COMPUTER SYSTEMS ORGANIZATION

ARM Processor Architecture -- Spring 2010 -- IIIT-H -- Suresh Purini

Algorithms, Data Structures and Programs

Algorithms + Data Structures = Programs

Niklaus Wirth

Programming Languages

A Programming Language provides

- Data Abstractions
  - int, float, bool etc. data types.
  - Mechanism for hierarichal composition of new Data Abstractions
    - Structures, arrays, Unions etc.
- Data Processing Abstractions
  - Arithmetic and Boolean Operations, String Operations etc.
- Control Abstractions
  - while, if, for constructs etc.

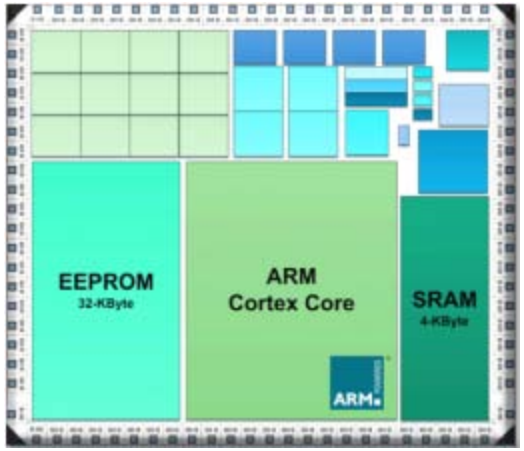
Question: What has all these things got to do with Assemble Language Programming, in particular ARM assembly language?

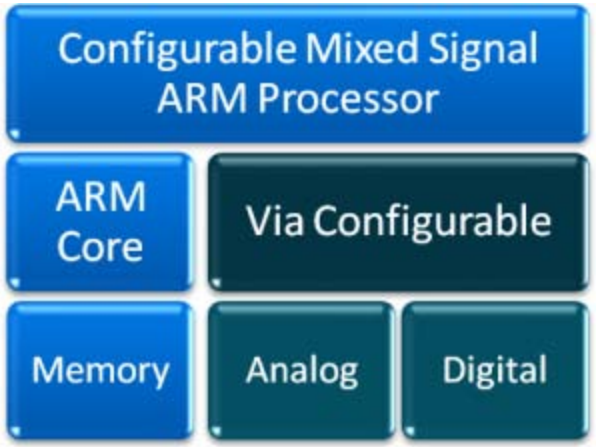
Key Idea: We should be able to realize these abstractions through the ISA of a given processor!

