**《计算机组成原理与系统结构》勘误**

1. 第6章，所有的MYM都改为$。（这个应该是排版问题，比较多，比较难看）
2. P259页习题6的6.9 ，改为“选择题”
3. P260页，第9行，“C. 前者是确定程序顺序执行，后者是取操作数地址。”改为：

C. 前者是确定程序执行顺序，后者是确定操作数地址。

1. P260页倒数第6行，改为“该指令格式最多可定义多少种不同的指令?”
2. P273页，图7.9上的存储器左边的一个总线应该是双向的。



1. P275页上，图7.11中，AR方框上方的总线应该是单向向下的。



1. P279页图7.16中黄底红色字为改过的（包括黑色箭头）：



1. P284页：图7.24 多端口寄存器堆和运算器，应该改为：图7.24 R型指令的数据通路。
2. P313~P314：表7.10中ALU\_OP信号漏掉，或者表达不正确；修正如下：黄底红色的为修改过的；

| 时钟周期 | 操作 | 发送控制信号 |
| --- | --- | --- |
| ① R型指令 | | |
| M0 | Mem[PC]→IR,  PC+4→PC | I\_D\_s=0,Mem\_read,IR\_write;ALU\_OP=+;  ALU\_A\_s=0, ALU\_B\_s=00, PC\_s=00, PC\_write; |
| M1 | Reg[rs]→A,Reg[rt]→B | 无 |
| M2 | A （op）B→F | ALU\_A\_s=1, ALU\_B\_s=01, ALU\_OP=\*\*\*; |
| M3 | F→ Reg[rd] | rd\_rt\_s=0,alu\_mem\_s=0, Reg\_write; |
| ② I型访存指令：lw | | |
| M0 | Mem[PC]→IR,  PC+4→PC | I\_D\_s=0,Mem\_read,IR\_write; ALU\_OP=+;  ALU\_A\_s=0, ALU\_B\_s=00, PC\_s=00, PC\_write; |
| M1 | Reg[rs]→A,Reg[rt]→B | 无 |
| M2 | A + offset →F | ALU\_A\_s=1, ALU\_B\_s=10, imm\_s=1,ALU\_OP=+; |
| M3 | Mem[F]→MDR | I\_D\_s=1,Mem\_read; |
| M4 | MDR→ Reg[rt] | rd\_rt\_s=1,alu\_mem\_s=1, Reg\_write; |
| ③ I型访存指令：sw | | |
| M0 | Mem[PC]→IR,  PC+4→PC | I\_D\_s=0,Mem\_read,IR\_write; ALU\_OP=+;  ALU\_A\_s=0, ALU\_B\_s=00, PC\_s=00, PC\_write; |
| M1 | Reg[rs]→A,Reg[rt]→B | 无 |
| M2 | A + offset →F | ALU\_A\_s=1,ALU\_B\_s=10,imm\_s=1, ALU\_OP=+; |
| M3 | B → Mem[F] | I\_D\_s=1,Mem\_write; |
| ④ I型分支指令：beq | | |
| M0 | Mem[PC]→IR,  PC+4→PC | I\_D\_s=0,Mem\_read,IR\_write;ALU\_A\_s=0, ALU\_B\_s=00, ALU\_OP=+,PC\_s=00, PC\_write; |
| M1 | Reg[rs]→A,Reg[rt]→B  PC+offset\*4→F | ALU\_A\_s=0,ALU\_B\_s=11, ALU\_OP=+,imm\_s=1; |
| M2 | A - B ，产生zero  zero=1，则F→PC  zero=0，空操作 | ALU\_A\_s=1, ALU\_B\_s=01, ALU\_OP=-;  zero=1:PC\_s=01,PC\_write |
| ⑤ J型跳转指令：J | | |
| M0 | Mem[PC]→IR,  PC+4→PC | I\_D\_s=0,Mem\_read,IR\_write;ALU\_A\_s=0,ALU\_B\_s=00, ALU\_OP=+,PC\_s=00, PC\_write; |
| M1 | {PC[31:28],address, 2‘b00}→PC | PC\_s=10,PC\_write |

