Standard Microcontroller

1-day ARM Training

Paolo Bernasconi

NXP Semiconductors, FAE

Agenda

9:30 LPC2000 Technical Training Part I

- Introduzione
- LPC2000 devices and roadmaps dev tools. I nuovi dispositivi della famiglia LPC2300 e LPC2400. Novità per il 2007

☺ **Break**

- Presentazione dell'Architettura ARM7 (mappa di memoria, system control, peripherals)

12:30 ☺ Pranzo

13:30 LPC2000 Technical Training Part II

- Inizializzazione delle periferiche di sistema PLL, Vector Interrupt Controller e USB

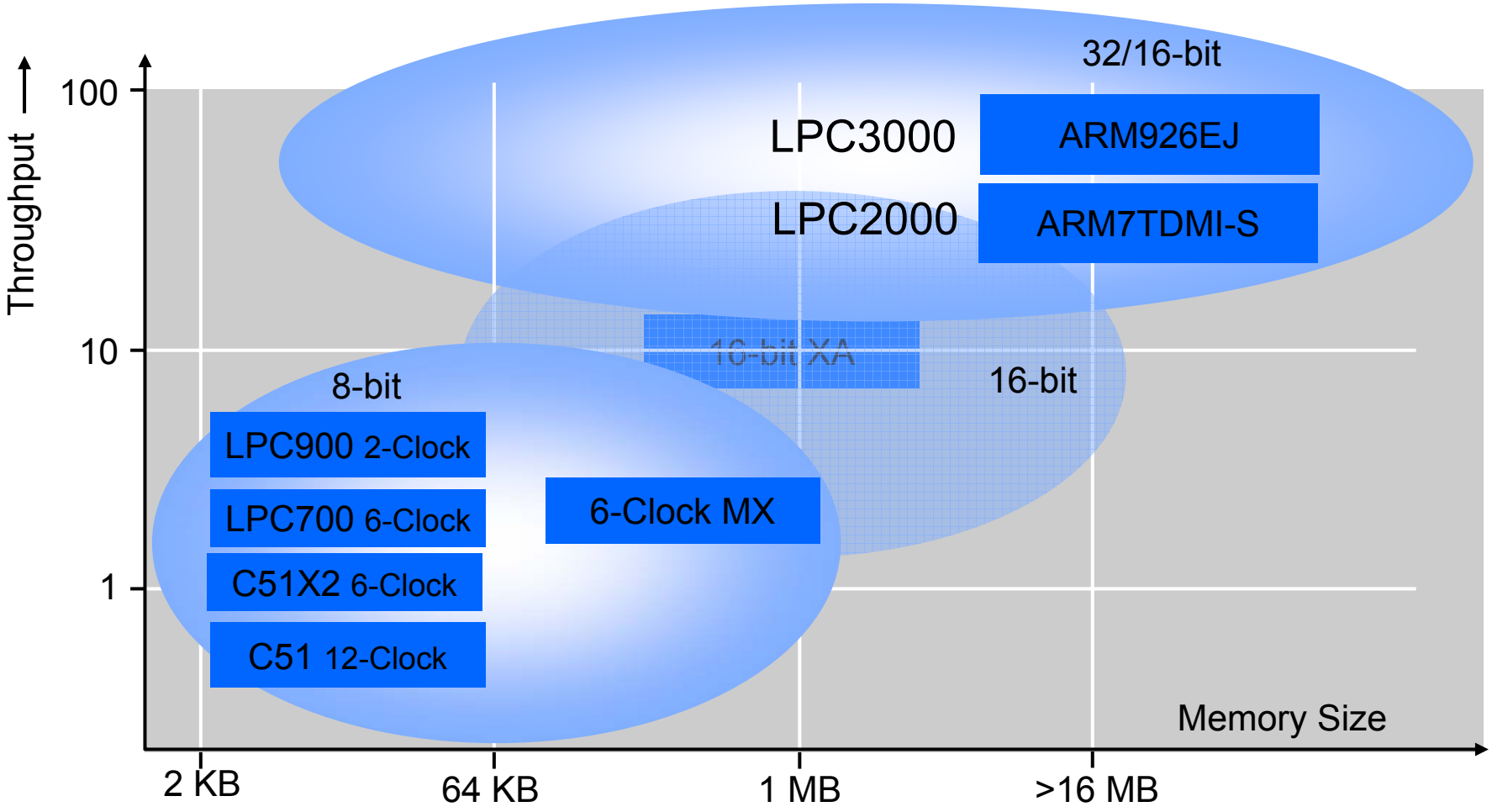
- presentazioni di esempi con tool Keil e scheda di valutazione Keil

☺ **Break**

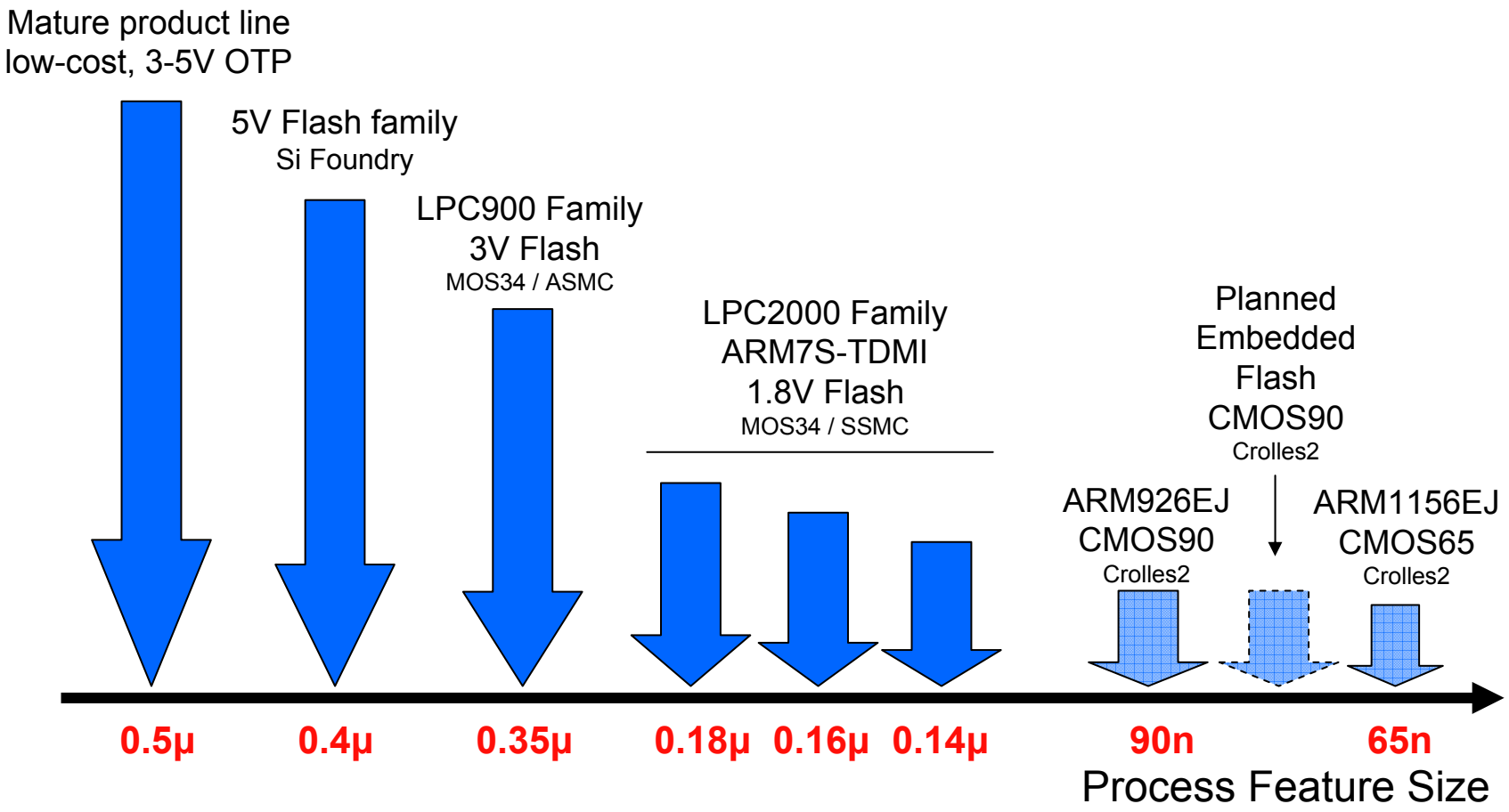
- Implementazone dell'architettura ARM7 nella famiglia LPC23/24
- Q&A

17:30 Chiusura

Standard Microcontroller Cores



NXP Embedded Flash Process Roadmap

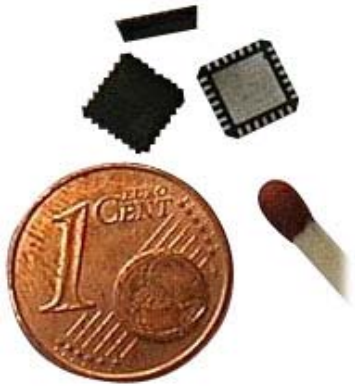


Standard Microcontroller

Strategy Summary

Develop Innovative and Cost Effective Products

- Focus on 16/32-bit market with wide range of ARM7 & ARM9 based products
- Expand the successful LPC Family approach:
  - New peripherals like USB, Ethernet, ...
- Use highly competitive flash based processes:
  - 0.35 μm and 0.18 μm Flash in production
  - Shrink path down to 0.14 μm
  - First products in 90nm in 2005
- Introduction of innovative packages:
  - Chip scale packages like HVSON10, TFBGA256



LPC2000 Family
16/32-bit ARM7TDMI-S Products

ARM Microcontrollers

NXP has developed a family of ARM-based Microcontrollers

For

- Low-Cost High Volume Applications

With

- Embedded Flash and SRAM
- On-board AMBA-bus Peripherals (Adv. µC Bus Architecture)
- Real-Time Deterministic behavior (no Cache required)
  - **High performance NXP specific Flash Memory matrix**
  - **and access design**
- Full Debug, Real-time Monitoring and Trace facilities

To

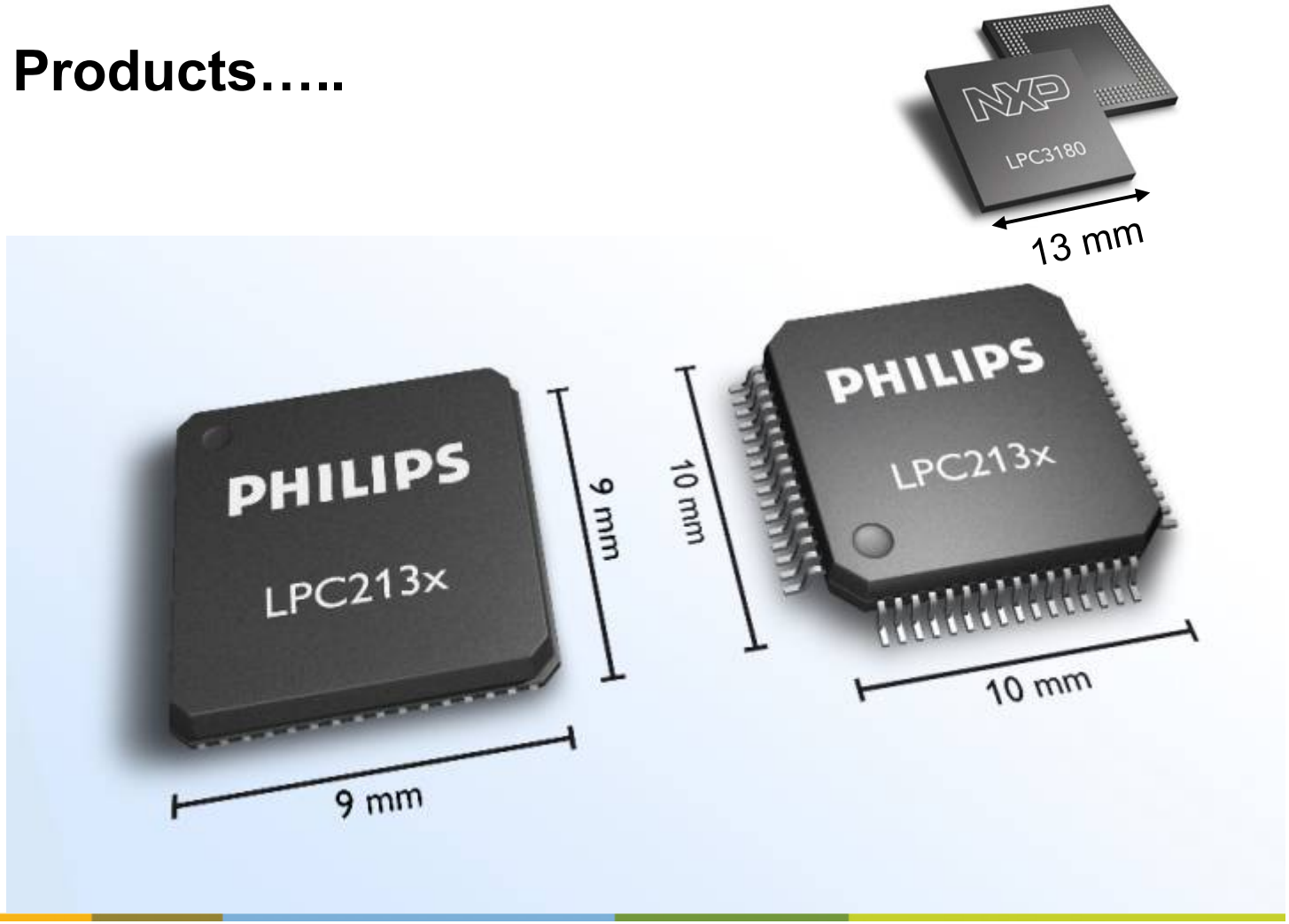
- Continue on from our successful 8-bit 80C51 Family
- Enable new low-cost 16/32-bit Microcontroller-based applications

NXP Standard Microcontrollers

The Ultimate Products…..

LPC2000

LPC3000



History - NXP a leader in ARM

NXP relationship with ARM Ltd. spans a decade

- One of the three founding partners of ARM
- Development with cores starting from ARM2 through ARM11

NXP offers the most experience

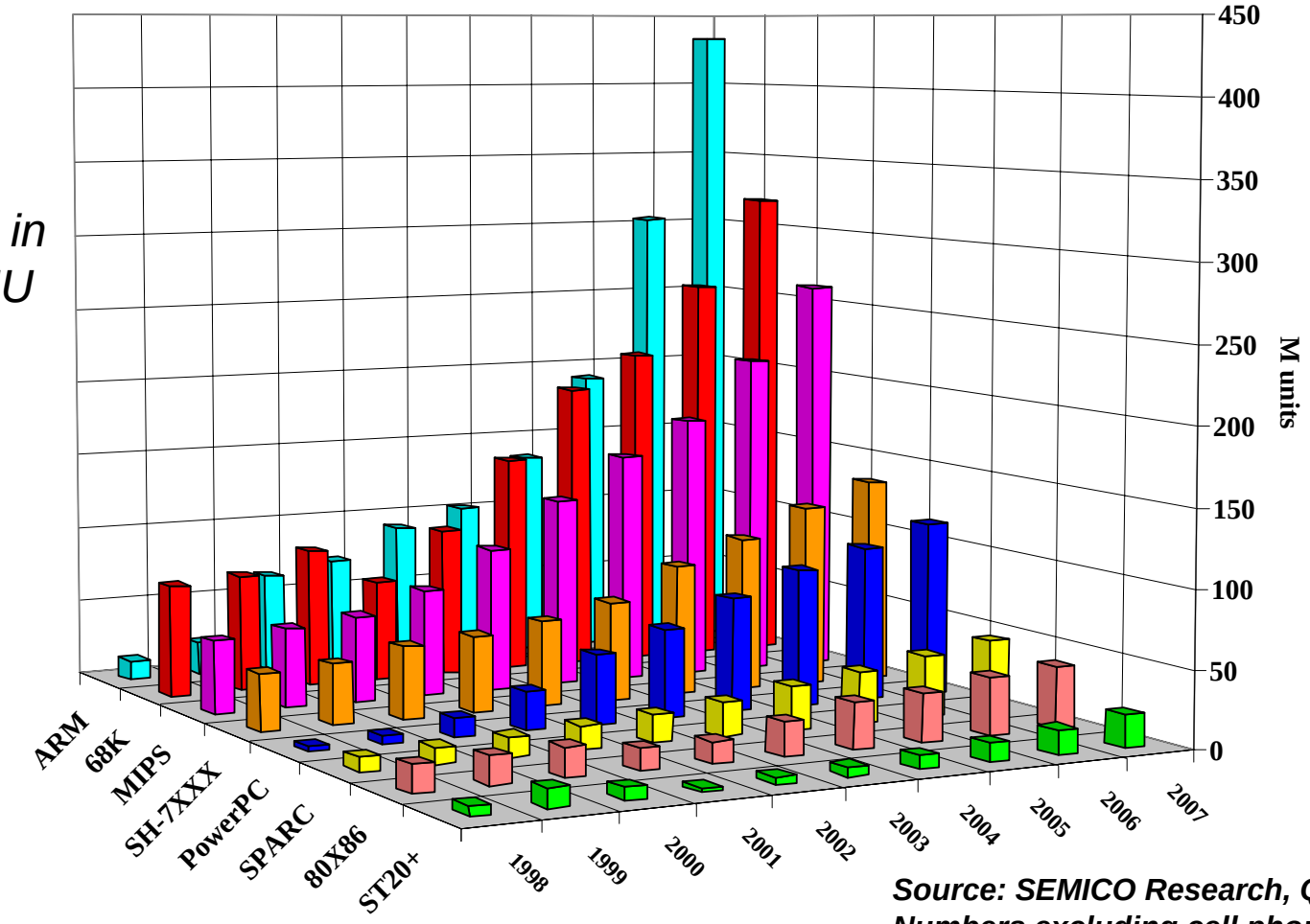
- Over 250 ARM designs - more than anyone else in the industry
- In Top 3 for ARM shipments worldwide
- More than a dozen ARM cores in over 7 CMOS processes

NXP is a long-term ARM licensee

- Extensive license relationship provides continuous access to all architectures
- Announcing off-the-shelf ARM microcontrollers with embedded Flash

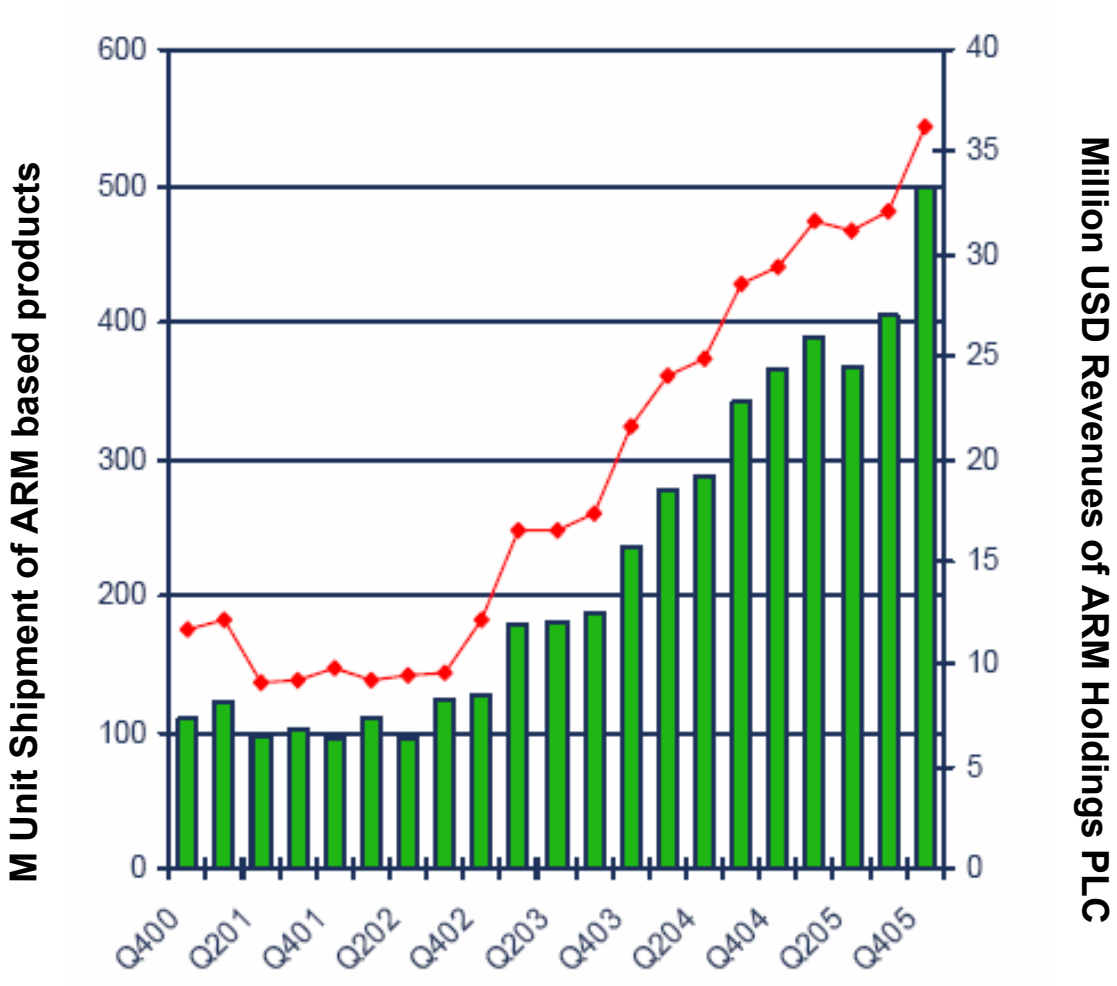
ARM vs. other 32/16-bit Emb. Architectures

ARM Volume in Y2004: 190 MU



Source: SEMICO Research, Q4 2002
Numbers excluding cell phone handsets

Shipments of all ARM products



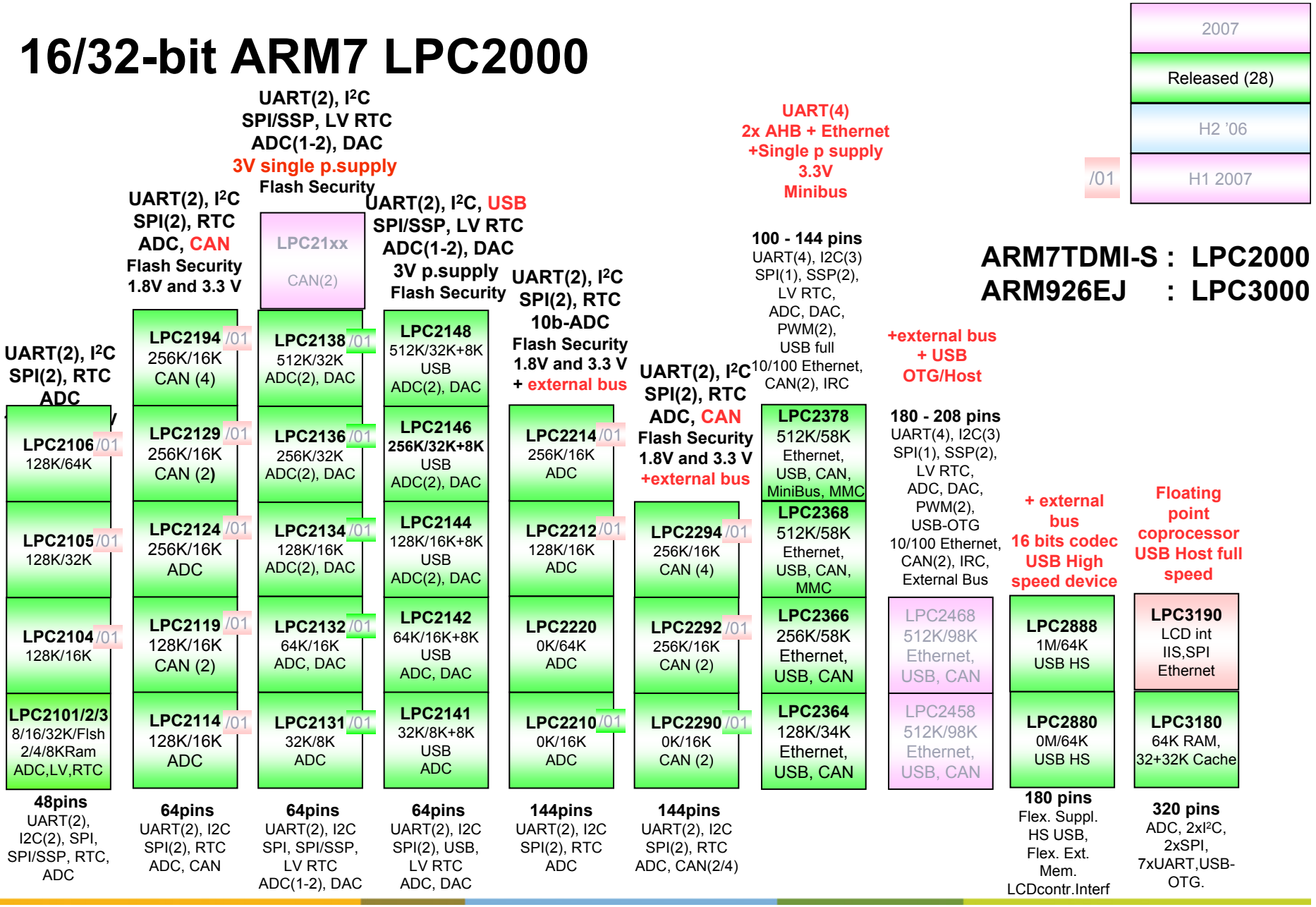
ARM's partners shipped 1,662 Mpcs in 2005 (+31%)

ARM7TDMI-S NXP Choice

The ARM7TDMI-S is based on an ARM7 core

- T- Thumb architecture extension
  - ARM Instructions are all 32 bit
  - Thumb instructions are all 16 bit
  - Two execution states to select which instruction set to execute
- D- Core has debug extensions
- M- Core has enhanced multiplier
- I- Core has Embedded ICE Macrocell
- S- Fully synthesis able

16/32-bit ARM7 LPC2000



Timing/features/packages of non released parts may change without prior notification

Roadmap – 32-bit portfolio

Recently released

Roadmap

Functionality →

LPC31xx: ARM9
- SPI
- IIS interface
- Ethernet MAC controller
- LCD interface

LPC3190

LPC3000

LPC3000

LPC24xx: ARM7
- Ethernet (MII+RMII)
- USB FS Device
- USB Host/OTG
- 2 x CAN
- Ext. Memory (SDRAM, SRAM)
- 96K SRAM

LPC2468

LPC2458

LPC2000 LCD

LPC2368

LPC2378

LPC2366

LPC23xx: ARM7
- Ethernet (RMII)
- USB FS Device
- 2 x CAN

LPC2364

LPC22xx/01

LPC21xx/01

LPC210x/01

Recently released

New releases

Feature and performance improvements

2007

Time →

/00 Versions Status

- Reset.1 bug fixed
- LPC2104/00 and LPC2105/00 are Indus. Temp. range qualified

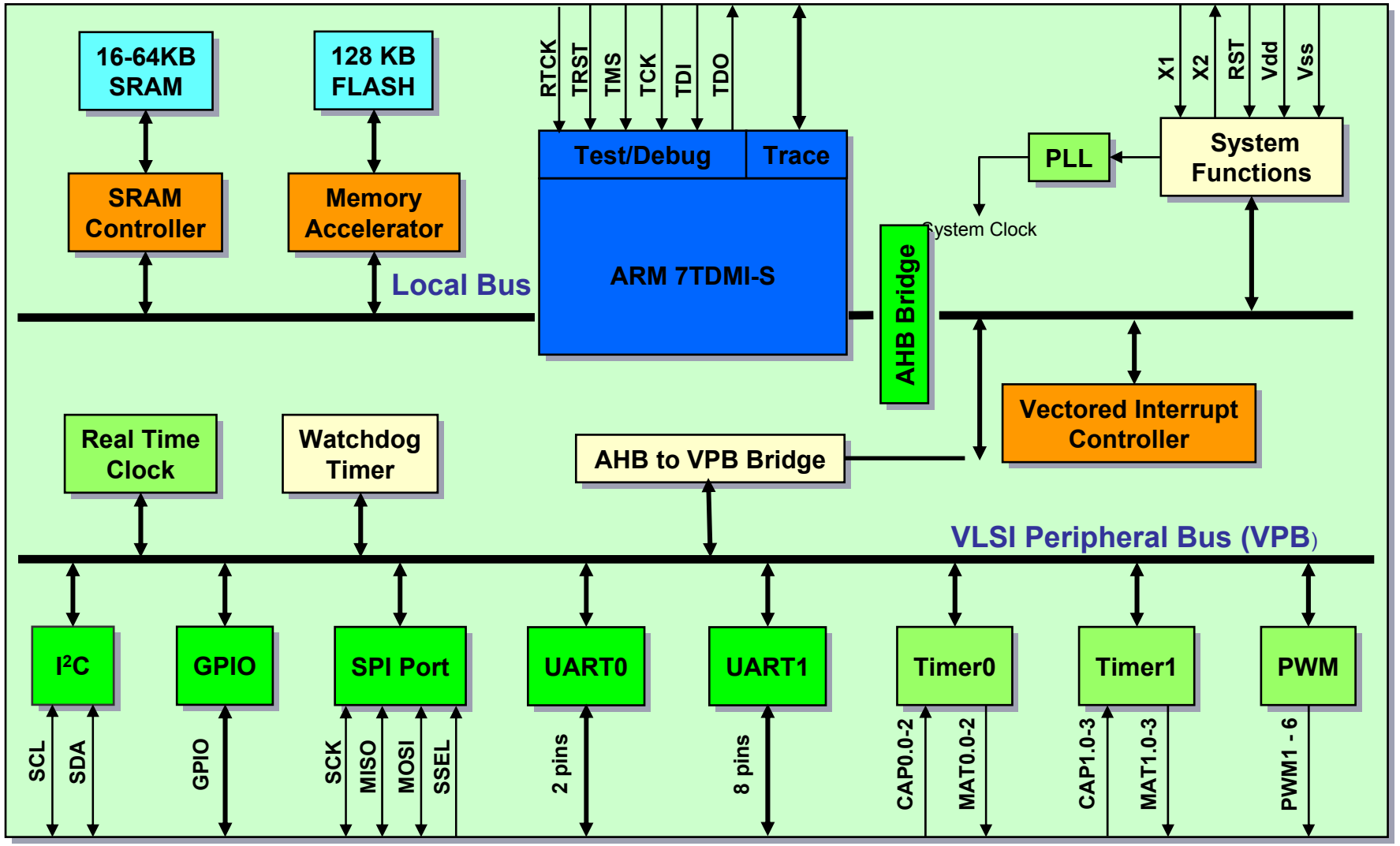
Parts	Samples	Status
LPC2104FBD48/00	Yes	RFS
LPC2105FBD48/00	Yes	RFS
LPC2106FBD48/00	Yes	RFS
LPC2106FHN48/00	Yes	RFS
LPC2114FBD64/00	Yes	RFS
LPC2124FBD64/00	Yes	RFS
LPC2119FBD64/00	Yes	RFS
LPC2129FBD64/00	Yes	RFS
LPC2194HBD64/00	Yes	RFS
LPC2212FBD144/00	Yes	RFS
LPC2214FBD144/00	Yes	RFS
LPC2292FBD144/00	Yes	RFS
LPC2292FET144/00	Yes	RFS
LPC2294HBD144/00	Yes	RFS

/01 Versions Status

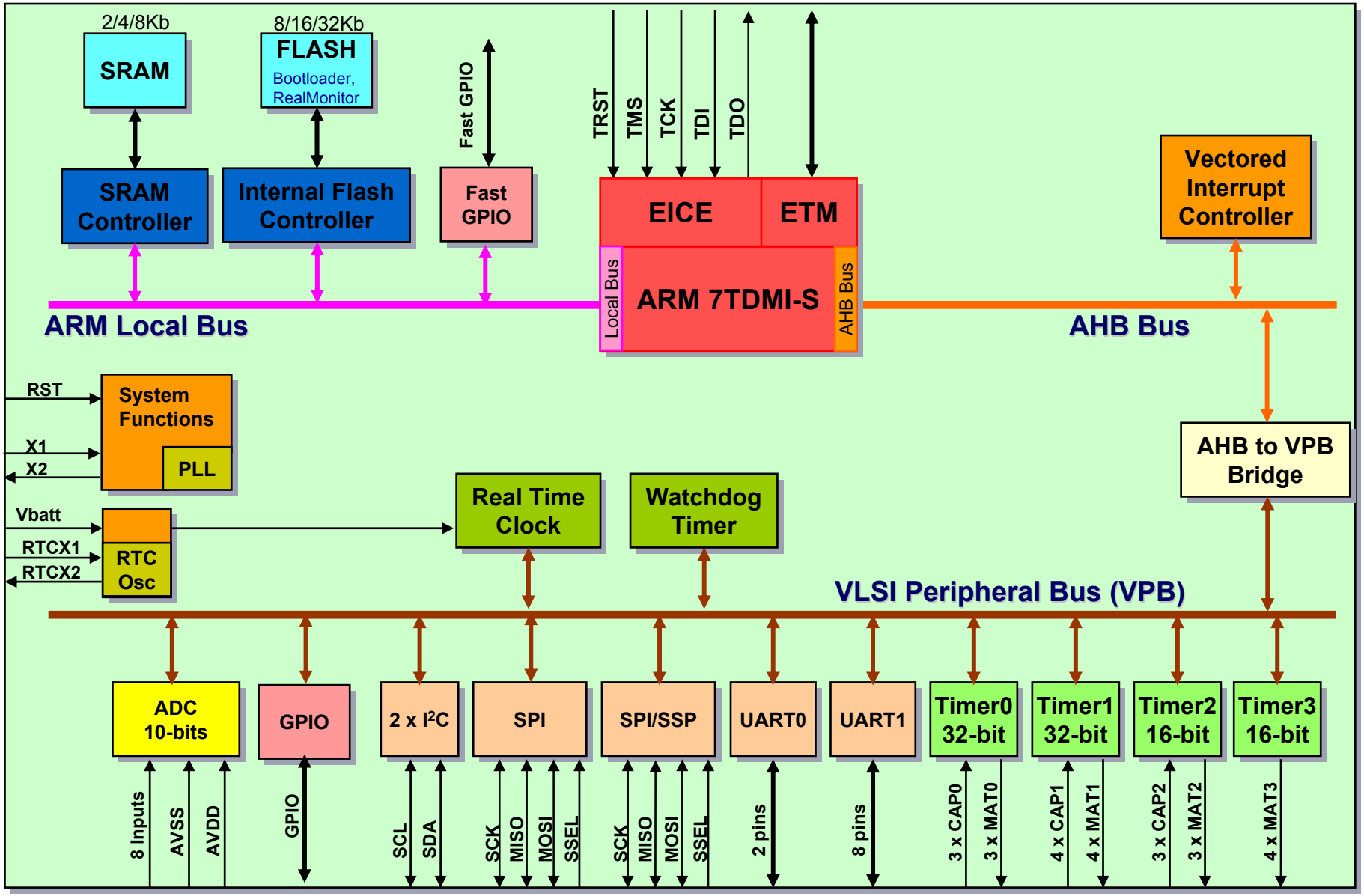
- All bugs corrected excepted core.1
- Some enhanced features:
  - ✓ Fast I/O (3-4 times faster than standard)
  - ✓ Counter inputs
  - ✓ Dedicated result registers per ADC input
  - ✓ UART improvements
  - ✓ Program security (for the LPC210x)
  - ✓ …

Parts	Samples	Status	RFS date
LPC2104FBD48/01	No	Dev	H1/07
LPC2105FBD48/01	No	Dev	H1/07
LPC2106FBD48/01	No	Dev	H1/07
LPC2106FHN48/01	No	Dev	H1/07
LPC2114FBD64/01	No	Dev	H1/07
LPC2124FBD64/01	No	Dev	H1/07
LPC2119FBD64/01	No	Dev	H1/07
LPC2129FBD64/01	No	Dev	H1/07
LPC2131FBD64/01	Yes	RFS	Now
LPC2132FBD64/01	Yes	RFS	Now
LPC2132FHN64/01	Yes	RFS	Now
LPC2134FBD64/01	Yes	RFS	Now
LPC2136FBD64/01	Yes	RFS	Now
LPC2138FBD64/01	Yes	RFS	Now
LPC2138FHN64/01	No	RFS	Now
LPC2194HBD64/01	No	Dev	H1/07
LPC2210FBD144/01	Yes	RFS	Now
LPC2290FBD144/01	Yes	RFS	Now
LPC2212FBD144/01	No	Dev	H1/07
LPC2214FBD144/01	No	Dev	H1/07
LPC2292FET144/01	No	Dev	H1/07
LPC2294HBD144/01	No	Dev	H1/07

LPC2104/5/6 Block Diagram



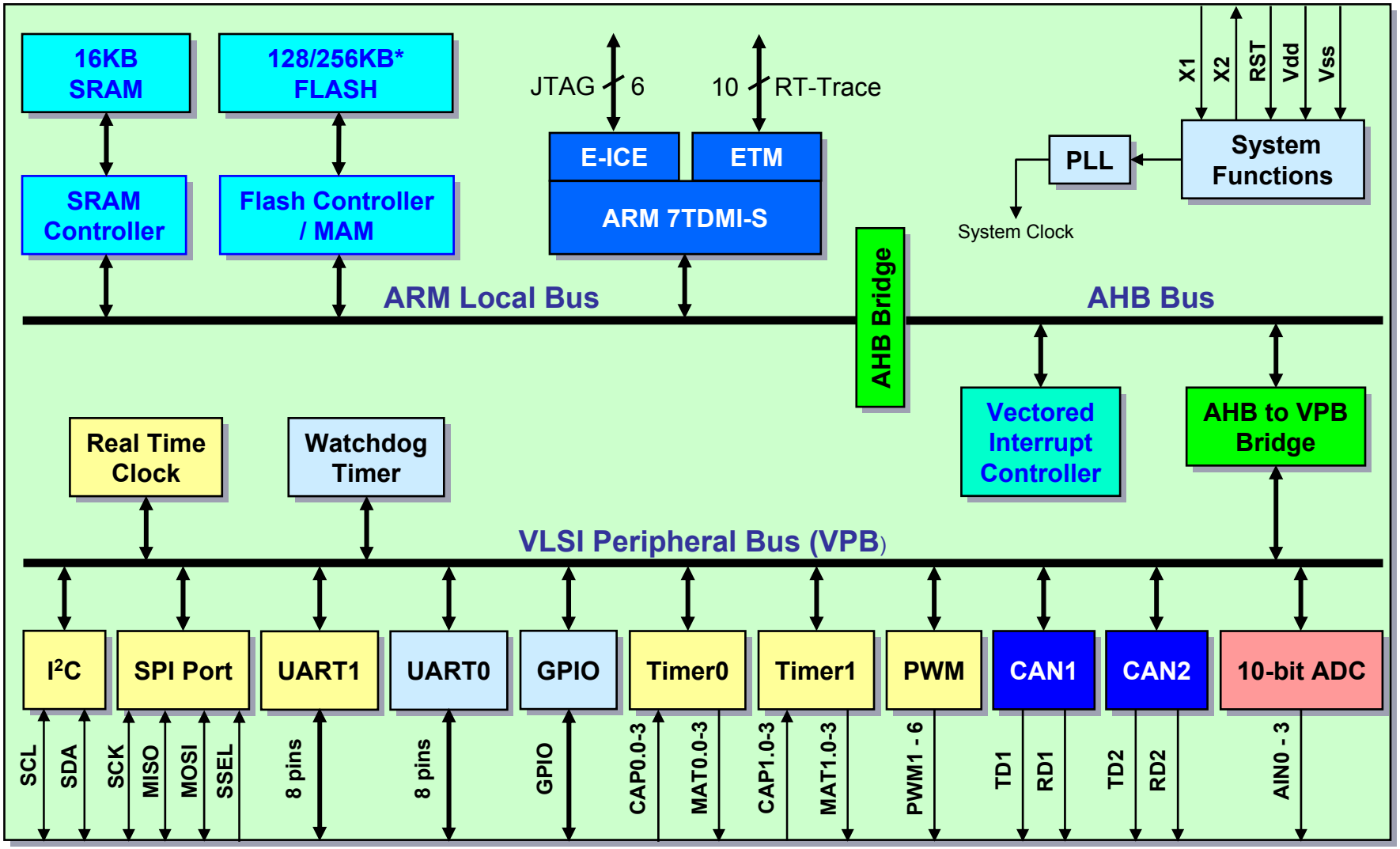
LPC2101/2/3 Blocks



*Flash size: LPC2119 128KB
LPC2129,94 256KB

LPC2119, LPC2129,LPC2194

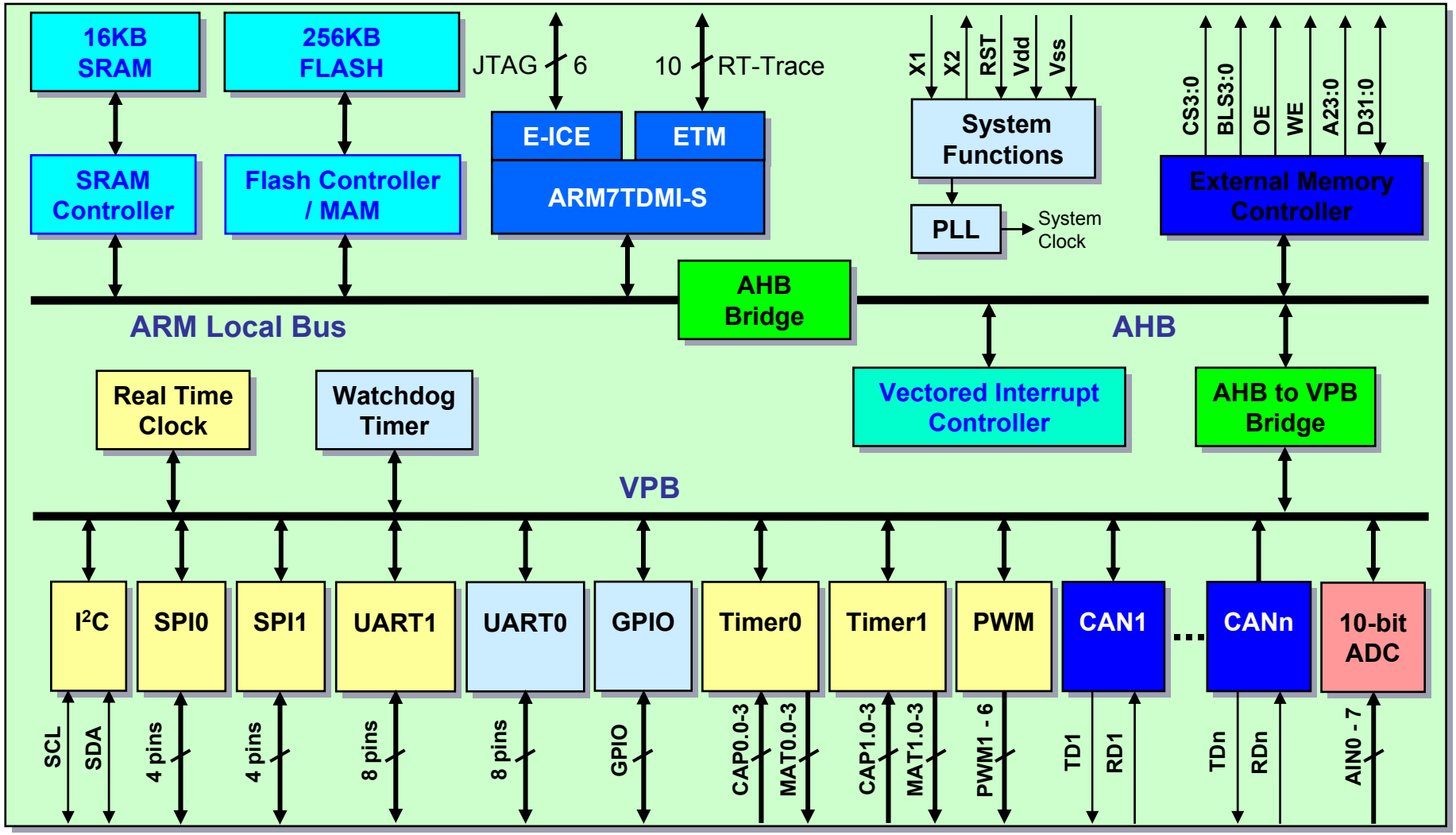
Package:
LQFP64



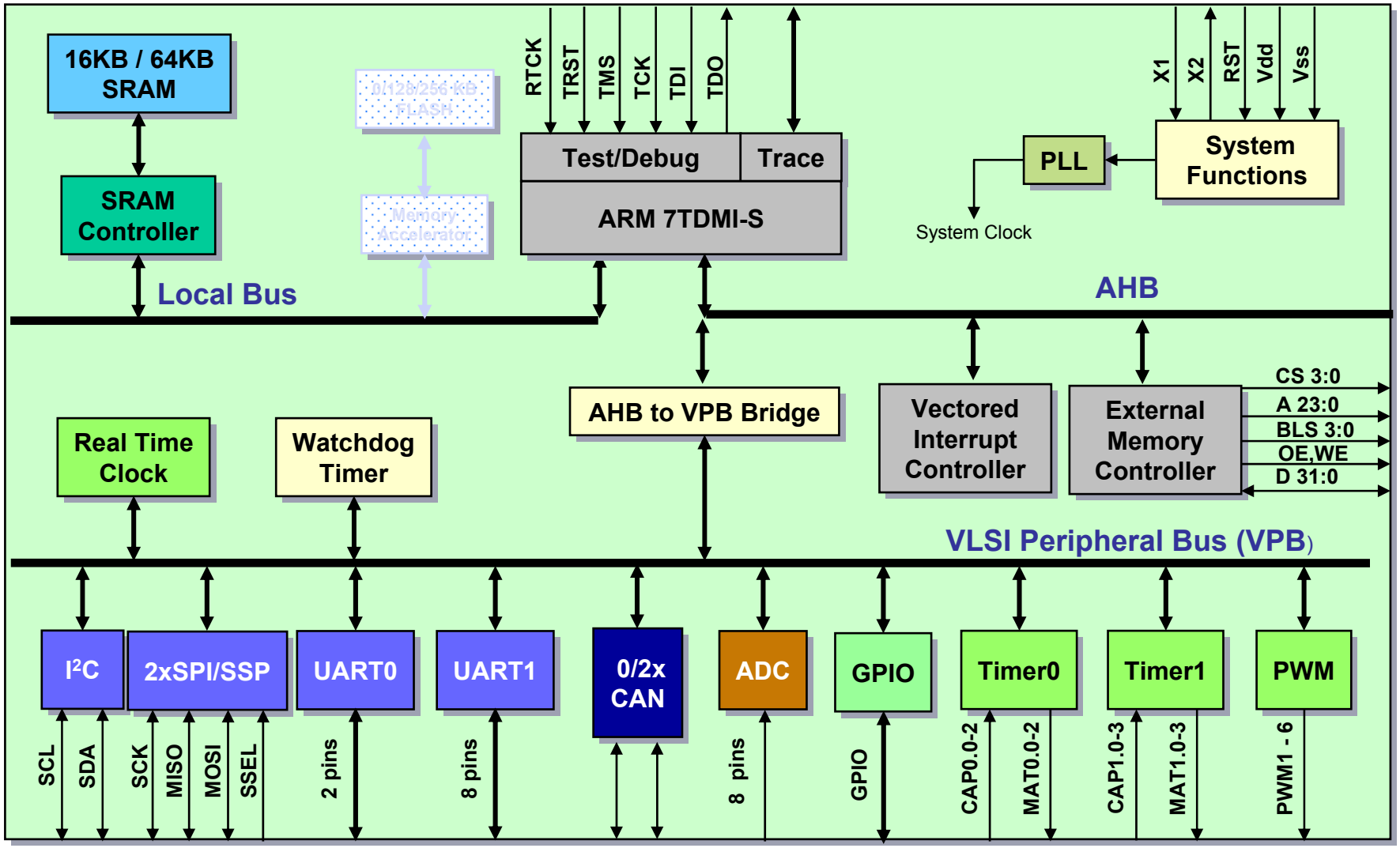
n: LPC2292 = 2
LPC2294 = 4

LPC2292, LPC2294

144-Pin Packages

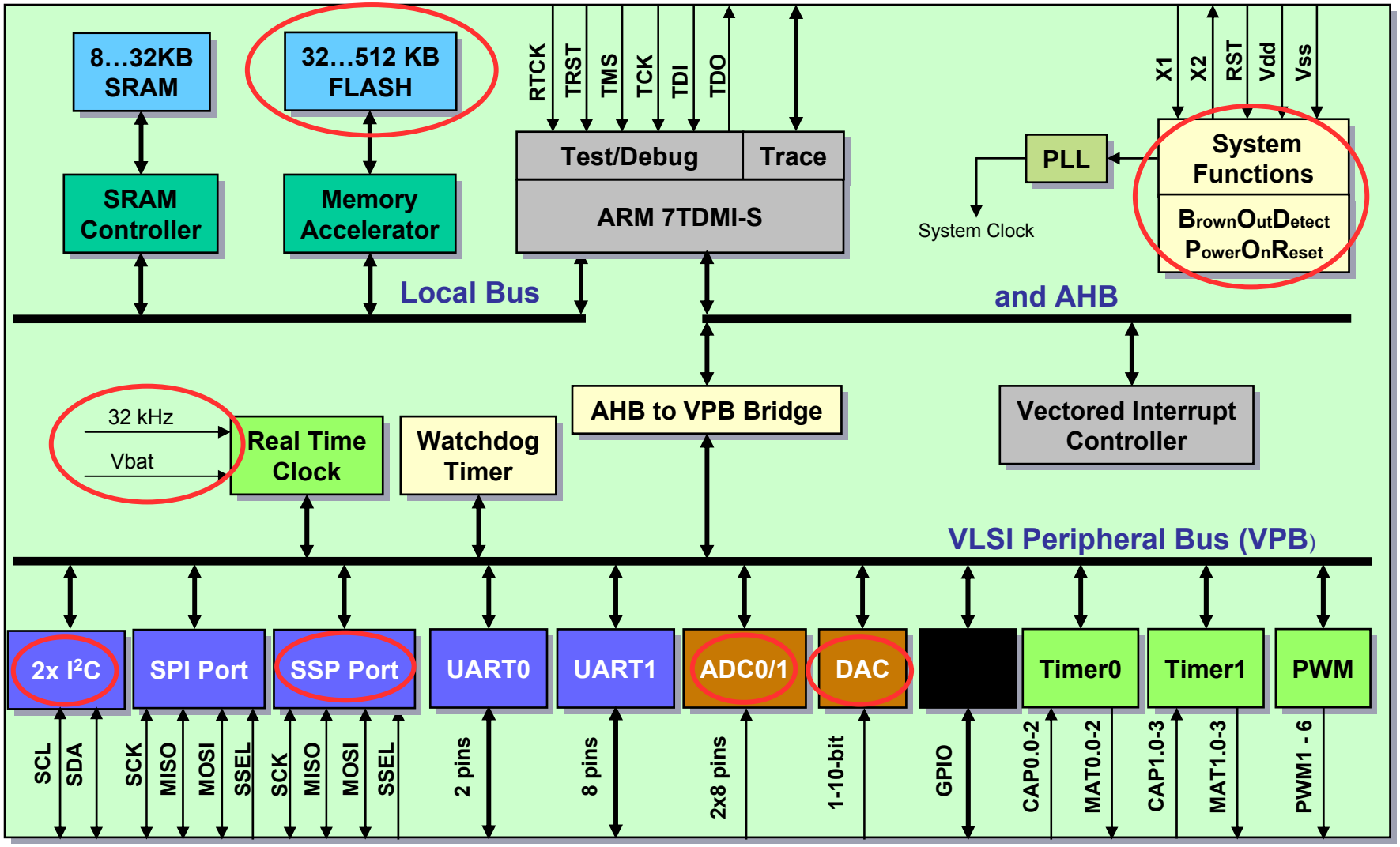


LPC2220 Flashless 72 MHz



Package: LQFP144

LPC213x Block Diagram



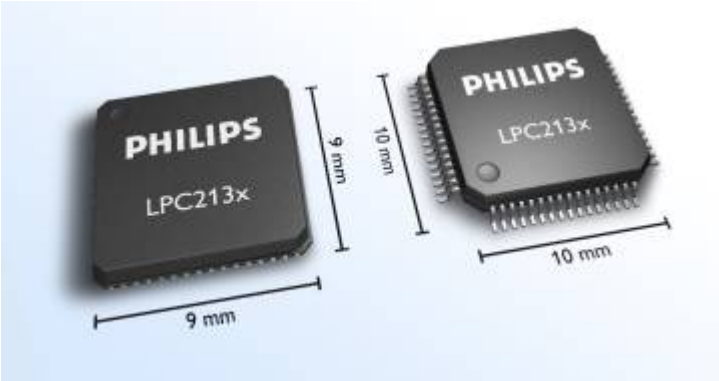
LPC2131, LPC2132 ONLY ONE

NOT LPC2131

Package: LQFP64/HVQFN64

LPC213x Series Overview

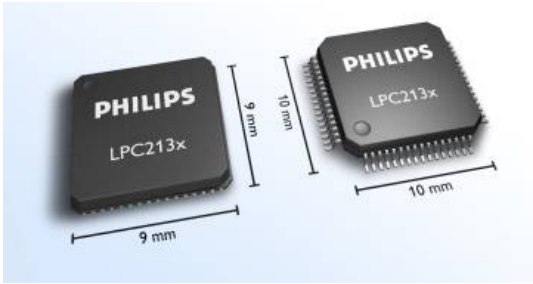
- 60 MHz Operation (54MIPS) from both on-chip Flash and SRAM
- 2 I^2C, 2 UARTs, 1 SPI, 1 SPI/ SSP
- Two 8-channel 10-bit ADCs
- One 10-bit DAC
- 4 Timers (Capture/Match/PWM/WDT)
- 47 I/O pins (5V tolerant)
- 3.3V Single-Voltage Supply
- 32KHz RTC, BOD, POR
- User-code security
- Real-time Debugging & Trace
- ISP, IAP, Parallel Programmer Support
- Tiny Packages: QFP64 (10 x 10 x 1.4 mm), HVQFN64 (9 x 9 x 0.85 mm)



	Flash	SRAM	ADC	DAC	Pkg
LPC2131	32KB	8KB	1		QFP64
LPC2132	64KB	16KB	1	1	QFP64 QFN64
LPC2134*	128KB	16KB	2	1	QFP64
LPC2136*	256KB	32KB	2	1	QFP64
LPC2138	512KB	32KB	2	1	QFP64 QFN64

* Available Q1 2005

LPC213x Series Overview



Single Supply Voltage

- 3.3V Single-Voltage Supply
- CPU operating voltage range of 3.0V to 3.6V (3.3V +/- 10%) with 5 Volt tolerant I/O pads.

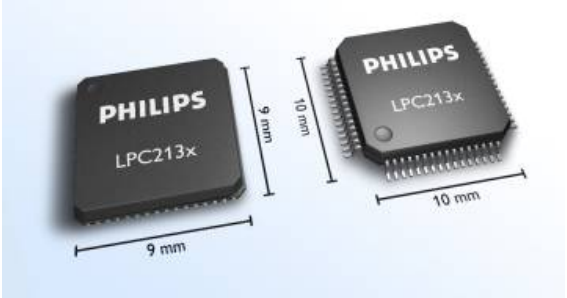
Brown Out Detection (BOD @ 2-stage monitoring of the voltage)

- **Stage 1:** Vdd < 2.9V, the Brown-Out Detector (BOD) asserts an interrupt signal to the VIC (Vectored Interrupt Controller).
- **Stage 2:** Vdd < 2.6 V LPC213x will be reset to prevent alteration of the Flash as operation of the various elements of the chip would otherwise become unreliable due to low voltage.

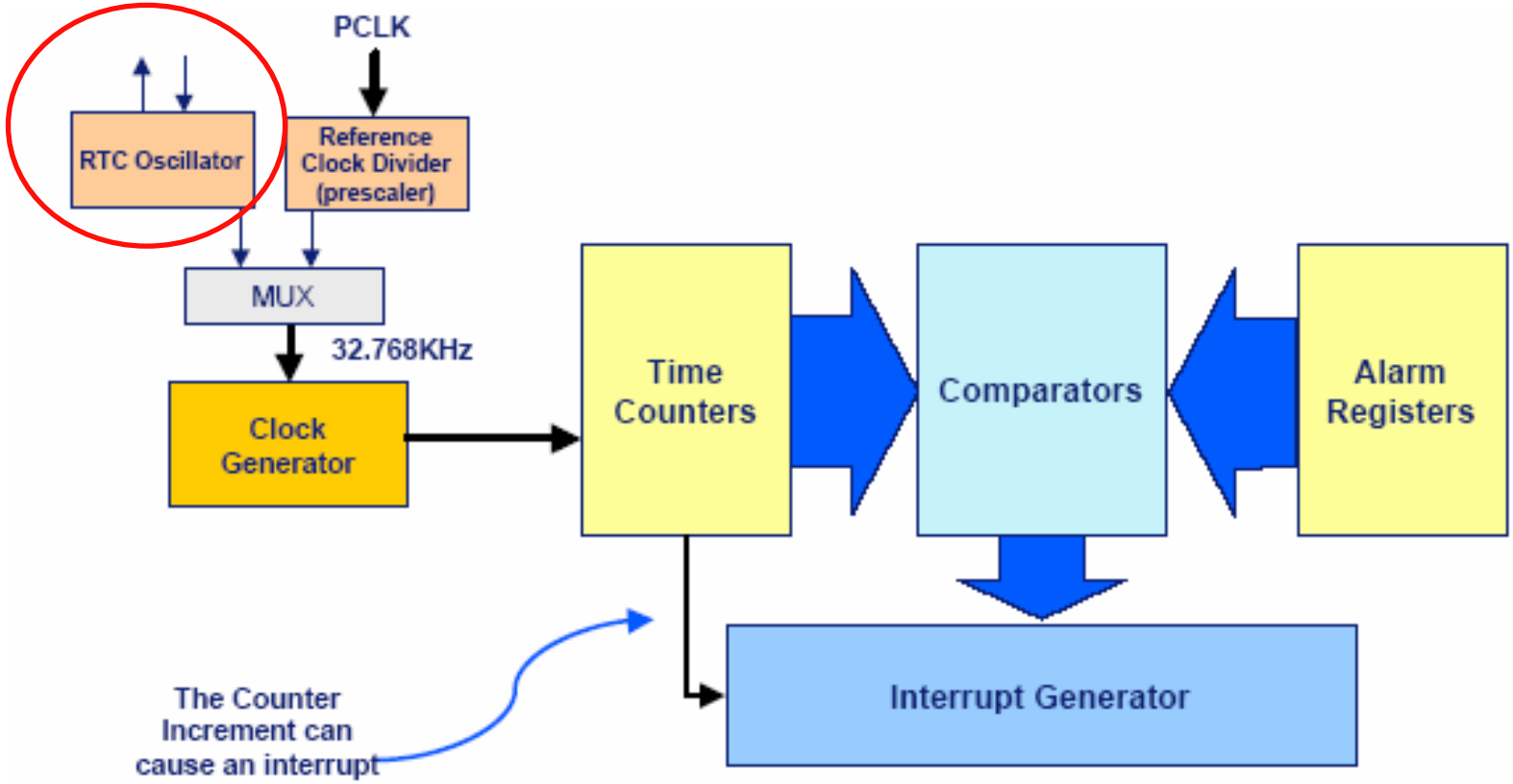
Power On Reset (POR)

- The BOD circuit maintains this reset down below 1V, at which point the Power-On Reset circuitry maintains the overall Reset.

