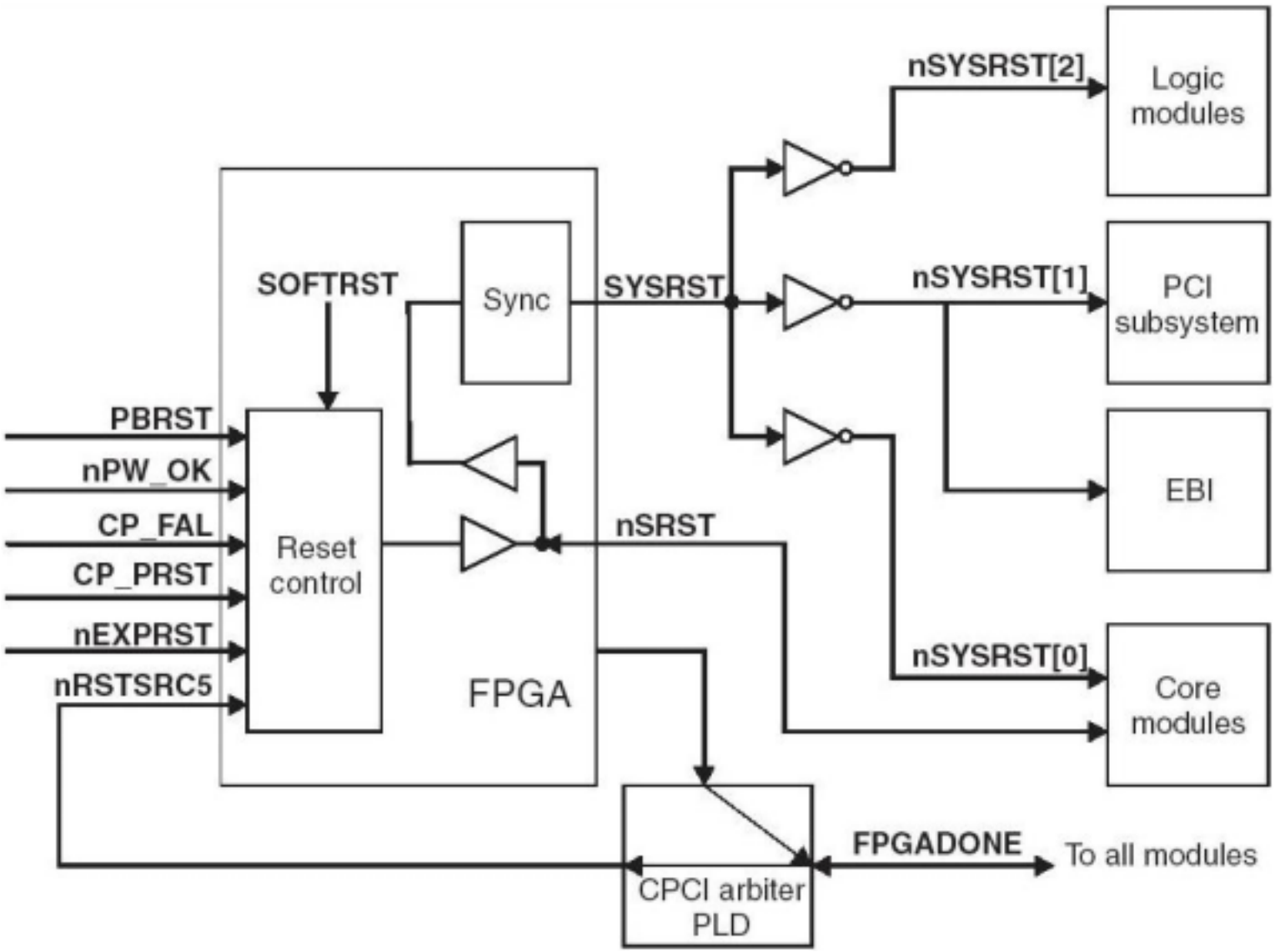
System Controller FPGA Diagram



Reset Controller

- A reset controller is incorporated into the system controller FPGA.
- The hardware reset sources are as follows:
  - Push-button **PBRST** and CompactPCI signal **CP_PRST**
  - ATX power OK signal **nPW_OK** and CompactPCI power fail signal **CP_FAL**
  - **FPGADONE** signal (routed through CPCI arbiter to become **nRSTSRC5**)
  - Logic modules using **nEXPRST**
  - Core modules (and Multi-ICE) using **nSRST**

