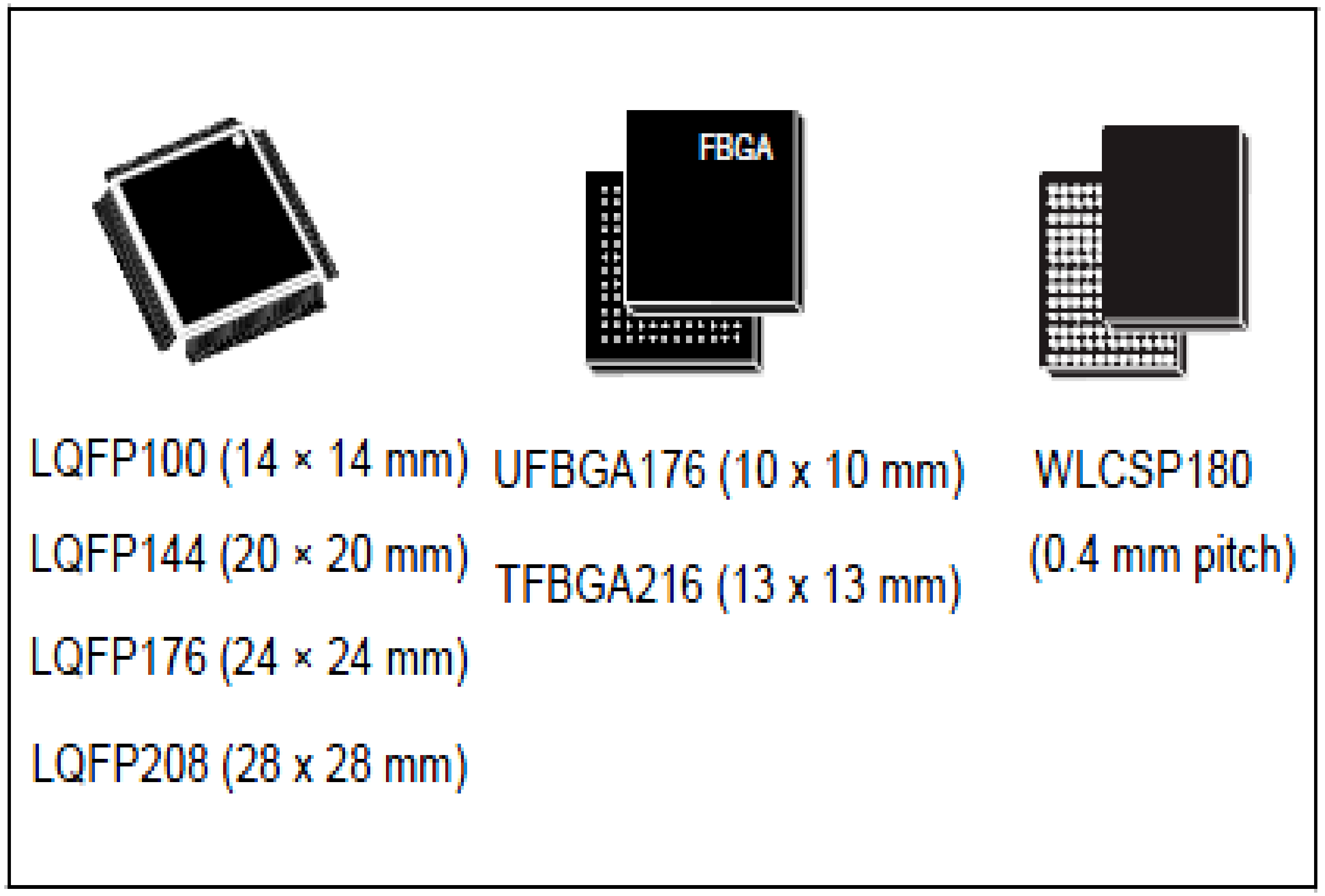
معرفی میکروکنترلر STM32F

ریزپردازنده ۱
محمد مهدی همایون پور
دانشکده مهندسی کامپیوتر، دانشگاه صنعتی امیرکبیر

مشخصات عمومی میکروکنترلر STM32F



کاربردهای میکروکنترلر STM32F

Suitable for a wide range of applications:

- Motor drive and application control
- Medical equipment
- Industrial applications: PLC, inverters, circuit breakers
- Printers, and scanners
- Alarm systems, video intercom, and HVAC
- Home audio appliances
- Mobile applications
- Internet of Things
- Wearable devices: smart watches

محصولات حاصل از میکروکنترلر ARM

مشخصات عمومی میکروکنترلر STM32F

- Core: ARM® 32-bit Cortex®-M7 CPU with
  - DPFPU, ART Accelerator™ and L1-cache.
  - L1-cache:
    - 16 Kbytes I/D cache,
    - allowing 0-wait state execution from embedded Flash and external memories
- DSP instructions

- The Chrom-Art Accelerator™ (DMA2D) is a graphic accelerator

DPFPU: Double-Precision Floating Point Unit
MPU: Memory Protection Unit which enhances the application security.
I/D cache: I (Instruction) cache and D (Data) Cache

مشخصات عمومی میکروکنترلر STM32F

The memory protection unit (MPU) is used to manage the CPU accesses to memory to prevent one task to accidentally corrupt the memory or resources used by any other active task.

مشخصات عمومی میکروکنترلر STM32F

- up to 216 MHz, MPU,
- 462 DMIPS/2.14 DMIPS/MHz (Dhrystone 2.1),
- DSP instructions

Dhrystone MIPS (Million Instructions per Second), or DMIPS, is a measure of computer performance relative to the performance of the DEC VAX 11/780 minicomputer of the 1970s.

STM32F مشخصات عمومی میکروکنترلر

Internal Memories:

- Up to 2 Mbytes of Flash memory organized into two banks allowing read-while-write
- SRAM: 512 Kbytes (including 128 Kbytes of data TCM RAM for critical real-time data)
- + 16 Kbytes of instruction TCM RAM (for critical real-time routines) + 4 Kbytes of backup SRAM
- Flexible external memory controller with up to 32-bit data bus: SRAM, PSRAM, SDRAM/LPSDR SDRAM, NOR/NAND memories
- Tightly Coupled Memory and cache memory are both fast access memories.

