The ARM Architecture

Agenda

- Introduction to ARM Ltd
- Programmers Model
- Instruction Set
- System Design
- Development Tools

ARM Ltd.,

- Founded in 1990
  - Spun out of Acorn Computers
- Designs the ARM range of RISC processor
- Licenses ARM core for Design Partners
  - ARM does not fabricate Silicon on own
- Develops technology to assist with ARM Arch.
  - Software Tools
  - Boards | Debug Hardware | Peripherals
  - Application Software | Bus Architecture

IP – Intellectual Property

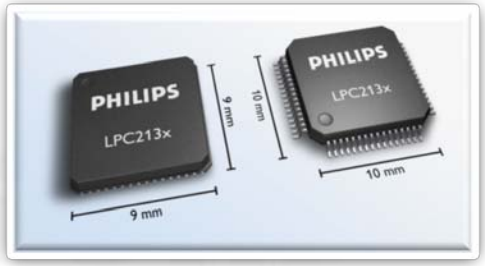
- ARM provides hard and soft views to licensees
  - RTL and Synthesis Overflow
  - GDSII Layout
- Licensees have the right to use hard or soft views of the IP
  - Soft views include gate level net-list
  - Hard view are DSMs
- OEMs must use Hard Views
  - To protect ARM IP

ARM Overview

ARM

IP Core
RTL / GDSII
Layout

Silicon
Manufacturer
NXP | Atmel |
SAMSUNG | INTEL



Development
Tools
IDE/Debugger
ICE

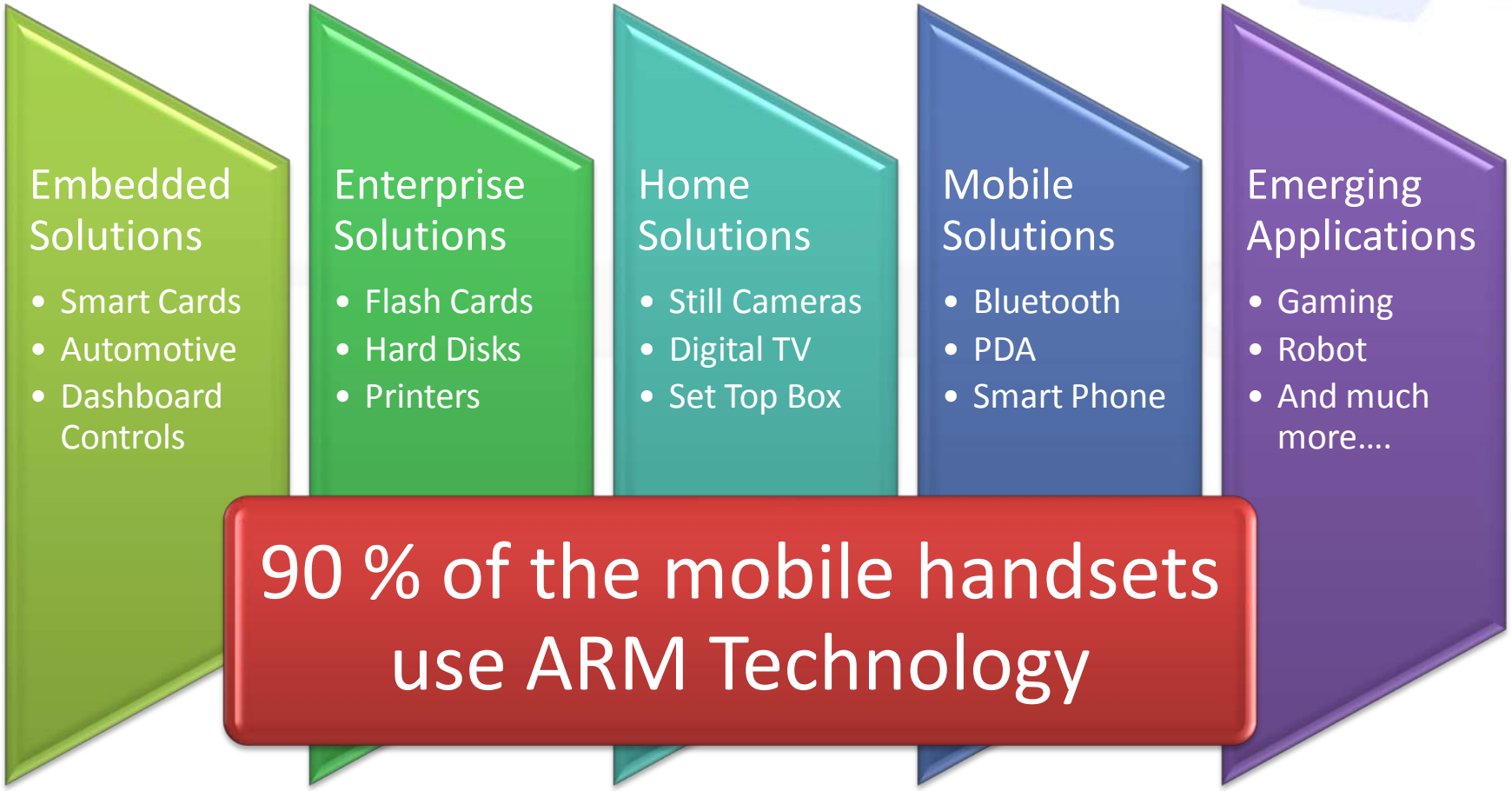
IAR
KEIL
Realview…

ARM Overview

- Shipped in excess of 10 billion units since 1990.
- Over 1300 new chip designs with ARM IP in 2007
- Licensed by majority of world's leading Semiconductor manufacturers
- At least 90 processors are shipped every second.
- *isuppli* predicts over 5 billion chips/year will ship in 2011.

Markets

The FIVE key markets are:



Processor Licensee

- ARM licenses its microprocessor IP to majority of the semiconductor companies.

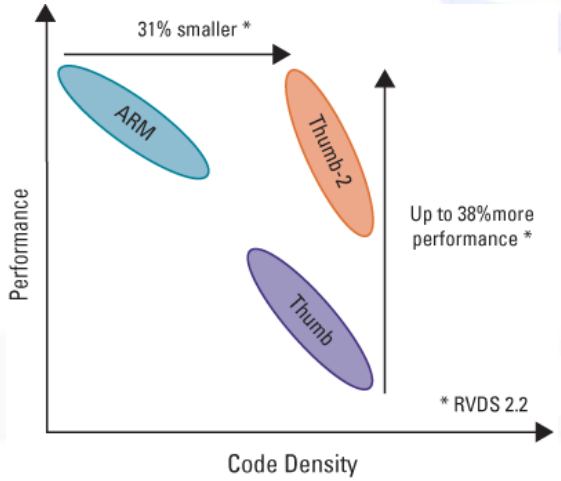
Family	Licenses
Cortex™	47
ARM11™	66
ARM9™	247
ARM7™	155

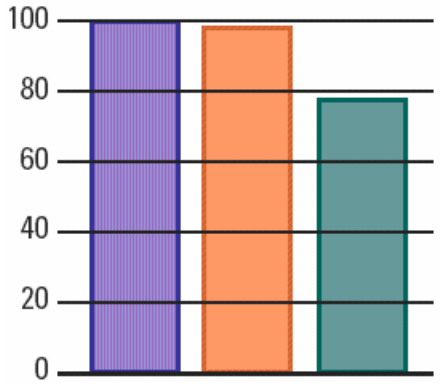
Why ARM ?