1. P321页，倒数第6行中：将①与②换一下，改为：微程序控制器的设计主要完成两个任务：①产生正确的微指令序列（即上述CPU状态转换序列；②产生正确的微命令。
2. P322页图7.50中，RAM →DA2，应改为DR →DA2，如下图红字所示。



1. P323页表7.12中，应按如下修改：黄底红色的为修改过的

表7-12 简单微控器中微指令的下址设计表

|  |  |  |  |
| --- | --- | --- | --- |
| **微地址** | **状态** | **判别测试字段（J1#）** | **下址字段** |
| 000 | 取指1：PC🡪AR,PC+1 | 1 | 001 |
| 001 | 取指2：RAM🡪IR,译码 | 0 | ××× |
| 010 | ADD1：PC🡪AR,PC+1 | 1 | 011 |
| 011 | ADD2：RAM🡪DA1 | 1 | 100 |
| 100 | ADD3：DR🡪DA2 | 1 | 101 |
| 101 | ADD4：DA1+DA2🡪R0 | 1 | 000 |
| 110 | JMP1：PC🡪AR,PC+1 | 1 | 111 |
| 111 | JMP2：RAM🡪PC | 1 | 000 |

1. P323页，第1行中，应改为：“至此，已经完成了微程序控制器设计任务中的第1个任务，产生了正确的微指令序列。然后，考虑设计任务中的第2个任务，”；第6行中，应改为：“从而完成微程序控制器设计任务中的第2个任务。”
2. P323页表7.13中，应按如下修改：黄底红色的为修改过的
3. 表7-13 简单计算机系统的控制信号一览表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **序号** | **控制信号** | **功能** | **序号** | **控制信号** | **功能** |
| 1 | PC-B# | 指令地址（程序计数器）送总线 | 10 | S1 | 同上 |
| 2 | B-AR | 总线内容打入地址寄存器 | 11 | S0 | 同上 |
| 3 | PC+1 | 程序计数器内容加一 | 12 | M | M为“1”选择ALU做逻辑运算，M为“0”选择ALU做算数运算 |
| 4 | B-PC | 总线内容打入程序计数器 | 13 | B-DA1 | 总线内容打入暂存器DA1 |
| 5 | B-IR | 总线内容打入指令寄存器 | 14 | B-DA2 | 总线内容打入暂存器DA2 |
| 6 | M-W＃ | 存储器写 | 15 | ALU-B＃ | 运算器ALU内容送总线 |
| 7 | M-R＃ | 存储器读 | 16 | Ci | ALU进位输入 |
| 8 | S3 | S3- S0选择ALU16种运算之一 | 17 | B-DR | 总线内容打入目的寄存器DR |
| 9 | S2 | 同上 | 18 | DR-B | 目的寄存器DR内容送总线 |

1. P324页图7.51中将SR-B改为DR-B，相应地译码器输入线也改为从I1I0引出。



1. P324页倒数第4-7行中，应改为：黄底红色的为修改过的

表7-12中微地址为100的这条微指令ADD3：DR🡪DA2需要**R0-B**信号，而表7-13中只有第18微命令DR-B，此时还需要ADD指令码第1字的I1I0 （表示目的寄存器DR编号）2位通过译码来分别产生R0-B、R1-B、R2-B、R3-B，因前述ADD指令码第1字中的I1I0为00，所以经译码产生的是**R0-B**信号。

1. P324页图7.52中，应该为如下：红字的为修改过的



1. P324页图7.53中，应改为如下：红字的为修改过的



1. P326页第20行“一次也只能有一个有效；这些信号相互都是互斥的。”应改为：“虽然理论上可以同时有效，但在实际应用中不会或很少同时有效，因此，可以将这些信号视为互斥的。”
2. P327页第6-9行中，应改为如下：黄底红色的为修改过的

综上所述，表7.-14模型机的微指令格式包含了上述3种，其中第M13位至第M8位的S3S2S1S0MCi 均为控制信号，即采用直接控制法；第M23位到第M18位BTO字段和OTB字段为字段直接编译法，BTO字段定义的是信息从总线到部件的控制信号，OTB字段定义的是部件的信息到总线的控制信号；第M17位至第M14位是字段间接编译法，FUNC字段编码有二组定义，由FS字段１位来确定二组中的一组定义信号。