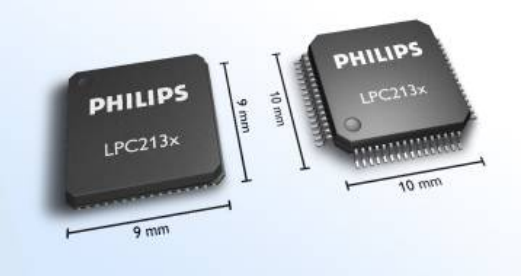
LPC213x Series Overview



RTC with additional crystal pins



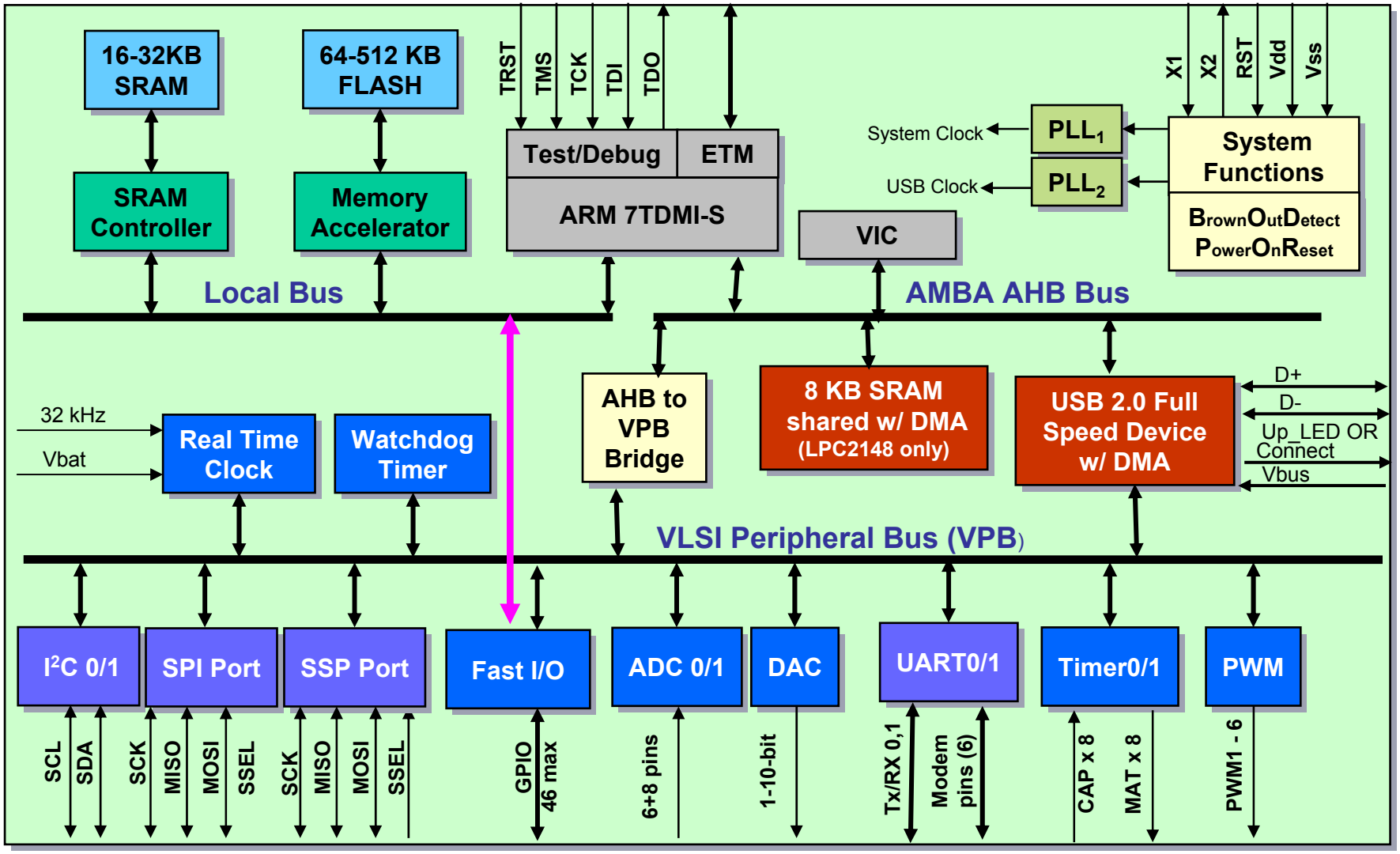
LPC213x Series Overview



RTC:

- Can be clocked by a separate 32.768KHz or by prescaler divider based on VPB clock
- So RTC can run in Power Down mode
- Has got its own supply pin Vbat which can be connected to battery or to the (2.0… 3.3… 3.6 V) supply.
- Typical power consumption is 14-20uA (@25 degree, with Vbat 2.5 to 3.6V respectively) when the device is in Power Down Mode

LPC2142/44/46/48 Block Diagram



Has 1.8V Regulator. Only 3V input needed

64-pin LQFP

LPC214x Series Overview

- **60 MHz Operation from both on-chip Flash and SRAM**
- **2 I^2C, 2 UARTs, 1 SPI, 1 SPI/ SSP**
- **Up to 14-channels 10-bit ADCs**
- **One 10-bit DAC**
- **4 Timers (Capture/Match/PWM/WDT)**
- **45 I/O pins (5V tolerant)**
  - **3.5 times faster than older I/O!**
- **3.3V Single-Voltage Supply**
- **32KHz RTC with Vbat input**
- **Brown Out Detect, Power On Reset**
- **User-code security**
- **Real-time Debugging & Trace**
- **ISP, IAP, Parallel Programmer Support**
- **Tiny Packages: LQFP64 (10 x 10 x 1.4 mm), HVQFN64 (9 x 9 x 0.85 mm)**

Spec	LPC2142	LPC2148
Internal Flash	64 KB	512 KB
Internal SRAM	16 KB	32 KB + 8 KB shared
10-bit ADC	1 x 6-chan	1 x 8-chan 1 x 6-chan
UARTs	2 x 16C550	2 x 16C550 (one with auto CTS/RTS plus fractional baud rate divisor)

LPC214x - USB Features

- USB 2.0 Full Speed Device
- Supports 32 physical (16 logical) endpoints
  - Supports Control, Interrupt, Bulk and Isochronous endpoints
- 2kB of endpoint RAM for communication only (not general purpose)
- 8kB block of general purpose SRAM usable by USB DMA (LPC2148 only)
- USB controller has dedicated PLL (functionally same as other PLL)
- USB registers are accessed via the VPB bus, but the 8kB block is accessible via the AHB bus
- Customer can choose between the UP_LED (Good Link™) OR the CONNECT (Soft Connect™) functionality

Extending the success to LPC214x

LPC213x Features:

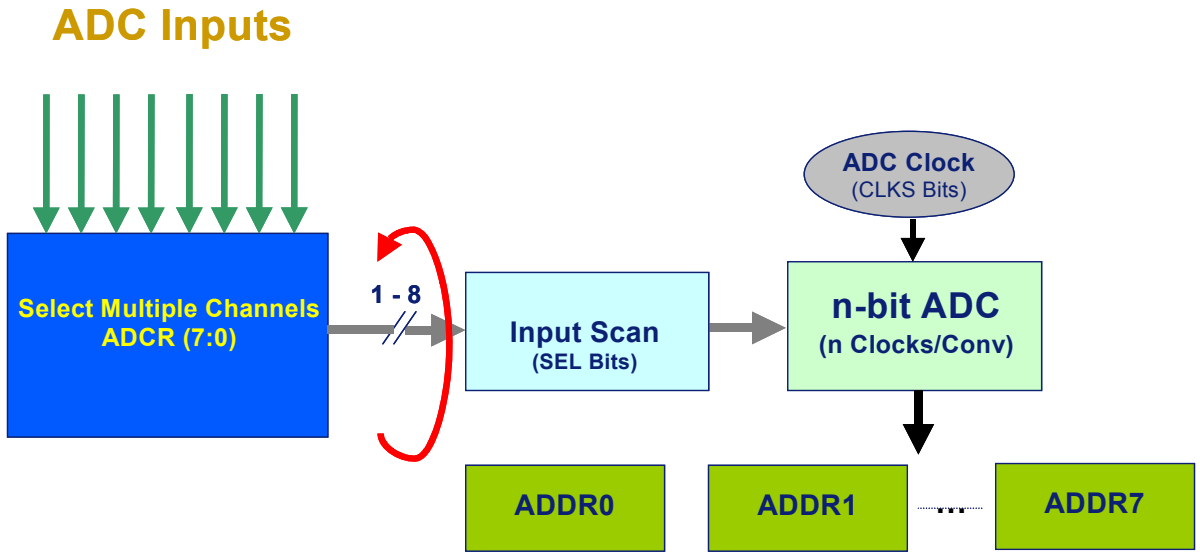
- Fast Embedded Flash: Up to 512K Bytes of nearly 60 MHz zero wait state execution from 128-bitx2 wide Flash with Memory Acceleration
- Single-voltage supply: on-chip DC-DC converter takes a single 3.3V supply with POR and BOD capabilities
- Many standard peripherals: Real-time-clock with power domain, SPI, I2C, UARTS, Timers

LPC214x Adds:

- USB 2.0 Full-speed 12 Mbits/sec with full USB standard compliance and DMA
- Fast I/O Capability; speeds up Software controlled I/O by 3.5X, up to 15Mhz port-pin toggling frequency
- 2 10-bit ADCs and a 10-bit DAC with individual result registers
- Enhanced UART with hardware handshake plus fractional baud rate divider. -> crystal frequency can be set to a independent value of to the baud rate generator clock

2 10-bit Analog-to-Digital Converters with:

- 400 Kbits/Sec Sampling frequency with 8 Channels
- Each Channel has its own Result Register thus reducing CPU Interrupt Overhead by a factor of 8
- ADCs can Operate in Burst Mode with autonomous signal acquisition
- ADCs can be synchronized (e.g. for simultaneous current & voltage measurement) and triggered by an input pin or Timer match



NXP USB versus Competition

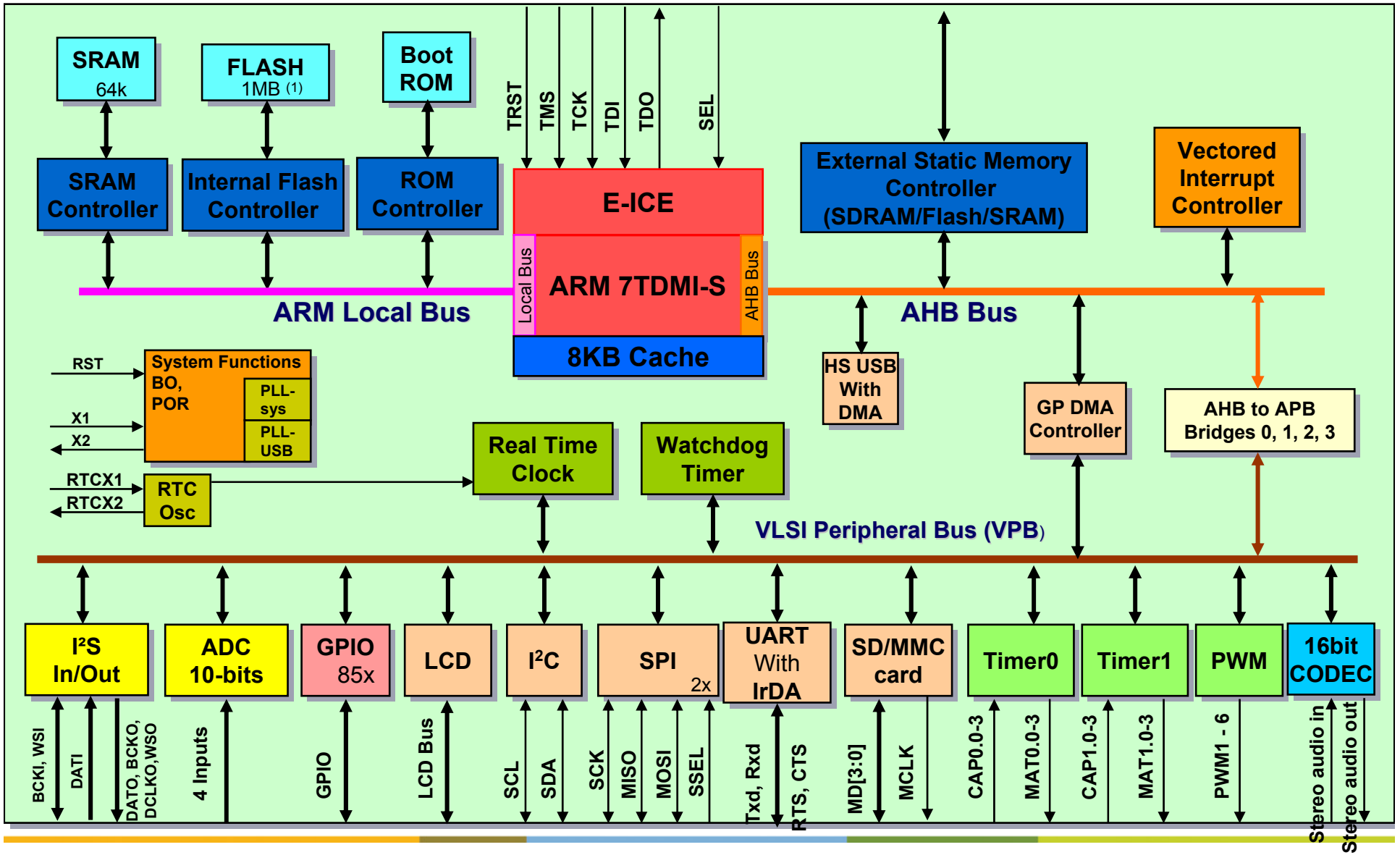
	USB Standard	Philips	Atmel SAM7S64	ST STR71X
Bi-directional* Endpoints supported	**16 max/device**	**16**	**4**	**8**
Modes Supported	Control, Interrupt,Bulk, Isochronous	Control, Interrupt,Bulk, Isochronous	Control, Interrupt,Bulk, Isochronous	Control, Interrupt,Bulk, Isochronous
Max. Control Buffer size.	64 bytes	64 bytes	64 bytes	64 bytes
Max. Interrupt Buffer Size	64 bytes	64 bytes	64 bytes	64 bytes
Max. Bulk Buffer Size	64 bytes	64 bytes	64 bytes	64 bytes
Max. Isoch. Buffer Size	**1023 bytes**	**1023 bytes**	**64 bytes**	**512 bytes**
DMA Capability		**Yes**	**No**	**No**

* Separate input/output

LPC2880/8 Family for
MP3 player and audio management best fit

LPC2880/88 Blocks

180 pins



(1) LPC2888 only

LPC2800 Family

Advanced and Flexible Memory Capability:

- On-chip: 1Mbyte of Flash, 64KB of RAM, and 32KB of ROM
- External Memory Controller for SDRAM, NOR and NAND Flash, and SRAM
- 8KBytes of Cache for enhanced performance from external memory

LPC2800 Peripherals:

- 480 Mbits/sec High-Speed USB device with on-chip PHY
- Multi-Media Card, I2S, and LCD controller interfaces
- General Purpose DMA
- ATA interface

LPC2800 Power Supply sub-systems:

On-chip High-efficiency Switching regulator and Linear Regulators allow:

- Operation from single AA(A) Battery cell (0.9 to 1.6V)
- Operation from 5V USB input

LPC2800 Key Features (1)

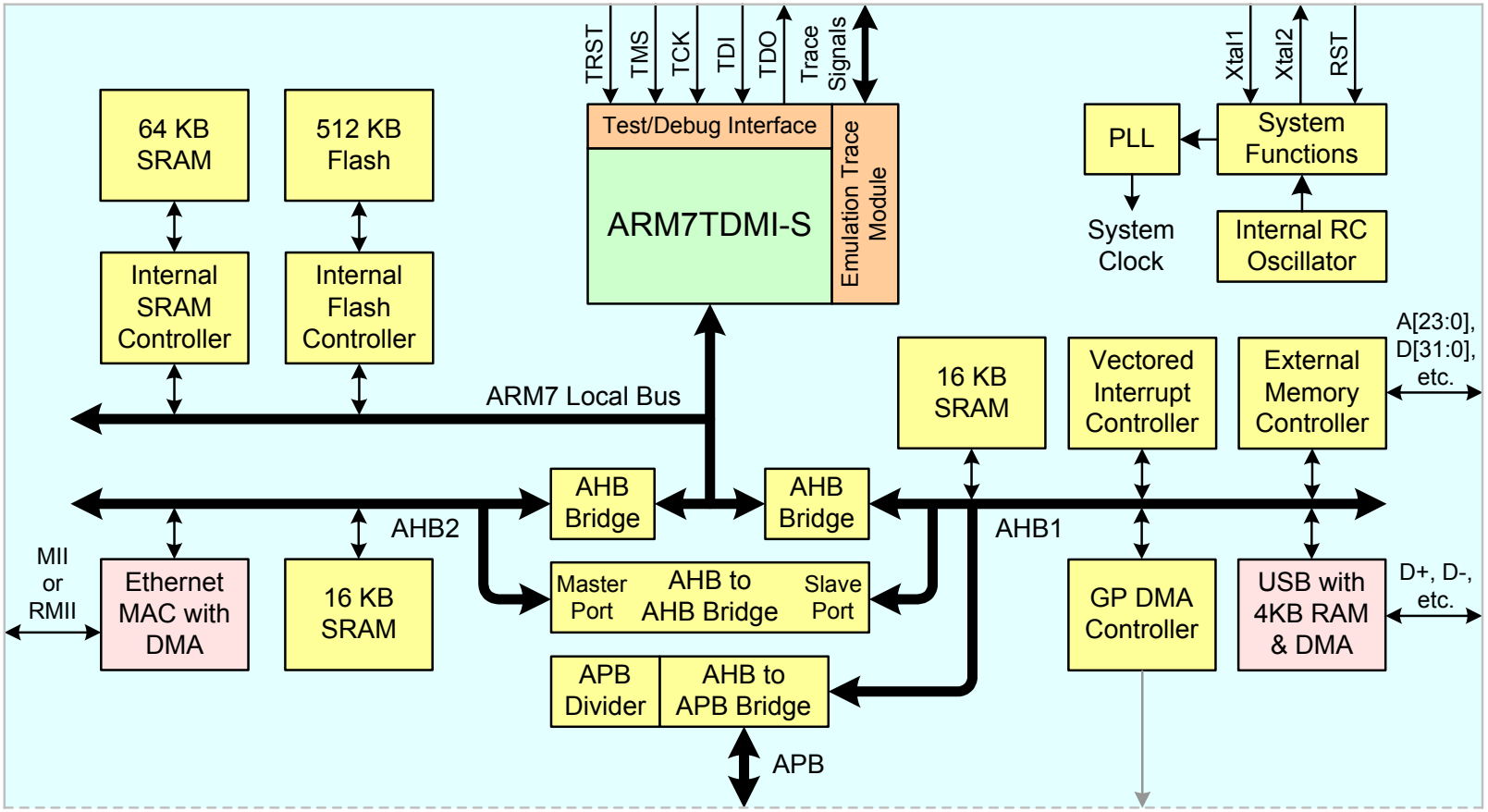
- On-chip 1Mbyte of flash on LPC2888 (0Kbyte on LPC2880), 64Kbyte RAM, 8 Kbytes cache
- Boot ROM allow execution of Flash Code, external code, or flash programming via USB
- External Memory Controller for SDRAM, NOR and NAND flash and SRAM.
- 8Kbyte of cache for enhanced performance from external memory.
- 480 Mbits/sec High-Speed USB with on-chip PHY
- LCD Interface glue logic
- 8 channels General Purpose DMA, (can be connected to the LCD interface too)
- SD/MMC Card Interface
- 10Bit A/D converter + 16 Bit A/D and DA converters with amplification and gain control
- UART with fractional baud rate generator, IrDA IIC and IIS interfaces

… LPC2800 Key Features (2)

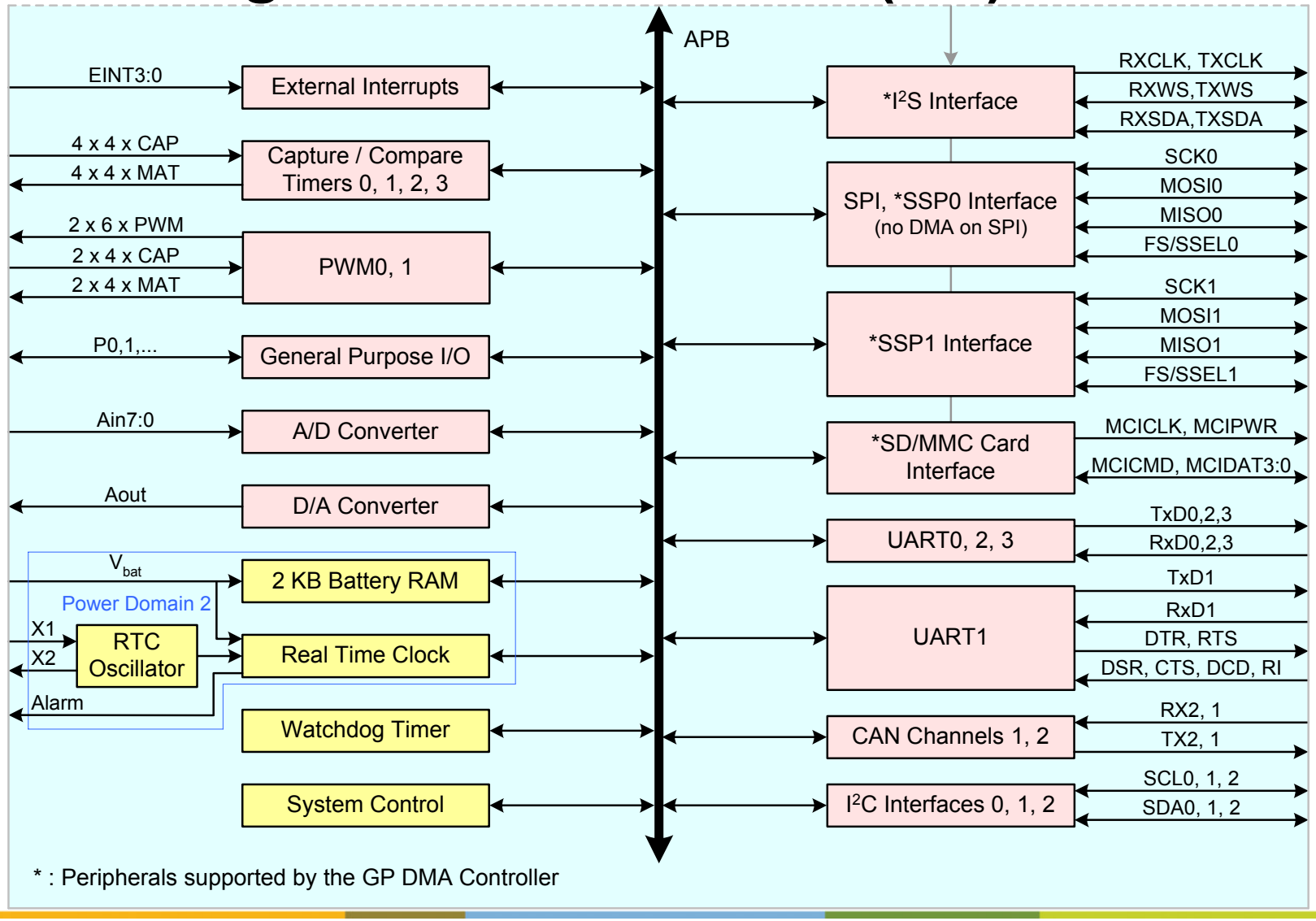
- Innovative Event Router allows interrupt, power-up- and clock start capabilities from up to 107 sources. Each signal can act as an interrupt source, or a clock enable or reset source for the LPC2880/88 modules
- Advanced clock generation: CPU clock can be obtain from the RCT 32Khz clock now
- Integrated DC/DC converter can generate all required voltages from a single battery or from USB power

LPC2300/24000 Families
32-bit ARM7 Products

Block Diagram LPC23xx/24xx (1/2)



Block Diagram LPC23xx/24xx (2/2)



Features: Core

- 72 MHz ARM7TDMI
- Advanced Vectored Interrupt Controller (32 vectors, 16 priorities)
- Single 3.3V power supply (3.0V to 3.6V).
  - On-chip DC/DC converter for internal 1.8V
- 4 reduced power modes, Idle, Sleep, Power Down, and Deep Power Down.
  - Processor wakeup from Power Down mode via any interrupt able to operate during Power Down mode (includes external & GPIO interrupts, RTC, Ethernet wakeup).
- On-chip Power On Reset, and Brownout detect with separate thresholds for interrupt and forced reset.
- On-chip crystal oscillator with an operating range of 1 MHz to 24 MHz.
- 4 MHz internal RC oscillator that is the default system clock.
- On-chip PLL allows CPU operation up to the maximum CPU rate without the need for a high frequency crystal. May be run from the main oscillator, the internal RC oscillator, or the RTC oscillator.

System Architecture Issues

Challenges to preventing bottlenecks:

- Ethernet requires support of 2 concurrent 100Mbits/sec data streams with up to 1500 byte packets
- USB & Ethernet Streams are asynchronous and must both be supported including Isochronous mode USB (1024 byte data bursts)
- CAN, SPI, SSP, I^2C, SDIO, I^2S, UARTs, Timers, PWMs, ADC, DAC, etc, must also be supported but these are all lower bandwidth with smaller packet sizes than Ethernet & USB (increases MCU core involvement)
- CPU, Ethernet, and USB clock domains (72, 25/50, and 48 MHz) are all separate and need to communicate through memory
- Multi-ported memory is expensive, complex, and not desirable

The Goals of the LPC2300 Family

Preventing overflow, underflow, and contention:

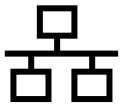
- Manage multiple Asynchronous high-speed channels with no channel performance bottlenecks
- Provide Ethernet, USB, CAN, UART, SPI, I^2S, and I^2C channels
- Offer a one chip system for high performance and low power at a low cost
- Include industry-leading Embedded Flash operating at SRAM speeds combined with ECC (error correction) for the best performance and power, security and reliability
- Develop a large derivative family in different packages and with different capabilities and memory configurations at different price points, but allowing for easy transitions between family members (preserve code re-use)

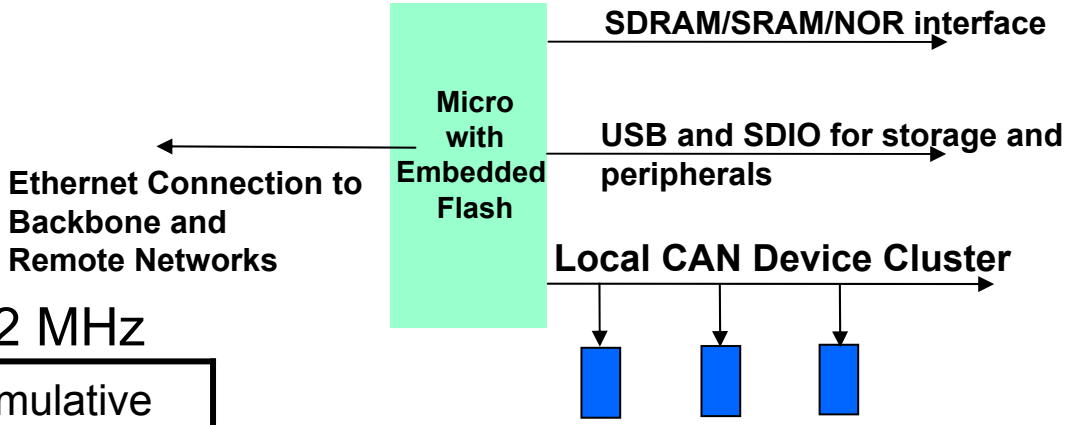
Building the LPC2300 – CPU & Memory

- ARM7TDMI-S Processor – 72 MHz
- Up to 512 KB Flash on-chip Flash
  - zero wait-state (execute code from Flash or SRAM)
  - 128-bit wide bus with patented Memory Accelerator Module (MAM)
  - 8-bits Error Correction Code (ECC) for every 128-bit word
  - Automotive qualified Flash process for high reliability
- Up to 58 KB on-chip Static RAM (all portions accessible by CPU and DMA)
  - 8 KB – 32 KB SRAM exclusively for CPU
  - 16 KB for Ethernet buffering
  - 8 KB for USB device (code or data)
  - 2 KB for RTC is for data only
  - Additional 4 KB USB FIFO buffer
- Advanced Vectored Interrupt Controller (VIC) – 32 IRQ sources
- Emulation Trace Module supports real-time trace
- Low power - 4 reduced power modes including Deep Power Down

Start with the best embedded Flash in the market

Bus Bandwidth required for Industrial network





Bus bandwidth usage at 72 MHz

Peripheral	AHB Bus bandwidth %	Cumulative AHB Bandwidth
Ethernet	50	50
USB	4	54
Ext, DRAM	20	74
SSP	4	79
$I^2C(2)$	2	81
CAN (2)	3	84
UARTS(2)	4	88
ADC	8	96

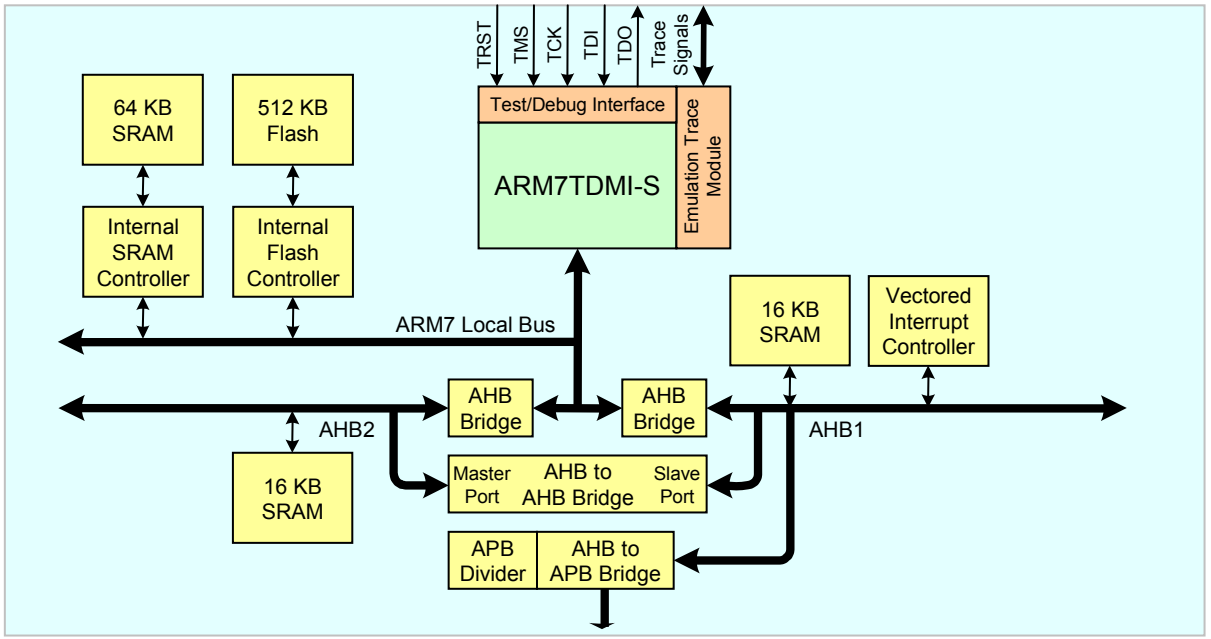
- More than 60% usage of bus bandwidth causes collisions
- Loaded system bandwidth at 72Mhz is 96% for application

Conclusion:

1 AHB Bus is not sufficient
Multiple Busses and concurrent DMA processing is required

Building the LPC2300 – Dual AHB Bus

- Two separate but not isolated AHBs
  - Any bus can still reach any other bus through bridges when needed
- High-bandwidth peripherals on different AHBs will not overwhelm the CPU or other peripherals



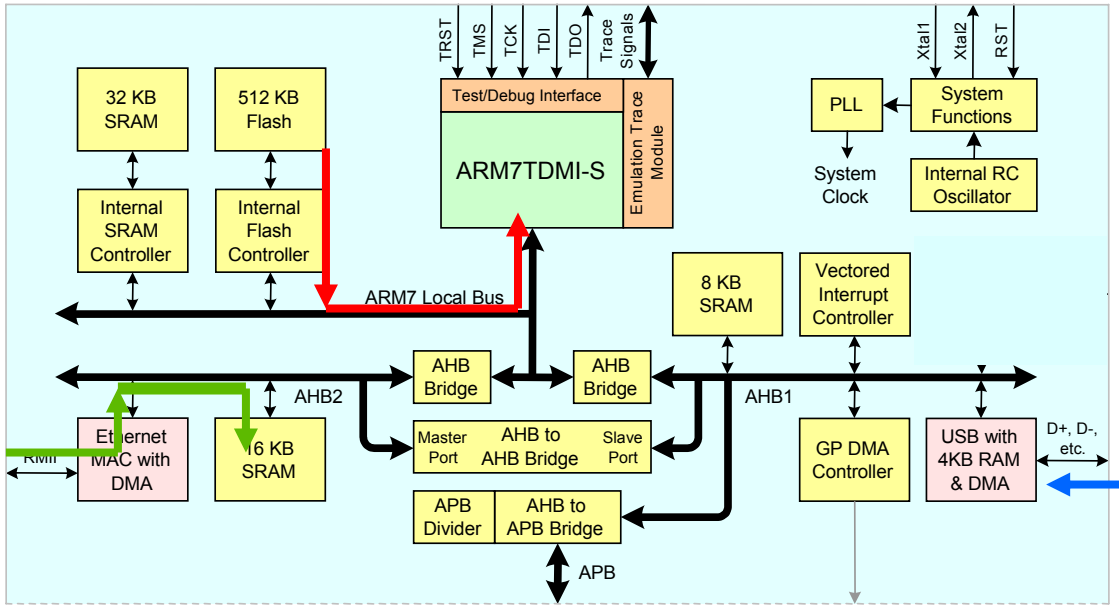
No communications “traffic jams”!!

The LPC2300 Advantage - parallel buses

Concurrent operations become possible:

- Ethernet packet reception and transfer to SRAM
- CPU Instruction Fetch
- USB packet reception and transfer to SRAM

Dedicating AHB Bus to Ethernet is required to guarantee 100 Mbits/sec Ethernet throughput without contention with other peripherals



Ethernet

USB

Features: I/O Peripherals

- 160 GPIO pins, all implemented as fast GPIOs, with 64 GPIO interrupts (plus 4 other external interrupts).
- 10 bit A/D Converter with input multiplexing among 8 pins.
- 10 bit D/A converter.
- Four general purpose Timers with capture inputs, compare outputs, and external count inputs.
- Two linkable PWM / Timer blocks with support for 3 phase motor control with "dead time". Each PWM has an external count input.
- Real Time Clock with separate power pin, alarm output, and 2K SRAM.
- Watchdog Timer. The watchdog timer can be clocked from the internal RC oscillator, the RTC oscillator, or the APB clock.

Features: Serial Interfaces

- Ethernet MAC with associated DMA controller.
  These functions reside on an independent AHB bus.
- USB Device, Host (OHCI compliant), and OTG block with on-chip Host/Device PHY and associated DMA controller.
- Four UARTs with fractional baud rate generation, one with modem control I/O, one with IrDA support, all with FIFO.
- CAN controller with two channels.
- SPI controller.
- Two SSP controllers, with FIFO and multi-protocol capabilities. One is an alternate for the SPI port, sharing its interrupt and pins.
- Three I2C Interfaces. The second and third I2C interfaces are expansion I2Cs with standard port pins rather than special open drain I2C pins.
- I2S (Inter-IC Sound) interface for digital audio input or output.

Power Options

- Power options:
  - On-chip DC-DC converter supplies 1.8V power to all internal logic, except in the RTC power domain.
  - 1.8V power can be supplied from off-chip for some pinouts.

- Power reduction modes:
  - Idle mode.
  - Power Down mode.
  - Sleep mode.
  - Deep Power Down mode.

- Power reduction modes are entered via an encoding of the IDL and PD bits in PCON, plus an additional new power control bit.

Idle Mode

- CPU is halted, reducing power used by:
  - The CPU itself.
  - Memories and their controllers used by the CPU.
  - Internal buses used by the CPU.

- Peripheral clocks and functions continue to run.
- All registers and memories retain their state.
- Wakeup from any enabled interrupt, or Reset.

Sleep Mode

- All clocks are stopped to the CPU and peripherals.
  If enabled, the main oscillator and PLL are shut down.
  - All dynamic operation of the device is suspended.
  - The DC-DC converter remains operational.
  - The Flash memory remains on, allowing for fast wakeup.

- All registers and memories retain their state.
- Wakeup from any enabled interrupt that can occur without clocks, or Reset.

Power Down Mode

- All clocks are stopped to the CPU and peripherals.
  If enabled, the main oscillator and PLL are shut down.
  - All dynamic operation of the device is suspended.
  - The DC-DC converter remains operational.
  - The Flash memory is turned off.

- All registers and memories retain their state.
- Wakeup from any enabled wakeup source that can occur without clocks, or Reset.

Deep Power Down Mode

- Similar to Power Down mode, but the DC-DC converter is turned off.
  - This is only useful if 1.8V power is supplied externally.
  - Device state is lost (but, see RTC and Battery RAM).
- Wakeup can only be accomplished by a chip reset or an Alarm interrupt from the RTC.
- During Deep PD mode, power may be removed from the entire device, except for the RTC.
  - Restoring power causes a POR.
- Wakeup requires that external circuitry restores power.
  - The alarm output of the RTC can signal external circuitry when power should be restored, or some external means may be used.

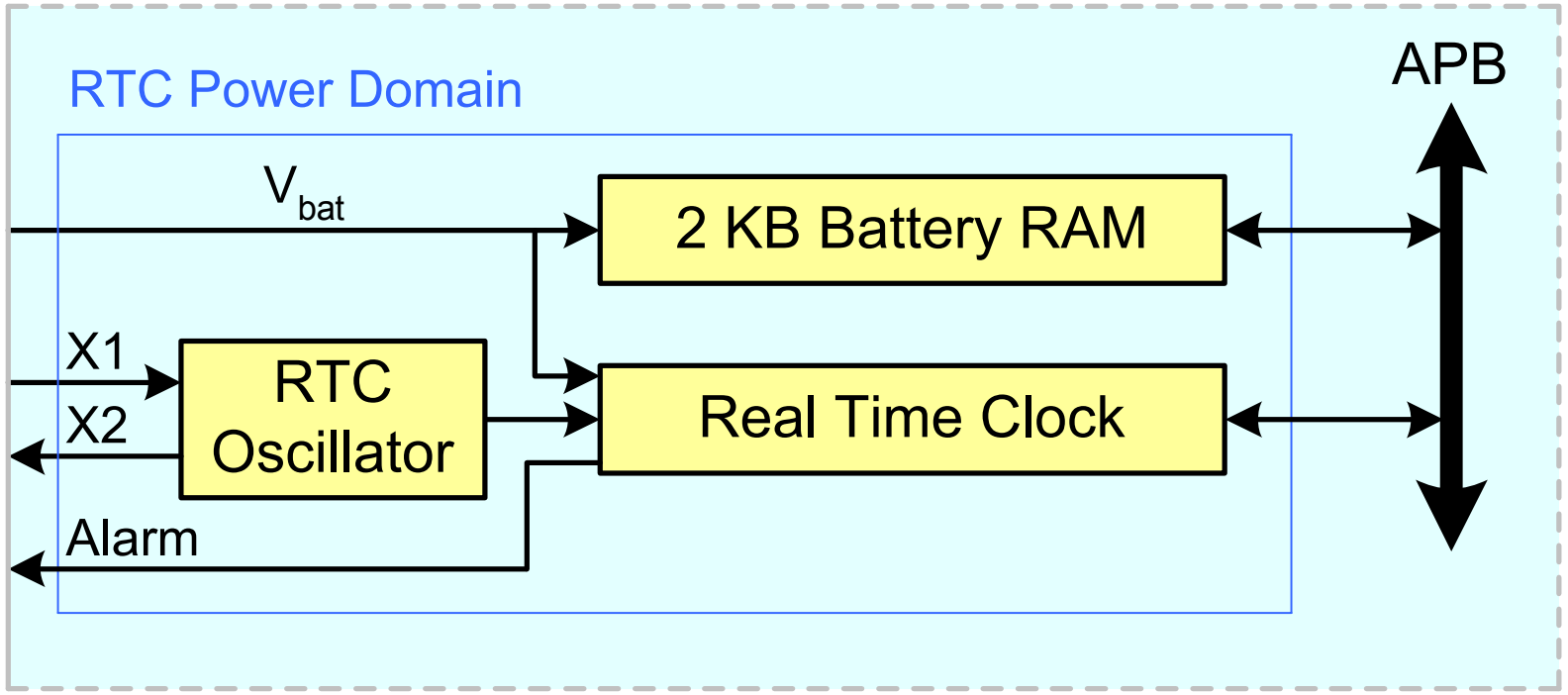
Wakeup from Power Down or Sleep Mode

- External interrupts
  - (EINT0 through EINT3) and GPIO interrupts
- Ethernet wakeup
  - (portions of the Ethernet block receive clocks from the external PHY)
- USB or CAN activity (pin state change)
- Brown Out Detect
- RTC Alarm

Wakeup from Power Down or Sleep Mode

- External interrupts
  - (EINT0 through EINT3) and GPIO interrupts
- Ethernet wakeup
  - (portions of the Ethernet block receive clocks from the external PHY)
- USB or CAN activity (pin state change)
- Brown Out Detect
- RTC Alarm

RTC & Battery RAM



I_{BATact}	active mode battery supply current	DC-to-DC converter on	[9]	-	20	-	µA
		DC-to-DC converter off	[9]	-	28	-	µA

NicheLite for LPC by Interniche

- NicheLite for LPC is a fully featured TCP/IP stack
  - Requires as little as 12 KB of code.
- Support for the following protocols:
  - Address Resolution Protocol (ARP), Internet Protocol (IP), Internet Control Message Protocol (ICMP), User Datagram Protocol (UDP), Transmission Control Protocol (TCP), Dynamic Host Configuration Protocol (DHCP) Client, Domain Name System (DNS) Client, Bootstrap Protocol (BOOTP), Trivial File Transfer Protocol (TFTP)
- Includes NicheTask ™ a cooperative multi-tasking scheduler.
- Supports InterNiche's Light Weight API and a Zero-Copy option.
- Single Ethernet interface with device drivers optimized for the LPC2300 and LPC2400
- Example applications (TFTP Client, TFTP Server, HTTP Listener)
- **Source code is free to NXP customers**
- **License - Unlimited use with NXP LPC2000 and LPC3000 microcontrollers only**
- Support from Interniche at sales@interniche.com

LPC24xx

Comparison Table

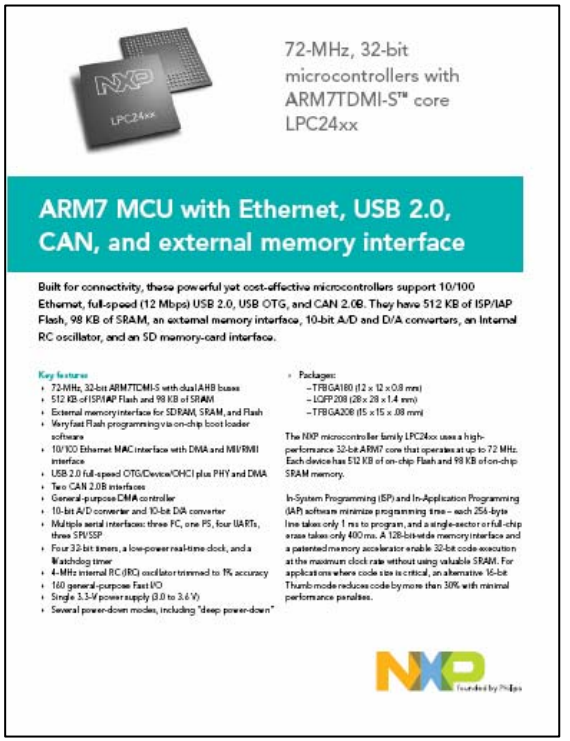
Product	Flash	SRAM	External Bus	10/100 Ethernet	USB 2.0	CAN	SD/MMC	Package(s)
LPC2458	512kB	98kB	Full 16-Bit	1	1	2	1	TFBGA180
LPC2468	512kB	98kB	Full 32-Bit	1	1	2	1	LQFP208 TFBGA208

LPC24xx features beyond those offered in the LPC23xx.

- More RAM
- Ethernet MII interface (in addition to RMII)
- USB Host and OTG functionality
- External Memory Interface with additional External Memory Controller circuitry
- Additional GPIO
- Additional PWM

LPC24xx

- Announcement on the web
- Datasheets by email (LogicPG@NXP.com)
- 2 Devices:
  - LPC2458: 512KB flash, 98KB SRAM, Ext bus (16-bit), TFBGA180
  - LPC2468: 512KB flash, 98KB SRAM, Ext bus (32-bit), LQFP208 & TFBGA208
- Will be available in Q1 2007
- LPC24xx Leaflet


72-MHz, 32-bit microcontrollers with ARM7TDMI-S™ core LPC24xx

ARM7 MCU with Ethernet, USB 2.0, CAN, and external memory interface

Built for connectivity, these powerful yet cost-effective microcontrollers support 10/100 Ethernet, full-speed (12 Mbps) USB 2.0, USB OTG, and CAN 2.0B. They have 512 KB of ISP/IAP Flash, 98 KB of SRAM, an external memory interface, 10-bit A/D and D/A converters, an Internal RC oscillator, and an SD memory-card interface.

Key features
- 72-MHz, 32-bit ARM7TDMI-S with dual AHB buses
- 512 KB of ISP/IAP Flash and 98 KB of SRAM
- External memory interface for SDRAM, SRAM, and Flash
- Very fast Flash programming via on-chip boot loader software
- 10/100 Ethernet MAC interface with DMA and MII/RMII interface
- USB 2.0 full-speed OTG/Device/OHCI plus PHY and DMA
- Two CAN 2.0B interfaces
- General-purpose DMA controller
- 10-bit A/D converter and 10-bit D/A converter
- Multiple serial interfaces: three I²C, one I²S, four UARTs, three SPI/SSP
- Four 32-bit timers, a low-power real-time clock, and a Watchdog timer
- 4-MHz internal RC (IRC) oscillator trimmed to 1% accuracy
- 160 general-purpose Fast I/O
- Single 3.3-V power supply (3.0 to 3.6 V)
- Several power-down modes, including "deep power-down"
- Packages:
  - TFBGA180 (12 x 12 x 0.8 mm)
  - LQFP208 (28 x 28 x 1.4 mm)
  - TFBGA208 (15 x 15 x .88 mm)

The NXP microcontroller family LPC24xx uses a high-performance 32-bit ARM7 core that operates at up to 72 MHz. Each device has 512 KB of on-chip Flash and 98 KB of on-chip SRAM memory.

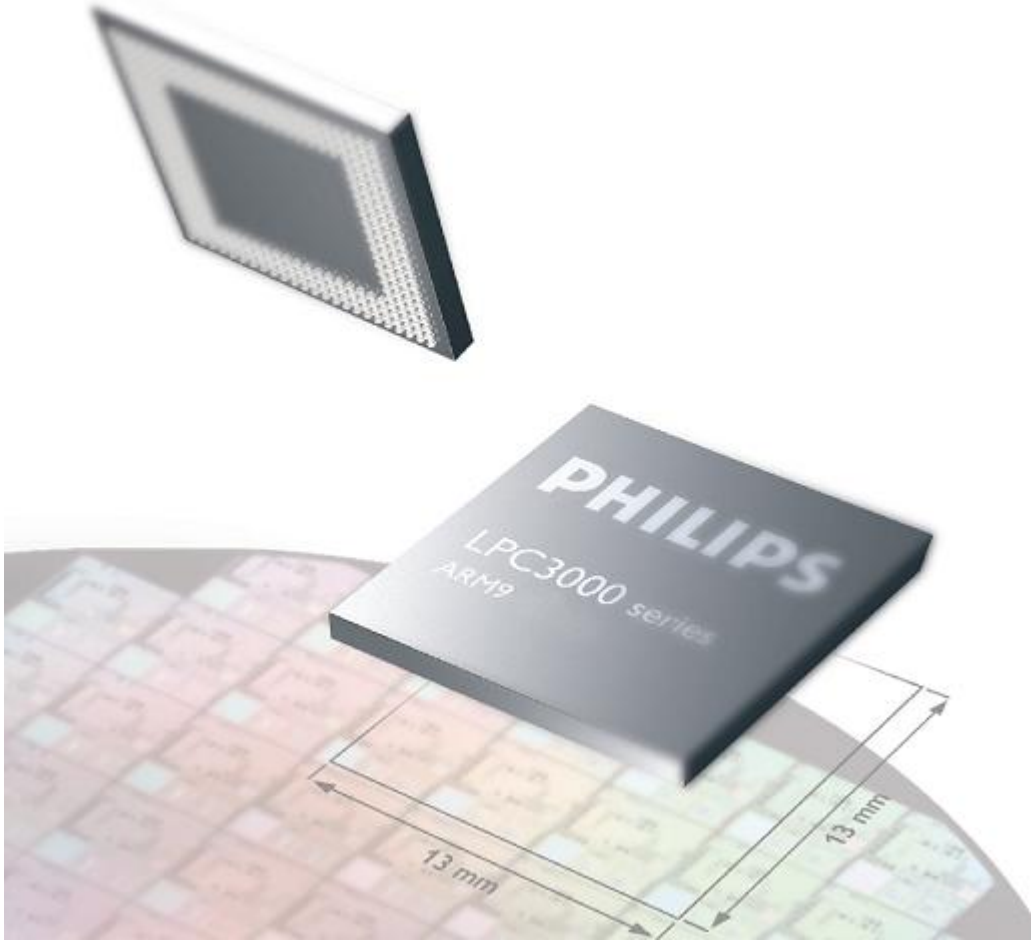
In-System Programming (ISP) and In-Application Programming (IAP) software minimize programming time – each 256-byte line takes only 1 ms to program, and a single-sector or full-chip erase takes only 400 ms. A 128-bit-wide memory interface and a patented memory accelerator enable 32-bit code execution at the maximum clock rate without using valuable SRAM. For applications where code size is critical, an alternative 16-bit Thumb mode reduces code by more than 30% with minimal performance penalties.

LPC3000 Family
32-bit ARM9 Products

NXP Standard Microcontroller

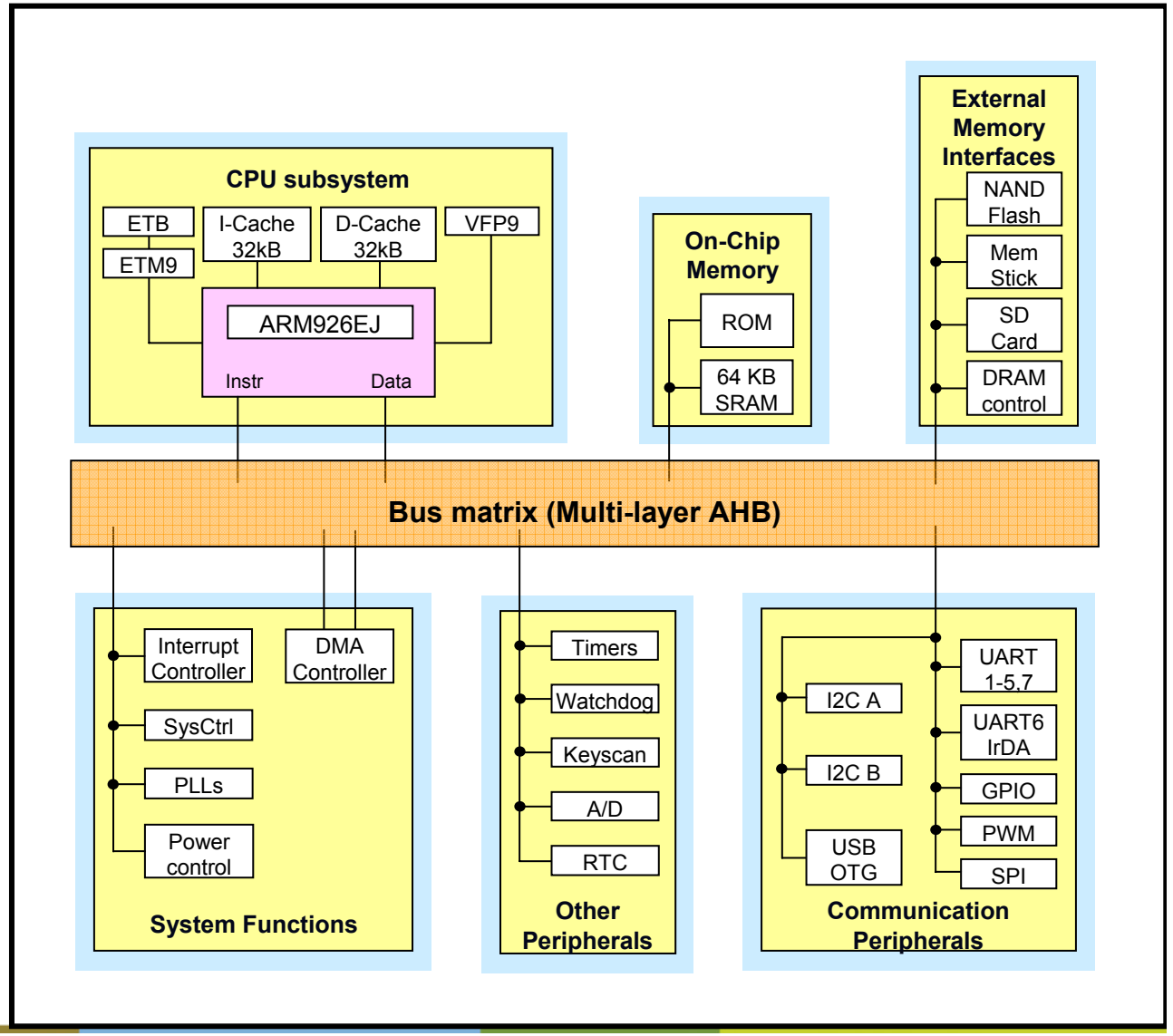
The Next Generation …..

LPC3000



Block Diagram LPC3180

- 90nm Process
- 208 MHz Core freq.
- 1.2V core operation
- 3.3V I/O
- 32KB I- & D- caches
- 64KB TCM SRAM
  (Tightly-Coupled Memory)
- Up to 1MB SRAM
  (On-Chip Memory)
- Standard E-ICE
  JTAG Interface
- 6KB ETB
  (Embedded Trace Buffer)
- 256-pin TFBGA package



Technology Announcement

First ARM9 family-based microcontrollers in 90nm technology

Press release done on Feb 23th during the Embedded World Show Nuremberg:

- LPC3000 Family based on 90nm technology industry's first 90nm ARM9* family-based 32-bit microcontroller family.
- Based on NXP Nexperia platform using the ARM926EJ-S* core

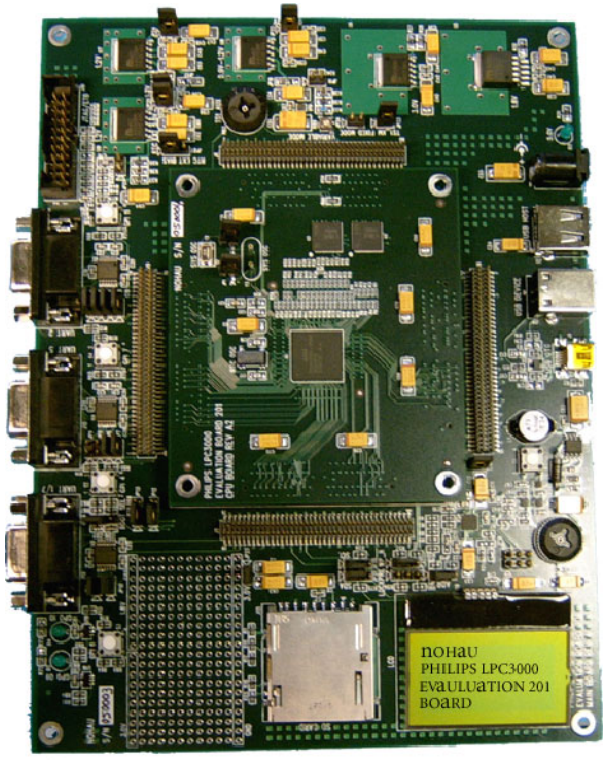
Features:

- Several power management benefits
- Peripherals such as integrated USB On-the-Go (OTG) and full USB Open Host Controller Interface (OHCI) host
- Multi-level NAND Flash interface
- Operating speed at 200MHz
- Standard communication peripherals like up to 7 UARTs, SPI, I^2C, USB, real-time clock, Ethernet - to follow.
- Floating point Vector Coprocessor
- 100mW power consumption @200Mhz with all peripherals switched on

Nohau's LPC3000 Evaluation Board

- A Single power supply input (5.0V), regulated on board to provide all the necessary EVB voltages.
- User Reset pushbutton switch.
- 20 Way JTAG/ETB connector.
- 32M (8M x 32) Bytes of SDRAM.
- 32M (32M x 8) Bytes of NAND FLASH.
- 1 - LCD Module with Philips PCF8558 built in.
- 1 - SD Card connector.
- 3 - USB connectors (USB A Receptacle Connector for USB Host; USB B Receptacle Connector for USB Device; USB Mini AB Receptacle Connector for USB OTG) with Philips ISP1301.
- 3 - UART (RS232) physical interface circuits connected to standard PC style DB9 female connectors.
- 4 - User input pushbutton switches.
- 2 - User output LEDs.

http://www.nohau.com/emularm/lpc3000_board.html



phyCORE®-ARM9/LPC3180
ARM9 with Vector Floating Point Unit

- **Technical Features**
- Carrier Board in EURO-card dimensions (100 x 160 mm)
- RS-232: 2x female DB-9 connectors support UART5 and UART2
- USB (Host/Device/OTG) connectors:
- Standard A type connector for USB host functionality, Standard B type connector for USB device functionality
- miniAB type connector for OTG functionality
- JTAG – 2.54mm pitch, 16-pin connector for JTAG debugging interface
- SD Card slot
- Reset push button, Boot jumper ,2x user buttons
- 4x user LEDs with jumpers to separate from I/O lines
- Battery receptacle for LPC3180 Real-Time Clock and SRAM back-up
- Keyboard – 2x8 2.54mm pitch connector for keyboard interface
- 1x potentiometer connected to one A/D input
- 5V. low-voltage socket for power supply connectivity
- 3V. and 5V. low voltage supplies for external devices and subassemblies
- Expansion Bus: address, data, interface and all applicable I/O signals route from implemented phyCORE module to 2x80-pin Molex connectors, enabling connectivity to Add-On hardware

- Memory configuration:
  - SDRAM: 16 to 64 MB synchronous SDRAM, max. access time of 10ns, 32-bit organization
  - Flash: 16 to 128 MB NAND-Flash in 8-bit mode
  - Serial: 1 to 32 KB I²C-EEPROM

- PART #: PCM-031
  - Configuration:
    208 MHz, 32 MB SDRAM,32 MB Flash, 32 KB EEPROM, USB OTG, JTAG interface
  - **$249.00 Module only**
  - **$349.00 basic package : Module + Carrier Board**
- http://www.phytec.com/products/rdk/ARM-XScale/phyCORE-ARM9LPC3180-Kits.html

Single Board Computer **Module**

PART #: PCM-031

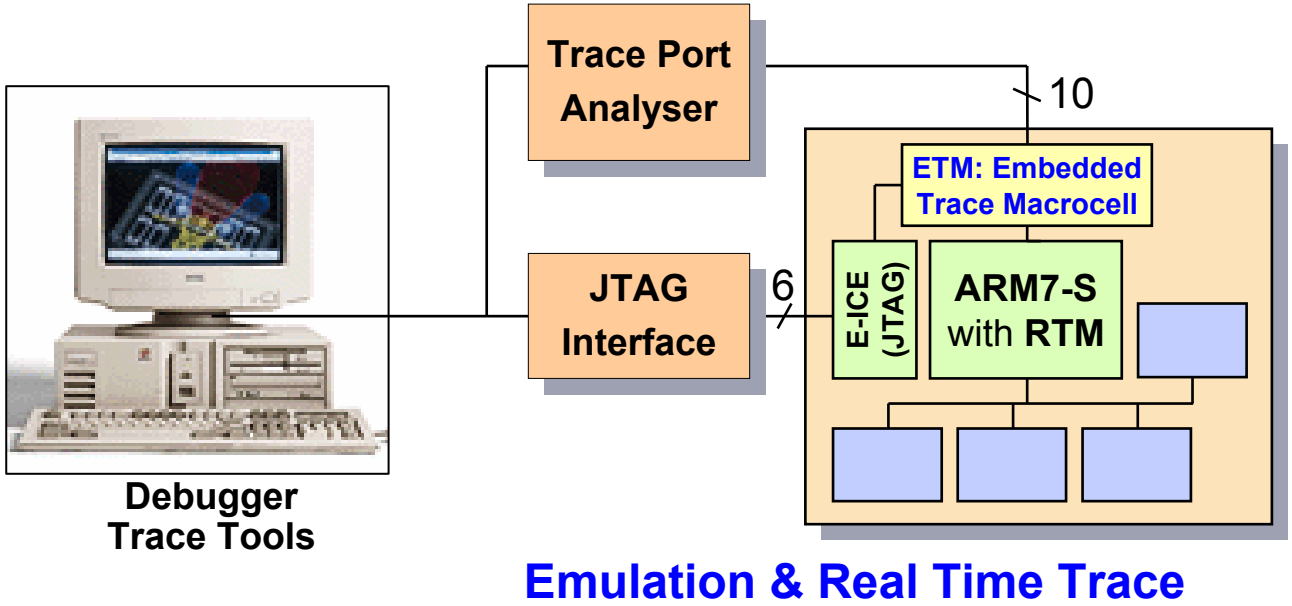
Carrier Board

PART #: KPCM-976

Tools and support 16/32-bit ARM7TDMI-S Products

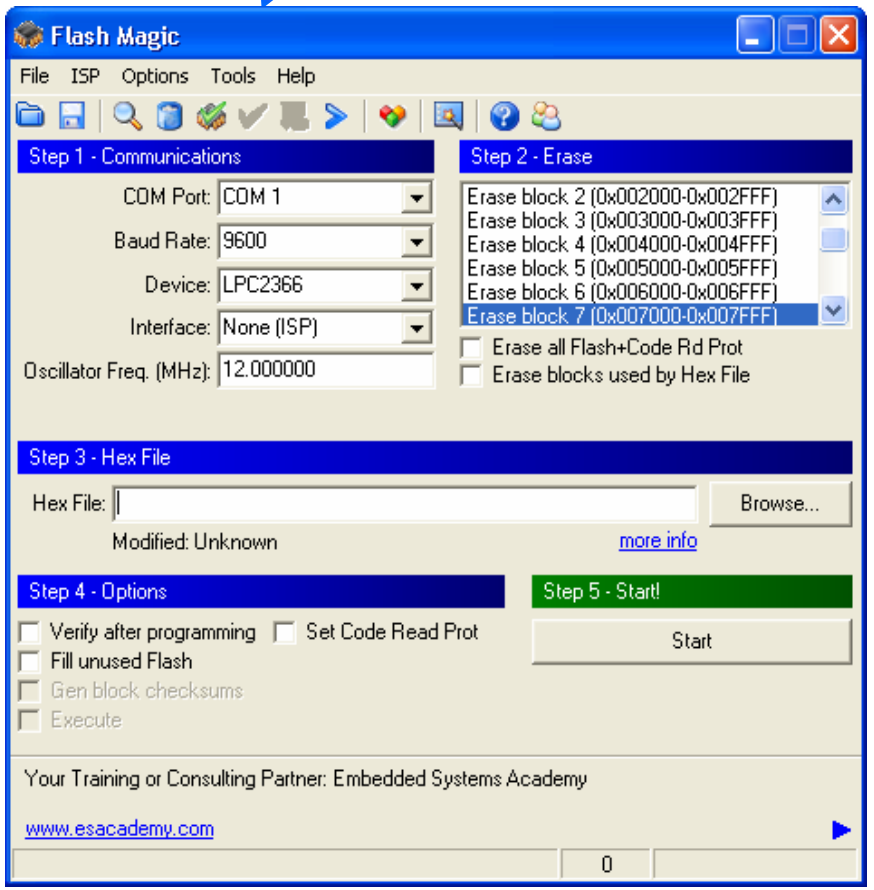
Development Tool Support

- EmbeddedICE-RT™ (JTAG)
- Embedded Trace Macrocell (ETM)
- Real Time Monitor with Debug Interrupt



Esacademy Flash Magic

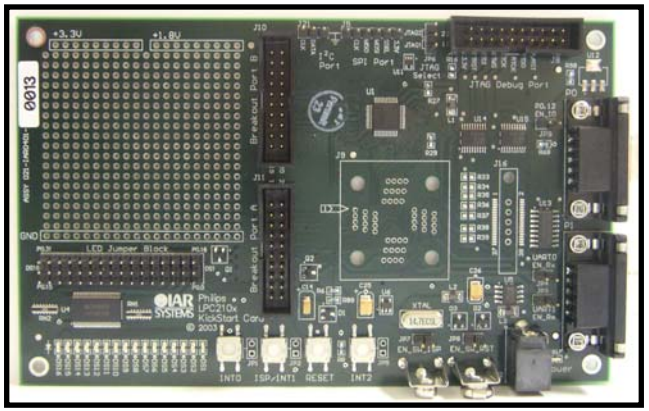
- Free Flashmagic Utility from Esacademy
- Features:
  - Program/Erase Flash
  - Verify
  - Blank check
  - Check ID
  - Fill Buffer
  - Save HEX file
  - Fo more info: http://www.esacademy.com/



Development tools - LPC2100/LPC2200

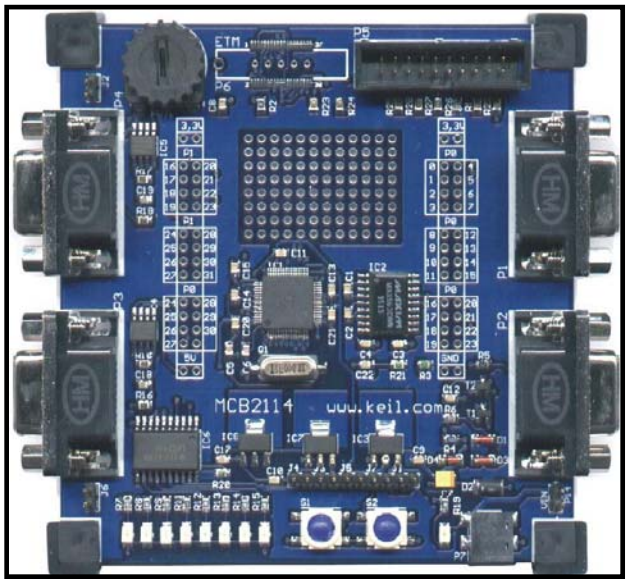
IAR/NXP board for LPC210x

- 2x 9-pin D-type Serial communications ports
- LED's can be connected to selected port-pins
- 3 switches for interrupts, Reset switch
- Breakout ports for Logic Analyzer connection
- Price projection $149 with 32k compiler