- Built in architecture extensions
  - THUMB®2 - Greatly improved code density
  - DSP - Signal process directly in the RISC core
  - Jazelle® - Java acceleration
  - Trustzone™ - Maximum Security Environment
- Core Performance
- Tools of Choice
- Wide Support
- Low Power Consumption

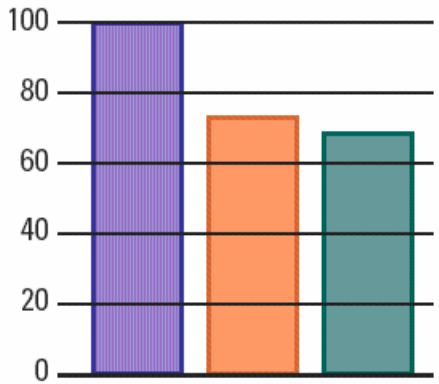
Architecture Extensions

- Improved Code Density
- Improved Performance
- Power Efficiency





Thumb-2 Performance
25% faster than Thumb

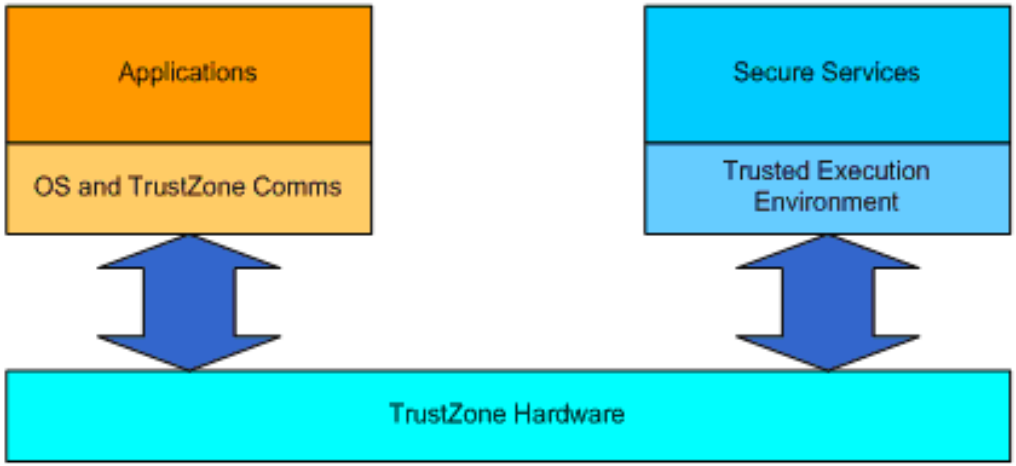


Thumb-2 code size
26% smaller than ARM

Architecture Extensions…

- The TrustZone®
  - Provide a Trusted Execution Environment (TEE)
  - Consistent programming model across platforms and applications
  - Hardware backed security environment

TrustZone®
Security Foundation by ARM®



Architecture Extensions…

- Jazelle®
  - Technology to enable Jazelle H/w in any existing JVM
  - Full featured multi-tasking Java Virtual Machine JVM
  - Enables acceleration of execution environments

jazelle®
Execution Environment
Acceleration by ARM®

Apple iPhone Key Design Win for NXP ARM

LPC2211 → PMU → 12 Logic Parts → 5$ – 6 $ content

2007
- 4 - 5 Mega Units

2008
- 10 Mega Units

PANTECH SOLUTIONS
Technology Beyound the Dreams

Significant Design-Win Americas: Q4 2006

Significance	Used LPC900 MCUs to beat Microchip on high volume consumer product	Automation Bank machine can be chosen as world platform by Diebold. Provides stronger security system in this kind of product.	
Customer	Diebold - Procomp		
MMS DW Product	LPC2144FBD64-S		
Other NXP/MMS part types			
End Product	Automation Bank Machine		
Application	Baking		
Quantity (EAU)	50k units		
Value (3yr/5yr auto)	$1.00 mil (3yr)		
NXP Competitive Advantage	NXP MCU is most competitive package product solution	Sales Engineer	Marcelo Gonçalves

ARM Partnership Model

ARM7 Family

- ARM7 Family Includes…

ARM7TDMI
- T - Thumb® instruction set
- D - Debug extension
- M-Enhanced multiplier (32x8)
- I-Embedded ICE

ARM7EJ-S
- E – DSP Extension
- J – Jazelle®, Java Extension
- S – Synthesizable

ARM7TDMI-S (ARM v4T)
- T - Thumb® instruction set
- D - Debug extension
- M-Enhanced multiplier (32x8)
- I-Embedded ICE
- S – Synthesizable Macro

ARM720T
- T – Thumb
- Uses ARM7TDMI-S CPU
- MMU Included

End Products



















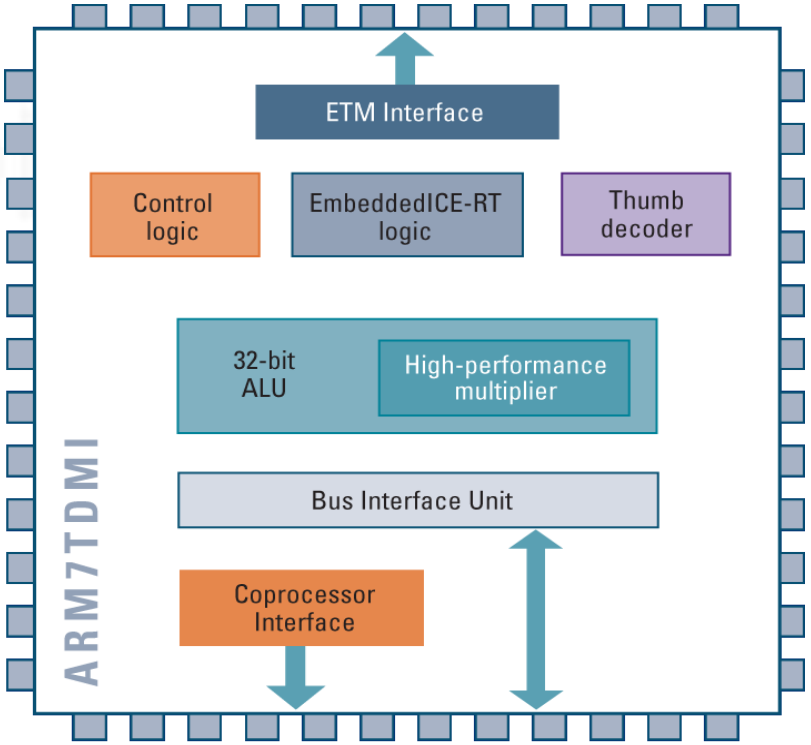
PANTECH SOLUTIONS
Technology Beyound the Dreams

Agenda

- Introduction to ARM Ltd
- Programmers Model
- Instruction Set
- System Design
- Development Tools

Bus Width

- ARM7 is a 32-bit architecture
- Data Paths and Instructions (ARM) are 32 bits wide
- Von-Nuemann Architecture
  - Instructions and Data use same bus
- Thumb Mode
  - Subset of 16-bit instructions
  - Code density optimization

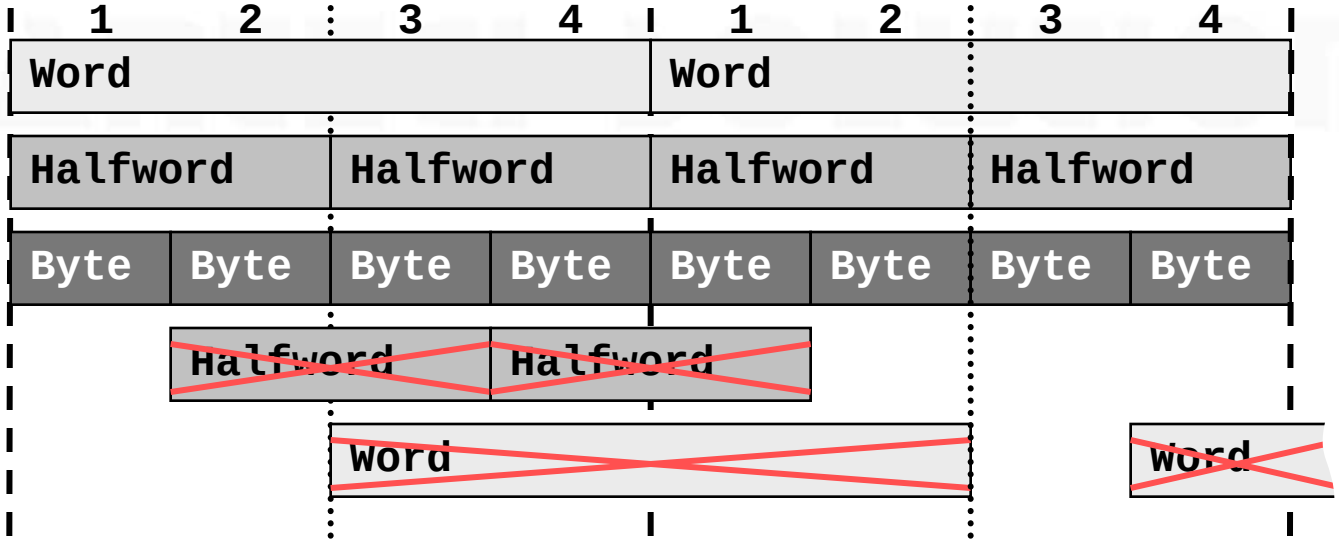


Thumb State

- 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 endian are options):
  - Word = 32 bits (four bytes)
  - Halfword = 16 bits (two bytes)
  - Byte = 8 bits



