- Advanced RISC Machine
  - Reduced Instruction set Computer
- CISC
  - Intel AMD
  - Complex Instruction set computer

- Intel: 400~ instrucciones
- ARM: 30~

RISC { 1 instrucción / 1 ciclo

CISC { 1 instrucción / 20 ciclos

CISC -> Hace muchas operaciones complejas
-> Cosas complicadas
-> CPU más costoso
-> solo +, -

* Ciclo de reloj: Como la CPU mide el tiempo
* Ideal: Que la instrucción se haga en 1 ciclo de reloj.
* Problemas: CPU más rapida que memoria
  - viceversa -> Genera cuello de botella. } En Berkley investigación como hacer operaciones más ligeras.

* First RISC microprocessor for commercial use
* Market-leader for low-power and cost-sensitive embedded applications } Se calienta menos, más baterías.
* Se encuentran en casi todo el mundo, mayoría de celulares usan ARM.
* Apple: 25% de la empresa

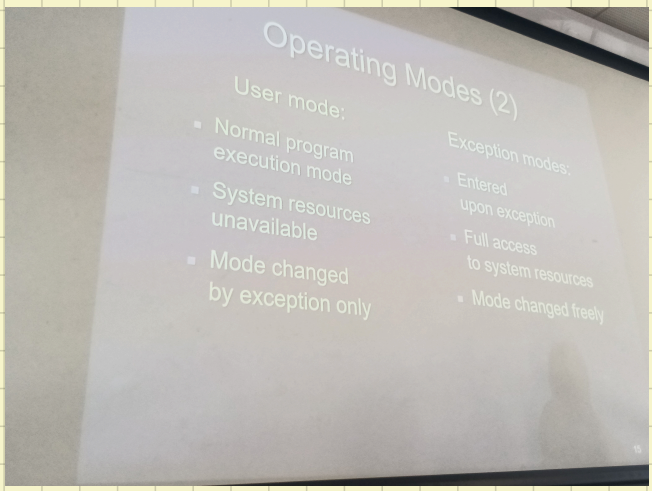
- Features:
  - Arquitectural simplicity

- The history of ARM
  - Developed at Acorn Computers Limited
  - Solution – the Berkeley RISC I:
    * Competitive
    * Easy to develop (< 1 year)
    * Cheap
    * Pointing the way to the future

- Typical RISC architectures
  - Large uniform register file
  - Load/Store architectures
  - Simple addressing modes
  - Uniform and fixed-length instruction fields

- Results:
  - High performance
  - Low code size
  - Low power consumption
  - Low silicon area

- Operating Modes:
  - User
  - Privileged:
    * System (V4 +)
    * FIQ
    * IRQ
    * Abort
    * undefined
    * Supervisor
    } Exceptions



- ARM Registers:
  - 31 general-purpose 32-bit registers
  - 16 visible, R0 – R15
  - Others speed up the cpu

  - Special roles:
    * R14 – Link Register (LR): } Salvar en memoria si cambiamos algo
      - optionally holds return address for branch instructions
    * R15 – Program Counter (PC)

  - Software:
    * R13 – Stack Pointer (SP) } Salvar en memoria si cambiamos algo

- Current Program Status Register (CPSR)

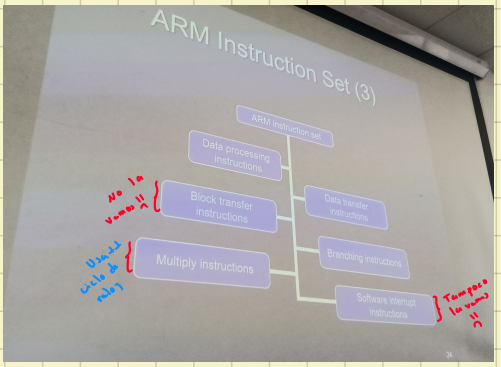
Marie: SKipcond → Revisa condición / Salta si se cumple la condición

RISC: Revisa condición ① → CPSR
Saltar ②

- Saved Program Status Register (SPSR) → No lo necesitamos

- Instruction Set
  - ARM
    - Standard 32-bit instruction set
  - THUMB
    - 16-bit compressed form
    - Code density better than most CISC
    - Dynamic compression in pipeline

- Features:
  - Load/Store architecture
  - 3-address data processing instructions

