UNIVERSITY OF Waterloo

TPPE 000 Technical Presentation Proficiency Examination

ARM7TDMI vs. ARM Cortex-M3 Microcontroller Cores

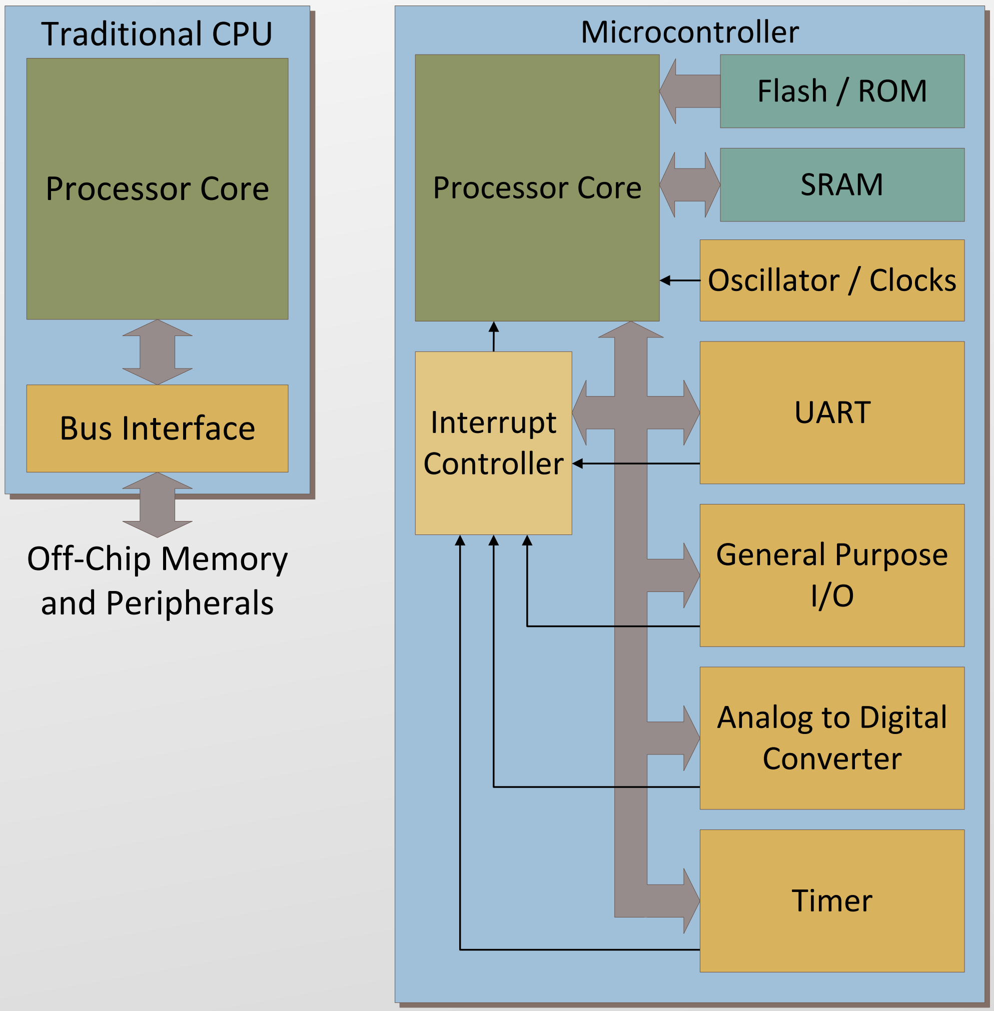
Paul Roukema
Department of Electrical and Computer Engineering
University of Waterloo
2A Candidate for B.A.Sc. in Computer Engineering
March 22, 2010

Outline

- In this presentation we will discuss
  - » Microcontrollers
  - » ARM, architectures vs. implementations
    - ARM7TDMI
    - ARM Cortex-M3
  - » Comparison
    - Initialization
    - Interrupt Handling
    - Performance
    - Availability
  - » Conclusions

Microcontrollers

- Integrates
  - » Processor
  - » Memory
    - Volatile
    - Non-Volatile
  - » Peripherals
    - Serial (UARTs)
    - ADCs
    - Timers
    - Ethernet (sometimes)
  - » Oscillator Block
    - Internal
    - External



ARM, Architectures vs. Implementations

- ARM Inc. produces
  - » Architecture Specifications
  - » Implementations of Architectures
  - » General IP (UARTS, I2C Controllers, Bus Controllers)
- Architectures ≈ Instruction Sets
  - » ARM v4T
  - » ARM v7-{M|R|A}
- Implementations
  - » ARM7 (This is an ARM v4T implementation)
  - » Cortex-M{0|1|3} (ARM v7-M)
  - » Cortex-A{5|8|9} (ARM v7-A)
  - » xScale (Intel/Marvell)