Data and Instruction

- DATA when used in ARM
  - Byte : 8 bits
  - Half Word : 16 bits (2 Bytes)
  - Word : 32 bits (4 Bytes)
- Instruction Set of most ARMs
  - 32 bits ARM Instruction Set
  - 16 bits Thumb Instruction Set

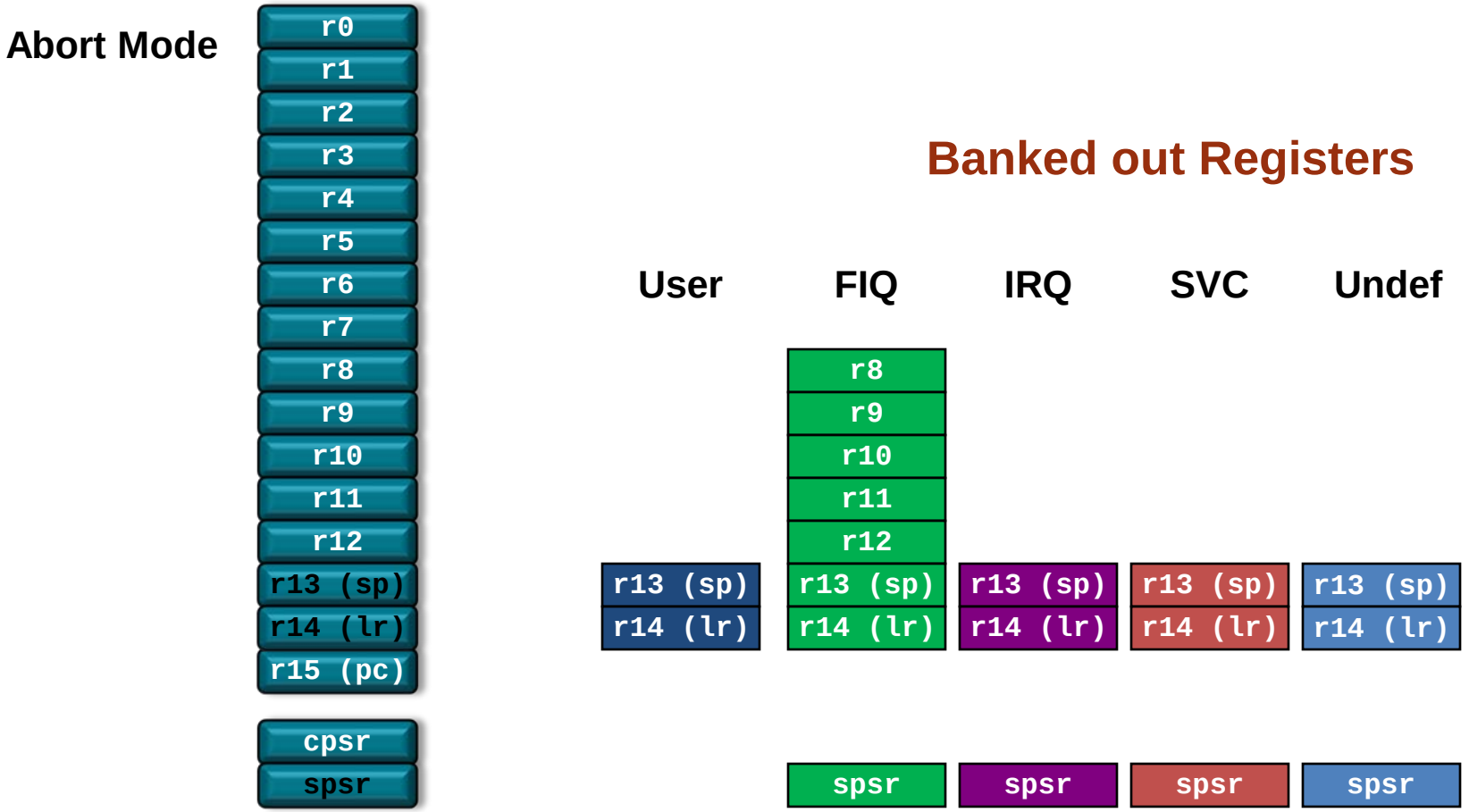
Processor Modes

ARM

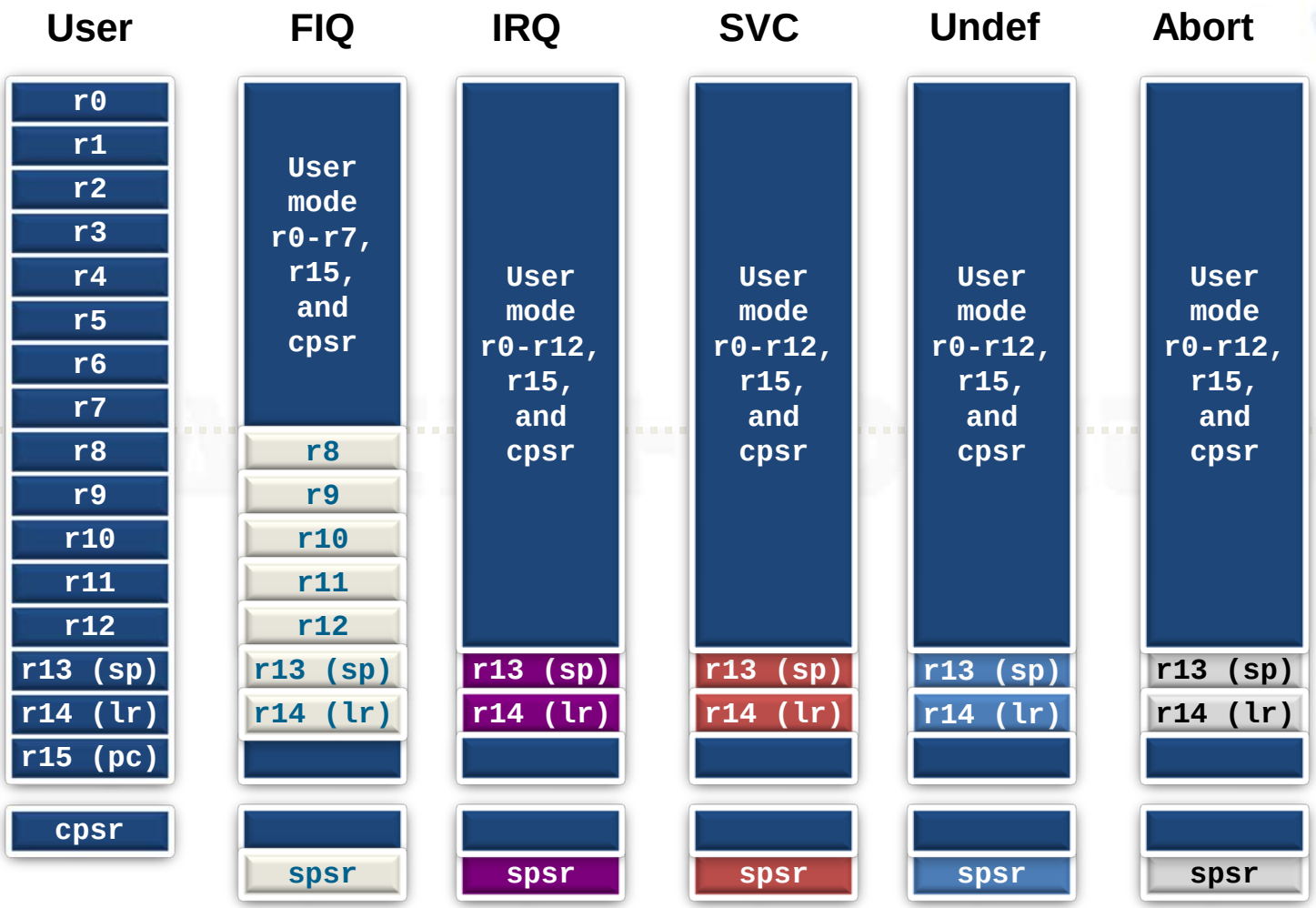
User	• Unprivileged Mode • Most tasks run on this mode
FIQ	• Entered when a high priority Interrupt is occurred
IRQ	• Entered when a low priority interrupt (normal) is occurred
Supervisory	• Entered on Reset • Entered on Software Interrupt
Abort	• To handle memory access violations
Undefined	• To handle un-defined instructions
System	• Privileged mode using the same registers as user mode

