COE718: Embedded Systems Design

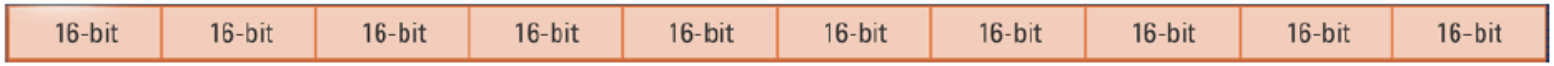
Lecture 3:
Cortex-M3 CPU Architecture

Anita Tino

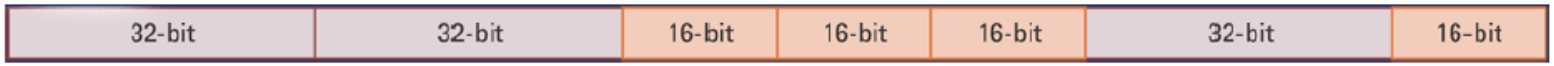
Recapping ARM and Thumb

ARM now called AArch32

Thumb (actually includes all ARM 32 bit instructions)



Thumb-2



A64 AArch64

Recapping ARM and Thumb

	ARM	Thumb*
Instruction Size	32 bits	16 bits 16bits/32bits
Core instructions	58	30
Conditional Execution	most	Only branch instructions or in an IT block
Data processing instructions	Access to barrel shifter and ALU	Separate barrel shifter and ALU instructions
Program status register	Read/write in privileged mode	No direct access
Register usage	15 general purpose registers + pc	8 general purpose registers + 7 high registers + pc

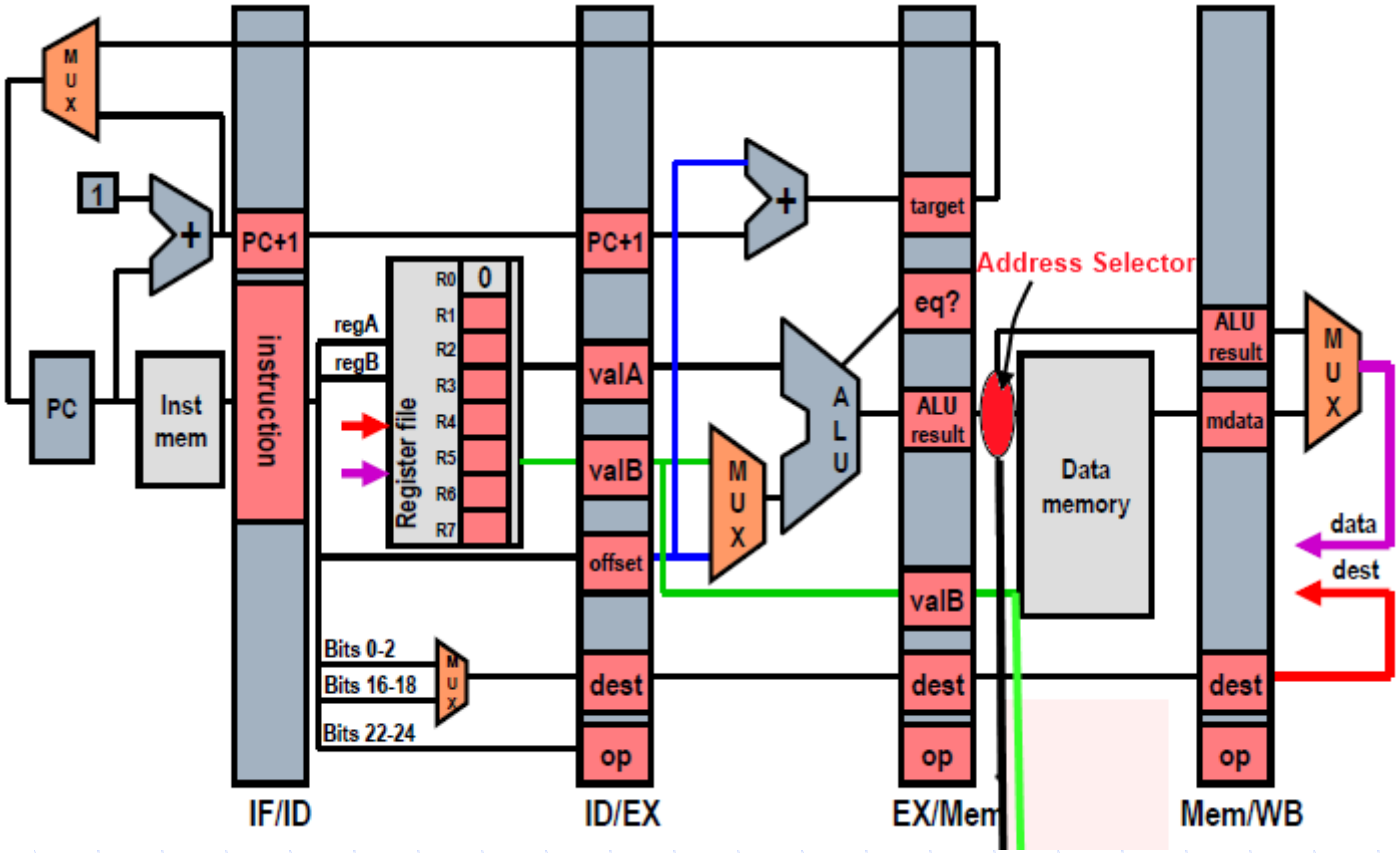
Bit Banding – allows for performance improvement and code compaction (especially depending on the application)