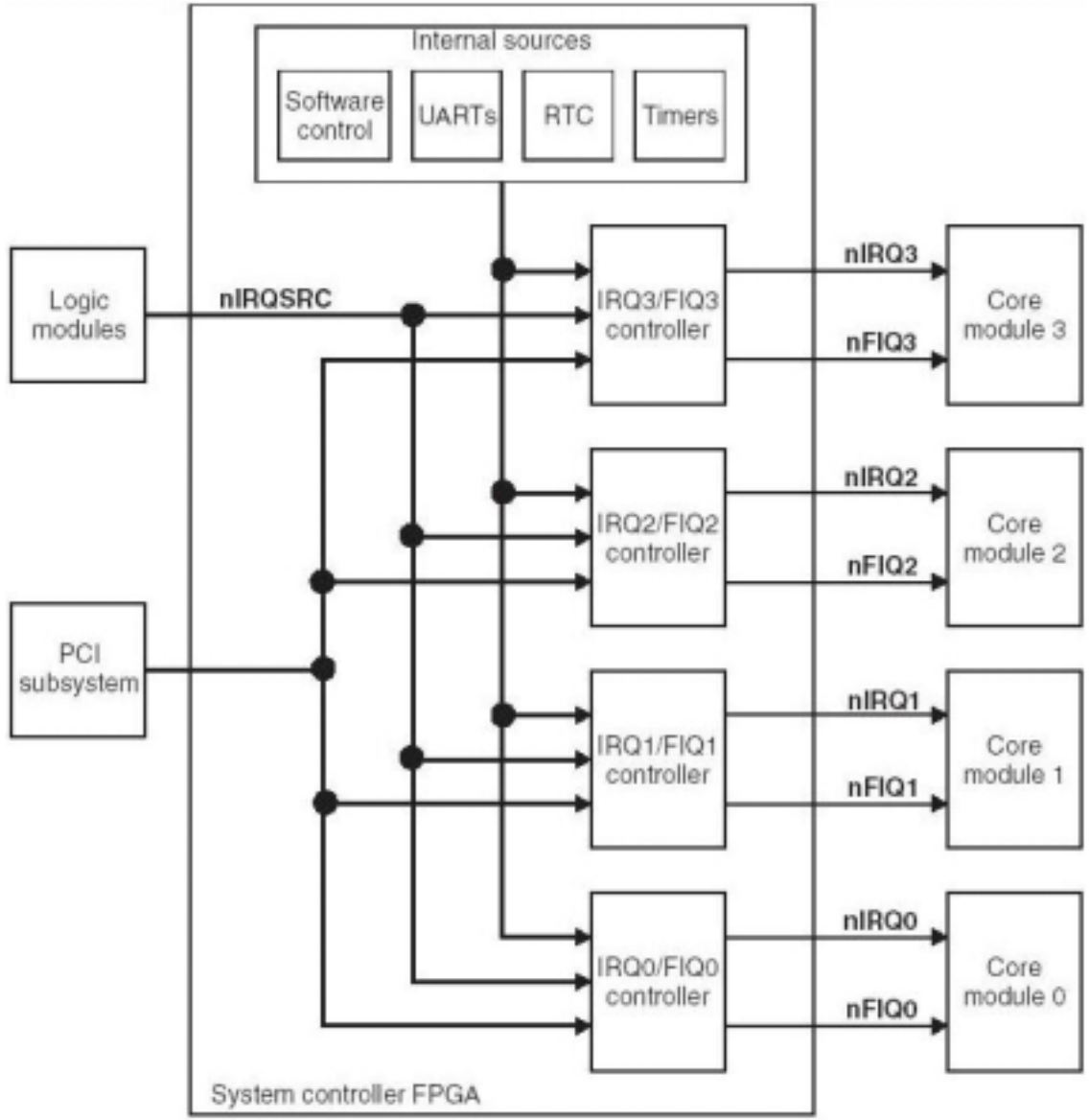
Integrator/AP Reset Control



Interrupt Controller

- The system controller FPGA contains **four** interrupt controllers.
- The system controller incorporates a separate IRQ and FIQ controller for each core module.
- Interrupts are masked enabled, acknowledged, or cleared via registers in the interrupt controller.
- Main sources of interrupts:
  - System controller's internal peripherals
  - LM's devices
  - PCI subsystem
  - Software