The ARM Register Set

Current Visible Registers



Register Organization Summary

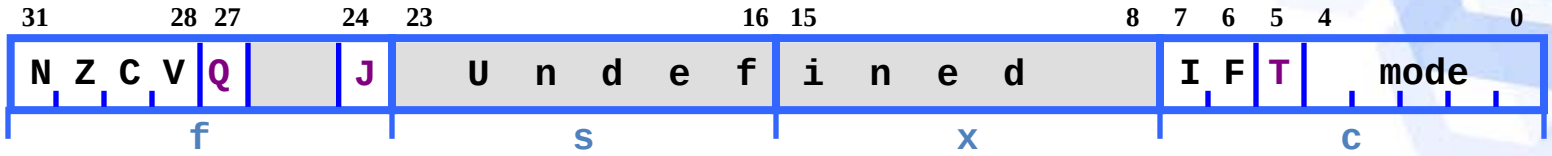


Note: System mode uses the User mode register set

The Registers

- ARM has 37 registers all of which are 32 bits long
  - 1 dedicated Program Counter
  - 1 dedicated Current Program Status Register (CPSR)
  - 5 dedicated Saved Program Status Register (SPSR)
  - 30 General Purpose Register
- The current processor mode governs the access of register banks
  - A particular set of r0-r12 registers
  - A particular r13 (stack pointer) and r14 (link register, LR)
  - The program counter (PC)
  - The current program status register (CPSR)
- Privileged Mode can access
  - A particular SPSR (Saved Program Status Register)

Program Status Registers



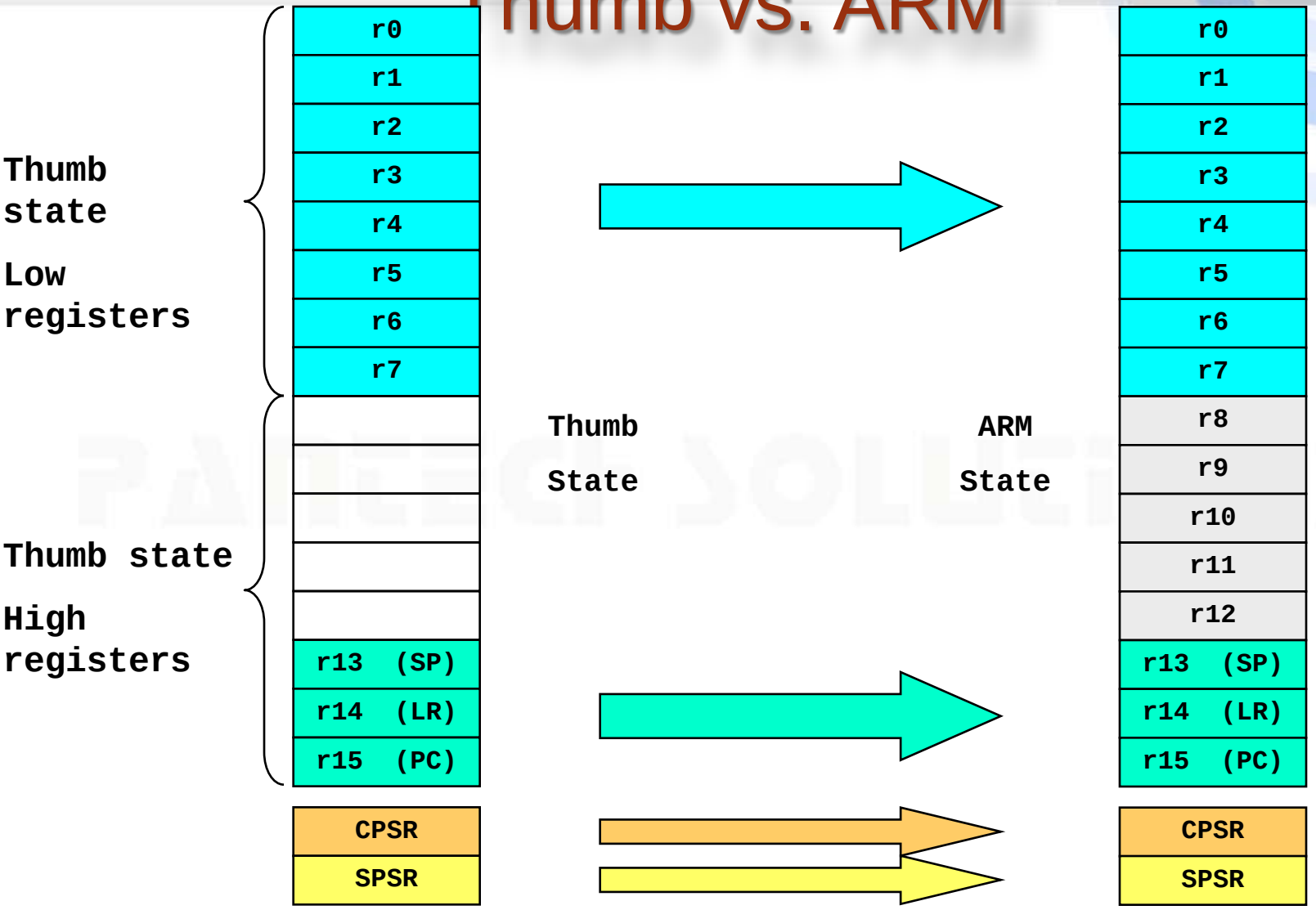
- Condition code flags
  - N = Negative result from ALU
  - Z = **Z**ero result from ALU
  - C = ALU operation **C**arried out
  - V = ALU operation overflowed
- Sticky Overflow flag - Q flag
  - Architecture 5TE/J only
  - Indicates if saturation has occurred
- J bit
  - Architecture 5TEJ only
  - J = 1: Processor in Jazelle state

- Interrupt Disable bits.
  - I = 1: Disables the IRQ.
  - F = 1: Disables the FIQ.
- T Bit
  - Architecture xT only
  - T = 0: Processor in ARM state
  - T = 1: Processor in Thumb state
- Mode bits
  - Specify the processor mode

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 - r12) 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



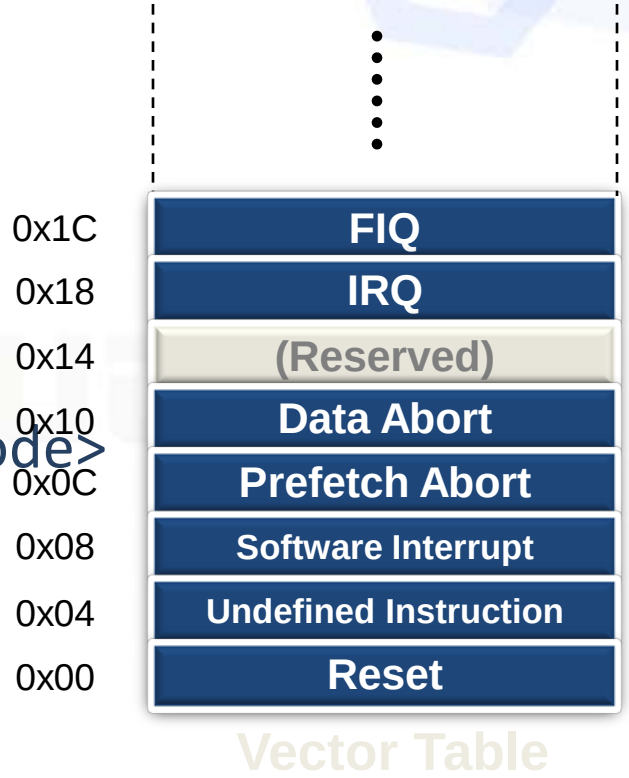
Program Counter (r15)

- When the processor is executing in ARM state:
  - All instructions are 32 bits wide
  - All instructions must be word aligned
  - Therefore the **pc** value is stored in bits [31:2] with bits [1:0] undefined (as instruction 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
  - Therefore the **pc** value is stored in bits [31:1] with bit [0] undefined (as instruction cannot be byte aligned).
- When the processor is executing in Jazelle state:
  - All instructions are 8 bits wide
  - Processor performs a word access to read 4 instructions at once