Harvard and Von Neumann

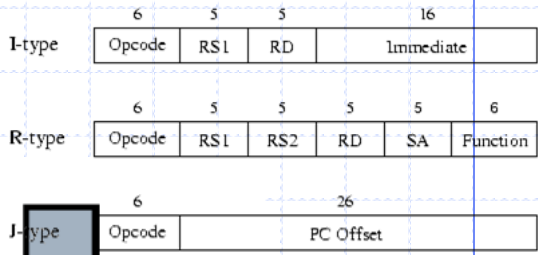
ARM Cortex-M optional components[6][7]

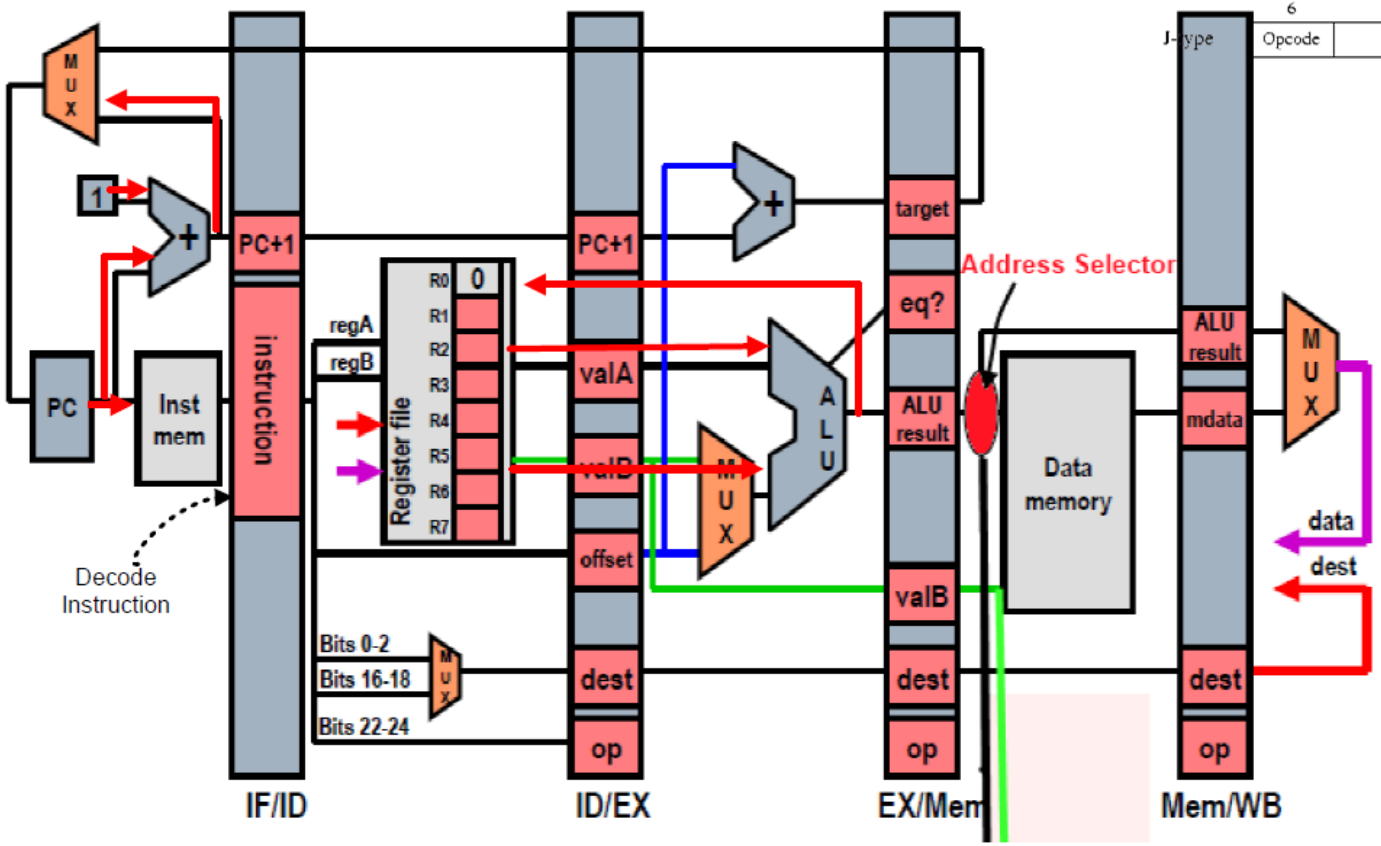
ARM Cortex-M	SysTick Timer	Bit-banding	Memory Protection Unit (MPU)	Tightly-Coupled Memory (TCM)	CPU cache	Memory architecture	ARM architecture
Cortex-M0[1]	Optional*	Optional[9]	No	No	No[10]	Von Neumann	ARMv6-M
Cortex-M0+[2]	Optional*	Optional[9]	Optional (8)	No	No	Von Neumann	ARMv6-M
Cortex-M1[3]	Optional	Optional	No	Optional	No	Von Neumann	ARMv6-M
Cortex-M3[4]	Yes	Optional*	Optional (8)	No	No	Harvard	ARMv7-M
Cortex-M4[5]	Yes	Optional*	Optional (8)	No	⊘ Possible[11]	Harvard	ARMv7E-M
Cortex-M7	Yes	No	Optional (8 or 16)	Optional	Optional	Harvard	ARMv7E-M

