

$$\begin{aligned} 5. x &= (-17.25)_{10} \\ &= (-10001.01)_2 \\ &= -0.1000101 \times 2^5 \\ &= 1.0111011 \times 2^{-10} \end{aligned}$$

$$\begin{aligned} X_{16} &= 0000101 \ 1.0111011 \\ &= 05BBH \end{aligned}$$

$$6. Y_8 = 11111110 \ 0111000$$

$$\begin{aligned} Y &= 0.1111000 \times 2^{-10} \\ &= 0.001111 \end{aligned}$$

$$\begin{aligned} 7. E_x &= 101 \\ E_y &= -10 \quad E_x - E_y = 7 > 0 \end{aligned}$$

M_y 右移 7 位, 阶数加 7

$$M_{y_{补}} = 0.0000000(1111)$$

$$\begin{array}{r} M_{x_{补}} \quad 1.0111011 \\ + M_{y_{补}} \quad 0.0000000(1111) \\ \hline 1.0111011(1111) \end{array}$$

令入

$$Z_{16} = 0000101 \ 1.0111100$$

$$8. PC \rightarrow AR, PC+1 \rightarrow PC$$

↓
RAM → IR

↓
 $R_0 \rightarrow DA_1$

↓
 $R_1 \rightarrow DA_2$

↓
 $DA_1 + DA_2 \rightarrow R_0$

2. 14, 15, F4H

3. (8)
0000 0101
0001 0000

(9) 指令助记符 源操作数 寻址方式 结果

1. MOV 58H 立即寻址 $R_2 = 58H$

2. ADD R_2 和 (27H) 变址 $R_2 = 21H + 58H = 79H$

3. SUB R_2 和 11H 直接 $R_2 = 68H$

4. HALT

4条指令

$$4. (14 + 4 + 3 + 7) \times 2^7 \text{ 位} \quad (18, 3, 7)$$

$$\geq 28 \times 2^7 \text{ 位}$$

5. 控制器:

① 程序计数器(PC):

存放当前指令或下一条指令的地址

② 指令寄存器(IR):

控制器从内存中取出指令, 放 IR 中, 以便进行指令译码。

③ 指令译码器:

将操作码译码或控制信号

④ 操作控制信号形成部件:

产生各部件所需的操作控制信号, 完成各种指令的正确操作。

⑤ 时序信号产生器:

提供时钟信号和机器周期信号

⑥ 地址寄存器

容纳各种来源的地址, 统一在
规定节拍地址送上总线。

⑦ 数据寄存器

13. 频率高 指令短

依. 长

14. 极间电容, 慢, 依

$$15. 50 \times 128 = 6400ns$$

$$16. 0000000H \sim 3FFFFFFH$$

64K 芯片

图略

$$6. \frac{1700}{2000} = 85\%, \quad 20.5ns \quad \checkmark \text{ 标记 6, 组地址 8 位}$$

$$64K = 2^{16} \quad 64K/4K = 16 = 2^4$$

4 8 2 2

$$7A8H = 0111 \ 1010 \ 1000$$

$$2^1 + 2^3 + 2^5 + 2^6 + 2^7$$

$$= 2 + 8 + 32 + 64 + 128$$

$$= 234 \text{ 组}$$

$$7. 200 + 20 \times 50 = 1200ns$$

$$200 \times 21 - 1200 = 3000ns$$

$$8. 100 \text{ 字节}, 6400 \text{ 字节}$$

9.

指令	w_r_s	imm_s	rt_imm_s	wr_data_s	ALU_OP	Write_Reg	Mem_Write	IO_R	IO_W	PC_s
5)	00	-	0	00	0100	1	0	0	0	00
7)	01	1	1	00	0101	1	0	0	0	00
8)	-	1	0	-	0101	0	0	0	0	00/10
9)	-	-	-	-	-	0	0	0	0	11
10)	-	1	0	-	1000	0	1	0	0	00

10. in rt imm 从 [imm] 输入到 rt

out rt imm 从 rt 输出到 [imm]

指令	w_r_s	imm_s	rt_imm_s	wr_data_s	ALU_OP	Write_Reg	Mem_Write	IO_R	IO_W	PC_s
in	-	0	1	11	-	1	0	1	0	00
out	01	0	1	-	-	0	0	0	1	00