Exception Handling

- When an exception occurs, the ARM:
  - Copies CPSR into SPSR_<mode>
  - Sets appropriate CPSR bits
    - Change to ARM state
    - Change to exception mode
    - Disable interrupts (if appropriate)
  - Stores the return address in LR_<mode>
  - Sets PC to vector address
- To return, exception handler needs to:
  - Restore CPSR from SPSR_<mode>
  - Restore PC from LR_<mode>

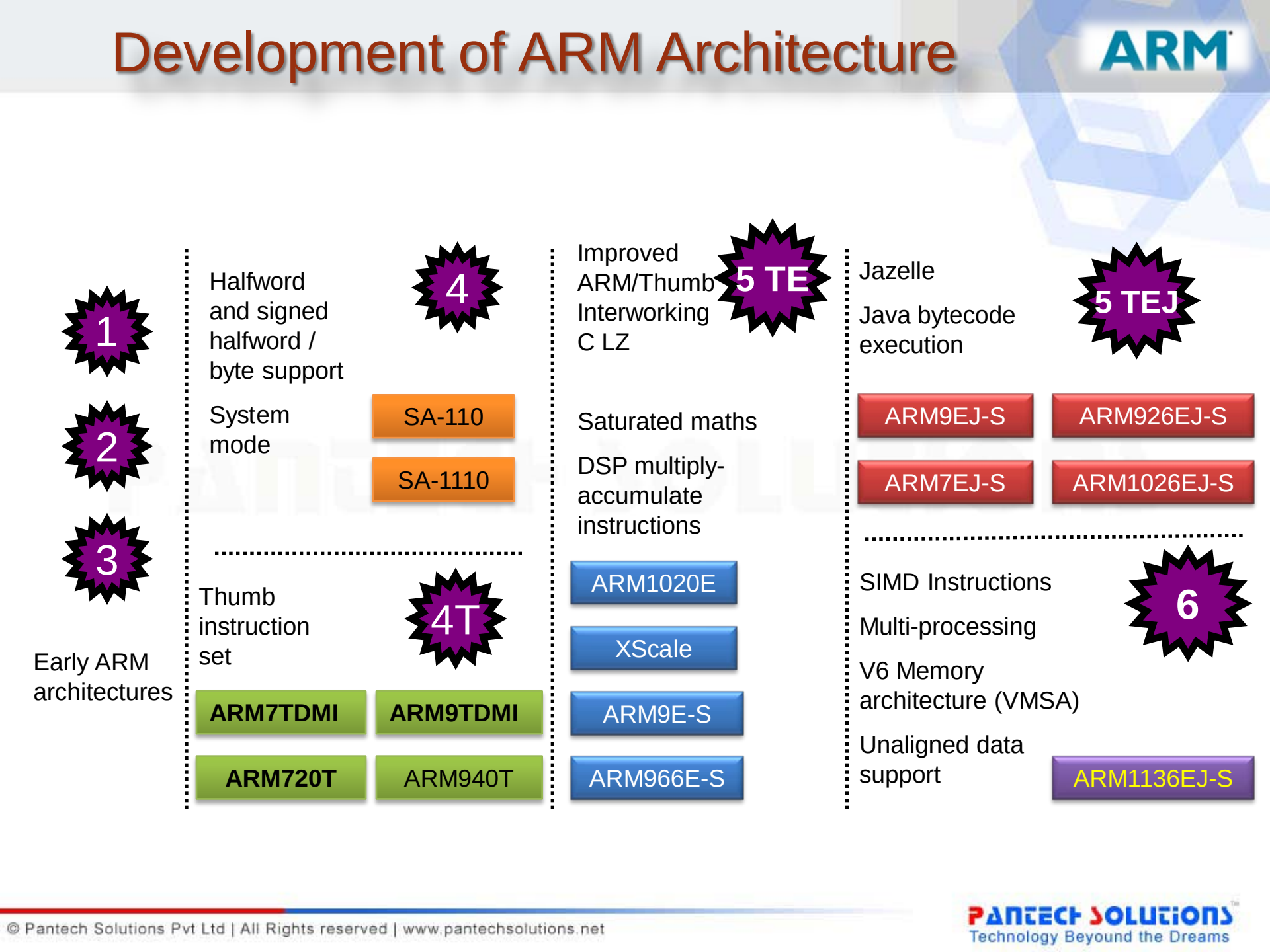
This can only be done in ARM state.



Vector Table

Vector table can be at **0xFFFF0000** on ARM720T and on ARM9/10 family devices

Development of ARM Architecture



Agenda

- Introduction to ARM Ltd
- Programmers Model
- Instruction Set
- System Design
- Development Tools

ARM7 TDMI - S

- The ARM7TDMI-S is based on ARM7 Core
  - 3 Stage Pipeline
  - Von-Neumann Architecture
  - CPI ~ 1.9
    - T – Thumb instruction Sets
    - D – includes debug extensions
    - M – Enhanced Multipliers
    - I – Core has Embedded ICE logic extensions
    - S – Fully Synthesizable soft IP

Instruction Pipeline

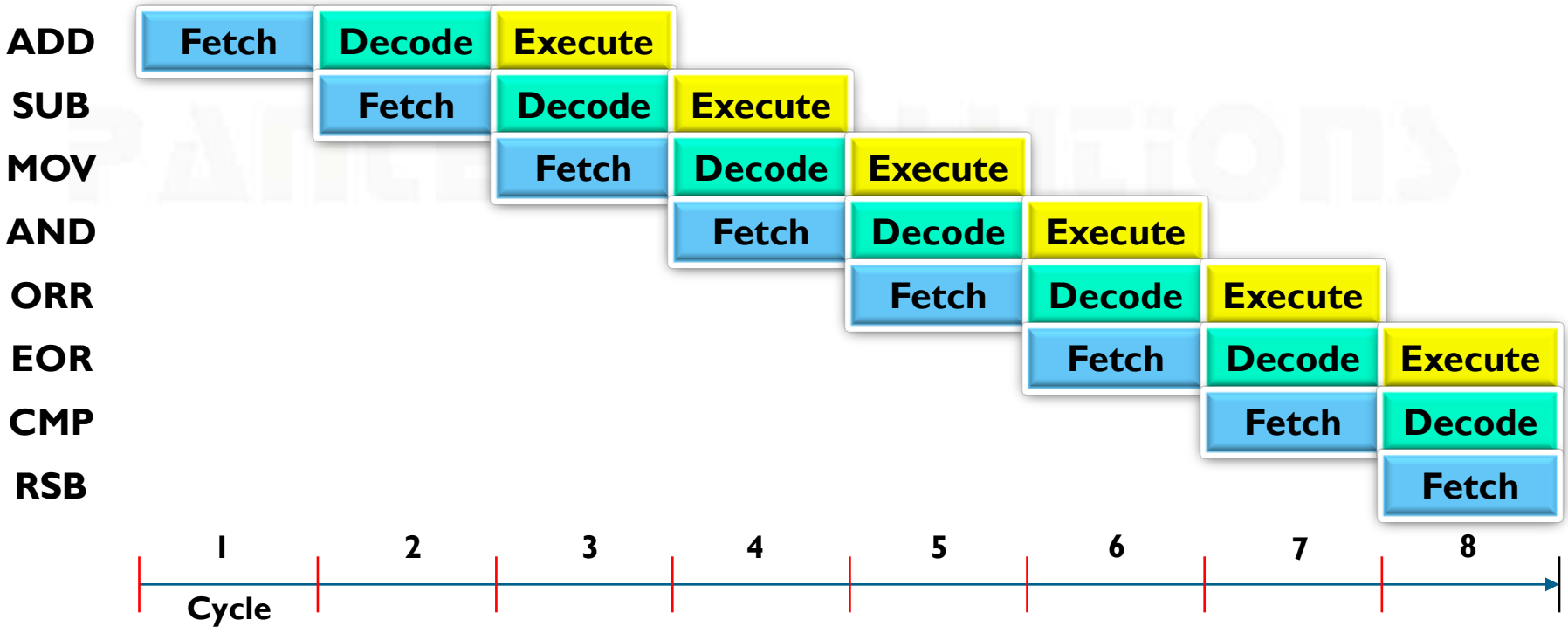
- The ARM7TDMI-S core uses a pipeline
  - Increase the speed of the flow of instructions
  - Enables several operations to take place simultaneously
  - Program Counter (PC) points to instruction being fetched rather than that being executed
- During normal pipelined operation
  - An instruction x is executed
  - Instruction (x-1) is being decoded
  - Instructino (x-2) is being fetched