TCM: Tightly Coupled Memory interface
PSRAM: Pseudo-static random-access memory
SDRAM: Synchronous dynamic random-access memory

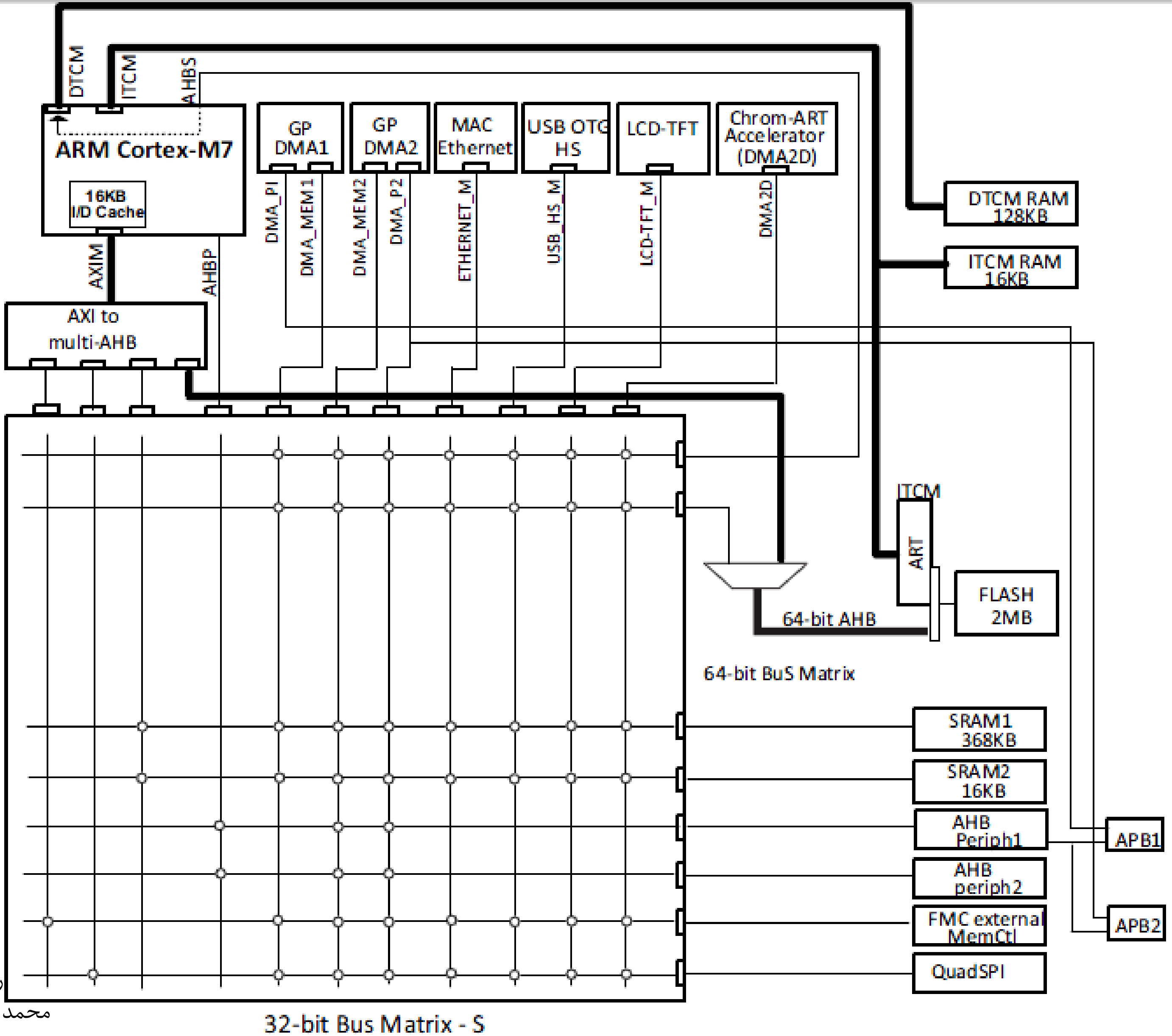
مشخصات عمومی میکروکنترلر STM32F

Extensive range of enhanced I/Os

- connected to two APB buses
- two AHB buses
- a 32-bit multi-AHB bus matrix
- a multi layer AXI interconnect supporting internal and external memories access.

- The ARM **Advanced Microcontroller Bus Architecture (AMBA)** is an open-standard, on-chip interconnect specification for the connection and management of functional blocks in system-on-a-chip (SoC) designs.

مشخصات عمومی میکروکنترلر STM32F



مشخصات عمومی میکروکنترلر STM32F

AXI, the third generation of AMBA interface defined in the AMBA 3 specification,

AXI is targeted at high performance, high clock frequency system designs and includes features that make it suitable for high speed sub-micrometer interconnect,

Advanced High-performance Bus (AHB) is a bus protocol introduced in Advanced Microcontroller Bus Architecture version 2 published by ARM Ltd company.

Advanced Peripheral Bus (APB) is designed for low bandwidth control accesses.

مشخصات عمومی میکروکنترلر STM32F

Peripherals:

- Three 12-bit ADCs
- 2 DACs
- 1 low-power RTC
- 12 general-purpose 16-bit timers including
  - two PWM timers for motor control
  - two general-purpose 32-bit timers
- 1 true random number generator (RNG),
- 1 cryptographic acceleration cell.

مشخصات عمومی میکروکنترلر STM32F

Standard and advanced communication interfaces:

- 4 I2Cs
- 6 SPIs
- 3 I2Ss in half-duplex mode
- 4 USARTs plus four UARTs
- 1USB OTG full-speed
- 1 USB OTG high-speed with full-speed capability

مشخصات عمومی میکروکنترلر STM32F

Standard and advanced communication interfaces:

- Three CANs
- Two SAI serial audio interfaces
- Two SDMMC host interfaces
- Ethernet interface
- Camera interface for CMOS sensors
- LCD-TFT display controller
- Chrom-ART Accelerator™
- SPDIFRX interface
- HDMI-CEC
- 1 flexible memory control (FMC) interface
- a Quad-SPI Flash memory interface
- A cryptographic acceleration cell
- –40 to +105 °C temperature range
- 1.7 to 3.6 V power supply

مشخصات عمومی میکروکنترلر STM32F

- USB OTG introduces the concept of a device performing both master and slave roles – whenever two USB devices are connected and one of them is a USB OTG device, they establish a communication link. The device controlling the link is called the master or host, while the other is called the slave or peripheral.

