

# Nintendo DS

## The Nintendo DS

Daniel McCarney & Jason Gagnon



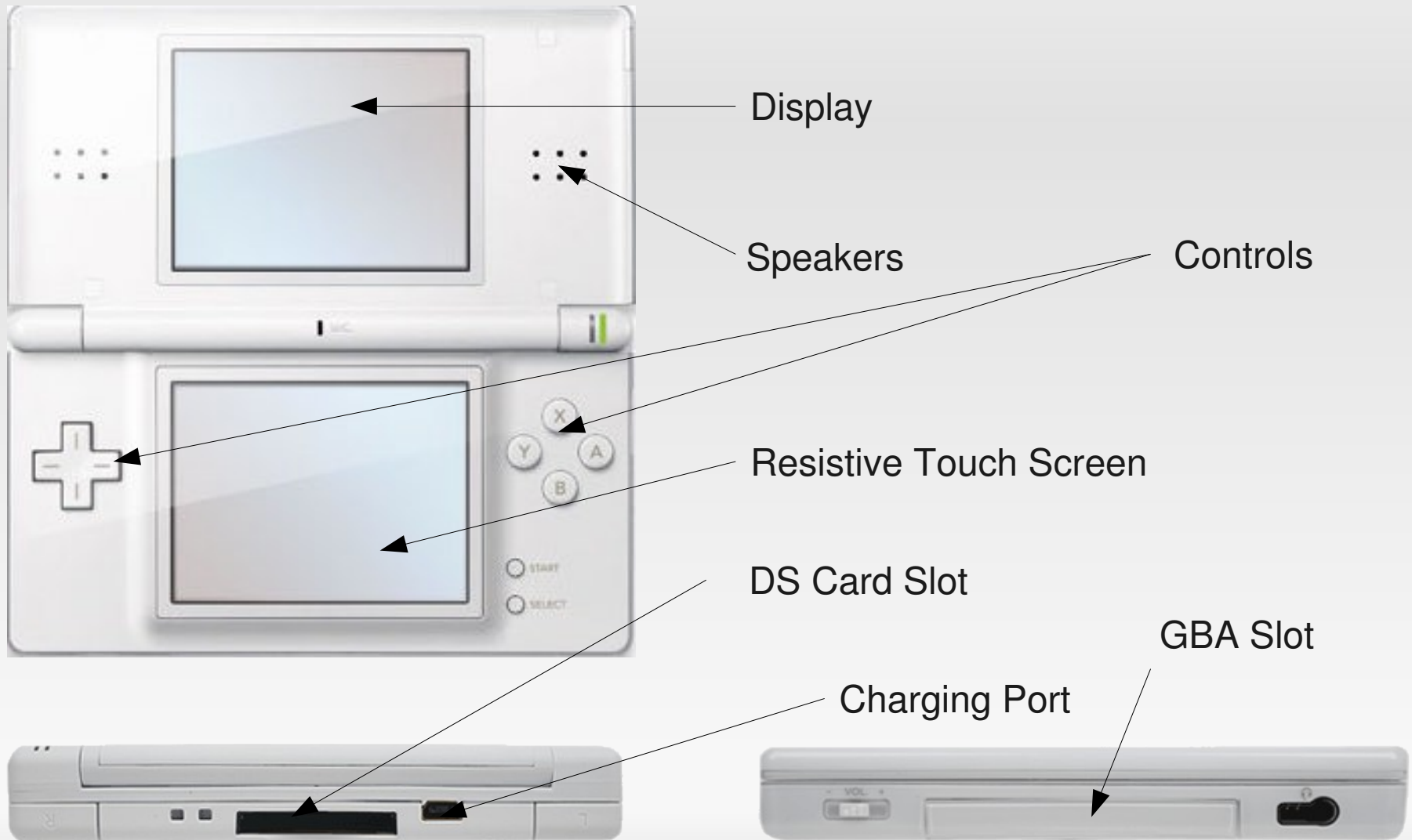
Dual Screens

Clamshell Design

Speakers

GBA Cart Slot & DS Slot

# Design Overview



# Hardware Specifications

Dual TFT LCD screens

256 x 192 pixels each

CPUs

Arm 7

