ARM Microprocessor and ARM-Based Microcontrollers

Nguatem William

24th May 2006
A Microcontroller-Based Embedded System
# Roadmap

1. **Introduction**
   - ARM
   - ARM Basics

2. **ARM Extensions**
   - Thumb
   - Jazelle
   - NEON & DSP Enhancement
   - Summary

3. **ARM Processor Cores**

4. **ARM based System**
   - Microcontroller
   - ARM Products
Roadmap

1. **Introduction**
   - ARM
   - ARM Basics

2. **ARM Extensions**
   - Thumb
   - Jazelle
   - NEON & DSP Enhancement
   - Summary

3. **ARM Processor Cores**

4. **ARM based System**
   - Microcontroller
   - ARM Products
# ARM History

- ARM - Acorn RISC Machine from Acorn Computers Ltd. of Cambridge, UK.
- In 1990, ARM Ltd. was established and ARM was renamed as Advanced RISC Machines.

## ARM Ltd.

- A semiconductor IP - Intellectual Property company.
- Licenses IP cores to partner companies e.g. Nokia, Philips Semiconductors.
- Also develop technologies to assist with the designing of the ARM architecture.
- ARM is not a chip producer.
IP - Intellectual Property

- ARM provides hard and soft views to licensees (RTL and synthesis flows GDSII layout) IP
- Soft views include gate level netlists (RTL source code) → synthesizable from licensees using a suitable gate library
  Hard IP are the final GDSII layout given to the customer
- OEMs must use hard views to protect ARM IP
ARM Basics

**ARM feature (I)**

Partly from Berkeley RISC concept

- A load-store architecture.
- Fixed-length 32-bit instructions
- 3-address instruction format
- Pipelined architecture
- Conditional execution of all instructions
- Extensible ISA through hardware Coprocessors
- The ability to perform a general shift operation and a general ALU operation in a single instruction that executes in a single clock cycle
rejected from Berkeley RISC concept

- Register Window.
- Delayed branches
Modes (I)

7 operating modes

- user: Unprivileged, normal execution mode
- FIQ: High priority (fast) interrupt raised
- IRQ: Low priority Interrupt
- svc: Software interrupt (SWI) is executed
- Abort: Handling of memory access violations
- system: Run privileged task
- Undefined: Undefined instructions
Question
Why FIQ and IRQ ?

Answer
- FIQ has a higher priority than IRQ
- Gives a better mapping of different interrupt sources
- FIQ has extra bank registers than IRQ
  → faster than IRQ
ARM Basics

Registers

ARM has 37 registers all of which are 32-bits long
- 1 dedicated program counter
- 1 dedicated current program status register
- 5 dedicated saved program status registers
- 30 general purpose registers
Accessible Registers

These registers cannot all be seen at once. The processor state and operating mode dictate which registers are available to the programmer.
**Processor Status Register - CPSR**

- **Condition code flags**
  - \( N = \) Negative result from ALU
  - \( Z = \) Zero result from ALU
  - \( C = \) ALU operation Carried 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
Data Sizes and Instruction Sets

Data sizes

The ARM is a 32-bit architecture. When used in relation to the ARM:

- Byte means 8 bits
- Halfword means 16 bits (two bytes)
- Word means 32 bits (four bytes)

ISA

Most ARMs implement two instruction sets.

- 32-bit ARM Instruction Set
- 16-bit Thumb Instruction Set

Jazelle cores can also execute Java bytecode
Endianess

- Neutrality to Endianess.
- Can be configured at power-up as either little- or big-endian mode.
- Default alignment is little-endian due to many little-endian peripheral component available.
Coproprocessors

Coprocessors interface
Provide support for hardware coprocessors.

Advantages
- Extends the instruction set, e.g. On-Chip control of MMU and Cache, floating point arithmetic
Roadmap

1 Introduction
   - ARM
   - ARM Basics

2 ARM Extensions
   - Thumb
   - Jazelle
   - NEON & DSP Enhancement
   - Summary

3 ARM Processor Cores

4 ARM based System
   - Microcontroller
   - ARM Products
Thumb

Thumb (I)

- Thumb is a 16-bit instruction set.
- Core has additional execution state - Thumb
- Switch between ARM and Thumb using BX instruction
- Not a complete ISA

Difference to ARM Inst.

- Conditional execution is not used
- Source and destination registers identical
- Thumb bit in CPRS is set
- Inline barrel shifter not used
Thumb (II)

Example: ADD Rd. #constant

```
1110 001 0100 1 0 Rd 0 Rd 0000 #imm8
```

```
15 13 12 11 10 8 7 0
0 0 1 10 Rd #imm8
```

- **Thumb (II)**
- **Example:** ADD Rd, #constant
- **Thumb code:**
  - 15 13 12 11 10 8 7 0
  - 0 0 1 10 Rd #imm8
- **ARM code:**
  - 1110 001 0100 1 0 Rd 0 Rd 0000 #imm8

This diagram illustrates theThumb (II) format and how it relates to the ARM code.
Thumb (III)

Thumb-2

16-bit coding of more ARM instruction.

