Concerto™ College

Topic 3: Host Subsystem

Agenda

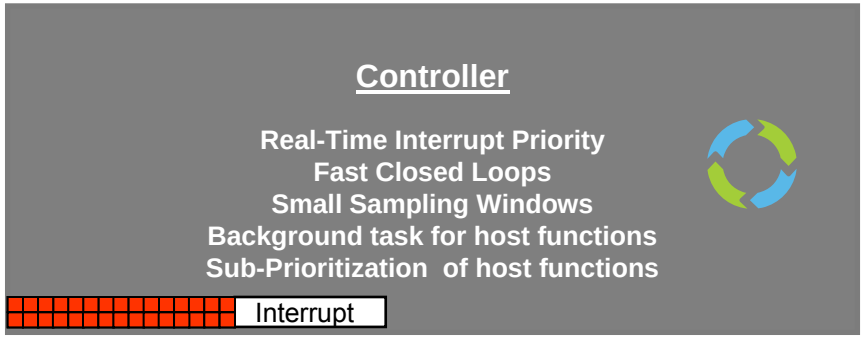
- Concerto architecture
- M3 subsystem overview
- Interrupt mechanism
- Memory
- Serial communication (UART, SSI, I²C)
- CAN interface
- Ethernet interface
- USB interface
- µDMA
- µCRC

Agenda

- **Concerto architecture**
- M3 subsystem overview
- Interrupt mechanism
- Memory
- Serial communication (UART, SSI, I²C)
- CAN interface
- Ethernet interface
- USB interface
- µDMA
- µCRC

Concerto Family: New System Architectures

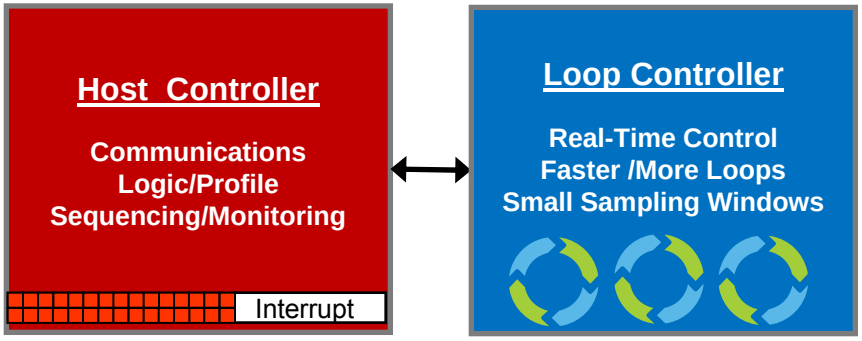
FIELDBUS
(ENET, CAN, SERIAL)



Classic MCU

- Compromise between ideal host and control capability
- Complex tasking / prioritization
- Still appropriate for deeply embedded systems

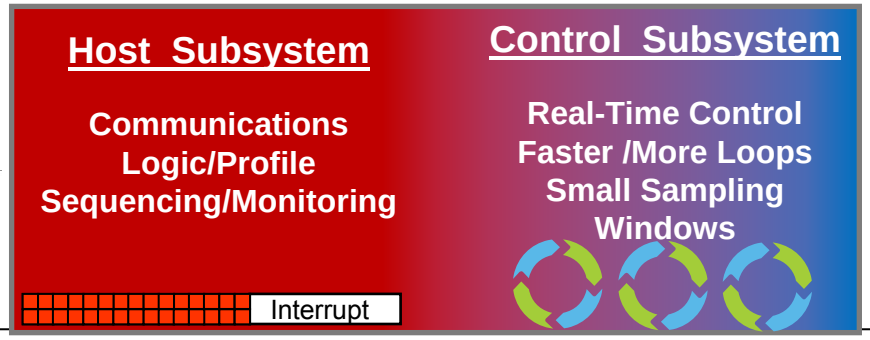
FIELDBUS
(ENET, CAN, SERIAL)



Two Chip

- Additional complexity
- Dual developments plus interface challenges / latency
- Necessary solution depending on isolation boundary trade-offs

FIELDBUS
(ENET, CAN, SERIAL)



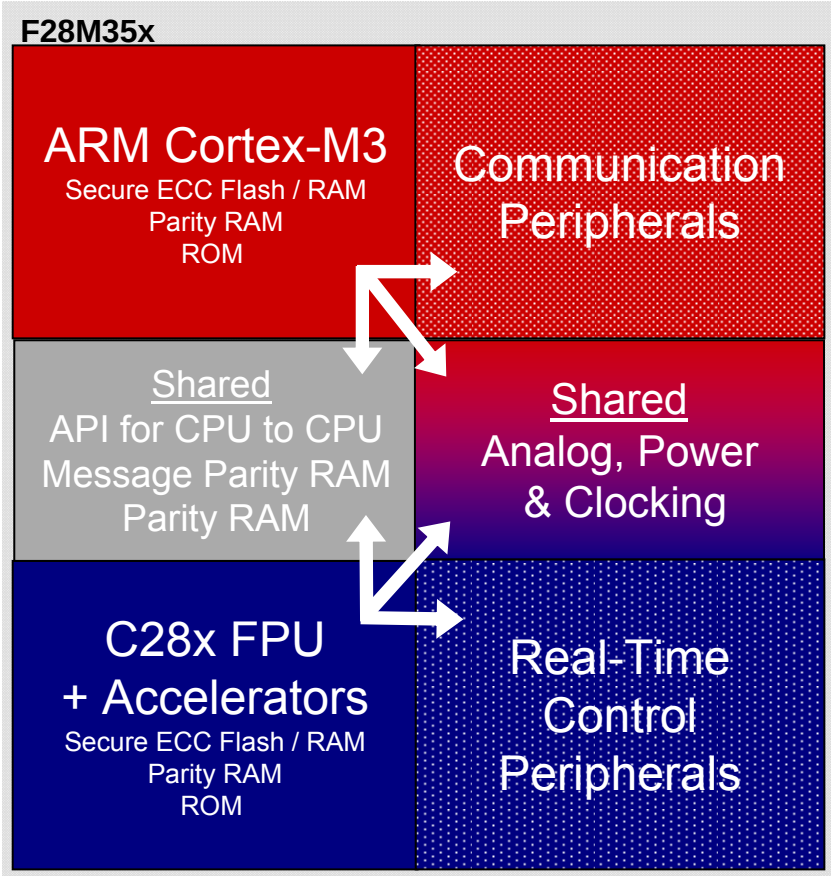
Concerto

- Independent subsystems on a single device
- Optimized capability
- Single platform for development
- No compromises