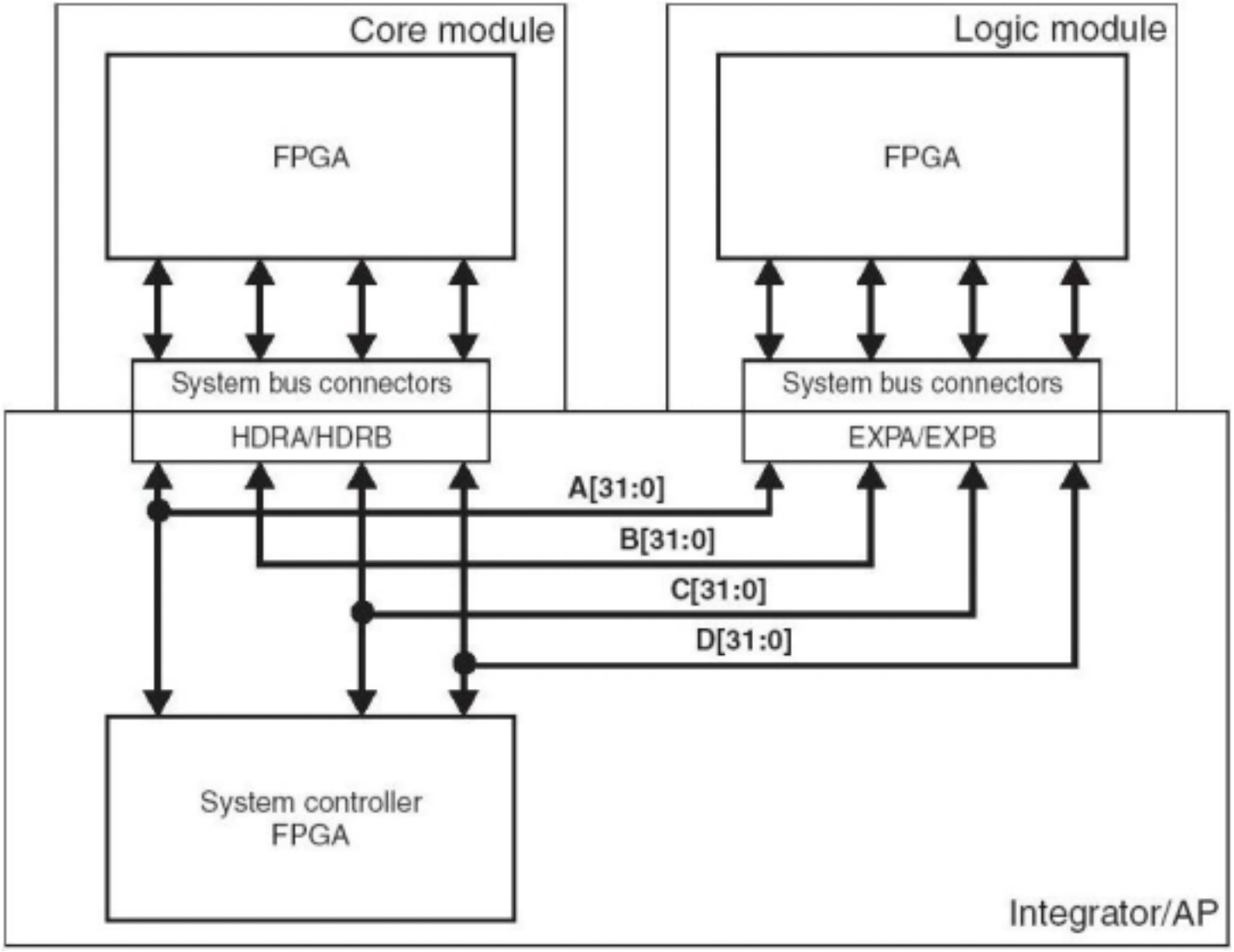
Interrupt Controller Architecture



System Bus

- The HDRA/HDRB and EXPA/EXPB connector pairs are used to connect the system bus between the AP and other modules
  - Core modules on the connectors HDRA and HDRB
  - Logic modules on the connectors EXPA and EXPB
- There are three main system bus (**A[31:0]**, **C[31:0]**, and **D[31:0]**) and fourth bus **B[31:0]**
  - **A[31:0]**: This is the address bus