Pros and Cons of Thumb

+ Excellent code density for minimal system size.
- No direct access of Status registers while in Thumb state
- No conditional execution, excepting branch instructions
- With 32-bit memory, the ARM code is 40% faster than Thumb code
+ With 16-bit memory, the Thumb code is 45% faster than ARM code
A typical embedded system, e.g. a mobile phone, will include a small amount of fast 32-bit memory (to store speed-critical DSP code) and 16-bit off-chip memory to store the control code.
Jazelle (I)

- Hardware accelerated java code mechanism.
Jazelle

Jazelle (II)

Features

- Processor fetches one word containing 4-javabytes.
- J-bit of status register set

Jazelle types

- Jazelle DBX - Direct Byte eXecution: supports only javabyte codes
- Jazelle RCT - Run time CompilaTion: extension of Thumb-2, supports different VM.
TrustZone

- Hardware based security mechanism
- Complete code separation

TrustZone adds a “parallel world” to allow trusted programs and data to be safely separated from the operating system and applications.
NEON & DSP Enhancement

NEON.

- Hardware acceleration for multimedia applications
- Combines 64-bit and 128-bit hybrid SIMD
- Separate execution i.e complete instruction architecture.

DSP Extension

Added DSP instructions to ARM ISA
Conclusion

Note
- Extra logic needed for the hardware extension: e.g. Thumb decompression unit.
- Increase die size and cost.
- Simple implementation
- Very efficient
Roadmap

1. Introduction
   - ARM
   - ARM Basics

2. ARM Extensions
   - Thumb
   - Jazelle
   - NEON & DSP Enhancement
   - Summary

3. ARM Processor Cores

4. ARM based System
   - Microcontroller
   - ARM Products
ARM7 Core

Features

- 32-bit RISC Architecture
- Von Neumann Architecture
- 3-Stage Pipeline - Fetch, Decode, Execute
- Most instructions execute in a single cycle.
- ARMv4 ISA
- Supports up to 16 coprocessors.
ARM7 Family

**ARM740T**
- Embedded RTOS Core
- 60-75 MHz (0.18μm w/c)
- 2.5mm² (0.18μm)
- 0.65mW/MHz
- Hard Macro IP
- 8K 4-way SA cache

**ARM7TDMI-S**
- Base Integer core
- 100 MHz (0.18μm w/c)
- ~0.7mm² (0.18μm)
- Synthesizable (Soft IP)

**ARM7TDMI**
- Base Integer core (Hard Macro IP)
- 60-100 MHz (0.18μm w/c)
- 0.53mm², 0.25mW/MHz (0.18μm)
- NEW: low-voltage layout

**ARM720T**
- Platform OS Core
- 60-75 MHz (0.18μm w/c)
- 2.93mm² (0.18μm)
- 0.65mW/MHz
- Hard Macro IP
- 8K 4-way SA cache

**MMU**
- 8K Cache
- ARM7
- ETM7 i/interface
- ASB i/interface

**Added Q4 2000**

Stated operating frequencies represent worst-case speed for a range of processes.
ARM9 Core

Features

Performance of the ARM7 Von Neumann architecture limited by the available bandwidth - memory accessed on almost every cycle either to fetch an instruction or to transfer data.

- Harvard architecture improves CPI - Clock cycles Per Instruction
- Higher performance core than ARM7
- Five-stage pipeline - Fetch, Decode, Execute, Memory, and Write
ARM9E

Features

- ARM9 core + DSP extensions
- Enhanced multiplier for DSP performance
- On-Chip debug hardware

Benefits

- Single engine for both DSP and control code
- Simple single memory system.
- Reduced chip complexity, die size and power consumption
- Single toolkit support with ARMs Development and Debug tools, giving faster time-to-market
Five Families of ARM Processor IP

ARM preserves SW & HW investment through code and process portability

- **ARM7™**
  - Low Power
  - Area Efficient
  - Code Density
  - 3 Stage Pipe
  - 4 product offerings

- **ARM9™**
  - Dual Caches
  - Performance
  - 3 product offerings

- **ARM9E™**
  - Config $/TCM
  - DSP instr.
  - Java™ in HW
  - 3 product offerings

- **ARM10E™**
  - Multiple $/TCM
  - Dual 64bit I/O
  - 6 Stage Pipe
  - 2 product offerings
  - New: ARM1026EJ-S

- **ARM11™**
  - ARMv6 Architecture
  - Secure Apps
  - Performance
  - Power, Area
  - Code Density
  - 4 product offerings
Key technology additions by architecture generation

- Dynamic compiler support
- VFPv3
- NEON™ advanced SIMD

- Thumb®-2 (option)
- Thumb-2 (mandated)
- TrustZone™
- SIMD

- DSP
- VFPv2
- Jazelle®

- V5 ARM9E
- V6 ARM11
- V7 A&R Cortex A/R
- V7 M Cortex M

Execution environments: Improved memory use

Improved media and DSP

Low cost MCUs
Roadmap

1. Introduction
   - ARM
   - ARM Basics

2. ARM Extensions
   - Thumb
   - Jazelle
   - NEON & DSP Enhancement
   - Summary

3. ARM Processor Cores

4. ARM based System
   - Microcontroller
   - ARM Products
ActelCoreMP7 and Subsystem
- ARM7TDMI-S
- 32/16-bit RISC architecture
- ARM and Thumb instruction sets
- Embedded real-time debug and JTAG
ARM Products

ARM Powered Products
Summary

- ARM Cores are IPs
- Hybrid instruction encoding offers better efficiency in embedded applications
- Backward software compatibility for all new cores
Questions?