- For instance, a mobile phone may read from removable media as the host device, but present itself as a USB Mass Storage Device when connected to a host computer.

مشخصات عمومی میکروکنترلر STM32F

- **S/PDIF (Sony/Philips Digital Interface Format)** is a type of digital audio interconnect used in consumer audio equipment to output audio over reasonably short distances.

مشخصات عمومی میکروکنترلر STM32F

Consumer Electronics Control (CEC) is a feature of HDMI designed to allow users to command and control devices connected through HDMI[1][2] by using only one remote control.

For example, by using the remote control of a television set to control a set-top box and or DVD player.

Up to 15 devices can be controlled.

مشخصات عمومی میکروکنترلر STM32F

- LCD-TFT controller
  - The LCD-TFT display controller provides a 24-bit parallel digital RGB (Red, Green, Blue) and delivers all signals to interface directly to a broad range of LCD and TFT panels up to XGA (1024x768) resolution with the following features:
    - 2 display layers with dedicated FIFO (64x32-bit)
    - Color Look-Up table (CLUT) up to 256 colors (256x24-bit) per layer
    - Up to 8 input color formats selectable per layer
    - Flexible blending between two layers using alpha value (per pixel or constant)
    - Flexible programmable parameters for each layer
    - Color keying (transparency color)
    - Up to 4 programmable interrupt events

مشخصات عمومی میکروکنترلر STM32F

Flexible memory controller (FMC)

FMC includes three memory controllers:

- The NOR/PSRAM memory controller
- The NAND/memory controller
- The Synchronous DRAM

مشخصات عمومی میکروکنترلر STM32F

Quad-SPI memory interface (QUADSPI)

- is a specialized communication interface targeting Single, Dual or Quad-SPI Flash memories.

- It can work in:
    - Up to 256 Mbytes external Flash are memory mapped, supporting 8, 16 and 32-bit access. Code execution is supported.

مشخصات عمومی میکروکنترلر STM32F

DMA controller (DMA)

- The devices feature two general-purpose dual-port DMAs (DMA1 and DMA2) with 8 streams each.

- They are able to manage memory-to-memory, peripheral-to-memory and memory-to-peripheral transfers.

مشخصات عمومی میکروکنترلر STM32F

Embedded SRAM

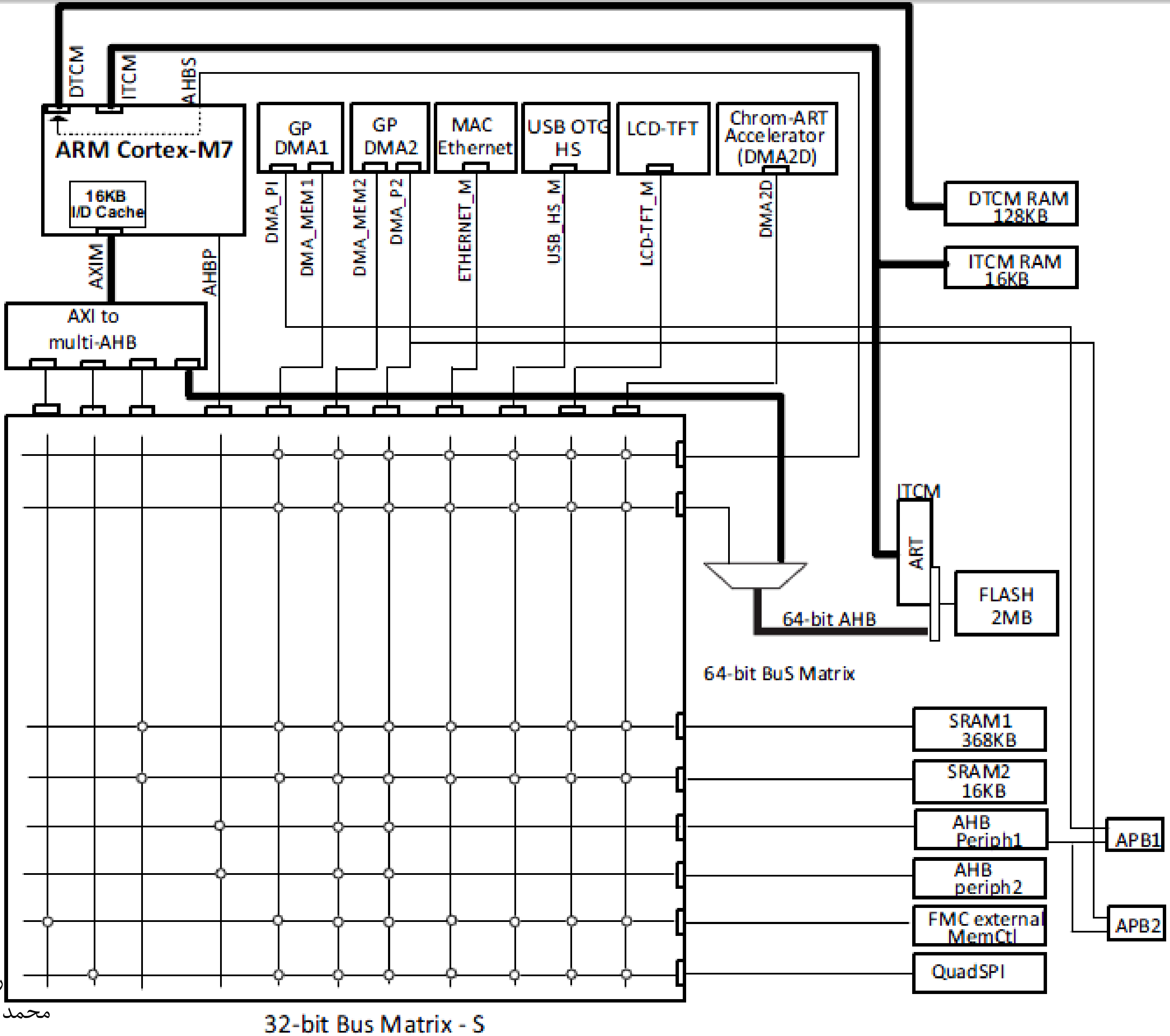
All the devices feature:
The Data TCM RAM is accessible by the GP-DMAs and peripherals DMAs through specific AHB slave of the CPU.The instruction TCM RAM is reserved only for CPU. It is accessed at CPU clock speed with 0 wait states. • System SRAM up to 512 Kbytes: – SRAM1 on AHB bus Matrix: 368 Kbytes – SRAM2 on AHB bus Matrix: 16 Kbytes – DTCM-RAM on TCM interface (Tighly Coupled Memory interface): 128 Kbytes for critical real-time data. • Instruction RAM (ITCM-RAM) 16 Kbytes: – It is mapped on TCM interface and reserved only for CPU Execution/Instruction useful for critical real-time routines.
• 4 Kbytes of backup SRAM This area is accessible only from the CPU. Its content is protected against possible unwanted write accesses, and is retained in Standby or VBAT mode