F28M35x – First Series in *Concerto*

Markets: Advanced Metering, Automotive EPS, Motion Control & Drives, UPS, Renewable Energy, Power & Protection, Medical Process Control, Smart Sensors

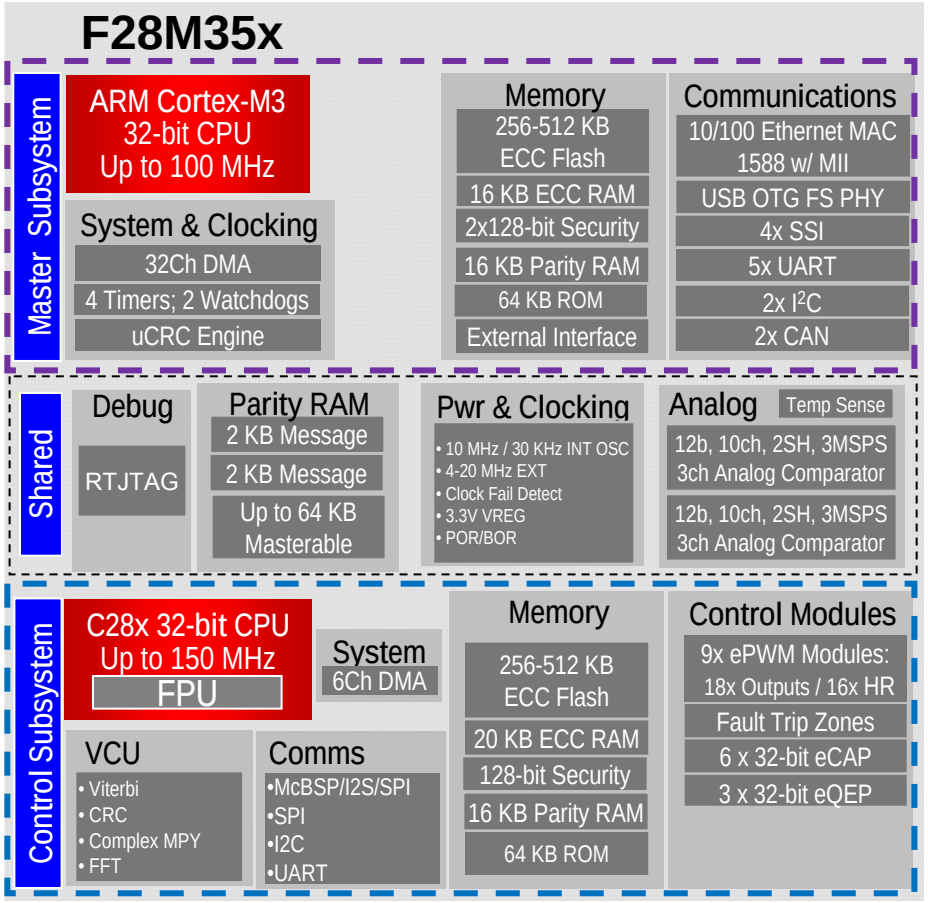
ARM Cortex-M3 for Host
TI C28xFPU for Control
75/150, 100/100, 60/60 MHz Options
1024 / 512 KB Flash
136 / 72 KB RAM
ECC and Parity Memories
HW-BIST & Additional Safety Checks
Functional Safety Documentation
Automotive Qualified Packages



F28M35x – First Series in *Concerto*

- **Multiple Performance Options**
  - 60, 100, 150 MHz C28x Floating Point
  - 60, 75, 100 MHz Cortex-M3 CPU
- **Large Internal Memory**
  - 512kB to 1MB Embedded Flash
  - 72kB to 132kB Embedded SRAM
  - ECC, Parity, and HW BIST
- **Robust Communications**
  - 10/100 Ethernet MAC with 1588
  - USB 2.0 OTG w/ integrated PHY
  - 2 x CAN
  - Multiple SPI, UART, I^2C
  - 8/16/32-bit External Memory Interface
- **Flexible Control Peripherals**
  - Enhanced PWMs w/ ~150ps resolution per channel and new flexible fault management
  - Two high-speed 12-bit ADCs each with 2S/H
  - Analog Comparators w/ Internal DAC Reference
- **144-pin PowerPad QFP**
- **Industrial and Automotive Temp**
  - -40 to 105^0C, -40 to 125^0C (AEC Q100)
- **Functional Safety Documentation**