3-Stage Instruction Pipeline

ARM	Thumb		
PC	PC	Fetch	Instruction Fetched from Memory
PC - 4	PC - 2	Decode	Thumb only: Thumb instruction decompressed to ARM instruction Instruction decoded
PC - 8	PC - 4	Execute	Registers read from Register Bank, Shift and ALU operations performed, Registers written back to Register Bank

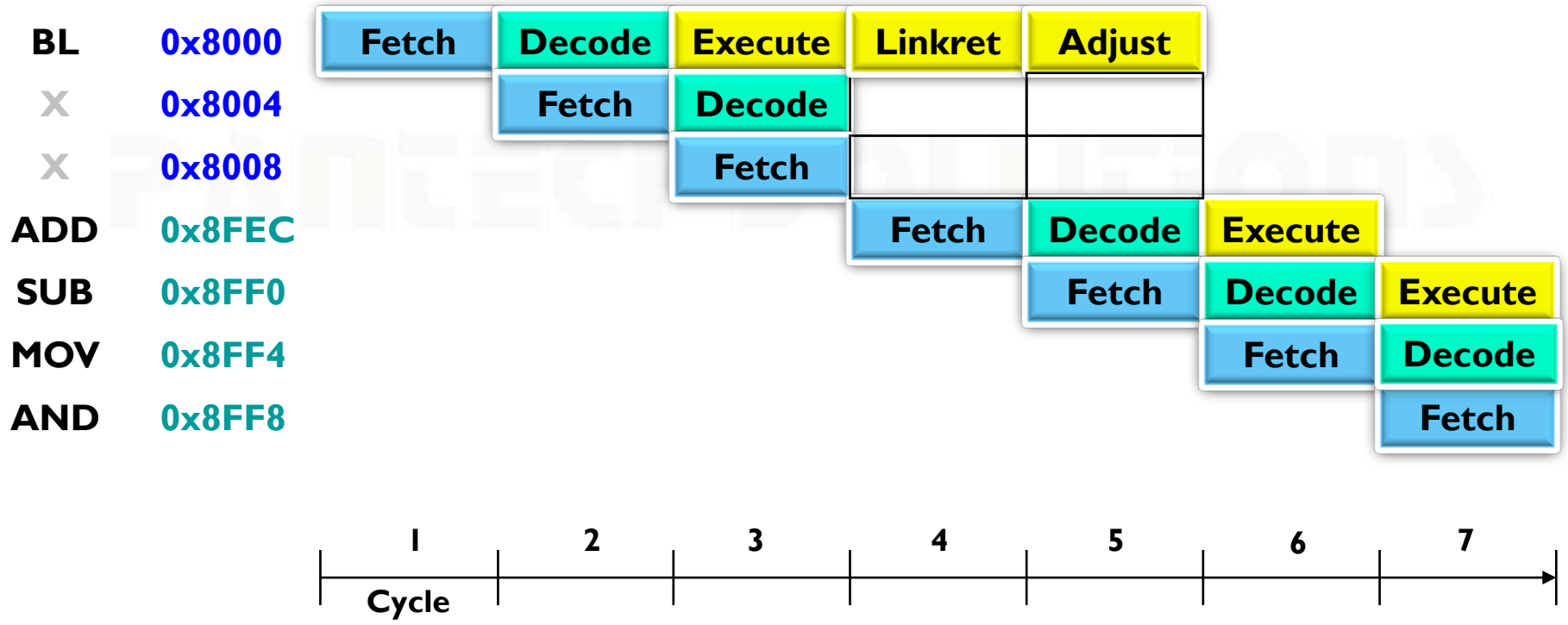
Optimal Pipelining

- In this example it takes 6 clock cycles to execute 6 instructions
- All operations are on registers (single cycle instructions)
- Clock cycles per instruction (CPI) = 1



Branch Pipeline Example

- Branches break the pipeline
- Example in ARM state



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

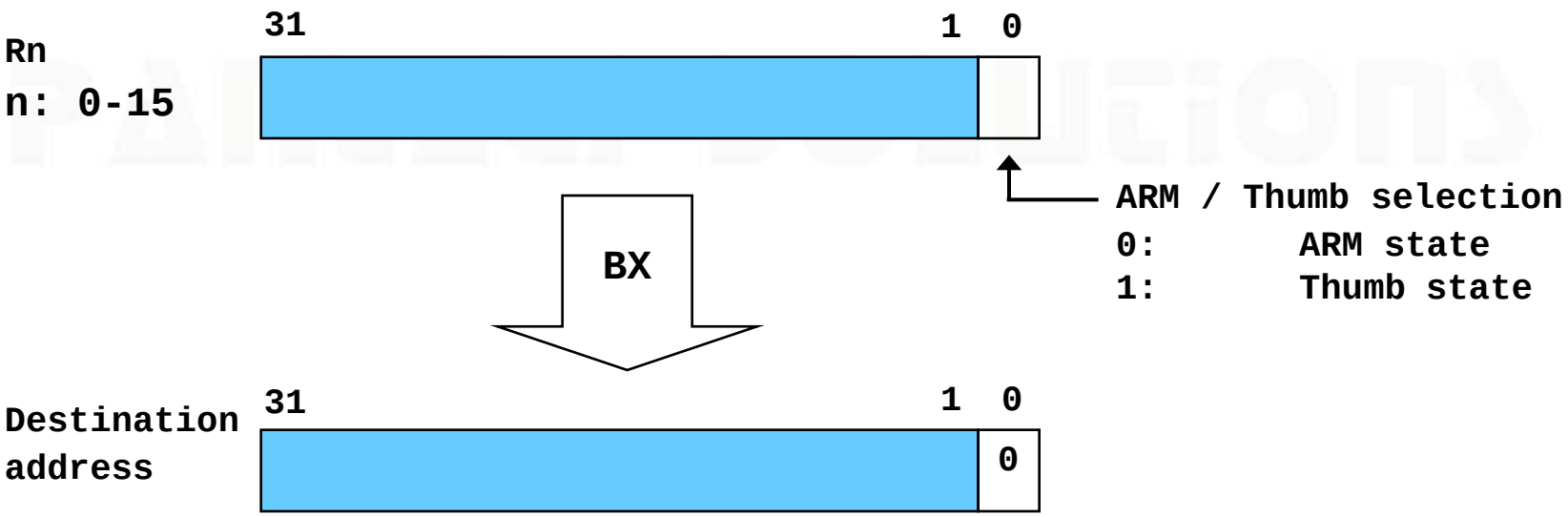
Mnemonic	Description
EQ	Equal
NE	Not equal
CS / HS	Carry Set / Unsigned higher or same
CC / LO	Carry Clear / Unsigned lower
MI	Negative
PL	Positive or zero
VS	Overflow
VC	No overflow
HI	Unsigned higher
LS	Unsigned lower or same
GE	Signed greater than or equal
LT	Signed less than
GT	Signed greater than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

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)