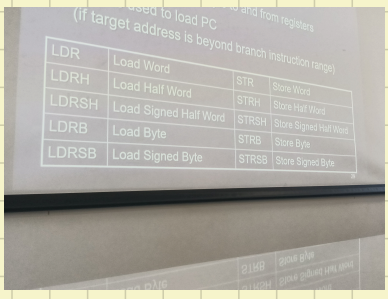


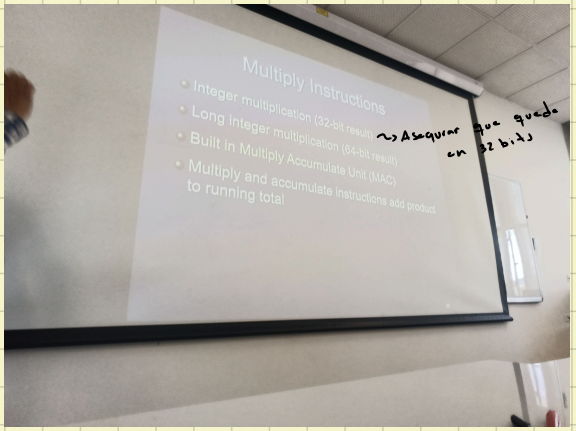
- Data Processing Instructions
  - Arithmetic and logical operations
  - 3-address format:
    - Two 32-bit operands (OP1 is register, OP2
    -
  - Arithmetic operations:
    - ADD, ADDC, SUB, SUBC, RSB, RSC
  - Bit-wise logical operations:
    - AND, EOR, ORR, BIC
  - Register movement operations:
    - MOV, MVN
  - Comparison operations:
    - TST, TEQ, CMP, CMN

Marie	ARM
Add	ADD
Subt	SUB
	MOV
	CMP

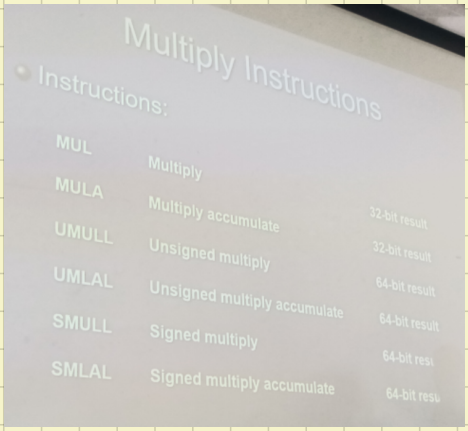
Marie	ARM
Load	LDR
Store	STR
	MUL

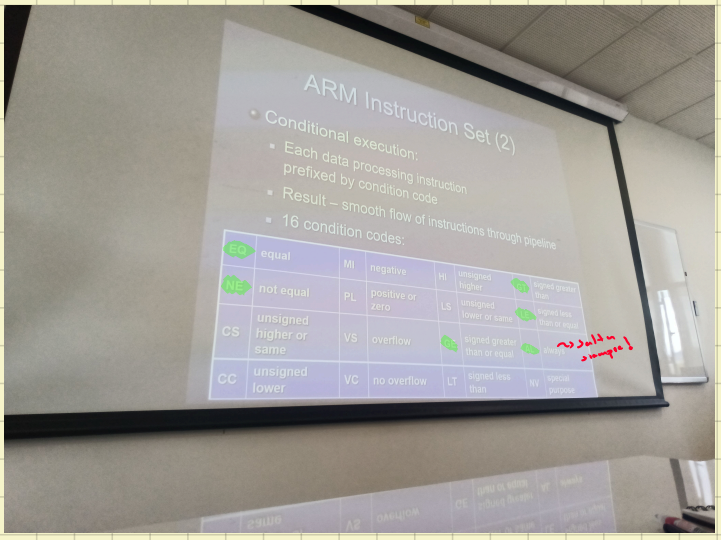
- Data Transfer instructions
  - Load/Store instructions
  - Used to move signed and unsigned Word, Half Word and byte to and from registers





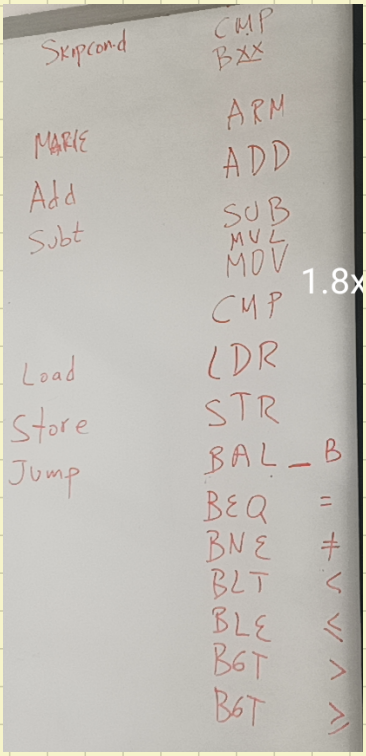
→ Asegurar que quede en 32 bits





→ Saltan siempre

- Branch siempre mira la condición



1.8x

Jn S
Jump I

BL
BX

- Branching Instructions:
  - Branch (B):
    - jumps forwards/backwards up to 32 Mb.
  - Branch link (BL)
    - Same + saves (PC + 4) in LR
  - Branch exchange (BX) and
    - Branch link exchange (BLX):
      - Same as B/BL +
      - change instruction set (ARM ↔ THUMB)
  - Only way to swap sets

Combin Ring por BLS