مشخصات عمومی میکروکنترلر STM32F

AXI-AHB bus matrix:
The STM32F system architecture is based on 2 sub-systems:

- An AXI to multi AHB bridge converting AXI4 protocol to AHB-Lite protocol:
  - 3x AXI to 32-bit AHB bridges connected to AHB bus matrix
  - 1x AXI to 64-bit AHB bridge connected to the embedded Flash memory
- A multi-AHB Bus-Matrix
  - The 32-bit multi-AHB bus matrix interconnects all the masters (CPU, DMAs, Ethernet, USB HS, LCD-TFT, and DMA2D) and the slaves (Flash memory, RAM, FMC, Quad-SPI, AHB and APB peripherals) and ensures an efficient operation even when several high-speed peripherals work simultaneously.

مشخصات عمومی میکروکنترلر STM32F



ARM Cortex-M7F

Key features of the Cortex-M7 core are:[6]

- 6-stage pipeline with branch speculation (پیش‌بینی انشعاب‌ها).
- Instruction sets:
  - Thumb-1 (entire).
  - Thumb-2 (entire).
  - 32-bit hardware integer multiply with 32-bit or 64-bit result, signed or unsigned, add or subtract after the multiply.
  - 32-bit hardware integer divide (2-12 cycles).
  - Saturation arithmetic support.
  - DSP extension: Single cycle 16/32-bit MAC, single cycle dual 16-bit MAC, 8/16-bit SIMD arithmetic.
- 1 to 240 interrupts, plus NMI.
- 12 cycle interrupt latency.
- Integrated sleep modes.

- MAC: Multiply–accumulate operation

- SIMD: Single instruction, multiple data, multiple processing elements that perform the same operation on multiple data points simultaneously.

ARM Cortex-M7F

In computer architecture, a **branch predictor** is a digital circuit that tries to guess which way a branch (e.g. an if-then-else structure) will go before this is known definitively.

Saturation arithmetic is a version of arithmetic in which all operations such as addition and multiplication are limited to a fixed range between a minimum and maximum value

For example, if the valid range of values is from -100 to 100, the following operations produce the following values:

- $60 + 30 = 90$
- $60 + 43 = 100$
- $(60 + 43) - (75 + 75) = 0$
- $10 \times 11 = 100$
- $99 \times 99 = 100$
- $30 \times (5 - 1) = 100$
- $30 \times 5 - 30 \times 1 = 70$

Data Sizes and Instruction Sets

- 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)
- Most ARM's implement two instruction sets
  - 32-bit ARM Instruction Set
  - 16-bit Thumb Instruction Set
- Jazelle cores can also execute Java bytecode

Java bytecode is the instruction set of the Java virtual machine (JVM).

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

Processor Modes

- The ARM has seven basic operating modes:
  - **System** : privileged mode using the same registers as user mode
  - **User** : unprivileged mode under which most tasks run
  - **FIQ** : entered when a high priority (fast) interrupt is raised
  - **IRQ** : entered when a low priority (normal) interrupt is raised
  - **Supervisor** : entered on reset and when a Software Interrupt instruction is executed
  - **Abort** : used to handle memory access violations
  - **Undef** : used to handle undefined instructions

Privileged Mode: The software can use all the instructions and has access to all resources.

