

## 2017年半期考试

1. 在扩展的同步控制中, ④。

- ① 无时钟周期划分
- ② 无总线周期划分
- ③ 允许时钟周期长度可变
- ④ 允许总线周期长度可变

2. 操作  $M \rightarrow \text{MDR} \rightarrow C$  与  $C \rightarrow \text{MDR} \rightarrow M$  是否都可以在一个时钟内完成? ( ③ )

- ① 两者都不能
- ② 两者都能
- ③ 前者能, 后者不能
- ④ 后者能, 前者不能

3. 汇编程序的功能是 ④。

- ① 将汇编语言程序翻译成微程序
- ② 将高级语言程序翻译成机器语言程序
- ③ 将机器语言程序翻译成微程序
- ④ 将汇编语言程序翻译成机器语言程序

4. CPU 从数据总线上获取的信息, 依据 ④ 来判断该信息是指令还是数据。

- ① 对该信息进行译码的结果
- ② 指令和数据来自内存的不同的地址
- ③ 寻址方式的不同
- ④ 获取该信息的时间

5. 在补码一位乘法中, 有效位连同符号位共  $N+1$  位, 因此需要做 ① 步操作。

- ①  $N$  或者  $N+1$
- ②  $N$
- ③  $N+2$
- ④  $N+1$

6. CPU 的程序状态字(或标志位寄存器), 其状态是根据 ① 来进行设置。

- ① 指令执行结果
- ② 指令的寻址方式
- ③ 指令的格式
- ④ 指令的类型

7. 若某浮点数为  $2^3 \times 1.10101$ , 其尾数为补码表示, 则该数 ②。

- ① 已是规格化数
- ② 需将尾数左移一位规格化

③ 需右移规格化

④ 需将尾数左移两位规格化

8. 如果一条微指令的命令字段共 18 位, 分为 6 个字段(分段直接编译法), 长度分别为 2、2、3、3、4、4, 则最多可表示的微命令总数为 ① 个。

① 56      ② 18      ③  $2^{18}$       ④ 6

9. 根据模型机指令系统, 假设一条双操作数指令(该指令的地址为 A), 其源和目的均采用变址寻址, 则该指令执行完成时, PC 值等于 ③。

① A+1      ② A+2      ③ A+3      ④ A+4

10. 假设 R0 的内容为 2000H, R1 的内容为 2500H, (2000H)=3000H, (3000H)=3500H, 则指令“MOV R1, @(R0)+”执行完成后的结果是 ②。

① R0 的内容为 3500H      ② R1 的内容为 3500H  
③ 2000H 单元的内容为 3501H      ④ 3000H 单元的内容为 3501H

11. 一条机器指令中, 用 3 位表示操作码, 3 位表示源操作数寻址方式, 3 位表示源操作数寄存器编号。那么采用微程序设计方式时, 如果按源操作数寻址方式实现分支转移, 则最多可实现 ④ 分支。

① 3 个      ② 6 个      ③ 9 个      ④ 8 个

12. 与直接寻址相比, 采用寄存器寻址或寄存器间接寻址可缩短指令中地址信息的位数, 其原因是 ( ④ )。

① 一个寄存器中的数据位数比一个存储单元的数据位数少  
② 因为存储器容量很大, 很难用直接寻址访问到所有存储单元  
③ 一条指令的执行只处理少量的几个数据, 因此用少量寄存器就可存放。  
④ 因为寄存器数量远少于存储器单元的数量, 因此寄存器编号的位数远比存储器地址位数少。

1. 对外设采用单独编址方式(编址到寄存器), 则 CPU 进行 I/O 操作时, 只需用一般的数据传输指令即可完成 I/O 数据传输: ( )

外设单独编址时, 其地址码不能区分访存还是访问外设, 因此不能用一般的数据传输指令即可完成 I/O 数据传输

2. 为了标明当前指令的执行处于在哪个工作周期, 需要在程序状态字中设置相应的状态位来表征当前的工作周期状态。 ( )

为了标明当前指令的执行处于在哪个工作周期, 是通过设置相应的工作周期触发器, 而不是在在程序状态字中设置。

3. 假设一条数据传输指令的源和目的都采用变址寻址，指令长度为 48 位(包含形式地址)，数据总线和地址总线位数也均为 16 位，则该指令的执行需要从存储器取指令、取源操作数并送到目的存储单元，所以需要 3 次访问存储器。 ( )

由于取指令就需要3三次访问存储器，再加上数据传输还要访存，因此指令执行完成不止3次访存。

4. 对定点除法运算中的补码不恢复余数法，之所以不需要恢复余数，是因为上商时，只需要根据部分余数的正负即可决定商 0 或者商 1，因此不需要恢复余数就能得到正确的商值。

补码不恢复余数法是根据余数和除数的符号的异同来决定上商，而不是根据余数的正负来决定上商



5. 在微程序设计中,就确定后续微指令地址可采用增量方式和断定方式。所谓断定方式,其基本思想是:根据当前执行的微指令的执行情况,要么顺序执行下一条微指令,要么由当前指令提供转移的地址。( )

断定方式是由后续地址字段提供的后续地址形成方式来形成后续地址,不是顺序执行或者由当前指令提供转移的地址

6. 浮点数和乘法和除法运算的结果都需要进行规格化,因此,这两种运算的运算步骤的最后一操作都是进行规格化处理。

浮点除法运算的结果已经是规格化数,不需要对结果进行规格化处理

### 三、写出无条件转移指令“**JMP (R1)**”的指令流程和操作时间表 (16分)

FT0: M→IR, PC+1→PC	EMAR, R, SIR, PC→A, A+1, DM, 1→ET, CPET( $\bar{P}$ ), , CPT( $\bar{P}$ )
ET0: R1→MAR	R1→A, 输出 A, DM, CPMAR, T+1, CPT( $\bar{P}$ )
ET1: M→MDR→PC, MAR	EMAR, R, SMDR, MDR→B, 输出 B, DM, CPPC, CPMAR, 1→FT, CPFT( $\bar{P}$ ), CPT( $\bar{P}$ )