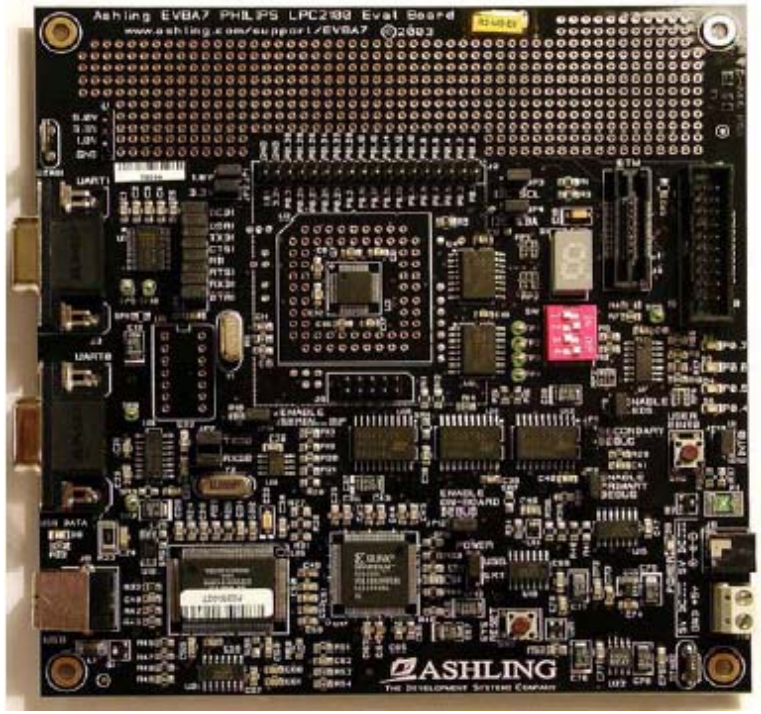
Keil/NXP board for devices with ADC and optional CAN (LPC2129)

- 2x 9-pin D-type Serial for serial co. ports
- 2x 9-pin D-type Serial for CAN
- 8 status LED's
- Switches for interrupt and Reset
- Potentiometer for ADC demos
- Price projection $149 with 16k compiler
  (no time limit)



Ashling ASK-2000 for LPC210x, LPC211x, LPC212x, LPC22xx

- LPC2104, LPC2105, LPC2106, LPC2114, LPC2119, LPC2124, LPC2129, LPC2210, LPC2212, LPC2214, LPC2290, LPC2292 and LPC2294, LPC213x
- **Jtag connector available**
- **Price: $295 with LPC2106**
- **Add $90 for LPC2129 or LPC2294**
- **Adapter available for the other 21xx and 22xx derivatives**
- **Emulator integrated in the same board !**

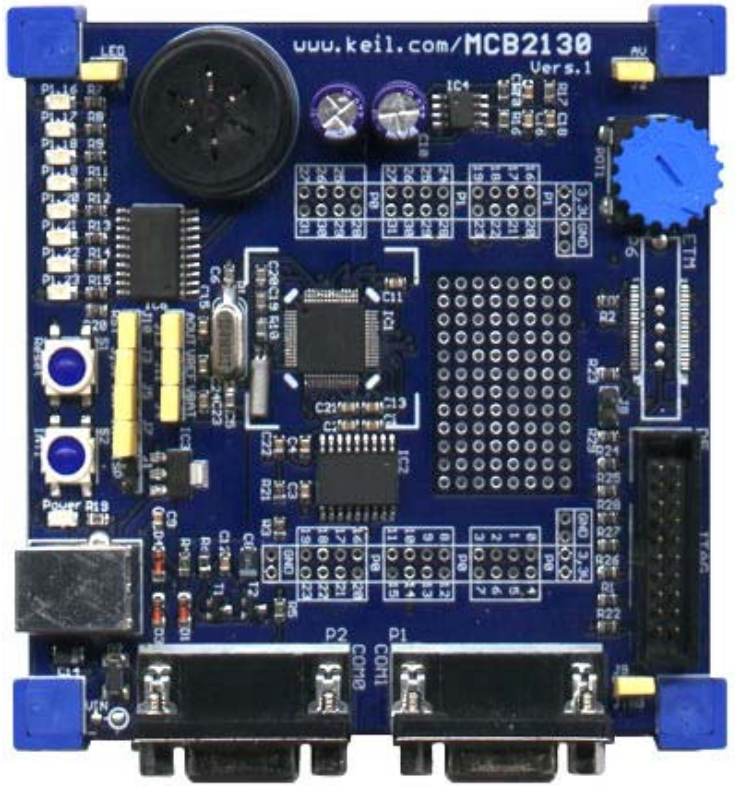


http://www.ashling.com/support/lpc2000/eval_kits.html

ASHLING
THE DEVELOPMENT SYSTEMS COMPANY

Keil MCB2130 Evaluation Board LPC2138

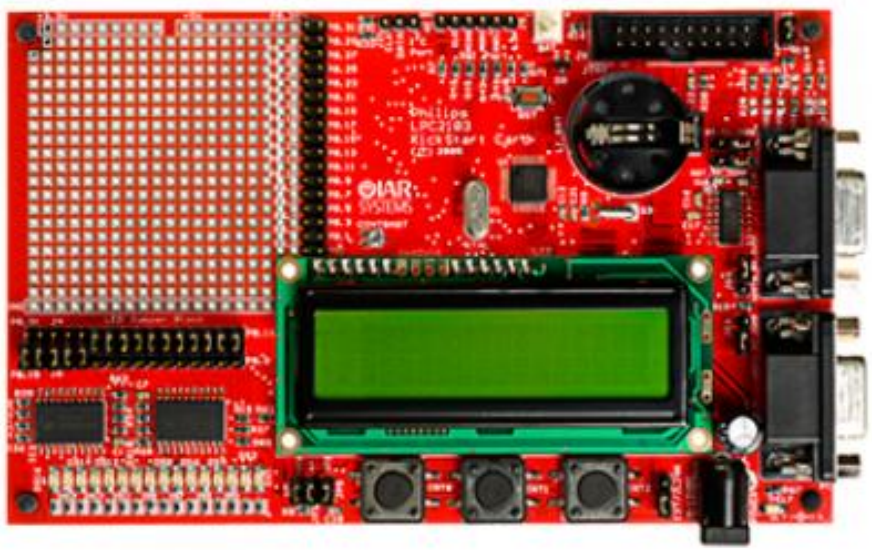
- 2x 9-pin D-type Serial for serial communications ports
- 8 status LED's
- Speaker on DAC output
- Buttons for interrupts and Reset
- Potentiometer for ADC demos
- Plated-through-hole prototyping matrix
- **Price: $149 with 16K compiler (no time limit)**



KEIL™ SOFTWARE

IAR KS2103 Evaluation Board

•LPC2103 Development board

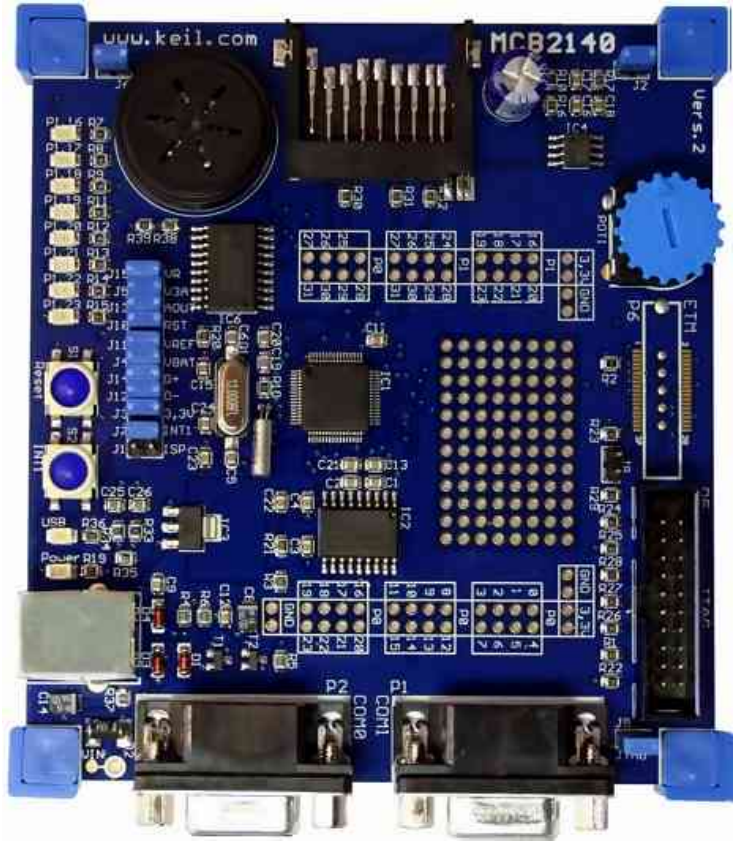


•LPC2103 MCU
• Two serial ports
• Reset button
• In-system programming (ISP) button
• Three user-defined buttons
• 16 fully configurable LEDs
• 16 character x 2 row LCD screen
• Power-on LED
• Can be powered via IAR J-Link-KS or external 9-12V DC power supply (not included)
• Lithium back-up battery holder
• 20-pin JTAG interface connector
• Breakout headers for all pins (suitable for mounting daughter boards)
• 20x20 array of plated holes for prototyping

Keil MCB214x Evaluation Board

- LPC2148 microcontroller
- 2x 9-pin D-type Serial for serial communications ports, power amplifier and on board loudspeaker
- 8 status LED's
- Speaker on DAC output
- Potentiometer for ADC demos
- SD Card Interface
- Software support for USB
- USB Soft Connect feature NOT supported

- **Price $149.00**



KEIL™ SOFTWARE

IAR KS214x Evaluation Board

- LCD interface
- RS232 ports for UART's
- Push buttons for external interrupts
- USB Soft Connect feature supported
- SD card interface for USB
- **Price $149.00**

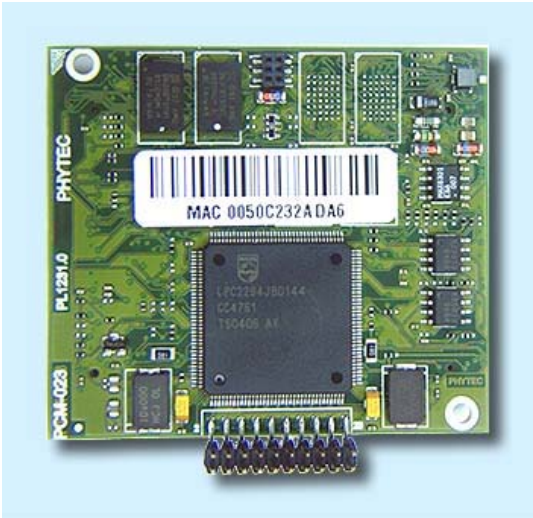
IAR SYSTEMS

PHYTEC phyCORE®ARM7/LPC229x

- phyCORE-ARM7/LPC229x
  PCM-023-SK-2294
- 10/60 MHz,
- 1 MB SRAM, 2 MB Flash,
  2 KB EEPROM,
- SMSC LAN91C111 10/100 Mbit/s
  Ethernet,
- JTAG interface

- $199.00

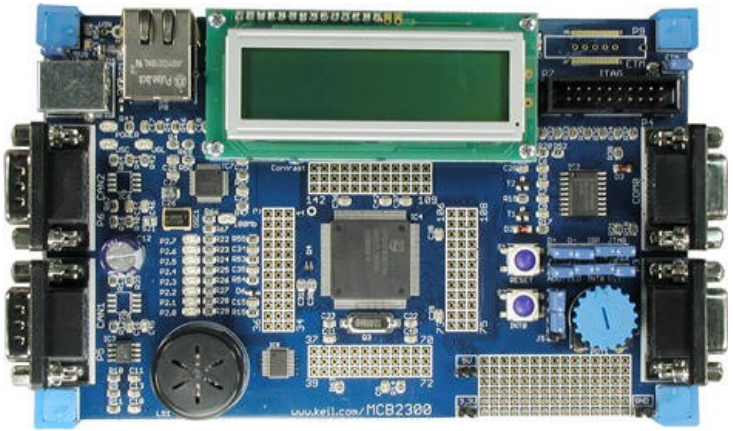
- http://www.phytec.com/sbc/32bit/pclpc229x.htm



PHYTEC
TECHNOLOGIE HOLDING AG

MCB23xx Keil for LPC2300 family

- Evaluation boards from Keil:
  - **MCB2360 (LPC2364/66/68)**
  - **MCB2370 (LPC2378)**



- Schematic
- Code sample on the web
- **Price: 199$**
- Two serial interfaces,
- a speaker, analog input (via potentiometer),
- two CAN interfaces,
- LCD, SD card interface
- USB, Ethernet,
- and eight LEDs make this board a great starting point for your next ARM project.

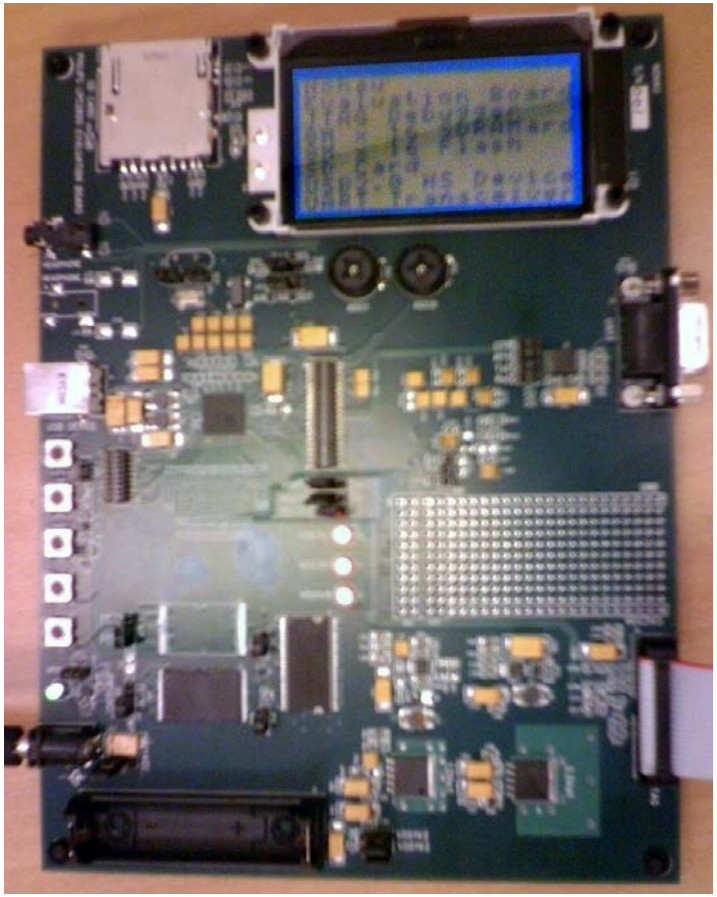
IAR Kick Start Development Kit for LPC2378

Contains:

- LPC2378 development board
- IAR J-LINK-KS JTAG debugger with USB connector
- IAR Embedded Workbench with a 32KB version of the IAR C/C++ Compiler
- **IAR PowerPac: 3 tasks (RTOS) and 1 file (Flash File System) evaluation version**
- 20-state version of visualSTATE

Nohau's LPC2800 evaluation board

- SD card Connector
- USB connector
- LCD Module
- 16MB SDRAM, 8MB Flash
- Headphone jack
- Price on Nohau's web site: 995$



http://www.nohau.com/emularm/lpc2800.html

ARM Emulators

JTAG emulators

noHau

ASHLING
THE DEVELOPMENT SYSTEMS COMPANY

IAR SYSTEMS

KEIL™ SOFTWARE

SIGNUM SYSTEMS

Green Hills
• SOFTWARE, INC. •

hitex
DEVELOPMENT TOOLS

Emulator with Trace

LAUTERBACH

http://www.ashling.com/datasheets/armtools.html

ASHLING
THE DEVELOPMENT SYSTEMS COMPANY

IAR SYSTEMS

Jlink + Trace

Debuggers

Pathfinder debugger for the LP2100

http://www.ashling.com/datasheets/armtools.html

ARM RealView debugger

http://www.arm.com/devtools/ads?OpenDocument&View=defaultBody2.

Seehau debugger ARM

http://www.nohau.com/downloads.html

C-SPY debugger ARM

Debuggers

Chameleon Debugger

http://signum.com/Signum.htm?p=ARM.htm

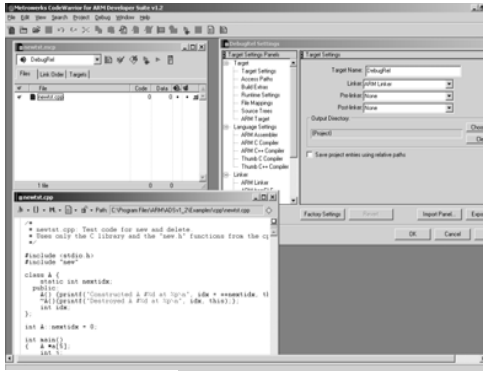
Keil debugger & simulator µVision3

http://www.keil.com/arm

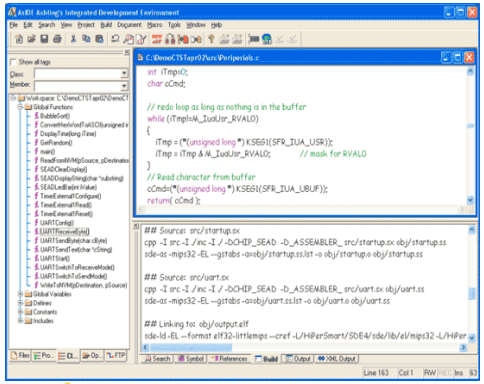
Universal Debug Engine (UDE)

MULTI Debugger

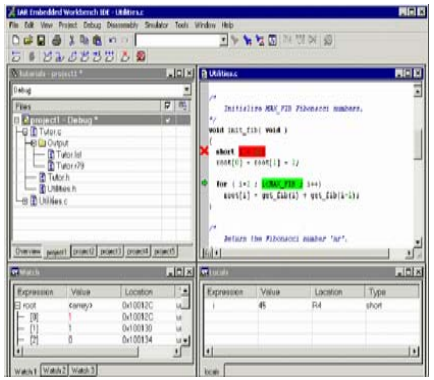
Integrated Development Environment



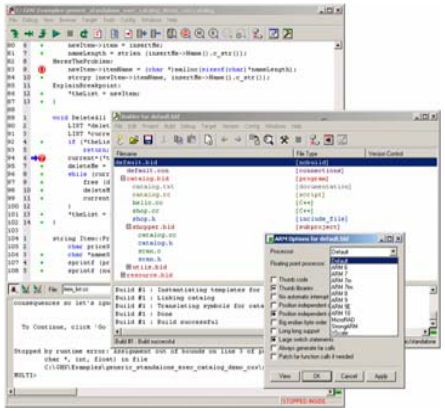
ARM RealView



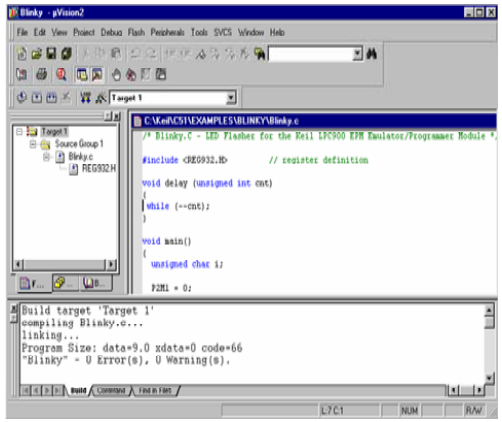
ASHLING THE DEVELOPMENT SYSTEMS COMPANY AsIDE



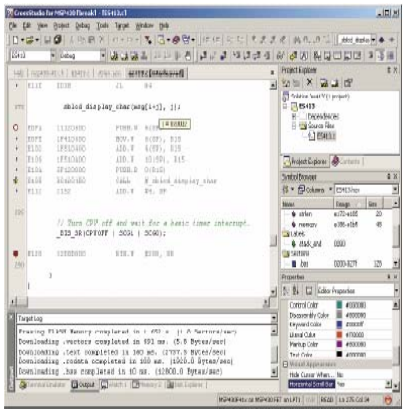
IAR SYSTEMS EWARM



Green Hills SOFTWARE, INC. Multi 2000

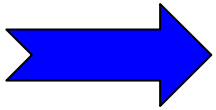


KEIL SOFTWARE™ µVision3



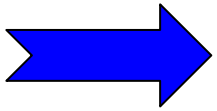
Rowley Associates Smart Software for Smart People CrossWorks

Compilers



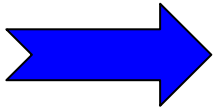
ARM Compiler

http://www.arm.com/devtools/soft_dev_tools?OpenDocument.



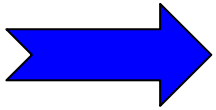
GHS Compiler

http://www.ghs.com/products/arm_development.html



IAR Compiler

http://www.iar.com/Products/?name=EWARM



GNU GCC

ARM RTOS

RTOS Listing and Ported to …

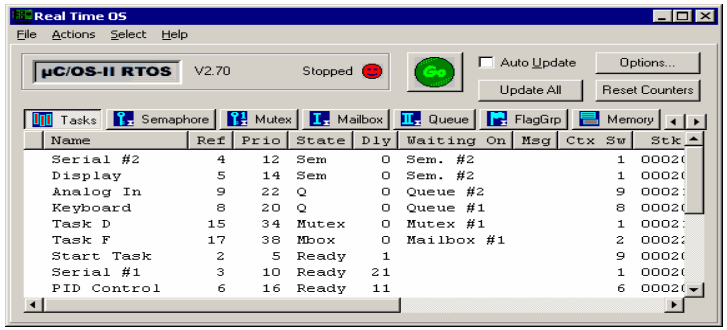
Nucleus	Nucleus Tools, Keil board
CMX	ARM,Keil
Keil ARTX	Keil
µCOS-II	IAR,Nohau,Keil (appnotes available online)
FreeRTOS	http://www.freertos.org/

RTOS Listing

eCos	Ashling
NicheTask	
ThreadX	IAR,ARM
Pumpkin Salvo	Keil
µClinux	See www.uclinux.org

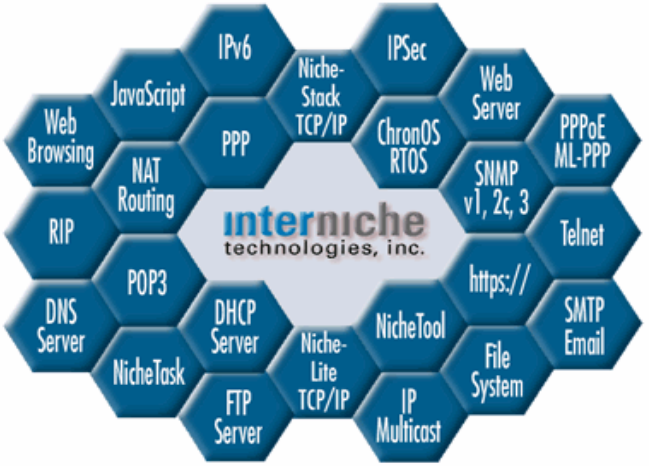
RTOS Support

Nucleus from
Accelerated Technology
a Mentor company



Micriμm

ChronOS™
from InterNiche



ARM Doc and software examples

Free link and tools:

- Free GNU C compiler for ARM7 + related manuals
  - http://www.gnuarm.com/
- Free LPC210x software for flash programming at:
  - http://www.lpc2100.com/
    http://www.lpc2000.com/
- Link to the ARM processor core documentation directly from ARM site at:
  - http://www.arm.com/documentation/ARMProcessor_Cores/index.html
- NXP ARM selection tools page:
  - http://www.nxp.com/products/microcontrollers/support/development_tools/tools_by_type/
- Yahoo support groups
  - http://groups.yahoo.com/group/lpc900_users
  - http://groups.yahoo.com/group/lpc2000/
- Free online Introduction to ARM
  - http://www.techonline.com/community/ed_resource/course/14612/
- Free LPC2000 Insider's Guide To The NXP ARM7-Based Microcontrollers
  - http://www.hitex.co.uk/arm/index.html

LPC21xx Driver Support (for free)

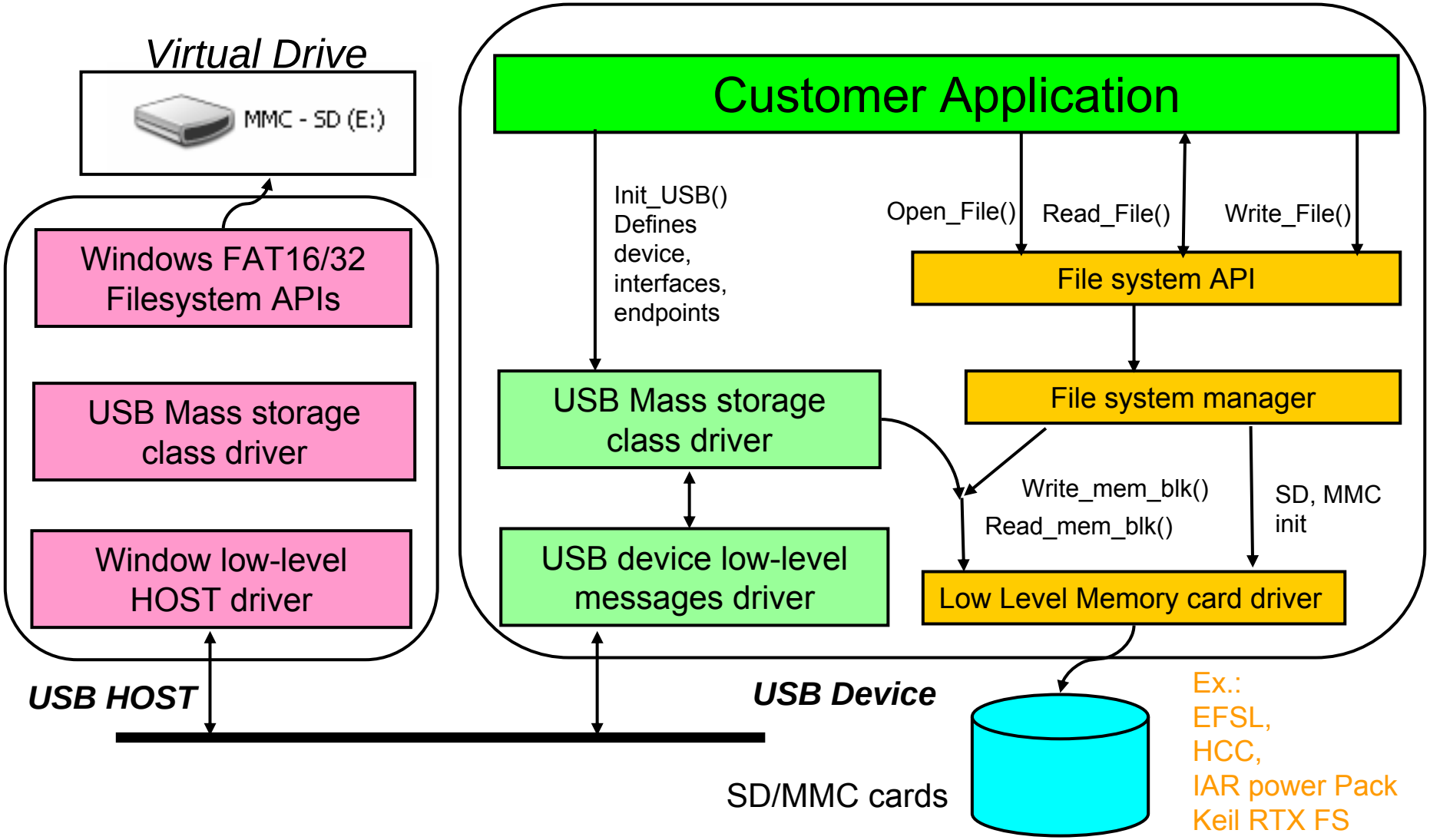
- Keil (using uVusion + Keil compiler)
  - USB mem: a Standard Memory Mass Storage driver example
    USB audio: an Audio Device driver example
  - USB HID: a Human Interface Device class driver example
- NXP
  - Virtual COM port: Maps a virtual COM port communication in a USB pipe (see LPC2000 news group, file section)
  - MMC driver: MMC memory card driver, application note (see NXP web site, microcontroller section)
- IAR:
  - USB mouse class driver example
  - USB MassStorage: Manage a MMC card and a vurtual RAM disk via USB
  - USB Audio: manage an input and an output audio stream via USB
  - USB CDC: USB communication device class via virtual COM
    CDC = Class Definition for Communication Devices
- Rowley :
  - TCP/IP, uIP stack for the Olimex LPC-E2124 board @ http://www.rowley.co.uk
- Free lib: EFSL http://efsl.de (FAT12/16/32 file system with short file names)

Keil and IAR examples don't require any Windows USB class driver. Windows XP supports all mentioned class drivers. NXP supplies the Virtual COM driver for Windows XP, IAR supplies the .inf file for the CDC windows driver

COMMERCIAL FILE SYSTEM SUPPORT

- • HCC
  - – Supports fat 16/32
  - – Wear levelling
  - – About 2000 Eur
- • Keil
  - – Supports Fat 16
  - – 8.3 file naming
  - – Part of RTL
- IAR
  - – Power PACK

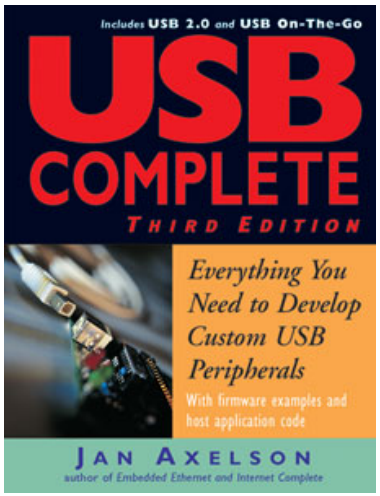
USB Masss-storage class drv. and Filesystem relationship



USB books

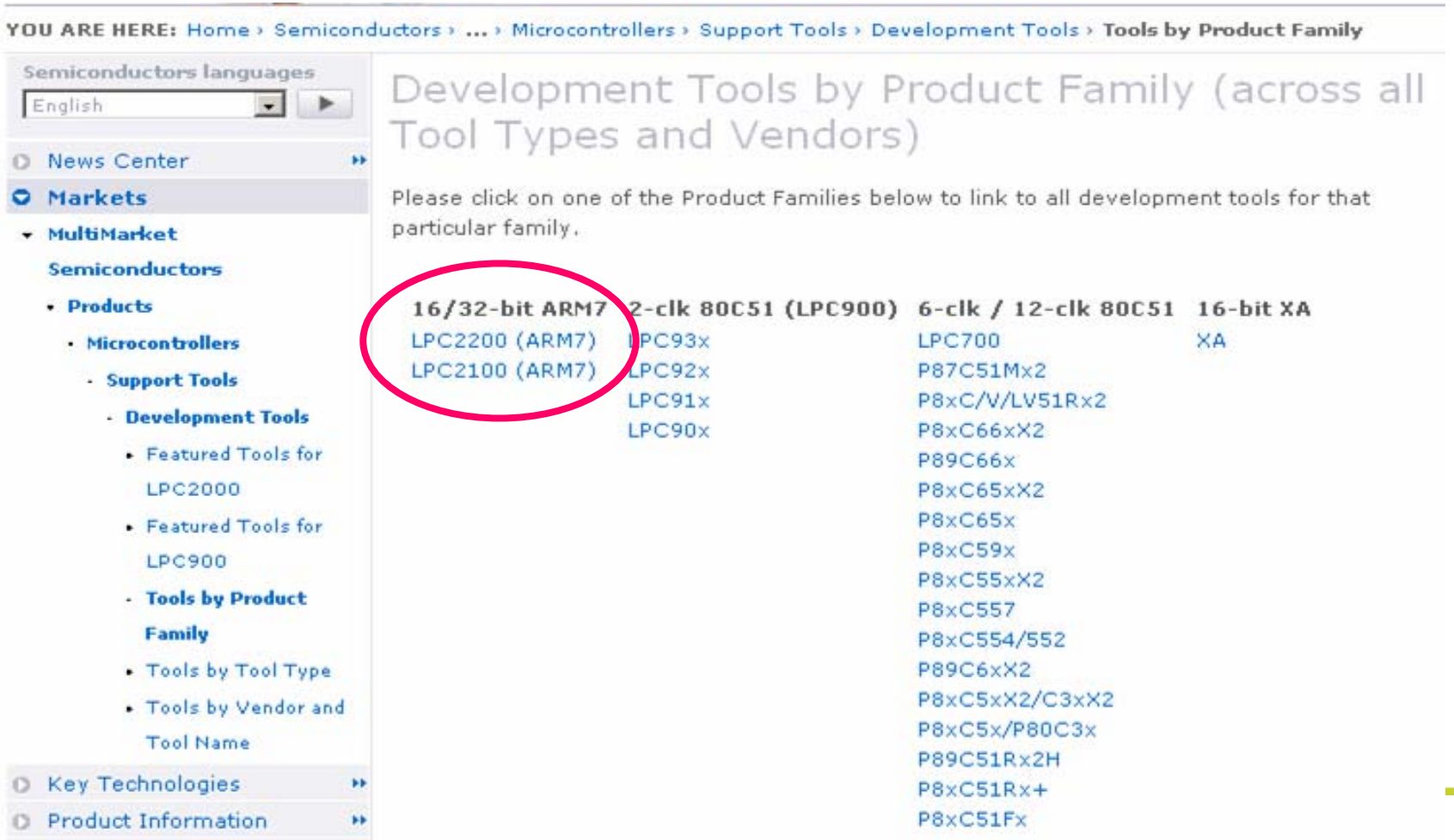
- USB Design By Example
  A pratical Guide to Building I/O devices
  John Hyde, Intel University Press

- USB complete third edition
  Jan Axelson, www.Lvr.com
  Lakeview Research



ARM-Tools on the NXP website

http://www.semiconductors.NXP.com/products/microcontrollers/support/development_tools/tools_by_family/



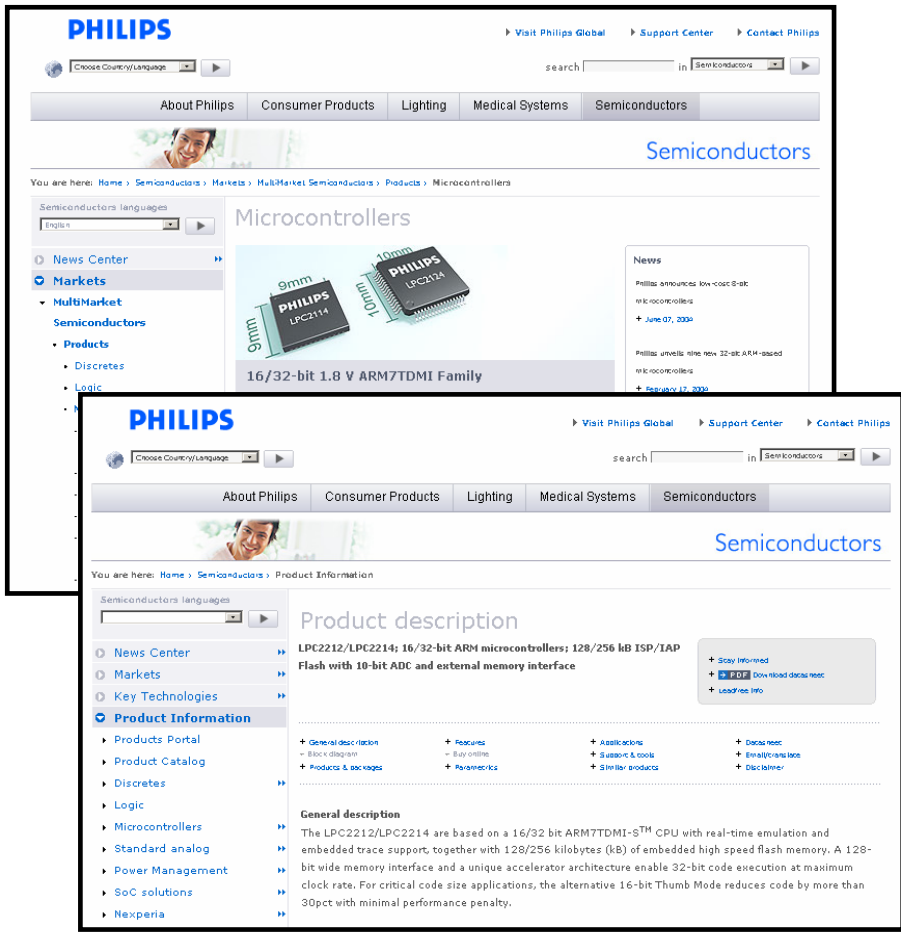
LPC2000 forum

- Started by Leon Heller, an engineering consultant from England: “The NXP LPC2100 family of ARM MCUs is sufficiently different from other ARM variants that I decided that a forum dedicated to it would be useful.”
- Direct URL http://groups.yahoo.com/group/lpc2000/
- Founded Nov 17, 2003
- Already about 4000 members!

Get the latest Information .....

- Product Selection Guide
- Datasheets
- Production Status
- Tools
- Application Notes
- And much more …

http://www.standardics.nxp.com/products/microcontrollers/



Insider's Guide to LPC2000

- 200 page guide to LPC2000 featuring chapters on:
  - ARM7 Core
  - Software Development
  - System Peripherals
  - User Peripherals
  - Keil Tutorial
  - GNU Tutorial
- Perfect for engineers without ARM experience

http://www.hitex.co.uk/arm/lpc2000book/book_downloadform.html

Application examples

Philips ARM Design Contest 2005 - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address http://www.jandspromotions.com/philips2005/ Go

Links Customize Links AltaVista Google Pages Jaunes Philips France Eurêka Yahoo! Yahoo! France Best of the Web Channel Guide Free Hotmail

LPC213x Philips ARM Design Contest 2005

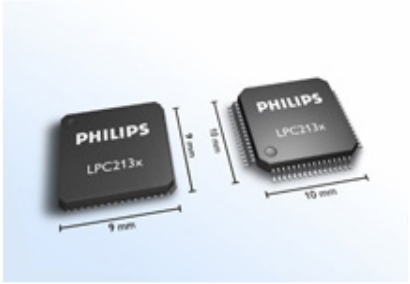
We've squeezed more in...

how much can you squeeze out?

The **Philips ARM Design Contest 2005** is your introduction to Philips's powerful LPC213x microcontroller family. These RISC MCUs are based on the high-performance 32-bit ARM7TDMI-S™ microprocessor, which features low-power consumption and 16-bit Thumb code for efficient and flexible designs. Other key features include a wide range of serial communications interfaces, on-chip 8/16/32-KB SRAM, 32-bit timers, single or dual 10-bit eight-channel ADCs, a 10-bit DAC, PWM channels, and 47 GPIO lines. These features coupled with a small footprint make the LPC2131/2132/2134/2136/2138 microcontrollers ideally suited for countless applications, such as access control, point of sale, communication gateways, voice recognition, industrial control, and medical systems.

Don't miss this major opportunity to show off your design talent.
Your design skills could win you a share of $15,000 in cash prizes!

Deadline: October 24, 2005



Contest Information:
- > Contest Home
- > Prizes
- > Rules
- > FAQ
- > Components
- > Forums
- > Request a Sample
- > Evaluation Board
- > Registration Form

PHILIPS

About the Sponsor | Philips | Circuit Cellar

http://www.jandspromotions.com/philips2005/

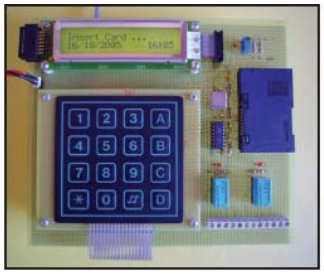
Internet

NXP ARM Design Contest

TAM-TAM

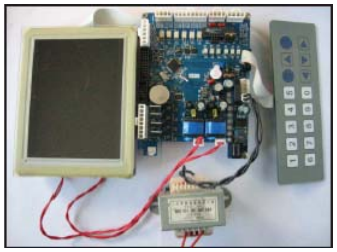
NoPC

Flash card …



Charlie

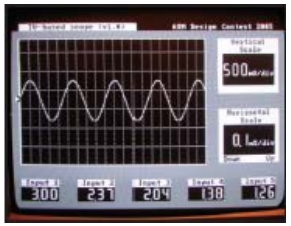
Magnetometer



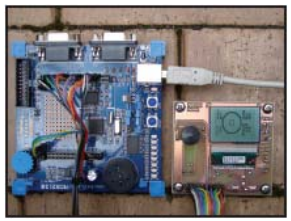
Nuclear Weighing Measurement

LAURIN

Ethernet Acquisition

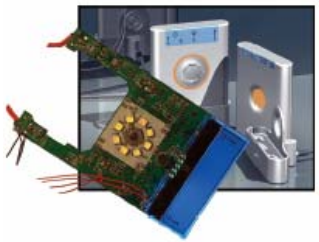


TV-Oscilloscope



Dual-Axis Level Sensor

Buckymeter



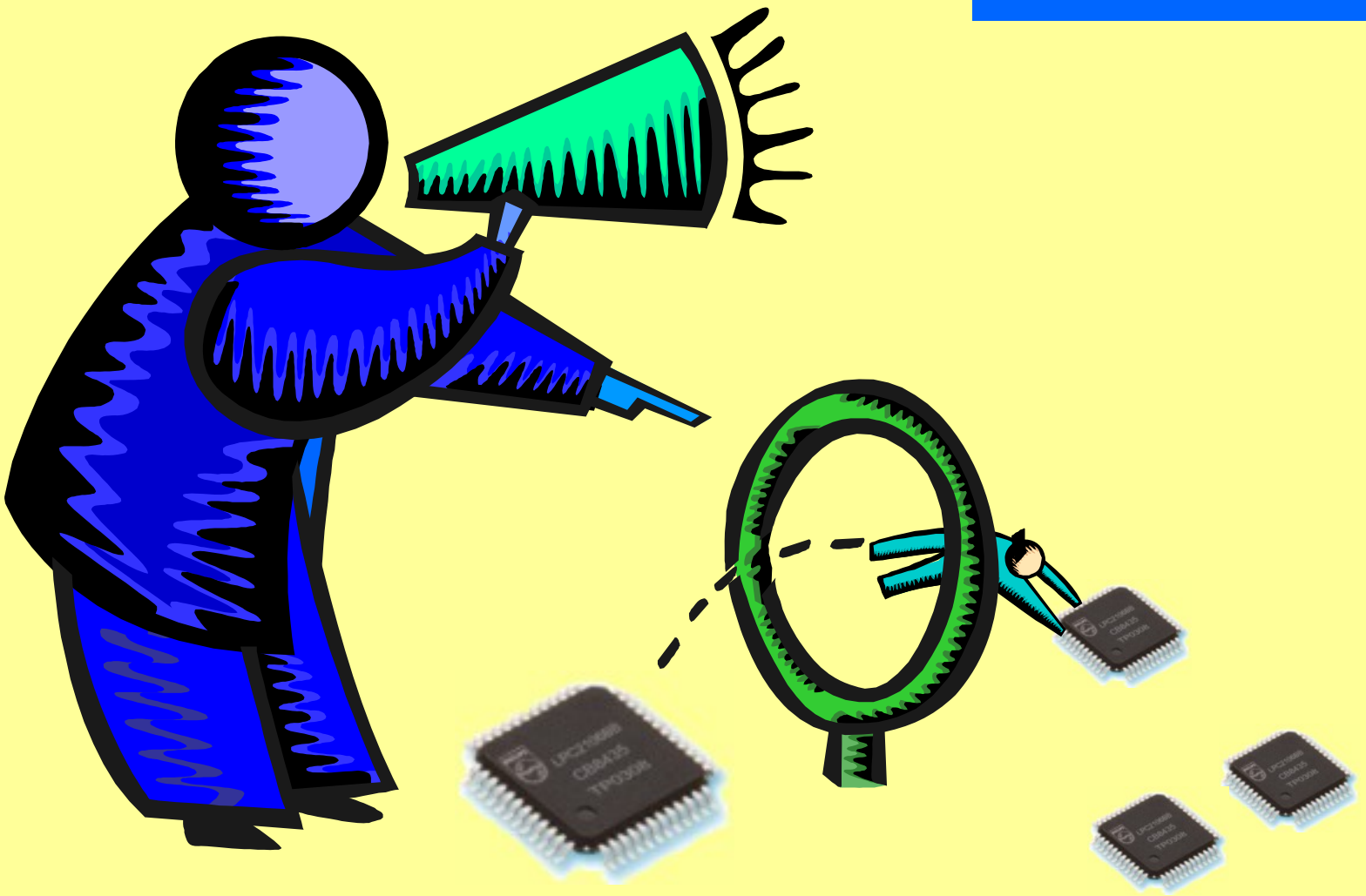
Duux Babycall

Distinctive Excellence

LPC2000 – Key Points

- ***Why ARM?***
  - **ARM7 is an open architecture**
  - **Already the preferred 32-bit solution in Automotive, Communication and Industrial markets.**
  - **Same tools for all ARM7 of different manufacturer. Often the same tools can be used for ARM9/10/11. Customers can save money**
  - **ARM is a scalable and uniform architecture. ARM7 is binary code compatible with ARM9/10. Your customer can chose the right product for its target application saving time due to the reduced learning curve**
- ***Why NXP?***
  - **First ARM7TDMI supplier with on-board flash in 0.18 μm process, with the largest ARM7 product portfolio**
  - **Memory sizes from 8k up to 1M on-chip flash**
  - **Highest flash performance with nearly zero wait states due to internal MAM (Memory Accelerator Module)**
  - **Widest selection of devices and of the integrated peripherals: 4*CAN, 2*ADC, SPI, I^2C…)**
  - **Lowest pin count and smallest packages available**
  - **Price level allows to address mid/high end 16-bit applications, and high end 8 bit application**

Hands-On



Hands-On Index

Hands-On

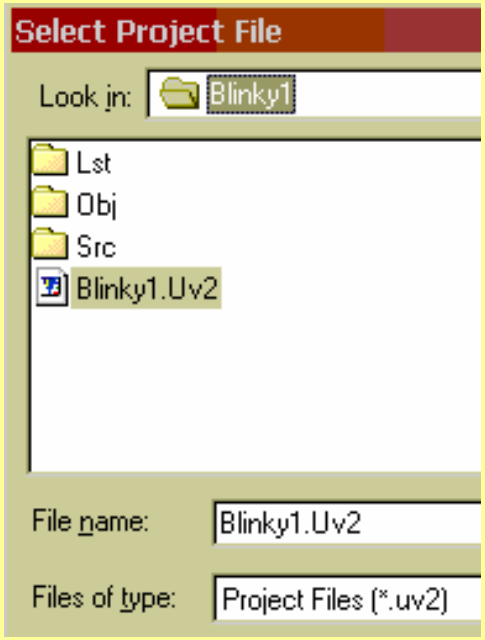
1. Tools Setup
2. Oscillator / PLL / MAM / GP I/O / Flash
3. ADC
4. Interrupts / Timer
5. UART / CAN

Start Keil uVision3

Hands-On

- Menu Selection:
  Project /
  Open Project

- Select
  Blinky1



- Compile Current File

- Make Project

- Rebuild Project (all)

- Open Options Dialog

- Start/Stop Debugging

Build and Debug

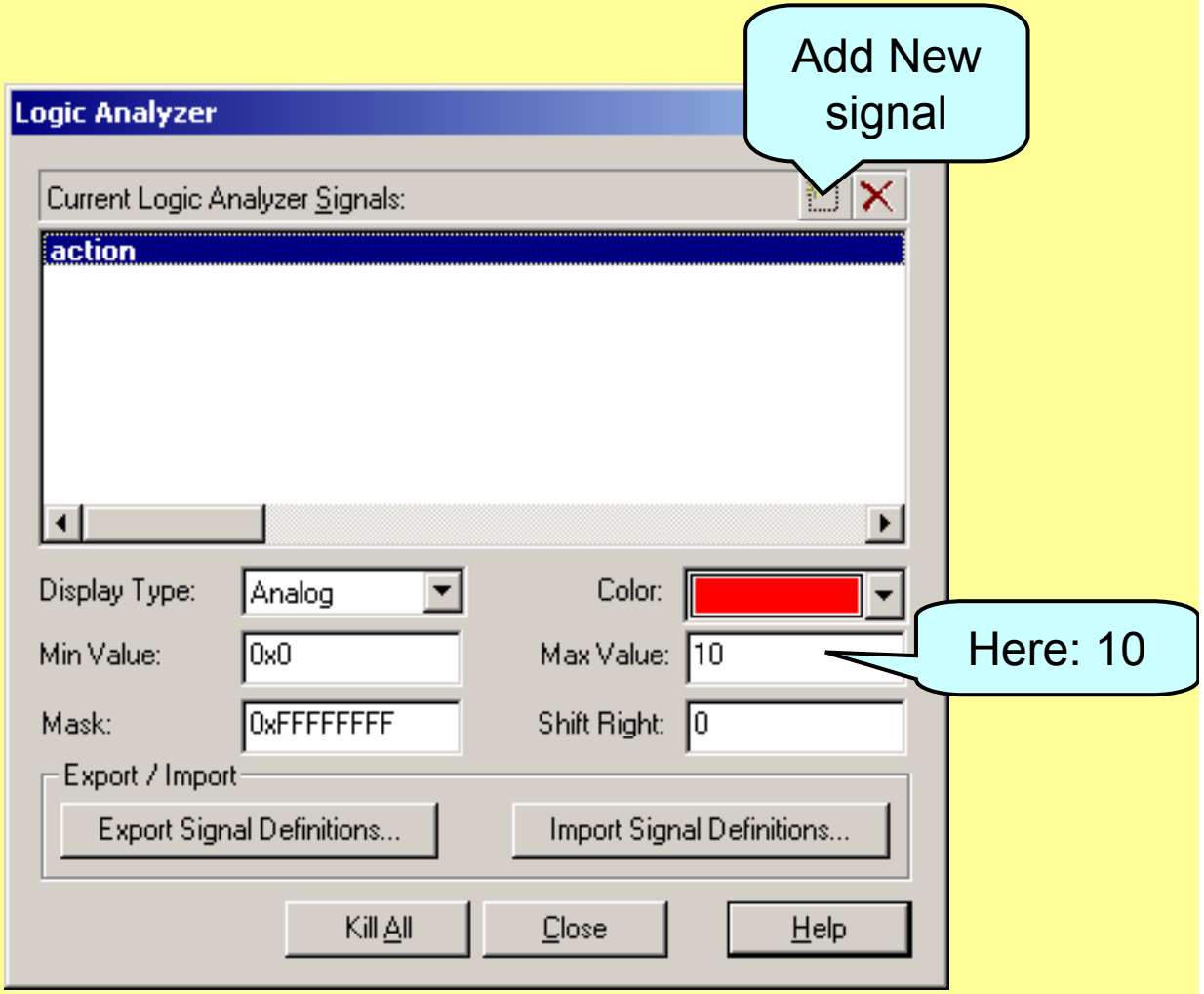
Hands-On

-  Build Project
-  Start Debugging
-  Reset
-  Run
-  Halt
-  Step into
-  Step over
-  Performance Analyzer
-  Signal Analyzer

Setup of Logic Analyzer

Hands-On

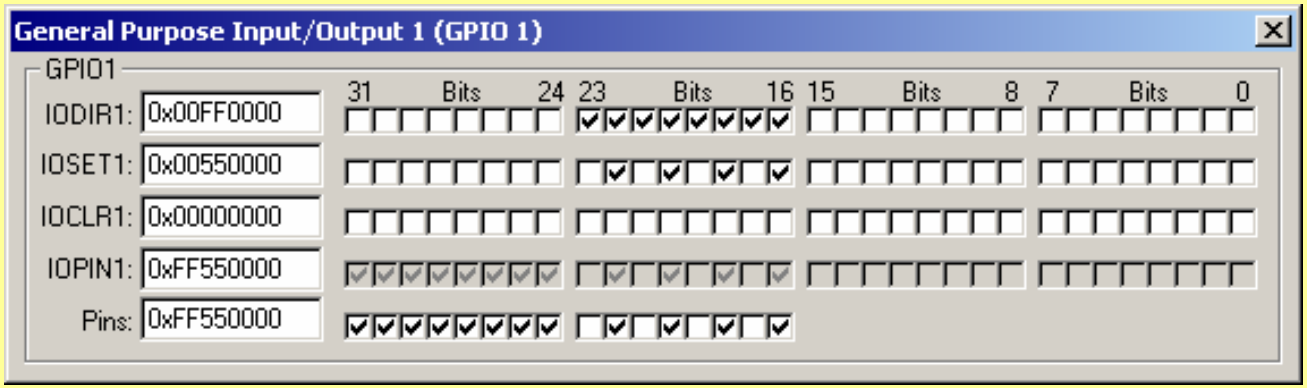
- Logic Analyzer
  - Setup
  - Add signal “action”



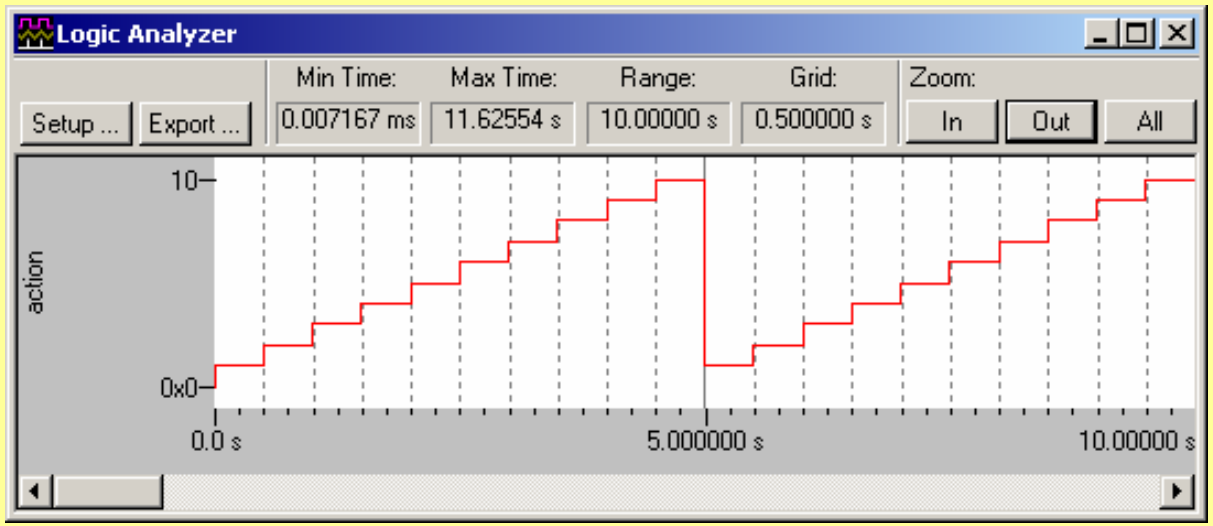
Run Program
View Variables and Registers

Hands-On

- Menu:
  Peripherals /
  GPIO /
  Port1



- Zoom:
  In & Out



Run Program on Hardware

Hands-On

- Halt

- Stop Debugging

- Options

- Make Settings:
  - Using JTAG/ULINK
    - Select Debug Tab
    - Use "ULINK ARM Debugger"
  - With Flash ISP Utility
    - Select Utilities Tab
    - Use External Tool
    - Modify Command Line
    - Run Independent

- Press Load Button

Introduction to the ARM architecture

1. **ARM - The Company**
2. Architecture
3. Instruction Set
4. ARM7TDMI-S
5. Systems
6. Other ARM cores
7. NXP Implementation
8. Q & A

ARM Holdings plc. (1)

- Established as **Advanced RISC Machines Ltd.** in 1990 as a UK based joint venture between Apple Computer, Acorn Computer Group and VLSI Technology*
  - Apple and VLSI provided funding
  - Acorn supplied technology and first 12 engineers
- Introduction of ARM6™ family in 1991, VLSI initial licensee
- In April 1998 listed on the London Stock Exchange and Nasdaq

*: part of Philips since 1999

ARM Holdings plc. (2)

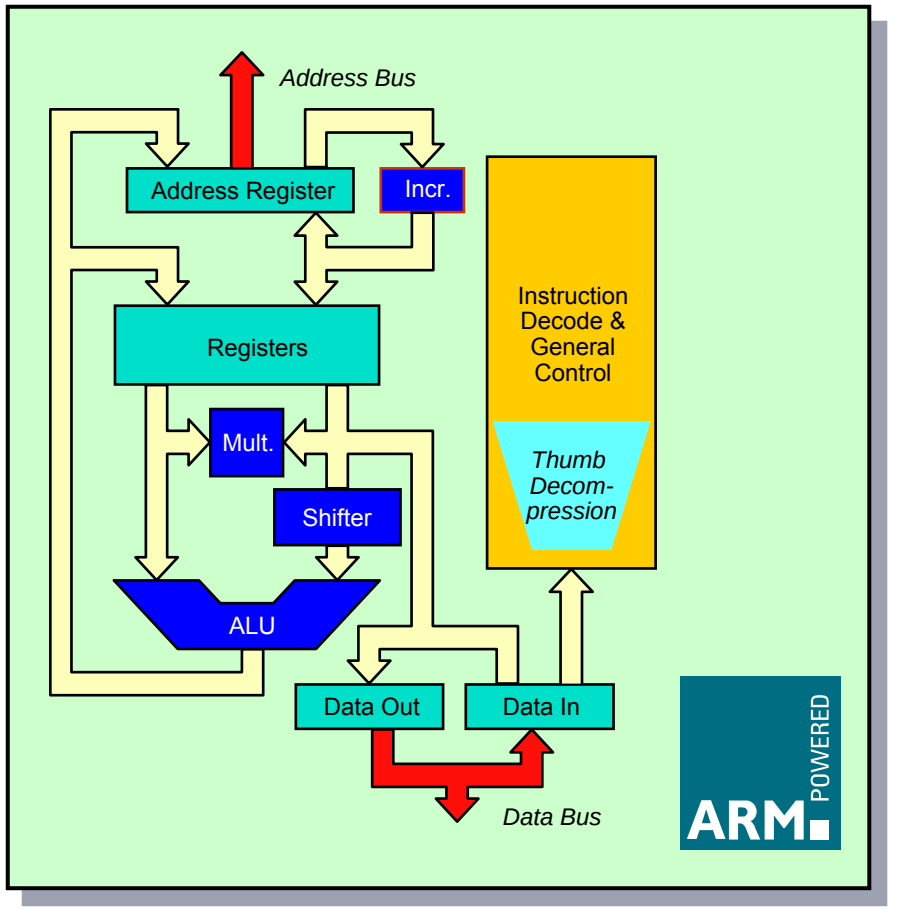
- Develops the ARM range of RISC processor cores
- Licenses its RISC microprocessor core and SoC IP to a network of partners; semiconductor and system companies
- ARM does not manufacture silicon itself
- Also licenses architectural extensions, development tools, peripheral IP and SoC solutions
- ARM's market share of the embedded RISC microprocessor market is approx. 75% and to date, ARM Partners have shipped more than one billion ARM core-based microprocessors

1. ARM - The Company
2. **Architecture**
3. Instruction Set
4. ARM7TDMI-S
5. Systems
6. Other ARM cores
7. NXP Implementation
8. Q & A

Bus Width

- ARM7 is a 32-bit architecture
  - Data pathes and (**ARM**) instructions are 32 bits wide
  - Von Neumann architecture
    - instructions and data use the same 32-bit data bus
  - There is a subset of 16-bit instructions (**Thumb**) optimized for code density*

*: from C code

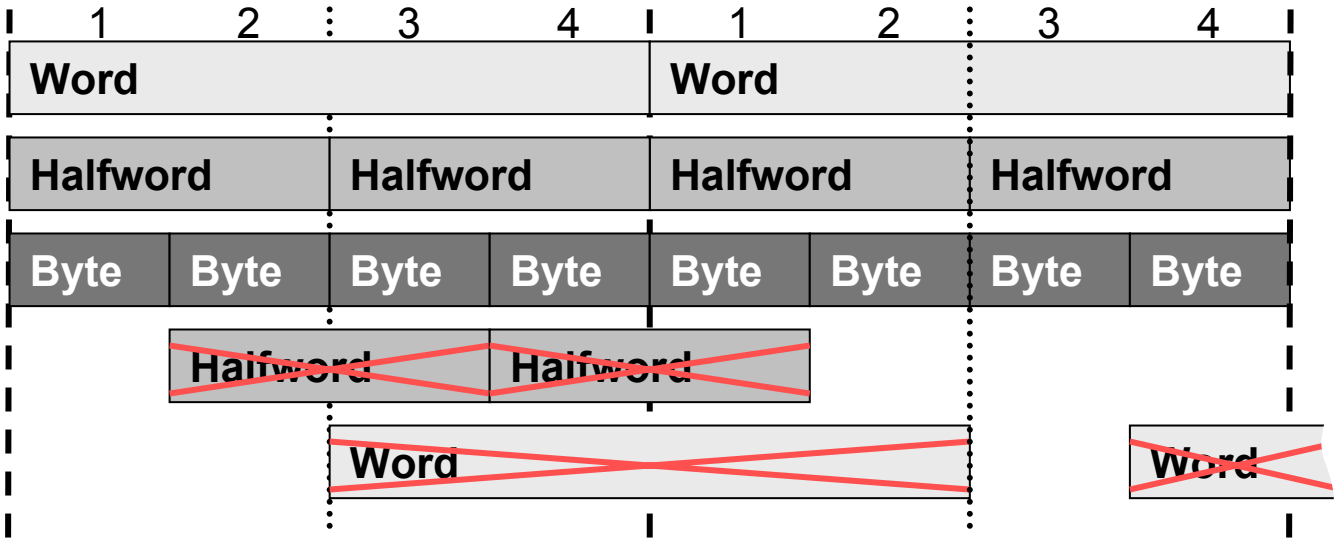


Thumb State

- Set of instructions re-coded into 16 bits
  - Improved code density by ~ 30%
  - saving program memory space
- In Thumb state only the program code is 16-bit wide
  - after fetching the 16-bit instructions from memory, they are de-compressed to 32 bit instructions before they are decoded and executed
  - all **operations are still 32-bit** operations

Data Types and Alignment

- Definitions (Little endian or big endiand are options):
  - Word = 32 bits (four bytes)
  - Halfword = 16 bits (two bytes)
  - Byte = 8 bits



Processor Modes

ARM has seven operating modes

1. **User** unprivileged mode under which most applications run
2. **FIQ** entered, when a high priority (fast) interrupt is raised
3. **IRQ** general purpose interrupt handling
4. **Supervisor** protected mode for the operating system

   entered on reset or software interrupt instruction
5. **System** privileged mode using the same registers as user mode

   (not in ARM architectures 1, 2 and 3)
6. **Abort** used to handle memory access violations
7. **Undefined** used to handle undefined instructions

Registers (1)

An ARM core has 37 registers (32-bits wide)

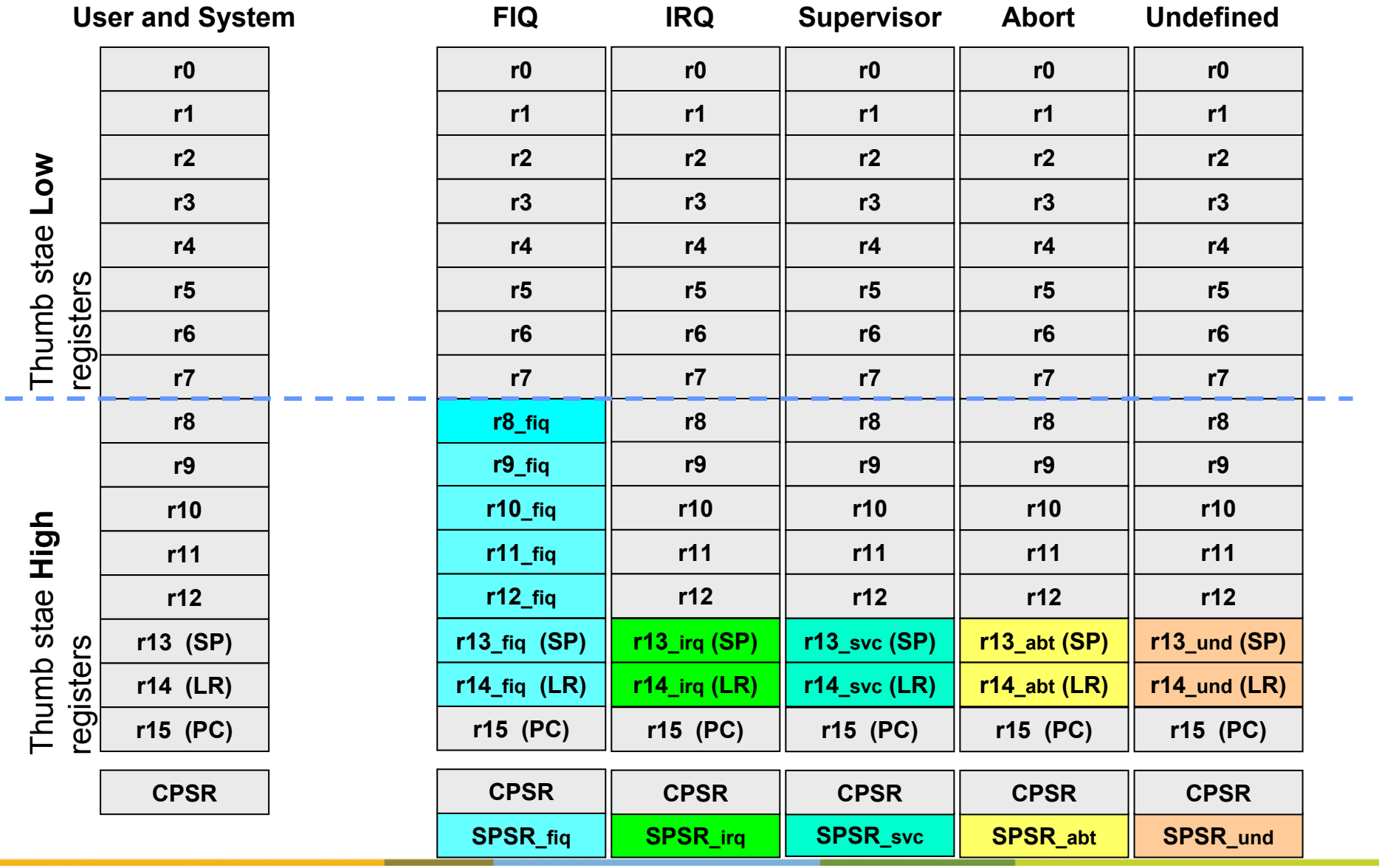
- General purpose registers
  - 1 program counter
  - 30 general purpose registers
- Status registers
  - 1 current program status register
  - 5 saved program status registers

These registers are not all accessible at the same time. The processor state and operating mode determine which registers are available to the programmer.