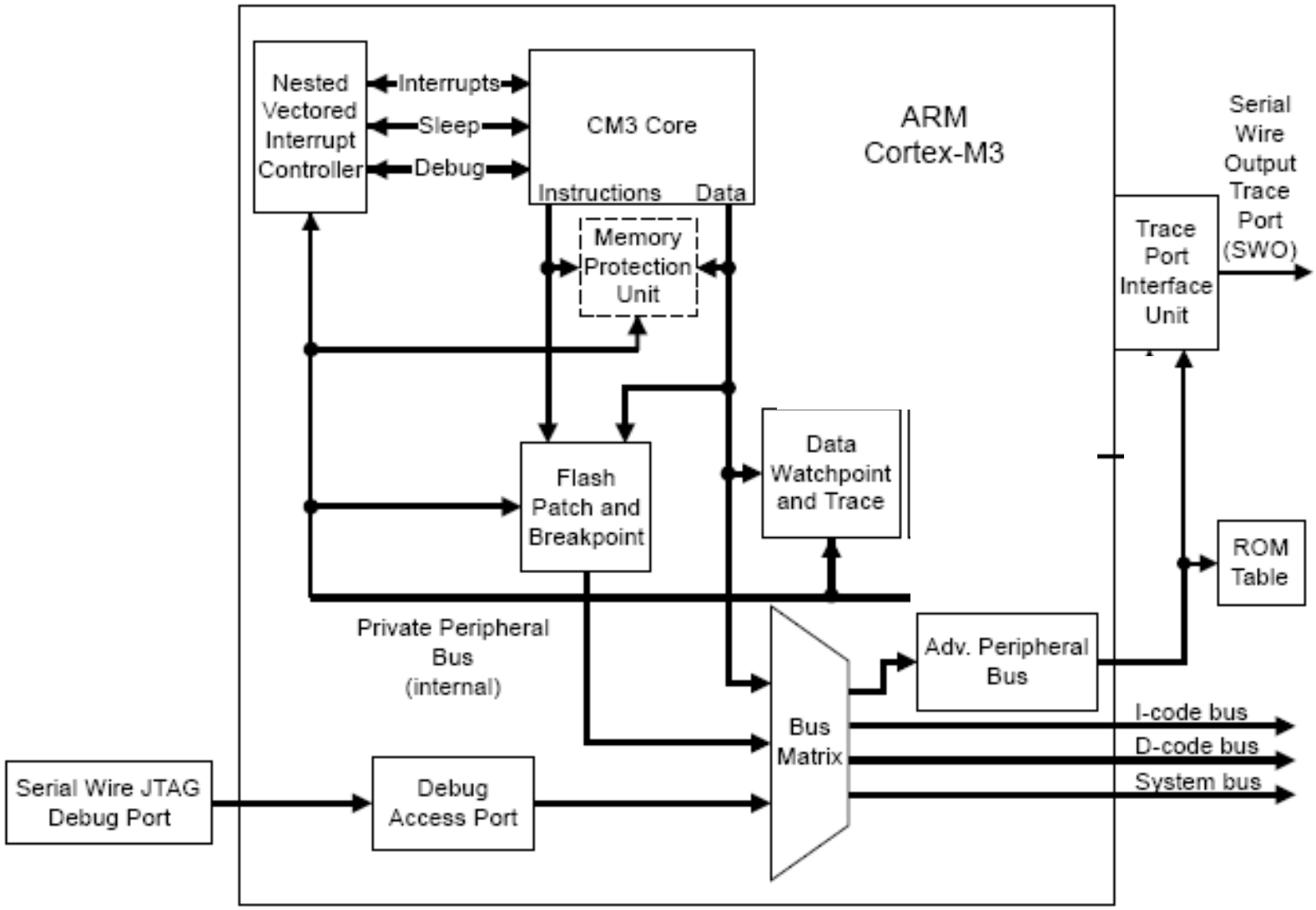
Markets: *Advanced Metering, Automotive EPS, Motion Control & Drives, UPS, Renewable Energy, Power & Protection, Medical Process Control, Smart Sensors*



144 QFP 0.5mm; 256 BGA 1mm (15x15) – Under Eval $105^0C/125^0C$ and Q100

Cortex M3 core: architecture

- Harv
- Privil
  and u
- Little



Agenda

- Concerto architecture
- M3 subsystem overview
- **Interrupt mechanism**
- Memory
- Serial communication (UART, SSI, I²C)
- CAN interface
- Ethernet interface
- USB interface
- µDMA
- µCRC

CortexM3 Interrupts

- *Nested Vectored Interrupt Controller* (NVIC):
  - prioritize and handle all exceptions
  - Automatic interrupt service
  - Pre-emptive/Nested interrupts implementation
  - full access from privileged mode

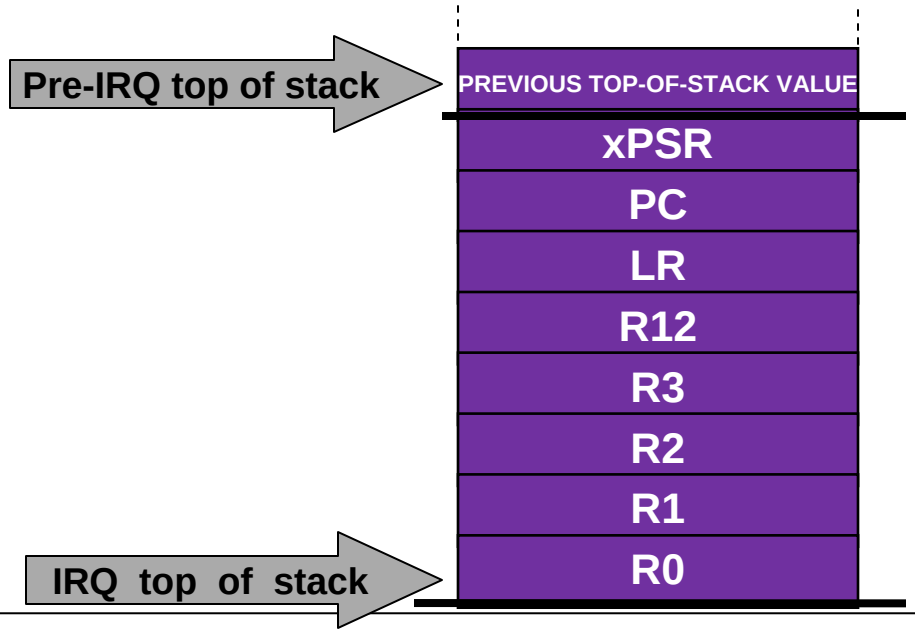
- Exceptions:
  - 10 Cortex M3 core exceptions types
  - Up to 91 peripherals interrupts (GPIOs, UART, USB,..)
  - Priority grouping

