

Arquitetura do conjunto de instruções (ISA)

Sistemas de Computação

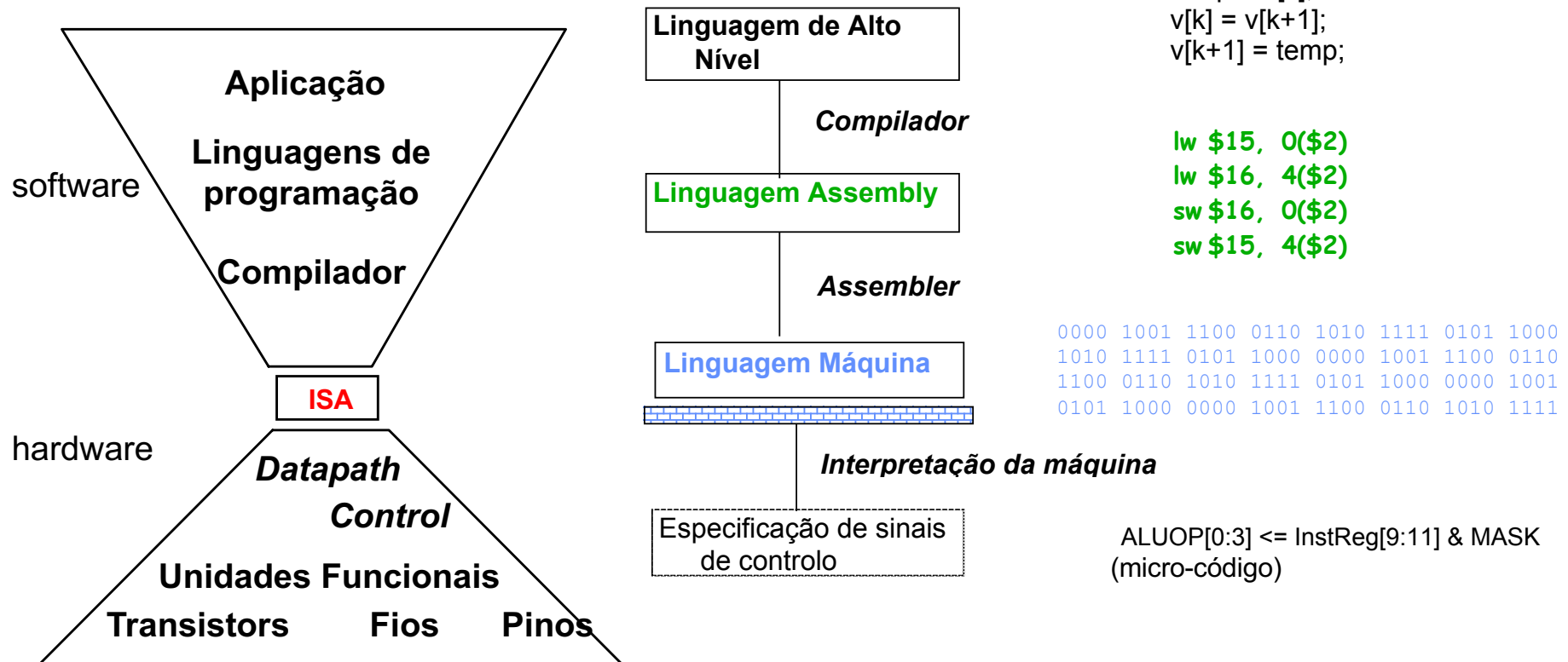
Lic. em Ciências da Computação

João Luís Sobral

Mar/2024

Instruction Set Architecture (ISA)