Registers (II)

- Depending on processor mode one of several banks is accessible. Each mode can access
  - the program counter **r15** (**PC**)
  - a particular **r13** (stack pointer **SP**)
  - a particular **r14** (subroutine link register, **LR**)
  - a set of **r0-r7** registers, and a particular set of **r8-r12**
  - the current program status register (**CPSR**)
- Privileged modes (except Sytem mode) can also access
  - a particular **SPSR** (saved program status register)

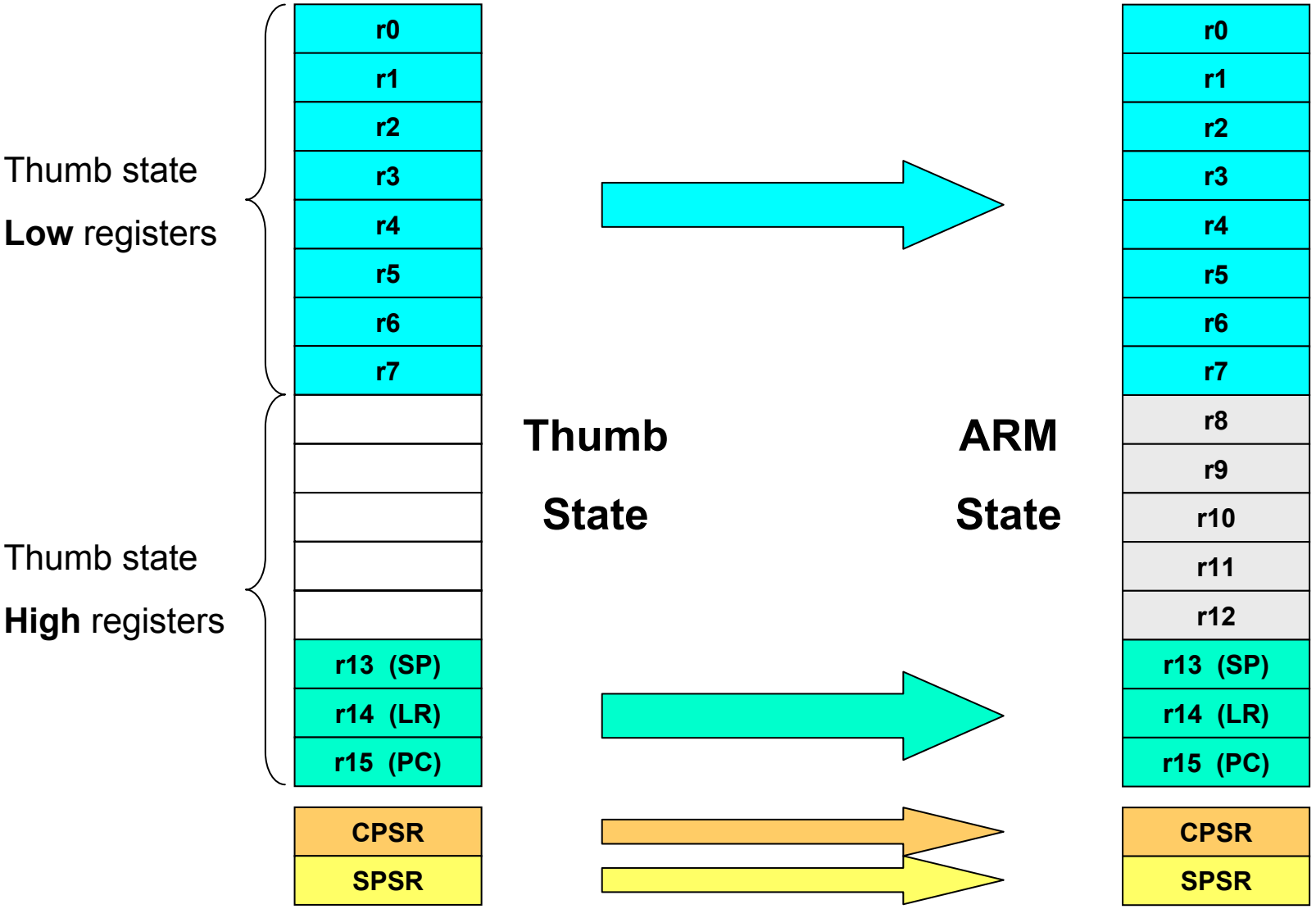
Register Overview



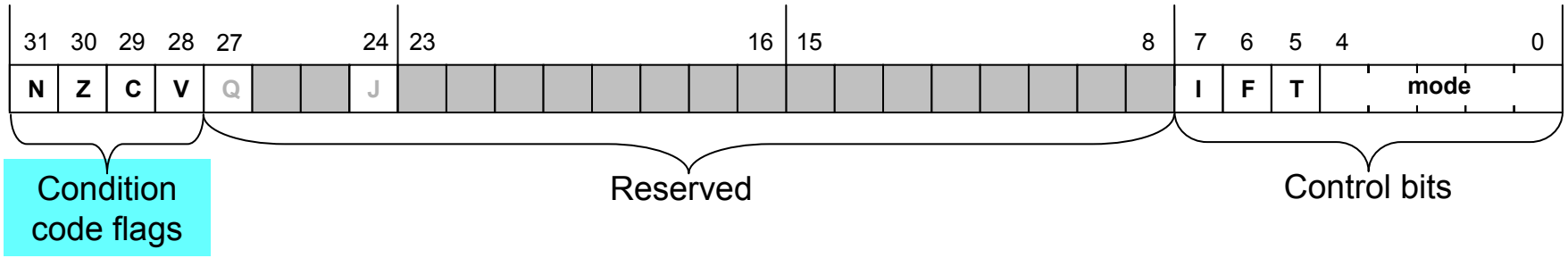
Registers in Thumb State

- The Thumb state register set is a subset of the ARM state set. The programmer has direct access to:
  - eight general registers r0 - r7
  - the program counter PC
  - a Stack pointer SP
  - a Link register LR
  - the current program status register CPSR
- In Thumb state, the high registers (r8 - r15) are not part of the standard register set. The assembly language programmer has limited access to them, but can use them for fast temporary storage

Thumb vs. ARM


Thumb state
Low registers
Thumb state
High registers
r0
r1
r2
r3
r4
r5
r6
r7
r13 (SP)
r14 (LR)
r15 (PC)
CPSR
SPSR
Thumb
State
ARM
State
r0
r1
r2
r3
r4
r5
r6
r7
r8
r9
r10
r11
r12
r13 (SP)
r14 (LR)
r15 (PC)
CPSR
SPSR

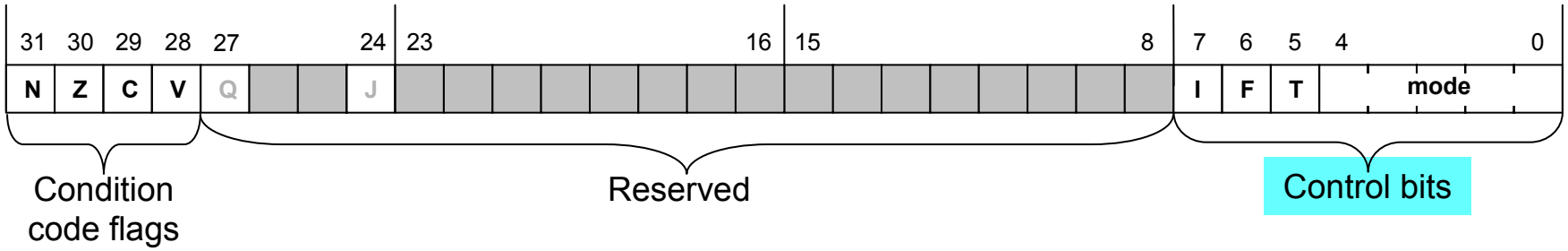
Program Status Register (1)



- Condition Code Flags
  - N: Negative or less than
  - Z: Zero
  - C: Carry or borrow or result of the shift operations
  - V: Overflow

To not disturb reserved bits, a read-modify-write strategy should be applied to change PSR bits.

Program Status Register (2)



- Interrupt Disable Bits
  - I: IRQ interrupts **disable**
  - F: FIQ interrupts **disable**
- T Bit
  - Thumb mode (when set)
  - ARM mode (when cleared)

- Mode Bits

10000	User
10001	FIQ
10010	IRQ
10011	Supervisor
10111	Abort
11011	Undefined
11111	System

Program Counter (r15)

- When the processor is executing in **ARM** state
  - all instructions are 32 bits wide
  - all instructions must be word aligned
  - bits [31:2] contain the PC, bits [1:0] are zero

    (instructions cannot be halfword or byte aligned)
- When the processor is executing in **Thumb** state
  - all instructions are 16 bits wide
  - all instructions must be halfword aligned
  - bits [31:1] contain the PC, bit [0] is zero

    (instructions cannot be byte aligned)

Exception

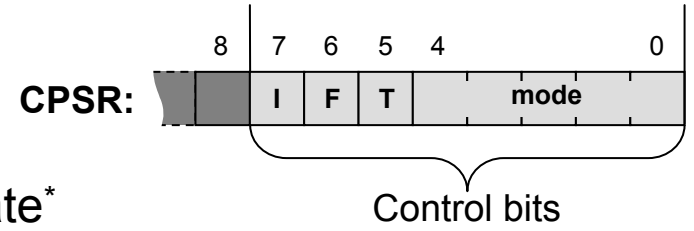
- Exceptions result whenever the normal flow of a program has to be halted temporarily, for example to service an interrupt from a peripheral. Before attempting to handle an exception, the ARM7TDMI-S preserves the current processor state so that the original program can resume when the handler routine has finished.

Exception Handling

- Entering an exception the ARM core
  - saves the address of the next instruction in the appropriate LR

  - copies the CPSR into the appropriate SPSR

  - sets appropriate CPSR bits
    - interrupt disable bits
    - mode field bits
    - if running in Thumb state, enter ARM state*

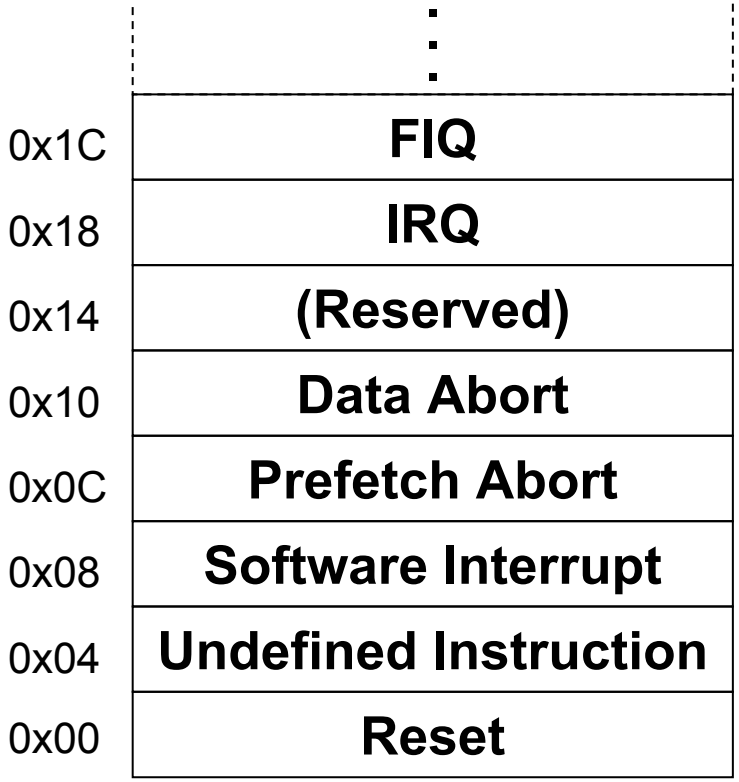


  - forces PC to fetch next instruction from relevant exception vector

*: all exceptions are handled in ARM state!

Exception Vectors

- Vector Table



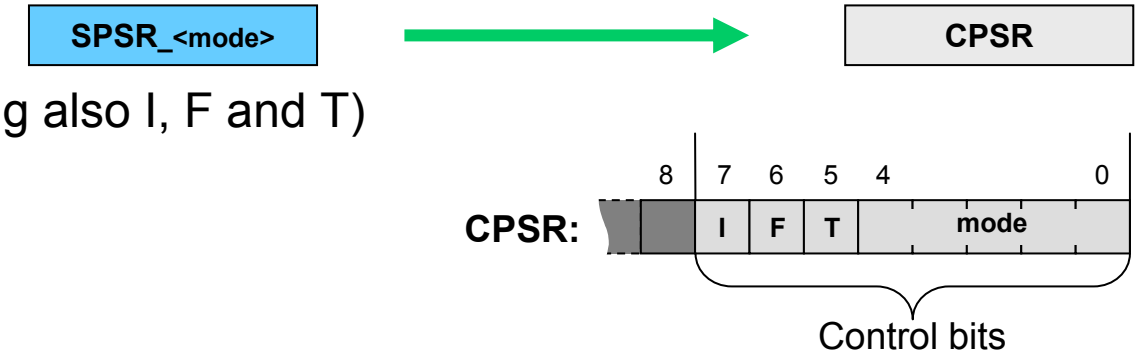
Multiple Exceptions

- Exception priorities
  - When multiple exceptions arise at the same time, a fixed priority sytem determines the order in which they are handled

1.	Reset	highest priority
2.	Data Abort (data memory access cannot be completed)	
3.	FIQ	
4.	IRQ	
5.	Prefetch Abort (instruction memory access cannot be completed)	
6.	Undefined Instruction	
7.	SWI - Software Interrupt (to enter supervisor mode)	lowest priority

Leaving Exception

- To leave an exception, the exception handler must
  - copy SPSR back into CPSR



(automatically restoring also I, F and T)

  - move contents of current LR minus offset* to PC

  - *: varies according to type of exception: 2, 4 or 8

1. ARM - The Company
2. Architecture
3. **Instruction Set**
4. ARM7TDMI-S
5. Systems
6. Other ARM cores
7. NXP Implementation
8. Q & A

Instruction Set

- All instructions are 32-bits long
- Many instructions execute in a single cycle
- Instructions are **conditionally** executed
- ARM is a load / store architecture
  - via registers => RISC
- Load or store multiple registers in a single instruction

  using <register list>

Conditional Execution

Mnemonic	Description
EQ	Equal
NE	Not equal
CS / HS	Carry Set / Unsigned higher or same
CC / LO	Carry Clear / Unsigned lower
MI	Negative
PL	Positive or zero
VS	Overflow
VC	No overflow
HI	Unsigned higher
LS	Unsigned lower or same
GE	Signed greater than or equal
LT	Signed less than
GT	Signed greater than
LE	Signed less than or equal
AL	Always (normally omitted)

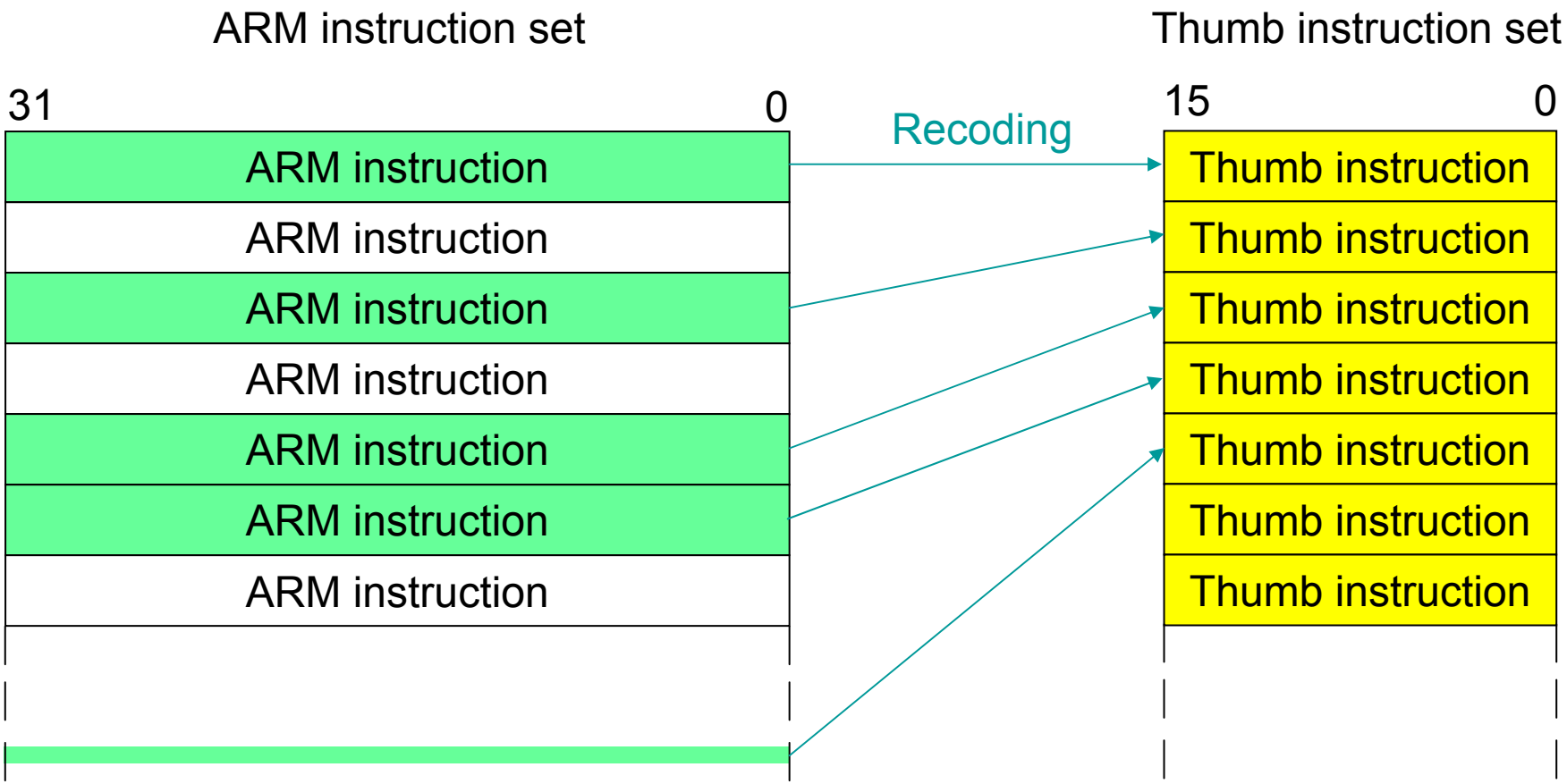
Instruction Examples

- Data processing instructions
  - SUB r0, r1, #5 — *r0 := R1 - 5*
  - ADD r2, r3, r3, LSL #2 — *r2 := r3 + (r3, LSL #2)*
  - ADDS r4, r4, #0x20 — *r4 := r4 + 32 and set flags*
  - ADDEQ r5, r5, r6 — *r5 := r5 + r6 if equal*
- Specific memory access instructions
  - LDR r0, [r1, #4] — *r0 := [r1 +4]*
  - STRNEB r2, [r3, r4] — *[r3 + r4] := r2 Byte operation if Z = 0; ignores r2[31:8]*
  - LDRSH r5, [r6, #2]! — *r5 := [r6 + 2] Halfword sign-ext. set bit [31:16] to bit 15 then r6 := r6 + 2*

Thumb Instruction Subset

- Subset of most commonly used 32-bit ARM instructions
  - 2 address format: destination register same as one source registers
- Compressed into 16-bit wide code
  - Improved code density
- Decompressed on execution to full 32-bit instructions
  - transparently
  - in real-time
  - no performance loss
- ARM code can be combined with Thumb code for maximum flexibility

Thumb Instructions



Thumb Instruction Set (1)

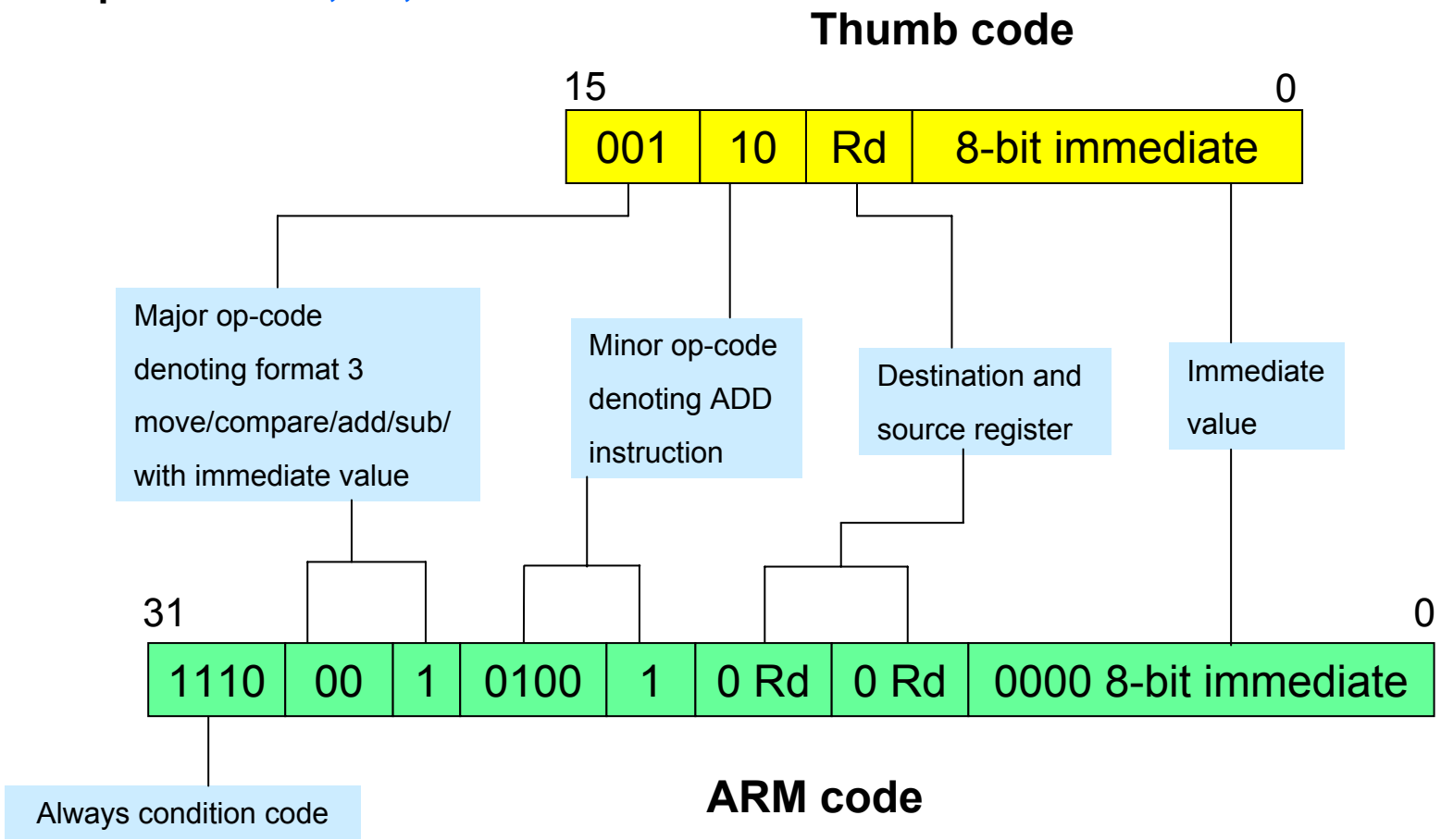
- Instruction Types
  - Branch
    - Unconditional ± 2KBytes
    - Conditional ± 256Bytes
    - Branch with Link ± 4MBytes (2 Instructions!)
    - Branch and exchange change to ARM state if Rm[0] = 0
    - Branch and exchange with Link
  - Data Processing
    - Subset of ARM data processing instructions
    - Not conditionally executed (but some update flags)

Thumb Instruction Set (2)

- Instruction Types
  - Load and Store
    - Register plus 5-bit (PC,SP plus 8) immediate addressing
    - Register plus Register addressing
  - Load and Store Multiple
    - Load / Store list of registers
    - Push / Pop (ARM equivalent: STMDB SP!, <registers>)
  - Exception Generating Instructions
    - SWI (switch to ARM mode and privileged mode)
    - Breakpoint (prefetch abort, with debug monitor)

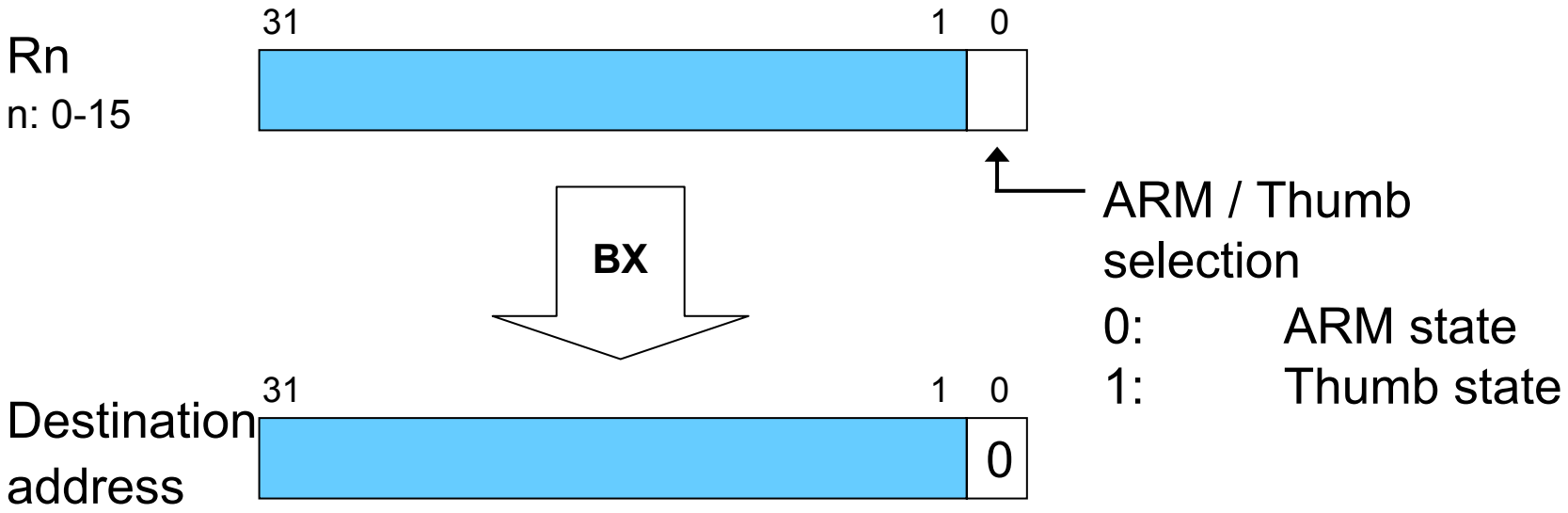
Translation of Thumb Instruction

Example: ADD Rd, Rd, # Constant



ARM and Thumb Interworking

- Switch between ARM state and Thumb state using BX instruction
  - In ARM state: BX<condition> Rn
  - In Thumb state: BX Rn



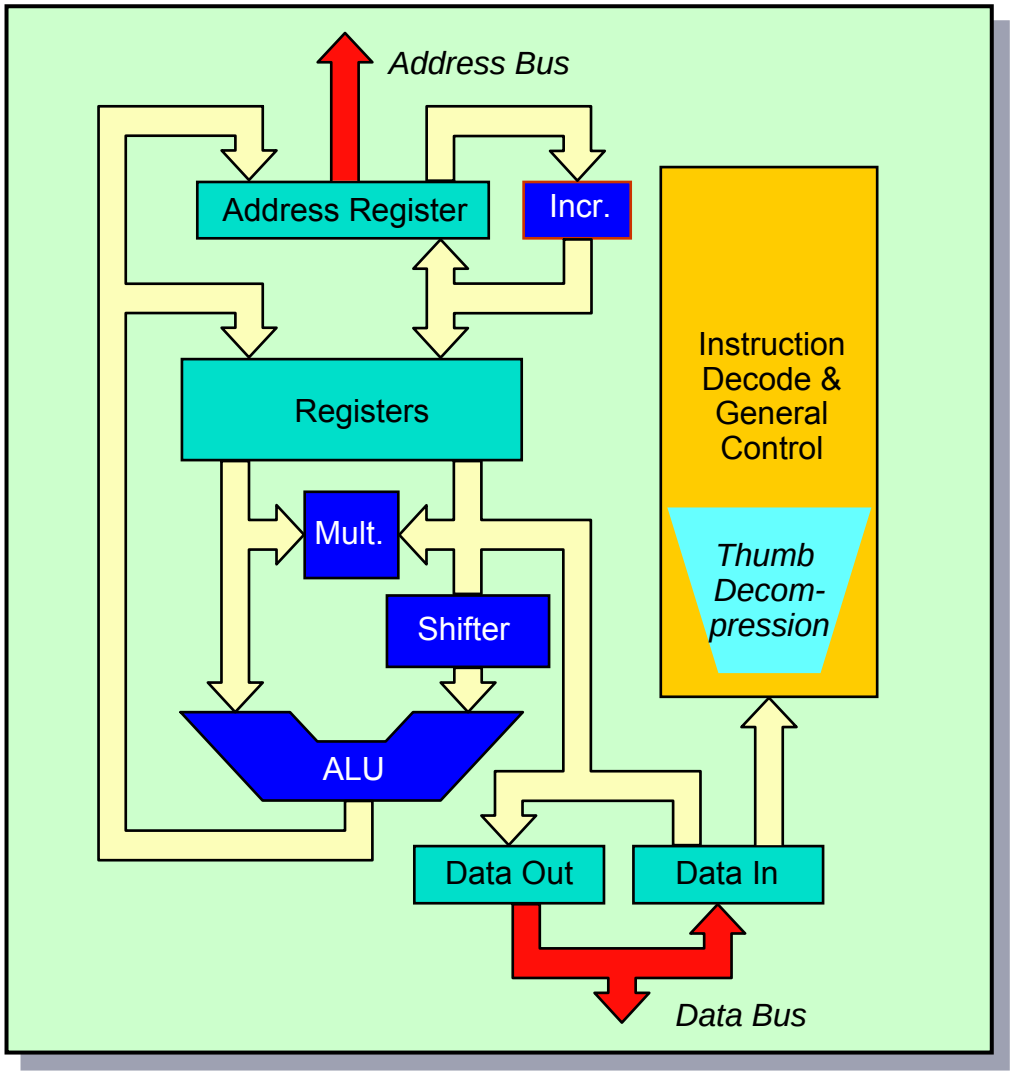
1. ARM - The Company
2. Architecture
3. Instruction Set
4. **ARM7TDMI-S**
5. Systems
6. Other ARM cores
7. NXP Implementation
8. Q & A

ARM7TDMI-S

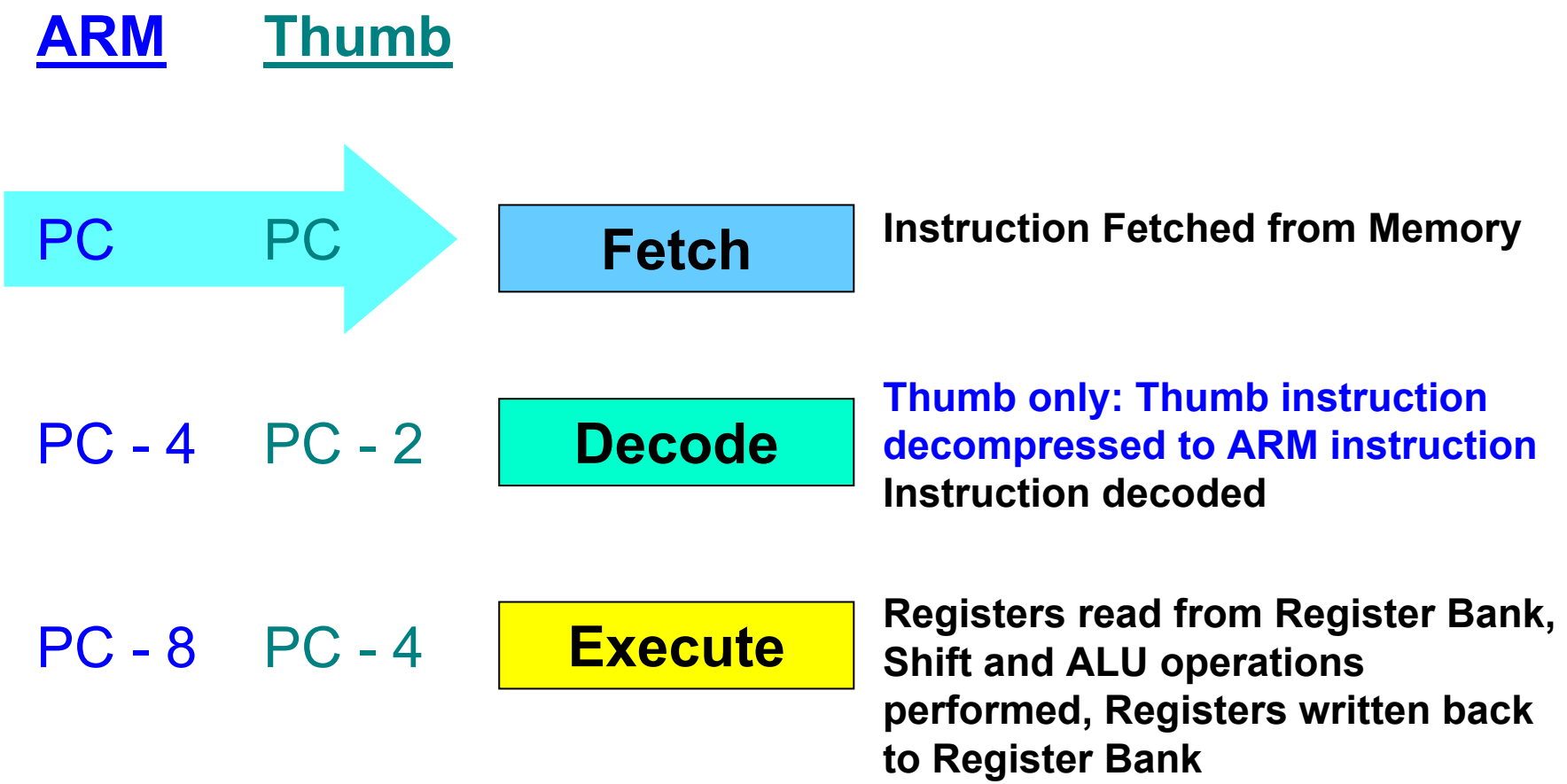
The ARM7TDMI-S is based on ARM7 core

- 3 stage pipeline
- Von Neumann architecture
- CPI ~1.9
- **T:** Thumb instruction set
- **D:** includes debug extensions
- **M:** enhanced multiplier (32x8) with instructions for 64-bit results
- **I:** core has EmbeddedICE logic extensions
- **S:** fully synthesisable (soft IP)

ARM7TDMI-S Core

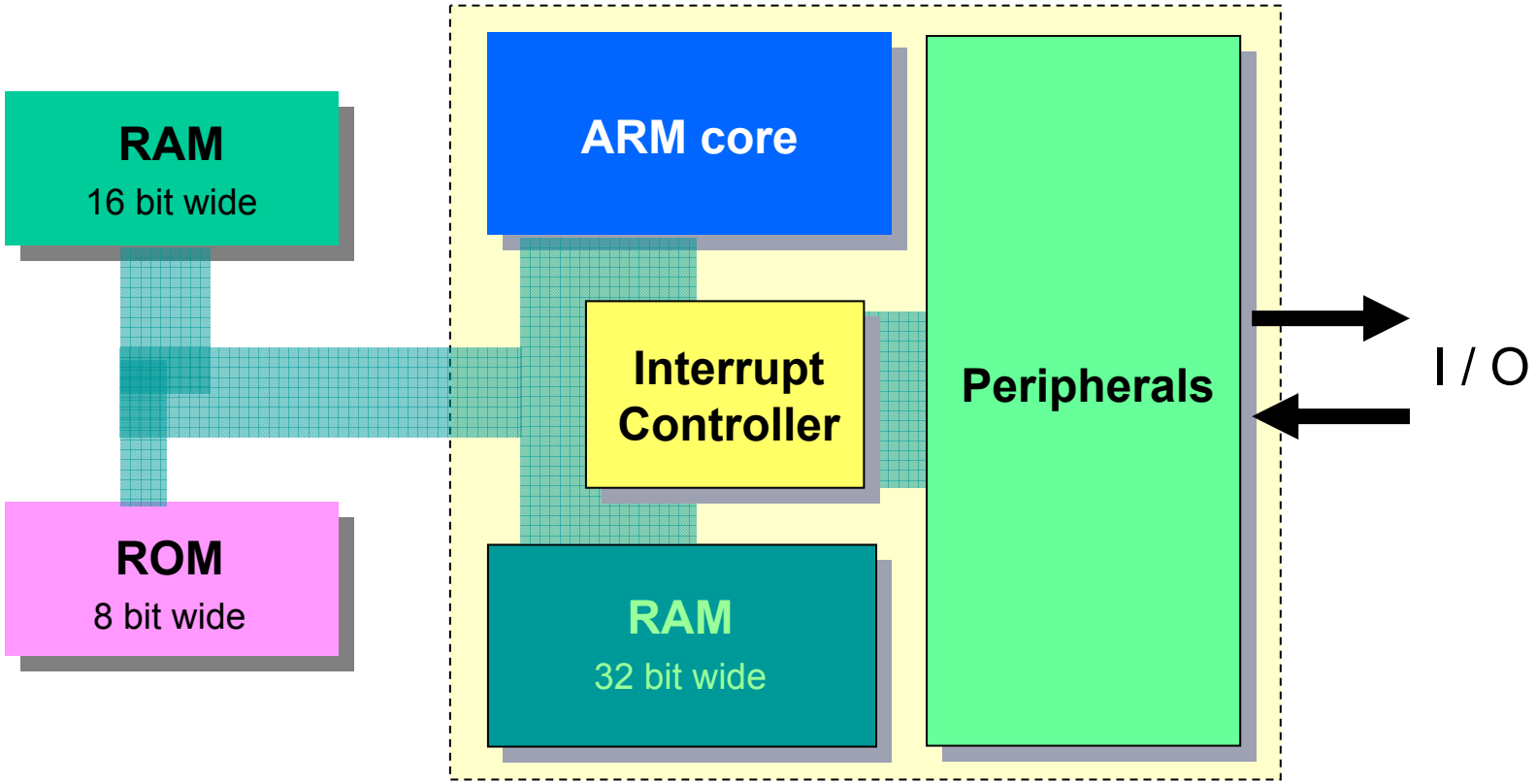


3-Stage Instruction Pipeline



1. ARM - The Company
2. Architecture
3. Instruction Set
4. ARM7TDMI-S
5. **Systems**
6. Other ARM cores
7. NXP Implementation
8. Q & A

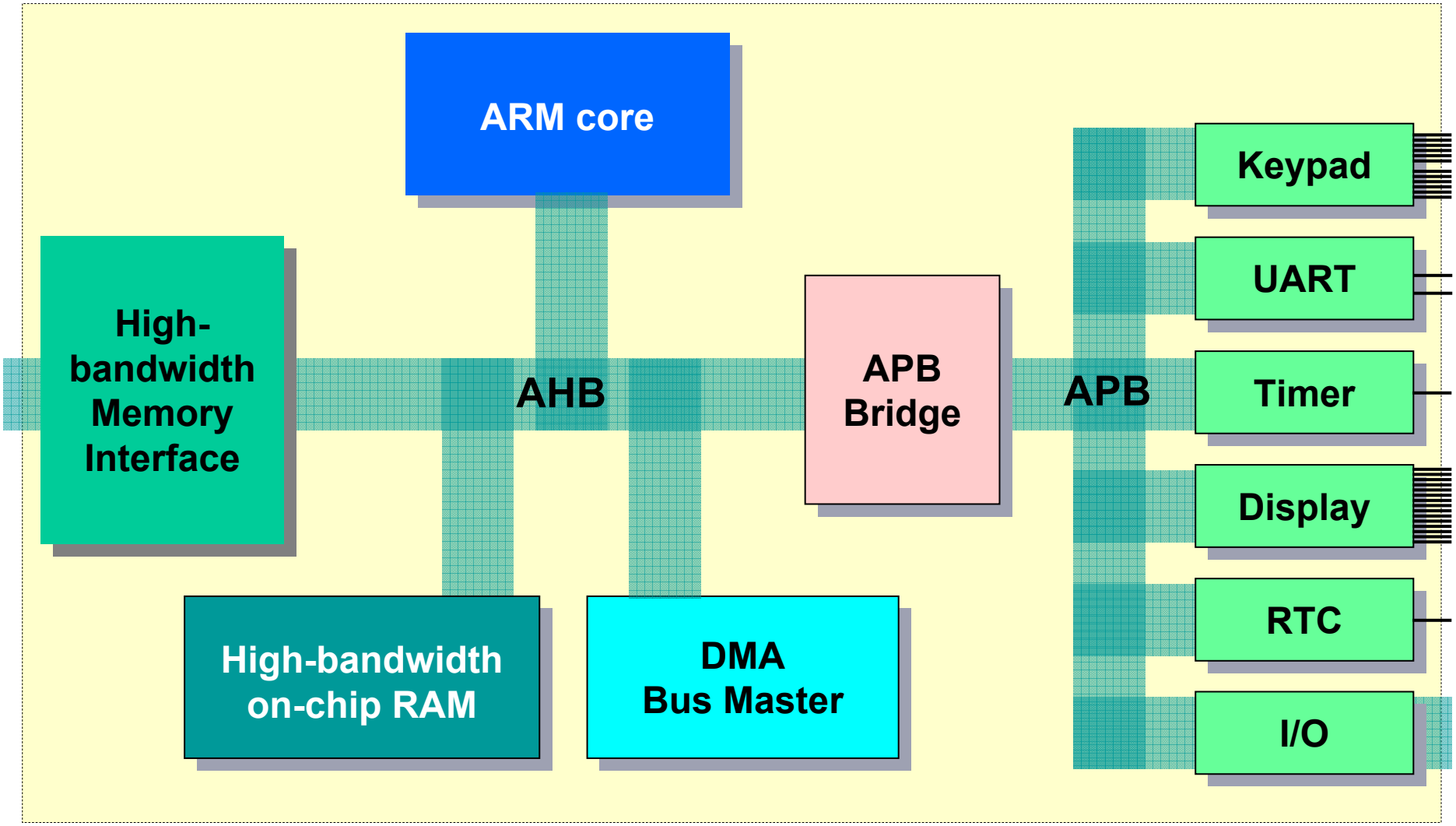
Example ARM based System



AMBA

- **A**dvanced **M**icrocontroller **B**us **A**rchitecture
  - on-chip interconnect
  - established, open specification
  - framework for SoC designs
  - enabler for IP reuse
  - ‘digital glue’ that binds IP cores together

Example AMBA System



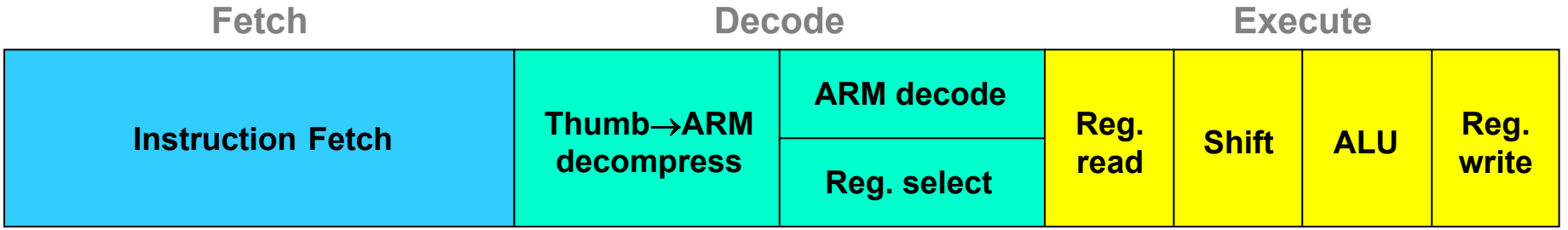
AHB and APB / VPB

- **A**dvanced **H**igh-Performance **B**us
  - high-performance
  - pipelined
  - fully-synchronous backplane
  - multiple bus masters
- **A**dvanced **P**eripheral **B**us / **V**LSI **P**eripheral **B**us
  - low-power
  - non-pipelined
  - simple interface
  - wait support (VPB)

1. ARM - The Company
2. Architecture
3. Instruction Set
4. ARM7TDMI-S
5. Systems
6. **Other ARM cores**
7. NXP Implementation
8. Q & A

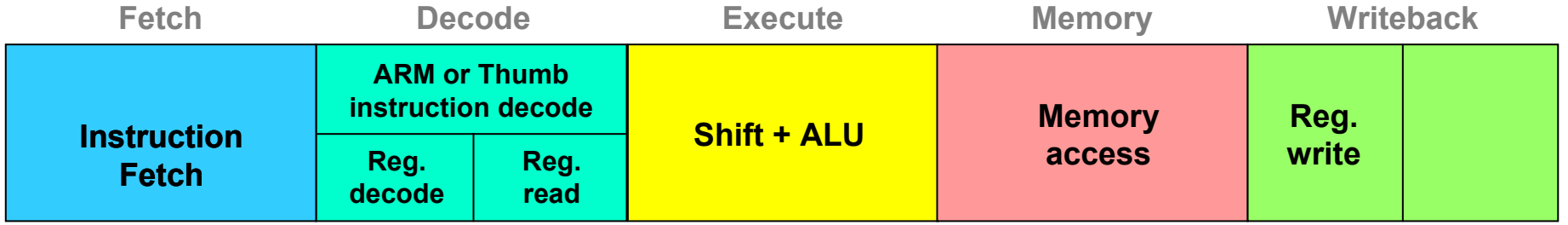
Pipeline-Changes for ARM9TDMI

ARM7TDMI



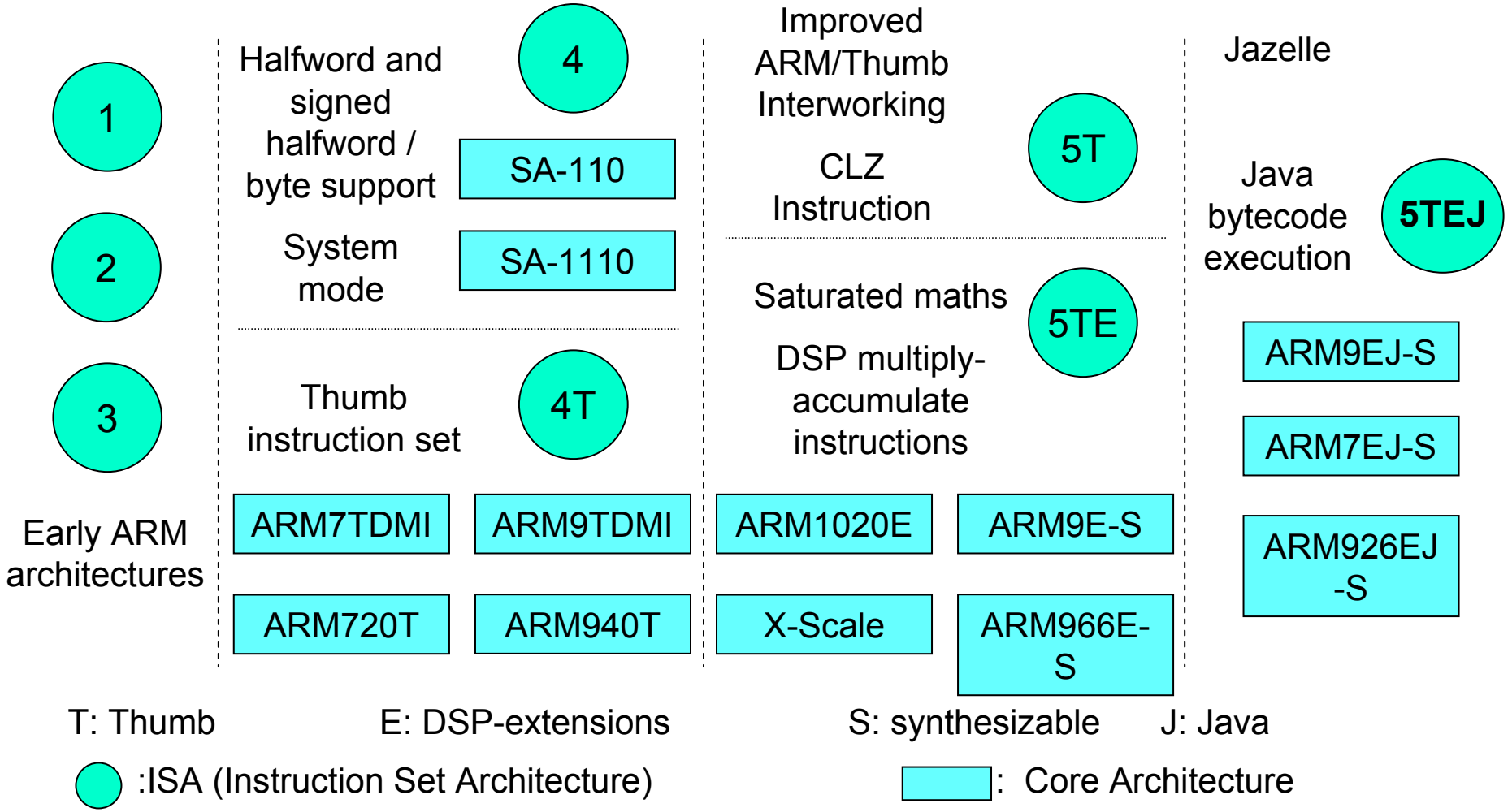
CPI:
~1.9

ARM9TDMI

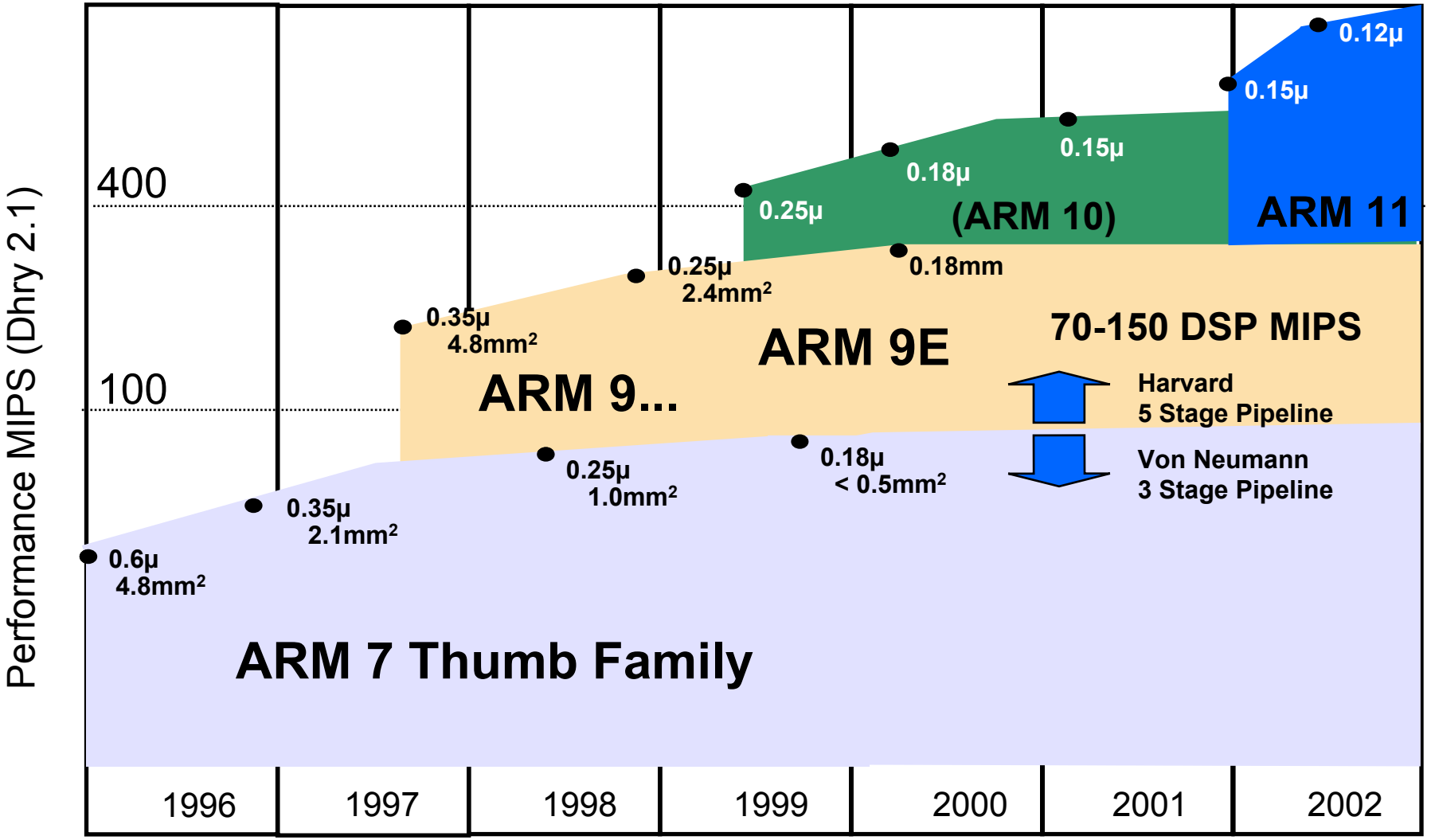


CPI:
~1.5

Development of the ARM Architecture



ARM Technology Roadmap



1. ARM - The Company
2. Architecture
3. Instruction Set
4. ARM7TDMI-S
5. Systems
6. Other ARM cores
7. **NXP Implementation**

LPC2xxx minimum peripherals set

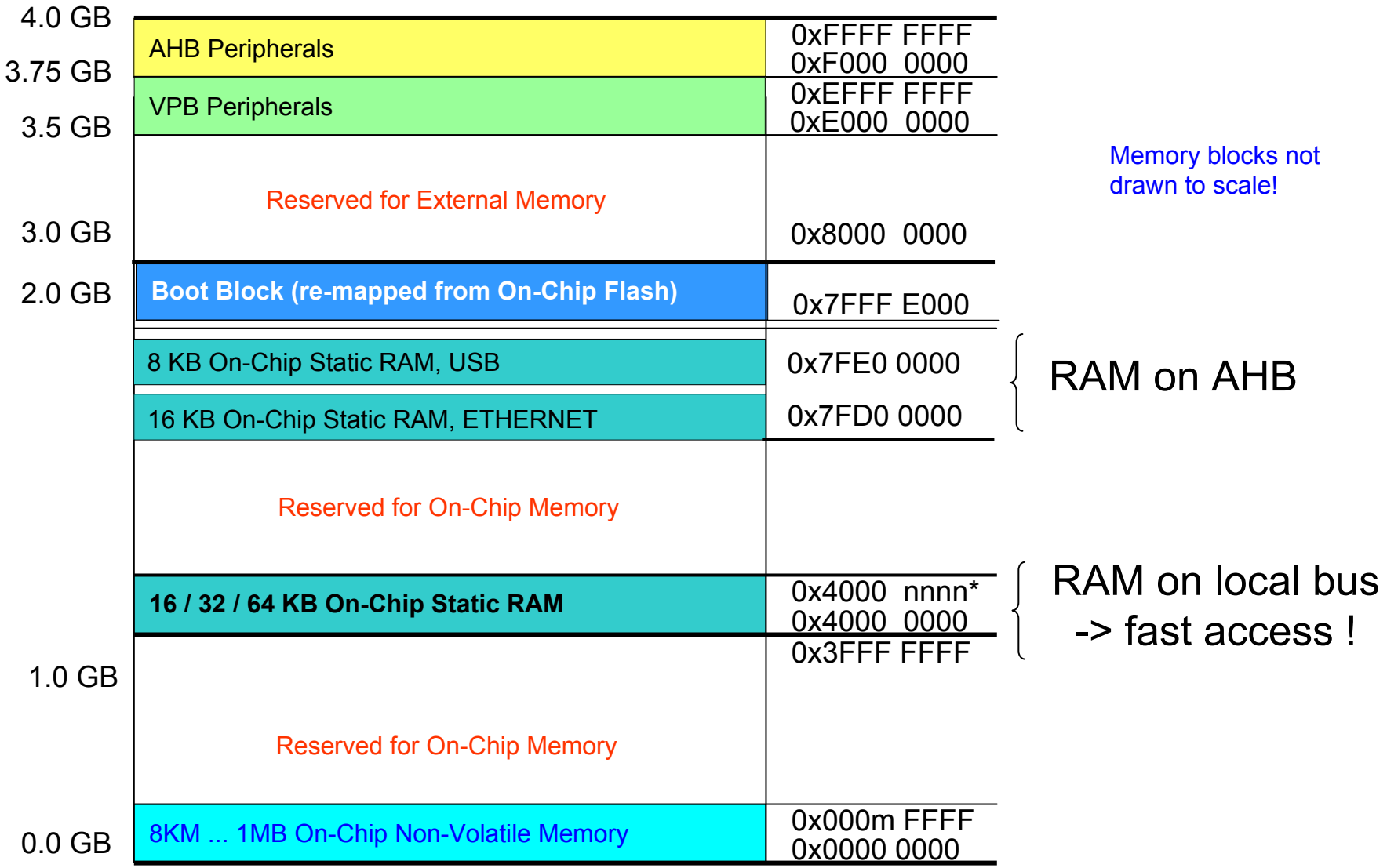
- Some common features
  - ARM7TDMI-S core with E-ICE RTM™ / ETM™
  - Operation up to 60MHz
  - 32-bit timers
    - 2 (4 capture and 4 compare channels each)
    - PWM (6 outputs)
    - RTC
    - Watchdog
  - 2 UARTs (16C550)
  - I²C (400kb/s)

NXP Implementation

1. **Memory Addressing**
2. System Control Block
3. Memory Accelerator Module (MAM)
4. General Purpose I/O / Pin Connect Block
5. Vectored Interrupt Controller
6. Integrated Peripherals

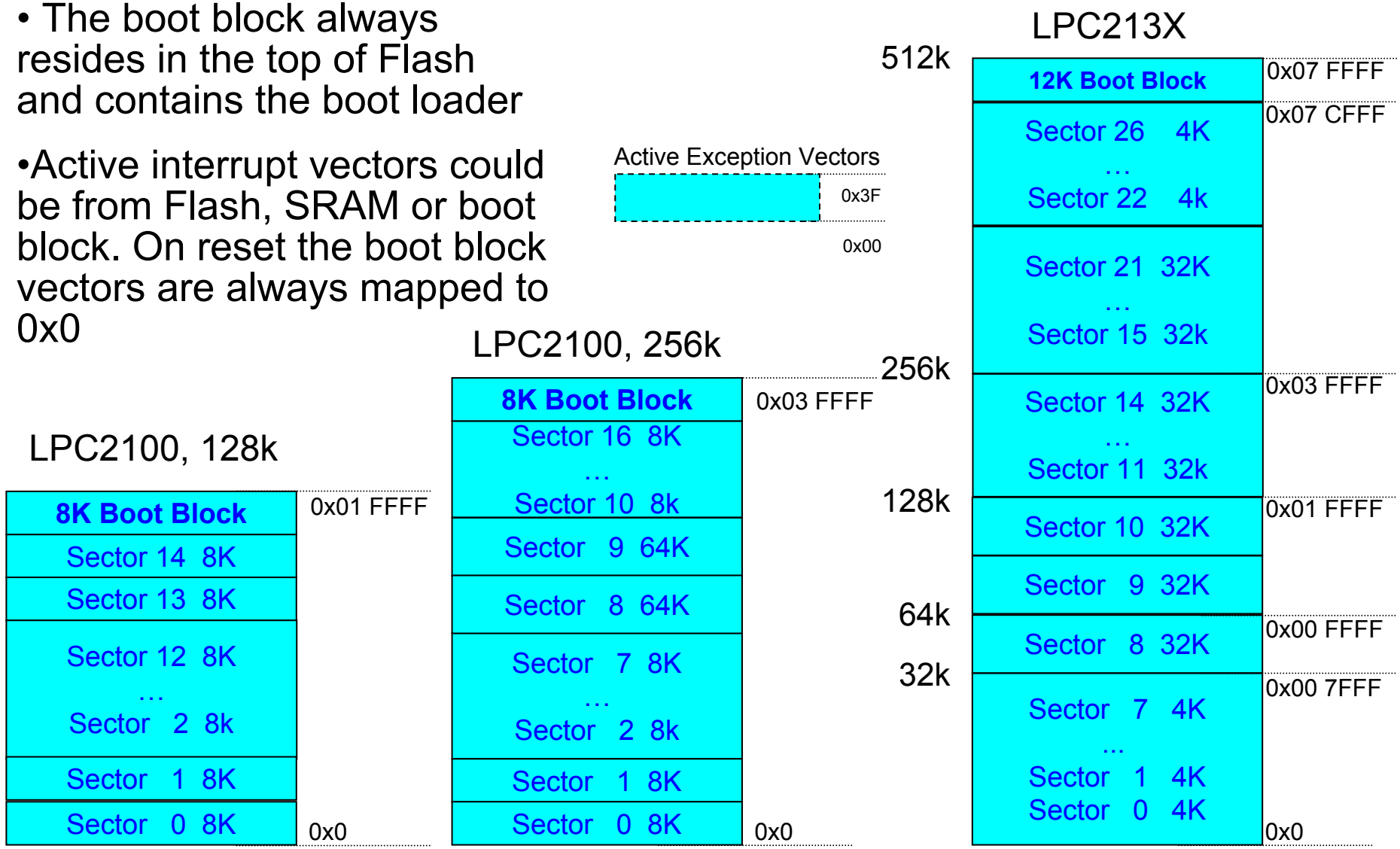
   Timer 0 / Timer 1, UART 0 / UART 1, I²C, SPI, PWM, RTC, Watchdog, ADC, USB, CAN, Ethernet, SD, IIS, GPDMA

LPC2000 Memory Map

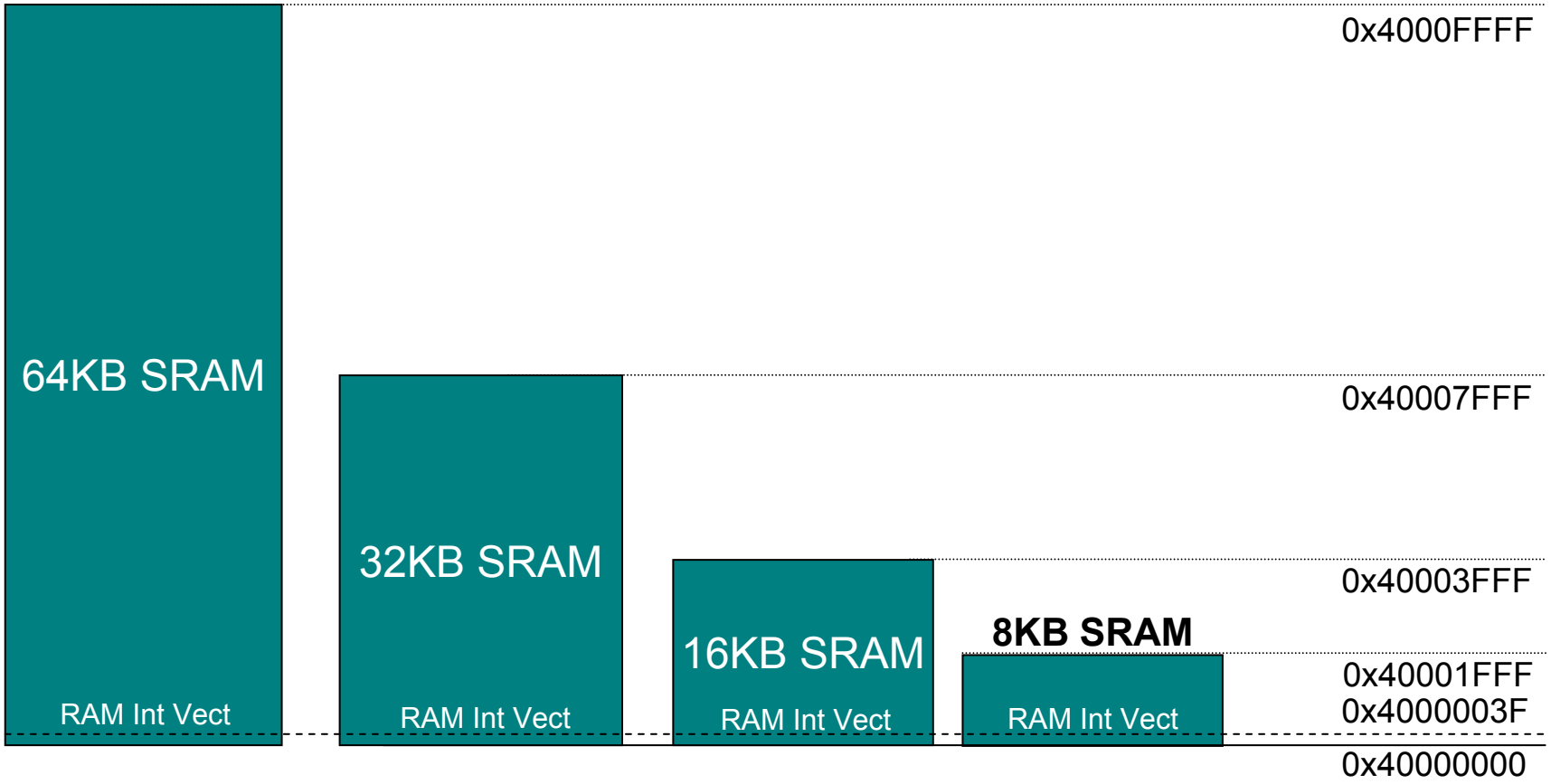


Flash Memory Organization

- The boot block always resides in the top of Flash and contains the boot loader
- Active interrupt vectors could be from Flash, SRAM or boot block. On reset the boot block vectors are always mapped to 0x0



SRAM: 8, 16, 32 or 64 KB



1. Memory Addressing
2. **System Control Block**
3. Memory Accelerator Module (MAM)
4. General Purpose I/O / Pin Connect Block
5. Vectored Interrupt Controller
6. Integrated Peripherals

   Timer 0 / Timer 1, UART 0 / UART 1, I²C, SPI, PWM, RTC, Watchdog, Ethernet, SD, IIS, GPDMA

System Control

- Includes a number of important system features
  - Power Control
  - Memory mapping configuration
  - Oscillator
  - PLL
  - VPB (VLSI Pheriperal Bus) divider
  - Reset (active low)
  - Wakeup Timer
  - External Interrupts

Power Control 1

- Power Control Register [PCON – 0xE01FC0C0] R/W

PCON[0]	IDL	Idle mode - processor clock stopped, on-chip peripherals remain active, interrupts cause wakeup
PCON[1]	PD	Power Down mode - oscillator and on-chip clocks stopped, wakeup by external interrupt

20 uA at room temperature,
50 uA with single voltage supply

For example 5 mA with most peripherals powered down

Biggest factors:
temperature, clock rates
Peripheral Clock Divider: 20%

Power Control 2

- When disabled, peripherals are switched off to conserve power
- Power Control for Peripherals Register [PCONP – 0xE01FC0C4] R/W

PCONP 1	PCTIM0	Enable Timer0
PCONP 2	PCTIM1	Enable Timer1
PCONP 3	PCURT0	Enable UART0
PCONP 4	PCURT1	Enable UART1
PCONP 5	PCPWM0	Enable PWM0
PCONP 7	PCI2C	Enable I2C
PCONP 8	PCSPI	Enable SPI
PCONP 9	PCRTC	Enable RTC
.....		