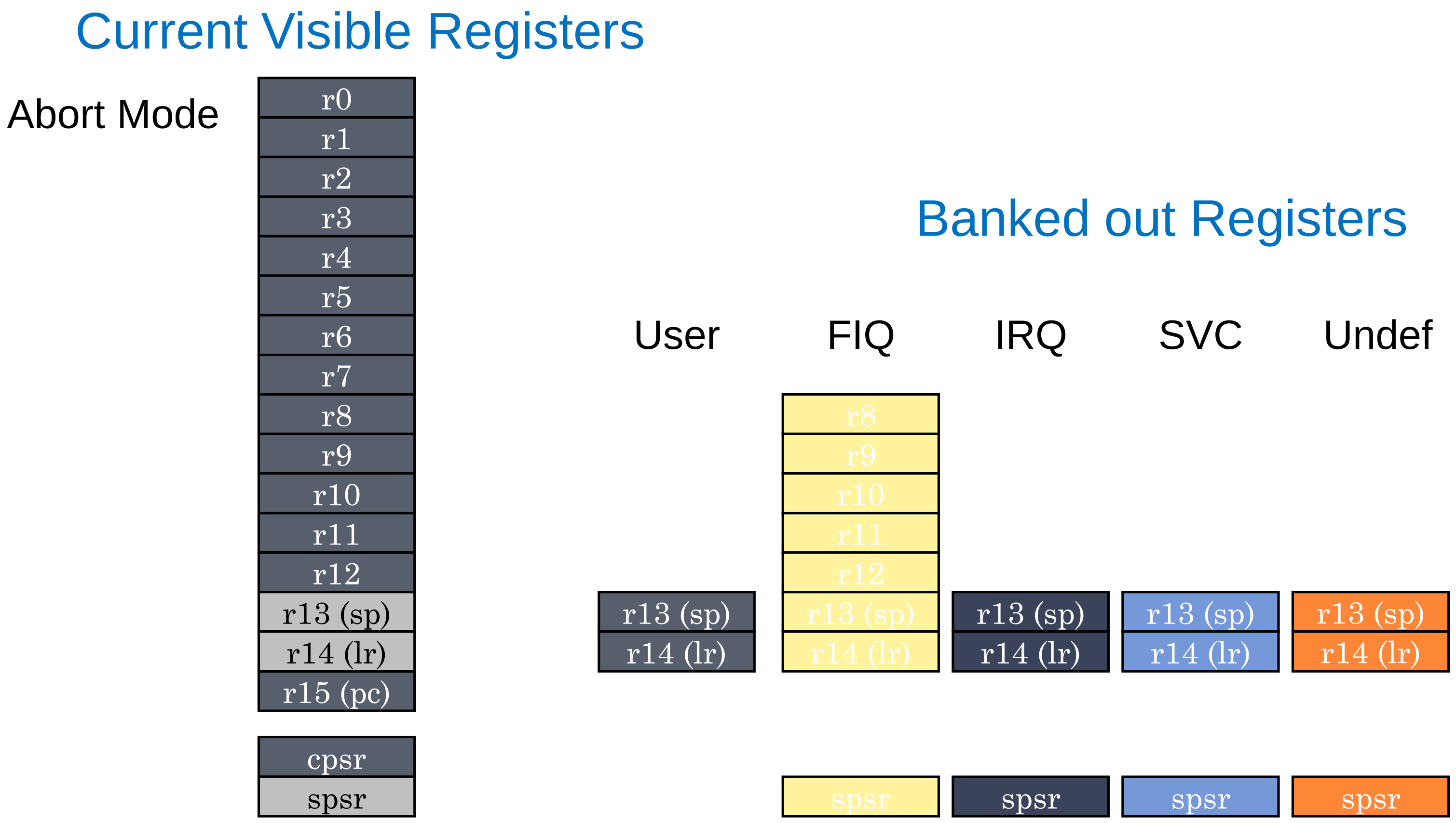
Exceptions: Bus faults, Usage Faults

The 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

- The current processor mode governs which of several banks is accessible. Each mode can access
  - a particular set of r0-r12 registers
  - a particular r13 (the stack pointer, sp) and r14 (the link register, lr)
  - the program counter, r15 (pc)
  - the current program status register, cpsr

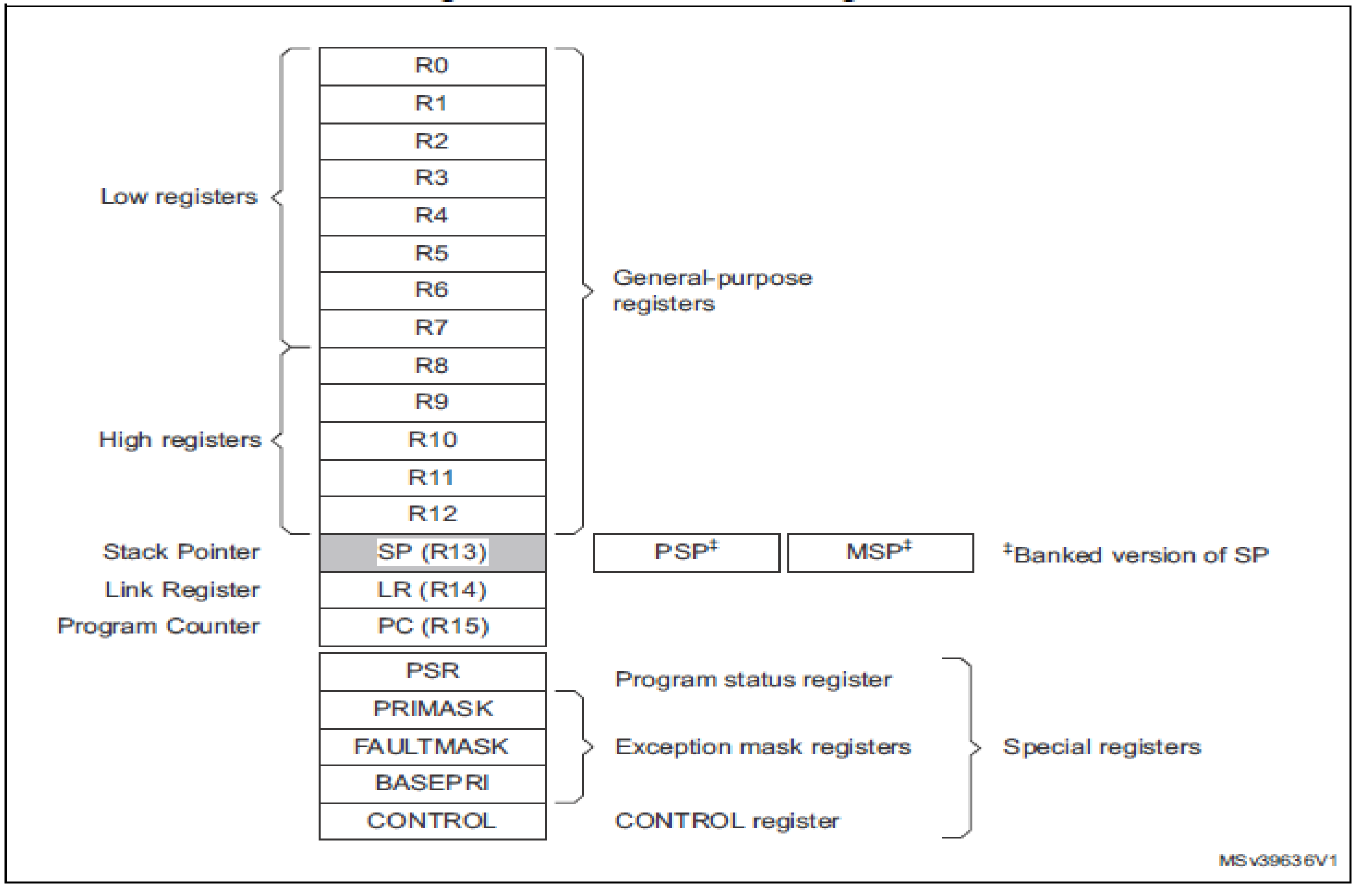
The ARM Register Set



LR: Link register
cpsr: Current Program Status Register
Spsr: Saved Program Status Register