  - **B[31:0]**: Only connects HDRA to EXPA and reserved for future use
  - **C[31:0]**: Used to implement a system control bus
  - **D[31:0]**: This is the data bus

System Bus Architecture

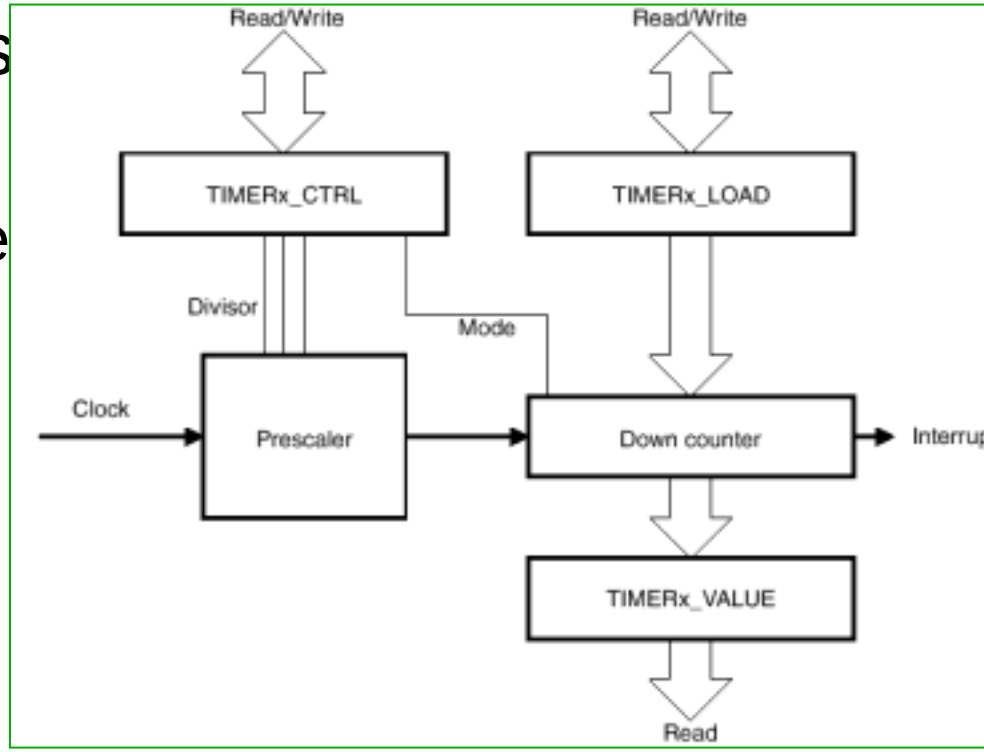


Peripherals

- The peripheral devices incorporated into the system controller FPGA
  - *Counter/timers*
  - *Real-time clock*
  - *UARTs*
  - *Keyboard and mouse interface*
  - *GPIO*