Each peripheral typically below 1mA

Power Control (3)

- Power Control for Peripherals Register cont'd

...		
PCONP 8	PCSP0	Enable SPI0
PCONP 9	PCRTC	Enable RTC
PCONP 10	PCSPI1	Enable SPI1
PCONP 11	PCEMC	Enable External Memory Controller
PCONP 12	PCAD	Enable A/D-Converter
PCONP 13	PCCAN1	Enable CAN Controller 1
PCONP 14	PCCAN2	Enable CAN Controller 2
PCONP 15	PCCAN3	Enable CAN Controller 3
PCONP 16	PCCAN4	Enable CAN Controller 4

Acceptance Filter enabled with any CAN Controller

CAN peripheral typically below 2mA

Boot Block

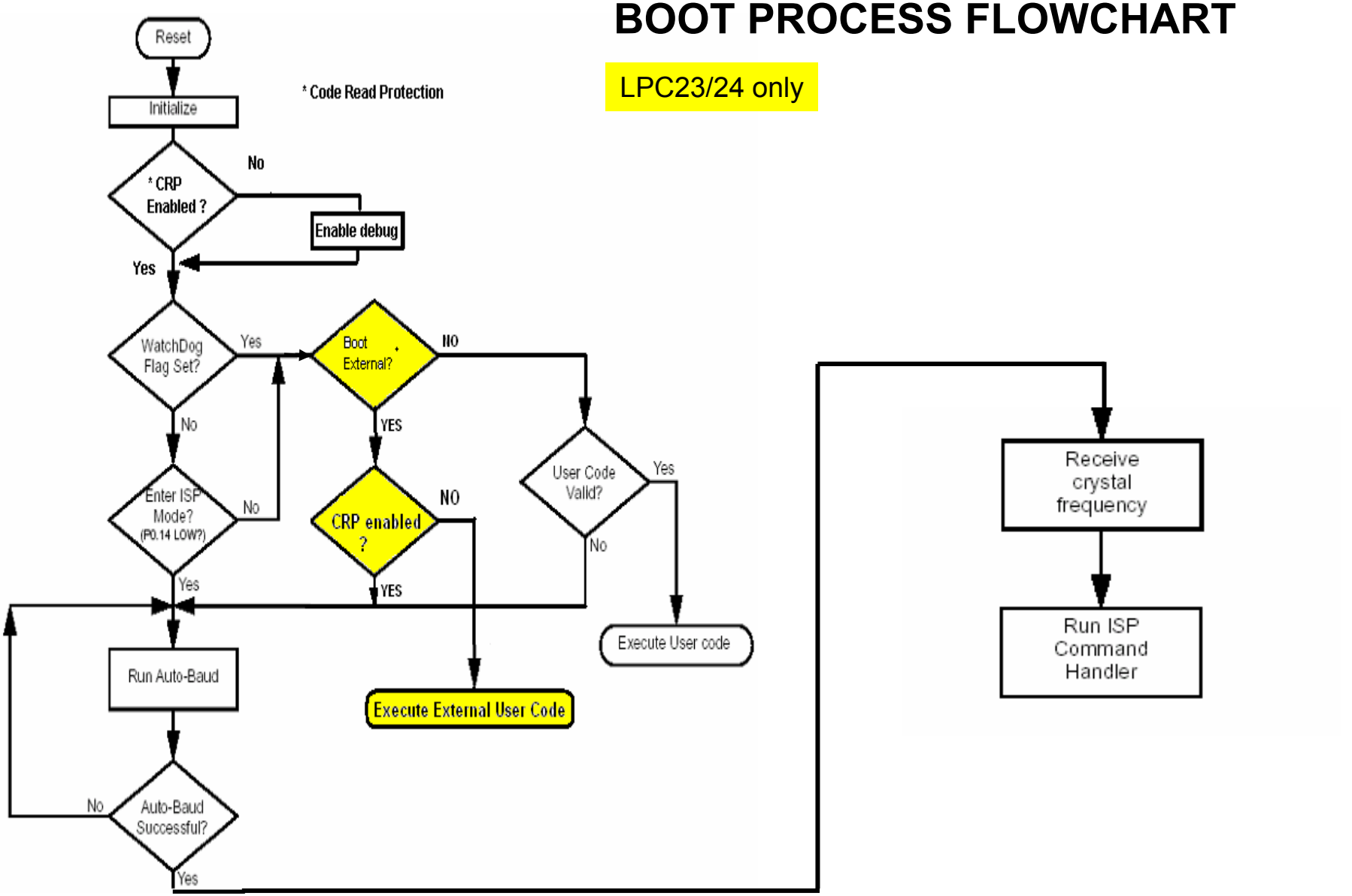
- The uppermost Flash sector contains the Boot Loader
  - controls physical interface for programming and erasing the Flash
  - supports ISP (In System Programming) mode for initial programming of customer code
  - supports In-Application Programming in a running system under the control of customer software
  - buffers an entire Flash line (512 bytes) at once to keep programming time to a minimum
- The Boot Loader is automatically run following reset
  - checks for a “Valid User Program” key to prevent running code on incorrectly programmed devices

Flash Memory IAP Programming

- IAP: In Application Programming
- Bootloader contains flash programming routines
  - Erase Sectors
  - Write blocks (of 512 bytes)
    - Re-write to blocks possible, if bits are cleared in 32 byte groups (see hands-on example)
  - Common entry point for IAP calls: `0x7fffffff1`
- During calls, all interrupts must be disabled
  - Note: hands-on example only disabled IRQ, not FIQ

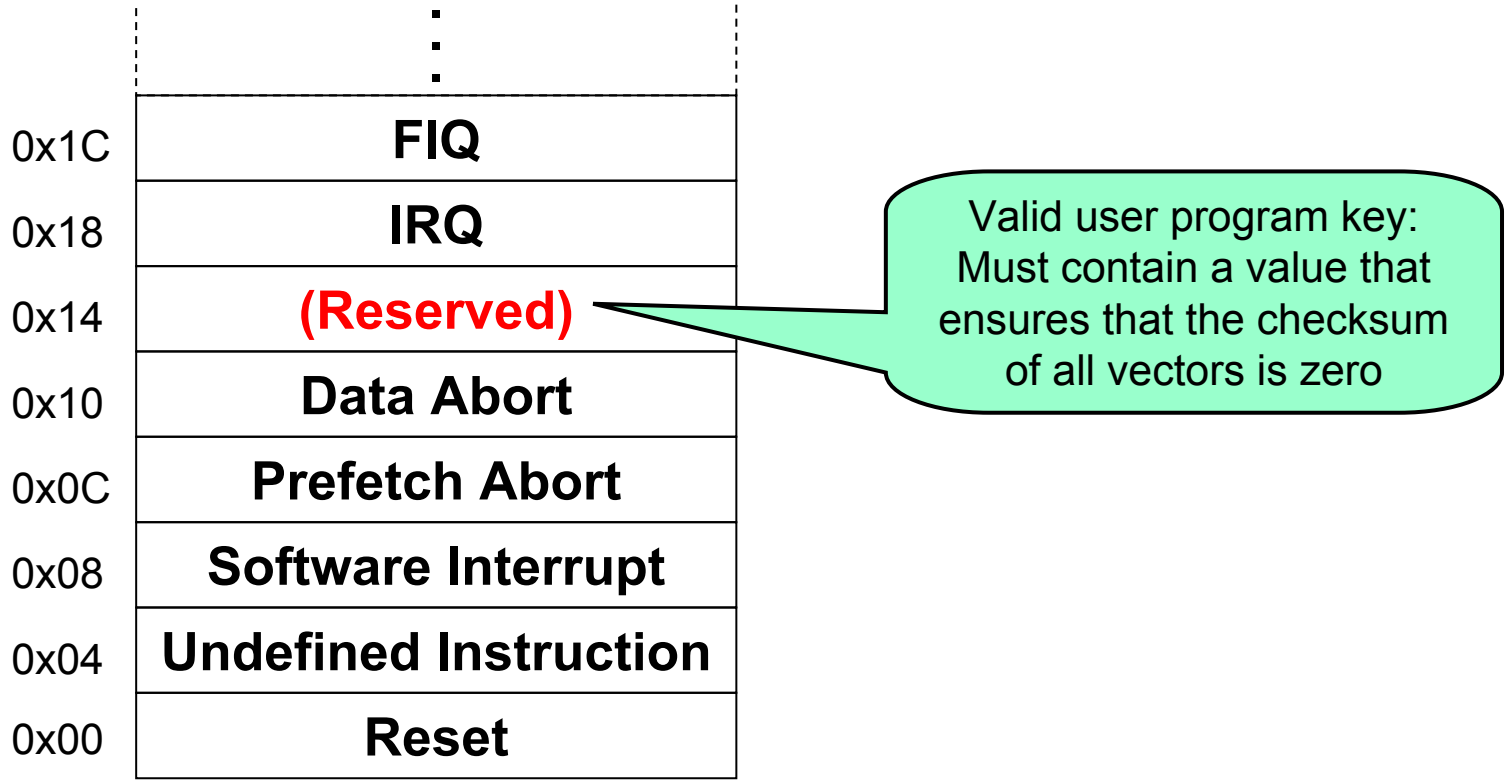
BOOT PROCESS FLOWCHART

LPC23/24 only



Exception Vectors

- Vector Table

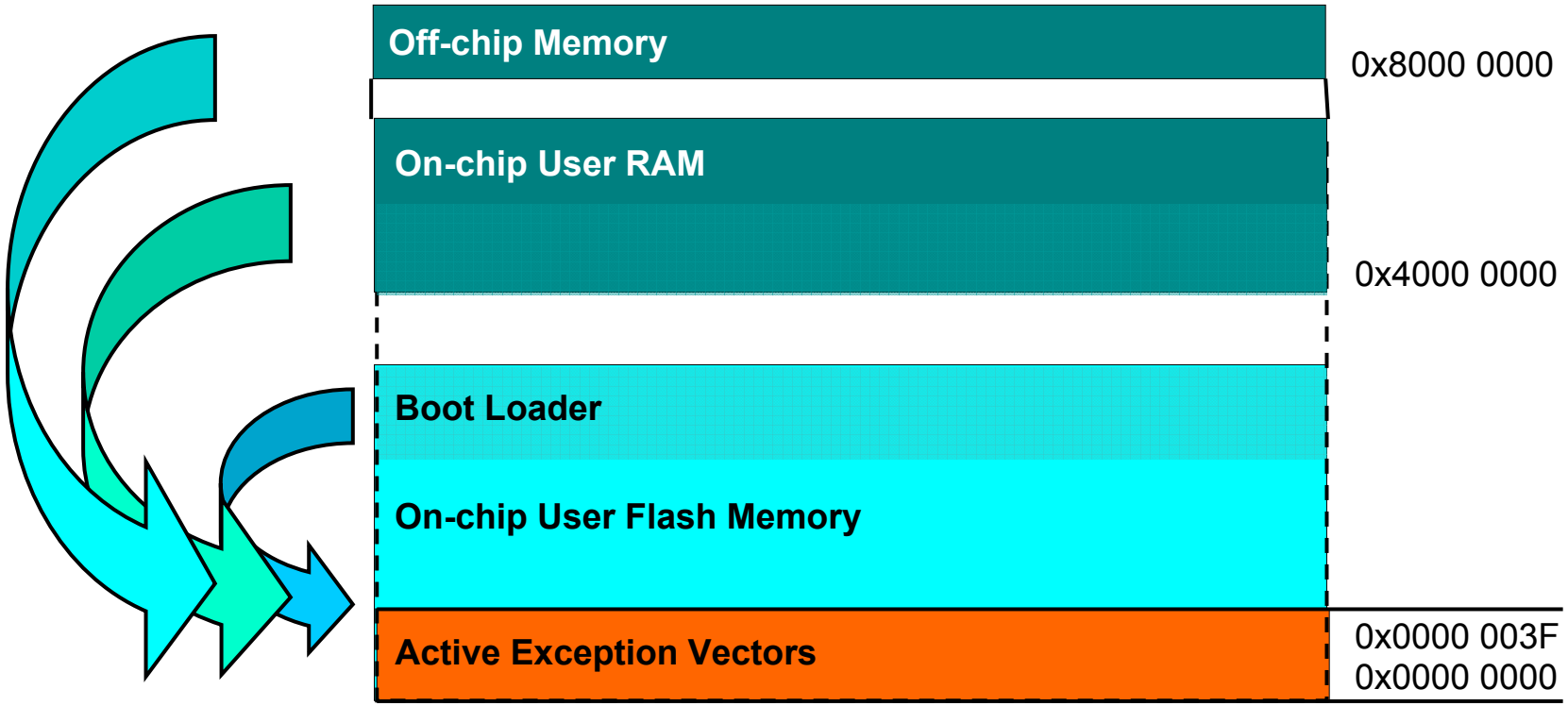


Memory Mapping Control 1

- Re-mapping of **Exception Vectors**
  - always appear to begin at 0x0000 0000
  - but can be mapped from different sources:
    - User Flash
      - **Exception Vectors** are **not** re-mapped and reside in Flash

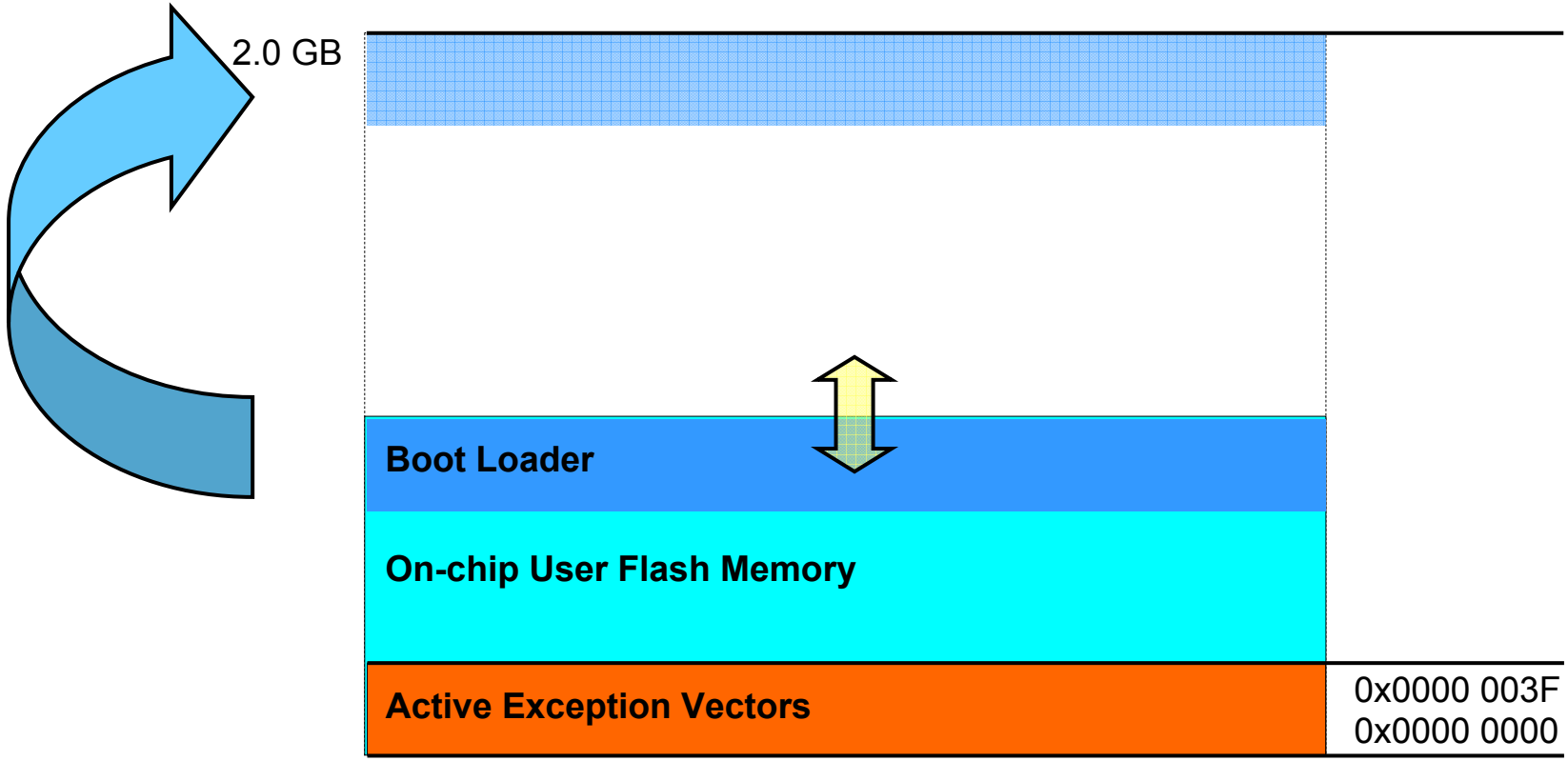
Memory Mapping Control (2)

- Boot Loader
  - Always executed after reset. **Exception Vectors** re-mapped from Boot Block
- User RAM
  - **Exception Vectors** are re-mapped from RAM

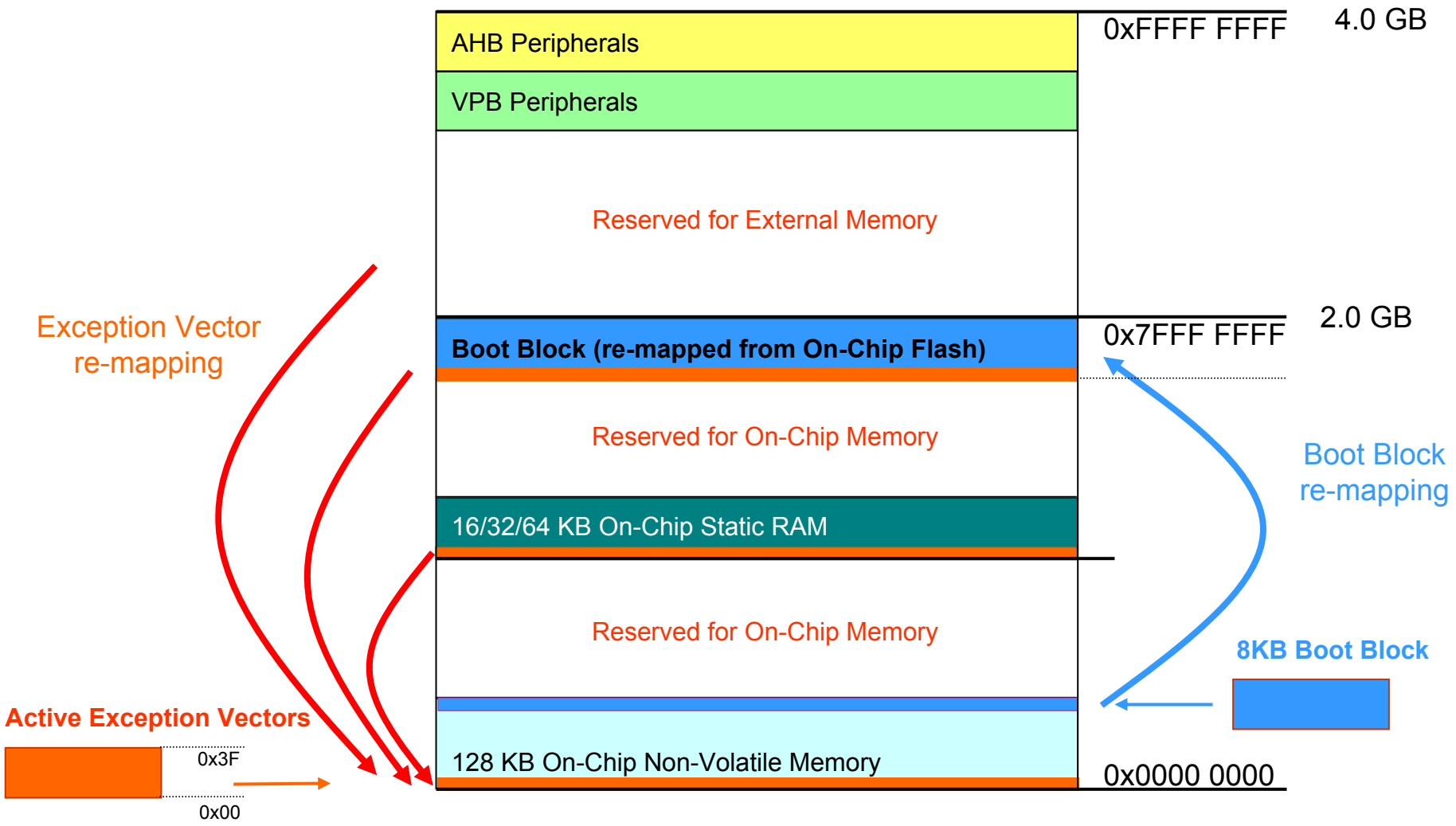


Memory Mapping Control (3)

- Re-mapping of **Boot Block**
  - mapped from top of Flash to top of on-chip memory space



Memory Mapping Overview



Memory Mapping Control Register

- Memory Mapping Control [MEMMAP – 0xE01FC040] R/W

MEMMAP 1:0	MAP 1:0	00: Boot Loader Mode 01: User Flash Mode (no re-mapping) 10: User RAM Mode 11: External Memory

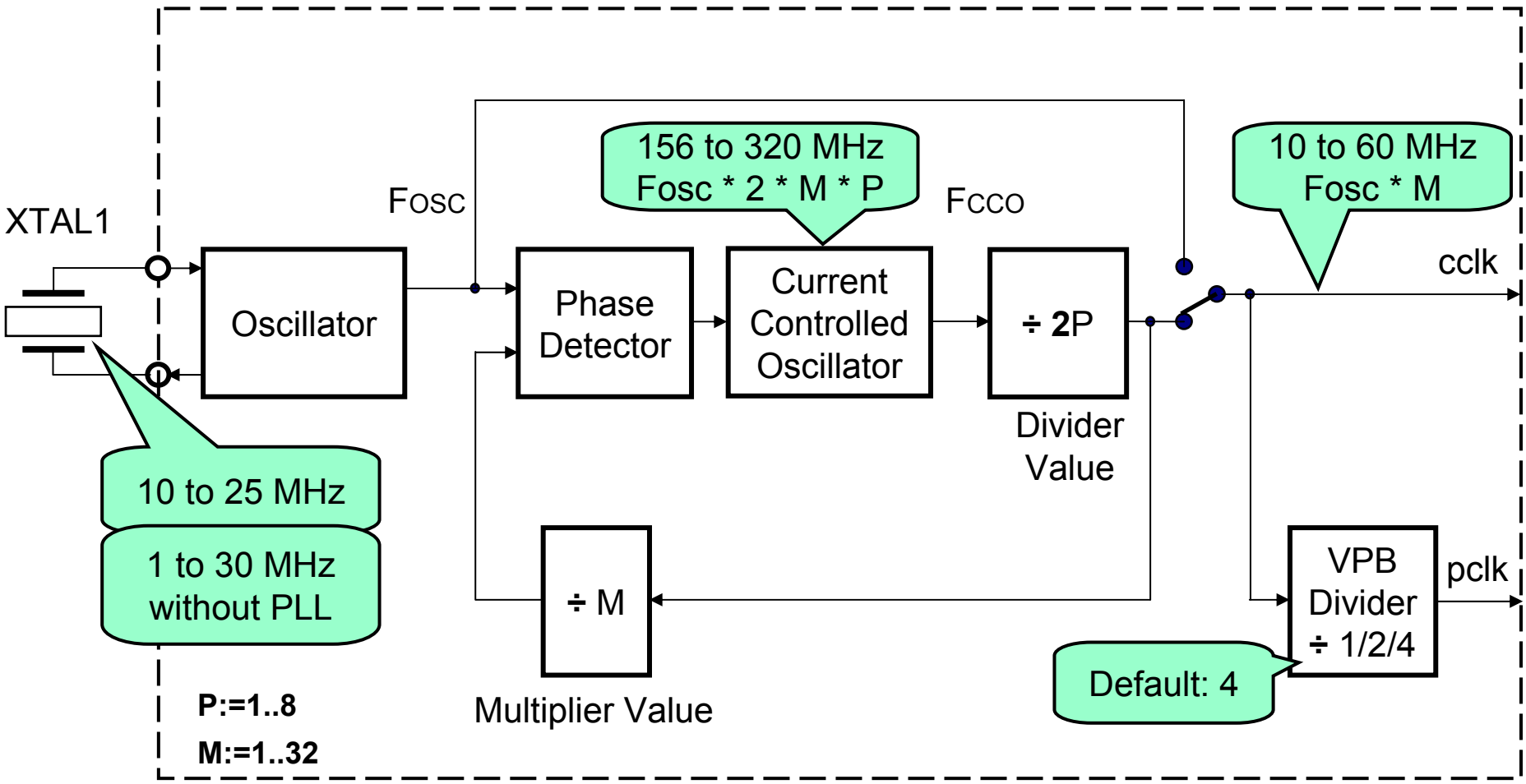
Selects the memory being mapped to address zero

Phase Locked Loop (1)

- 10 to 25 MHz input clock frequency
- Output frequency from 10 MHz up to the max. CPU rate

  (LPC2xxx: 60MHz)
- Programmable frequency multiplication
- PLL bypassed on reset
- PLL lock indicator can be used as an interrupt to connect the PLL once it is locked
- PLL programming requires a special feed sequence (like the watchdog) for safety

Phase Locked Loop (for old families LPC21xx and LPC22xx)



Phase Locked Loop (3)

cclk:	processor clock frequency	10 ... 60-70	MHz
Fosc:	crystal oscillator or input frequency	10 ... 25	MHz
Fcco:	PLL CCO-frequency	156 ... 320	MHz

$cclk = M \bullet Fosc$	$cclk = \frac{Fcco}{2 \bullet P}$
$Fcco = cclk \bullet 2 \bullet P$	$Fcco = Fosc \bullet M \bullet 2 \bullet P$

Example: Fosc = 10 MHz (crystal controlled) Target: cclk = 60 MHz

$$M = \frac{cclk}{Fosc} = \frac{60\ MHz}{10\ MHz} = 6$$

Select suitable value for P: Fcco = 60 MHz • 2 • P = 240 MHz (P = 2)

PLL Registers (1)

- PLL Control Register [PLLCON – 0xE01FC080] R/W

PLLCON0	PLLE	PLL Enable
PLLCON1	PLLC	PLL Connect

- PLL Configuration Register [PLLCFG – 0xE01FC084] R/W

PLLCFG 4:0	MSEL 4:0	PLL Multiplier value "M" (M-1)
PLLCFG 6:5	PSEL 1:0	PLL Divider value "P" 00: 1 01: 2 10: 4 11: 8

PLL Registers (2)

- PLL Feed Register [PLLFEED – 0xE01FC08C] WO

PLLFEED 7:0	PLLFEED	Consecutive writes 0xAA then 0x55

- PLL Status Register [PLLSTAT – 0xE01FC088] RO

PLLSTAT 4:0	MSEL 4:0	Readback for Multplier value "M" (M-1)
PLLSTAT 1:0	PSEL 1:0	Readback for Divider value "P" (1,2,4,8)
PLLSTAT 8	PLLE	Readback for PLL Enable bit
PLLSTAT 9	PLLC	Readback for PLL Connect bit *
PLLSTAT10	PLOCK	Reflects PLL Lock status

* Cleared when Power Down mode activated

VPB Divider Register

- VPB Divider Register [VPBDIV – 0xE01FC100] R/W

VPBDIV 1:0	VPBDIV	cclk / pclk ratio	00: 4 01: 1 10: 2

1. Memory Addressing
2. System Control Block
3. **Memory Accelerator Module (MAM)**
4. General Purpose I/O / Pin Connect Block

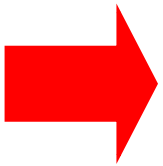
   Pin Connect Block / External Memory Controller
5. Vectored Interrupt Controller
6. Integrated Peripherals

   Timer 0 / Timer 1, UART 0 / UART 1, I²C, SPI, PWM, RTC, Watchdog, ADC, CAN

On-chip Memory with 0-wait States

- ARM7TDMI-S is a 1-clock core
  - CPI of ~1.9, but many instructions execute in 1 cycle
  - CPU requires one instruction per clock cycle

For **highest performance**



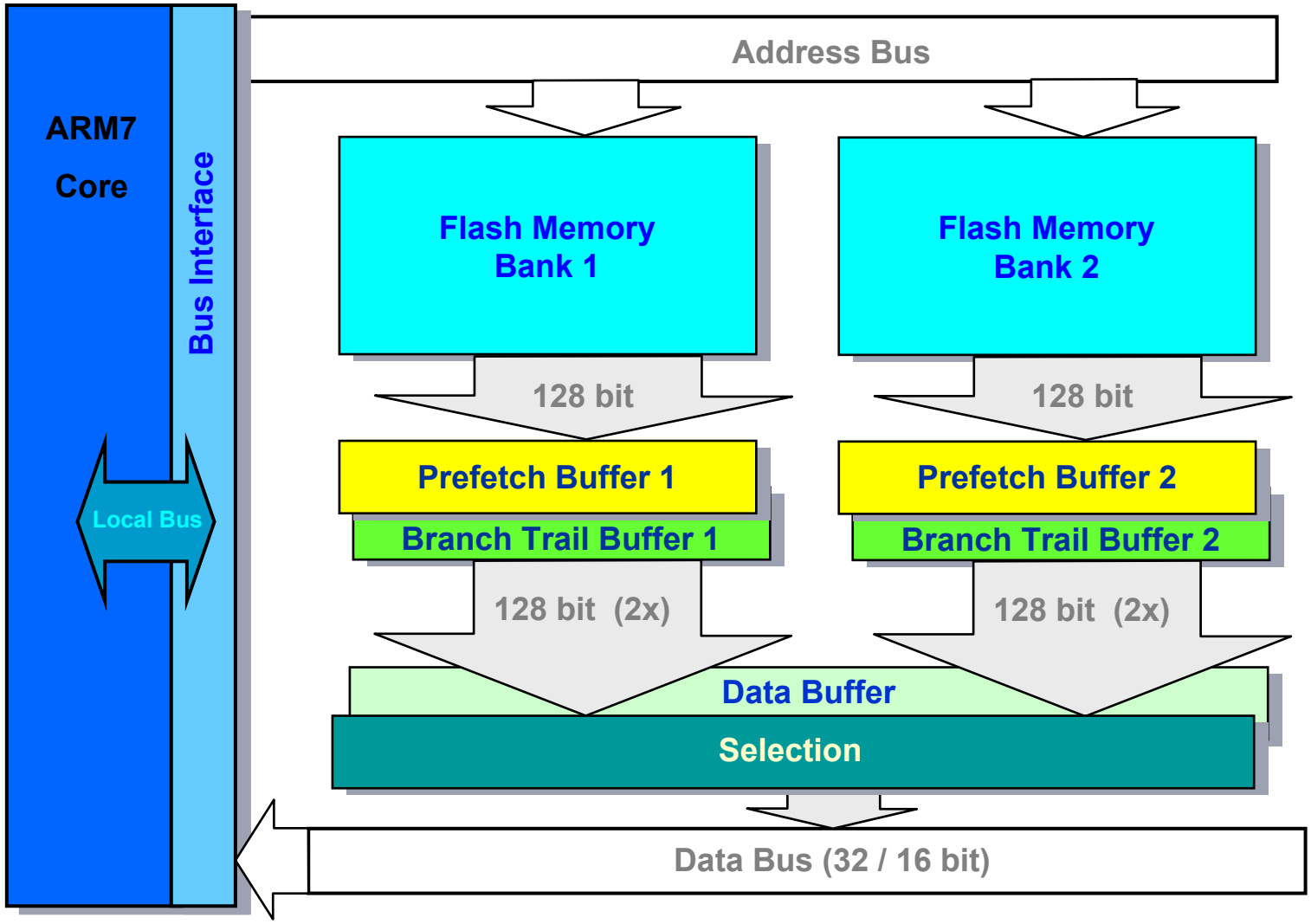
- 32 bits needed with every clock
- Memory access time **<** 17ns @ 60MHz

Memory Accelerator Module

- CPU performance limited by Flash access time (<50 ns)
  - ARM needs 32 bits every clock for 1 clock instructions
  - Without MAM operation possible up to 20MHz
- Flash Architecture
  - Flash is split into two blocks of 128 bits each
    - 128 bits are accessed at once
  - Separate Branch Trail Buffer holds 128 bits of history of each block

  => 4-times speed improvement for linear code; branches will cause a variable delay, depending on target address
  - Additional Data Buffer for data accesses to Flash

Memory Accelerator Module



MAM Registers

- MAM Control Register [MAMCR – 0xE01FC000] R/W

MAMCR 1:0	MAM mode control	00: MAM functions disabled 01: MAM functions partially enabled * 10: MAM functions fully enabled 11: reserved

* Only sequencial code via MAM

- MAM Timing Register [MAMTIM – 0xE01FC004] R/W

MAMTIM 2:0	MAM Fetch Cycle timing	000: reserved 001: MAM fetch is 1 clock cycle ... 111: MAM fetch is 7 clock cycles

Hands-On Index

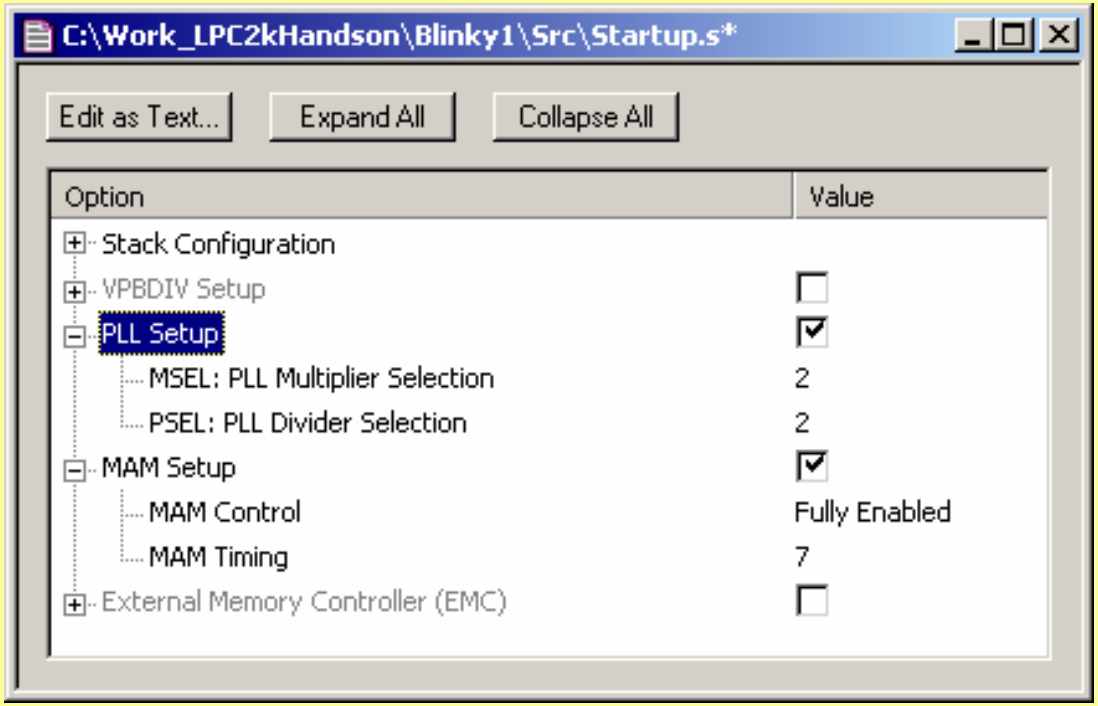
Hands-On

1. Tools Setup
2. PLL / MAM / GP I/O / Flash
3. ADC
4. Interrupts / Timer
5. UART / CAN

PLL and MAM Setup with Keil uVision, Project Blinky1

Hands-On

- The PLL and MAM settings are part of the startup code
- Settings can be made using a convenient input mask
- Exercise: Setup the PLL to 60Mhz



MAM Setup

Hands-On

- Enabling the MAM for 60Mhz clock and 50ns Flash
- Setup MAM
- “Fully Enable”
- MAM Timing “3”
  3 * 17ns = 51ns

- Explanation of MAM Register values
  - “Partially Enable”: Code fetches only
  - “Fully enable”: Code and data fetches
  - MAM Timing Number of clock cycles used for one flash memory access

Verify
PLL and MAM Settings

Hands-On

Open disassembly window and single step through while loop in function “wait”

Every loop needs 6 cycles:
- 3 instructions at 1 CPI
- 1 instruction (branch) at 3 CPI

```
ADD         R2,#0x01
SUB         R3,#0x01
CMP         R3,#0x00
BNE         0x0000014E
```

- In Simulation == In Hardware if MAM is full enabled
  - If the define LOOP_DELAY is 1.000.000
  - 6.000.000 cycles are realized in the delay()
  - “action” incremented every 100ms at 60Mhz

Flash IAP

Hands-On

- Project FlashSWInt
- Load and single step
  - Watch memory at 0x8000, ASCII mode

- Exercise:
  Change program to use address 0x10200 for storing the string

1. Memory Addressing
2. System Control Block
3. Memory Accelerator Module (MAM)
4. **General Purpose I/O / Pin Connect Block**
5. Vectored Interrupt Controller
6. Integrated Peripherals

   Timer 0 / Timer 1, UART 0 / UART 1, I²C, SPI, PWM, RTC, Watchdog, ADC, CAN

General Purpose I/O (1)

- Pins available for GPIO:
- LPC21/22
  - 48-pin devices: 32
  - 64-pin devices: 46
  - 144 pin devices: 76 (max.) (with external memory)
    112 (w/o external memory)
- LPC23/24
  - Up to 160 GPIO pins, all implemented as fast GPIOs, with 64 GPIO interrupts (plus 4 other external interrupts).
- Shared with
  - Alternate functions of all peripherals
  - Data/address bus and strobe signals for external memories

General Purpose I/O (2)

- Direction control of individual bits
- Separate set and clear registers
- Pin value and output register can be read separately
- Slew rate controlled outputs (10 ns)
- 5 registers used to control I/Os

General Purpose I/O 2

Register

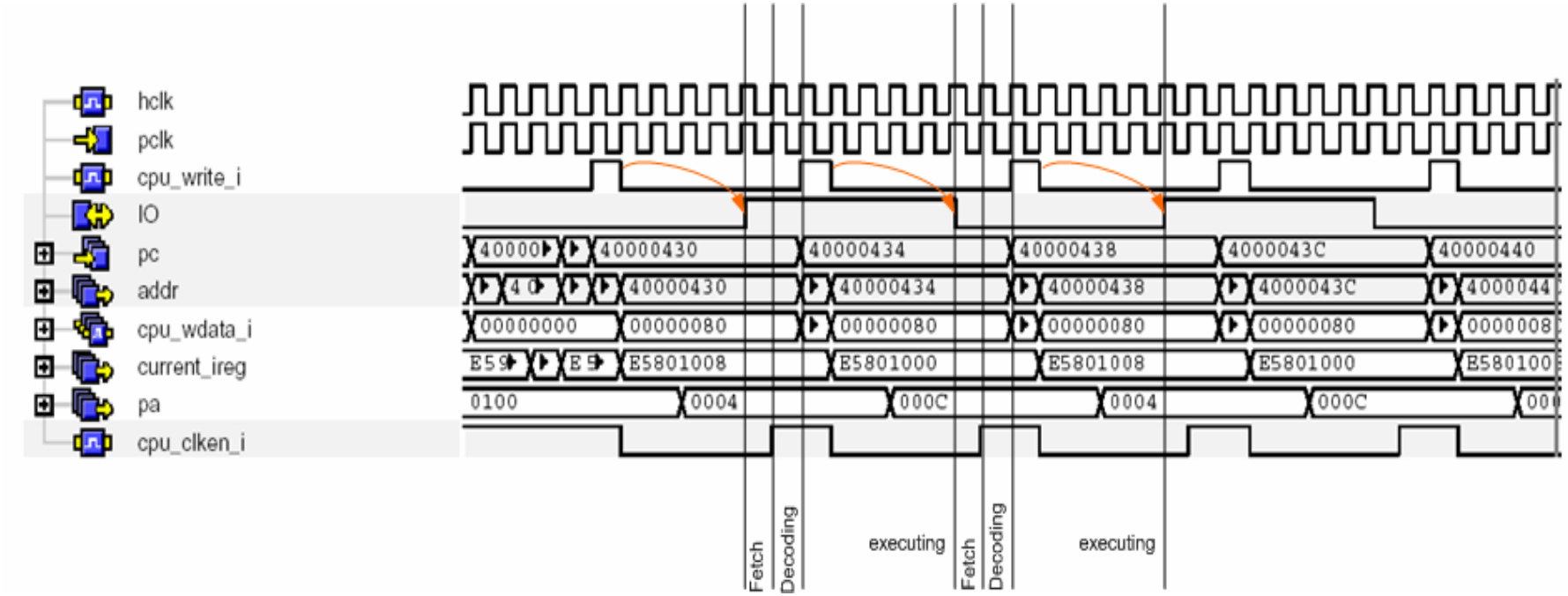
IOPIN	The current state of the port pins is read from this register
IOSET	Writing "1" sets pins **high**, writing "0" has no effect
IOCLR	Writing "1" sets pins **low** and clears corresponding bits in IOSET
IODIR	Port pin direction: 0 = INPUT 1 = OUTPUT
PINSEL0/1	Selects function of pins (Pin Connect Block)

Conventional GPIO Implementation Drawbacks

- Conventional ARM GPIO is implemented on the APB peripheral bus
- Toggling speed of the GPIO is limited due to the 3-stage pipeline, AHB bridge and the APB bus

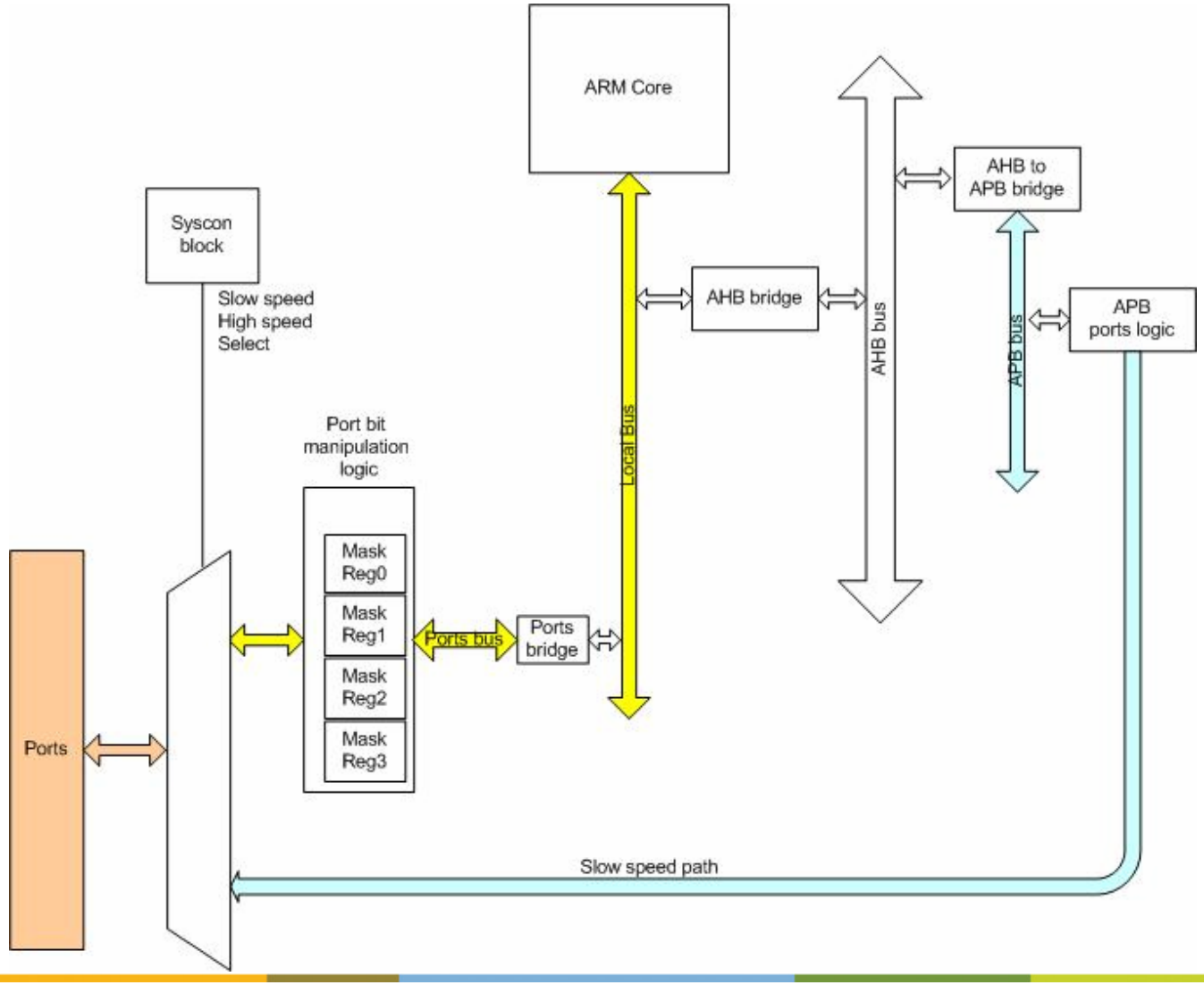
Understanding the slow port behavior

- It takes 5 clocks to execute a port write
- Total time from instruction fetch to port change is 7 clocks
- Maximum achievable period is 14 clocks (cclk/14) = (60/14) = 4.28MHz



When cpu_clken_I is low indicates core is stalled

Block diagram of new port configuration



Improving Speed

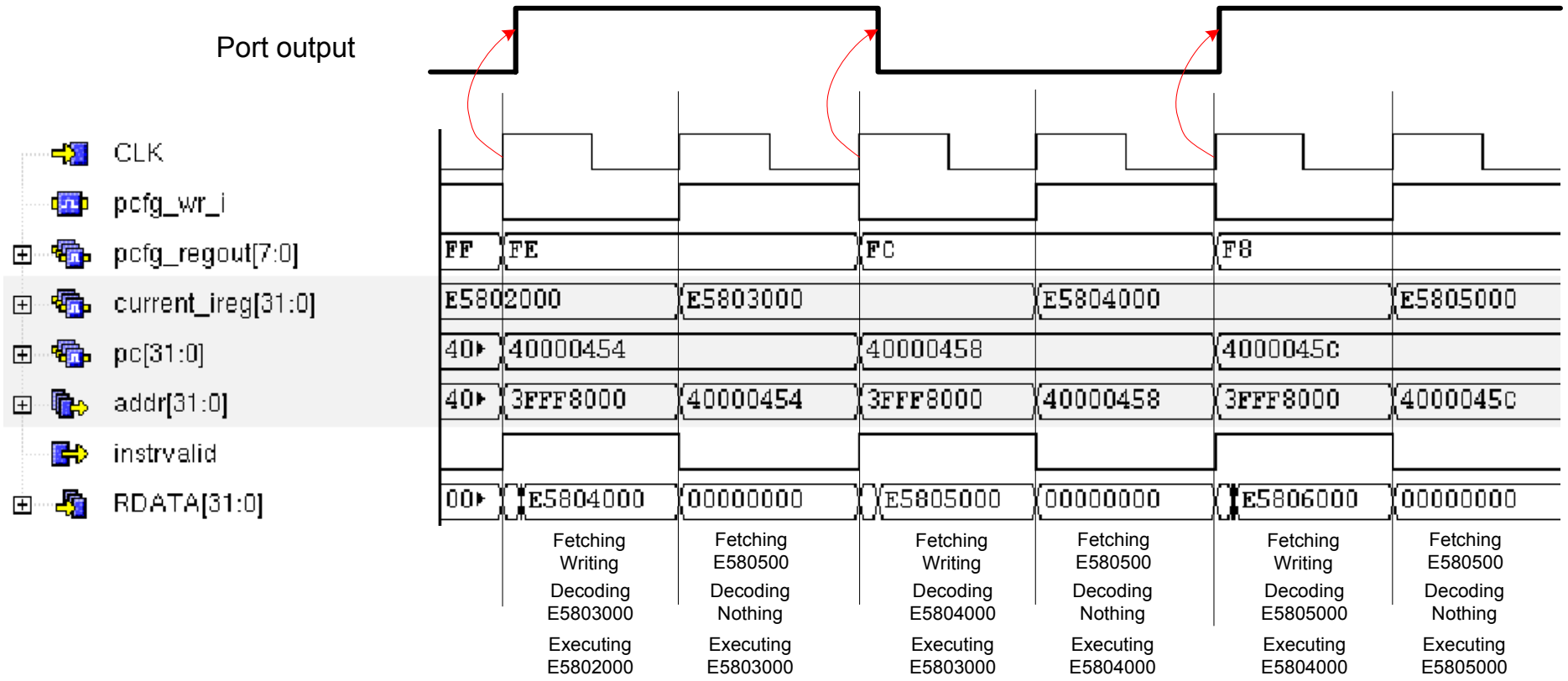
- GPIO registers are now interfaced directly to the ARM7 Local bus

- Ports can now toggle every 2 clocks giving a clock period of cclk/4= 15Mhz

This is a 3.5x speed increase

Enables faster ‘soft’ peripherals

Results of change (speed of writes)



Fast GPIO

- Special features
  - GPIO registers accessed via ARM local bus in addition to conventional peripheral bus access
  - Mask registers allow treating sets of port pins as a group, leaving other bits unchanged
  - Local bus GPIO registers are now byte addressable
  - Entire port value can be written in one instruction using the IOPIN register

Mask Register- Advantages

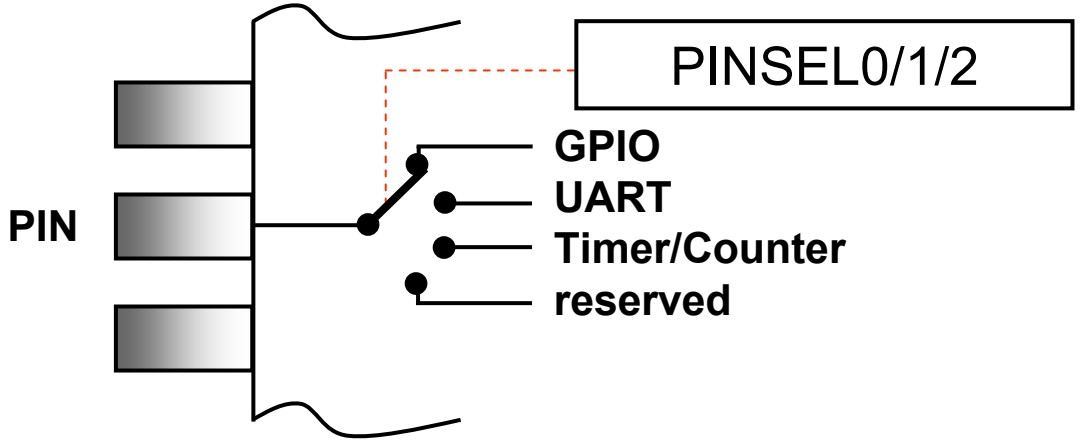
- Provides the capability to the user to separate the GPIO pins into groups
- Any modifications to the FIOSET,FIOCLR and FIOPIN is only effected if the corresponding bits in the FIOMASK are set
- Using Mask registers…

  Individual I/O pins can be addressed separately

Pin Connect Block (1)

- Many on-chip functions can use I/O pins
- Number of I/O-pins is limited

  ⇒ I/Os can be configured to adapt various functions
- Configuration done by Pin Connect Block



Pin Connect Block (2)

- Pin Function Select Registers
  - PINSEL0 and PINSEL1
    - Configuration of **P0**
    - Assign P0.0 ... P0.31 to GPIO or an alternate function (1 of max. 3)
  - PINSEL2 (**not** available in 48-pin devices)
    - Configuration of **P1** (64/144-pin devices) and **P2, P3** (144-pin devices)
    - Select availability of debug and trace ports on Port1 pins
    - Controls use of address/data bus and strobe pins (144-pin devices)
    - Selection of additional ADC-inputs (144-pin devices)

Pin Connect Block (3)

Example:

...

- Pin Function Select Register 0 [PINSEL0 - 0xE002C000)] R/W

...	...	...
PINSEL0 21:20	P0.10	00: GPIO Port 0.10 01: RTS (UART1) 10: Capture 1.0 (Timer 1) 11: reserved
...	...	...