Example 5 Stage Pipeline CPU



Example 5 Stage Pipeline CPU

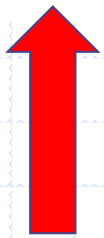


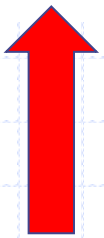


CISC vs RISC

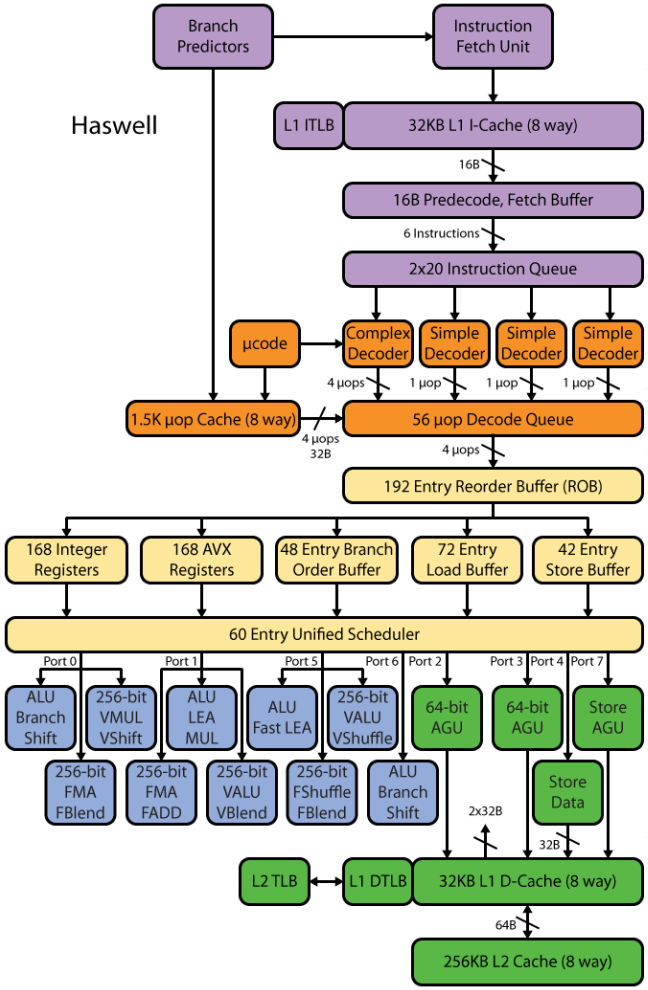
$$\frac{\text{time}}{\text{program}} = \frac{\text{time}}{\text{cycle}} \times \frac{\text{cycles}}{\text{instruction}} \times \frac{\text{instructions}}{\text{program}}$$