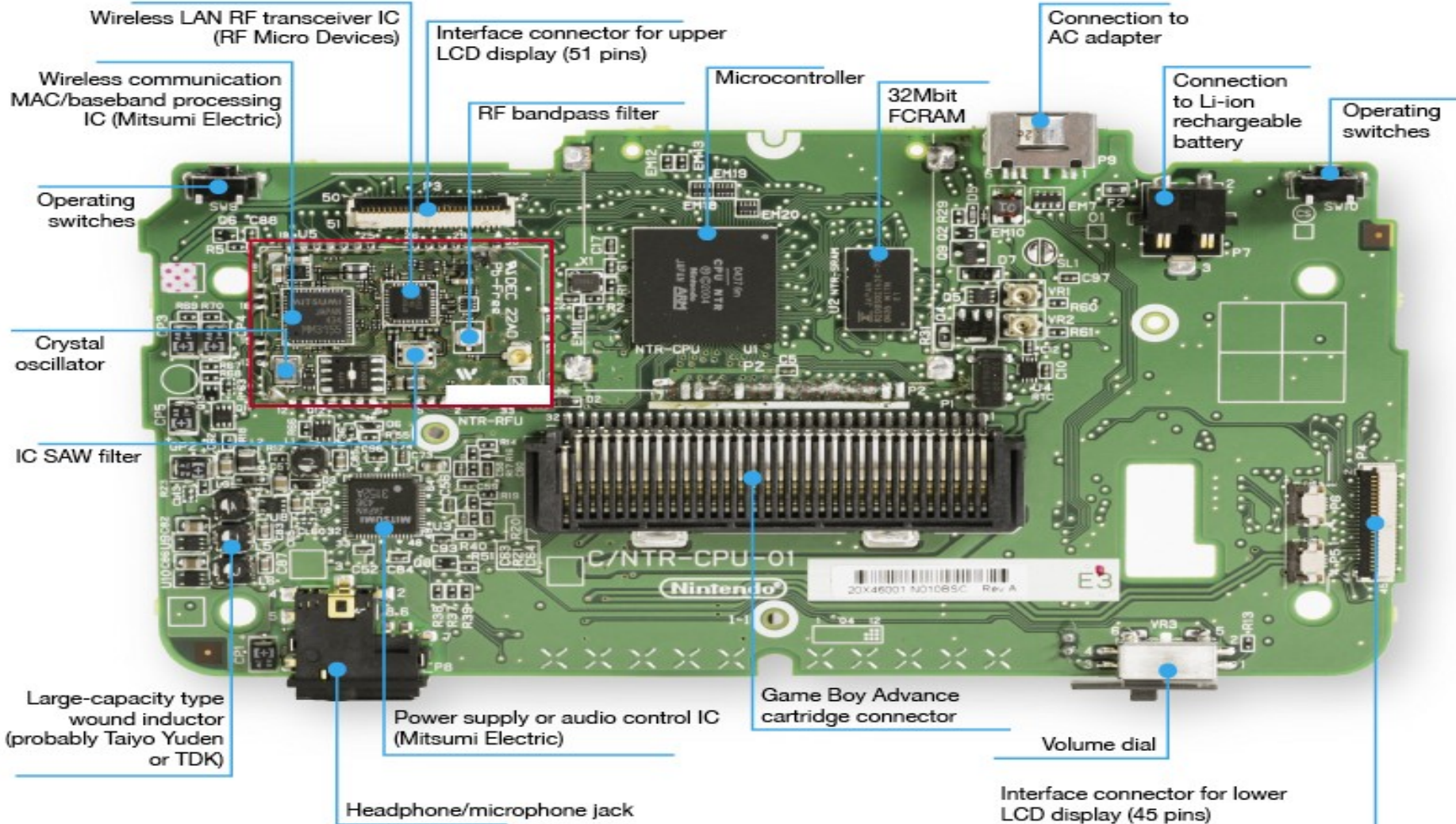
Arm 9

4 MB of RAM

1.65v internal power

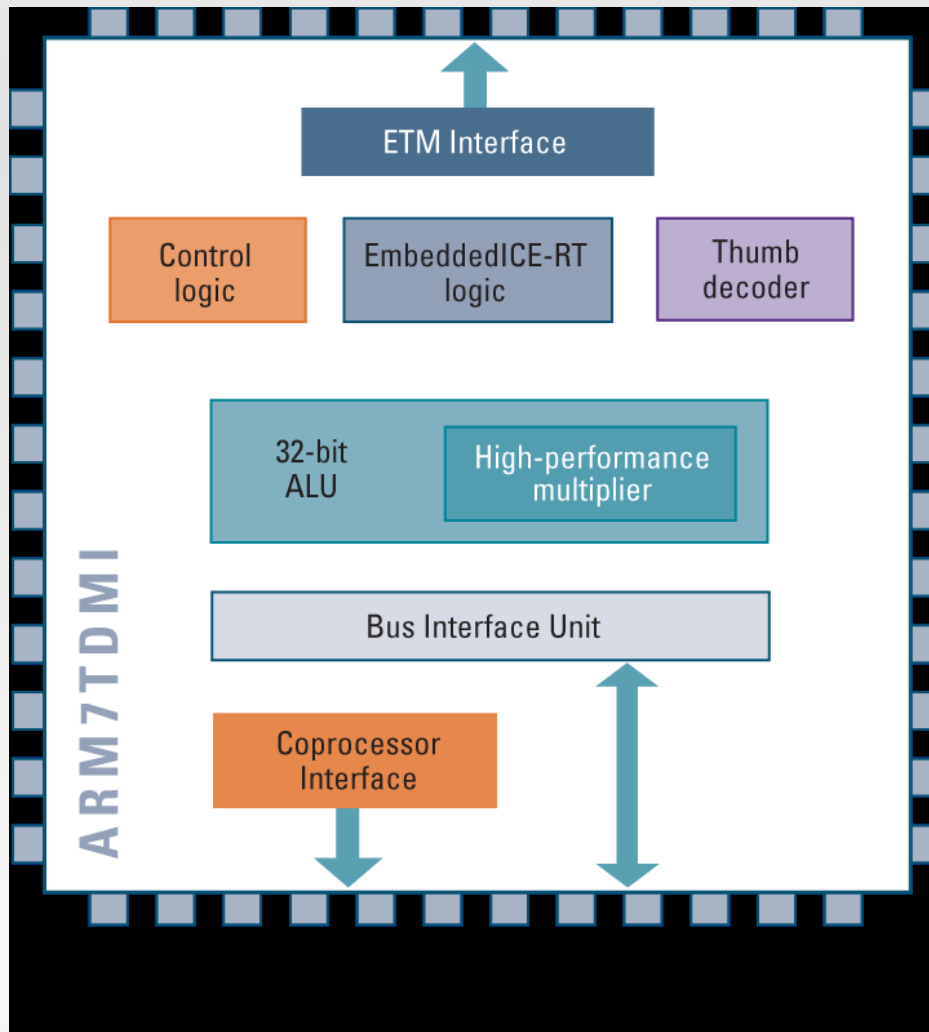
256kb Flash Memory (Firmware etc)

# Hardware



**Fig 1 Key Circuitry Clumped on One Side** The photo is close to the actual size – 141 x 76mm (longest area). The wireless module