External Memory Controller* (1)

- Supports static memory devices (RAM, ROM, external I/O)
- Up to 4 independent banks, each up to 16M Bytes
- Programmable
  - Bus turnaround (idle) cycles (1 to 16)
  - Read and write WAIT states (up to 32)
  - Write protection
  - Burst mode operation
  - External data width: 8, 16, or 32 bits

*: LPC2000 144-pin devices only, LPC2378, LPC24xx

External Memory Controller* (2)

- Available signals

Pin name	Type	Description
D [31:0]	Input / Output	External memory data lines
A [23:0]	Output	External memory address lines
OE	Output	Output Enable (active low)
BLS [3:0]	Output	Byte Lane Select (active low)
WE	Output	Write Enable (active low)
CS [3:0]	Output	Chip Select (active low)

*: 144-pin devices only

External Memory Controller* (3)

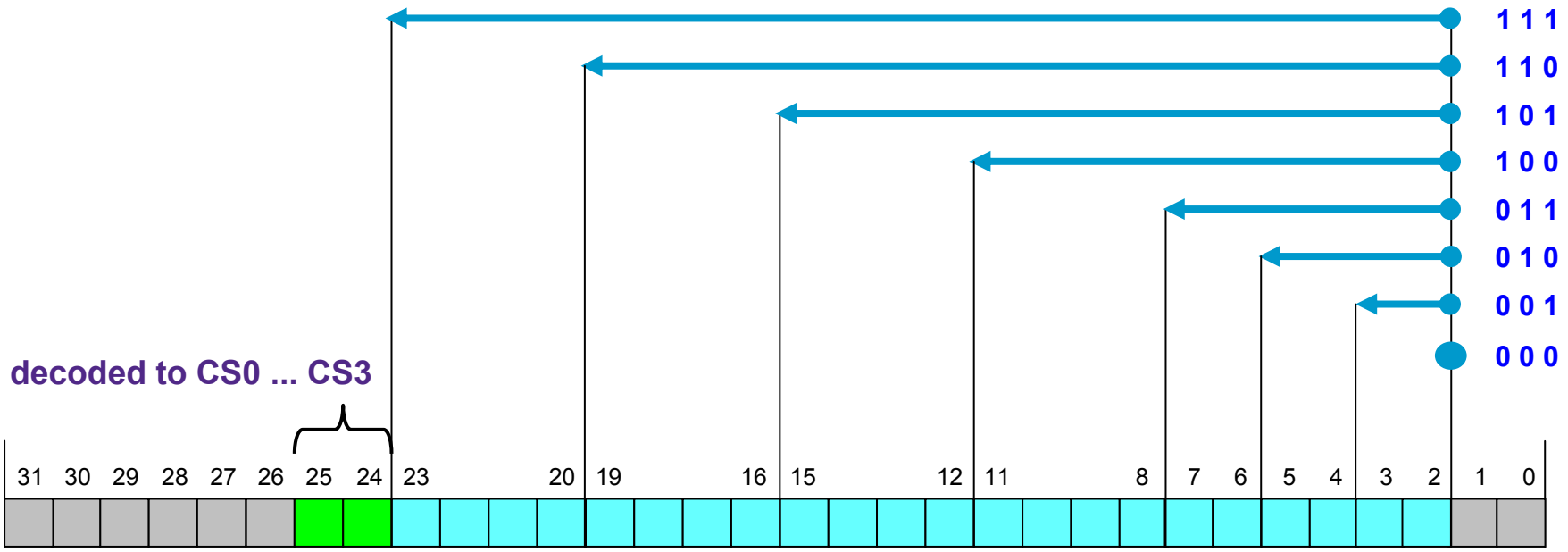
- Selection of **external data bus width** (144-pin devices only)
  - Determined by state of pins **Boot0** and **Boot1** during Reset

P2.27 / D27 / **Boot1**	P2.26 / D26 / **Boot0**	Boot from
0	0	8-bit memory on CS0
0	1	16-bit memory on CS0
1	0	32-bit memory on CS0
1	1	Internal Flash memory

*: 144-pin devices only

External Memory Controller* (4)

- Size of external memory
  - **Number of external address lines** defined by **PINSEL2 27:25**
  - Remaining port pins available as GPIO



*: 144-pin devices only

GPIO Exercise

Hands-On

- Exercise:
  Modify the example
  to use a macro or function
  for LEDs

- Optional Exercise:
  "Invent" a display pattern
  indicating seconds and
  125ms

- Hint:
  - SET_LEDS(byte)
  - LEDs are on
    Port 1, bits 16-23

- Examples:
  - Walking LED, or
  - Bar indicator

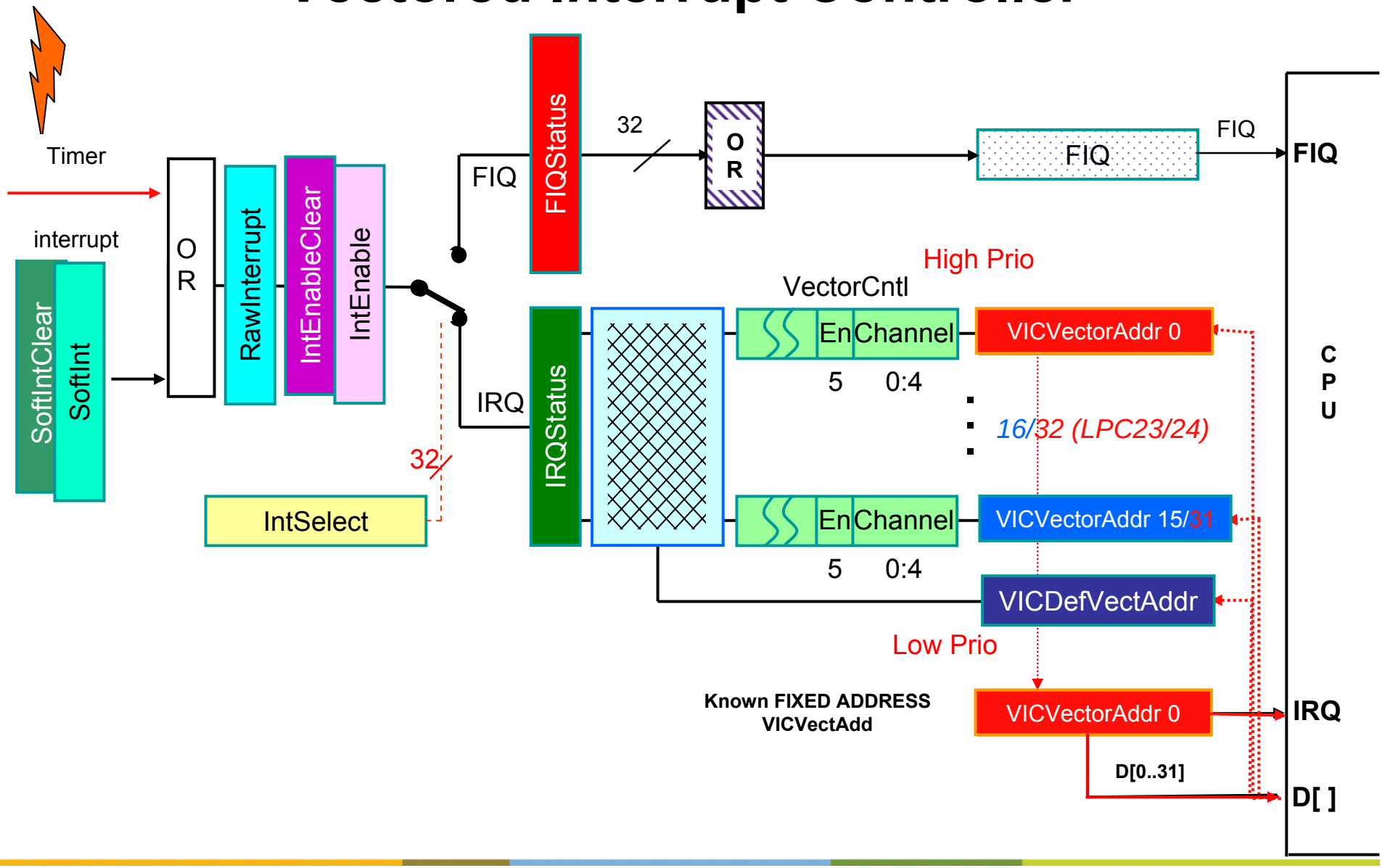
1. Memory Addressing
2. System Control Block
3. Memory Accelerator Module (MAM)
4. General Purpose I/O / Pin Connect Block / Memory Controller
5. **Vectored Interrupt Controller**
6. Integrated Peripherals

    Timer 0 / Timer 1, UART 0 / UART 1, I²C, SPI, PWM, RTC, Watchdog

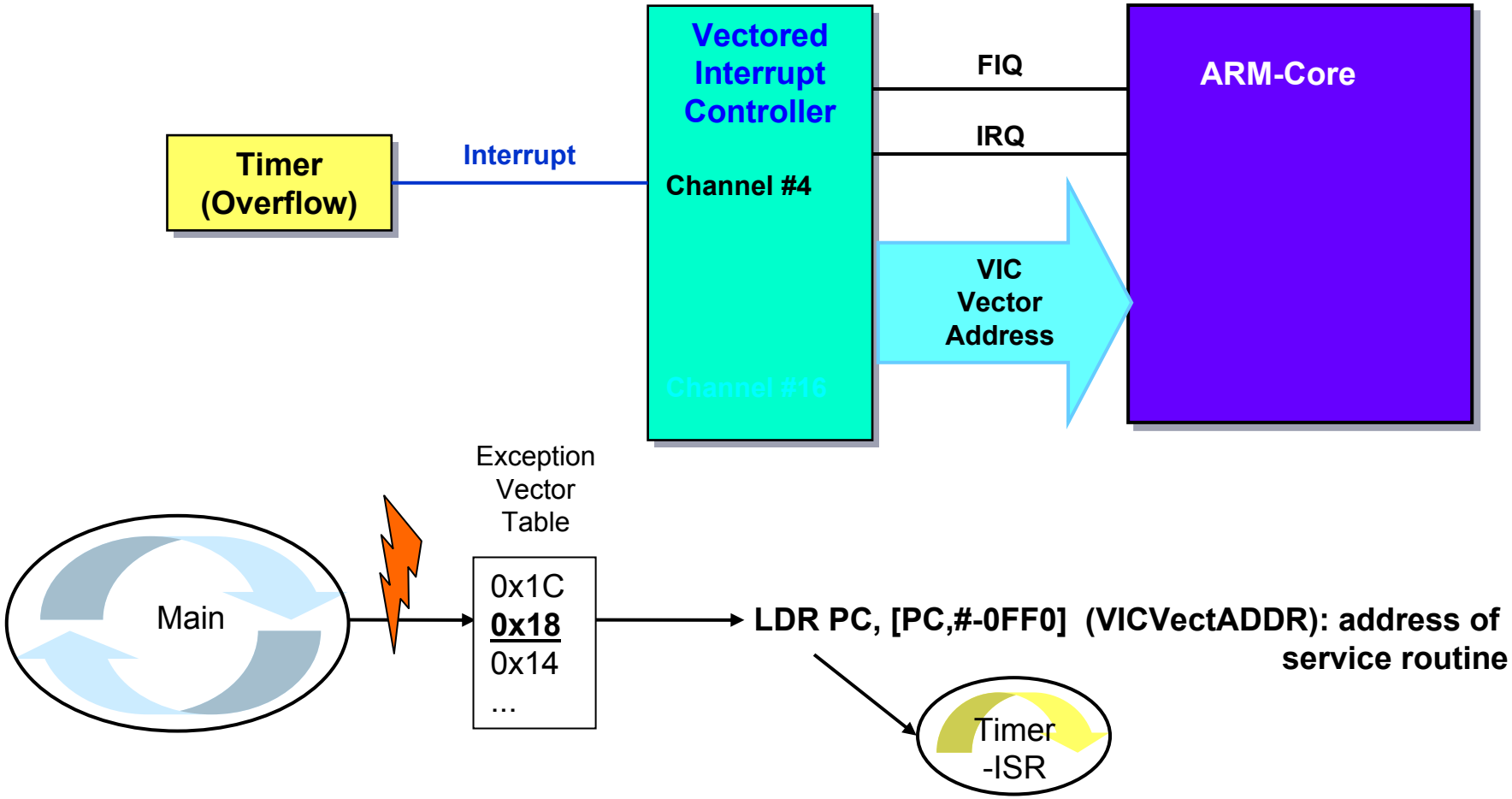
Vectored Interrupt Controller

- ARM PrimeCell™
- 32 interrupt request inputs
- 16/32 IRQ interrupts can be auto-vectored (in the LPC23/24)
  - single instruction vectoring to ISR
  - dynamic software priority assignment
- 32 FIQ non-vectored interrupts
- 32 Software interrupts

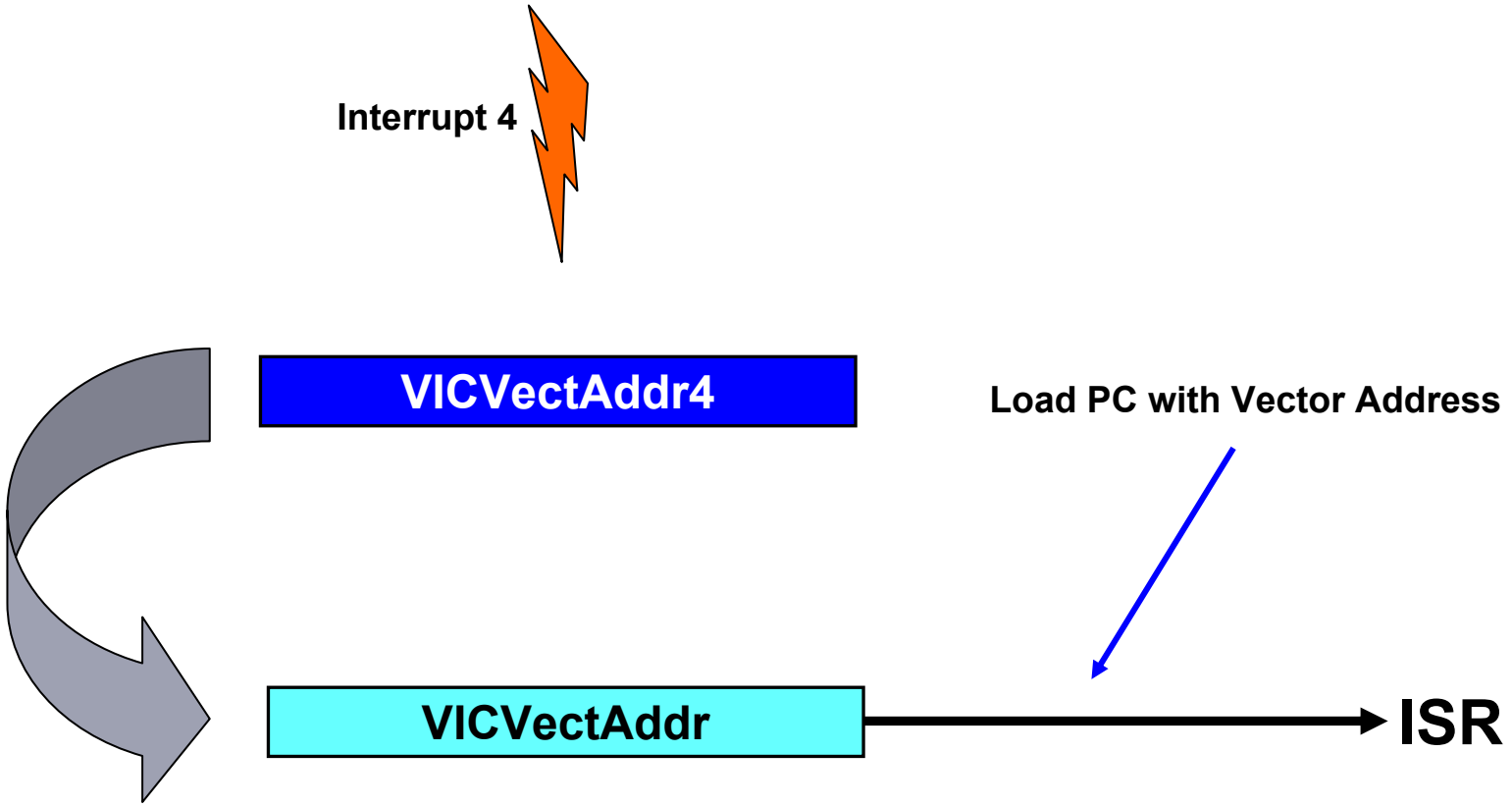
Vectored Interrupt Controller



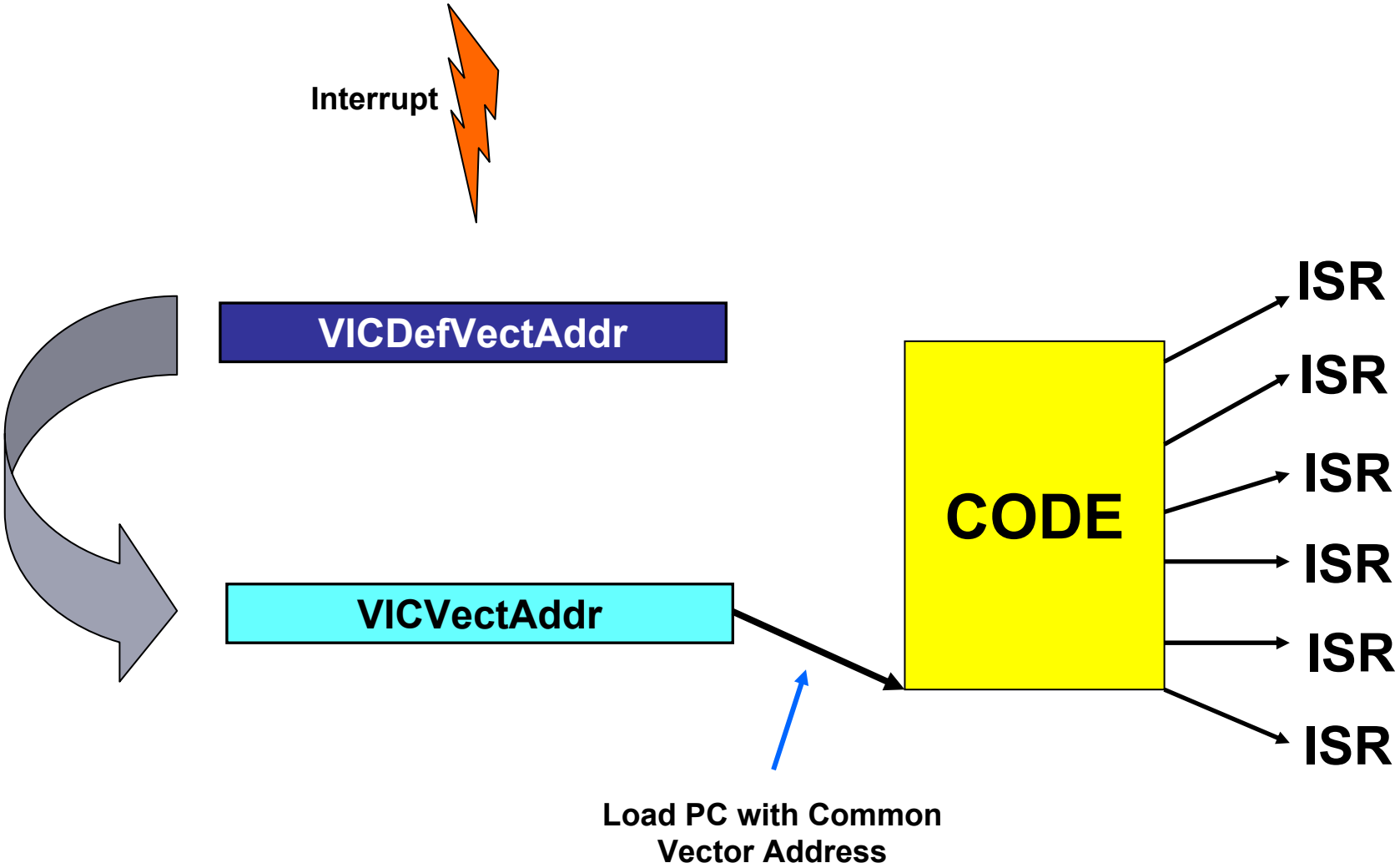
IRQ Interrupts



VIC - Vectored Interrupts



VIC - Non-Vectored Interrupts

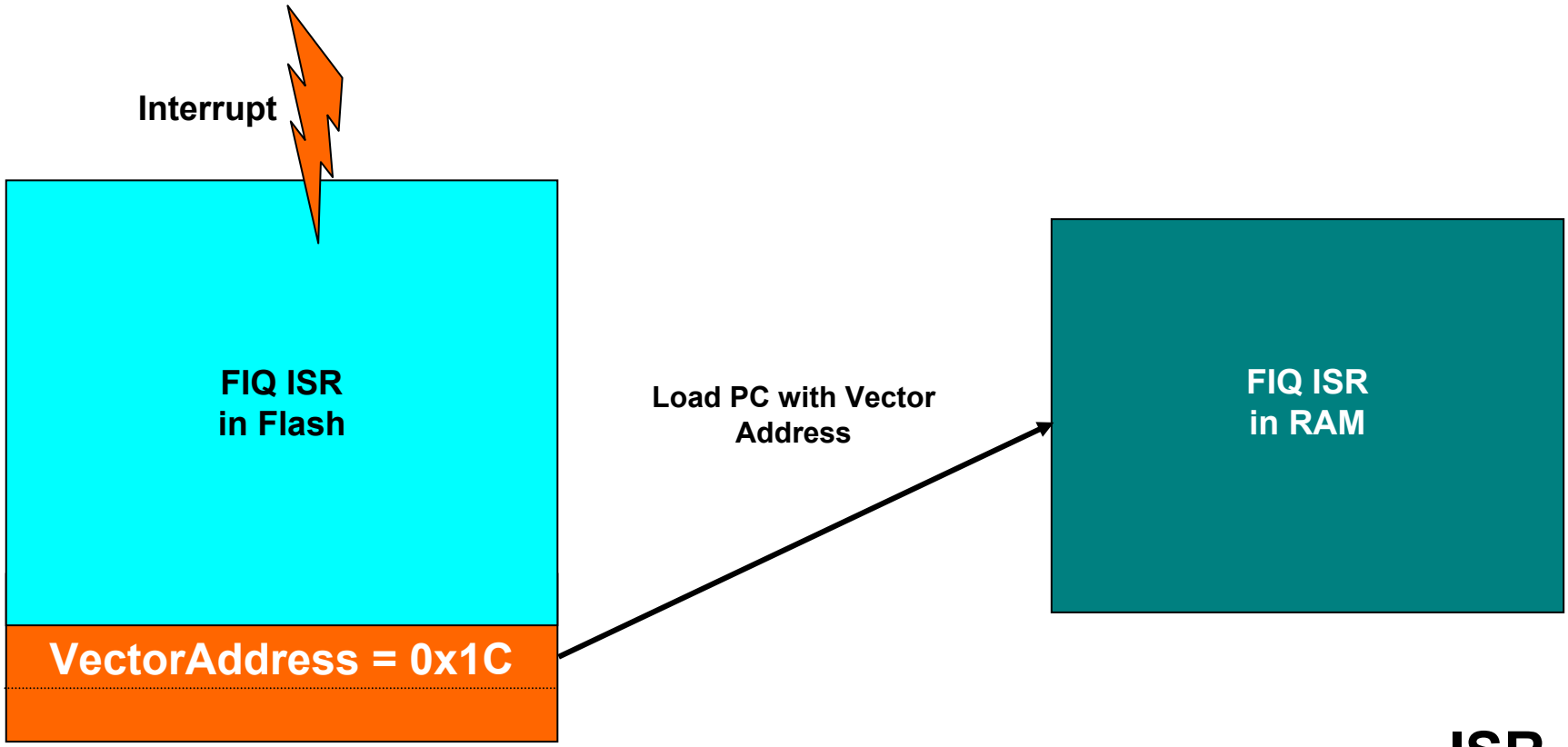


VIC - FIQ Interrupt

- FIQs have higher priority than IRQs
  - Serviced first
  - FIQs disable IRQs

- FIQ Vector is last in vector table (allows handler to be run sequentially from that address)
- FIQ mode has 5 extra banked registers, r8-12 (interrupt handlers must always preserve non-banked registers)

VIC – FIQ (Fast Interrupt)



ISR

Interrupt Prioritizing with Nesting

IRQ Service Routine of “lower” priority:

1. Push SPSR to LR and on stack
2. Switch to System Mode (write CPSR, enable IRQ)
3. Push system mode link register on stack
4. Execute “real” service routine
5. Pop system mode link register from stack
6. Switch back to IRQ mode
7. Pop SPSR to LR and restore SPSR
8. Clear IRQ source
9. Reset VIC

For Keil compiler, see knowledgebase at www.keil.com

Hands-On Index

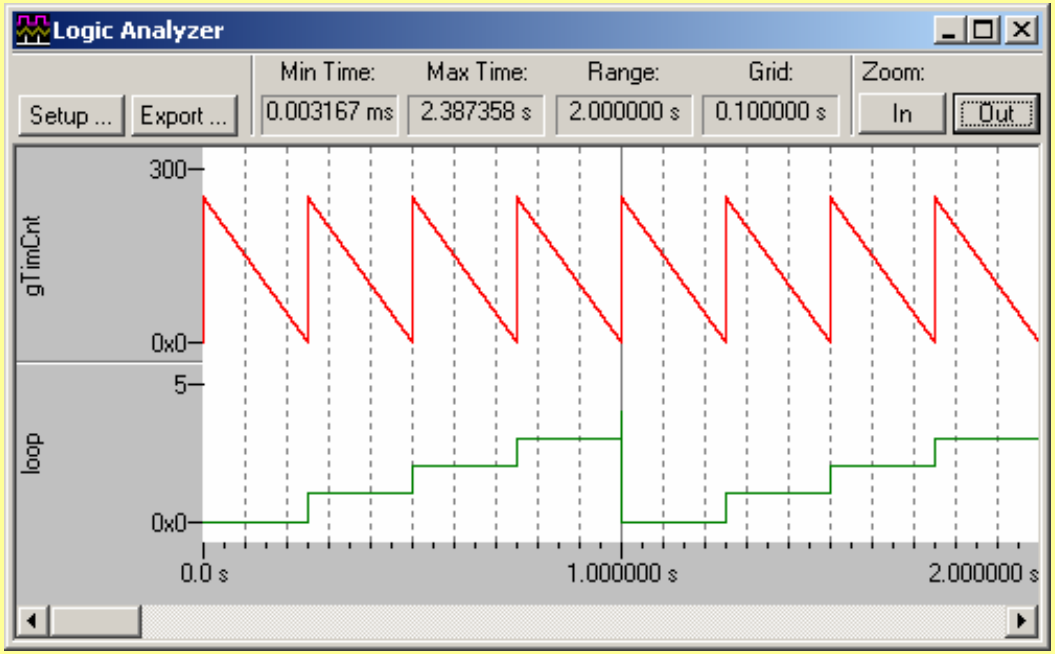
Hands-On

1. Tools Setup
2. Oscillator / PLL / MAM / GP I/O
3. ADC
4. Interrupts / Timer
5. UART / CAN

Project: BlinkyIRQ

Hands-On

- Load Project BlinkyIRQ
- Build and Debug
- Simulation
  - Use Logic Analyzer
    - gTimCnt
    - loop
  - Peripheral window
    - Vectored Int. Contr.



Exercise:
Timer & Interrupt

Hands-On

- Add timer 1 interrupt to BlinkyIRQ
- Interrupt all 333us
- Use Vector 1
- In ISR, display some LED pattern on port 1.20 to 1.23 every 333ms

- Test and debug in simulator using:
  - Performance Analyzer
  - Signal Analyzer
  - Peripheral GPIO window
  - Peripheral Timer window
  - Peripheral VIC window
- Test program on hardware

Interrupts

Hands-On

1. Tools Installation
   - Emulator/Debugger: Emul-ARM Evaluation System (Nohau)
   - Compiler: Embedded Workbench (IAR)
   - Flash Tool: Flash ISP Utility (NXP)
   - Hardware: LPC2106 Target Board (Nohau)
2. Oscillator / PLL
3. Memory Acellerator Module / General Purpose I/O
4. Interrupts
   a. Single Interrupt
   b. Nested Interrupts
5. UART
6. RTC

Single Interrupts (1)

Hands-On

- IAR Embedded Workbench
  - Load workspace Chapter 4a

    File -> Open Workspace -> Chapter 4a_single_Interrupt -> Chapter_4a.eww
  - Select project Training
  - Target Debug (= run from RAM!)
  - Make OK?
    - Correct error in Timer.c (User Manual !)
    - Try again

Single Interrupts (2)

Hands-On

- Start Seehau (Tools -> Seehau)
  - Run project (RUN -> Go or Button)
- What does Blinkie do in this case?
  - Browse through source codes
    - Blinkie_Main.c
    - Timer.c

Single Interrupts (3)

Hands-On

- Browse through source codes
  - VectorTabDebug.s (Assembler!)

Exception	Program Vector Table		
Reset	ldr	pc,=?cstartup	SUPERVISOR mode
Undefined	ldr	pc,=0x0000024	UNDEFINED mode
SWI	ldr	pc,=0x0000028	SUPERVISOR mode
Prefetch	ldr	pc,=0x000002c	ABORT mode
Data	ldr	pc,=0x0000030	ABORT mode
	nop		reserved
IRQ	ldr	pc,[pc,#-0xff0]	IRQ mode
FIQ	ldr	pc,=0x1C	IRQ mode

→ **Load contents of the ...**

... VICVectAddr. register into PC

Single Interrupts Solution

Hands-On

Timer.c

– T0_MCR= **0x03**;

//Interrupt on Match0, reset timer on match

Function

LEDs are changed whenever variable **TimerIsUp=1**

This variable is

- set by Timer 0 interrupt routine after Timer 0 match
- cleared by main routine before LEDs are changed

Nested Interrupts (1)

Hands-On

- IAR Embedded Workbench
  - Load workspace Chapter 4b

    File -> Open Workspace -> Chapter 4b_single_Interrupt -> Chapter_4b.eww
  - Select project Training
  - Target Debug (= run from RAM!)
  - Make OK?

Nested Interrupts (2)

Hands-On

- Start Seehau (Tools -> Seehau)
  - Run project (RUN -> Go or Button)
- Press SW4
- Release SW4

- What does Blinkie do in this case?

Nested Interrupts

Solution 1

Hands-On

– Two interrupt sources

- Timer 0 IR: blinks LEDs
- External IR 2 (SW4): stay in ISR until SW4 released

– Priorities

- External Interrupt > Timer

⇒ **SW4 blocks Timer 0**

(LEDs control)

– SW2, SW3 change direction (polled in main program)

Nested Interrupts (3)

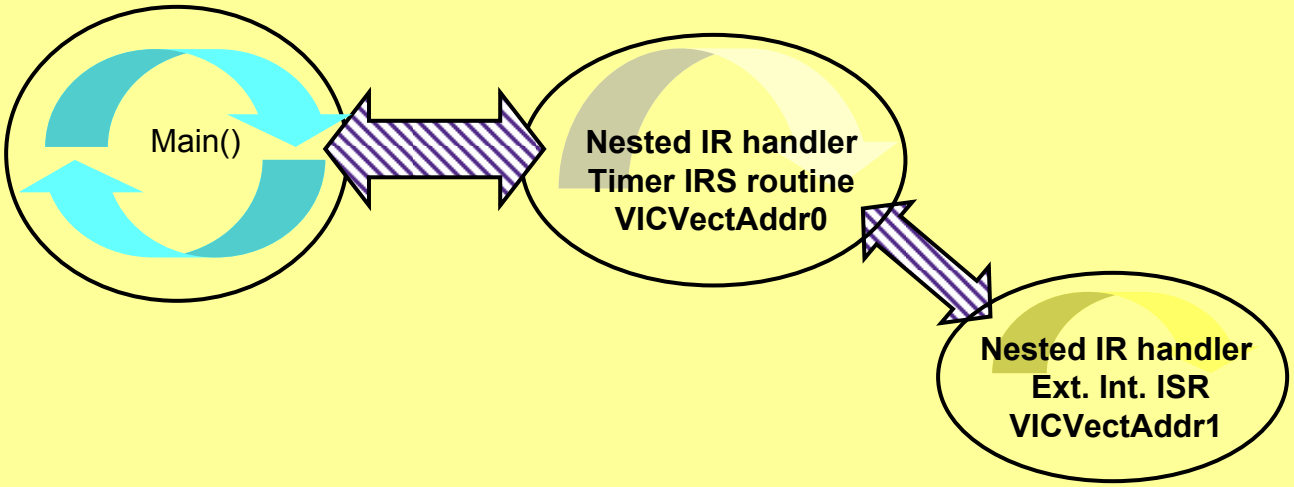
Hands-On

- Browse through source codes
  - Blinkie_main.c
  - ExtInt.c InitExtInt2()
  - Timer.c InitTimer0()
  - VectorTabDebug now initializes SP for different modes

Nested Interrupts (4)

Hands-On

– Nested IR-Handler now save status and re-enable further interrupts before entering ISR
  ⇒ nested interrupts possible!



Nested Interrupts (5)

Hands-On

– Browse through source codes (cont'd)

- NestedIrqHandler.s

Timer 0 nested	External Interrupt 2 nested
save work registers	save work registers
	ExtInt2-pin to GPIO
reset IRQ source	reset IRQ-source
switch to System mode, interrupts enabled	switch to System mode, interrupts enabled
branch to **distinct ISR**	branch to **distinct ISR**
switch to IRQ mode, IRQ disabled	switch to IRQ mode, IRQ disabled
	re-enable ExtInt2
reset VIC	reset VIC
restore work registers	restore work registers

Nested Interrupts (6)

Hands-On

– Change priorities: External Interrupt < Timer

- Edit Timer.c and ExtInt.c:

		VICVectAddr**n**	VICVectCntl**m**
InitTimer0()	is	1	1
InitExtInt2()	is	0	0
InitTimer0()	new	0	0
InitExtInt2()	new	1	1

Note: lower value (n, m) is higher priority!

Nested Interrupts (7)

Hands-On

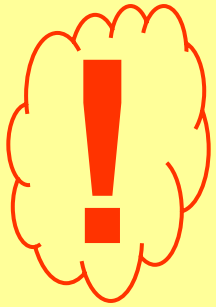
- Make OK?
- Start Seehau (Tools -> Seehau)
  - Run project (RUN -> Go or Button)

- What does Blinkie do now?

Nested Interrupts

Solution 2

Hands-On



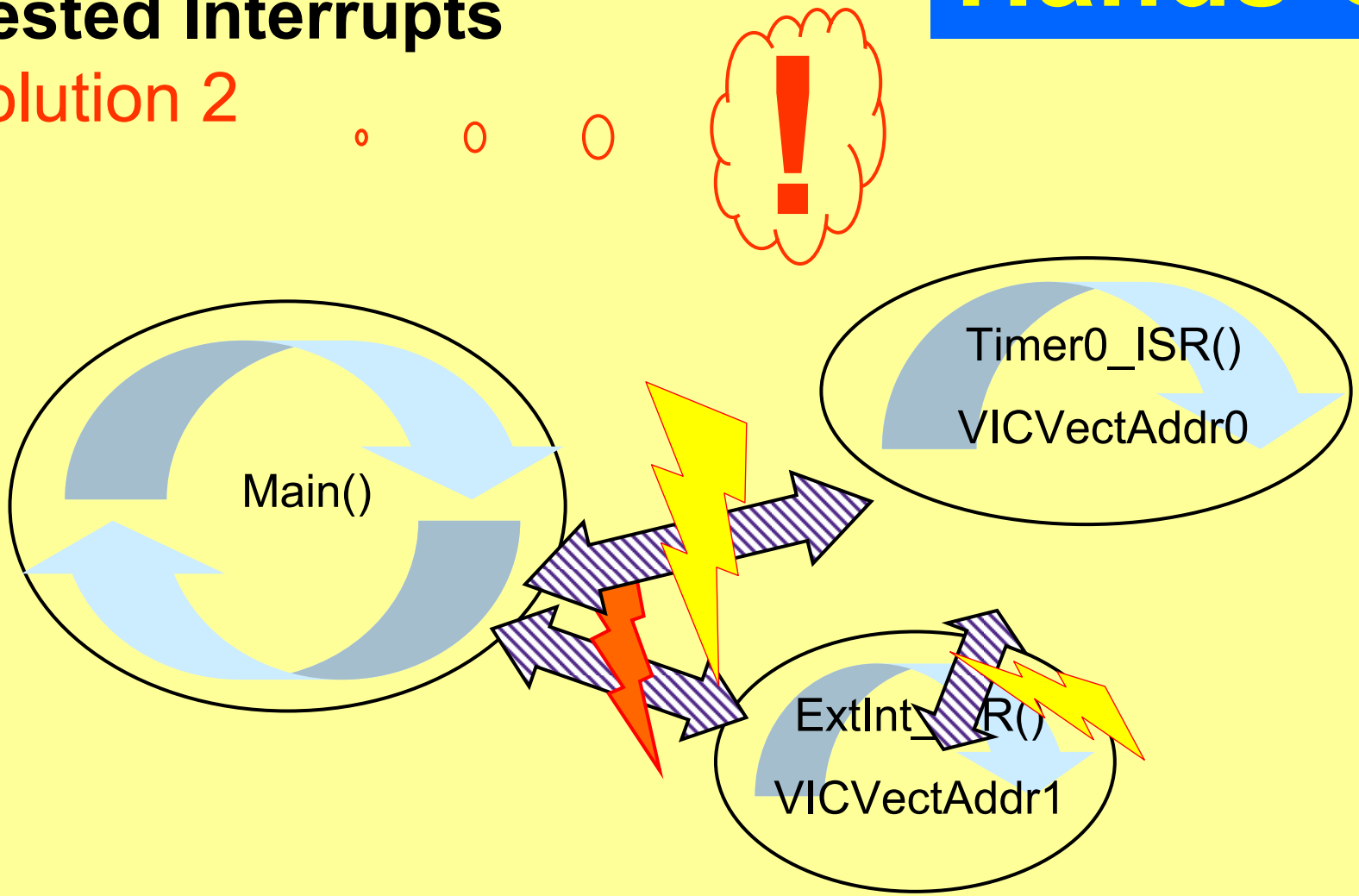
- Priorities
  - External Interrupt < Timer
    - **Blinkie continues, even while SW4 is pressed!**
    - LED5 toggles to indicate **recognition of External Interrupt**
- Two interrupt sources
  - Timer 0 IR: blinks LEDs
  - External IR 2 (SW4): stay in ISR until SW4 released

    (but can be interrupted!)
- SW2, SW3 change direction (polled in main program)

Hands-On

Nested Interrupts

Solution 2



VIC
IR Sources

LPC2104
LPC2105
LPC2106

Block	Flag(s)	VIC IR Source #
WDT	Watchdog Interrupt (WDINT)	0
-	Reserved for software interrupts only	1
ARM Core	Embedded ICE, DbgCommRx	2
ARM Core	Embedded ICE, DbgCommTx	3
Timer 0	Match 0 -2 (MR0, MR1, MR2, MR3) Capture 0 - 2 (CR0, CR1, CR2)	4
Timer 1	Match 0 -3 (MR0, MR1, MR2, MR3) Capture 0 - 3 (CR0, CR1, CR2, CR3)	5
UART 0	Rx Line Status (RLS) Transmit Holding Register empty (THRE) Rx Data Available (RDA) Character Time-out Indicator (CTI)	6
UART 1	Rx Line Status (RLS) Transmit Holding Register empty (THRE) Rx Data Available (RDA) Character Time-out Indicator (CTI) Modem Status Interrupt (MSI)	7
PWM 0	Match 0 - 6 (MR0, MR1, MR2, MR3, MR4, MR5, MR6)	8
I2C	SI (state change)	9
SPI	SPIF, MODF	10
-	reserved	11
PLL	PLL Lock (PLOCK)	12
RTC	RTCCIF (Counter Increment), RTCALF (Alarm)	13
System Control	External Interrupt 0 (EINT0)	14
System Control	External Interrupt 1 (EINT1)	15
System Control	External Interrupt 2 (EINT2)	16

add'l

VIC IR Sources

LPC2114/24
LPC2119*/29*
LPC2194**
LPC2210/12/14LPC2290/92****
LPC2294***

Block	Flag(s)	VIC IR Source #
...	...	...
System Control	External Interrupt 3 (EINT3)	17
A/D	A/D Converter	18
* CAN	ORed CAN Acceptance Filters	19
* CAN	CAN 1 Transmitter	20
* CAN	CAN 1 Receiver	21
** CAN	CAN 2 Transmitter	22
** CAN	CAN 2 Receiver	23
*** CAN	CAN 3 Transmitter	24
*** CAN	CAN 3 Receiver	25
*** CAN	CAN 4 Transmitter	26
*** CAN	CAN 4 Receiver	27

plus more flags for UART0, Timer1, Capture 0 - 3 (CR0, CR1, CR2, CR3)

VIC Interrupt Latency

- Some instructions take multiple cycles to execute and cannot be interrupted
  - The longest STM and LDM instructions take 20 cycles
  - Subsets can be reduced to 7 cycles
- FIQ: 12 (25) cycles, 200ns (416ns) @ 60MHz
- First Vectored IRQ (assuming no FIQ pending):
  26 (39) cycles, 433ns (650ns) @ 60MHz
- Default IRQ Vector (assuming no other IRQ pending):

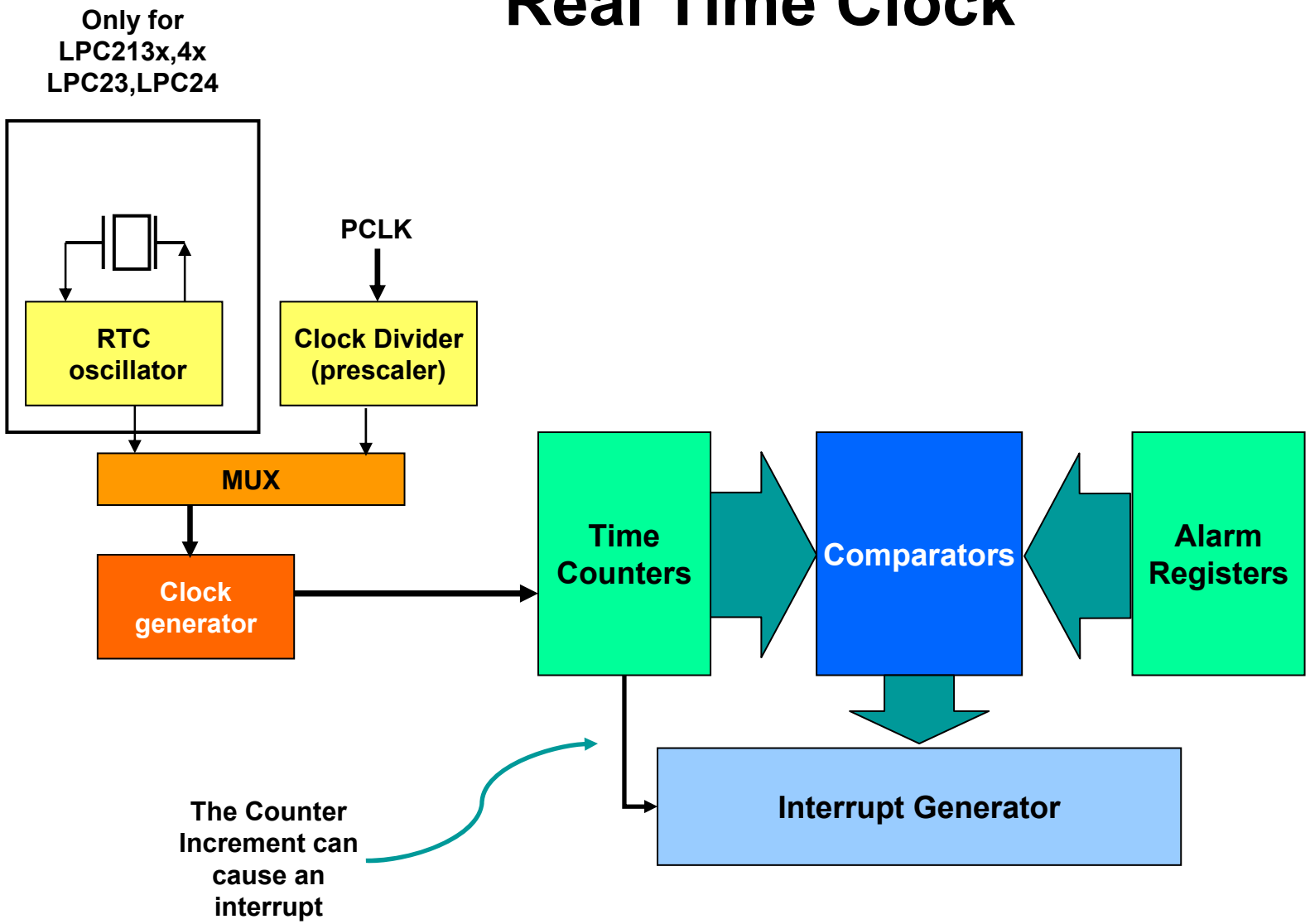
  42 (55) cycles, 700ns (916ns) @ 60MHz

1. Memory Addressing
2. System Control Block
3. Memory Accelerator Module (MAM)
4. General Purpose I/O / Pin Connect Block
5. Vectored Interrupt Controller
6. **Integrated Peripherals**

   Timer 0 / Timer 1, UART 0 / UART 1, I²C, SPI, PWM, RTC, Watchdog

RTC

Real Time Clock



Real Time Clock

- Full Clock/Calendar function with alarms
  - generates its own 32.768 kHz reference clock from any crystal frequency
  - counts seconds, minutes, hours, day of month, month, year, day of week and day of year
  - can generate an interrupt or set an alarm flag for any combination of the counters

Power consumption of RTC (LPC213x)

Core	RTC	Current consumption from V_{bat}
Power down	Power down	20-30µA
Power down	Running from V_{bat}	20-30 µA
Active (pclk=15MHz)	Running	around 80 µA

IAP programming

4 ways of programming the LPC2000 Flash

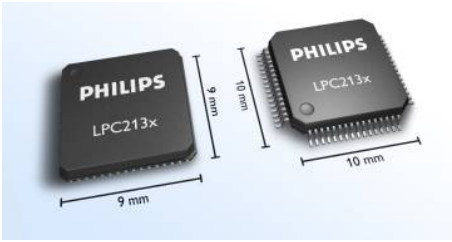
In-Application Programming (IAP)

In-System Programming
(ISP via NXP Boot Loader, UART0)

In-System Programming
(through LPC HW macro cells/
and custom software protocol)

Parallel programmer

LPC213x Series Overview



FLASH MEMORY

- **ISP:** In-System programming is programming or reprogramming the on-chip flash memory, using the boot loader software and a serial port. This can be done when the part resides in the end-user board.
- **IAP:** In-Application programming is performing erase and write operation on the on-chip flash memory, as directed by the end-user application code.
- **User-code security :** Code read protection is enabled by programming the flash address location 0x1FC (User flash sector 0) with value 0x87654321 (2271560481 Decimal).

  Disabled:

  - JTAG debug port
  - External memory boot
  - Following ISP commands:

    Read Memory

    Write to RAM

    Go

    Copy RAM to Flash

ADC

A/D Converter

- Features
  - 10 bit successive approximation analog to digital converter
  - Multiplexed inputs
    - 4 pins (64-pin devices)
    - 8 pins (144-pin devices)
  - Power down mode
  - Measurement range 0V ... 3V
  - Minimum 10 bit conversion time: 2.44 µS
  - Burst conversion mode for single or multiple inputs
  - Optional conversion on transition on input pin or Timer Match signal
  - Programmable divider to generate required 4.5MHz from VPB clock

A/D Converter – Burst mode

- CLKS: bit 17, 18, 19 of ADCR select the number of clocks used per conversion and the accuracy

  – 000b: 11 clocks, 10 bits

  – 001b: 10 clocks, 9 bits

  – 010b: 9 clocks, 8 bits

  – 011b: 8 clocks, 7 bits

  – …

  – 111b: 4 clocks, 3 bits

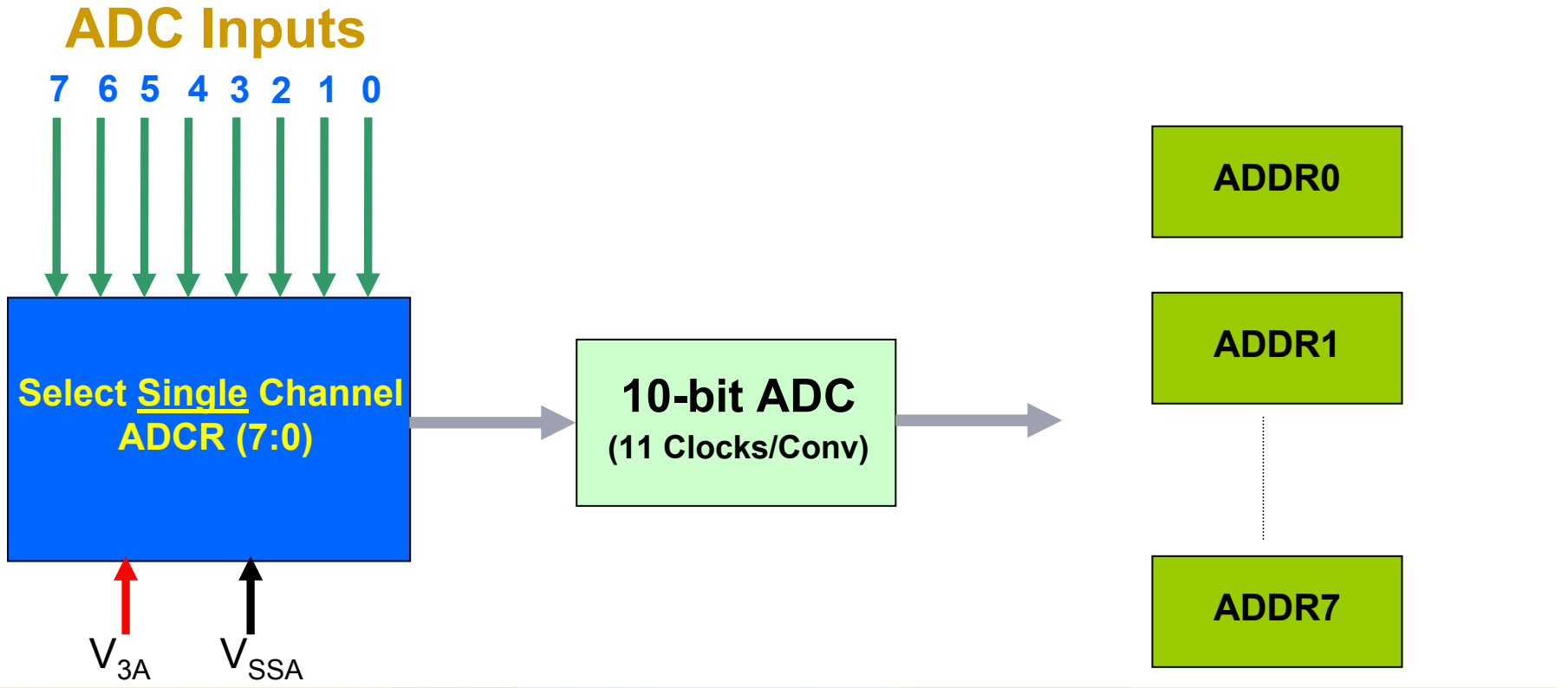
ADC LPC213X, LPC214X

- Separate result register for each channel
  - Reduces the interrupt overhead by a factor of 8

- Measurement range of 0 V to 3 V
  - Separate voltage pins for analogue 3V supply (V_{3A}) and analogue ground (V_{SSA})

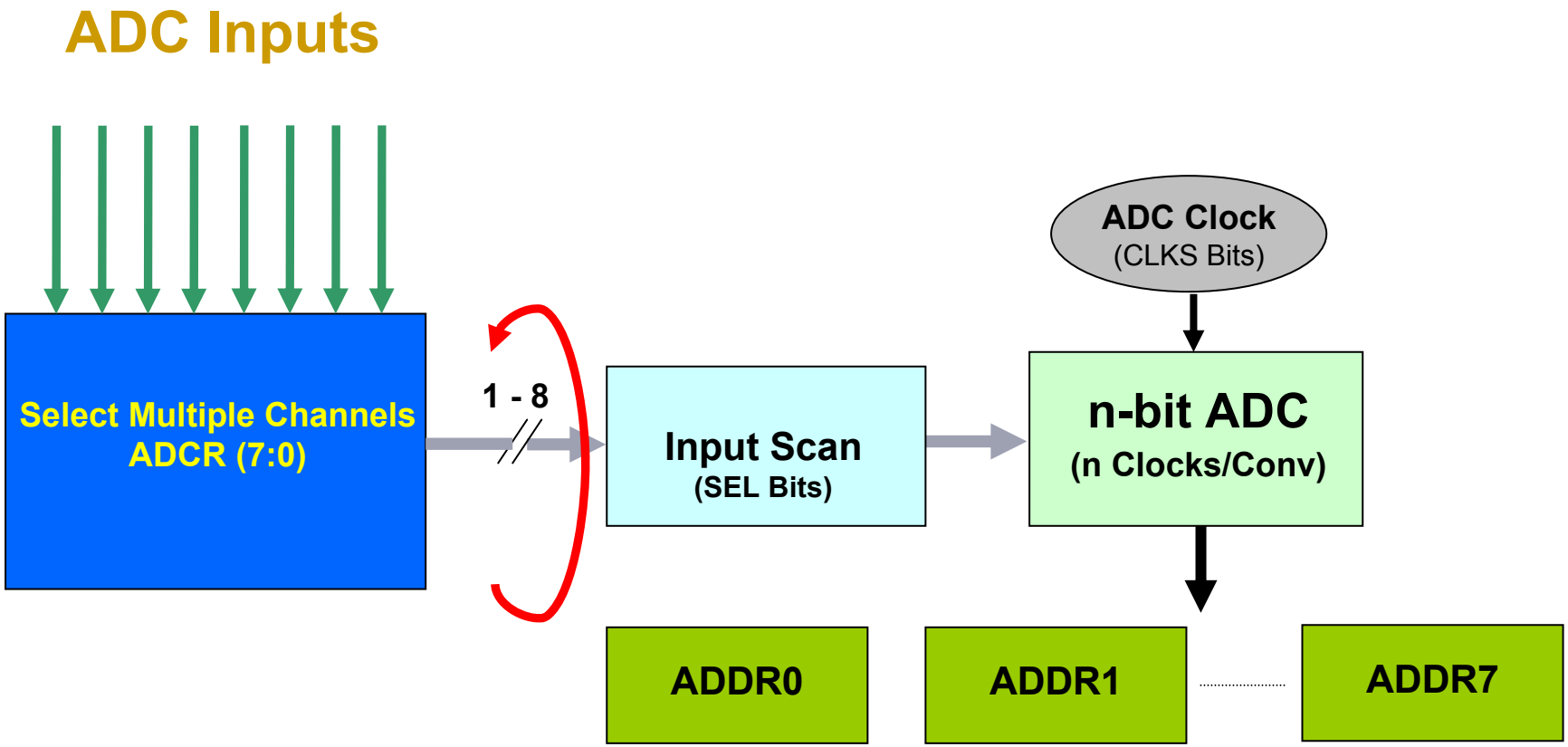
ADC – Software Controlled Mode

- All conversions are 10-bit and take 11 clocks
- 4.5Mhz Maximum Clock
- Allows conversion to start on an external edge



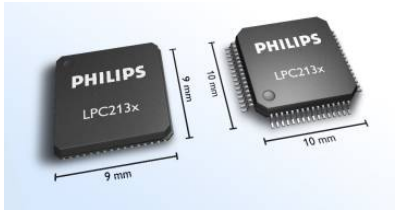
ADC – Burst Mode

- Result accuracy and speed are programmable
- Input selected by the SEL bits are scanned



DAC

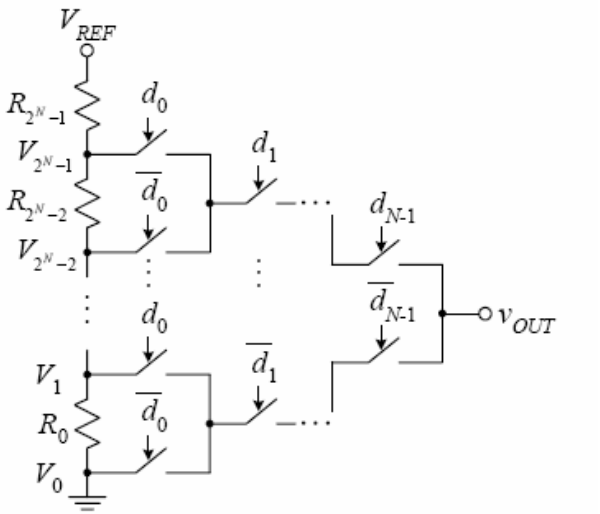
LPC213x Series Overview



DAC (10-bit)

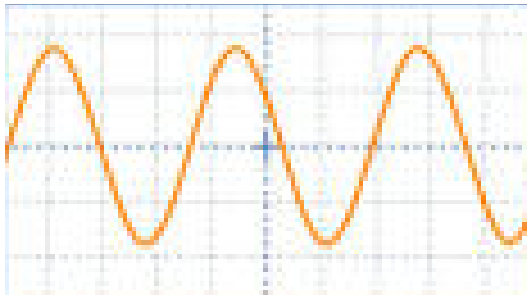
This peripheral is available in the LPC2138 only. The D/A converter enables the LPC2138 to generate variable analog output.

- 10 bit digital to analog converter
- Resistor string architecture
- Buffered output
- Power down mode
- Selectable speed vs. power



LPC213x DAC

- Digital-to-Analog Converter (DAC)
  - 10-bit resolution DAC with a buffered output
    - Last output value is held as long as DAC is on
  - Output from Zero Volt to Reference Voltage
    - In 1024 steps
  - Selectable Conversion speed vs. power
    - Settling time 1us, up to 350uA
    - Settling time 2.5us, up to 700uA
  - Selective power down



Hands-On Index

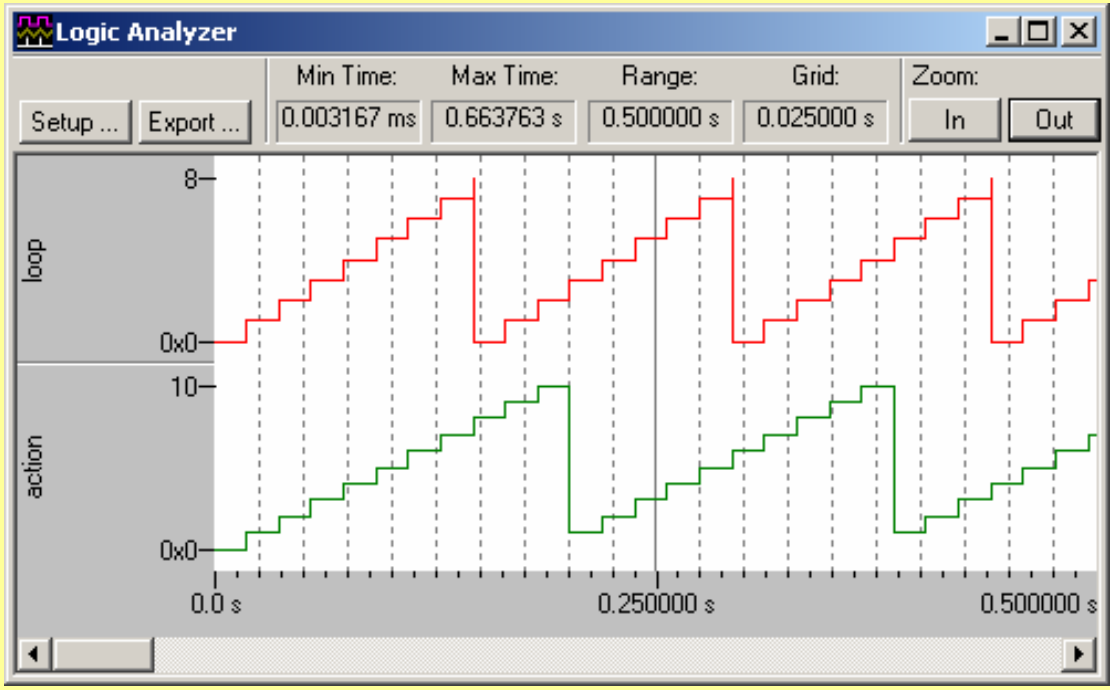
Hands-On

1. Tools Setup
2. Oscillator / PLL / MAM / GP I/O
3. ADC
4. Interrupts / Timer
5. UART / CAN

Project: BlinkyADC

Hands-On

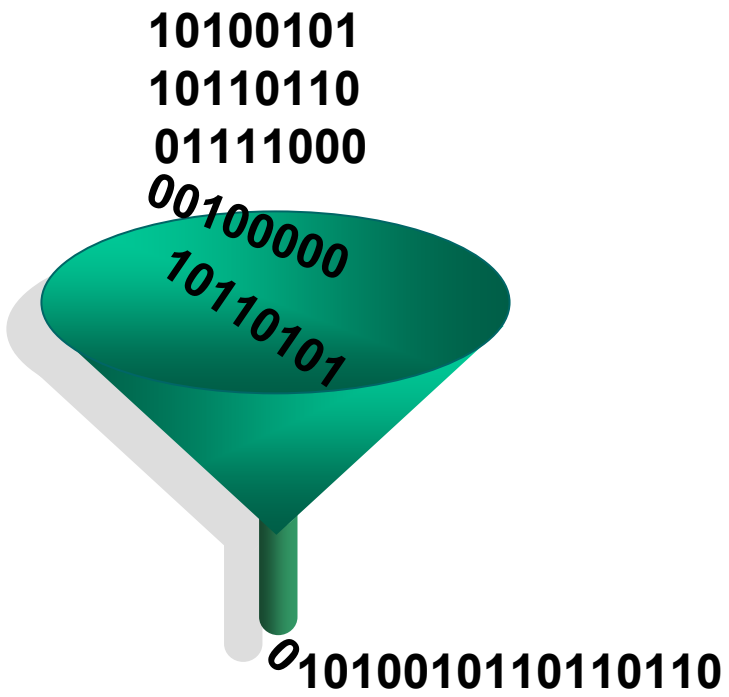
- Load Project BlinkyADC
- Build and Debug
- Simulation
  - Use Logic Analyzer
  - Peripheral windows
    - GPIO P1
    - ADC -> modify voltage
- Run on hardware
  - Verify change of AD values with potentiometer



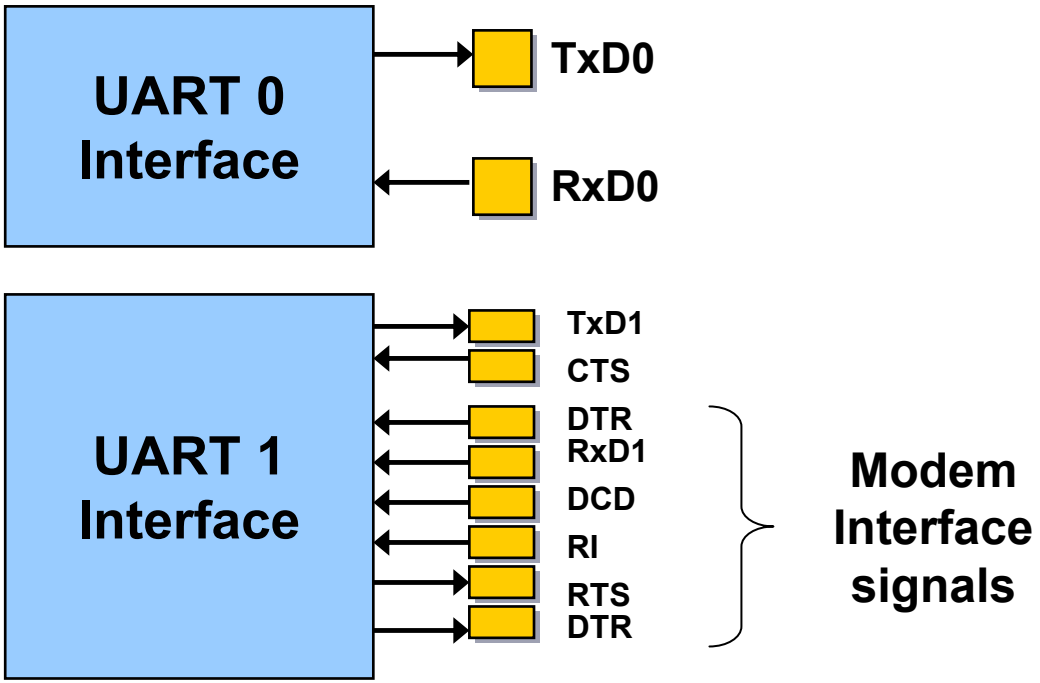
UART

Serial Communication Interfaces

- Two UARTs
- I²C
- SPI
- CAN
- SSP



UART0 / UART1



Maximum possible speed of the UART

3.75 Mbits/sec

UART0 / UART1

- Register locations conform to ‘550’ industry standard UART

- Built-in Baud Rate Generator
  - 16-bit baud rate generator clock divisor made from 2 8-bit divisor registers: DLM (MSB), DLL (LSB)
  - Required baud rate: pclk / (16 x Divisor*)
    (* if the Divisor is 0x0000, it is treated as 0x0001.)
  - Fractional baud rate generator (LPC214x, LPC2101-2-3)

- Error Detection
  - Parity, Framing and Overrun Errors detected
  - Break Interrupt detection

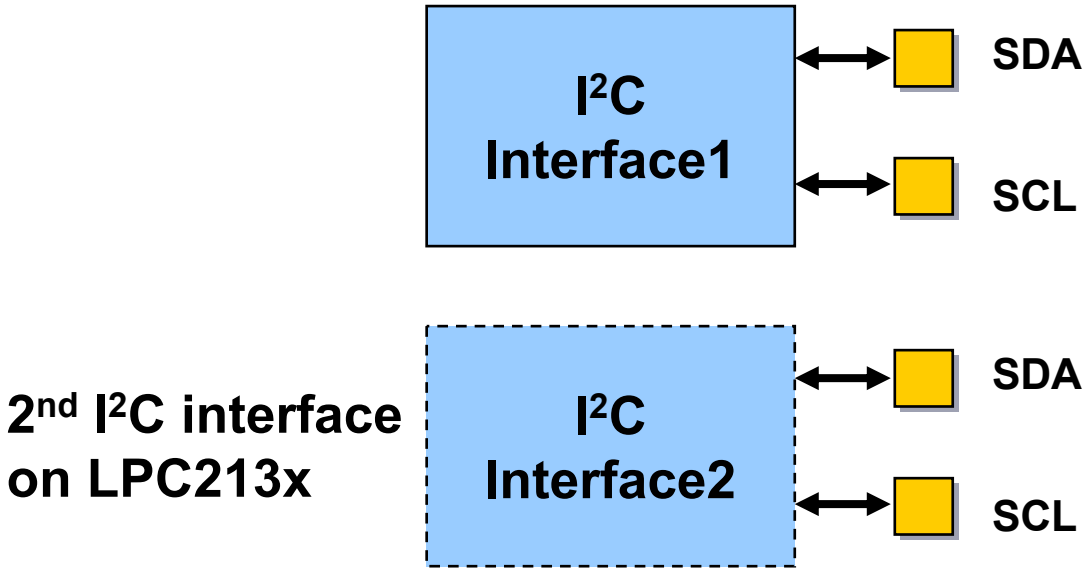
UART0 / UART1 (cont.)