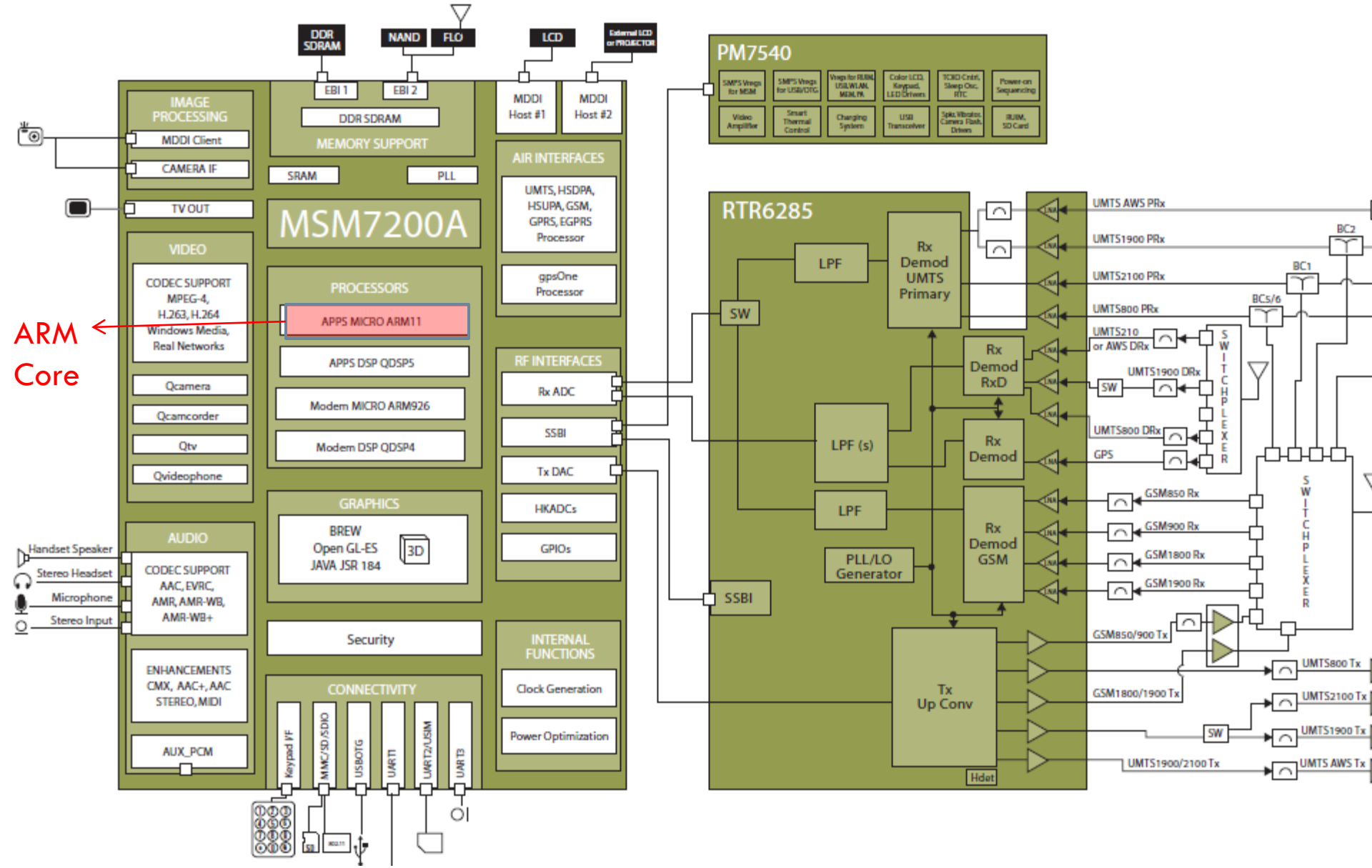
ARM Processors – History

- ARM Processor is a Reduced Instruction Set Computer (RISC)
  - First ARM processor was designed between 1983 and 1985 at Acorn Computers Ltd. of Cambridge, England.
  - Called Acorn RISC Machine then.
- In 1990
  - Acorn Computers Ltd. became ARM Limited.
  - Acorn RISC Machine is renamed Advanced RISC Machine.
- ARM Ltd. Licensees its processor design technology as Intellectual Property (IP)
  - It doesn't manufacture processors, it licenses other companies to use their ARM core designs in their custom chips.
  - Compare it with Intel line of processors