- Process and Main (Handler) stacks

Exception Model

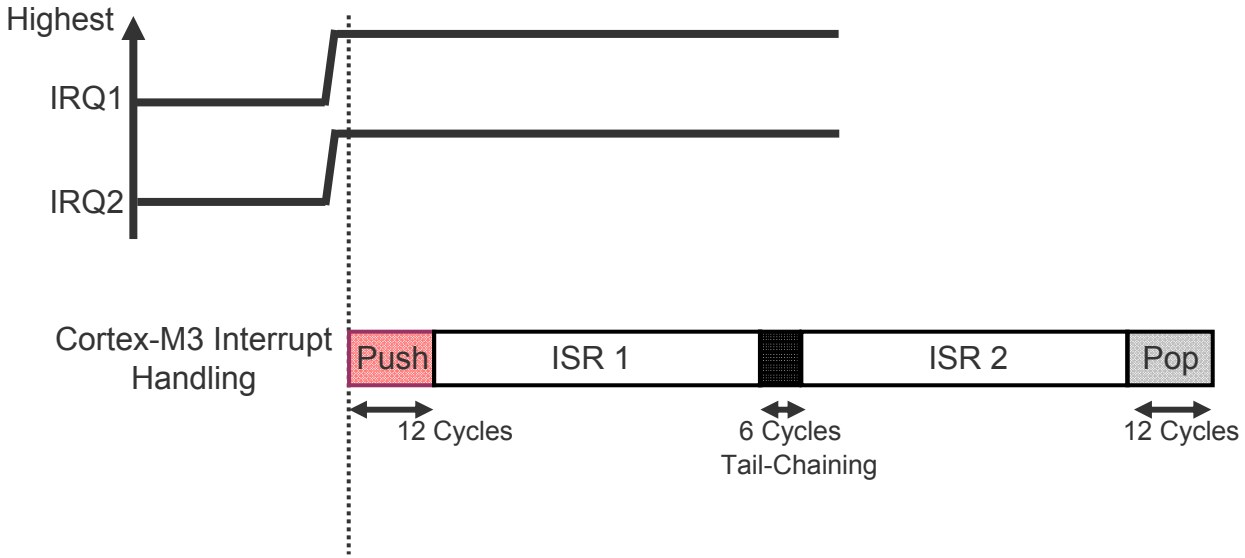
- **Exception Model handles all interrupts, synchronous faults and SVC exceptions**
  - Exceptions cause current machine state to be stacked
  - Stacked registers conform to EABI
- **Exception handlers are trivial as register manipulation carried out in hardware**
  - No assembler code required
  - Simple ‘C’ interrupt service routines:

```
void IRQ(void) { /* my handler */ }
```



Offset	Vector	IRQ number	Exception number
0x0118	IRQ91	91	107
0x0048	IRQ1	1	17
0x0044	IRQ0	0	16
0x0040	Systick	-1	15
0x003C	PendSV	-2	14
0x0038	Reserved		
0x002C	SVCall	-5	11
0x0018	Reserved		
0x0014	Usage fault	-10	6
0x0010	Bus fault	-11	5
0x000C	Memory mgt fault	-12	4
0x0008	Hard Fault	-13	3
0x0004	NMI	-14	2
0x0000	Reset		1
	Initial SP value		

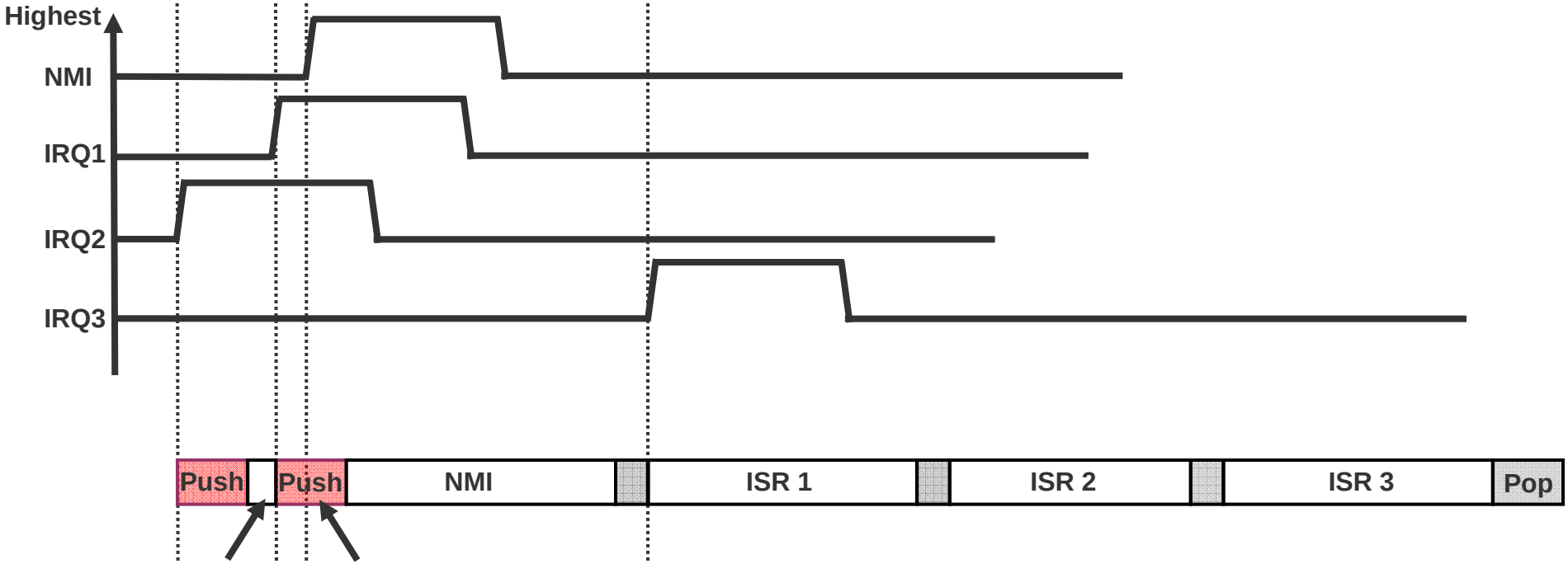
CortexM3 interrupts: Tail Chaining



Cortex-M3

- 12 cycles from IRQ1 to ISR1 (Interruptible/Continual LSM)
- 6 cycles from ISR1 exit to ISR2 entry
- 12 cycles to return from ISR2
- 65% cycle overhead saving vs ARM7