Processor Core Registers

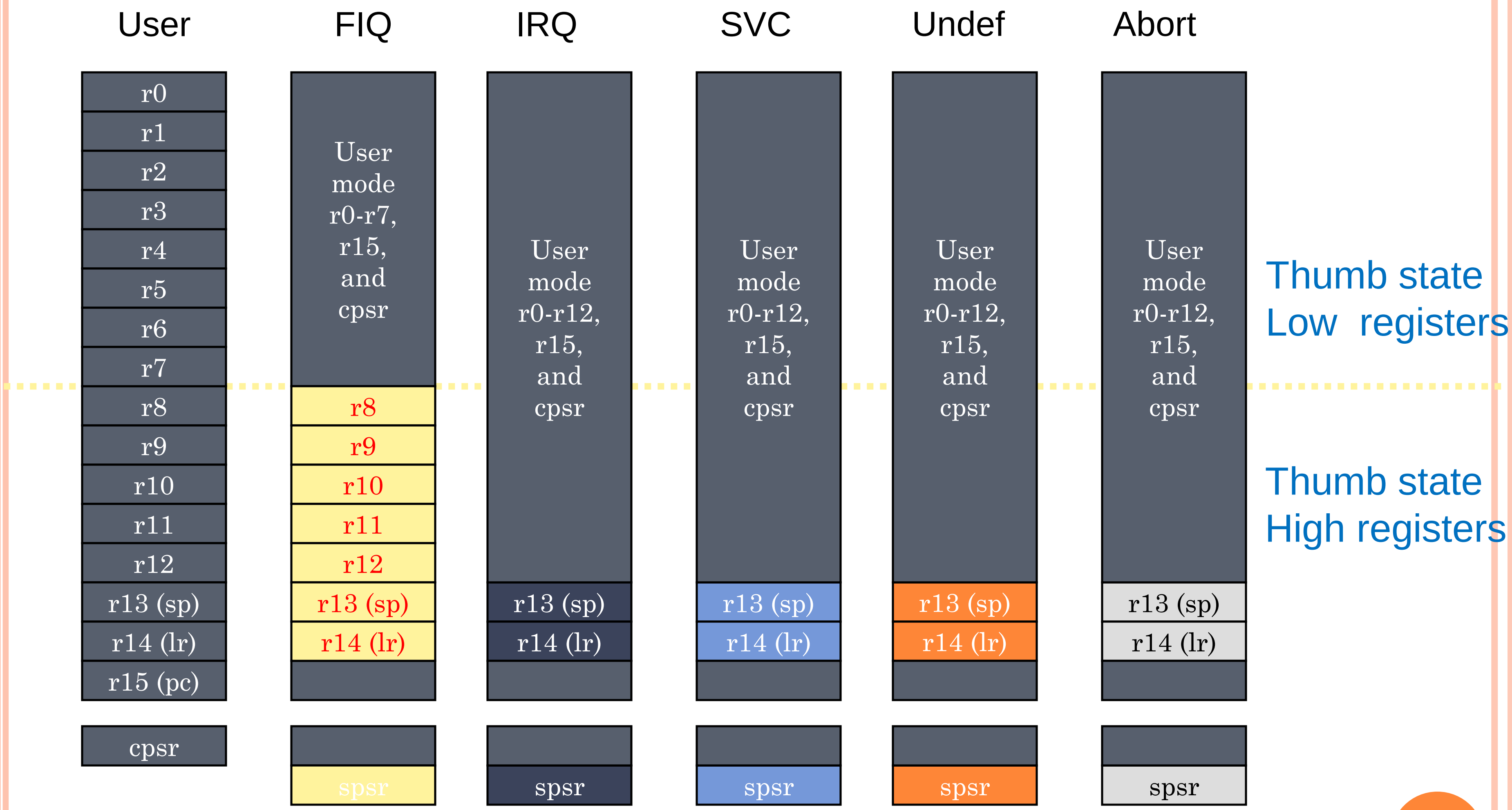
Figure 2. Processor core registers



MSP: Main Stack Pointer

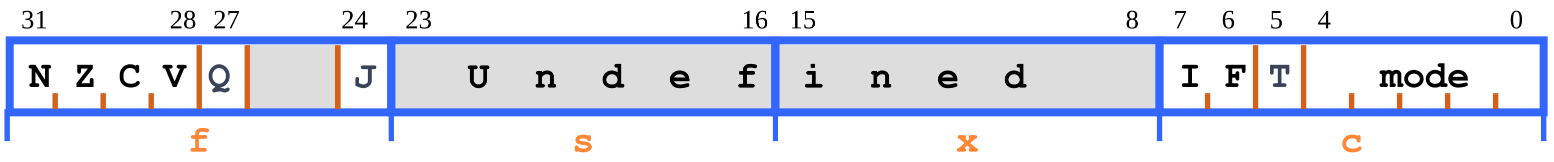
PSP: Process Stack Pointer

Register Organization Summary


User
FIQ
IRQ
SVC
Undef
Abort
r0
r1
r2
r3
r4
r5
r6
r7
r8
r9
r10
r11
r12
r13 (sp)
r14 (lr)
r15 (pc)
cpsr
User mode r0-r7, r15, and cpsr
r8
r9
r10
r11
r12
r13 (sp)
r14 (lr)
spsr
User mode r0-r12, r15, and cpsr
r13 (sp)
r14 (lr)
spsr
User mode r0-r12, r15, and cpsr
r13 (sp)
r14 (lr)
spsr
User mode r0-r12, r15, and cpsr
r13 (sp)
r14 (lr)
spsr
User mode r0-r12, r15, and cpsr
r13 (sp)
r14 (lr)
spsr
Thumb state Low registers
Thumb state High registers