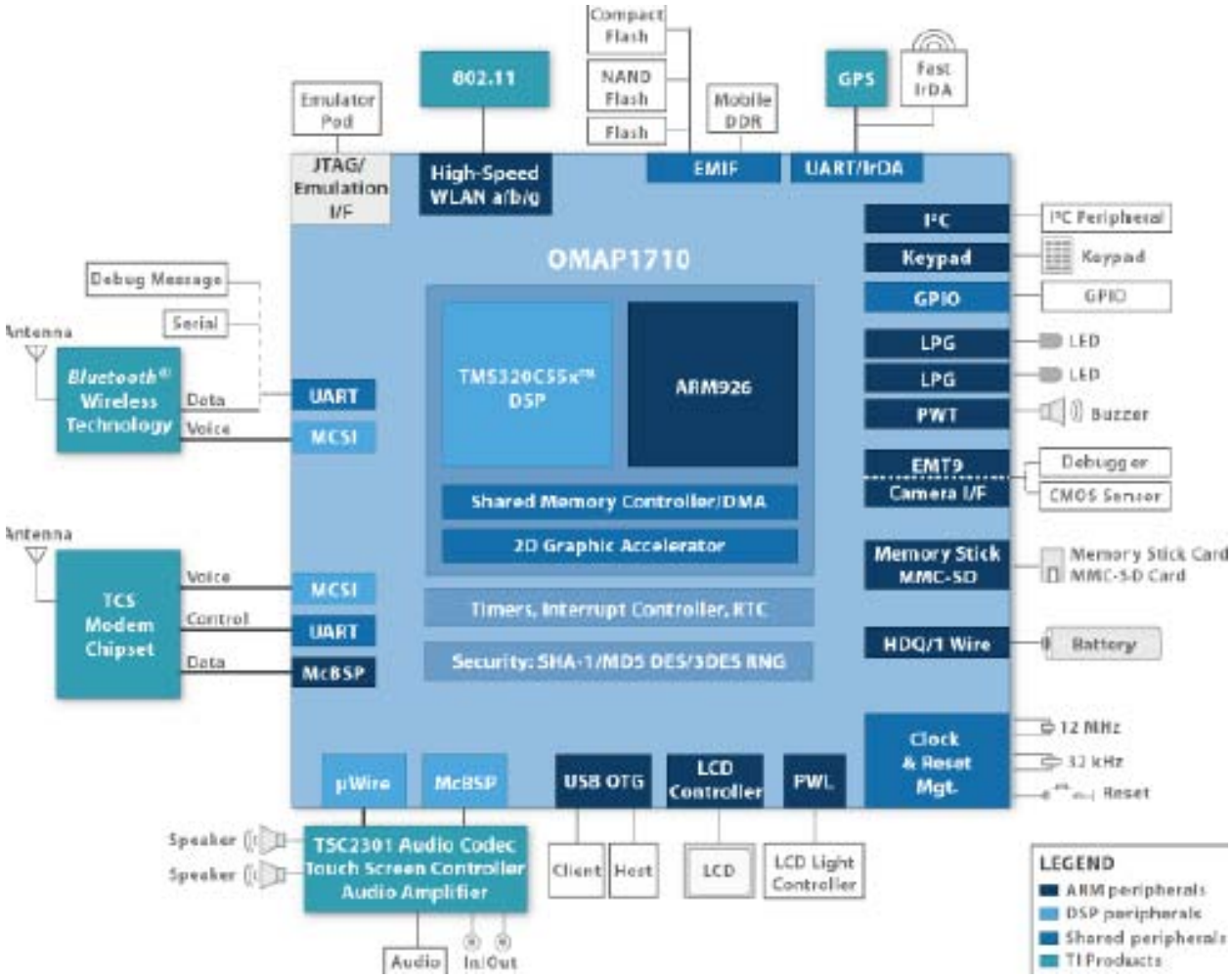
ARM Cores


EEPROM
32-KByte
ARM
Cortex Core
SRAM
4-KByte
ARM


Configurable Mixed Signal
ARM Processor
ARM
Core
Via Configurable
Memory
Analog
Digital


DDR SDRAM
NAND
FLO
LCD
External LCD or PROJECTOR
PM7540
IMAGE PROCESSING
MDDI Client
CAMERA IF
TV OUT
EBI 1
EBI 2
DDR SDRAM
MEMORY SUPPORT
SRAM
PLL
MDDI Host #1
MDDI Host #2
AIR INTERFACES
UMTS, HSDPA, HSUPA, GSM, GPRS, EGPRS Processor
gpsOne Processor
MSM7200A
VIDEO
CODEC SUPPORT MPEG-4, H.263, H.264 Windows Media, Real Networks
Qcamera
Qcamcorder
Qtv
Qvideophone
PROCESSORS
APPS MICRO ARM11
APPS DSP QDSP5
Modem MICRO ARM926
Modem DSP QDSP4
RF INTERFACES
Rx ADC
SSBI
Tx DAC
HKADCs
GPIOs
ARM Core
GRAPHICS
BREW
Open GL-ES
JAVA JSR 184
3D
AUDIO
CODEC SUPPORT AAC, EVRC, AMR, AMR-WB, AMR-WB+
ENHANCEMENTS CMX, AAC+, AAC STEREO, MIDI
AUX_PCM
Handset Speaker
Stereo Headset
Microphone
Stereo Input
Security
CONNECTIVITY
Keypad I/F
MMC/SD/SDIO
USBOTG
UART1
UART2/USIM
UART3
INTERNAL FUNCTIONS
Clock Generation
Power Optimization
RTR6285
LPF
Rx Demod UMTS Primary
SW
Rx Demod RxD
LPF (s)
Rx Demod
LPF
Rx Demod GSM
PLL/LO Generator
SSBI
Tx Up Conv
Hdet
UMTS AWS PRx
UMTS1900 PRx
UMTS2100 PRx
UMTS800 PRx
UMTS210 or AWS DRx
UMTS1900 DRx
UMTS800 DRx
GPS
GSM850 Rx
GSM900 Rx
GSM1800 Rx
GSM1900 Rx
GSM850/900 Tx
GSM1800/1900 Tx
UMTS1900/2100 Tx
UMTS800 Tx
UMTS2100 Tx
UMTS1900 Tx
UMTS AWS Tx
BC2
BC1
BCs/6
SWITCHPLEXER
SWITCHPLEXER