Counter/Timers

- There are **3** counter/timers on an ARM Integrator AP.
- Each counter/timer generates an IRQ when it reaches 0.
- Each counter/timer has
  - A 16-bit down counter with selectable prescale
  - A load register
  - A control register



Counter/Timers Registers (1/2)

- These registers control the **3** counter/timers on the Integrator AP board.
- Each timer has the following registers.
  - **TIMERX_LOAD**: a 16-bit R/W register which is the initial value in free running mode, or reloads each time the counter value reaches 0 in periodic mode.
  - **TIMERX_VALUE**: a 16-bit R register which contains the current value of the timer.
  - **TIMERX_CTRL**: an 8-bit R/W register that controls the associated counter/timer operations.
  - **TIMERX_CLR**: a write only location which clears the timer's interrupt.

Counter/Timers Registers (2/2)

Counter Timer Registers

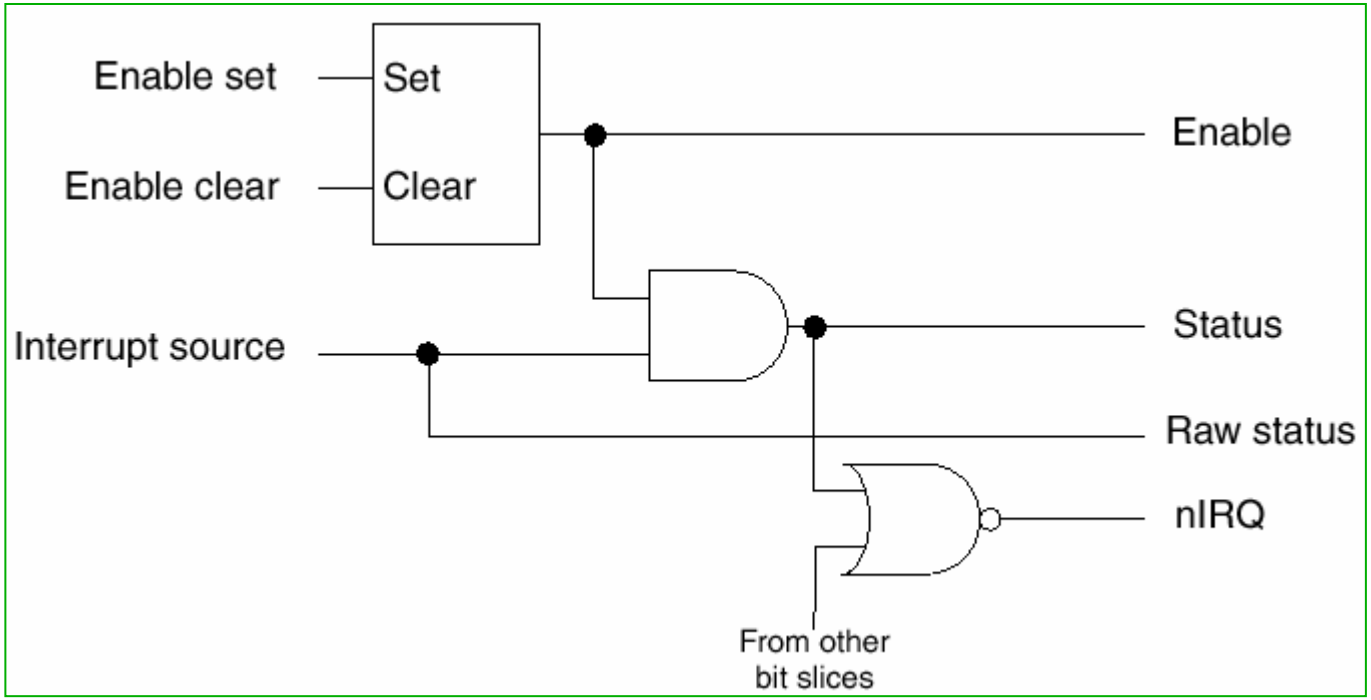
Address	Name	Type	Size	Function
0x13000000	TIMER0_LOAD	R/W	16	Timer0 load register
0x13000004	TIMER0_VALUE	R	16	Timer0 current value reg
0x13000008	TIMER0_CTRL	R/W	8	Timer0 control register
0x1300000C	TIMER0_CLR	W	1	Timer0 clear register

