

3.1

(1) 操作数在指令中, ~~有效地址, 即数~~

(2)  $EA = D = 7237H$

(3) 无有效地址, BX中为操作数

(4)  $EA = (BX) = 637DH$

(5)  $EA = (BX) + D = \cancel{637D}$ :

$$= 637DH + 7237H$$

$$= D5B4H$$

(6)  $EA = (BX) + (SI) = 8E18H$

(7)  $EA = (BX) + (SI) + D = 1004FH$

3.2

(1)  $ADD\ DX, BX$

(2)  $ADD\ AL, [BX], [SI]$

(3)  $ADD\ [BX + 0B2H], CX$

(4)  $ADD\ [0524H], 2A59H$

(5)  ~~$ADD\ [AL]$~~   $ADD\ AL, 0B5H$

3.3 (1)  $MOV\ DX, BLOCK$

$ADD\ BX, 12$

$MOV\ DX, [BX]$

(2) MOV ~~DBX~~, BLOCK  
MOV DX, [BX+12]

(3) MOV BX, BLOCK.  
MOV SI, 12  
MOV DX, [BX+SI]

3.4

(1) (AX) = 1200H

(2) (AX) = 0100H

(3) (AX) = 4C2AH

(4) (AX) = 3412H

(5) (AX) = 4C2AH

(6) (AX) = 7856H

(7) (AX) = 65B7H

## 题目 1

RISC-V 处理器的寻址方式: 立即寻址, 寄存器寻址, 基址寻址, 偏移量寻址, 绝对寻址, PC 相对寻址。

龙芯处理器的寻址方式: 立即寻址, 寄存器寻址, 基址寻址, 偏移量寻址, 绝对寻址, PC 相对寻址。

## 题目 2

表 1 MOD 为 00, 表示直接寻址。此时, [BP] 是一个有效的寄存器寻址, 但是不允许使用偏移量。

表 2 MOD 为 01 或 10, [BP] 可以与偏移量结合使用。这种情况下, BP 通常被用作栈帧的基址, 可以方便地访问栈中的局部变量和参数。

设计 R/M 表的意图为通过将 BP 用于栈操作, Intel 设计允许程序员轻松访问栈帧中的数据, 简化了局部变量和参数的管理, 减少了计算寻址的复杂性, 提高了执行效率。

## 题目 3

RISC 处理器使用较少的简单指令, 允许更快的指令解码和执行。相比之下, CISC 处理器的复杂指令集需要更复杂的解码逻辑, 增加了设计和实现的复杂性。RISC 处理器的指令通常是固定长度且简单的, 这使得流水线设计更为高效。而 CISC 处理器的复杂指令可能需要多个周期才能完成, 限制了流水线的效率。而且 RISC 处理器通常在功耗管理方面表现更好, 适合现代移动和嵌入式设备的需求。