ARM and Thumb Interworking

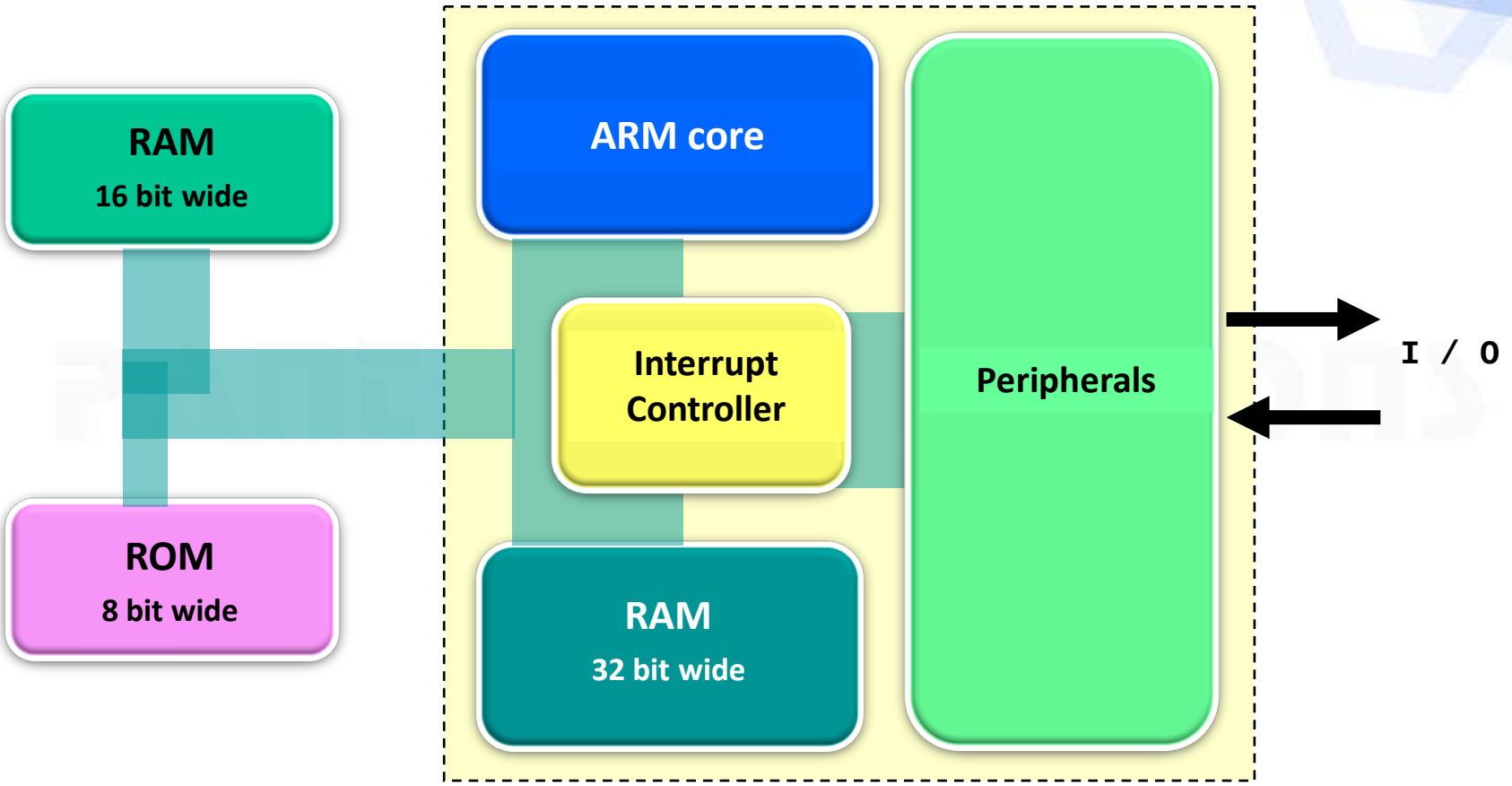
- Switch between ARM state and Thumb state using BX instruction
  - In ARM state: BX<condition> Rn
  - In Thumb state: BX Rn



Agenda

- Introduction to ARM Ltd
- Programmers Model
- Instruction Set
- System Design
- Development Tools

Example ARM based System

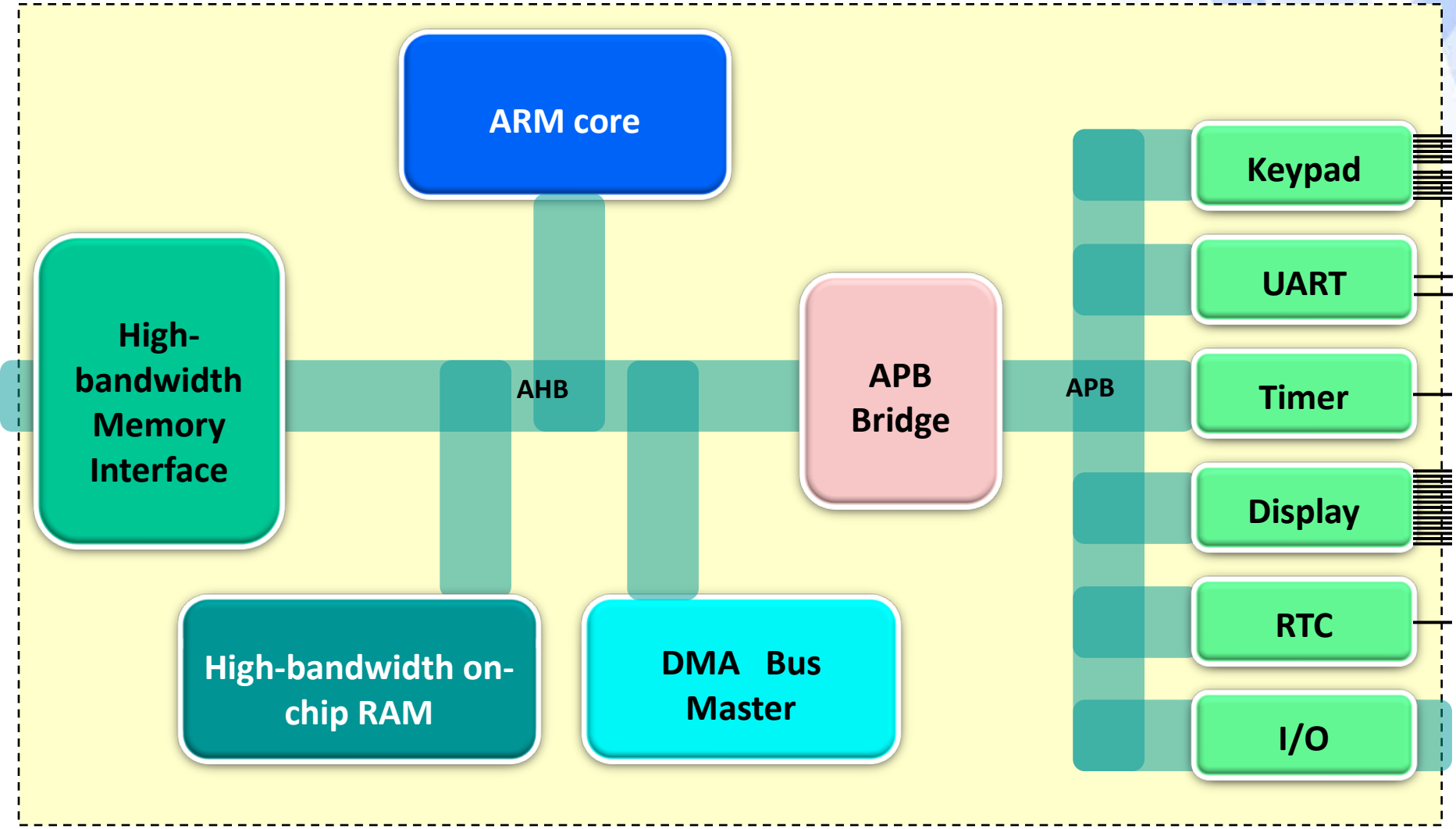


AMBA®
System Architecture

ARM

- Advanced Microcontroller Bus Architecture
  - 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

- Advanced High-Performance Bus
  - high-performance
  - pipelined
  - fully-synchronous backplane
  - multiple bus masters
- Advanced Peripheral Bus / VLSI Peripheral Bus
  - low-power
  - non-pipelined
  - simple interface
  - wait support (VPB)

Agenda

- Introduction to ARM Ltd
- Programmers Model
- Instruction Set
- System Design
- Development Tools

The RealView Product Families

Compilation Tools

ARM Developer Suite (ADS) – Compilers (C/C++ ARM & Thumb), Linker & Utilities

RealView Compilation Tools (RVCT)

Debug Tools

AXD (part of ADS)

Trace Debug Tools

Multi-ICE

Multi-Trace

RealView Debugger (RVD)