1. P334页倒数第5行“指令的执行过程如下。”改为：“参照图7.63微程序流程图，指令的执行过程如下。”
2. P336页表7.16中多了一个空格列。

表7-16 模型机微指令格式

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **M23M22M21** | **M20M19M18** | **M17 M16 M15** | **M14** | **M13** | **M12** | **M11** |  | **M10** | **M9** | **M8** | **M7** | **M6M5M4M3M2M1M0** |
| BTO(3) | OTB(3) | FUNC(3) | FS | S3 | S2 | S1 |  | S0 | M | Ci | 空 | MA6-MA0(7) |

1. P337页中第13行“
2. J1# =0时，根据机器指令的操作码（OP）I7~I4进行散转，产生该条指令的微程序入口地址，散转微地址确定如下：”下面第15-19行应改黄底红字。

* 当I7I6=11时(即指令格式4的指令)，则=0，即1→MA5，并且：
  + 若I5=1，则=0，即1→MA3；
  + 若I4=1，则=0，即1→MA2；
  + 若I3=1，则=0，即1→MA1；
  + 若I2=1，则=0，即1→MA0；
* 当I7I6≠11时(即指令格式1、2、3的指令) ，则：
  + 若I7=1，则=0，即1→MA3；
  + 若I6=1，则=0，即1→MA2；
  + 若I5=1，则=0，即1→MA1；
  + 若I4=1，则=0，即1→MA0；

1. P344页表应改为如下：最后一行删除，黄底红色的为修改过的

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 微指令 | 应发送的微操作控制信号 | 微指令编码（M24～M7） | | | | |
| BTO | OTB | FUNC | FS | S3～S0 MCi |
|  | PC→AR，PC+1 | PC-B#，B-AR，PC+1 | 110 | 111 | 000 | 1 | 000000 |
|  | RAM→IR | M-R#，B-IR | 011 | 000 | 010 | 0 | 000000 |
|  | J1# 方式散转 | J1#=0 | 000 | 000 | 001 | 1 | 000000 |
|  | SR→DR | SR-B#，B-DR | 100 | 011 | 000 | 0 | 000000 |
|  | DR→RAM | DR-B#，M-W# | 000 | 100 | 001 | 0 | 000000 |
|  | DA1+DA2→DR | ALU(F=A加B)，影响标志，ALU-B#，B-DR | 100 | 001 | 111 | 1 | 100101 |
|  | DA1→LED | ALU(F=A)，ALU-B#，I/O-W# | 000 | 001 | 011 | 0 | 000001 |

1. P346页第8行，“烦琐”应改为“繁琐”，倒数第5行“烦琐”改为“繁琐”。
2. P347页习题7.8的描述应改为：黄底红色的为修改过的

7.8 参见图7.27所示的单周期CPU数据通路，在其上执行下列MIPS指令，描述执行过程，写出译码与控制单元所需设置的控制信号，填入下表。假设ALU功能如下：

ALU\_OP=000：算术加；=001：算术减；=010：位异或；=011：逻辑左移；=100：小于置位。

| 指令 | w\_r\_s | imm\_s | rt\_imm\_s | wr\_data\_s | ALU\_OP | Write\_Reg | Mem\_Write | PC\_s |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| xor rd,rs,rt |  |  |  |  |  |  |  |  |
| sltu rd,rs,rt |  |  |  |  |  |  |  |  |
| sllv rd,rt,rs |  |  |  |  |  |  |  |  |
| xori rt, rs, imm |  |  |  |  |  |  |  |  |
| sw rt, offset(rs) |  |  |  |  |  |  |  |  |
| bne rs, rt, label |  |  |  |  |  |  |  |  |
| jal label |  |  |  |  |  |  |  |  |

1. P349页习题7.17描述应改为“参见图7.60数据通路，”。
2. P350页习题7.18描述应改为“根据图7.60所示的模型机结构和数据通路，”。“该指令完成（A）－R→（A），”应改为“该指令完成（A）－R→ A”。