Timer Control Register

Bits	Name	Function
7	ENABLE	Timer enable: 0=disable; 1=enable.
6	MODE	Timer mode: 0=free running; 1=periodic
5:4	unused	Unused, always 0
3:2	PRESCALE	Prescale divisor: 00=none; 01 = div by 16 10=div by 256; 11 = undefined
1:0	Unused	Unused,always 0

The IRQ and FIQ Control Registers

- Implemented in the system controller FPGA.
- Provides interrupt handling for up to 4 processors.
- There's a 22-bit IRQ and FIQ controller for each processor.



IRQ Registers (1/2)

- The registers control each processor's interrupt handler on the Integrator AP board.
- Each IRQ has following registers:
  - **IRQX_STATUS**: a 22-bit register representing the current masked IRQ status.
  - **IRQX_RAWSTAT**: a 22-bit register representing the raw IRQ status.
  - **IRQX_ENABLESET**: a 22-bit location used to set bits in the enable register.
  - **IRQX_ENABLECLR**: a 22-bit location used to clear bits in the enable register.

IRQ Registers (2/2)

❑ IRQ Registers

Address	Name	Type	Size	Function
0x14000000	IRQ0_STATUS	R	22	IRQ0 status
0x14000004	IRQ0_RAWSTAT	R	22	IRQ0 IRQ status
0x14000008	IRQ0_ENABLESET	R/W	22	IRQ0 enable set
0x1400000C	IRQ0_ENABLECLR	W	22	IRQ0 enable clear

