**[1.5s]**每条指令的执行都有取指令和执行指令两个过程，所有指令的取指令都是相同的，不同的是执行指令。

**[1.5s]**取指令的过程为：

指令计数器PC中的内容经选择器送至RAM的地址端口；在DL和地址的共同作用下，指令在时钟的上升沿从RAM中读出送至总线BUS；在LD IR信号的作用下，BUS上的指令在时钟的下降沿载入到指令寄存器IR；同时指令计数器PC自加1，SM变为1。

**[1.5s]**每条指令执行都需要两个时钟周期完成，第一个周期取址，第二个周期执行。SM=0，表示当前为取址周期。

**[1.5s]**MADD为00，PC中的数据00000000通过选择器传至RAM的地址端口。

**[1.5s]**DL为1,XL为0，F BUS，FL BUS，FR BUS都为0，时钟上升沿，00H地址单元中的指令11000001读至总线BUS。

**[1.5s]**LD IR、WE、LN PC为1、LD PC为0，时钟下降沿，将总线BUS上的指令写入指令寄存器IR，同时PC自加1，SM变为1，准备执行该指令。

**[1.5s]**寄存器之间的传输指令MOV A,B

**[1.5s]**其执行过程为：将编码01对应的寄存器B的数据经A口读出，经ALU、移位逻辑传送至总线BUS；由WE控制在时钟下降沿将BUS上的数据写入编码00对应的寄存器A中。

**[1.5s]**SM等于1，执行MOV A,B指令，具体的执行由控制信号决定。

**[1.5s]**MOV A,B的指令编码是1100 0001，RAA1、RAA0为01，寄存器B中的数据通过A口输出，S3-S0为1100，M为0，ALU将输入端A的数据传至输出，F BUS为1、FL BUS和FR BUS为0，DL和XL都为0，移位逻辑将输入传至总线BUS。

**[1.5s]**WE、LD IR、LD PC、IN PC都为0，RWBA1，RWBA0为00，时钟下降沿，将总线上的数据写入寄存器A，至此指令执行完成，同时SM变为0，准备取下一条指令。

**[1.5s]**寄存器数据传输至RAM的指令MOV M,B

**[1.5s]**其执行过程为：编码11对应的寄存器C中的数据34H从通用寄存器B口读出，经选择器到达RAM的地址端口；编码01对应的寄存器B中的数据26H从通用寄存器组的A口读出，在经ALU、移位逻辑和总线BUS送至RAM的数据端，在XL信号和地址的共同作用下，在时钟上升沿总线BUS上的数据26H写入RAM的34H存储单元。

**[1.5s]**SM等于1，执行MOV M,B，具体的执行由控制信号决定。

**[1.5s]**MOV M,B的指令编码是1100 1101，RWBA1、RWBA0为11，寄存器C中的数据34H从通用寄存器B口读出，MADD为10，通过选择器到达存储器RAM的地址端口。RAA1、RAA0为01，寄存器B中的数据26H从通用寄存器组A口读出，S3-S0为1100，M=0，ALU将输入端A的数据传至输出，F BUS为1、FL BUS和FR BUS为0，移位逻辑将输入传至总线BUS。

**[1.5s]**XL、WE为1、DL、LD IR、LD PC、IN PC都为0，时钟上升沿，将总线上的数据26H写入RAM的34H存储单元，至此指令执行完成。

**[1.5s]**时钟下降沿，SM变为0，准备取下一条指令。

**[1.5s]**RAM数据传输至寄存器的指令MOV A,M。

**[1.5s]**其执行过程为：编码11对应的寄存器C中的数据34H从通用寄存器A口读出，经选择器到达RAM的地址端口；在DL信号和地址共同作用下，时钟上升沿将34H内存单元中的数据12H读出，传送至总线BUS，在WE控制下，时钟下降沿将BUS上的数据12H写入编码00对应的寄存器A中。

**[1.5s]**SM等于1，执行MOV A，M，具体的执行由控制信号决定。

**[1.5s]**MOV A,M的指令编码是1100 0011，RAA1、RAA0为11，寄存器C中的数据34H从通用寄存器A口读出，MADD为01，通过选择器到达存储器RAM的地址端口。

**[1.5s]**DL为1，XL为0，F BUS、FL BUS和FR BUS都为0，时钟上升沿将34H存储单元中的数据12H读至总线BUS。

**[1.5s]**WE、LD IR、LD PC、IN PC都为0，RWBA1、RWBA0为00，时钟下降沿，将总线上的数据12H写入寄存器A，至此指令执行完成，同时SM变为0，准备取下一条指令。

**[1.5s]**算术逻辑指令ADD、SUB、AND的执行过程基本相同，不同的是在ALU中实现的运算不同，所以可以统一将它们写成CCC的形式。

**[1.5s]**以CCC A,B为例，介绍算术逻辑类指令的执行过程：将编码01对应的寄存器B中的数据从通用寄存器组A口读出；同时将编码00对应的寄存器A中的数据从通用寄存器组B口读出；在S3-S0和M的控制下，两个操作数在ALU中进行运算，结果经移位逻辑传输至总线BUS；在WE控制下，时钟下降沿将BUS上的运算结果写入寄存器A中。ADD、SUB指令影响状态位CF和ZF。