ARM Cores


Compact Flash
802.11
NAND Flash
Flash
GPS
Fast IrDA
Emulator Pod
Mobile DDR
JTAG/ Emulation I/F
High-Speed WLAN a/b/g
EMIF
UART/IrDA
I²C
I²C Peripheral
OMAP1710
Keypad
Keypad
GPIO
GPIO
Debug Message
Serial
LPG
LED
LPG
LED
PWT
Buzzer
Antenna
Bluetooth® Wireless Technology
Data
Voice
UART
McSI
TMS320C55x™ DSP
ARM926
EMT9
Camera I/F
Debugger
CMOS Sensor
Shared Memory Controller/DMA
2D Graphic Accelerator
Antenna
TCS Modem Chipset
Voice
Control
Data
McSI
UART
McBSP
Memory Stick MMC-SD
Memory Stick Card
MMC-SD Card
Timers, Interrupt Controller, RTC
HDQ/1 Wire
Battery
Security: SHA-1/MD5 DES/3DES RNG
Clock & Reset Mgt.
12 MHz
32 kHz
Reset
µWire
McBSP
USB OTG
LCD Controller
PWL
Speaker
Speaker
TSC2301 Audio Codec Touch Screen Controller Audio Amplifier
Client
Host
LCD
LCD Light Controller
Audio
In/Out
LEGEND
ARM peripherals
DSP peripherals
Shared peripherals
TI Products

RISC versus CISC

- Two ISA design philosophies
  - RISC – Reduced Instruction Set Computer
  - CISC – Complex Instruction Set Computer

	RISC	CISC
1.	Fixed Instruction size with few formats.	Variable Instruction Length.
2.	Load-Store Architecture. Instruction operands should always reside in registers.	Instruction Operands can reside in memory also.
3.	Large bank of general purpose registers.	Many special purpose registers.
4.	Simple Addressing Modes.	Can have complex addressing modes.
5.	Few Data Types (typically integer and float)	Could provide support for more data types like Strings.
6.	Berkeley RISC , Stanford MIPS, ARM, HP's PA-RISC	Intel x86 line of processors

RISC or CISC – Which way should we go?

- RISC pros
  - Simple, fast (pipelined) and power efficient hardware implementations.
  - Good for compiler writers.
- RISC cons
  - Less code density
- CISC pros
  - Makes the life of an Assembly language programmer better!
  - Good code density – Assuming the compiler is able to make good use of CISC features.
- CISC cons
  - Complex hardware, not so Power efficient, hard to come up with pipelined implementations.

ARM follows RISC philosophy but borrows some CISC ideas. ARM processor design tries to achieve " High Performance + Good Code Density + Low Power Consumption"