- 16 byte Receive and Transmit FIFOs
  - Receive FIFO trigger points at 1, 4, 8, and 14 bytes
  - Break signal can be transmitted
- Word Length Select: 5, 6, 7 or 8-bit characters
- Stop Bit Select: 1 or 2 stop bits
- Parity Select: Odd or Even parity
- Supports 6 modem control signals
  - CTS, RTS, DCD, DSR, DTR and RI functions are selectable
  - Note:

    On 48-pin devices UART 0 has Tx and Rx pins only

IIC

I²C Bus Interface(s)



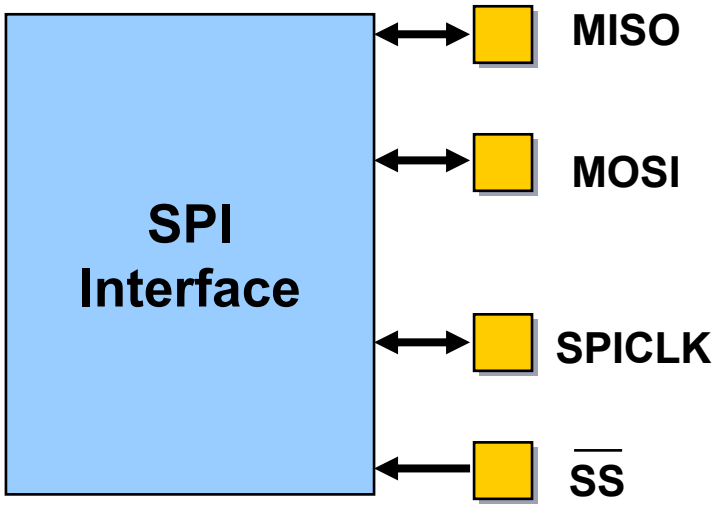
Maximum possible speed of the I²C

400Kbits/sec

I²C Bus

- Fast-I²C compliant bus interface
  - configurable as Master, Slave, or both
  - multi-master bus
  - bi-directional data transfer between masters and slaves
  - up to 400kb/s
  - arbitration between simultaneously transmitting masters without corruption of serial data on the bus
  - serial clock synchronization allows devices with different bit rates to communicate via one serial bus
  - programmable clock rate

SPI Interface



Maximum possible speed of the SPI

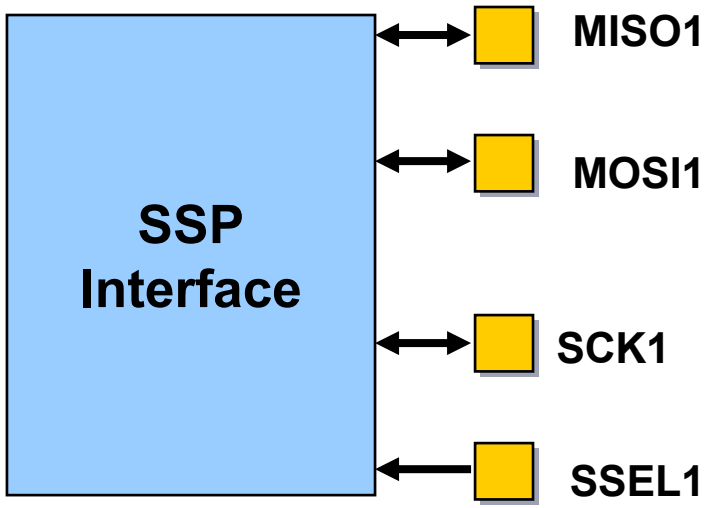
7.5 Mbits/sec

SPI Bus

- Compliant with Serial Peripheral Interface (SPI) specification
- Combined SPI master and slave function
- Maximum data bit rate of 1/8 of the peripheral clock rate
- Programmable clock polarity and phase for data transmit/receive operations
- No. of SPI channels:
  - 1 (48-pin devices)
  - 2 (64/144-pin devices)

SSP

Serial Synchronous Port (SSP) Interface



Maximum possible speed of the SSP

30 Mbits/sec

SSP Interface

- Compatible with Motorola SPI, 4-wire TI SSI and National Semiconductor Microwire bus
- Supports multiple slaves and master on the bus but at any point of time only one master and slave can communicate
- 8-Frame FIFO's for both Transmit and Receive
- Frame sizes can be from 4 bits to 16 bits

USB

USB 2.0

- Since USB is a standard doesn't that make all microcontrollers with USB the same?

  NO!!

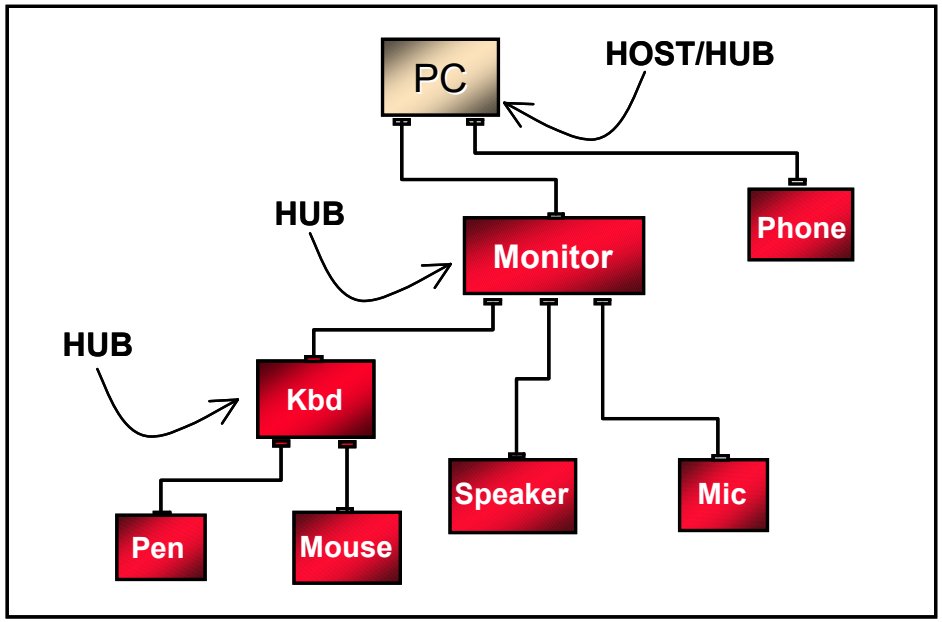
- Architectural choices and implementation details make a big difference in performance and ease of use.
- The LPC2148 is a high performance USB device.

LPC2148 key high performance USB features

- Flexible endpoint architecture
  - Supports all 32 USB endpoints
- Large data FIFO
  - Can double buffer full isochronous packets
- Flexible DMA capability
  - USB Buffer is present on the AHB bus

Introduction to USB

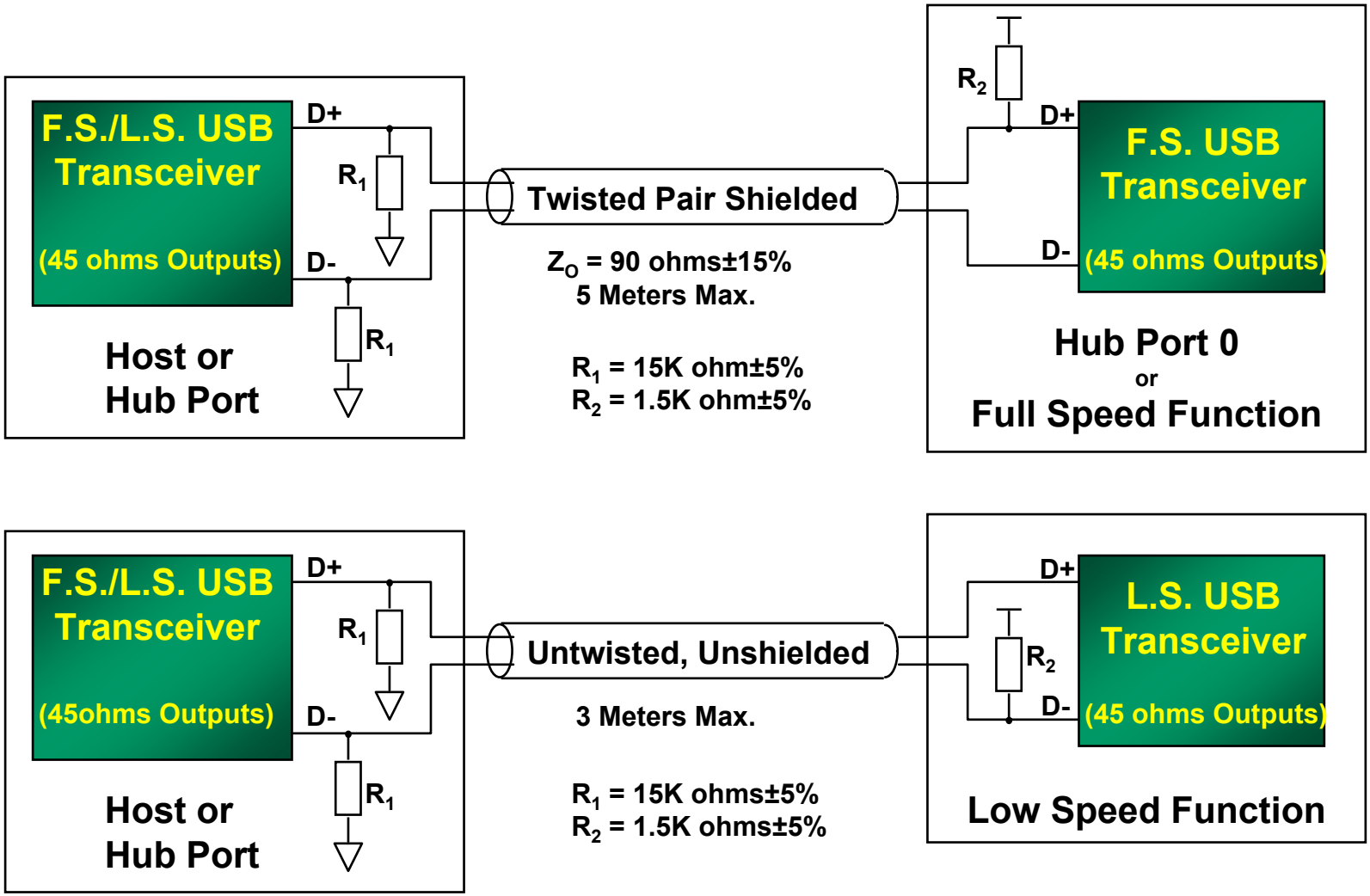
- An Auto-configuring Plug-and-play Serial Bus
- Single Master, Half-duplex, Time-multiplexed bus; Tiered Star topology
- All data communication is initiated and regulated by the Master; peer-to-peer not allowed
- USB 2.0 is the latest version of USB (downward compatible)



USB- A Brief History

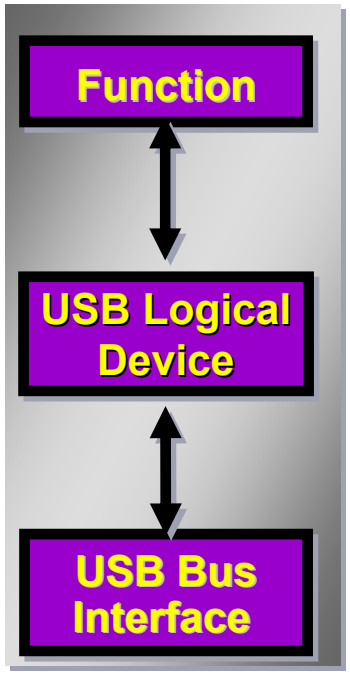
USB 1.1	USB 2.0	OTG
▸ Approved on 11/23/99 by the USB Core team ▸ 12 Mbps bus ▸ Full-speed (12 Mbps) ▸ Low-speed (1.5 Mbps) ▸ Standard A connector and standard B connector	▸ Original USB 2.0 specification released on April 27, 2000 ▸ 480 Mbps bus ▸ High-speed (480 Mbps) ▸ Full-speed (12 Mbps) ▸ Low-speed (1.5 Mbps) ▸ Backward compatible with USB 1.1 ▸ New mini-B connector	▸ Supplements the 2.0 specification ▸ Connects peripherals directly to each other ▸ New mini-A connector and mini-AB receptacle

USB Connections and Terminations



Device Layers

- Function
  - Represents capability delivered by the device, The primary „usage“ function, like mouse, audio output, printer, hub, etc.
- USB Logical Device
  - Defines common view of device by host
  - Manages high-level protocol
  - Typically controlled by system software
  - May have multiple functions with multiple endpoints
- USB Bus Interface
  - Physical interface to wire
  - Manages low-level protocol



Device Abstractions

- End Point
  - Ultimate data source or sink at the device end
  - Unique address, unidirectional, transfer characteristics
  - Each endpoint is unidirectional and has a transfer type associated with its peripheral
- Pipe
  - Association of endpoint with host SW owner
- Interface
  - Collection of pipes
  - Map to a capability
  - Owned by exactly 1 software client
  - More that 1 interface can be defined in a device

Endpoints

- Up to 32 (16 pairs) endpoints can reside within a device
  - All USB transfers are targeted to/from a device endpoint
  - An endpoint is a buffer used to transmit or receive data
- Each endpoint has a direction and an address
  - The direction is from the host's perspective
    - OUT transactions are to the device
    - IN transactions are from the device
- A Control endpoint contains an IN and OUT endpoint
  - Endpoint 0 is always the Control endpoint

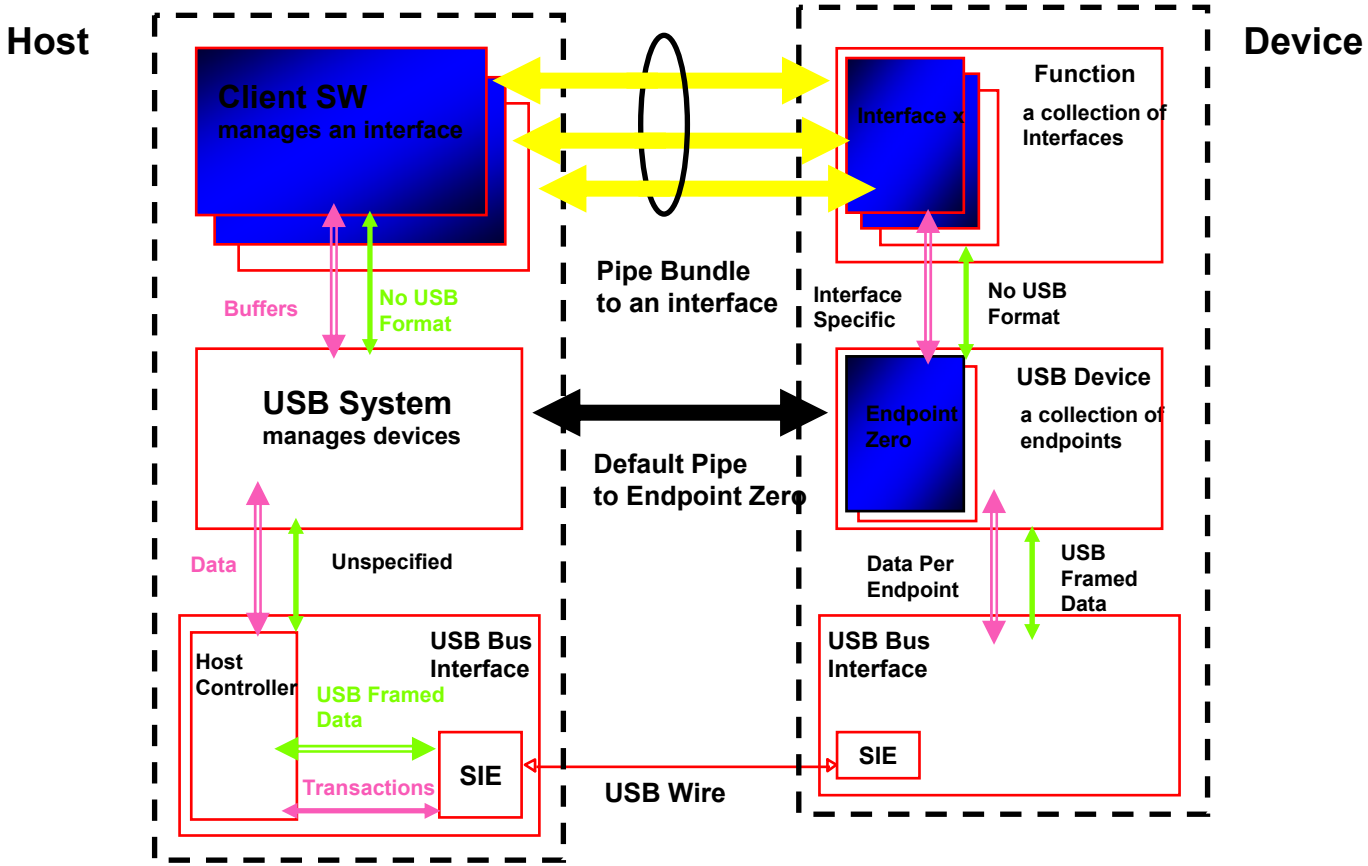
Interfaces

- Made of 0 or more pipes
- Has a client owner
  - Accesses individual pipes
  - Shares default pipe
- More dynamically configured than devices

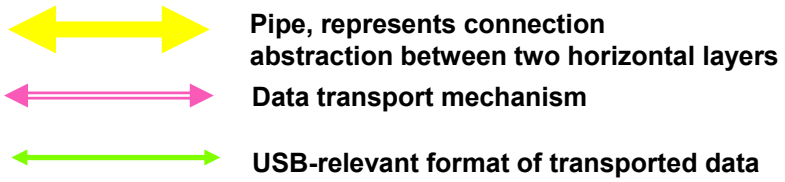
Pipes

- Connect host memory buffer to endpoint FIFO
- Stream Type
  - No USB imposed data format
  - Unidirectional
- Message Type
  - USB imposed data format
  - Bidirectional

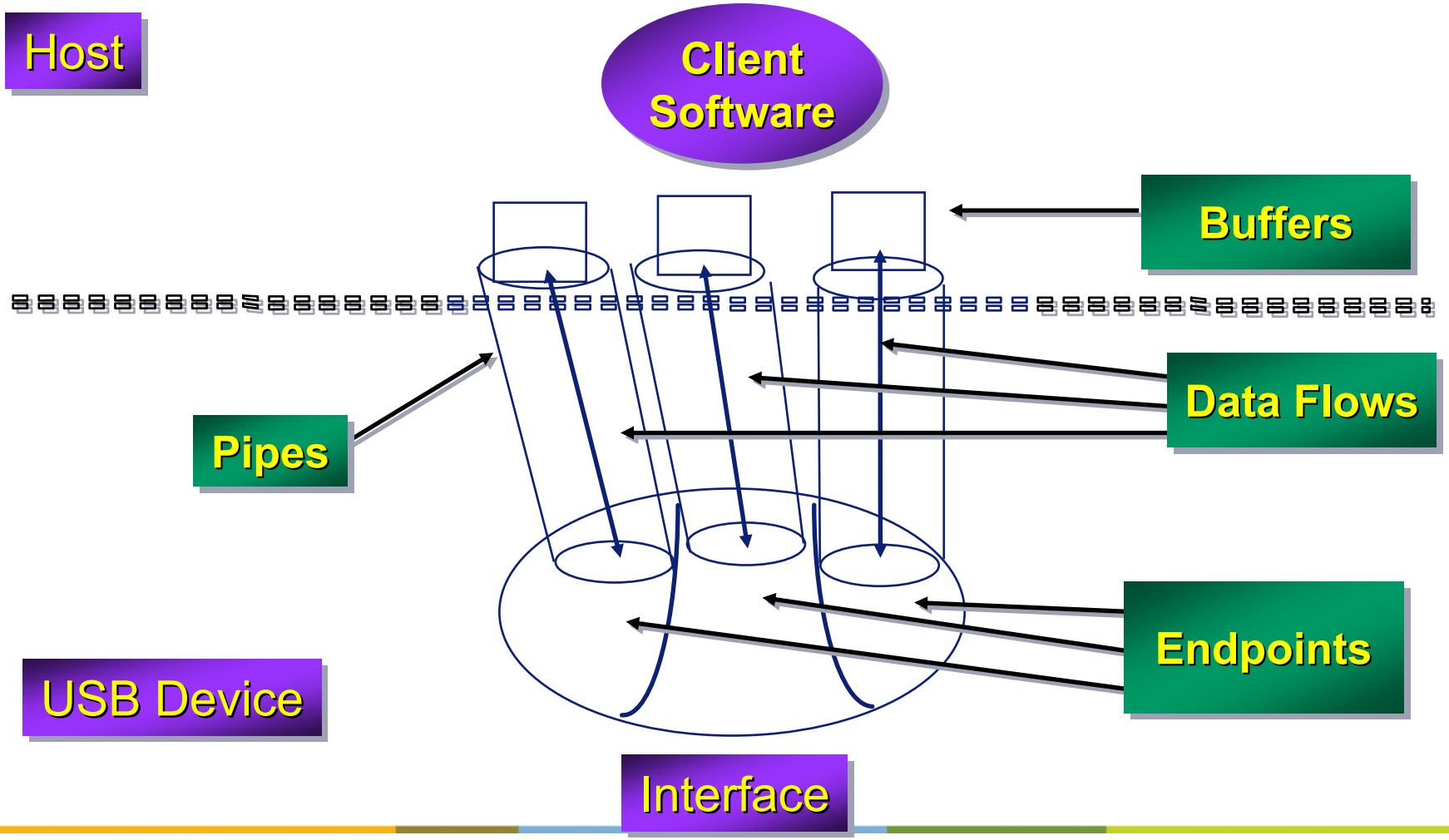
Detailed Host / Device View



Endpoint 0
- **Required, shared**
- **Configuration access**
- **Capability control**



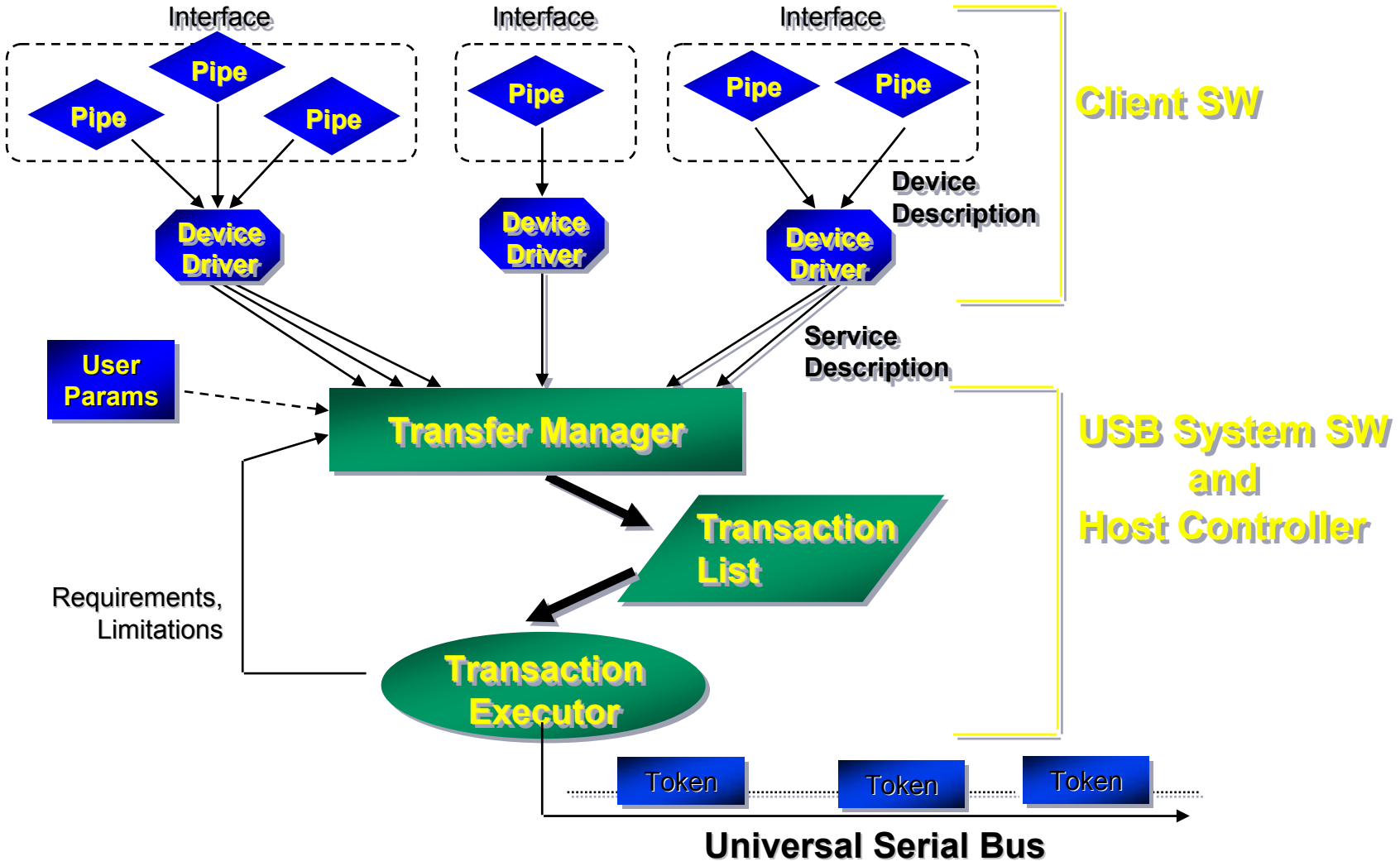
Client Software <-> Function



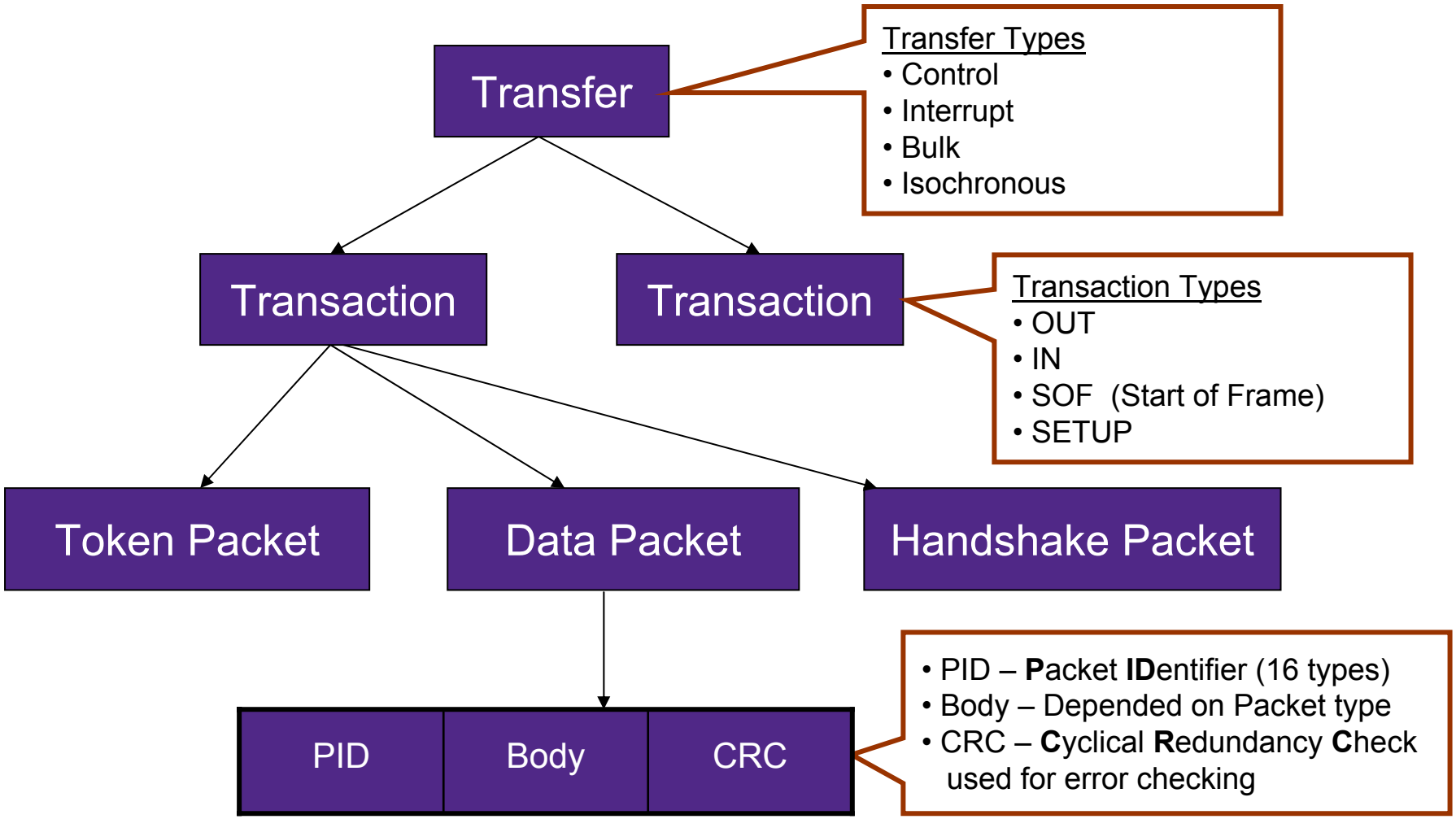
Communications Layers

- Physical
- Packets
- Transactions
  - 3 phases (token, data, handshake)
  - Token phase has token packet sent by host
    - Always present
    - Packet ID (PID) identifies transaction type
  - Other phases have 0 or more packets
- Transfers

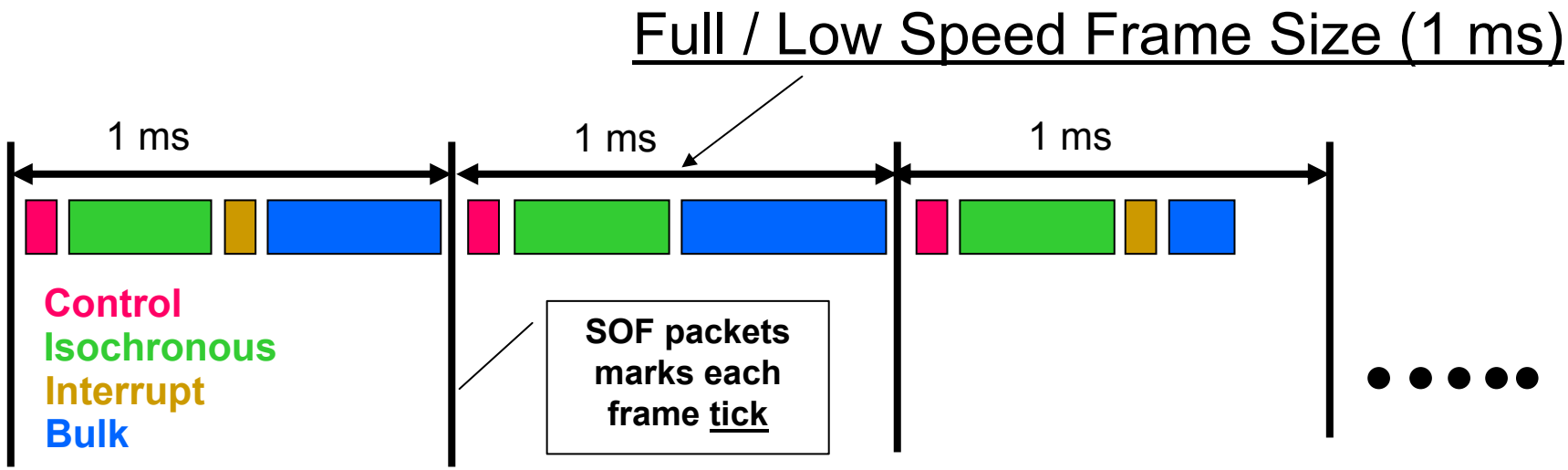
Transfers to Transactions



USB Protocol – Breakdown



Frames

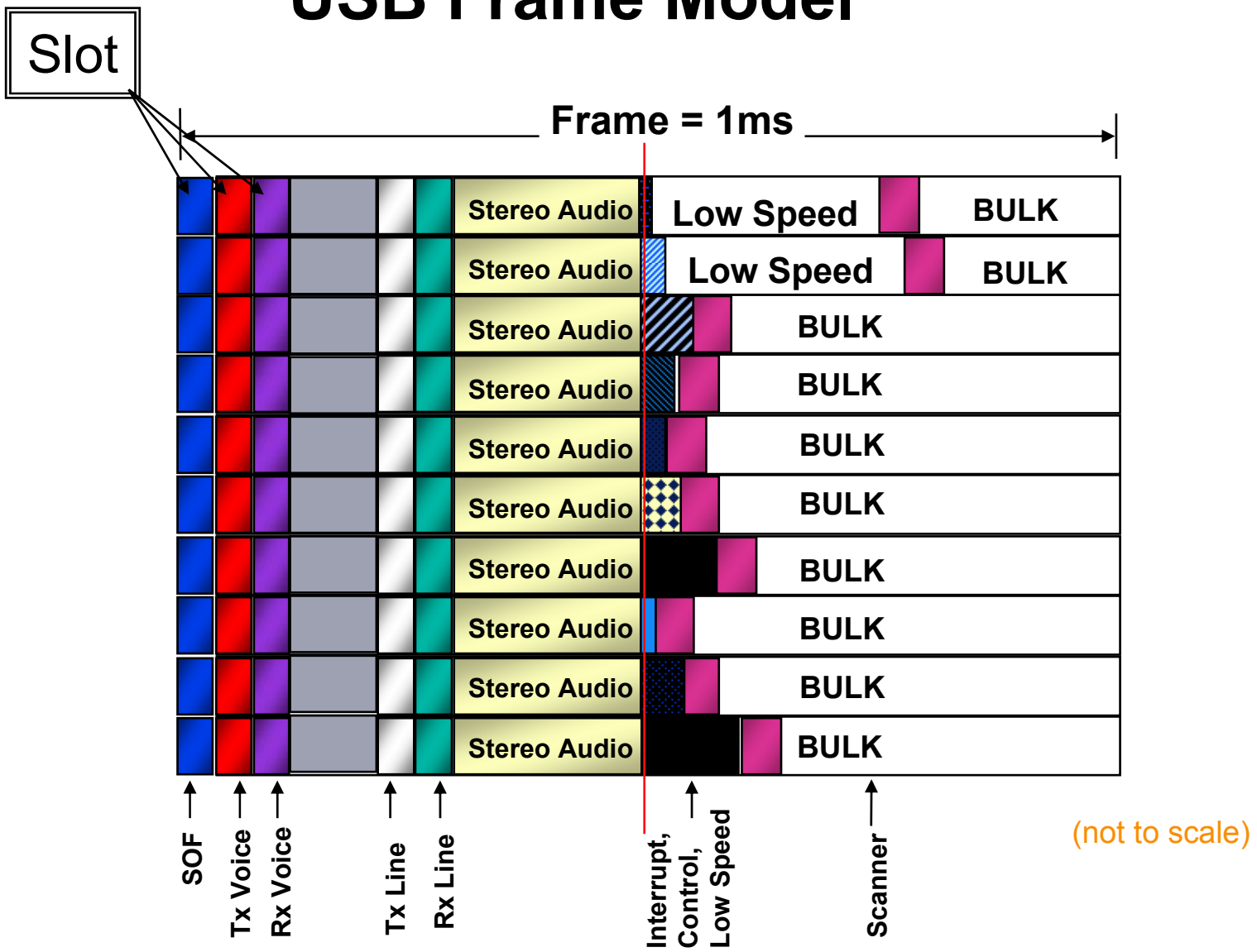

Full / Low Speed Frame Size (1 ms)
1 ms
1 ms
1 ms
Control
Isochronous
Interrupt
Bulk
SOF packets marks each frame tick

Data Transfer Types Comparison

Usage	Control	Interrupt	Bulk	Isochronous
Max Data bytes/msec in low-speed mode	24 bytes (three 8-byte transactions)	0.8 bytes (8-bytes per 10 msec)	no support	no support
Max Data bytes/msec in full-speed mode	832 (thirteen 64-byte transactions/frame)	64 (one 64-byte transaction/frame)	1216 (nineteen 64-byte transactions/frame)	1023 (one 1023-byte transaction/frame)
Error Correction	yes	yes	yes	no
Guarantee rate of delivery	no	no	no	yes
Guranteed time between transfers	no	yes	no	yes
Typical uses	Configuration	Mouse, keyboard	Printer, scanner	Audio, video

Note:
frame = 1msec @FS

USB Frame Model



Transactions

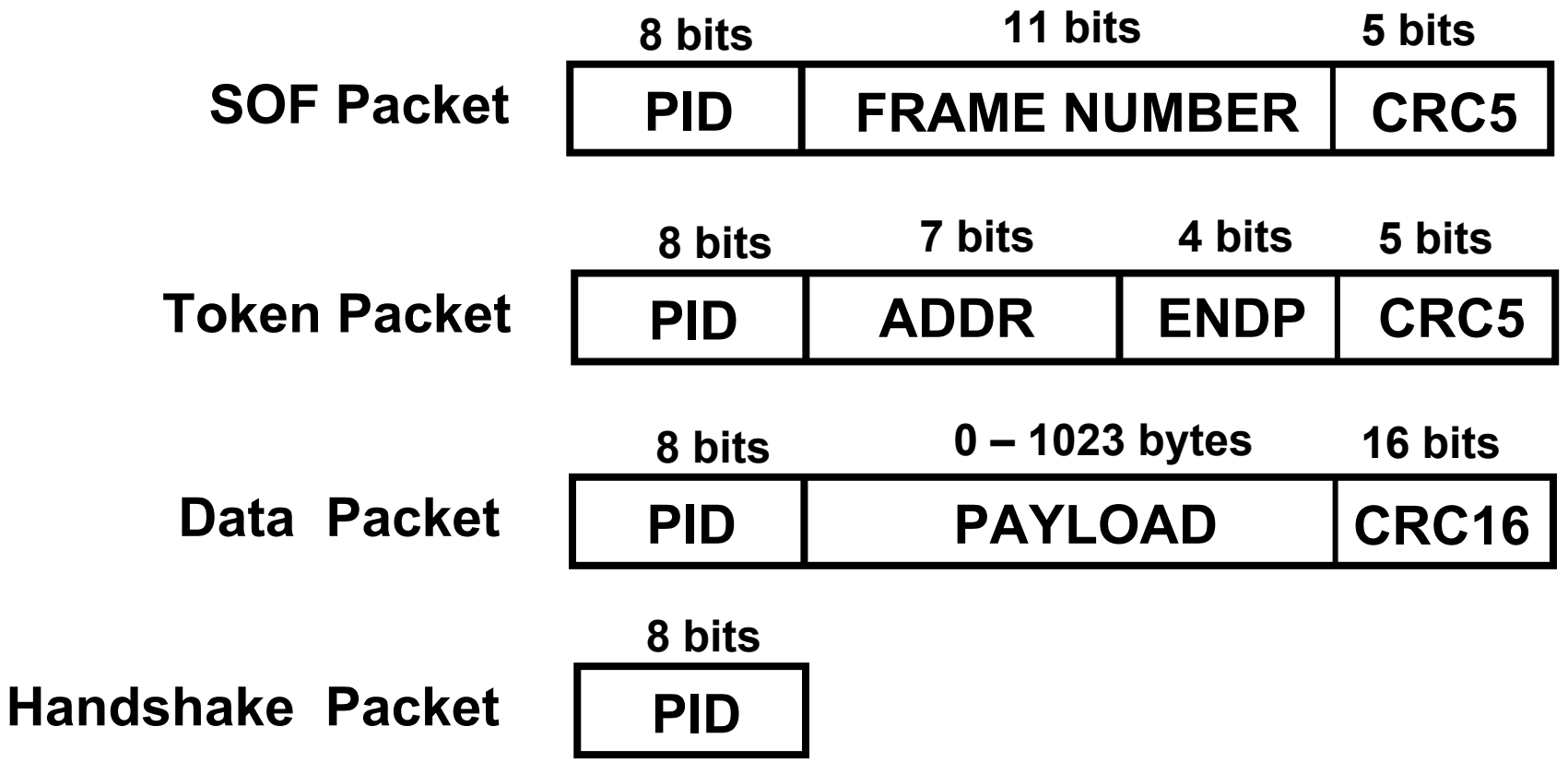
- Transactions are always initiated by the Host and therefore thought of from the Host's perspective

- A Transaction consist of multiple packets and begin when the host sends one of four Token Packets
  - OUT – notifies the DC that data is being send "out" from the Host
  - IN – requests that DC send data "in" to the Host
  - SOF – signals the "Start of Frame"
  - SOF – signals the "Start of Frame"

- A Data Packet follows the token packet

- The Transaction ends with Handshake Packet to report the Status of the Transaction.

Packet Formats



Endpoints in LPC2148

- Maximum of 32 (16 logical) endpoints
- Selectable Double Buffering for Bulk and Isochronous Data Transfer
- Any combinations of Endpoints allowed
- Maximum buffer size supported for all endpoints types

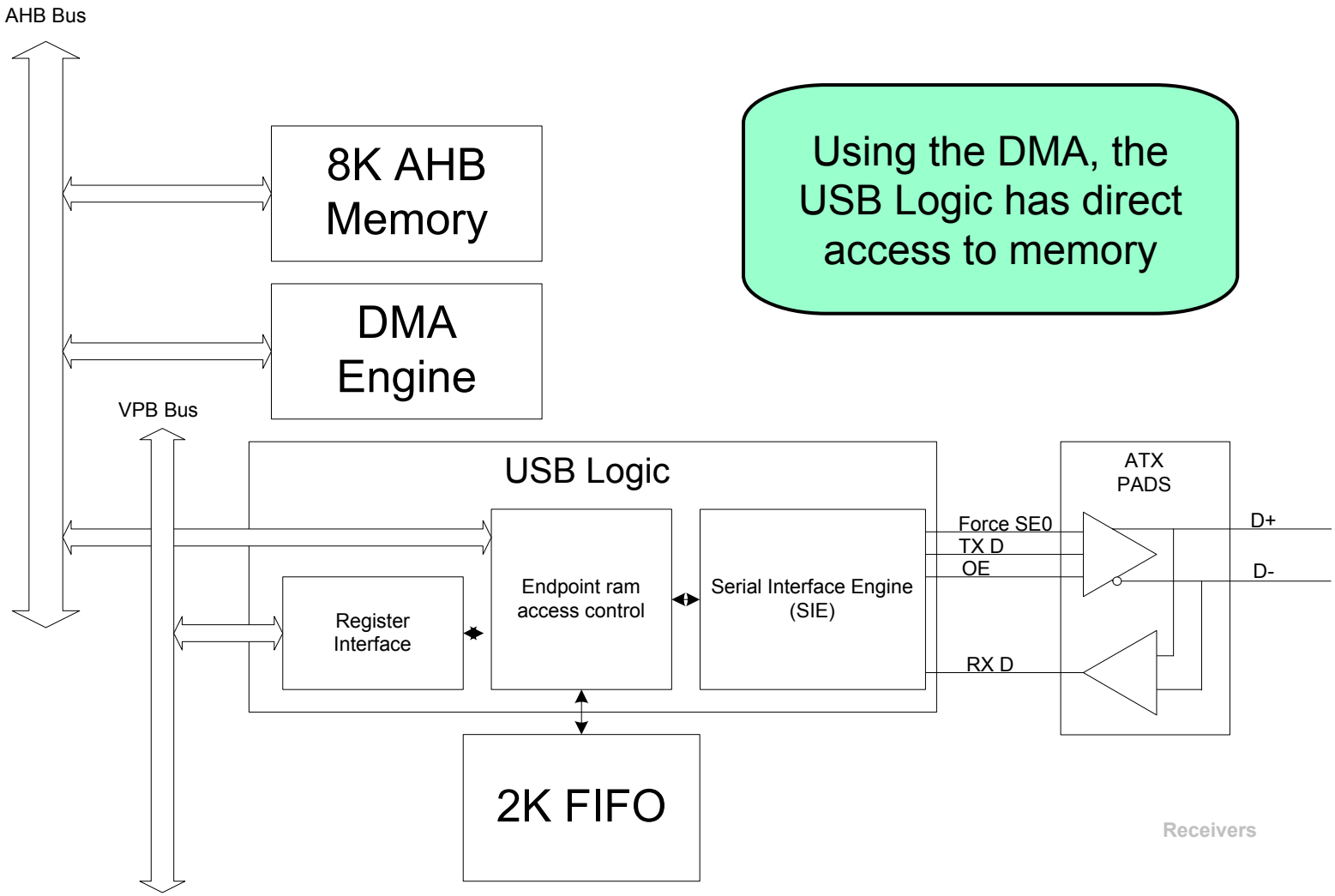
USB 2.0 in LPC2148

- Fully Compliant with USB 2.0 Spec
- Supports 32 physical endpoints
- Scalable realization of Endpoints during run time
- Double buffering supported for Bulk and Isochronous Endpoints
- Supports DMA transfer on all non-control endpoints

USB 2.0 in LPC2148

- Supports Control, Bulk, Interrupt and Isochronous endpoints
- Supports SoftConnect feature
- Supports Good link LED indicator
- Flexible clock architecture
- Remote wakeup

LPC2148 USB Block Diagram



Using the DMA, the USB Logic has direct access to memory

Transfer Modes

- Slave Transfer Mode
- DMA Transfer Mode

Slave Mode Transfer

- USB Block acts like a slave
- It can only issue interrupts to the CPU
- Control EP uses this mode of transfer exclusively

DMA Mode Transfer

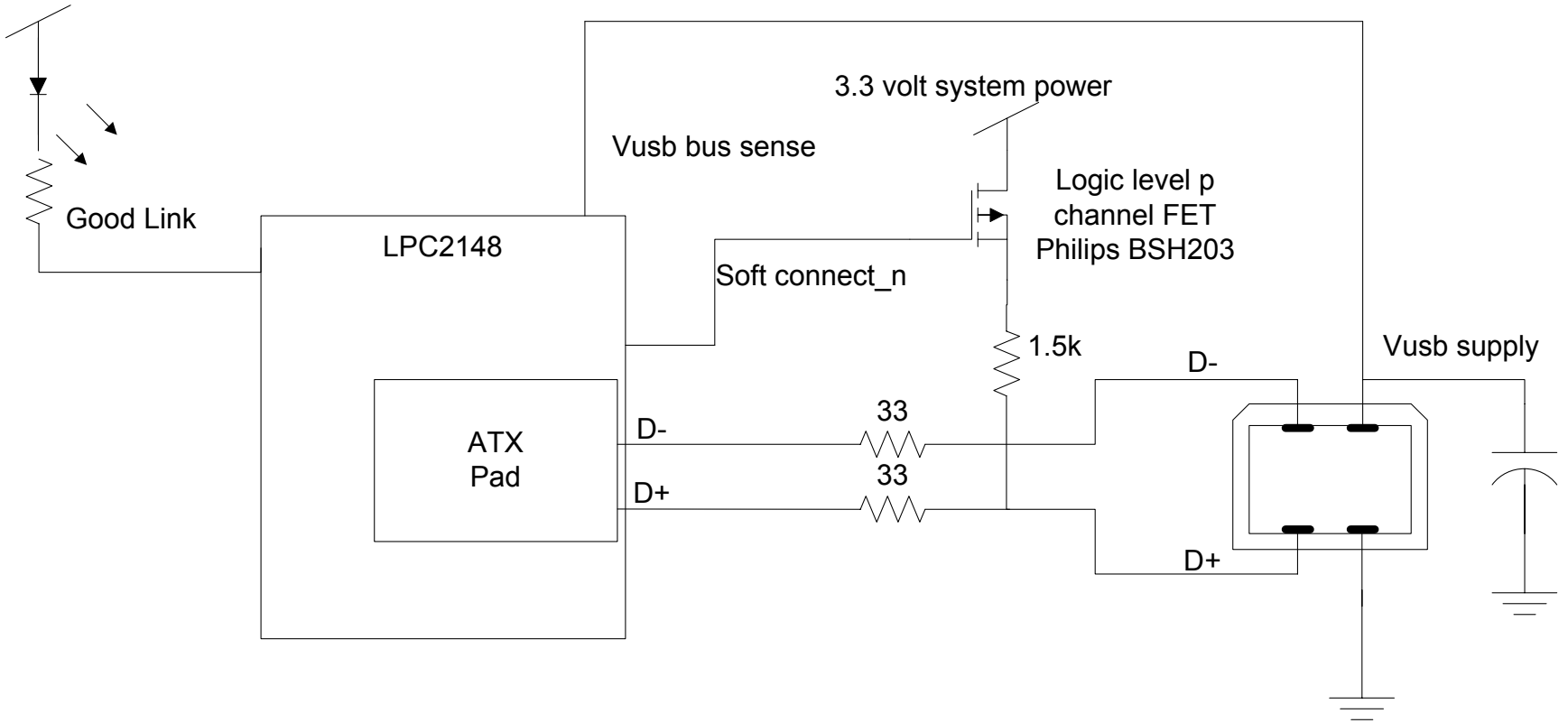
- USB Block acts like a Master
- Will transfer data directly from the 8K SRAM to the EP_RAM and vice versa.
- For Isochronous transfers, the DMA transfer is synchronized to the frame interrupt

Soft Connect and Good Link™ LED

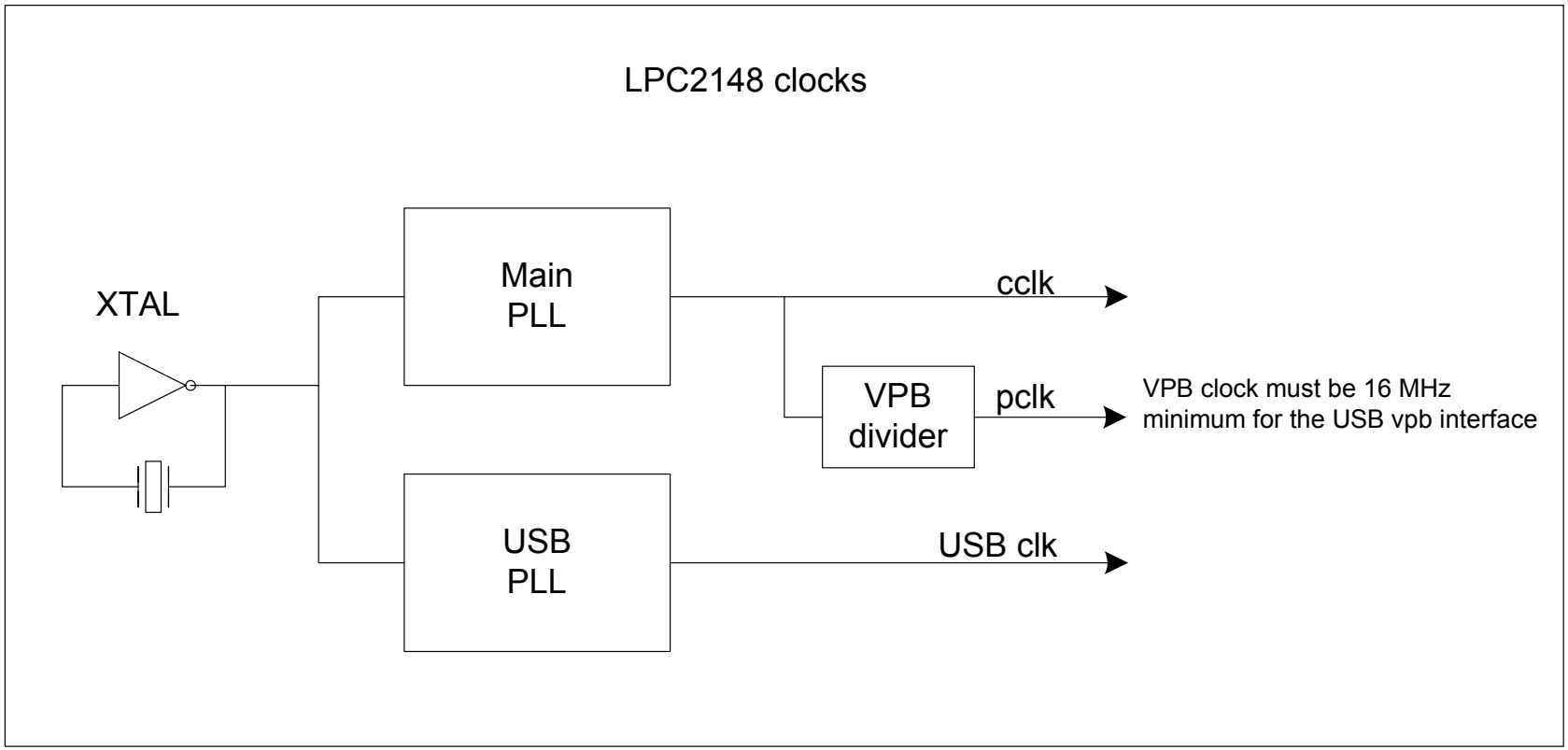
- Soft Connect
  - Can connect/re-connect to the host through software
  - No need to unplug and plug the cable back again

- Good Link™ LED
  - Needs a shared GPIO pin
  - Shows indication on a LED if connection is established

LPC2148 connections



LPC2148 Clock Architecture

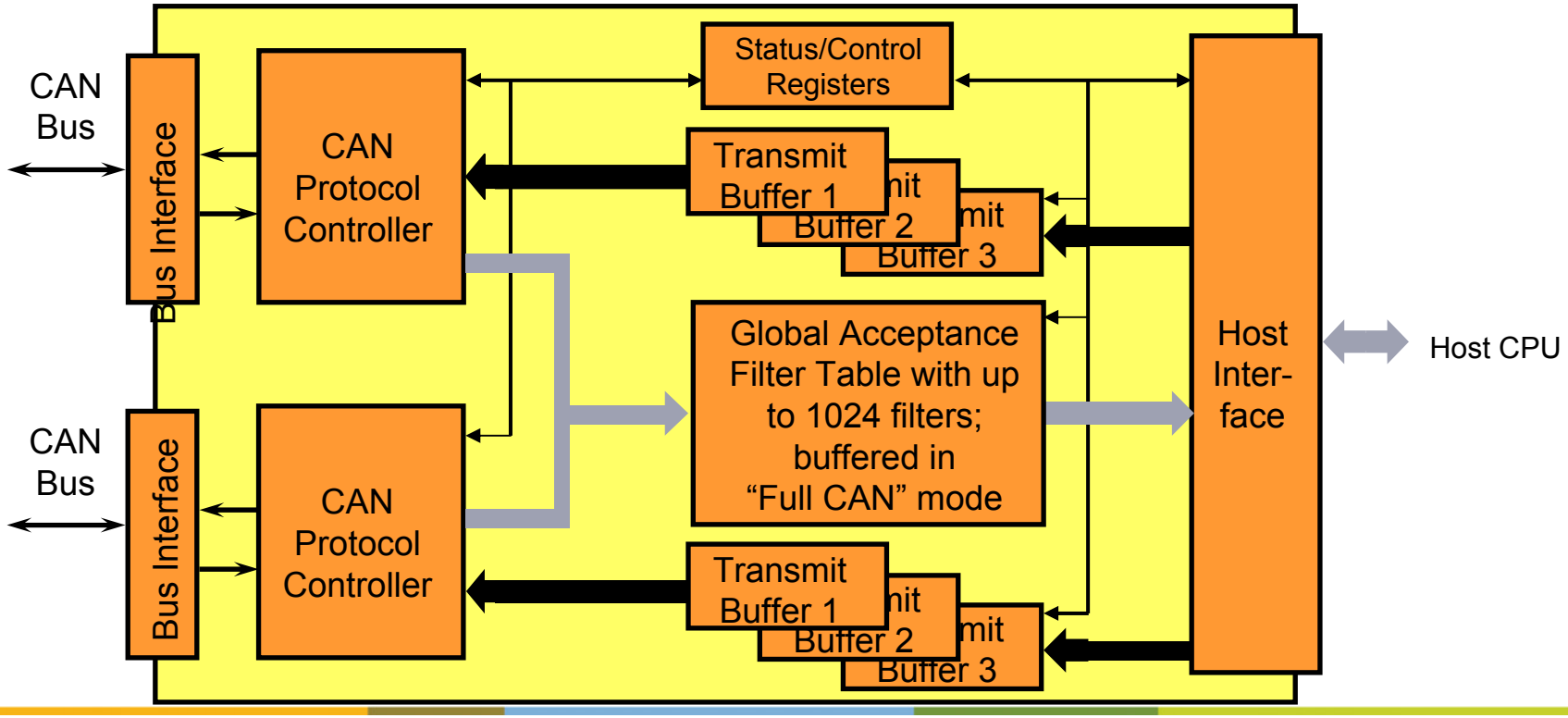


CAN

NXP LPC2000 Series CAN Controller

Extended Full CAN Controller

- 2 or 4 CAN interfaces
- 3 priority controlled transmit buffers per channel
- Global filter and buffer system with up to 1024 filters



LPC CAN controller and SJA1000

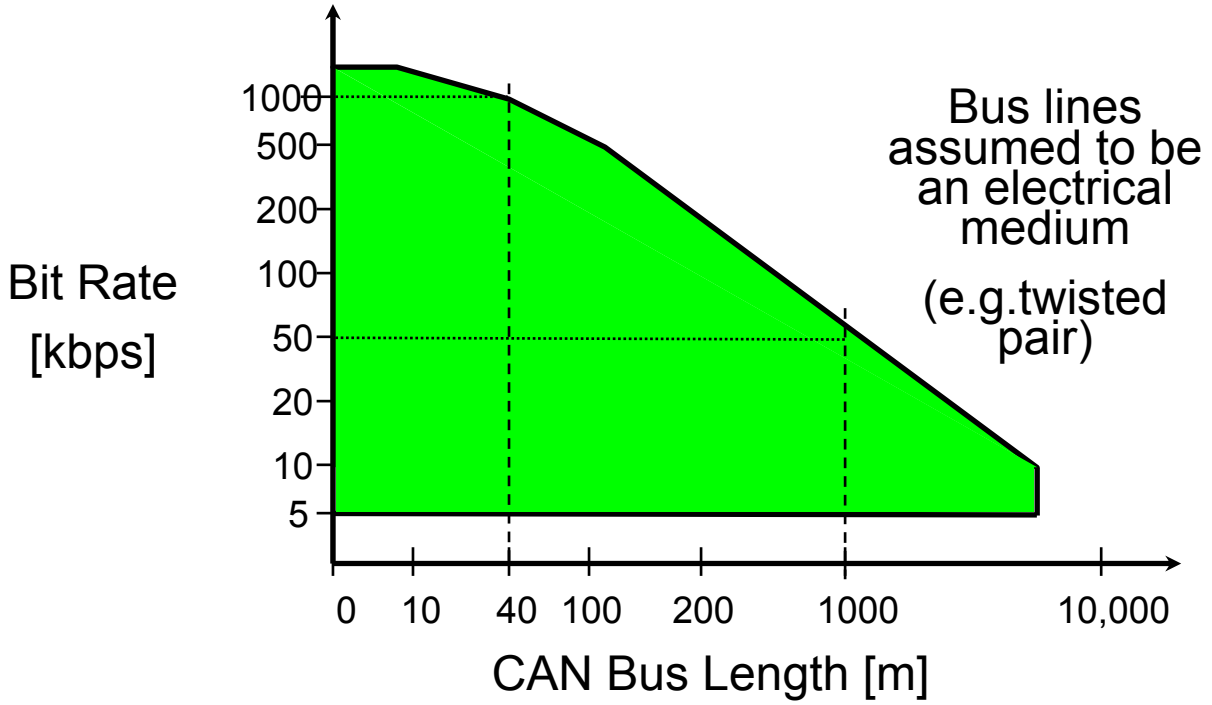
- Each CAN Controller has a register structure similar to the NXP SJA1000 and the PeliCAN Library block, but the 8-bit registers of those devices have been combined in 32 bit words to allow simultaneous access in the ARM environment. The main operational difference is that the recognition of received Identifiers, known in CAN terminology as Acceptance Filtering, has been removed from the CAN controllers and centralized in a global Acceptance Filter.