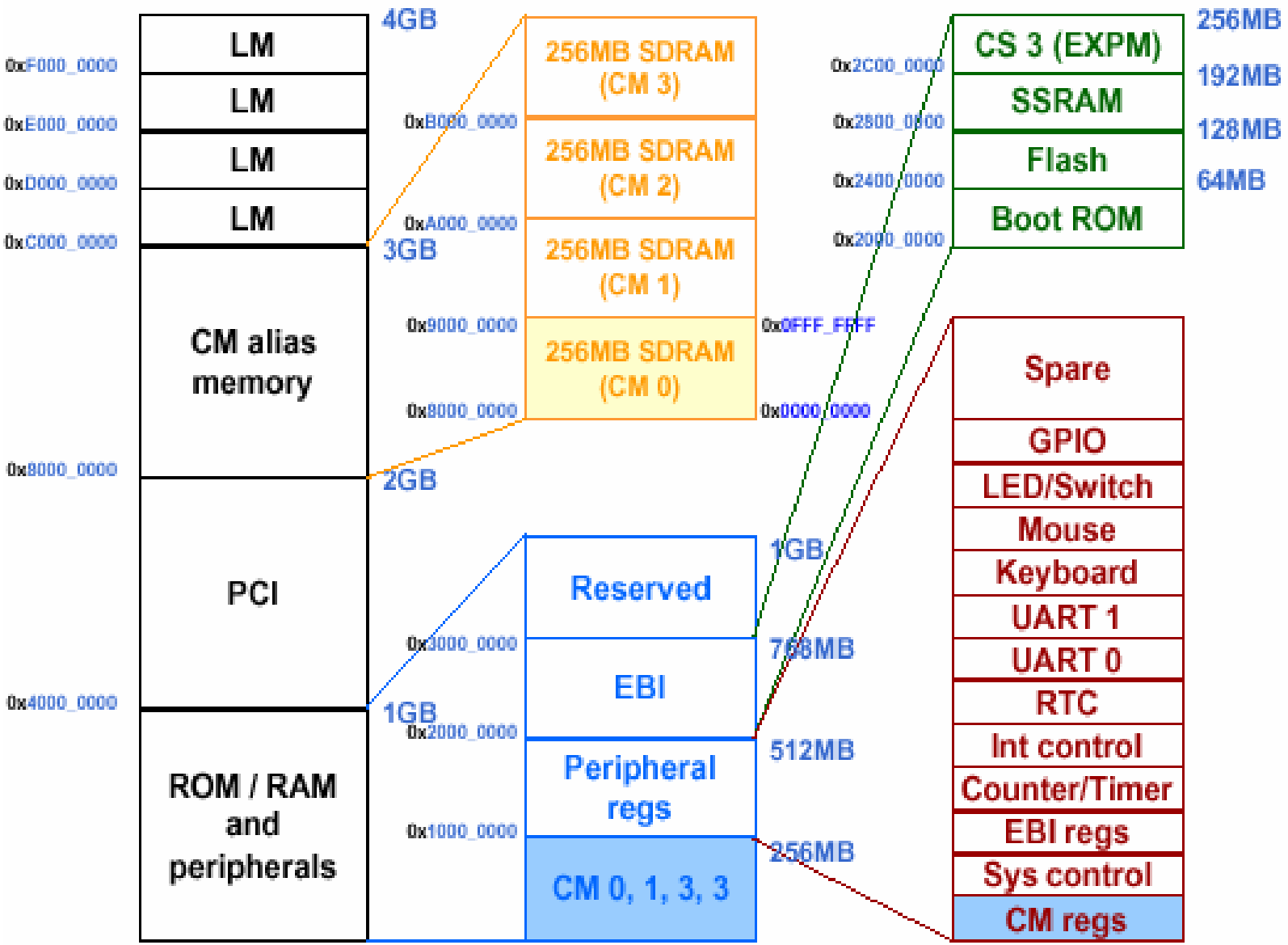
❑ IRQ Registers bit assignments

Bit	Name	Function
0	SOFTINT	Software interrupt
5	TIMERINT0	Counter/Timer interrupt
6	TIMERINT1	Counter/Timer interrupt
7	TIMERINT2	Counter/Timer interrupt

