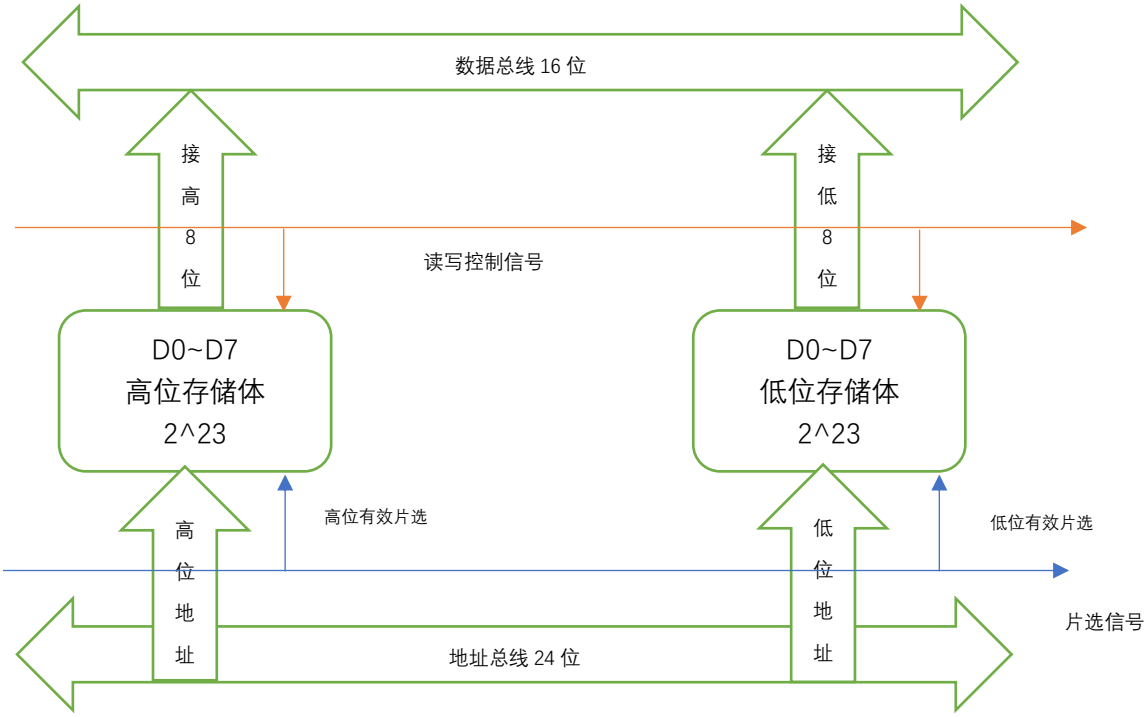
微嵌第二章作业参考答案

2.6

地址空间 = 2^24 = 16 777 216

示意图:



2.10

微操作码+执行顺序控制位=微指令。微程序是由多条微指令有序排列而成的。控制 ROM 用来存放每条指令对应的微程序。

2.13

1. 把第二条指令的地址 0x20000004 装入程序计数器 PC，程序计数器的内容 0x20000004 送到地址形成部件，地址形成部件产生的地址信号经地址缓冲器/驱动器和地址总线，被送到地址译码器进行译码，寻址指令存放的内存单元。

2. 操作控制器发读信号，将 0x20000004 单元的内容“LDR R1, [R3]”读出，由于是取指操作，“LDR R1, [R3]”经过数据总线被存入到指令寄存器 IR。

3. 如果程序计数器的单位是字节，则 PC 自动加 4，指向下一条指令的存放地址。

4. 指令译码器 ID 对指令操作码进行译码，操作控制器 OC 按照操作时序发出相应的控制信号。

5. 指令的地址码部分对应着汇编指令的操作数部分。本条指令中，存放源操作数的内存地址位于 R3 寄存器中，目的操作数是 R1 寄存器。

6. 在操作控制器输出的控制信号作用下， R3 寄存器的内容经地址形成部件和地址驱动器送到地址总线，再经地址译码后寻址到源操作数存放的内存单元.

7. 操作控制器发出读信号，将源操作数读出到数据总线，然后加载到 R1 寄存器。

2.14

不能。存储器单元之间不能直接传送数据，必须线将源操作数从存储器中读出到某个寄存器暂存，再将暂存的内容写入目的存储单元。（对应可以了解下 DMA，直接存储器访问）

2.15

WAR 为“读后写”，假设 Ij 为 SubR1，R2，R3，Ij+1 为 MoveR2，R4，若在 sub 指令读取 R2 之前 move 就将 R4 中数据转移到了 R2，则会发生“WAR”。

WAW 为“写后写”，假设 Ij 为 AddR1，R2，Ij+1 为 MoveR1，R3，若 move 先将 R3 中数据转移到了 R1，则最终 R1 存放的是 R2+R1 的结果，而程序本意是保留 R3 转移到 R1 的数据。

2.16

（1）转移目标指令：转移指令的目标指令，即下一条紧接着执行的命令。

（2）转移代价：假设在指令序列中，指令 Ij 是一条无条件转移指令，其在流水线上的执行步骤为：取指、译码、计算转移地址并更新程序计数器 PC。第三个周期阶数之后，第四个周期将读取转移目标指令 Ik。但在此之前，指令 Ij+1 和 Ij+2 也先后进入了流水线，转移必须将这两条指令丢弃。上述过程产生的两个流水线周期延迟被称为转移代价。

（3）转移延迟槽：转移指令 Ij 后面的一个时间片。

（4）BTB：转移目标缓冲器。BTB 收集和存储了近期所有转移指令的有关信息，并按照查找表的形式组织，为动态转移预测提供信息。

2.18

在发射前，必须将待处理的指令进行挑选，只有不相干的指令才能被发射到不同的流水线上。若流水线 2 的指令要用到流水线 1 的结果，那便会产生窝工。

2.20

同构多核处理器的内核普遍采用通用处理器，每个处理器的结构相同，地位相等。异构多核处理器具有不同功能和性能的内核以匹配实际应用需求，在提升芯片整体性能的同时，优化处理器结构，降低系统功耗。例如，对于通用的个人计算机，可以将图形处理器 GPU 与通用 CPU 集成在一颗芯片上，从而构成一种异构多核处理器。在这样的架构下，程序中必须串行执行的部分由 CPU 执行，可以并行的处理任务交由 GPU 内核进行提速。

2.24

300MHz×1.1MPIS/MHz=330MIPS

2.25

不够客观。原因有二：

1. 考虑到 RISC 与 CISC 架构处理器指令集繁简程度不同，使用 MIPS 衡量二者间性能差异不妥。

2. 对于同一处理器，MIPS 还依赖于所运行的测试程序。