Principles of ISA Design

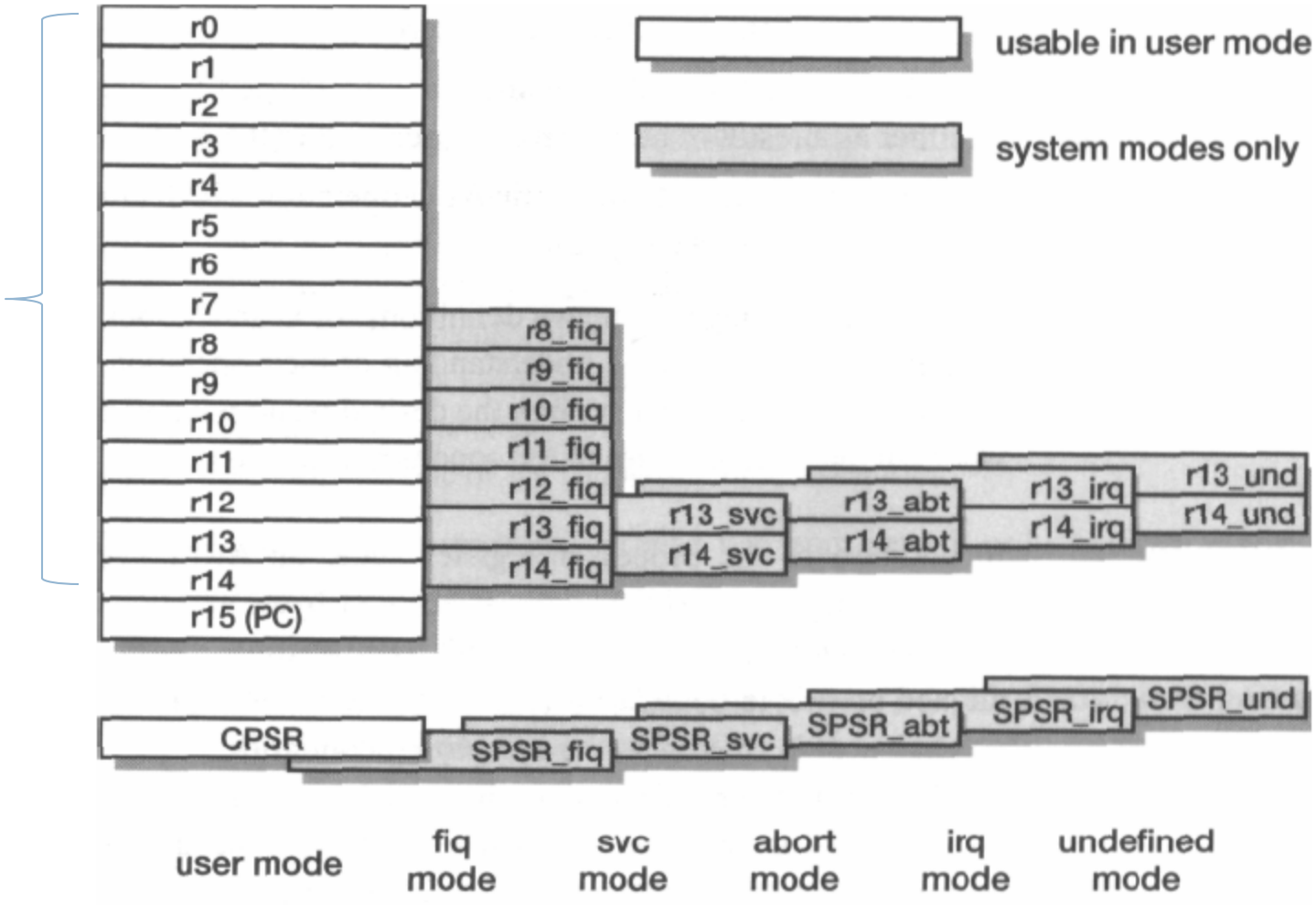
It is easy to see by formal-logical methods that there exist certain [instruction sets] that are in abstract adequate to control and cause the exception of any sequence of operations….The really decisive considerations from the present point of view, in selecting an [instruction set], are more of practical nature: simplicity of the equipment demanded by the [instruction set], and the clarity of its application to the actually important problems together with the speed of handling those problems.

- Burks, Goldstine and Von Neumann, 1947

ARM ISA – Register Set and Modes of Operation

ARM processor always runs either in user mode or one of the 5 System Modes.

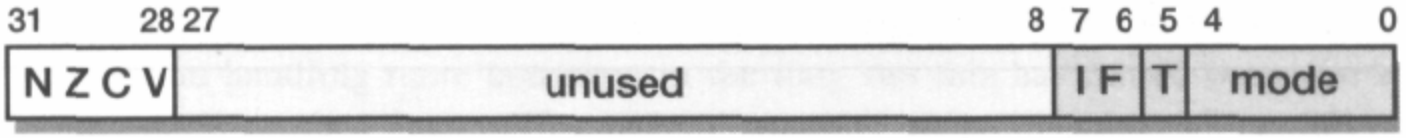
15 32-bit General Purpose Registers.