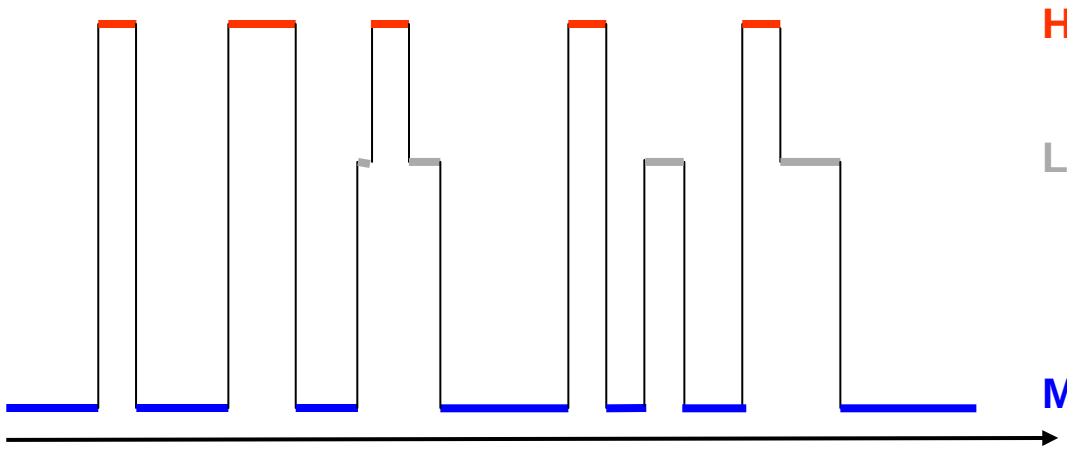
Interrupt Response – Example



- Push for ISR1 begins
- Pre-empted by NMI
- New instruction fetch in parallel minimises time to NMI

- Following NMI processor tail-chains into ISR1
- ISR2 Completed
- Pop only occurs on return to “Main”

CortexM3 Interrupts: tasks&priorities



- Main application runs as foreground (base level)
  - Easy to write since no “factoring” – just normal application or RTOS based
  - Can use PLC style state-machine poll loop safely: ISRs keep data available
- ISRs for System control are highest priority(ies)
  - Timer(s), Fault (may be highest), Temp sensor, etc
- ISRs for communications below that
  - Ethernet, CAN, and/or serial
- May use other priorities as needed
  - Very fast interrupt response time, true nested interrupts, priority masking, easy ISR setup all contribute to making an easy solution
  - Application uses priority masking vs. interrupt-disable if needs critical region

Agenda

- Concerto architecture
- M3 subsystem overview
- Interrupt mechanism
- **Memory**
- Serial communication (UART, SSI, I²C)
- CAN interface
- Ethernet interface
- USB interface
- µDMA
- µCRC

CortexM3 Memory Protection Unit

- Cortex M3 memory is split in 8 Protection regions + 1 background region

- Independent settings for each region
  - Location start address
  - Size configurable from 32 B to 4 GB
  - Independent attributes for each region

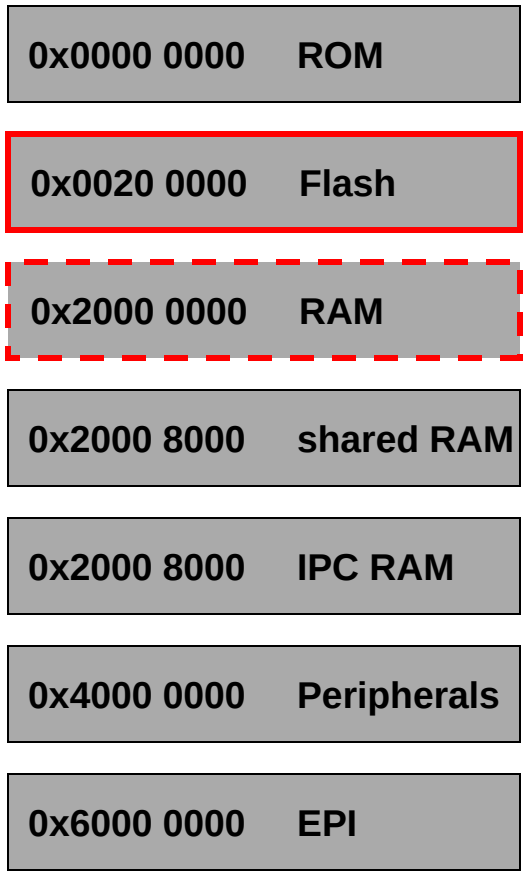
- Access Privilege (AP)
  - no access
  - R/W
  - Read only

Privilege Superuser	Unprivilege User	Permission fault
R/W	R/W	no
	R	write by unprivileged
	no	access from unprivileged
R	R	writes
	no	write + read from unpriv
no	no	all

- Overlapping protection regions with region priority

- MPU mismatches and permission violations invoke MemManage fault handler

Host side Memory map



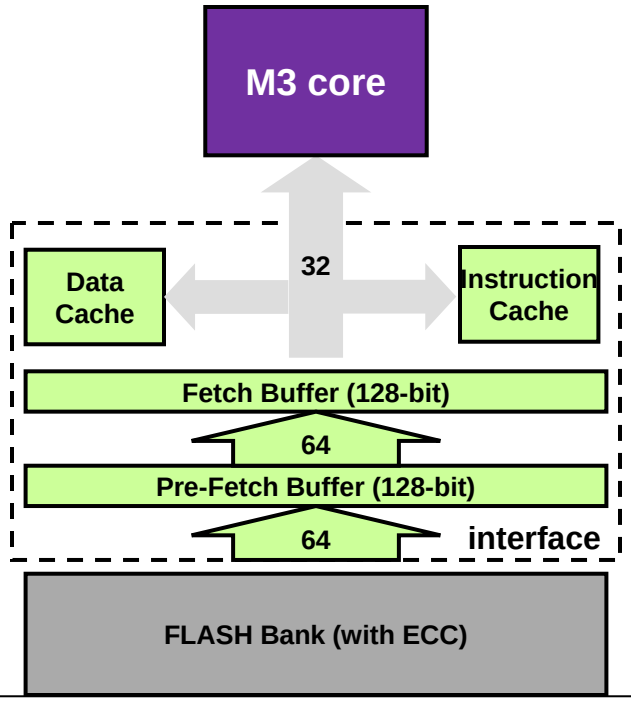
- 64 kB ROM
- 512 kB Flash
- Up to 96 kB RAM
- 4 kB RAM for IPC

