

## 习题 8

- 8.1 如果表 8.1 中 RV32I 指令集增加了 lb、lbu、lh、lhu 和 sb、sh 六条指令，考虑修改图 8.5 的数据存储器，以满足指令所需的多尺寸读取存储器。回答以下问题：
- (1) DM\_Addr 需要几位？按照字节、半字和字访问时，地址有什么特征？
  - (2) 读出数据 M\_R\_Data 要如何处理？添加什么部件和什么控制信号？
  - (3) 写入数据 M\_W\_Data 要如何处理？添加什么部件和什么控制信号？
  - (4) 画出能支持全部 8 条访存指令的数据存储器的结构框图，并给出读和写字节、半字、字数据操作时，控制信号如何设置。
- 8.2 在图 8.8 的基础上，再实现 auipc rd,imm20 指令，其功能是： $PC+\{imm20,12\{0\}\}\rightarrow rd$ 。请给出设计方案：
- (1) 描述添加或修改的部件，画出系统结构图；
  - (2) 给出 auipc 的指令流程图，写出各机器周期发送的控制信号；
  - (3) 分析上述修改，是否影响到其他指令的执行过程或发送的控制信号？如果有，请说明。
- 8.3 假设图 8.8 上不设置 PC\_Write 和 IR\_Write 信号，请问可以吗？说明原因或者解决方案。
- 8.4 图 8.8 支持了 R 型和 I 型运算指令、lui 指令的执行，分析它们所需要的机器周期数，仿照 7.2.4 节的图 7.24 和表 7.5 的设计过程，设计出类似图 7.25 的时序系统。
- 8.5 基于图 8.8，画出下面 4 条指令的指令流程图，写出发送的控制信号：
- (1) and rd, rs1, rs2
  - (2) ori rd, rs1, imm12
  - (3) srli rd, rs1, shamt
  - (4) lui rd, imm20
- 8.6 分析图 8.9 中的指令存储器 IM 和数据存储器 DM 的地址空间，是统一的，还是独立的？假设要求指令存储器 IM 的地址空间是 0000 0000H~0FFF FFFFH，而数据存储器空间是从 1000 0000H~1FFF FFFFH，请问应该如何进行连接？
- 8.7 将你设计的题 8.1 要求的存储器连接到图 8.9 中，请回答：
- (1) 指令 lw 和 sw 的执行过程有变化吗？
  - (2) 重新写出执行指令 lw 和 sw 所发送的控制信号。
  - (3) 支持的六条新指令 lb、lbu、lh、lhu 和 sb、sh，执行所需的机器周期数和过程，与 lw 和 sw 相同吗？发送的控制信号呢？
- 8.8 图 8.10 与图 8.11 只考虑实现了 beq 指令，如果添加另外的 5 条分支指令 bne、blt、bge、bltu、bgeu，请思考：
- (1) beq 发生转移的条件是 ZF=1，那么这 5 条指令发生转移的条件应考虑哪些标志？
  - (2) 如何修改或者设计电路实现它们？说明你的设计方案。
- 8.9 分析图 8.10，假设相对寻址的 jal 和 beq 指令，不采用专用的地址加法器，而是使用主 ALU 来实现，则应该如何修改电路？分析这两种设计方法的优缺点。
- 8.10 基于图 8.10，重新考虑实现指令 auipc，如何修改与连接？发送什么控制信号？
- 8.11 基于图 8.10，画出下面指令的指令流程图，写出发送的控制信号：
- (1) xor rd, rs1, rs2
  - (2) addi rd, rs1, imm12
  - (3) lw rd, imm12(rs1)
  - (4) sw rs2, imm12(rs1)

- (5) lui rd, imm20  
 (6) jal rd, offset20  
 (7) jalr rd, offset12(rs1)  
 (8) beq rs1, rs2, offset12

参考答案:

指令	机器周期	执行操作	发送的控制信号 (没有列出的无关)
所有指令	M0	IMem[PC]→IR, PC→PC0, PC+4→PC	IR_Write=1, PC_Write=1, PC0_Write=1, PC_s=00, (Mem_Write=0, Reg_Write=0)
xor rd, rs1, rs2	M1	Reg[rs1]→A, Reg[rs2]→B	无 (IR_Write=0, PC_Write=0, PC0_Write=0, Mem_Write=0, Reg_Write=0)
	M2	$A \oplus B \rightarrow F$	ALU_OP=0100, rs2_imm_s=0 (IR_Write=0, PC_Write=0, PC0_Write=0, Mem_Write=0, Reg_Write=0)
	M3	F→Reg[rd]	Reg_Write=1, w_data_s=00 (IR_Write=0, PC_Write=0, PC0_Write=0, Mem_Write=0)
addi rd, rs1, imm12	M1	Reg[rs1]→A, Reg[rs2]→B	无 (IR_Write=0, PC_Write=0, PC0_Write=0, Mem_Write=0, Reg_Write=0)
	M2	$A + \text{imm32} \rightarrow F$	ALU_OP=0000, rs2_imm_s=1 (IR_Write=0, PC_Write=0, PC0_Write=0, Mem_Write=0, Reg_Write=0)
	M3	F→Reg[rd]	Reg_Write=1, w_data_s=00 (IR_Write=0, PC_Write=0, PC0_Write=0, Mem_Write=0)
lw rd, imm12(rs1)	M1	Reg[rs1]→A, Reg[rs2]→B	无 (IR_Write=0, PC_Write=0, PC0_Write=0, Mem_Write=0, Reg_Write=0)
	M2	$A + \text{imm32} \rightarrow F$	ALU_OP=0000, rs2_imm_s=1 (IR_Write=0, PC_Write=0, PC0_Write=0, Mem_Write=0, Reg_Write=0)
	M3	DMem[F]→MDR	无 (IR_Write=0, PC_Write=0, PC0_Write=0, Mem_Write=0, Reg_Write=0)
	M4	MDR→Reg[rd]	Reg_Write=1, w_data_s=10 (IR_Write=0, PC_Write=0, PC0_Write=0, Mem_Write=0)
sw rs2, imm12(rs1)	M1	Reg[rs1]→A, Reg[rs2]→B	无 (IR_Write=0, PC_Write=0, PC0_Write=0, Mem_Write=0, Reg_Write=0)
	M2	$A + \text{imm32} \rightarrow F$	ALU_OP=0000, rs2_imm_s=1 (IR_Write=0, PC_Write=0, PC0_Write=0, Mem_Write=0, Reg_Write=0)
	M3	B→DMem[F]	Mem_Write=1 (IR_Write=0, PC_Write=0, PC0_Write=0, Reg_Write=0)
lui rd, imm20	M1	$\text{imm32} \rightarrow \text{Reg[rd]}$	Reg_Write=1, w_data_s=01 (IR_Write=0, PC_Write=0, PC0_Write=0, Mem_Write=0)
jal rd, offset20	M1	PC→Reg[rd], PC0+imm32→PC	Reg_Write=1, w_data_s=11, PC_Write=1, PC_s=01 (IR_Write=0, PC0_Write=0, Mem_Write=0)
jalr rd,	M1	Reg[rs1]→A, Reg[rs2]→B	无 (IR_Write=0, PC_Write=0, PC0_Write=0,

offset12(rs1)			Mem_Write=0, Reg_Write=0)
	M2	A+imm32→F	ALU_OP=0000, rs2_imm_s=1 (IR_Write=0, PC_Write=0, PC0_Write=0, Mem_Write=0, Reg_Write=0)
	M3	PC→Reg[rd] F→PC	Reg_Write=1, w_data_s=11, PC_Write=1, PC_s=10 (IR_Write=0, PC0_Write=0, Mem_Write=0)
bneq rs1, rs2, offset12	M1	Reg[rs1]→A, Reg[rs2]→B	无 (IR_Write=0, PC_Write=0, PC0_Write=0, Mem_Write=0, Reg_Write=0)
	M2	A-B→F/FR	ALU_OP=1000, rs2_imm_s=0 (IR_Write=0, PC_Write=0, PC0_Write=0, Mem_Write=0, Reg_Write=0)
	M3	ZF=1: PC0+Imm32→PC	ZF=1: PC_Write=1, PC_s=01 或者: PC_Write=ZF, PC_s=01 (IR_Write=0, PC0_Write=0, Mem_Write=0, Reg_Write=0)

8.12 一条指令的执行需要下面 4 个机器周期，请写出该条指令助记符，描述其功能。

M<sub>0</sub>: IMem[PC]→IR, PC+4→PC, PC→PC0

M<sub>1</sub>: Reg[rs1]→A, Reg[rs2]→B

M<sub>2</sub>: A-B→F/FR

M<sub>3</sub>: ZF=0, 则 PC0+imm32→PC

参考答案:

bne rs1, rs2, offset12; 不相等则跳转

8.13 一条指令的执行需要下面 5 个机器周期，请写出该条指令助记符，描述其功能。

M<sub>0</sub>: IMem[PC]→IR, PC+4→PC, PC→PC0

M<sub>1</sub>: Reg[rs1]→A

M<sub>2</sub>: A+imm32→F

M<sub>3</sub>: SE32[DMem16[F]]→MDR

M<sub>4</sub>: MDR→Reg[rd]