Note: System mode uses the User mode register set

Program Status Registers

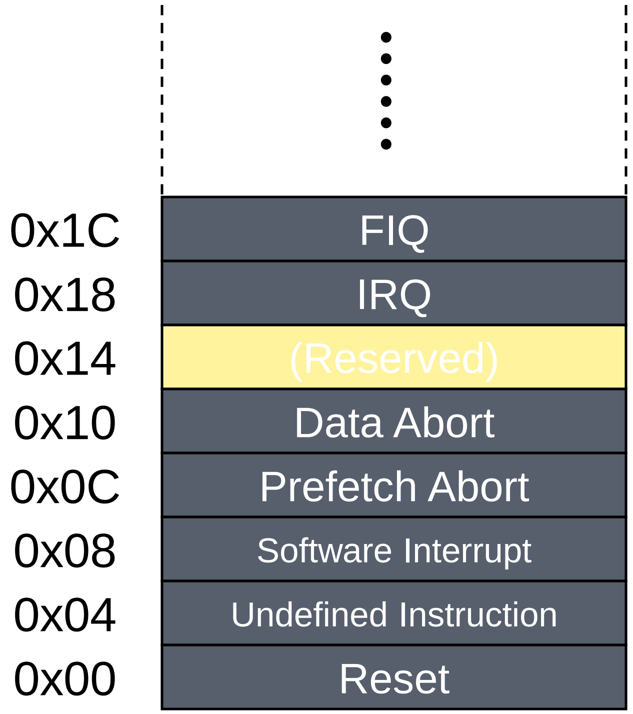


- Condition code flags
  - N = **N**egative result from ALU
  - Z = **Z**ero result from ALU
  - C = ALU operation **C**arried out
  - V = ALU operation o**V**erflowed
- Sticky Overflow flag - Q flag
  - Architecture 5TE/J only
  - Indicates if saturation has occurred
- J bit
  - Architecture 5TEJ only
  - 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

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

Development of the ARM Architecture

1

2

3

Early ARM architectures

4

Halfword and signed halfword / byte support

System mode

SA-110

SA-1110

4T

Thumb instruction set

ARM7TDMI

ARM720T

ARM9TDMI

ARM940T

5TE

Improved ARM/Thumb Interworking

CLZ

Saturated maths

DSP multiply-accumulate instructions

ARM1020E

XScale

ARM9E-S

ARM966E-S

5TEJ

Jazelle

Java bytecode execution

ARM9EJ-S

ARM926EJ-S

ARM7EJ-S

ARM1026EJ-S

6

SIMD Instructions

Multi-processing

V6 Memory architecture (VMSA)

Unaligned data support

ARM1136EJ-S

AGENDA

Introduction to ARM Ltd

Programmers Model

- Instruction Sets

System Design

Development Tools

Conditional Execution and Flags

- ARM instructions can be made to execute conditionally by postfixing them with the appropriate condition code field.
  - This improves code density *and* performance by reducing the number of forward branch instructions.

```
 CMP    r3,#0
  BEQ    skip
  ADD    r0,r1,r2
skip
```

```
CMP    r3,#0
ADDNE r0,r1,r2
```

- By default, data processing instructions do not affect the condition code flags but the flags can be optionally set by using “S”. CMP does not need “S”.

```
loop
  …
  SUBS r1,r1,#1
  BNE  loop
```

decrement r1 and set flags

if Z flag clear then branch

Condition Codes

- The possible condition codes are listed below:

Suffix	Description	Flags tested
EQ	Equal	Z=1
NE	Not equal	Z=0
CS/HS	Unsigned higher or same	C=1
CC/LO	Unsigned lower	C=0
MI	Minus	N=1
PL	Positive or Zero	N=0
VS	Overflow	V=1
VC	No overflow	V=0
HI	Unsigned higher	C=1 & Z=0
LS	Unsigned lower or same	C=0 or Z=1
GE	Greater or equal	N=V
LT	Less than	N!=V
GT	Greater than	Z=0 & N=V
LE	Less than or equal	Z=1 or N=!V
AL	Always	

Examples of conditional execution

- Use a sequence of several conditional instructions

```
if (a==0) func(1);
    CMP       r0,#0
    MOVEQ     r0,#1
    BLEQ      func
```

- Set the flags, then use various condition codes

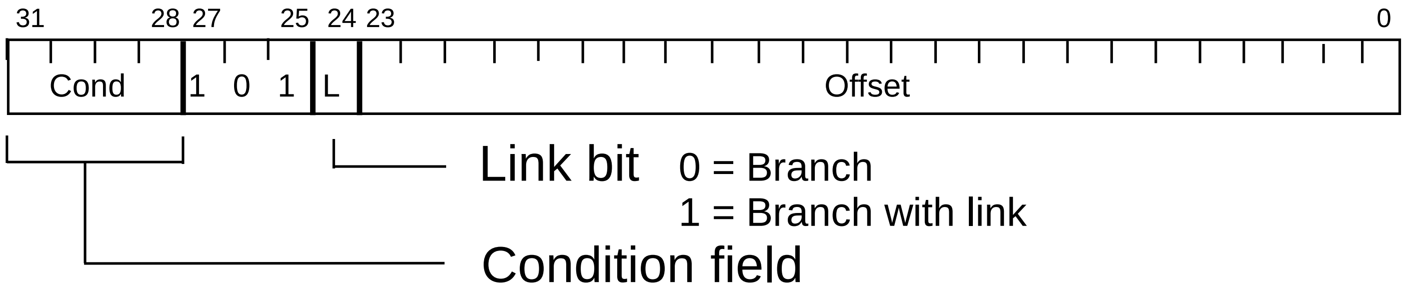
```
if (a==0) x=0;
if (a>0)  x=1;
    CMP       r0,#0
    MOVEQ     r1,#0
    MOVGT     r1,#1
```

- Use conditional compare instructions

```
if (a==4 || a==10) x=0;
    CMP       r0,#4
    CMPNE     r0,#10
    MOVEQ     r1,#0
```