Host side ROM

- Size 64 kBytes (start from 0x0000 0000)
- Access single cycle
- Contains
  - » M3 bootloader code
  - » Mathematic tables
  - » IPC code
  - » AES cryptography table
- Access from M3 only

Host side Flash

- Size 512 kBytes
  14 sectors
- Access 25 nsec
- Access from M3 only



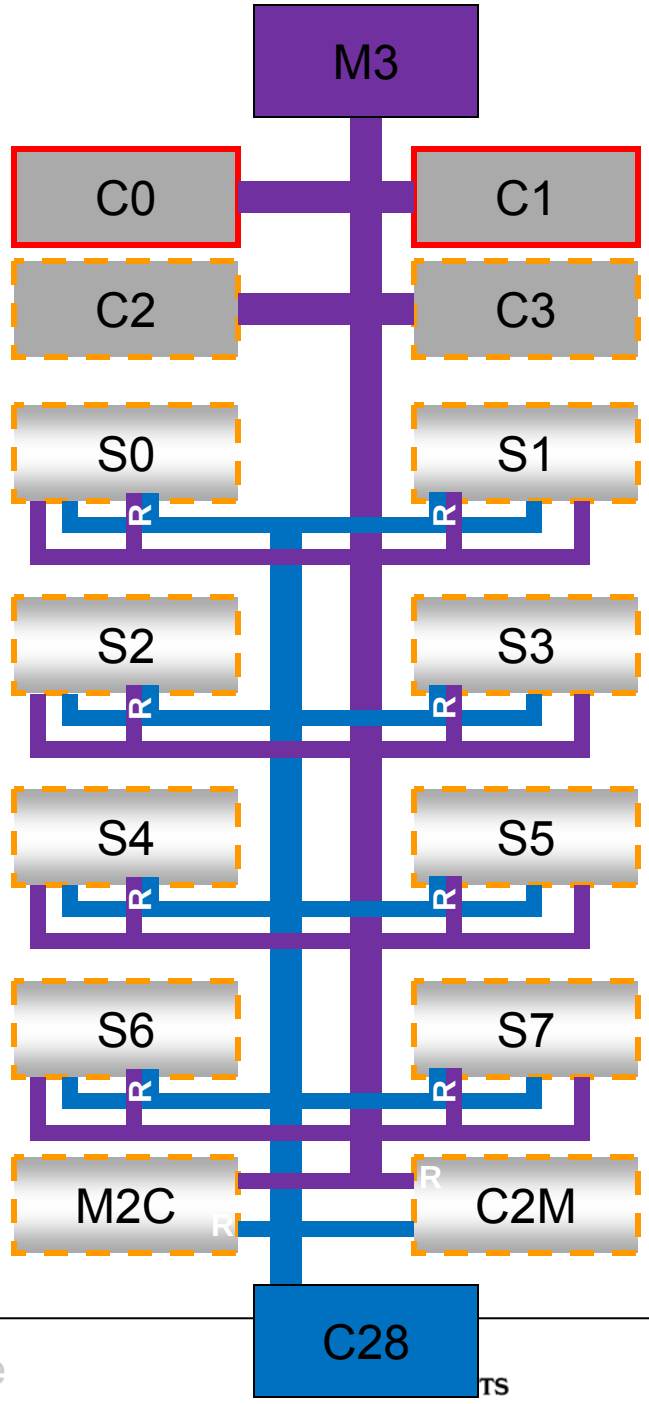
Flash performance

M3 frequency	Wait state	Efficiency	Effective speed
40 MHz	0	100 %	40 MHz
80 MHz	1	96 %	77 MHz
100 MHz	2	92 %	92 MHz

Host side RAM

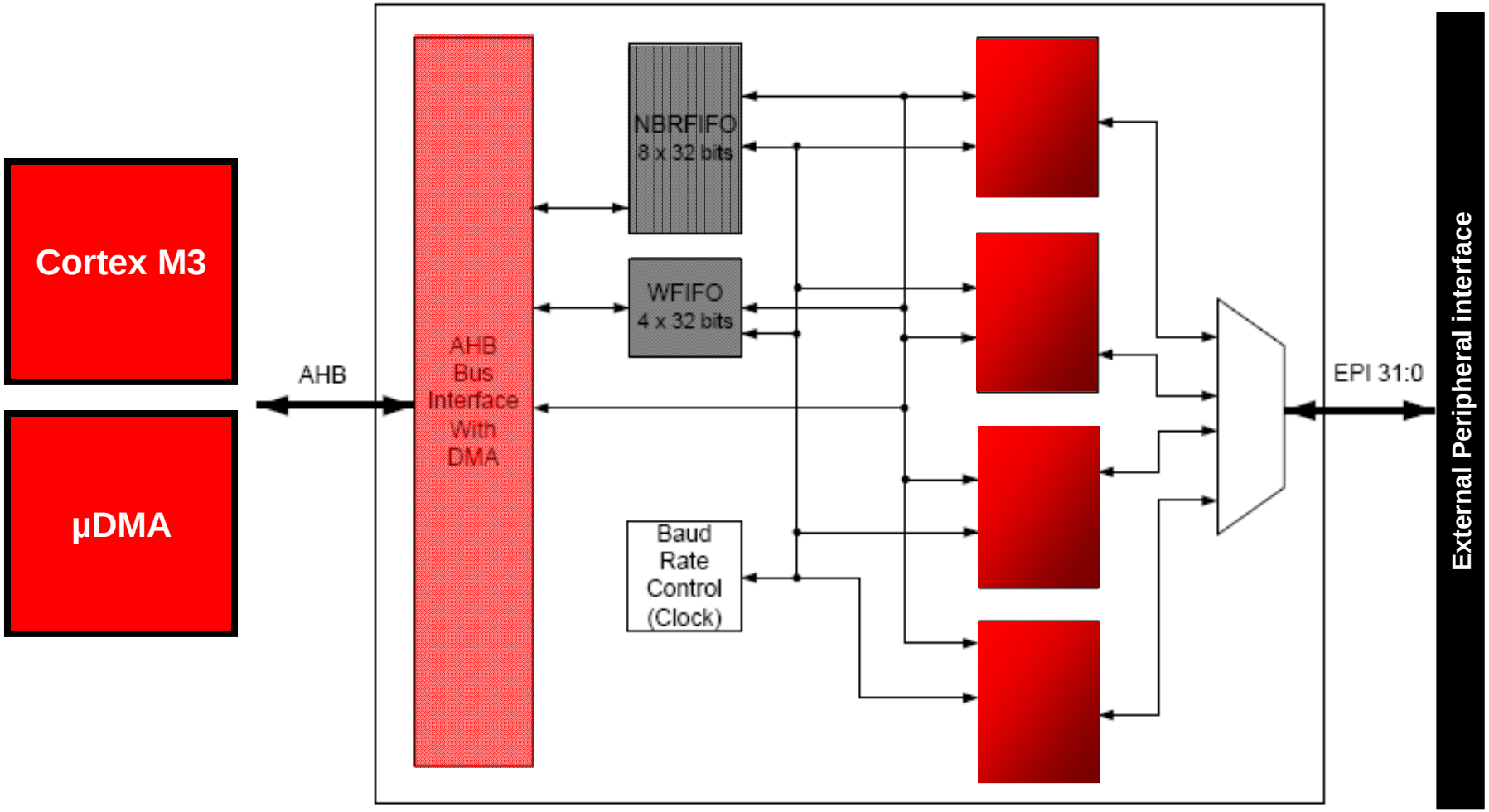
- Size 96 kB
- Access single cycle
- Each block is SARAM