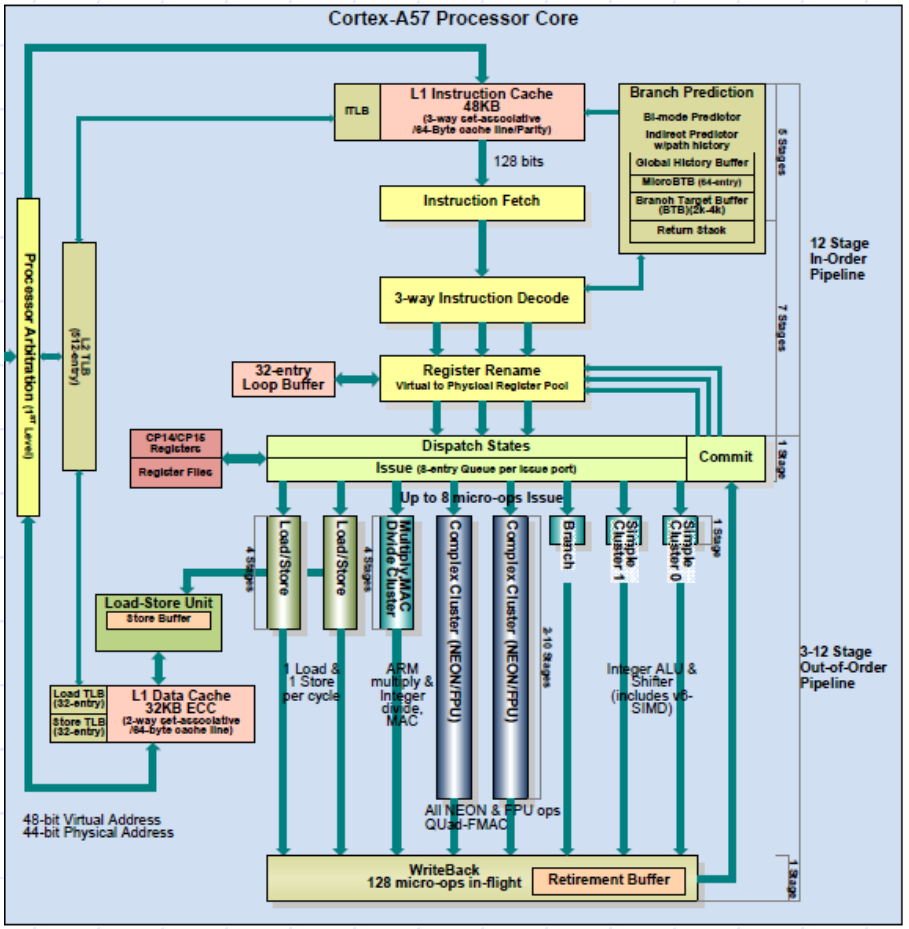
Execution Time



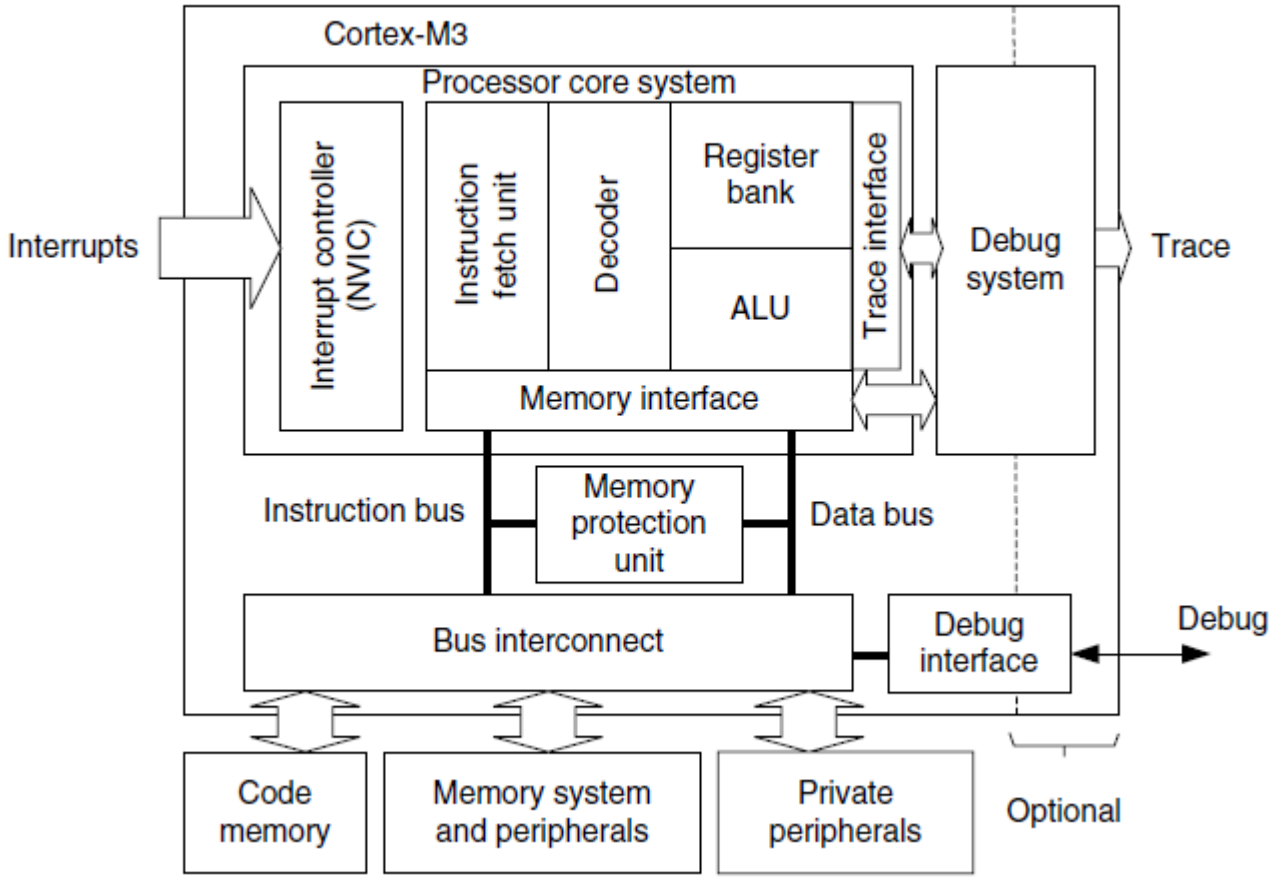


CISC vs RISC

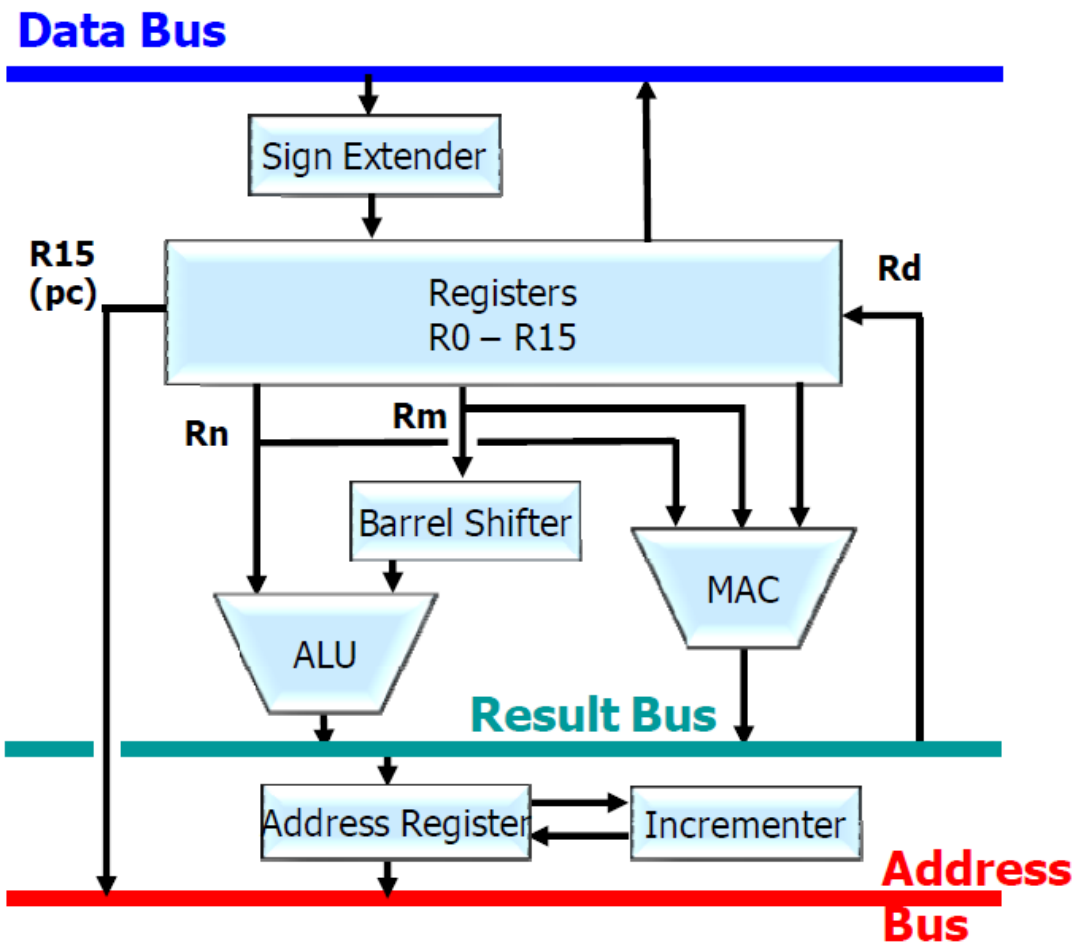

Haswell
Branch Predictors
Instruction Fetch Unit
L1 ITLB
32KB L1 I-Cache (8 way)
16B
16B Predecode, Fetch Buffer
6 Instructions
2x20 Instruction Queue
µcode
Complex Decoder
Simple Decoder
Simple Decoder
Simple Decoder
4 µops
1 µop
1 µop
1 µop
1.5K µop Cache (8 way)
4 µops
32B
56 µop Decode Queue
4 µops
192 Entry Reorder Buffer (ROB)
168 Integer Registers
168 AVX Registers
48 Entry Branch Order Buffer
72 Entry Load Buffer
42 Entry Store Buffer
60 Entry Unified Scheduler
Port 0
Port 1
Port 5
Port 6
Port 2
Port 3
Port 4
Port 7
ALU Branch Shift