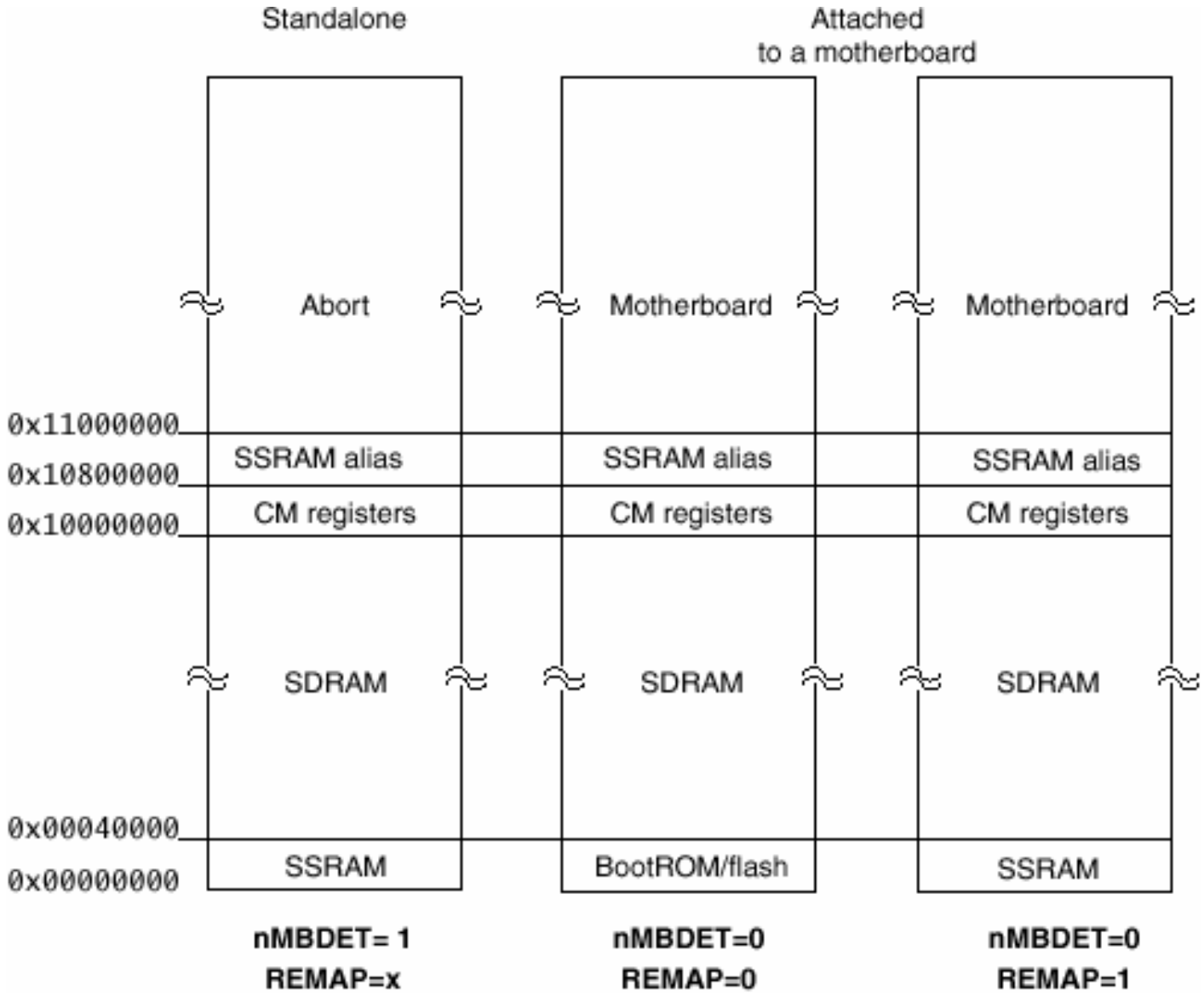
Outline

- ARM Integrator Core Module (CM) [1]
- ARM Integrator Logic Module (LM) [2]
- ARM Integrator ASIC Application Platform (AP) [3]
- ***System Memory Map [1]***
- Lab3 – Core Peripheral

System Memory Map



Core Module Memory Map



Outline

- ARM Integrator Core Module (CM) [1]
- ARM Integrator Logic Module (LM) [2]
- ARM Integrator ASIC Application Platform (AP) [3]
- System Memory Map [1]
- ***Lab3 – Core Peripheral***

Lab 3: Core Peripherals

- Goal
  - Understand available resource of ARM Integrator
    - Integrator/AP
    - Core Module (CM)
    - Logic Module (LM)
    - Memory-mapped device
    - Timer/Interrupt
- Principles
  - ARM ASIC Platform Resources
  - Semihosting
  - Interrupt handler
  - Architecture of Timer and Interrupter controller
- Guidance
  - Introduction to Important functions used in interrupt handler
- Steps
  - The same to that of code development
- Requirements and Exercises
  - Modified the C program. We use Real-Time Clock instead of timer to show our IRQ0 values.
- Discussion
  - How to use multi-timer/interrupt.

Timer/Interrupt example without uHAL

- Several important functions are used in this example:
  - **Install_Handler**: This function install the IRQ handler at the branch vector table at 0x18.
  - **myIRQHandler**: This is the user's IRQ handler. It performs the timer ISR in this example.
  - **enableIRQ**: The IRQ enable bit in the CPSR is set to enable IRQ.
  - **LoadTimer, WriteTimerCtrl, ReadTimer, ClearTimer**: Timer related functions.

References

[1] http://twins.ee.nctu.edu.tw/courses/ip_core_02/index.html
[2] DUI0126B_CM7TDMI_UG.pdf
[3] LM-XCV2000E.pdf
[4] DUI0098B_AP_UG.pdf.