Branch instructions

- Branch : `B{<cond>} label`
- Branch with Link : `BL{<cond>} subroutine_label`

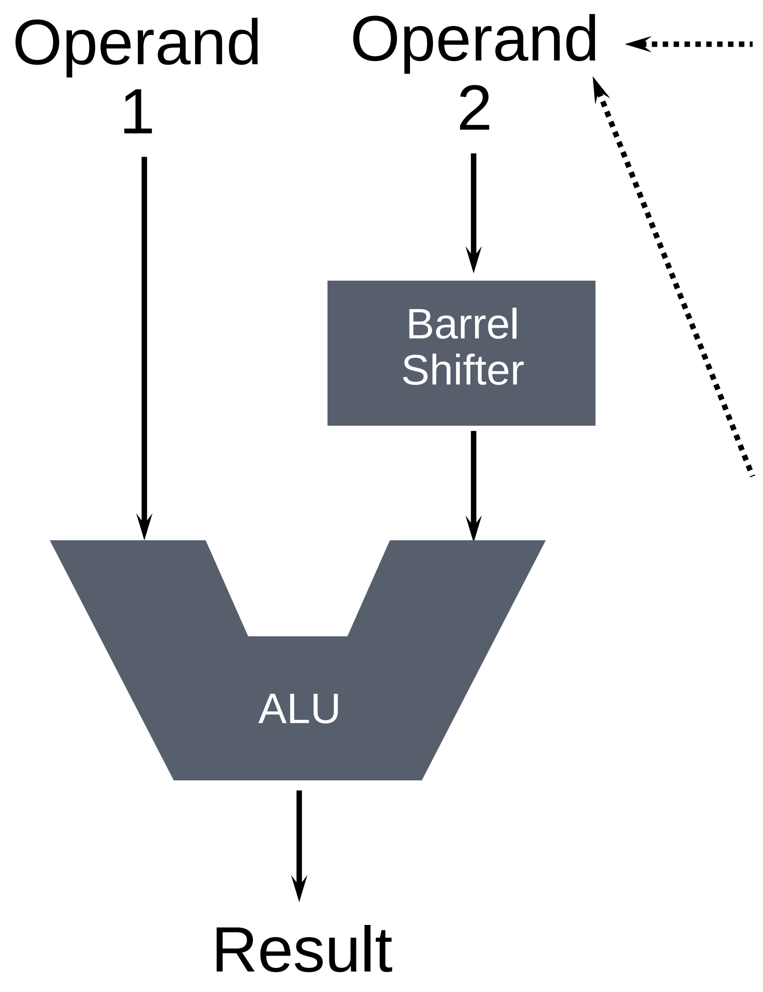


- The processor core shifts the offset field left by 2 positions, sign-extends it and adds it to the PC
  - ± 32 Mbyte range
  - How to perform longer branches?

Data processing Instructions

- Consist of :
  - Arithmetic: `ADD` `ADC` `SUB` `SBC` `RSB` `RSC`
  - Logical: `AND` `ORR` `EOR` `BIC`
  - Comparisons: `CMP` `CMN` `TST` `TEQ`
  - Data movement: `MOV` `MVN`
- These instructions only work on registers, NOT memory.

Using the Barrel Shifter: The Second Operand



Register, optionally with shift operation

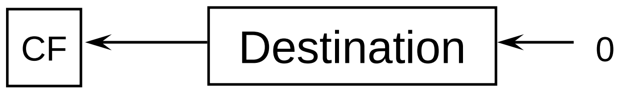
- Shift value can be either be:
  - 5 bit unsigned integer
  - Specified in bottom byte of another register.
- Used for multiplication by constant

Immediate value

- 8 bit number, with a range of 0-255.
  - Rotated right through even number of positions
- Allows increased range of 32-bit constants to be loaded directly into registers

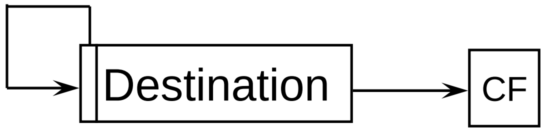
The Barrel Shifter

LSL : Logical Left Shift



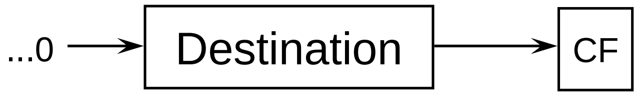
Multiplication by a power of 2

ASR: Arithmetic Right Shift



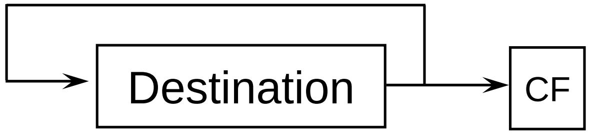
Division by a power of 2,
preserving the sign bit

LSR : Logical Shift Right



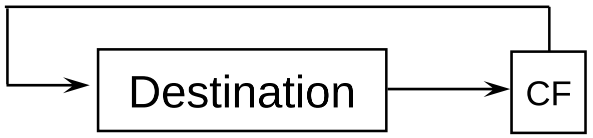
Division by a power of 2

ROR: Rotate Right



Bit rotate with wrap around
from LSB to MSB

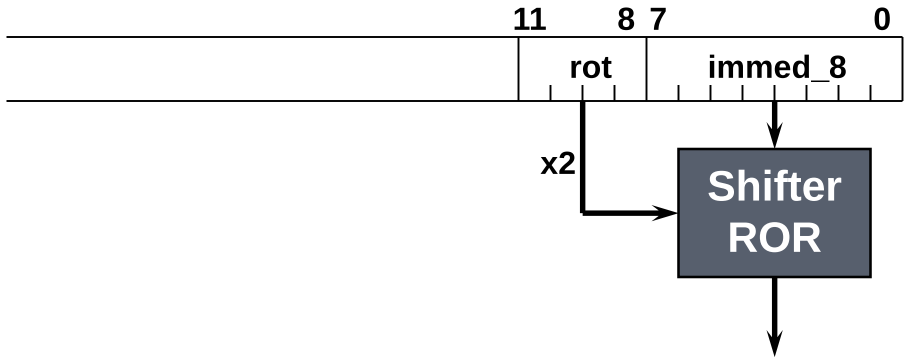
RRX: Rotate Right Extended



Single bit rotate with wrap around
from CF to MSB

IMMEDIATE CONSTANTS (1)

- No ARM instruction can contain a 32 bit immediate constant
  - All ARM instructions are fixed as 32 bits long
- The data processing instruction format has 12 bits available for operand2



Quick Quiz:

```
0xe3a004ff
MOV r0, #???
```

- 4 bit rotate value (0-15) is multiplied by two to give range 0-30 in steps of 2
- Rule to remember is "8-bits shifted by an even number of bit positions".