256-bit VMUL VShift
ALU LEA MUL
ALU Fast LEA
256-bit VALU VShuffle
64-bit AGU
64-bit AGU
Store AGU
256-bit FMA FBlend
256-bit FMA FADD
256-bit VALU VBlend
256-bit FShuffle FBlend
ALU Branch Shift
Store Data
2x32B
32B
L2 TLB
L1 DTLB
32KB L1 D-Cache (8 way)
64B
256KB L2 Cache (8 way)


Cortex-A57 Processor Core
ITLB
L1 Instruction Cache 48KB
(3-way set-associative /64-Byte cache line/Parity)
Branch Prediction
Bi-mode Predictor
Indirect Predictor w/path history
Global History Buffer
MicroBTB (64-entry)
Branch Target Buffer (BTB)(2k-4k)
Return Stack
128 bits
Instruction Fetch
3-way Instruction Decode
Processor Arbitration (1st Level)
L2 TLB (512 entry)
32-entry Loop Buffer
Register Rename
Virtual to Physical Register Pool
CP14/CP15 Registers
Register Files
Dispatch States
Issue (8-entry Queue per issue port)
Commit
Up to 8 micro-ops Issue
Load/Store
Load/Store
Multiply/MAC Divide Cluster
Complex Cluster (NEON/FPU)
Complex Cluster (NEON/FPU)
Branch
Simple Cluster 1
Simple Cluster 0
Load-Store Unit
Store Buffer
Load TLB (32-entry)
Store TLB (32-entry)
L1 Data Cache 32KB ECC
(2-way set-associative /64-byte cache line)
1 Load & 1 Store per cycle
ARM multiply & Integer divide, MAC
Integer ALU & Shifter (includes v8-SIMD)
All NEON & FPU ops QUad-FMAC
48-bit Virtual Address
44-bit Physical Address
WriteBack
128 micro-ops in-flight
Retirement Buffer
12 Stage In-Order Pipeline
3-12 Stage Out-of-Order Pipeline