ARM ISA – Register Set

- All the registers are of length 32-bit.
- In the User Mode (Application Level Programs)
  - Registers r0 - r15 are available to the programmer.
  - r13 – sp (Stack Pointer)
  - r14 – lr (Link Register)
  - r15 – pc (Program Counter)
  - CPSR – Current Program Status Register is also accessible.
- Rest of the registers are used only in System-Level Programming and for handling Exceptions (like Interrupts)

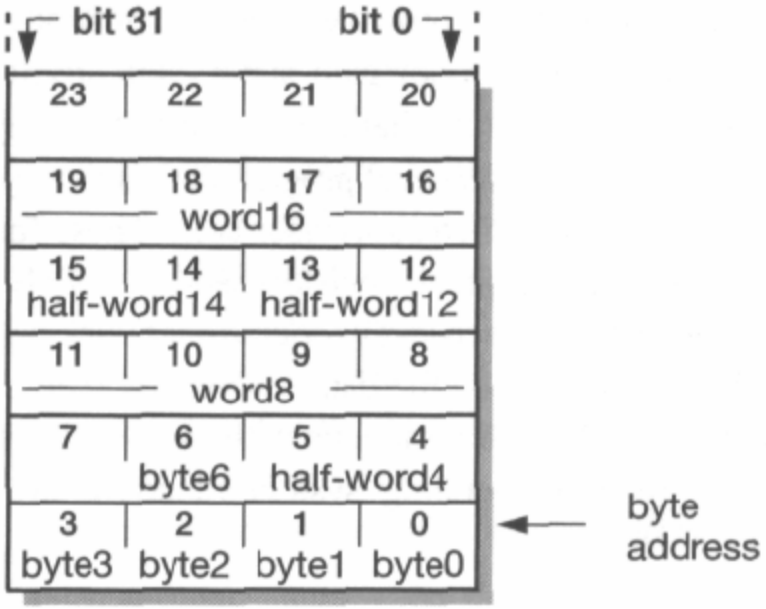
CPSR Register Format



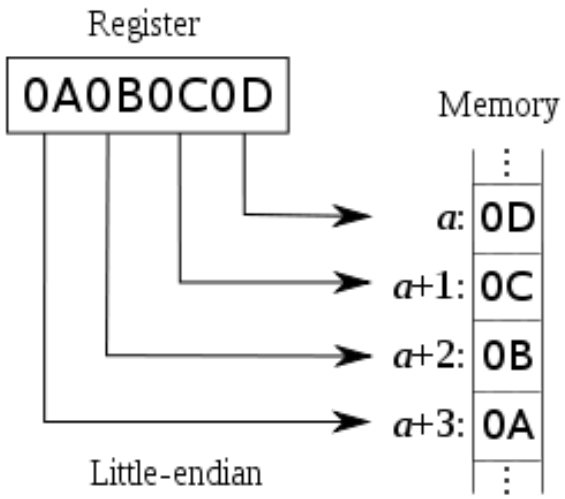
- N – Negative; the last ALU operation which changed the flags produced a negative result.
- Z – Zero
- C – Carry
- V – Overflow
- I and F – Interrupt enable flags (cannot be changed by programs running in User Mode)
- T – Thumb mode
- Mode – Mode bits indicates the Processor Mode.

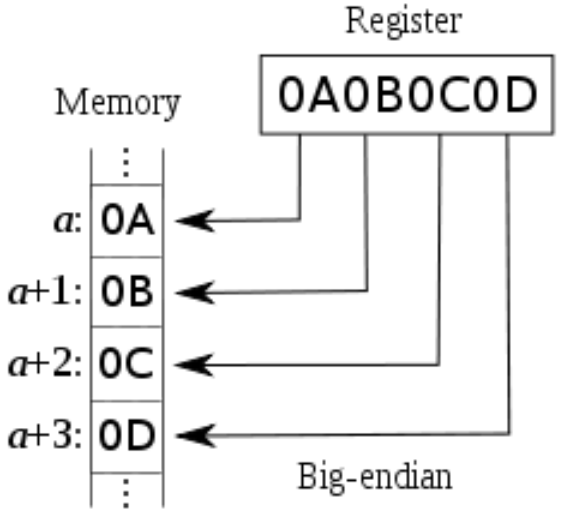
Memory System and Address Space

- Address Space Length: $2^{32} - 1$ bytes (4GB) .
- ARM Instruction can access
  - Byte sized data items
  - Half-word sized data items
  - Word-sized data items
- However, the half-words, words and also instructions should satisfy alignment restrictions.
- Follows Little-Endian Convention
- Can be configured to work with a Big-Endian Memory organization.