and microcontroller were covered by a metal shielding plate.

# ARM7



15 MIPS @ 33mhz

32/16-bit RISC architecture (ARM v4T)

32-bit ARM instruction set for maximum performance and flexibility

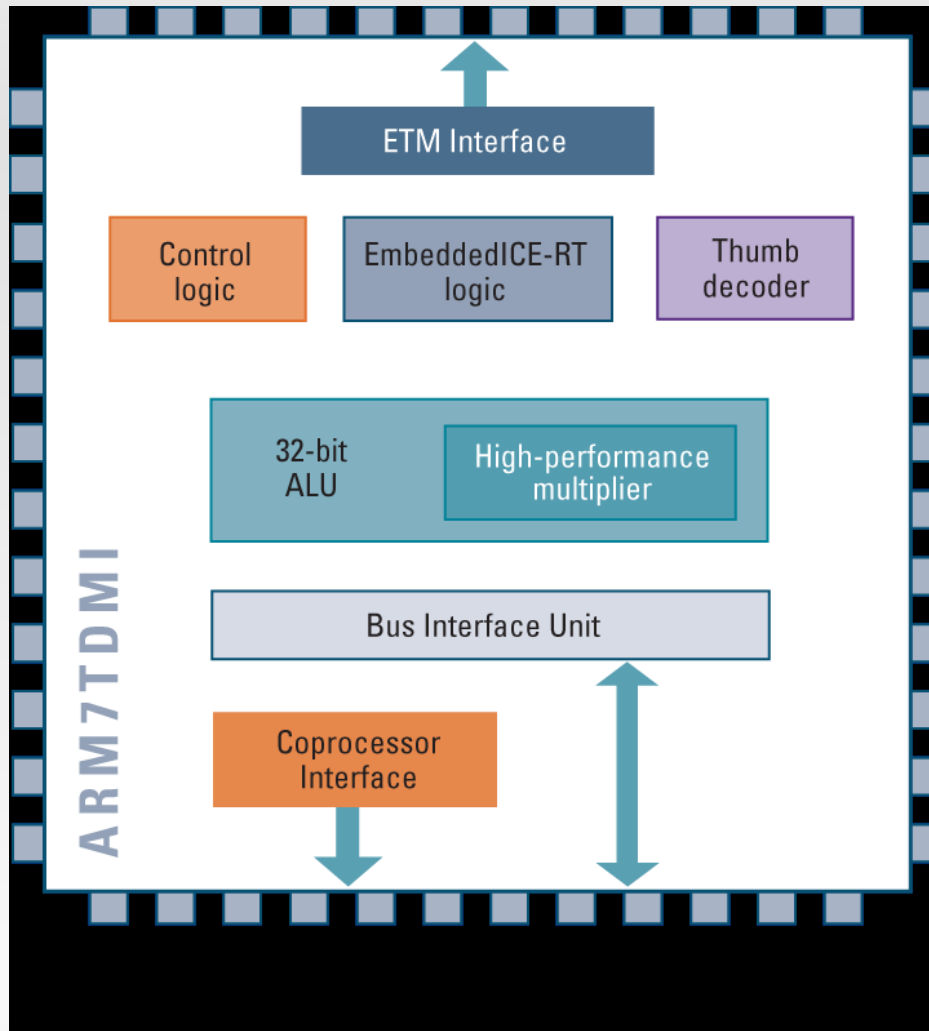
16-bit Thumb instruction set for increased code density