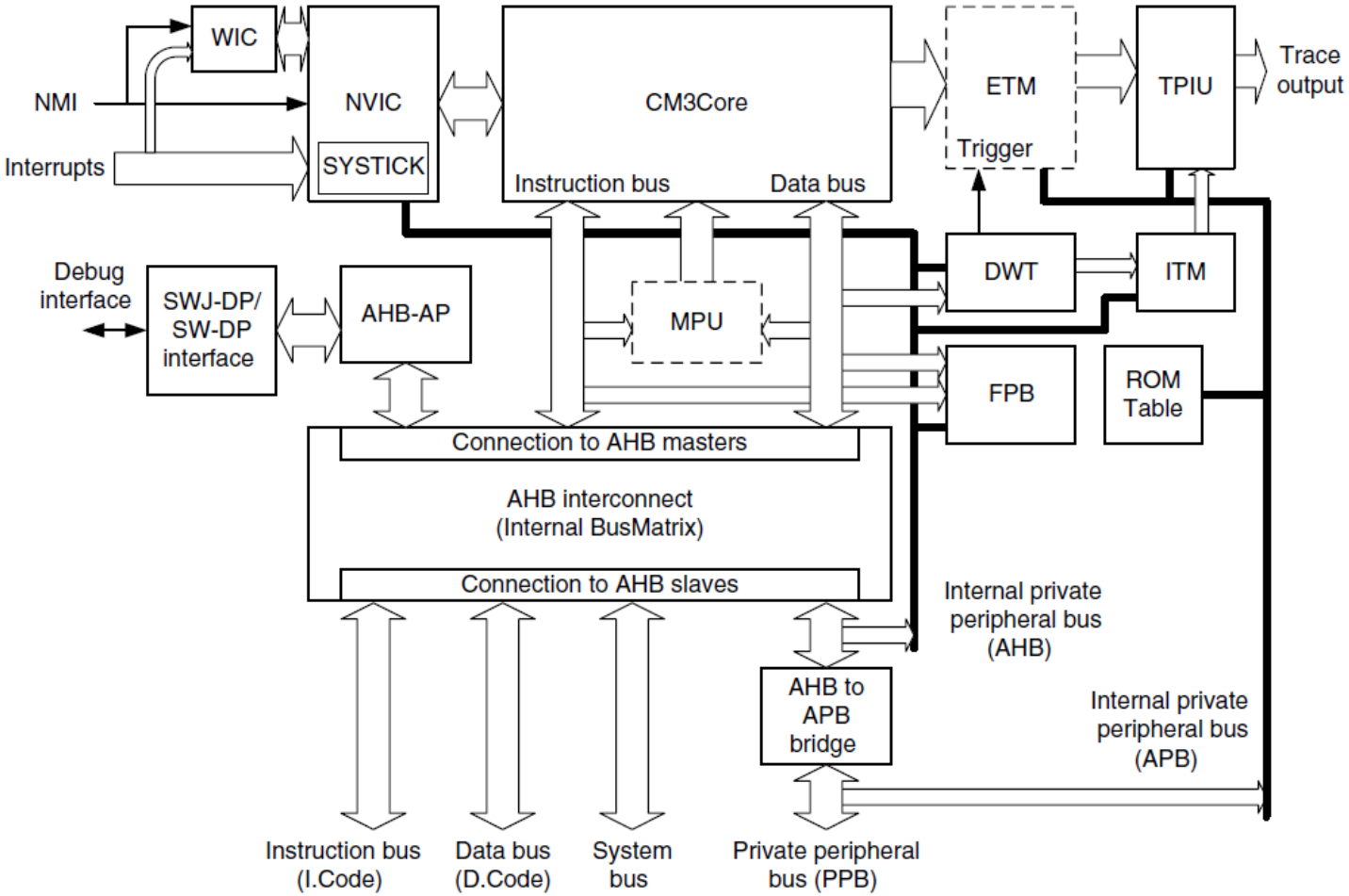
Cortex-M3 Core Overview



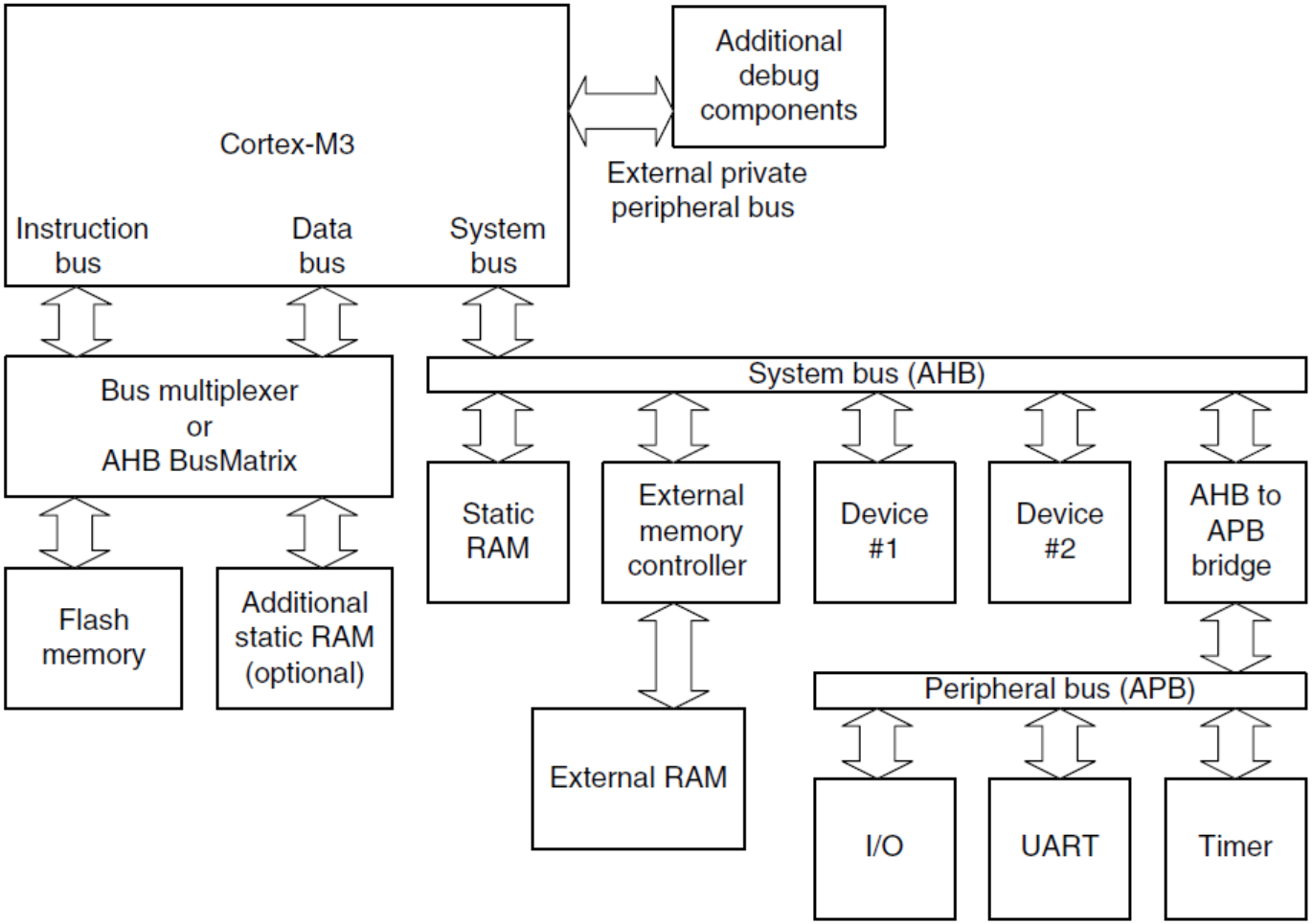
Cortex-M3 Backend



Cortex-M3 CPU Overview

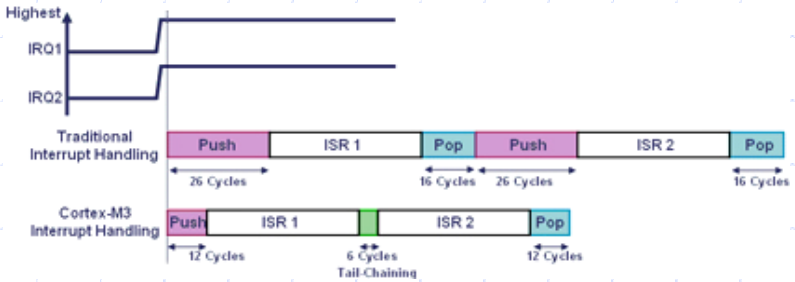


Cortex-M3 Bus System

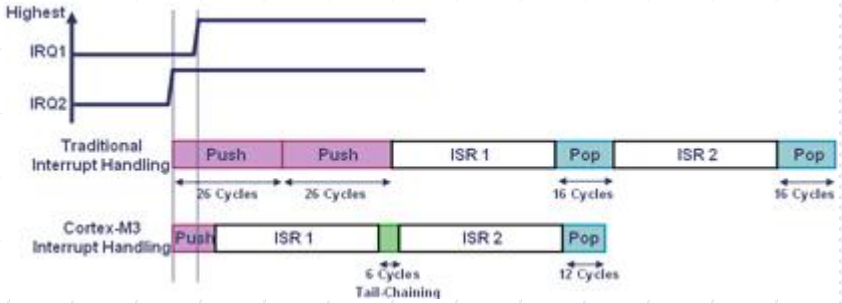


Tail Chaining

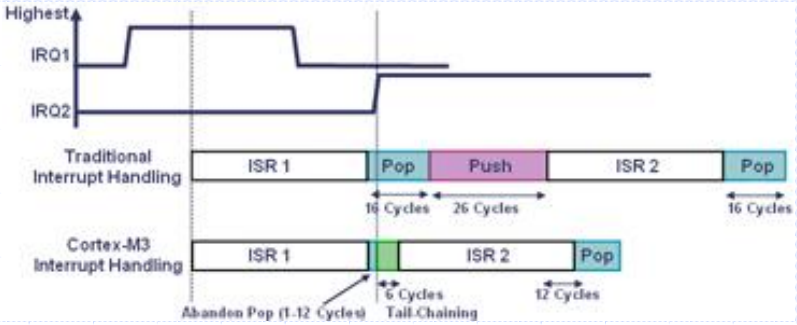
Back-to-Back



Late Arrival



Later Arrival (During Pop)



Embedded Systems

LETS PLAY.....

DESIGN THIS!

DESIGN THIS!

Intelligent Hanger
(Fashion & Engineering project)