

1. MCS-51单片机复位后PC和SP的值各是多少？

- 复位后PC值为0000H，SP为07H

2. 为什么通常建议将SP的值设为1FH或更大？

- SP复位后为07H，而片内RAM的08H-1FH单元常被用过工作寄存器组1-3；如果不设置SP为更大值的话，容易与工作寄存器组1-3冲突。

3. 为什么MCS-51需要4组工作寄存器？这4组工作寄存器实际对应在哪里的存储区？通过何种方式选择其中一组工作寄存器？采取这种方式的主要设计动机是什么？

- 需要4组工作寄存器是为了提高效率，比如有四个线程要交叉运行，都想用到r0-r7寄存器，就可以让它们各占一组，只要切换区域就行了，而不必频繁地保护现场，这样可以提高效率
- 4组工作寄存器实际对应应在RAM的寄存器区，地址为00H-1FH
- 通过RS1和RS0的值的组合 ( $2 \times 2 = 4$ 种) 来确定是4组中的哪一组
- 设计动机是支持四组独立的寄存器区，避免频繁保护现场，从而提高效率，并且可以在PSW中通过改变RS1和RS0便捷地切换工作寄存器

4. 求振荡频率为6MHz时，MCS-51的乘法指令执行时间。

- 乘法指令占用4个机器周期，1个机器周期对应6个时钟周期，1个时钟周期对应2个振荡周期；因此乘法指令总共对应48个振荡周期。考虑到一个振荡周期时长 $1/6\mu s$ ，则乘法指令执行时间为 $48 \times 1/6 = 8\mu s$

5. 什么是位操作？MCS-51具有位寻址功能的地址范围有哪些？

- 位操作是直接对数据位进行的操作，常见的位操作有位变量传送、逻辑运算、控制程序转移等
- MCS-51的部份RAM（20H-2FH这16字节）和11个SFR具有位寻址功能

6. 都是64K寻址空间，MCS-51的片外程序存储和片外数据存储空间的编址有何不同？

- 片外程序存储  
51片内有4K字节ROM，片外16位地址线最多可扩展64K字节ROM，两者统一编址。通过EA（外部访问允许）引脚来选择
  - EA=1，程序地址<4KB，即在0000H~0FFFH区间时，从片内程序存储器中取指。程序地址>4KB，即在1000H~FFFFH区间时，从片外存储器取指
  - EA=0，所有指令均从片外程序存储器中读取，这时片外存储器从0000H开始编址
- 片外数据存储
  - 片外RAM64KB，地址范围0000H~FFFFH

7. ARM体系结构有哪几种工作状态？又有哪几种运行模式？其中哪些为特权模式？哪些为异常模式？并指出处理器在什么情况下进入相应模式？

- ARM体系结构有ARM(32-bit)和Thumb(16-bit)两种
- ARM体系结构有七种运行模式
  - 用户模式(User)，这是程序正常执行的模式
  - 特权模式
    - 系统模式(System)，用于运行特权级的操作系统任务
    - 异常模式
      - 快中断模式(FIQ)，用于高速数据传输和通道处理
      - 外部中断模式(IRQ)，用于普通的外部中断请求处理

- 管理模式(Supervisor), 这是供操作系统使用的保护模式
- 数据访问中止模式(Abort), 用于虚拟存储和存储保护
- 未定义模式(Undef), 用于支持硬件协处理器软件仿真
- 特权模式分别为: 系统模式(System)、异常模式; 异常模式又分为: 快中断模式(FIQ)、外部中断模式(Irq)、管理模式(Supervisor)、数据访问中止模式(Abort)、未定义模式(Undef);
- 进入每种模式的情况参见上方每种模式的介绍

8. 请写出以下相关ARM指令语句运行后, 寄存器的值(20分)

提示: PC值已知取址阶段, 求执行阶段

- a. MOV R0, PC (PC=0010H, R0和PC的值)
- b. ADD R0, R1, #1 (R1=002FH, R0和R1的值)
- c. LDR R0, =0x56000010 (R0的值)
- d. LDR PC, [PC, #4] (PC=0020H, PC的值)
- e. BIC R0, R1, R2, LSL #1 (R1=0023H, R2=000FH, R0和R1的值)

- a. R0=0010H, PC在执行阶段已经比读到了下两条指令, 所以为PC=0018H
- b. 指令意思为R0:=R1+1, 因此R0=0030H, R1=002FH
- c. 将0x56000010加载入R0, R0=0x56000010
- d. 在decode阶段, PC为0024H, 之后加#4后, 读取0028H处内存的值, 因此PC=[0028H], 即为0028H内存处的值
- e.  $R2 \ll 1 = 001EH$ , 按位取反后和R1取AND, 得到0021H, 故R0=0021H, R1不变, 仍为R1=0023H

9. 假设R0=0x12345678,使用将R0存储到0x4000的指令存到存储器中, 若存储器为大端组织, 写出从存储器0x4000处加载一个字节到R2的指令执行后R2的值

- 大端情况下, 78排在高地址端, 12排在低地址端, 因此加载出一个字节为0x12

10. 在Bootloader的stage1阶段, CPSR如何设置?

- 禁用所有中断, 将FIQ和IRQ设为1
- 可能还需要将模式位全置为1进入系统模式, 并置T=0设置为ARM状态。