参考答案:

lh rd, imm12(rs1); 装有符号半字数据到 rd

8.14 一条指令的执行需要下面 4 个机器周期，请写出该条指令助记符，描述其功能。

M<sub>0</sub>: IMem[PC]→IR, PC+4→PC, PC→PC0

M<sub>1</sub>: Reg[rs1]→A

M<sub>2</sub>: A+imm32→F

M<sub>3</sub>: PC→Reg[rd], F→PC

参考答案:

jalr rd, offset12(rs1); 寄存器相对跳转

8.15 基于图 8.10，写出题 8.12 的 M<sub>3</sub> 周期、题 8.13 的 M<sub>0</sub> 和 M<sub>4</sub> 周期、题 8.14 的 M<sub>3</sub> 周期，要发送的控制信号（无关信号请写—或者 x）。

参考答案：

题 8.12	bne rs1, rs2, offset12	M <sub>3</sub>	ZF=0: PC0+imm32→PC	ZF=0: PC_Write=1, PC_s=01 或者: PC_Write= $\overline{ZF}$ , PC_s=01 (IR_Write=0, PC0_Write=0, Mem_Write=0, Reg_Write=0)
题 8.13	lh rd, imm12(rs1)	M <sub>0</sub>	IMem[PC]→IR, PC+4→PC, PC→PC0	IR_Write=1, PC_Write=1, PC0_Write=1, PC_s=00 (Mem_Write=0, Reg_Write=0)
		M <sub>4</sub>	MDR→Reg[rd]	Reg_Write=1, w_data_s=10 (IR_Write=0, PC_Write=0, PC0_Write=0, Mem_Write=0)
题 8.14	jalr rd, offset12(rs1)	M <sub>3</sub>	PC→Reg[rd] F→PC	Reg_Write=1, w_data_s=11 PC_Write=1, PC_s=10 (IR_Write=0, PC0_Write=0, Mem_Write=0)

8.16 某指令的机器周期完成操作：PC→Reg[rd]，PC0+imm32→PC。请基于图 8.10，写出需发送的控制信号（无关信号请写—或者 x）。

参考答案：

Reg\_Write=1, w\_data\_s=11, PC\_Write=1, PC\_s=01 (IR\_Write=0, PC0\_Write=0, Mem\_Write=0)

8.17 如果某条指令的某个机器周期，发送了如下的控制信号，请写出该周期执行的操作：  
PC\_Write=0, PC0\_Write=0, IR\_Write=0, Reg\_Write=1, Mem\_Write=0, rs2\_imm\_s=0,  
ALU\_OP=0000, PC\_s=00, w\_data\_s=10

参考答案：

有效的显式操作：MDR→Reg[rd]

隐式操作有三个：

①Reg[rs1]→A, Reg[rs2]→B

②A+B→F/FR

③DMen[F]→MDR

8.18 如果 rs2\_imm\_s 信号出现了恒 1 故障，请问下面哪些指令不能正确执行？恒 0 故障呢？

- A. add rd, rs1, rs2
- B. addi rd, rs1, imm
- C. beq rs1, rs2, offset
- D. lw rd, imm12(rs1)
- E. sw rs2, imm12(rs1)
- F. lui rd, imm20
- G. jal rd, offset

---

H. jalr      rd, offset(rs1)

参考答案:

若出现恒 1 故障: A, C 指令不能正确执行

若出现恒 0 故障: B, D, E, H 指令不能正确执行

8.19 如果 Reg\_Write 信号出现了恒 0 故障, 重新回答题 8.18。

参考答案:

若出现恒 0 故障: A, B, F, G, H 指令不能正确执行

8.20 图 8.10 是采用了五阶段操作完成指令执行, 经过改造可实现五级流水线 CPU。另一种三级流水线常用于嵌入式处理器: 取指令阶段 IF、指令译码与读寄存器阶段 ID、执行阶段 EX。IF 和 ID 阶段同图 8.10, 但是 EX 阶段操作完成 ALU 运算、读写数据存储器、写回目的寄存器。思考如何修改图 8.10, 完成三阶段操作的设计, 并写出 add、addi、lw、sw 和 jal 指令的执行过程。

8.21 在例题 6.39 的基础上, 进一步了解国产龙芯处理器的应用领域有哪些? 思考: 为何要自主研发处理器?

8.22 请调研: 国内生产研发 RISC-V 产品的企业有哪些? 它们的产品、应用领域、技术有何特点? 你认为采用开源的 RISC-V 架构有何优势和劣势?

8.23 请选择下面一种或者几种处理器进行调研或者对比, 内容包括指令集体系架构、系统结构、流水线、Cache 技术、处理器应用等方面, 撰写一篇新技术研学报告, 要求格式规范、逻辑清晰、内容充实, 附参考文献。

- (1) 龙芯处理器;
- (2) 华为麒麟、昇腾、鲲鹏系列处理器;
- (3) 平头哥玄铁系列处理器;
- (4) 申威处理器。