Unified bus interface, 32-bit data bus carries both instructions and data

Three-stage pipeline

32-bit ALU

# ARM7 (con't)



Low power consumption

Code written for ARM7TDMI is binary-compatible with other members of the ARM7 Family

Forwards compatible with the ARM9 and ARM9E families

ARM and Thumb instructions sets can be combined with low overhead.

# Harvard Architecture

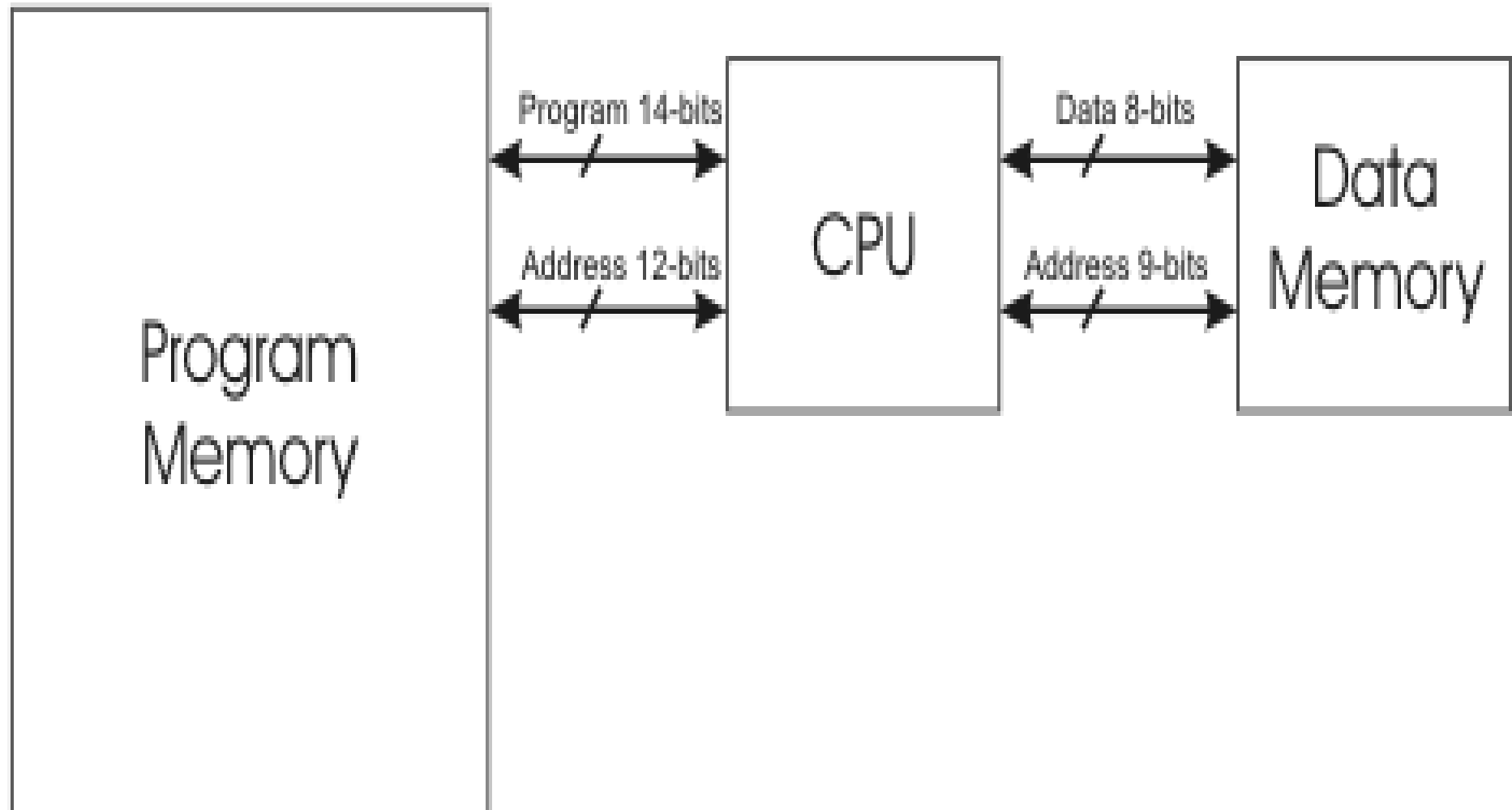