CAN Bus Baud Rate and Bus Length

- Up to 1Mbps @ 40m bus length (120 feet)

  OR

- Up to 1000m bus length (3000 feet) @ 50 kbps



CAN Bus (1)

- CAN **(Controller Area Network)** features
  - Serial communications protocol
  - Efficiently supports distributed real-time control
  - Very high level of security
  - Application domain: high speed networks to low cost multiplex wiring
- LPC2xxx with CAN have 2 or 4 CAN channels
  - Can be used as gateway, switch or router among CAN buses
  - Industrial or automotive applications

CAN Bus (2)

- Data rates to 1 Mbit/s on each bus
- 32-bit register and RAM access
- Compatible with CAN specification 2.0B, ISO 11898-1
- Global Acceptance Filter recognizes 11- and 29-bit Rx Identifiers for all CAN buses
- Acceptance Filter can provide FullCAN-style automatic reception for selected Standard Identifiers
- No. of CAN channels
  - 2 on LPC2119, LPC2129, LPC2290, LPC2292
  - 4 on LPC2194, LPC2294

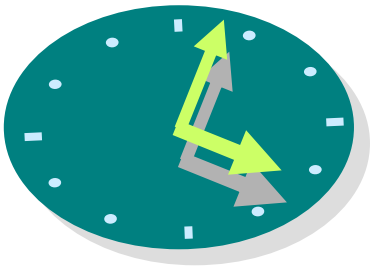
CAN Bus (3)

Acceptance Filtering

- Recognition of received Identifiers (Acceptance Filtering) removed from CAN controllers - now centralized in a global Acceptance Filter
- In a 2KB RAM (512 x 32) software maintains 1 ... 5 tables of Identifiers RAM can hold up to 1024 Standard Identifiers or 512 Extended Identifiers, or a mixture of both types

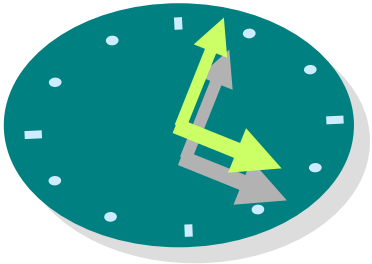
- TIMERS
- PWMs
- RTC
- WATCH-DOG

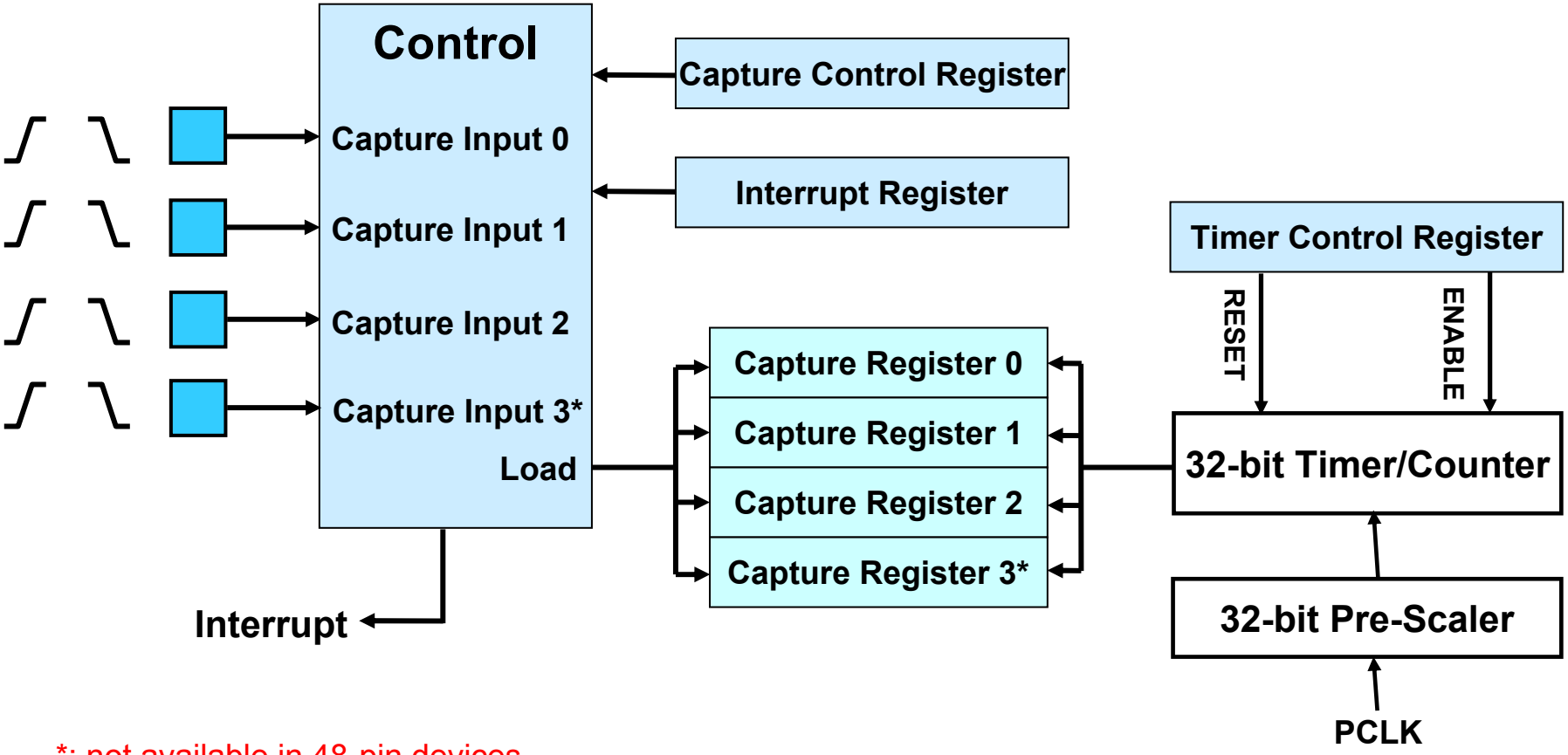
Timer 0 and 1



- 32-bit Timer
- 32-bit Capture Registers and Capture Pins
  - Four on each timer (48-pin devices three on Timer 0 and four on Timer 1)
  - Capture event can optionally trigger an interrupt
- 32-bit Match Registers and Match Pins
  - Four on each timer (48-pin devices three on Timer 0 and four on Timer 1)
  - Interrupt, timer reset or timer halt on match
  - Match output can toggle, go high, go low or do nothing

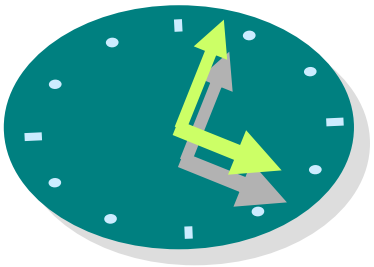
Timer Capture





*: not available in 48-pin devices

Timer Compare

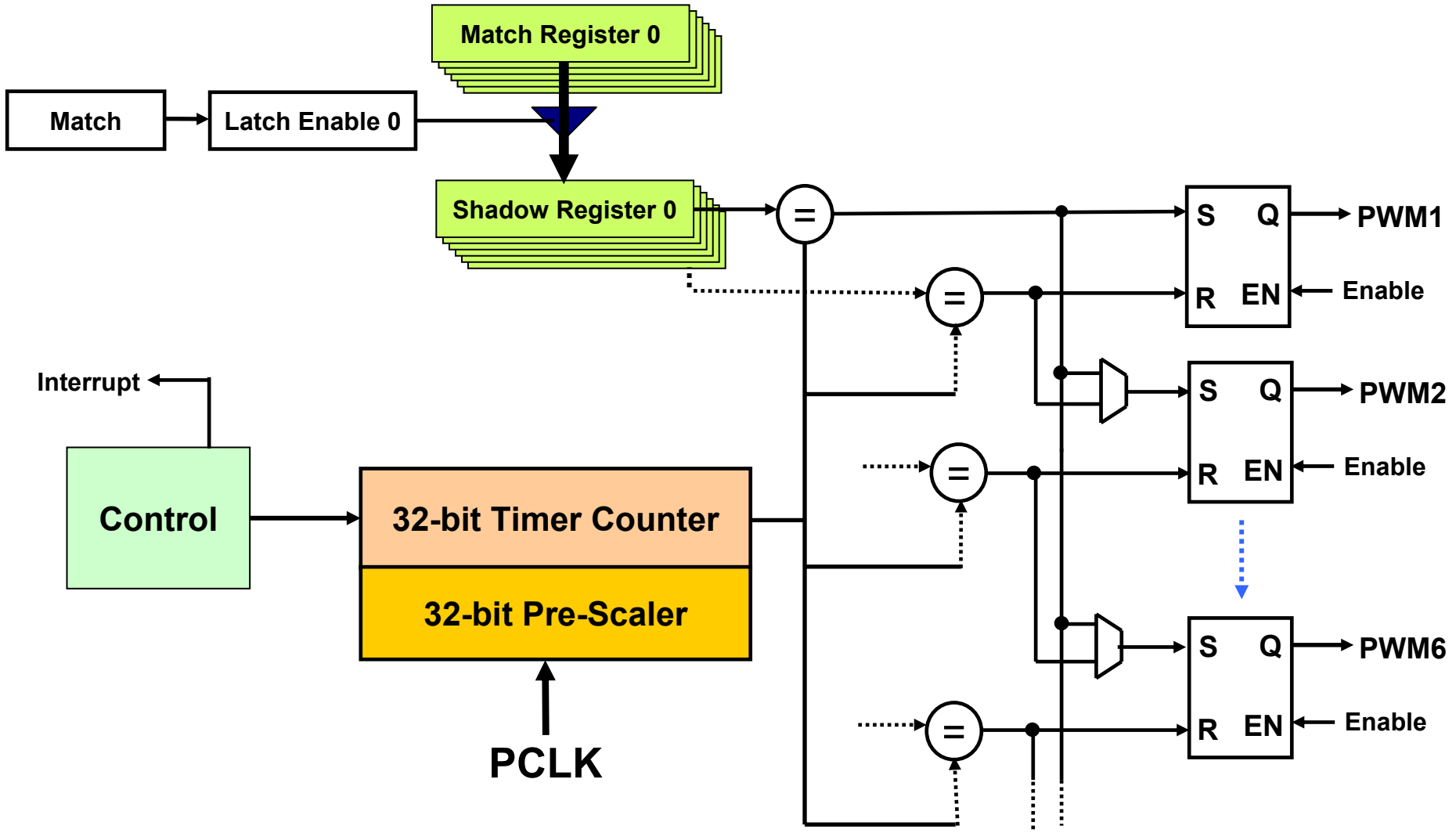


*: not available in 48-pin devices

Pulse Width Modulator

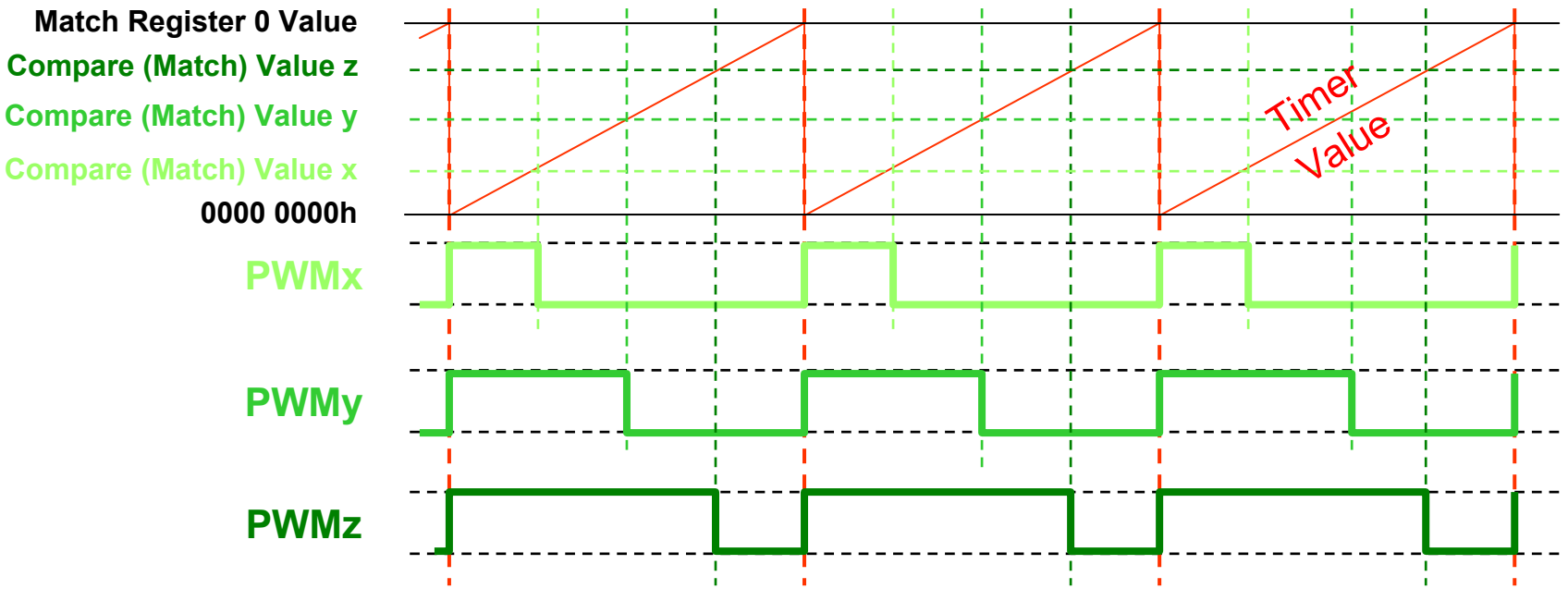
- Dedicated 32-bit PWM timer
  - similar functionality to Timer0 / Timer1
- Three additional match registers for a total of 7
  - all PWM outputs have the same rate, which is programmable
  - allows up to 6 single edge controlled or 3 double edge controlled PWM outputs in any combination
  - single edge controlled PWM outputs all go high at the beginning of each cycle and low at a programmed time
  - double edge controlled PWM outputs can be programmed to be either positive going or negative going pulses, with edges at any location in the cycle

Pulse Width Modulator



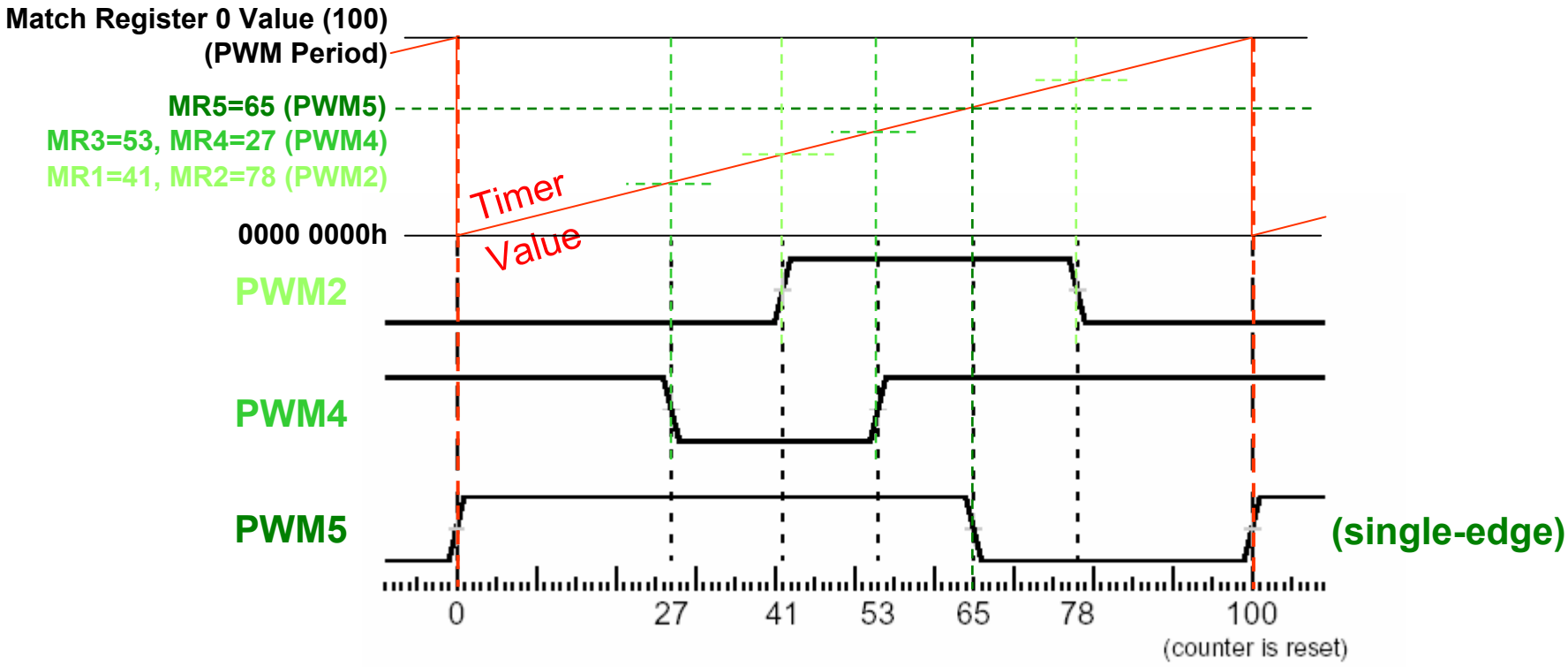
Single-Edge Controlled PWM

- PWM outputs all go high at the beginning of each cycle and go low on a Match



Double-Edge Controlled PWM

- Double edge controlled PWM outputs can have either edge occur at any position within a cycle



Hands-On Index

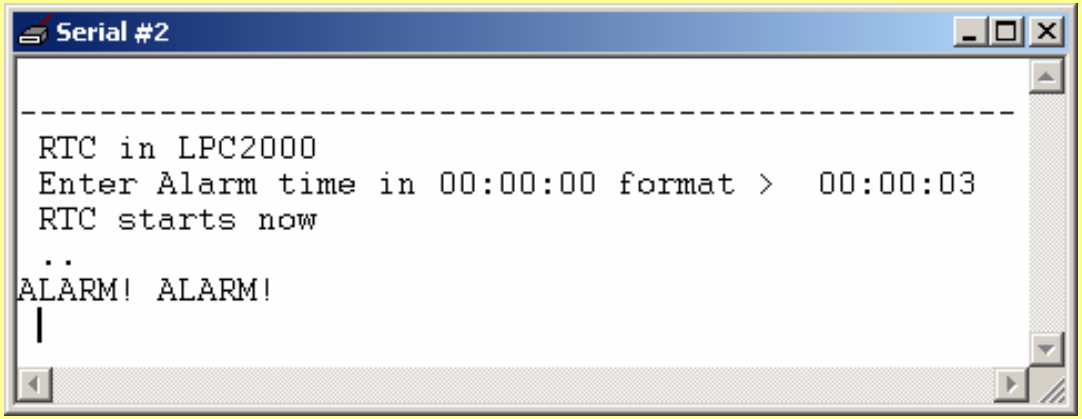
Hands-On

1. Tools Setup
2. Oscillator / PLL / MAM / GP I/O
3. ADC
4. Interrupts / Timer
5. UART / CAN

Project: SerialRTC

Hands-On

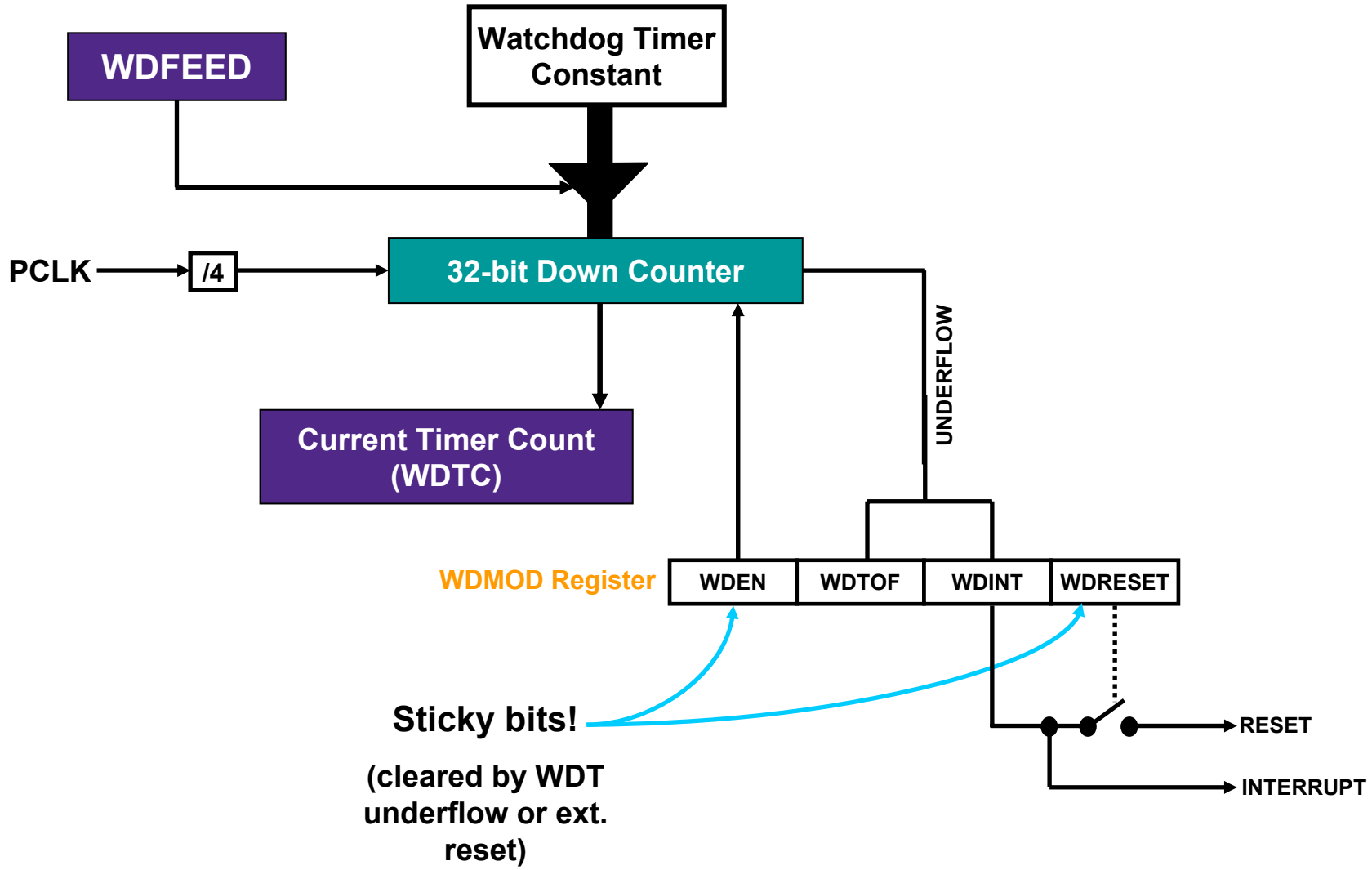
- Load Project SerialRTC
- Build and Debug
- Simulation
  - Use Serial Window #2
- Target Hardware
  - Use terminal program set to 9600bps



Watchdog Timer

- Once activated, the Watchdog will reset the entire chip if it is not fed regularly
- Feed is accomplished by a specific sequence of data writes
- Watchdog flag allows software to tell that a watchdog reset has occurred
- Selectable overflow time (µs ... minutes)
- Debug Mode generates an interrupt instead of a reset
- Secure: watchdog cannot be turned off once it is enabled
- Watchdog Timer value can be read in one cycle

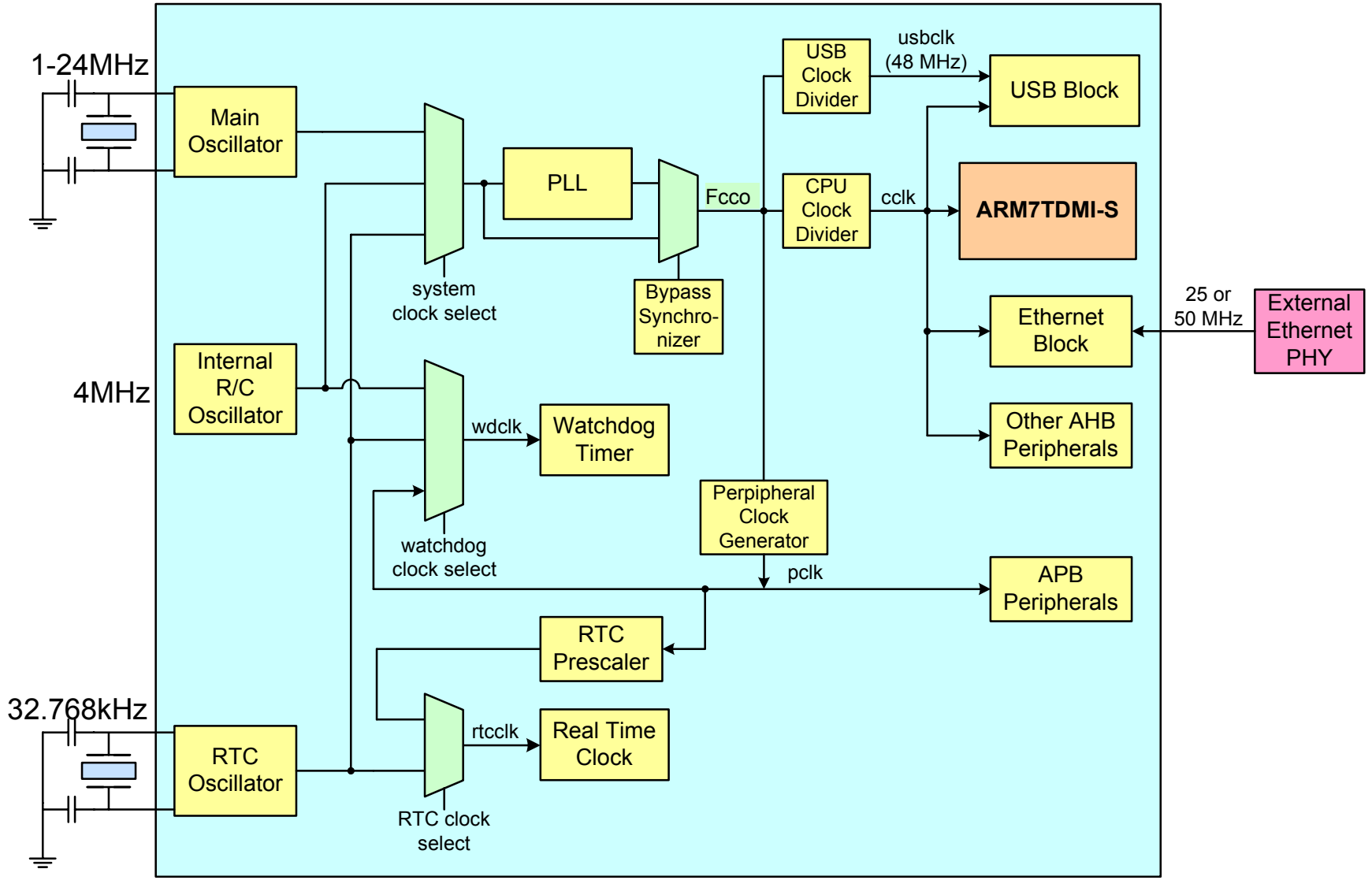
Watchdog Timer



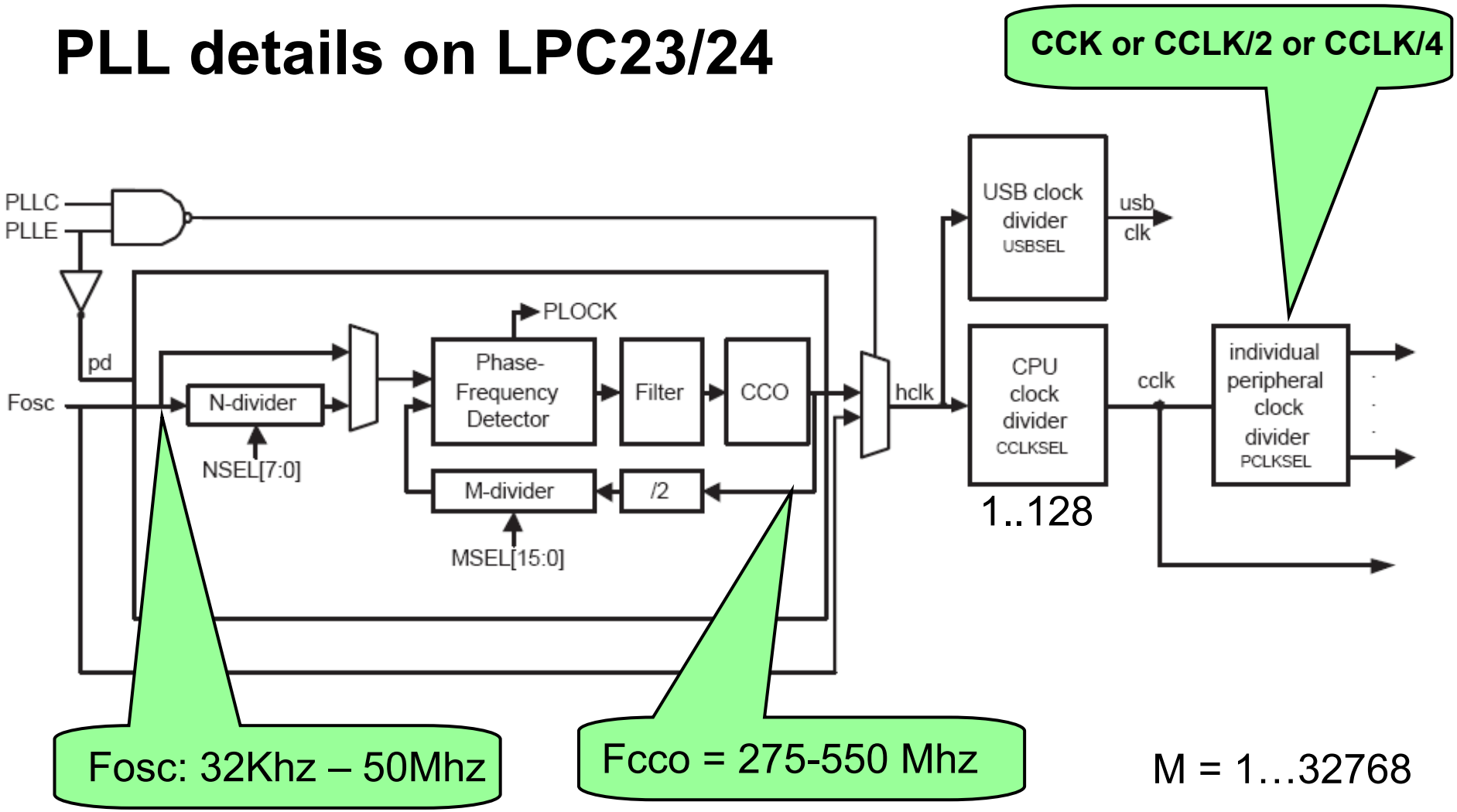
NXP LPC2300/2400
Selected Peripherals

Clock tree

Clocking (only for LPC23/24 families)



PLL details on LPC23/24



M = 1…32768

N = 1..256

LPC23/24 clocking

- The LPC2300 has three clock sources
  - External oscillator 1Mhz – 24 Mhz
  - External watch crystal 32.756 Khz
  - Internal RC oscillator approx 4 Mhz
- The LPC2300 has three clock sources Fcco = (2 x M x Fin)/N
  - Were 32Khz < Fin < 50Mhz
  - Were 275Mhz < Fcco < 550Mhz
  - Also Fcco must be kept to a minimum to reduce power consumption
- – The USB peripheral requires a precise 48Mhz clock source

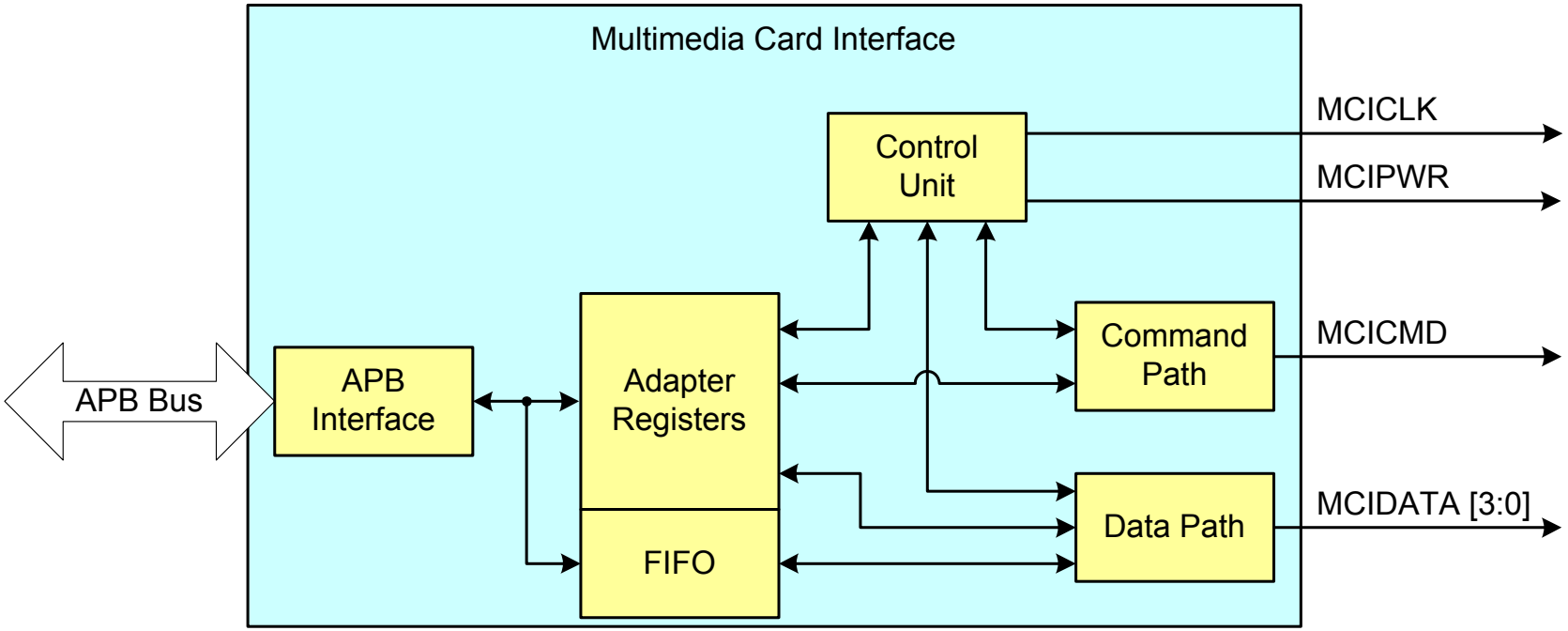
NXP LPC2300/2400
Selected Peripherals

LPC2300 Peripherals – SD/MMC

LPC2300 Peripherals – SD/MMC

- Conformance to Multimedia Card Specification v2.11.
- Conformance to Secure Digital Memory Card Physical Layer Specification, v0.96.
- Use as a multimedia card bus or a secure digital memory card bus host. It can be connected to several multimedia cards, or a single secure digital memory card.
- DMA supported through the General Purpose DMA Controller.

LPC2300 Peripherals – SD/MMC



LPC2300 Peripherals – BASIC TRAINING SD/MMC/SDIO

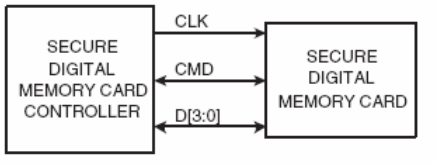
Technical comparison

Type	MMC	RS-MMC	MMC Plus	SecureMMC	SD	SDIO	miniSD	microSD
SD Socket	Yes	Mechanical adapter	Yes	Yes	Yes	Yes	Electro-mechanical adapter	Electro-mechanical adapter
Pins	7	7	13	7	9	9	11	8
Form factor	Thin	Thin/short	Thin	Thin	Thick (exceptions possible)	Thick	Narrow/short/thin	Narrow/short/extrathin
Width	24 mm	24 mm	24 mm	24 mm	24 mm	24 mm	20 mm	11 mm
Length	32 mm	18 mm	32 mm	32 mm	32 mm	32 mm+	21.5 mm	15 mm
Thickness	1.4 mm	1.4 mm	1.4 mm	1.4 mm	2.1 mm (exceptions possible)	2.1 mm	1.4 mm	1 mm
SPI mode	Optional	Optional	Optional	Required	Required	Required	Required	Optional
1 bit mode	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
4 bit mode	No	No	Yes	?	Optional	Optional	Optional	Optional
8 bit mode	No	No	Yes	?	No	No	No	No
Xfer clock	0–20 MHz	0–20 MHz	0–52 MHz	0–20 MHz?	0–25 MHz - 0–50 MHz	0–25 MHz	0–25 MHz?	0–25 MHz?
Max XFER	20 Mbit/s	20 Mbit/s	416 Mbit/s	20 Mbit/s?	100 Mbit/s - 200 Mbit/s	100 Mbit/s	100 Mbit/s	100 Mbit/s
Max SPI XFR	20 Mbit/s	20 Mbit/s	52 Mbit/s	20 Mbit/s	25 Mbit/s	25 Mbit/s	25 Mbit/s	25 Mbit/s
DRM	No	No	No	Yes	Yes	N/A	Yes	Yes
User encrypt	No	No	No	Yes	No	No	No	No
Simplified Spec	Yes	Yes	No	Not yet?	Yes	Yes	No	No
Memb cost	$2500/yr (not required)				$1500/yr (appears required)			
Spec cost	$500		?	?	Member	Member	Member	Member
Host license	No	No	No	No	$1000/yr+memb			
Mem card royalties	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
I/O card royalties	N/A	N/A	N/A	N/A	N/A	$1000/yr+memb	N/A	N/A
Open source compatible	Yes	Yes	Yes?	Yes?	SPI only	SPI only	SPI only	SPI only
Type	MMC	RS-MMC	MMC Plus	SecureMMC	SD	SDIO	miniSD	microSD

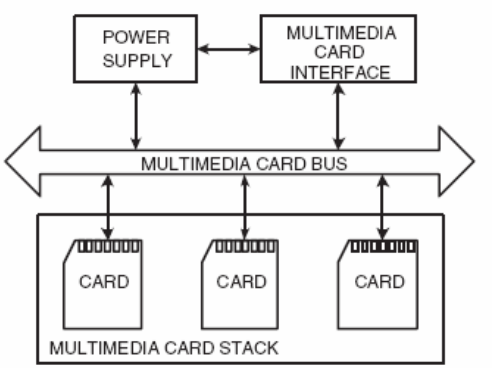
LPC2300 Peripherals – SD/MMC

SD/MMC memory card interface (LPC2368 & LPC2378)

- Conformance to Multimedia Card Specification v2.11.
- Conformance to Secure Digital Memory Card Physical Layer Specification, v0.96.
- Use as a multimedia card bus or a secure digital memory card bus host.
- It can be connected to several (~4 based on I/O pin loading) multimedia cards, or a single secure digital memory card.
- DMA supported through the General Purpose DMA Controller.

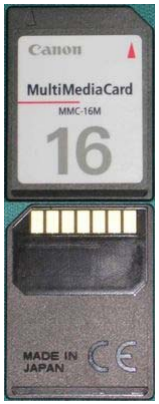


OR



LPC2300 Peripherals – SD/MMC BASIC TRAINING

- SD “Secure Digital” (LPC23xx = 25Mbit/s)
  - Developed as improvement on MMC
  - Up to 128 Gbyte per card
  - Low speed up to 400Kbit/s
  - High speed up to 100Mbit/s



- MMC “Multi-Media Card” (LPC23xx – 20Mbit/s)
  - 1, 4, or 8 bits per interface
  - Up to 8Gbyte per card
  - Slightly thinner than SD cards
  - Pin-compatible with SD cards

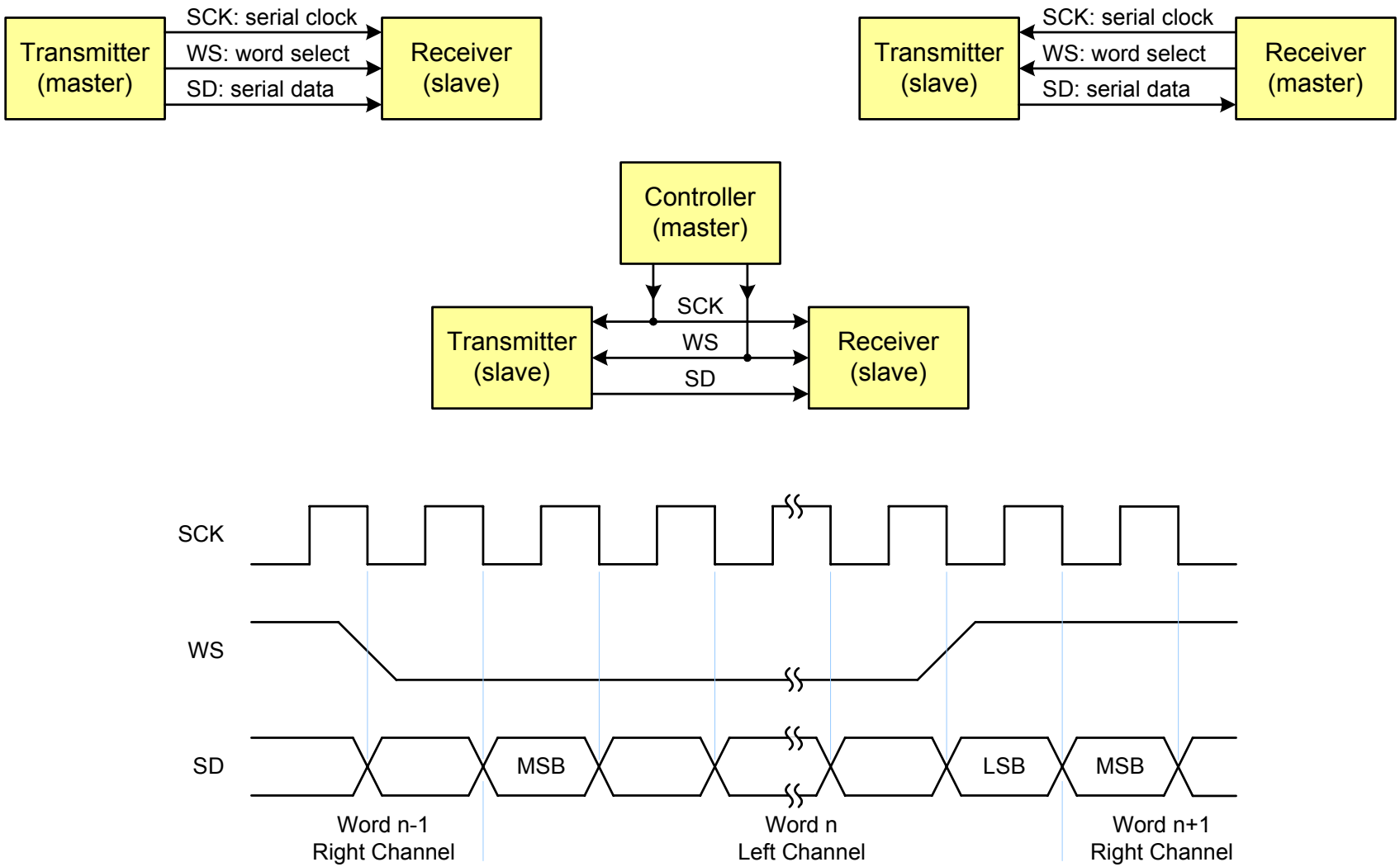
- SDIO – small devices that use the SD physical format for other functions beyond storage
  - GPS, WiFi, BlueTooth, Modems, FM Radio, RFID, Barcode, etc., etc.
  - Additional interconnect functionality required
  - May require interrupt line (SD interface does not provide)

- http://en.wikipedia.org/wiki/Secure_Digital_card

I²S Features

- The I²S input and output can each operate independently in both master and slave mode.
- Capable of handling 8, 16, and 32 bit word sizes.
- Mono and stereo audio data supported.
- The sampling frequency can range (in practice) from 16-48 kHz.
- Word Select period in master mode is configurable (separately for I²S input and I²S output).
- Transmit and receive functions each have an 8 byte data FIFO.
- Programmable FIFO level interrupts.
- Two DMA requests, controlled by programmable FIFO levels.
- Controls include reset, stop and mute options separately for I²S input and I²S output.

I²S Diagrams

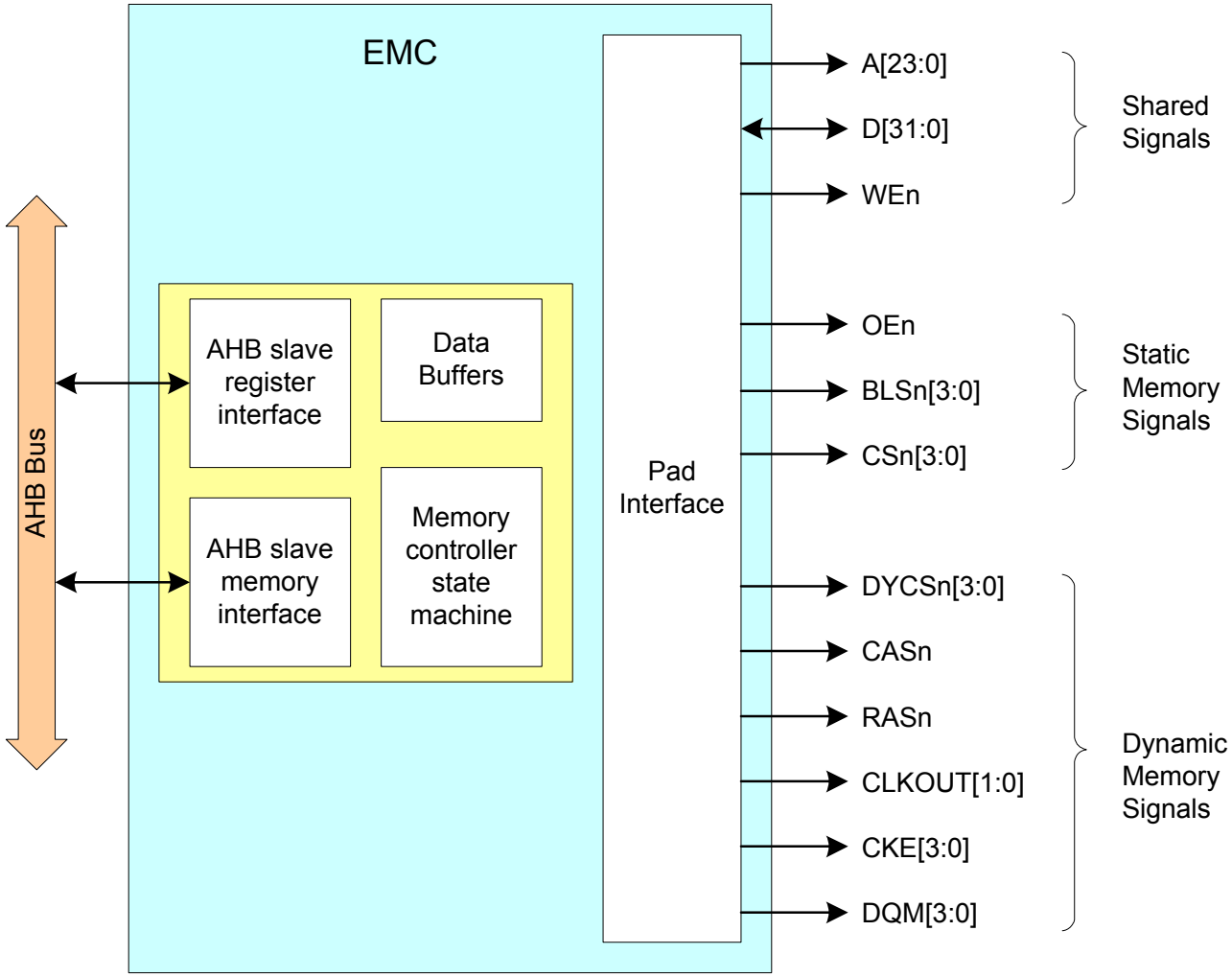


NXP LPC2300/2400
Ethernet

External Memory Controller, ARM PrimeCell™

- SDR SDRAM memory support.
- Asynchronous static device support including RAM, ROM, and Flash.
- Low transaction latency.
- Read and write buffers reduce latency and improve performance.
- 8-bit, 16-bit, and 32-bit wide static memory support.
- Static memory features include:
  - Asynchronous page mode read.
  - Programmable wait states.
  - Bus turnaround delay.
  - Output enable, and write enable delays.
  - Extended wait.
- Four chip selects each for SDRAM and static devices.
- Power-saving modes dynamically control CKE and CLKOUT.
- Support for dynamic memory self-refresh mode.
- Supports 2K, 4K, and 8K row address synchronous memory parts.
  - Typically 512, 256, and 128 MB parts with 4, 8, 16, or 32 bits per device.
- Separate reset domains allow for auto-refresh through chip reset.

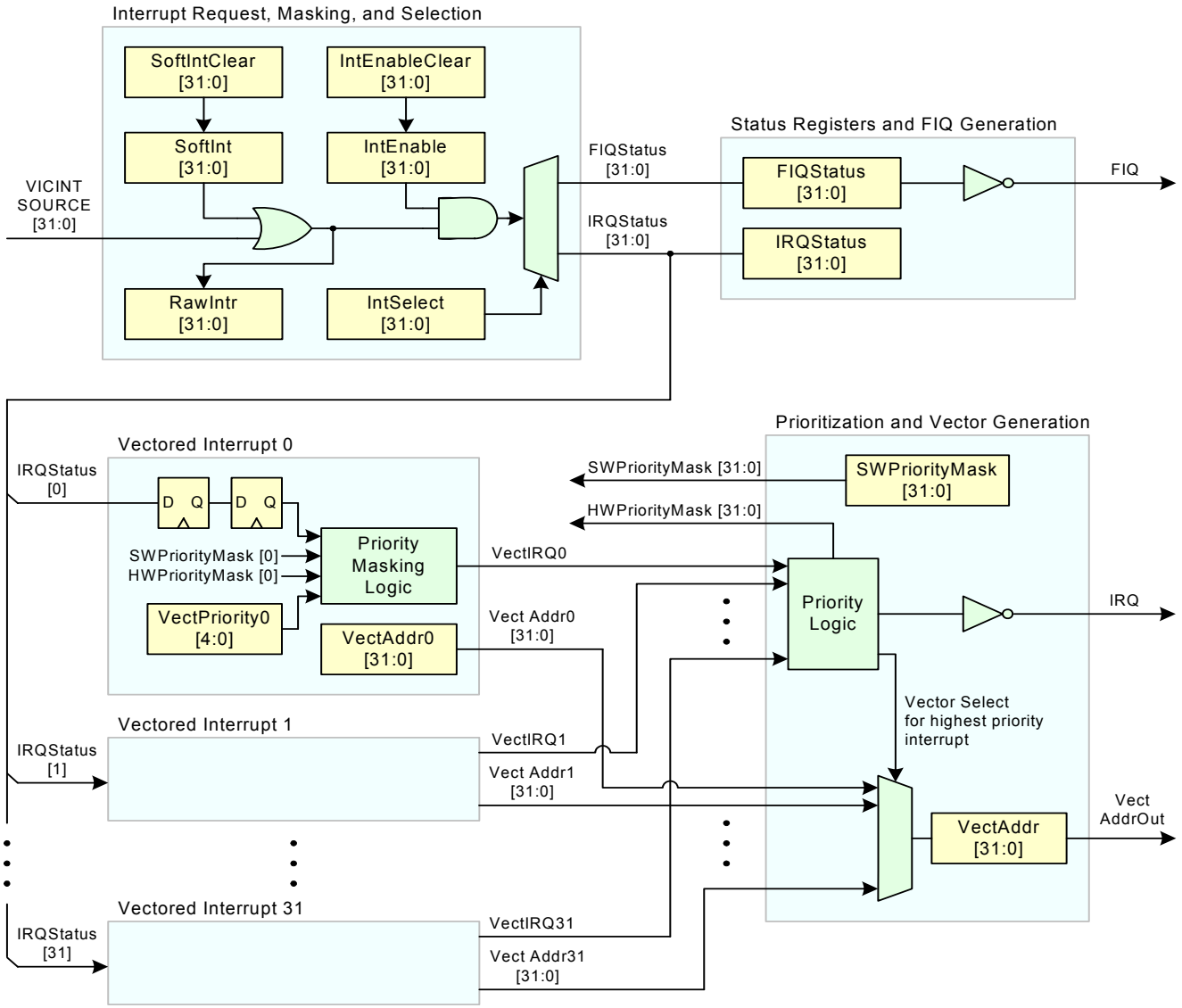
External Memory Controller Block Diagram



Advanced Vectored Interrupt Controller,

- Standard ARM PrimeCell™
- Mapped to AHB address space for fast access.
- Supports 32 vectored IRQ interrupts.
- 16 programmable interrupt priority levels.
- Fixed hardware priority within each programmable priority level.
- Any input can be assigned as an FIQ interrupt.
- Software interrupt generation.

Vectored Interrupt Controller



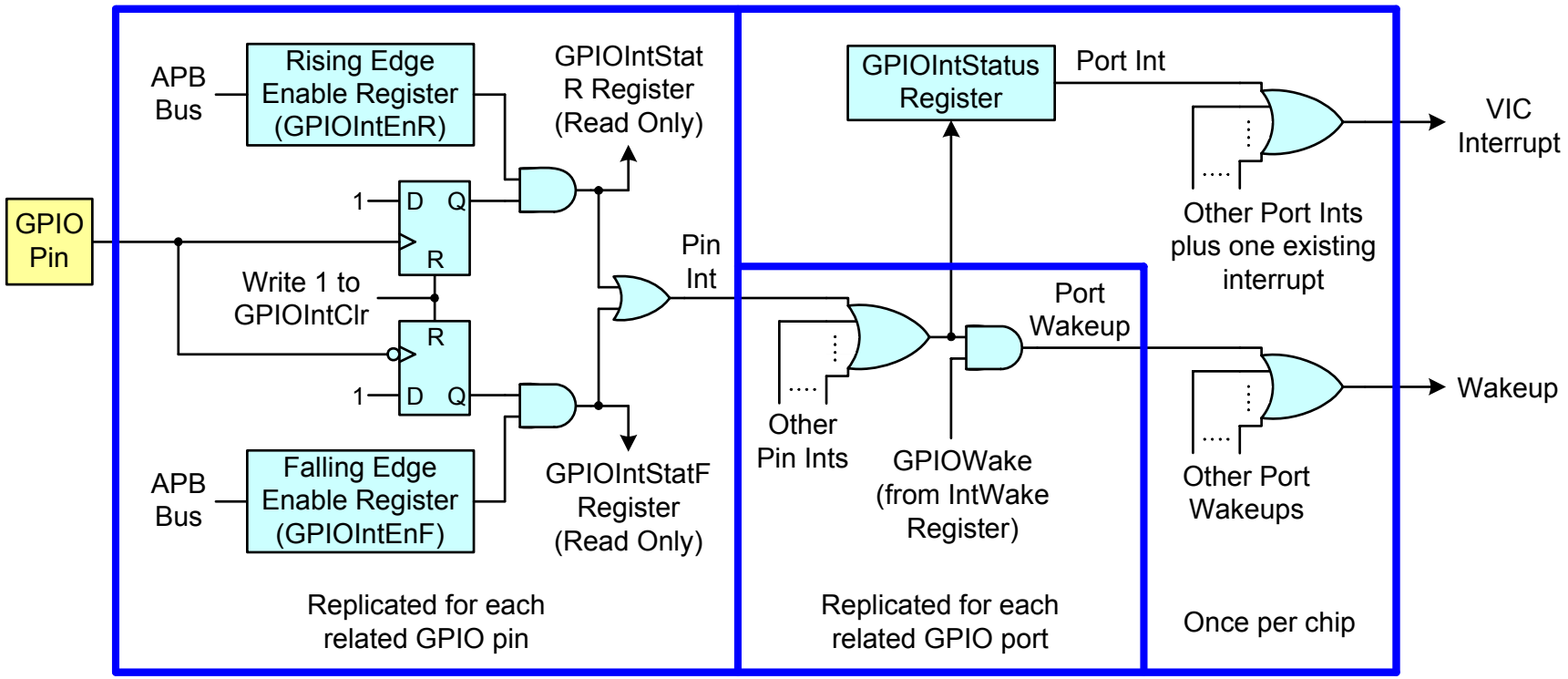
DMA (1)

- Two DMA channels, each with a four-word FIFO.
- 16 peripheral DMA request lines. Some of these are connected to peripheral functions that support DMA: the SD/MMC, two SSP, and I2S interfaces.
- One 32-bit AHB bus master interface.
- Single DMA and burst DMA request signals. Each peripheral connected to the GPDMA can assert either a burst DMA request or a single DMA request.
- Memory-to-memory, memory-to-peripheral, peripheral-to-memory, and peripheral-to-peripheral transfers are supported.

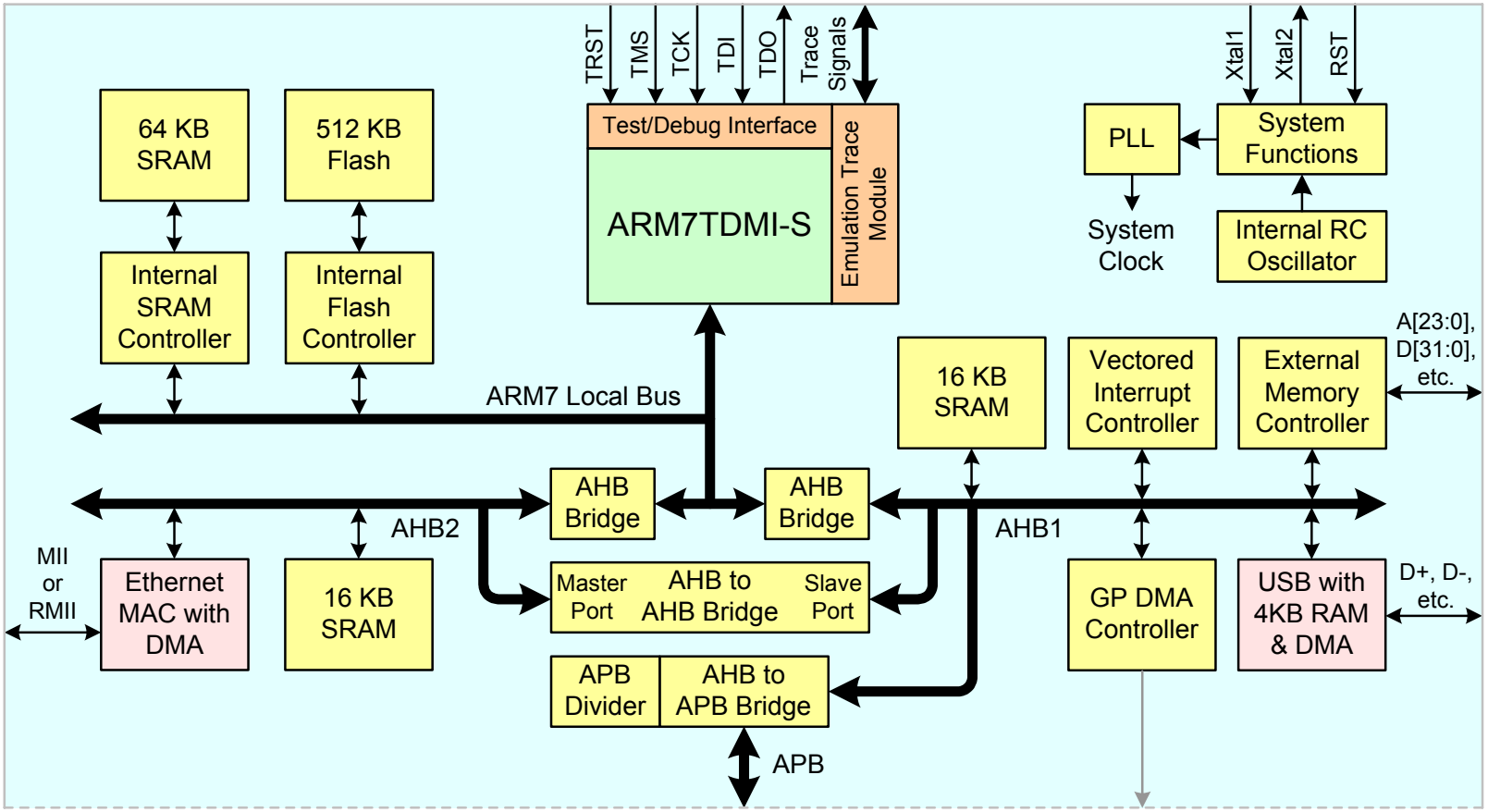
DMA (2)

- Scatter or gather DMA is supported through the use of linked lists.
- Hardware DMA channel priority: channel 0 is higher priority than channel 1.
- Incrementing or non-incrementing addressing for source and destination.
- Programmable DMA burst size.
- 8, 16, and 32-bit wide transactions.
- Big-endian and little-endian support. Little-endian is the default.
- An interrupt can be generated on a DMA completion or error.

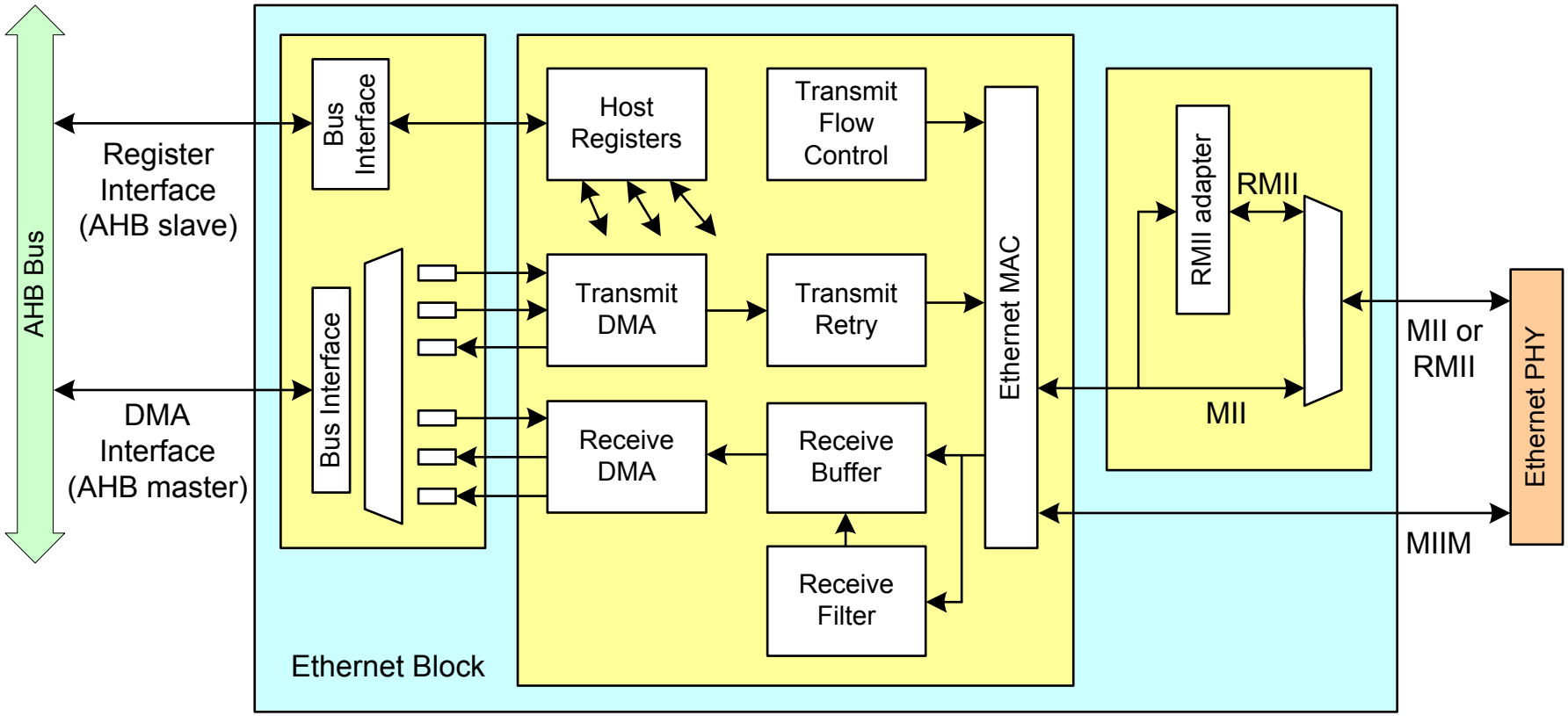
GPIO Interrupts



Ethernet MAC – System Connections



Ethernet Block Diagram



Ethernet Basic Features

- General:
  - 10/100MBps Ethernet MAC
  - PHY interface, including MII and RMII (on LPC2300 RMII only)
    - List of recommended devices in datasheet
  - Internal scatter/gather DMA controller
  - Semi-dedicated 16K byte on-chip RAM
  - Can access other memory areas (including off-chip) except the Flash and main SRAM

Ethernet Features: Standards

- Ethernet standards supported:
  - Supports 10 or 100 Mbps PHY devices including 10Base-T, 100Base-TX, 100 Base-FX, and 100Base-T4.
  - Fully compliant with IEEE standard 802.3.
  - Fully compliant with 802.3x Full Duplex Flow Control and Half Duplex back pressure.
  - Flexible transmit and receive frame options.
  - VLAN frame support.

Ethernet Features: Memory Transfers

- Memory management:
  - Independent transmit and receive buffers memory mapped to shared SRAM.
  - DMA managers with scatter/gather DMA and arrays of frame descriptors.
  - Memory traffic optimized by buffering and pre-fetching.

Ethernet Advanced Features

- Receive filtering.
- Multicast and broadcast frame support for both transmit and receive.
- Optional automatic FCS insertion (CRC) for transmit.
- Selectable automatic transmit frame padding.
- Over-length frame support for both transmit and receive allows any length frames.
- Promiscuous receive mode.
- Automatic collision backoff and frame retransmission.
- Includes power management by clock switching.
- Wake-on-LAN power management support allows system wake-up: using the receive filters or a magic frame detection filter.

Ethernet Features: Physical Layer

- Physical interface:
  - Attachment of external PHY chip through standard Media Independent Interface (MII) or standard Reduced MII (RMII) interface, software selectable.
  - PHY register access is available via the Media Independent Interface Management (MIIM) interface.

- NOTE: LPC23xx has RMII only

PHY Interface – MIIM

(Media Independent Interface Management)

MIIM Pin	# of Pins	Function
MDC	1	Clock
MDIO	1	Combined data input, data output, and data output enable. The protocol defines how and when the direction changes.
Total:	2	

PHY Interface – MII

(Media Independent Interface)

MII Pin	# of Pins	Function
TX_EN	1	Transmit data enable
TXD[3:0]	4	Transmit data output
TX_ER	1	Transmit error
TX_CLK	1	Transmitter clock
COL	1	Collision
CRS	1	Carrier sense
RX_DV	1	Receive data valid
RXD[3:0]	4	Receive data input
RX_ER	1	Receive error
RX_CLK	1	Receiver clock
Total:	16	

PHY Interface – RMII

(Reduced Media Independent Interface)

RMII Pin	# of Pins	Function
TX_EN	1	Transmit data enable
TXD[1:0]	2	Transmit data output
RXD[1:0]	2	Receive data input
CRS_DV	1	Carrier sense / Data valid
RX_ER	1	Receive error (optional, depending on application)
REF_CLK	1	Reference clock input
Total:	8	

CAN Interface incompatibilities within LPC2000 derivatives

- To conserve power, CAN peripherals are powered down after reset on the latest derivatives
  - Code must manually enable the power for the CAN interfaces used

- When using the filter tables, the numbering of CAN interfaces might vary
  - In some implementations CAN interfaces are numbered 1 and 2
  - In some implementations CAN interfaces are numbered 0 and 1

Other Peripherals

- USB with Device, Host, and OTG.
- UARTs (4), one with modem control, one with IrDA.
- CAN (2 channels).
- SD / MMC Card Interface.
- SPI (1) & SSP (2). SPI shares pins with SSP0.
- Timers (4), each with capture/compare.
- Watchdog Timer.
- DAC output.

Thank You!