	C0-C1	C2-C3	S1-S7 shared	M2C	C2M Read
Size	**8 kB**	**8 kB**	**8 kB**	**2 kB**	**2 kB**
µDMA	**No**	**Yes**	**Yes**	**Yes**	**Yes**
C28x	**No**	**No**	**Shared**	**Read**	**Yes**
Safety	**ECC**	**parity**	**parity**	**parity**	**parity**



Dedicated to M3
Shared with C28
Accessible from µDMA
ECC safety

EPI


Cortex M3
µDMA
AHB
AHB Bus Interface With DMA
NBRFIFO 8 x 32 bits
WFIFO 4 x 32 bits
Baud Rate Control (Clock)
EPI 31:0
External Peripheral interface

Agenda

- Concerto architecture
- M3 subsystem overview
- Interrupt mechanism
- Memory
- **Serial communication (UART, SSI, I²C)**
- CAN interface
- Ethernet interface
- USB interface
- µDMA
- µCRC

Serial communication peripherals - UART

- 5 independant UARTs
- Max baud rate 12.5 Mbps
- Fully programmable serial interface characteristics:
  - 5, 6, 7, or 8 data bits
  - Even, odd, stick, or no-parity bit generation/detection
  - 1 or 2 stop bit generation
- Efficient transfers using µDMA
- Flexible modes
  - IrDA serial-IR encoder/decoder for up to 115.2 Kbps half-duplex
  - ISO 7816 Support (SmartCard communication)
  - Full modem handshake (UART1 only)

Serial communication peripherals - I²C

- 2 independent I²C modules
- Master or Slave - Simultaneous operation as both a master and a slave
- 2 transmission speeds:
  - Standard (100 Kbps)
  - Fast (400 Kbps)
- Master and slave interrupts support
- Access from M3 only
- Loopback mode available

Serial communication - SSI

- 4 independent SSI
- Master or slave modes
- Max speed 25 Mbps
- Separate Tx and Rx FIFOs
  - 16 bits wide
  - 8 locations deep
- Frame size from 4 to 16 bits
- Efficient transfers using µDMA
- loopback mode available

Agenda

- Concerto architecture
- M3 subsystem overview
- Interrupt mechanism
- Memory
- Serial communication (UART, SSI, I²C)
- **CAN interface**
- Ethernet interface
- USB interface
- µDMA
- µCRC

CAN interface

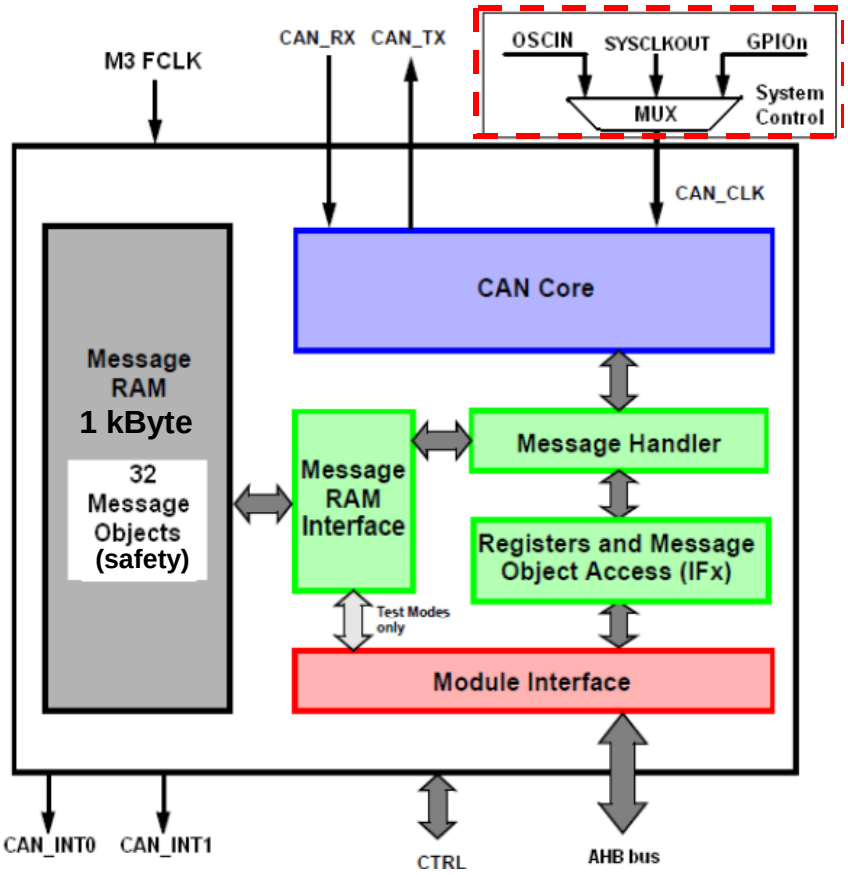
- 2 independent CAN modules compliant with CAN protocol version 2.0 part A/B
- Bit rates up to 1 MBit/s