**[1.5s]**SM等于1，执行ADD A,B，具体的执行由控制信号决定。

**[1.5s]**ADD A,B的指令编码是1001 0001，RAA1、RAA0为01，寄存器B中的数据从通用寄存器组A口读出，RWBA1、RWBA0为00，寄存器A中的数据从通用寄存器组B口读出，S3-S0为1001，M为1，ALU将寄存器A和寄存器B中的数据相加，如果和的低八位为0，则ZF等于1，否则为0，如果产生进位，则CF等于1，否则为0，同时输出相加的结果，F BUS为1、FL BUS和FR BUS为0，DL和XL都为0，移位逻辑将相加的和输出至总线BUS。

**[1.5s]**WE、LD IR、LD PC、IN PC都为0，RWBA1，RWBA0为00，时钟下降沿，将总线上的相加结果写入寄存器A，至此指令执行完成，同时SM变为0，准备取下一条指令。

**[1.5s]**NOT指令NOT B

**[1.5s]**其执行过程为：编码01对应的寄存器B中的数据从通用寄存器B口读出；在ALU中求反后经移位逻辑送入总线BUS；在WE控制下，时钟下降沿将BUS上求反的结果写入寄存器B。

**[1.5s]**SM等于1，执行NOT B，具体的执行由控制信号决定。

**[1.5s]**NOT B的指令编码是0101 0100，RWBA1、RWBA0为01，寄存器B中的数据从通用寄存器组B口读出，S3-S0为0101，M为1，ALU将输入端口B的数据取反后输出，F BUS为1、FL BUS和FR BUS为0，DL和XL都为0，移位逻辑将输入传至总线BUS，

**[1.5s]**WE、LD IR、LD PC、IN PC都为0，RWBA1，RWBA0为01，时钟下降沿，将总线上的数据写入寄存器B，至此指令执行完成，同时SM变为0，准备取下一条指令。

**[1.5s]**无条件跳转指令JMP 34H，其指令编码是00110000 00110100，其为双字节指令，取指周期取出了00110000存入指令寄存器IR，同时PC自加1，指向存放34H的地址单元。

**[1.5s]**其执行过程为：在DL信号控制下，时钟上升沿，从RAM中PC对应单元读出34H送至总线BUS，LD PC为1，时钟下降沿将总线BUS上的34H装载到PC中。

**[1.5s]**有条件跳转指令JZ 34H。

**[1.5s]**若ZF=1，则DL=1，LD PC=1，时钟上升沿从RAM中PC对应单元读出跳转地址34H，在时钟下降沿装载到PC中。若ZF=0，则IN PC=1，时钟下降沿PC自加1。

**[1.5s]**有条件跳转指令JC 34H

**[1.5s]**若CF=1，则DL=1，LD PC=1，时钟上升沿从RAM中PC对应单元读出跳转地址34H，在时钟下降沿装载到PC中。若CF=0，则IN PC=1，时钟下降沿PC自加1。

**[1.5s]**JMP与JZ、JC条件成立时的执行过程相同。JMP(JZ,JC) 34H的指令编码是0011 00CC 00110100，为双字节指令，存储在RAM的00H和01H两个存储单元中。

**[1.5s]**PC的值为0000 0001, MADD为00，PC的值经选择器传至RAM的地址端口。

**[1.5s]**DL为1，XL为0，F BUS、FL BUS和FR BUS都为0，时钟上升沿将01H单元中的0011 0100读取传至总线BUS。

**[1.5s]**WE、LD PC都为1，LD IR、IN PC为0，时钟下降沿，总线上的34H写入PC，至此指令执行完成。同时SM变为0，准备取下一条指令。

**[1.5s]**SM等于1，执行JZ(JC) 34H，此时条件不成立，DL，XL都为0，RAM不执行读写操作。

**[1.5s]**LD PC为0，IN PC为1，时钟下降沿，PC自加1，变为0000 0010，同时SM变为0，准备取下一条指令。

**[1.5s]**每条指令都需要两个周期完成，在控制信号的作用下，完成取指和执行操作。ADD A,B取指令时的控制信号为：SM为0，IN PC为1，LD PC为0, MADD为00, XL为0，DL为1, LD IR为1。

**[1.5s]**执行指令时的控制信号为：SM为1，RAA1 RAA0为01, RWBA1 RWBA0为00, WE为0，M为0，S3S2S1S0为1001，F BUS为1, FL BUS和FR BUS都为0。

**[1.5s]**控制信号产生逻辑中每个控制信号如何生成呢？前面分析了各条指令的执行过程，总结每个控制信号在各条指令取指和执行时所需的值，就能得到每个控制信号的逻辑表达式。以移位逻辑的F BUS为例说明，我们知道MOVA、MOVB、ADD、SUB、AND、NOT、OUT指令都需要将数据经移位逻辑直传至总线BUS，这时F BUS为1。

**[1.5s]**所以F BUS是MOVA、MOVB、ADD、SUB、AND0、NOT0和OUT0的逻辑和。