Separate data and instruction buses

Separate data and instruction storage

Allows for differing bus widths

Can read instructions and perform data manip. in parallel

# Harvard Architecture (Continued)



The given bus widths are examples only!

# Arm 9

32-bit RISC CPU

67 mhz clockspeed

Harvard Architecture

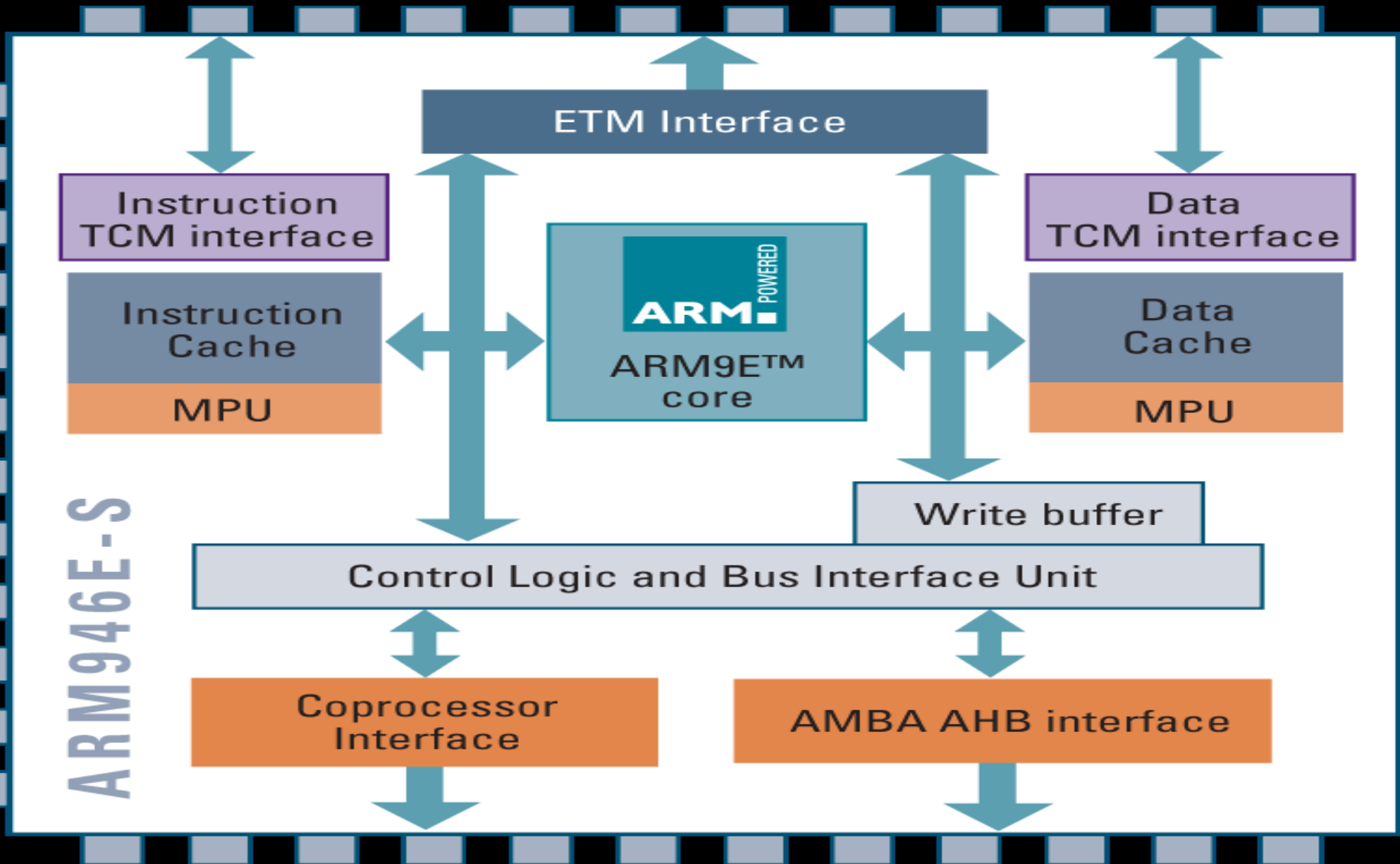
Dual instruction sets (ARM and Thumb)