ARM7TDMI

- Introduced in 1994
- Implements ARM v4T Architecture
  - » Thumb mode for reduced code size
- Von-Neumann Architecture
- Widely used
  - » Nintendo DS (ARM7 + ARM9)
  - » iPod
  - » Many Others
- Microcontrollers available from many suppliers
  - » Atmel, Analog Devices, NXP, STMicroelectronics …
- Wide software support

ARM Cortex-M3

- Introduced in 2005
- Implements ARM v7-M architecture
  - » Thumb 2, mixed 16/32 bit instructions
- Harvard Architecture
- Branch Prediction
- Few publicized users in commercial products

Initialization

- ARM7TDMI *requires* assembly code for initialization
  - » Makes writing your own startup code more difficult
- Cortex-M3 allows the use of either C/C++ or Assembly
  - » Simpler code to write and debug
  - » Easier to understand

	ARM7	**Cortex-M3**
Begins Execution	0x00000000	*(0x00000000)
Stack Pointer	Undefined	*(0x00000004)
Language	Assembly	Assembly or C

Interrupt Handling

- Only considering core interrupt handling
  - » The usual case for Cortex-M3
  - » Usually have a secondary interrupt controller for ARM7
    - Adds latency

	ARM7	Cortex-M3
Interrupt Types	FIQ, IRQ	Vectored
# Interrupts	2	Up to 240
Priorities	2	255
State saved	FIQ only	YES
Min Latency	5 Cycles	12 Cycles
Max Latency	29 Cycles	12 Cycles

Interrupt Handling (cont'd)

- Cortex M3 appears to have slower best-case interrupt latency
  - » Have to factor in state saving, interrupt routing
  - » As an example, Atmel SAM7S series ARM7 MCUs
    - Features an Advanced Interrupt Controller peripheral
    - Provides 8 priorities, nesting, vectoring
    - Adds 3 - 4.5 cycles latency
    - Still need to save some registers
  - » Best case, SAM7S could be faster
    - Requires correct instruction timing
    - Requires reserved interrupt registers or handler in assembly
  - » Cortex-M3 provides better guarantee in general case

Performance

- Clock speed
  - » Instruction Timings
  - » Pipeline Stalls
- Power
  - » Often the limiting factor
- ARM specific tradeoff
  - » ARM vs. Thumb vs. Thumb2 instruction set
    - ARM – 32 bit wide instructions, full set of capabilities
    - Thumb – 16 bit wide instructions, less flexible
    - Thumb2 – 16/32 bit mixed instructions, best of both worlds

Performance (cont'd)

	ARM7	Cortex-M3
Max Frequency	85 MHz	100 MHz
DMIPS/MHz	0.94	1.25
Max DMIPS	80	125
DMIP/mW	3.36	3.75
Area (mm^2)	0.62	0.37

- Speed optimized 180nm Process
- Important consideration not represented
  - » ARM7 can only reach full performance running ARM code
  - » Requires more code space, faster instruction bus
  - » Thumb mode ARM7 gives max. 63 DMIPS
  - » Cortex-M3 is exclusively Thumb2

Availability

	ARM7	Cortex-M3
Vendors	Atmel, Analog Devices, NXP, STMicro, Freescale, Texas Instruments, Samsung	Atmel, NXP, STMicro, Texas Instruments

- Market availability for Cortex-M3 has improved significantly over the last year
- Some vendors targeting low cost markets, 8/16 bit pricing
- ARM7 still has an edge in specialized applications
- Software ecosystem still maturing on Cortex-M3 front
  - » Compilers OK, RTOS support could be better

Conclusions

- Cortex-M3 microcontrollers represent an easier target from a software perspective
- Improved interrupt handling
- Higher peak performance
- Better power consumption
- ARM7 benefits from wider availability and larger support ecosystem
- Largest benefit in small, low cost situations replacing dsPIC, HC12 or even PIC and AVR
- One set of development tools and rules over a wider range of applications

References

[1] Cortex-M3 Technical Reference Manual, Issue E, ARM Ltd., 2007

[2] ARM7TDMI Technical Reference Manual, Issue C, ARM Ltd., 2004

[3] AT91SAM7S Series Preliminary, Version 6175I, Atmel Corp., 2008

[4] Stellaris LM3S6965 Microcontroller Data Sheet, Rev. 2550, Texas Instruments, 2009

[5] Using the GNU Compiler Collection, For Version 4.4.2, Free Software Foundation Inc. 2009

[6] ARM Ltd., "ARM7TDMI – ARM Processor", 2009; http://www.arm.com/products/CPUs/ARM7TDMI.html

[7] ARM ltd., "ARM Cortex-M3" – ARM Processor", 2009; http://www.arm.com/products/CPUs/ARM_Cortex-M3.html

[8] N. McGuire, P. Okech, G. Schiesser, "Analysis of inherent randomness of the Linux kernel," *11th Real-Time Linux Workshop* (RTLW 09), LWN.net, Oct. 2009.