



6.1 ④ a. 数据寄存器.

b. 指令寄存器.

c. 地址寄存器.

d. PC寄存器.

12. 主寄存器M \rightarrow b(GP) \rightarrow 微操作信号发生器.

③. 寄存器M \rightarrow a(DR) \rightarrow ALU \rightarrow AC.

存: AC \rightarrow a(DR) \rightarrow 寄存器M.

6.12 ① $CPI \times 2 \times 2 = 4$
 $f = 16 \times 10^6 \text{ Hz}$

c. 机器速度 = $\frac{f}{CPI} = \frac{16 \times 10^6}{4} = 4 \times 10^6 \text{ Hz} = 4 \text{ MIPS}.$

② 插入一个机器周期 则一个指令需要3个机器周期

机器速度 = $\frac{f}{CPI} = \frac{16 \times 10^6}{6} = 2.67 \text{ MIPS}$

③ A: $f = \frac{1}{0.125 \mu s} = 8 \text{ MHz} = 0.125 \mu s$

B: $\frac{1}{0.4} = 2.5 \mu s.$

C: $\frac{f}{0.4 \text{ MIPS}} = \frac{12}{X} \quad X = 0.6 \text{ MIPS}.$

A ① B ① ③ C ④



6.2.

1. JMP. 中取指 1010 1000 0000 0000 0000 110
12. 地址 0001 0001 0100 1000 0010 0xx

2. Load 中取指 1010 1000 0000 0000 0000 110
12.

13. 地址. $RS1 \rightarrow GR$. $RS1 \rightarrow ALU$. $DISP \rightarrow ALU$. $ALU + ALU \rightarrow AR$
0001 0001 0100 1000 0010 0xx

13. $AR \rightarrow AB$. $AB \rightarrow DR$. ADS^{21} MIO^{22} . ($W/R=0$)
0000 0100 0000 0000 0000 110

14. $DR \rightarrow ALL$ $RS1 \rightarrow GR$ + $ALU \rightarrow GR$ $PC \rightarrow AB$.
1000 0000 1001 1000 1000 0xx

13. store. 指令.

1. 取指 1010 1000 0000 0000 0000 110.

12. 指令有效地址 $RS1 \rightarrow GR$ $RS1 \rightarrow ALU$ $DISP \rightarrow ALU$ $ALU + ALU \rightarrow AR$
0001 0001 0100 1000 0010 0xx

13. $RS1 \rightarrow GR$ ~~$RS1 \rightarrow ALU$~~ $RS1 \rightarrow ALU$ $ALU \rightarrow DR$ ~~$DR \rightarrow DR$~~ +
0000 0000 1010 1000 0100 0xx

14. $AR \rightarrow AB$, $DR \rightarrow DB$, ADS , MIO , W/R $PC \rightarrow AB$
1000 0010 0000 0000 0001 111



6.5. 找出互反字节

a i b f g i j c f j d i j e f h j

0: a d e h

0: 无操作

1: a f d e h

0: 无操作

01: c

10: f

11: j

0: 无操作

01: b

10: g

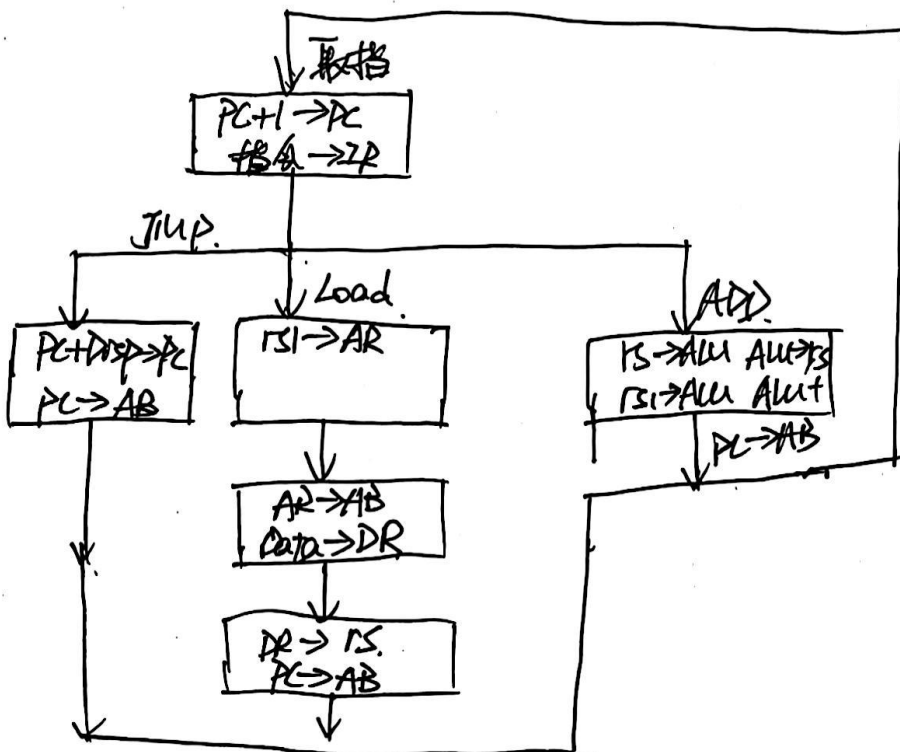
11: i

6.6. 有4个转移指令, 判断测试段4位

$\log_2 12 = 9$ 下地址段为9

刚做指令段 $48 - 9 - 4 = 35$ 位

6.7.





6.8 jmp 取指 译码 执行 送结果

指令执行后可以有直接送入下一个指令使用

取指 译码 执行 \rightarrow RS

取址 译码 执行 送结果

1 2 3 4 5 6 7 8

一共需要8个周期。

6.9 如果使用硬布线, 则组合逻辑在一个周期内输出会影响。

6.14 A: ① ~~200ns~~ B: ② 用两个机器周期完成。
A: ③ 200ns

6.17. 由reset信号设置R的初始值

6.18 字节, 0 1 2 3 4 5 6 7 8 9
Ni 4 6 3 11 9 5 7 1 8 15
字节位数 3 3 2 4 4 3 3 1 4 4
完全水平型 4 6 3 11 9 5 7 1 8 15

sum = 31

sum = 69

采用编码表示法需要31位

采用完全水平型的方式, 需要69位