Parte do processador visível ao programador/compilador



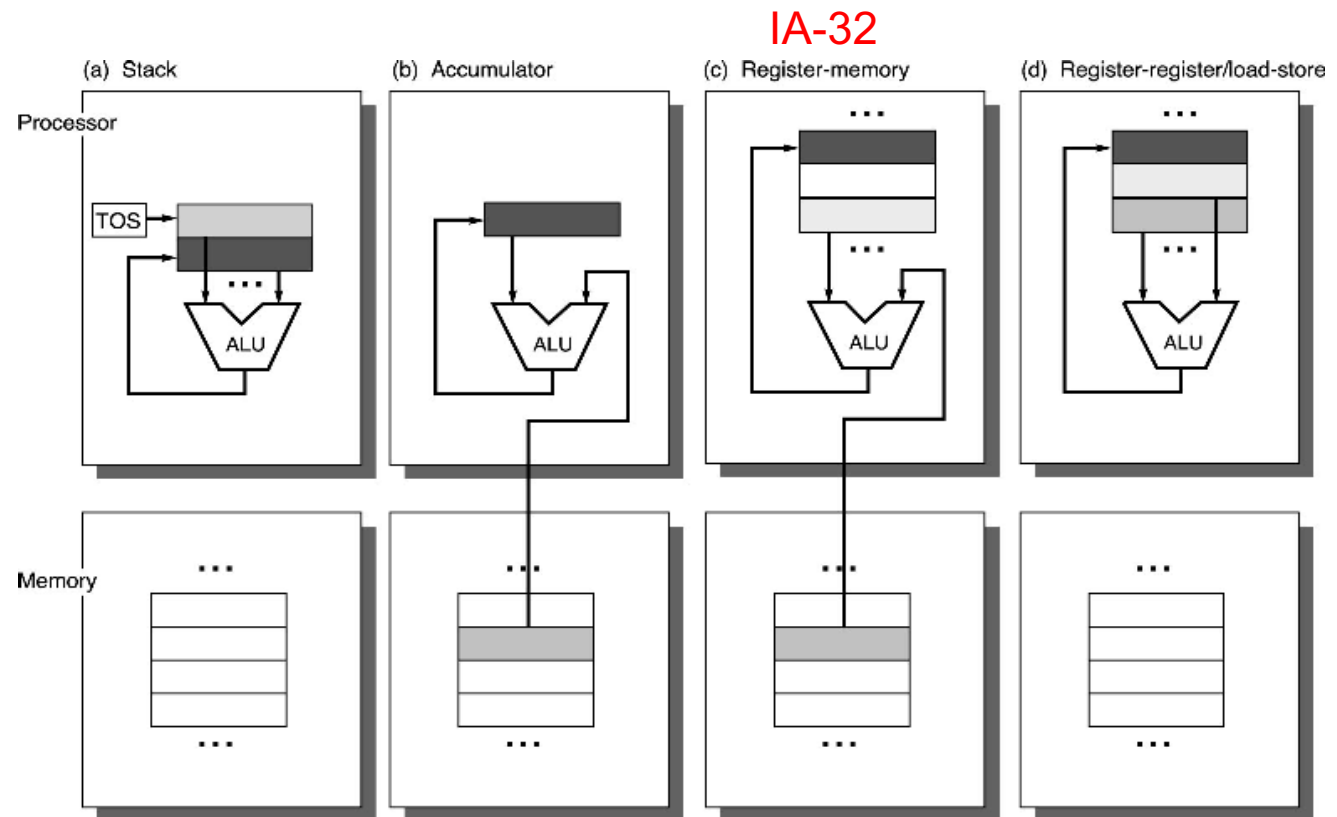
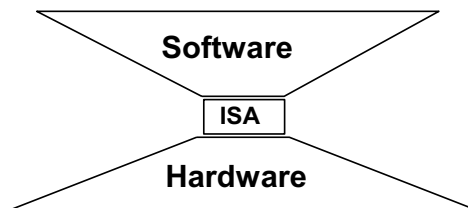
-Serve de Interface entre o software e o hardware

- Um conjunto de instruções possibilita a separação entre as ferramentas de desenvolvimento (compiladores) e a implementação desse conjunto de instruções (processadores)

- Na disciplina vai ser estudado em detalhe o ISA do IA-32

Instruction Set Architecture (ISA)

Tipos de ISA



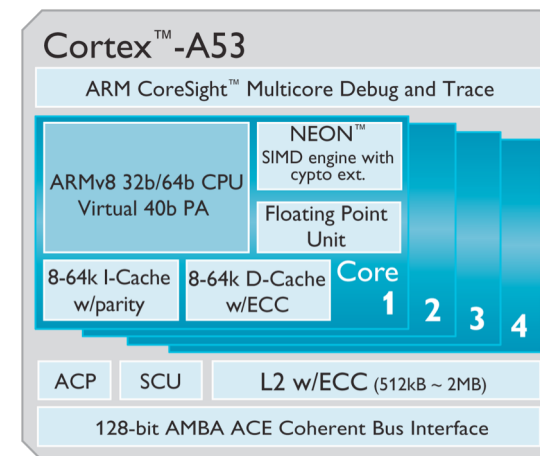
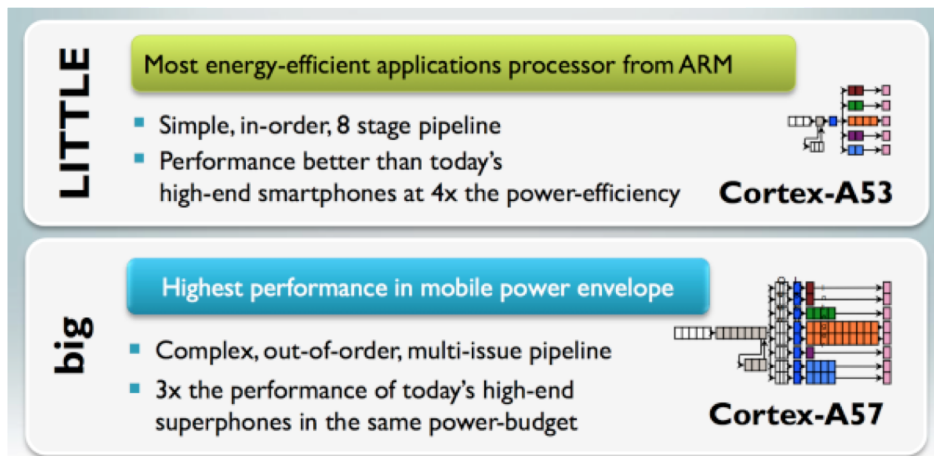
$$C = A + B$$