- 1kB message RAM
  - 32 message objects
  - with Individual identifier & mask
  - Programmable FIFO mode

- Test features:
  - Programmable loop-back modes
  - Direct access to Message RAM

- Automatic bus on with programmable delay
- Wake up from deep sleep mode

- No access from µDMA



Agenda

- Concerto architecture
- M3 subsystem overview
- Interrupt mechanism
- Memory
- Serial communication (UART, SSI, I²C)
- CAN interface
- **Ethernet interface**
- USB interface
- µDMA
- µCRC

Ethernet MAC

- MAC layer only
- 10BASE-T and 100BASE-TX/RX IEEE 802.3 Full/Half-Duplex support
- Programmable MAC address
- Hardware support for Precision Time Protocol (IEEE 1588 PTP Promiscuous mode support)
- 2KB Transmit FIFO / 2KB Receive FIFO
- Efficient transfers using µDMA

Agenda

- Concerto architecture
- M3 subsystem overview
- Interrupt mechanism
- Memory
- Serial communication (UART, SSI, I²C)
- CAN interface
- Ethernet interface
- **USB interface**
- µDMA
- µCRC

USB interface

- Integrated controller with integrated PHY
- USB 2.0 full-speed (12 Mbps) / low-speed (1.5 Mbps)
- Devices with OTG/Host/Device or Host/Device
- Transfer: Control, Interrupt, Bulk and Isochronous
- Up to 32 Endpoints
  - 1 dedicated control IN endpoint
  - 1 dedicated control OUT endpoint
  - 4 KB Dedicated Endpoint Memory
- µDMA efficient data transfer

Agenda

- Concerto architecture
- M3 subsystem overview
- Interrupt mechanism
- Memory
- Serial communication (UART, SSI, I²C)
- CAN interface
- Ethernet interface
- USB interface
- **µDMA**
- µCRC

µDMA

- 32-channels - dedicated for supported peripherals
  - 8,16 or 32 bits data sizes
  - Two levels of priority, Maskable device requests

Channel Control Structure

Offset	Description
0x0	Source end pointer
0x4	Destination end pointer
0x8	Control word
0XC	unused

Control Structure Memory Map

Offset	Channel	
0x0	0	Primary
0x10	1	
…		
0X1F0	31	
0x200	0	Alternate
0x210	1	
…		
0x2F0	31	

- Multiple transfer modes:
  - Basic
  - Ping-pong
  - Scatter-gather

- Interrupt on transfer completion with a separate interrupt per channel

Agenda

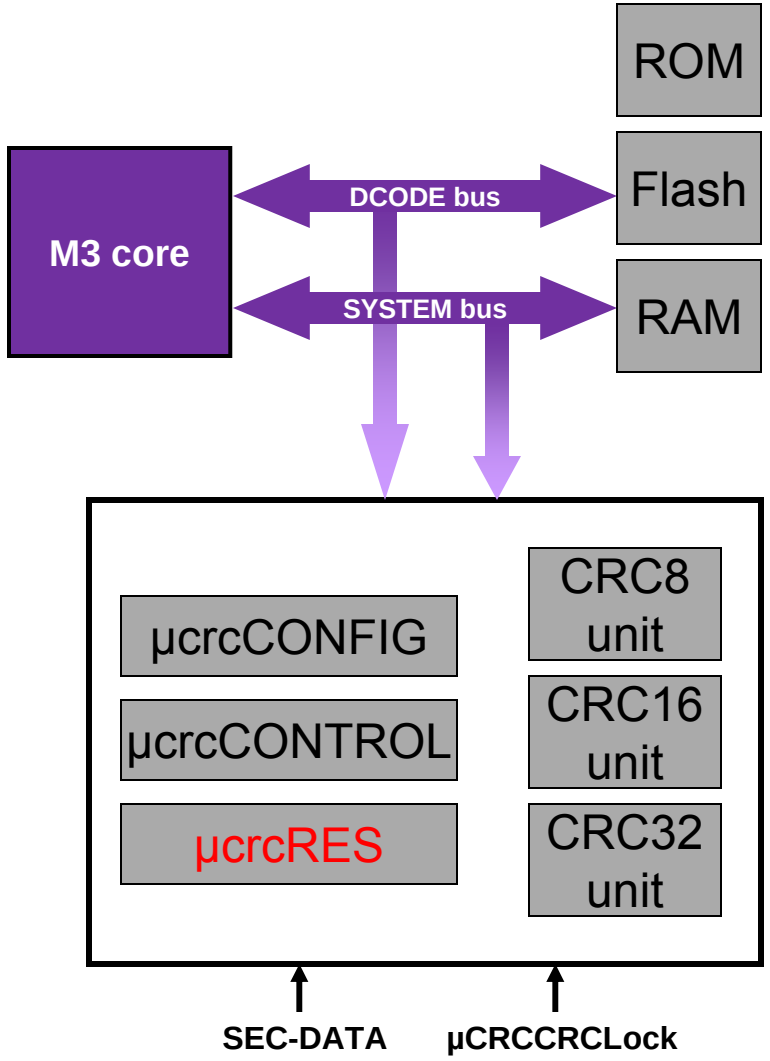
- Concerto architecture
- M3 subsystem overview
- Interrupt mechanism
- Memory
- Serial communication (UART, SSI, I²C)
- CAN interface
- Ethernet interface
- USB interface
- µDMA
- **µCRC**

uCRC

- Memory check on M3 side
  ROM, Flash and RAM (mirrored)

- Polynomials supported
  - CRC8 Poly 0x07
  - CRC16 Poly-1 0x8005
  - CRC16 Poly-2 0x1021
  - CRC32 Poly 0x04c11db7

- Ability to run on secured memory



Thanks you