Big-Endian Versus Little-Endian

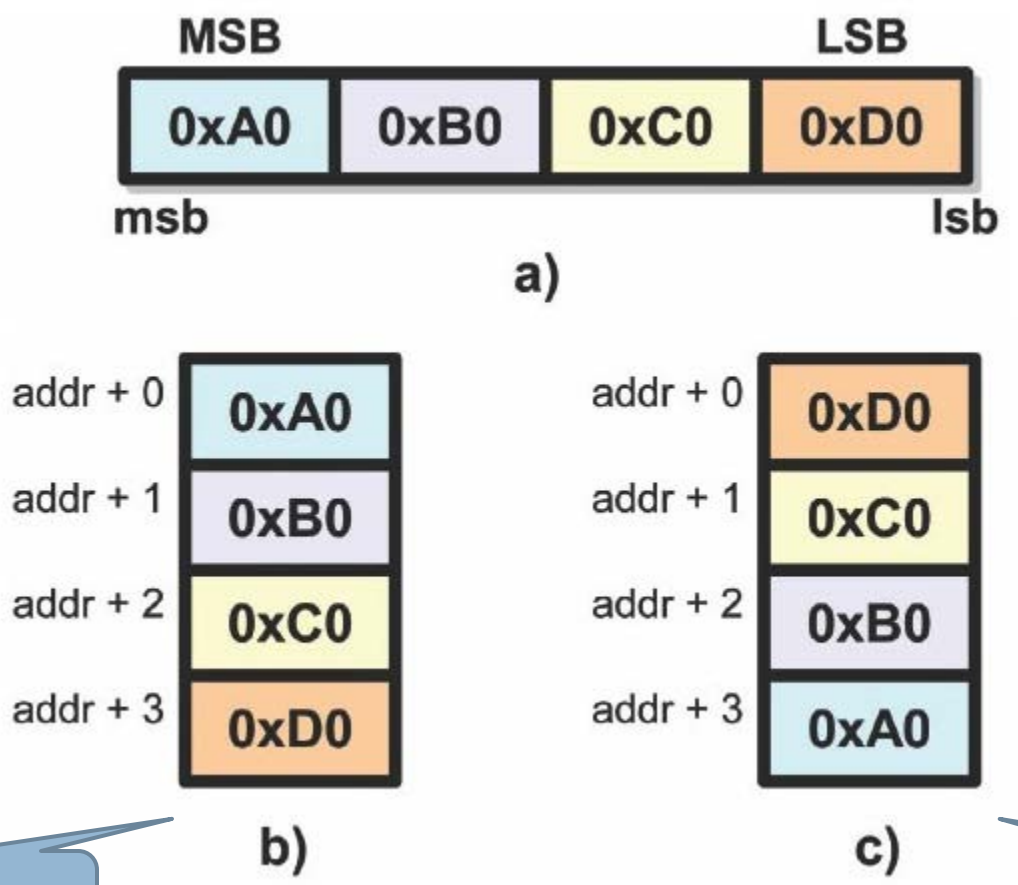




Little Endian: LSB goes to the smallest byte address.

Big Endian: MSB goes to the smallest byte address.

Big-Endian Versus Little-Endian



Big Endian

Little Endian

Big-Endian Versus Little-Endian

```
#include <stdio.h>
typedef unsigned char *byte_pointer;
void show_bytes(byte_pointer start, int len)
{
        int i;
        for (i = 0; i < len; i++)
                printf(" %.2x", start[i]);
        printf("\n");
}

main()
{
        int num = 0xF1F2F3F4;
        show_bytes((byte_pointer) &num, 4);
}
```

Is the output of this program same irrespective of whether it is run on a Little Endian machine or a Big Endian machine?

Load-Store Architecture

In Load-Store Architectures

- Instructions process (ADD, SUB, …) the data present only in the registers and result will also be placed in registers only.
- Only operations which apply to memory locations are ..
  - Load – Load from the memory location to a register.
  - Store – Store from the register to a memory location.

ARM Instruction Set

ARM Instructions can be classified into three categories.

1. Data Processing Instructions (like ADD, SUB, …)
2. Data Transfer Instructions (like LDR, STR, MOV, SWAP)
3. Control Flow Instructions

All Instructions are of 32-bit length.