Stack	Acumulador	Reg-Mem	Load-store
Push A	Load A	Load A, R1	Load A, R1
Push B	Add B	Add B, R1	Load B, R2
Add	Store C	Store R1, C	Add R1, R2, R3
Pop C			Store R3, A

Instruction Set Architecture (ISA)

Exemplo de um ISA RISC: ARMv8-A

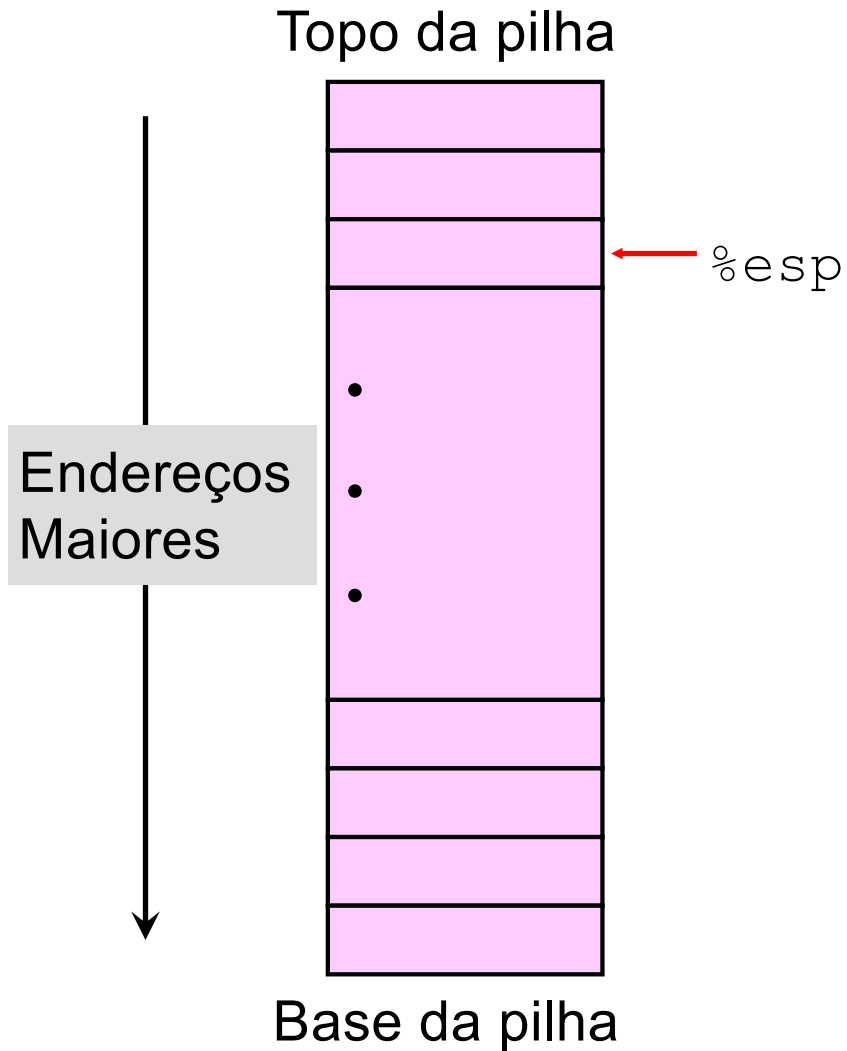
- Versão 64bit do ISA da ARM (Out/2011, sucessor do ARMv7 32bits)
 - 31 registos genéricos de 64 bits (+PC e SP)
 - Formato de instruções de 32 bits / arquitetura tipo *Load/Store*
 - Instruções para suporte a operandos de 64 bits
 - Endereços com 64 bits
- Implementações do ISA:



Localização dos operandos

- Imediato:
 - constante especificada na própria instrução (campo *Imm*)
- Registo:
 - registo(s) a utilizar especificados na instrução nos campos *rA*, *rB*, *etc*
- Memória principal
 - Base+Deslocamento:
 - Endereço da posição de memória a ler/escrever especificado como a soma do valor imediato (*Imm*) com o conteúdo de um (ou vários) registo(s) (*rX*)
 - Caso especial: Pilha

Pilha



- Região da memória com dados do programa
- Usada para suportar chamadas a funções e variáveis locais a funções
- Topo da pilha (“stack-pointer”) : `%esp`
 - Indica o endereço da posição ocupada
- Pilha cresce para endereços menores
 - Topo no endereço menor
 - `push` – coloca um valor na pilha - primeiro subtrai 4 ao `%esp`
 - `pop` – retira um valor da pilha - adiciona 4 ao `%esp` após leitura do valor