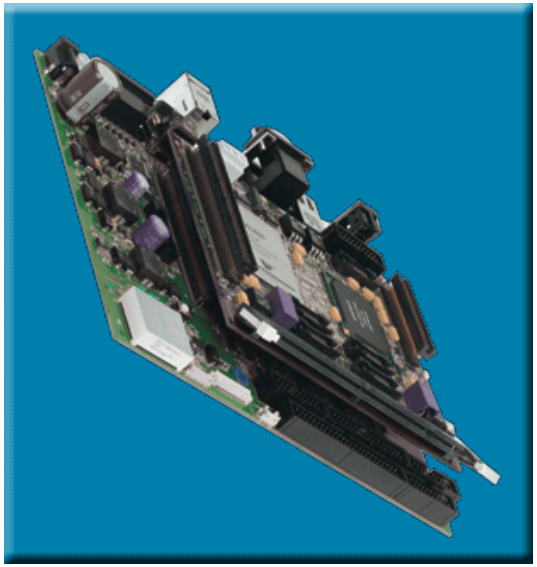
RealView ICE (RVI)

RealView Trace (RVT)

Platforms

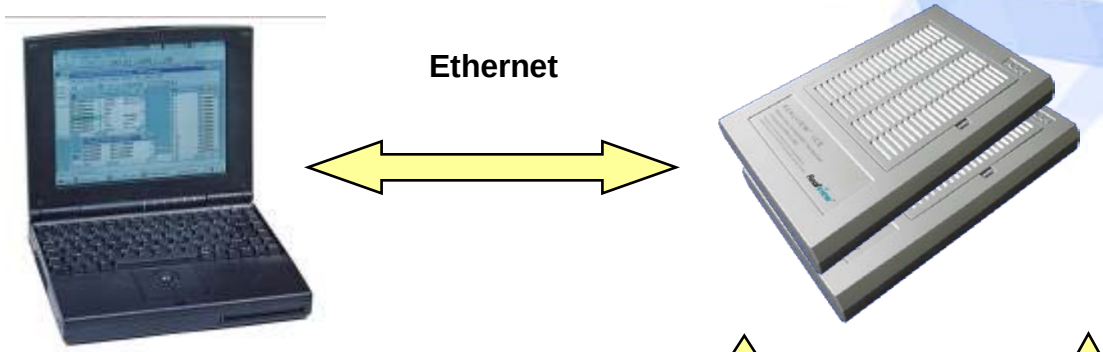
ARMulator (part of ADS)

Integrator™ Family

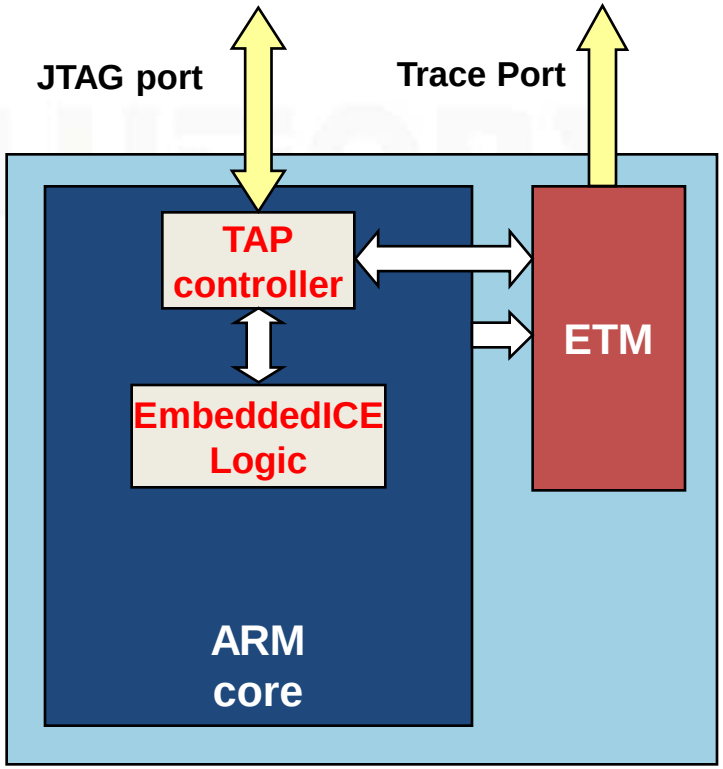


RealView ARMulator ISS (RVISS)

ARM Debug Architecture



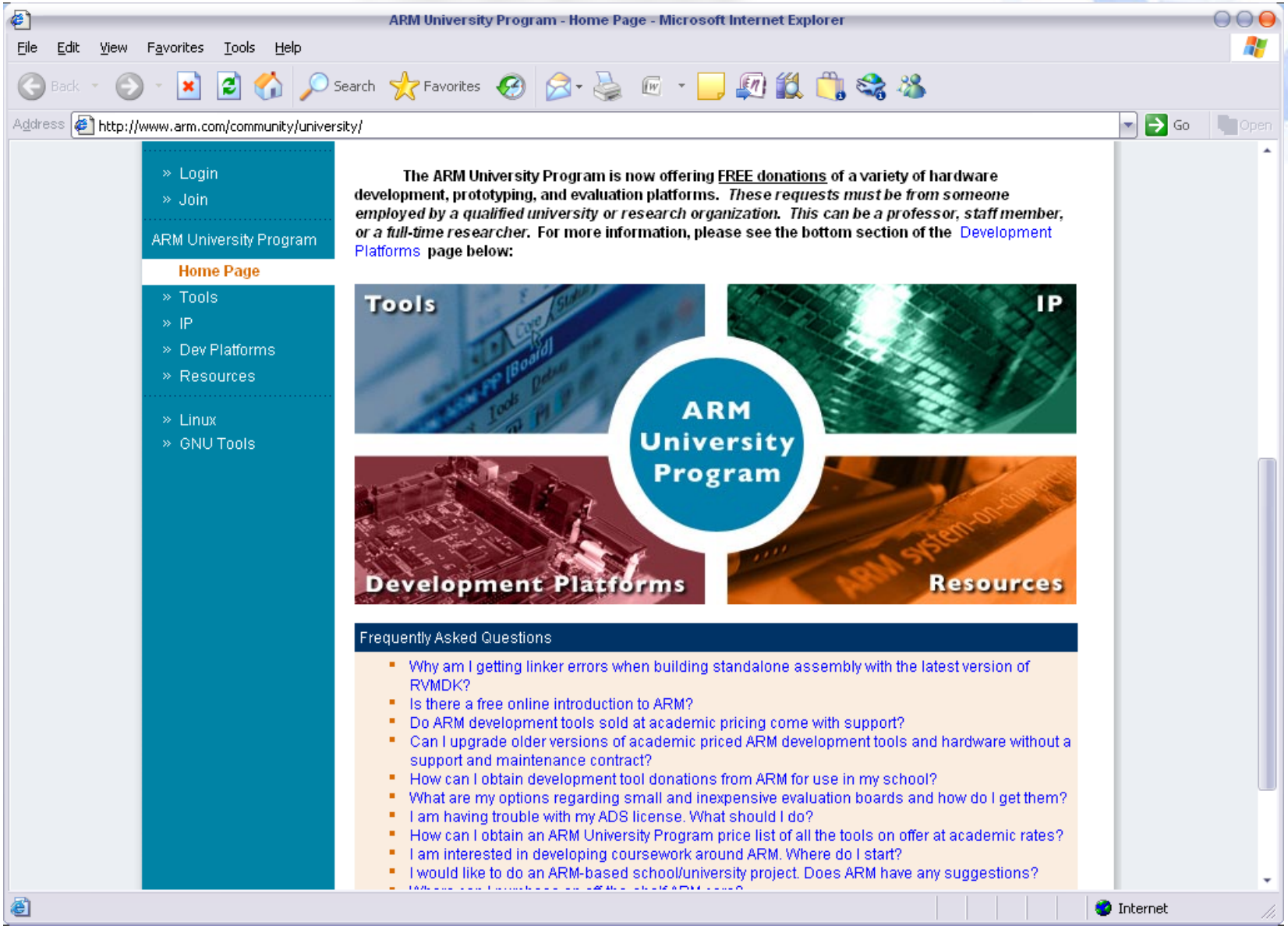
- EmbeddedICE Logic
  - Provides breakpoints and processor/system access
- JTAG interface (ICE)
  - Converts debugger commands to JTAG signals
- Embedded trace Macrocell (ETM)
  - Compresses real-time instruction and data access trace
  - Contains ICE features (trigger & filter logic)
- Trace port analyzer (TPA)
  - Captures trace in a deep buffer



References

- www.arm.com
- http://www.arm.com/markets/
- http://in.youtube.com/watch?v=4VRtujwa_b8
  - ARM Webinars | ARM Video Search
- http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.set.appnotes/

References



Questions ?

THANK YOU!