Features on-chip caching for

Instructions

Data

# Arm 9 (Continued)



# Main Memory

4 MB Block

Holds ARM9 executable + game data

ARM9 and ARM7 can access main memory  
simultaneously

Bus conflicts between processors delegated to  
processor with priority (ARM7 by default)

# Cache Memory

ARM7 has 64kb of fast ram (IWRAM)

Contains both the ARM7 executable and data

ARM9 has a data cache and instruction cache

Main memory is cache-able by default so all data and code from main memory will be stored in cache temporarily

# Memory Map

## Memory Map

ARM 9				
Name	Start Address	Stop Address	Size	Wait State
Main	0x02000000	0x023FFFFFF	4MB	?
BIOS	0xFFFF0000	0xFFFF7FFF	32KB	?
ITCM	0x00000000	0x00007FFF	32KB	?
DTCM	0x0B000000	0x0B003FFF	16KB	?
Shared WRAM Bank 0	0x03000000	0x03003FFF	16KB	?
Shared WRAM Bank 1	0x03004000	0x03007FFF	16KB	?
ARM 7				
Main	0x02000000	0x023FFFFFF	4MB	?
BIOS	0x00000000	0x00003FFF	16KB	?
IWRAM	0x03800000	0x0380FFFF	64KB	?
Shared WRAM Bank 0	0x03000000	0x03003FFF	16KB	?
Shared WRAM Bank 1	0x03004000	0x03007FFF	16KB	?
Video RAM				
Main OAM	0x07000000	0x070003FF	1KB	?
Sub OAM	0x07000400	0x070007FF	1KB	?
Main Palette	0x05000000	0x050003FF	1KB	?
Sub Palette	0x05000400	0x050007FF	1KB	?
Bank A	0x06800000	0x0681FFFF	128KB	?
Bank B	0x06820000	0x0683FFFF	128KB	?
Bank C	0x06840000	0x0685FFFF	128KB	?
Bank D	0x06860000	0x0687FFFF	128KB	?
Bank E	0x06880000	0x0688FFFF	64KB	?
Bank F	0x06890000	0x06893FFF	16KB	?
Bank G	0x06894000	0x06897FFF	16KB	?
Bank H	0x06898000	0x0689FFFF	32KB	?
Bank I	0x068A0000	0x068A3FFF	16KB	?
Virtual Video RAM				
Main Background	0x06000000	0x0607FFFF	512KB	?
Sub Background	0x06200000	0x0621FFFF	128KB	?
Main Sprite	0x06400000	0x0643FFFF	256KB	?
Sub Sprite	0x06600000	0x0661FFFF	128KB	?

## ARM

ITCM = Instruction Cache

DTCM = Data Cache

IWRAM = Instruction/Data Cache

## Video Ram

Colour Palettes

Video banks A-I can store graphic sprites etc.

# Instruction Set

Dual RISC Instruction Sets:

32 Bit ARM Instruction Set

16 Bit Thumb Instruction Set

Thumb Instruction Set:

Subset of most common RISC instructions in 16 bit opcodes

Improves code size

Decompressed to ARM instructions

# Dsi – Next Generation



Digital Cameras inside  
and out

No GBA Slot

*Single* ARM processor  
clocked at 133 MHz

16 MB of RAM

256 MB of internal  
Flash memory