

手写笔记

add	000000	ori	000000	100000	①或并错了 ②memory寻址
sub	000000	ori	000000	100000	
ori	000000	ori	000000	100000	
lw	000000	ori	000000	100000	00 加
sw	000000	ori	000000	100000	01 减
beq	000000	ori	000000	100000	10 或
lui	000000	ori	000000	100000	ExtOp
nop	000000	ori	000000	100000	00 无符号 01 有符号 10 lui

设计文档: jal 000011 / 跳转并链接 PC ← GRP[31] ← PC + 4

jr 000000 00/000 PC ← GRP[rs]

nPC 4种 + 4: other + 4

01 beq: imm16.

10 jal: imm26

11 jr: 要GRP[rs]

控制写地址: rd: 01

Regdst rs: 00 Mux

[31]: 10

同步复位

IM = 4096字

2¹⁰ = 1024

2¹² = 4096 [13:2]

DM = 3072字

0 - 3000 = 00... 0000

0 - 3004 = 0... 0004

00000004

00000000

jr: 其余 = 0. nPCsel = 11

jal: RegWrite = 1 WhichToReg = { 0 ALU, 1 Mem, 2 PC+4 }

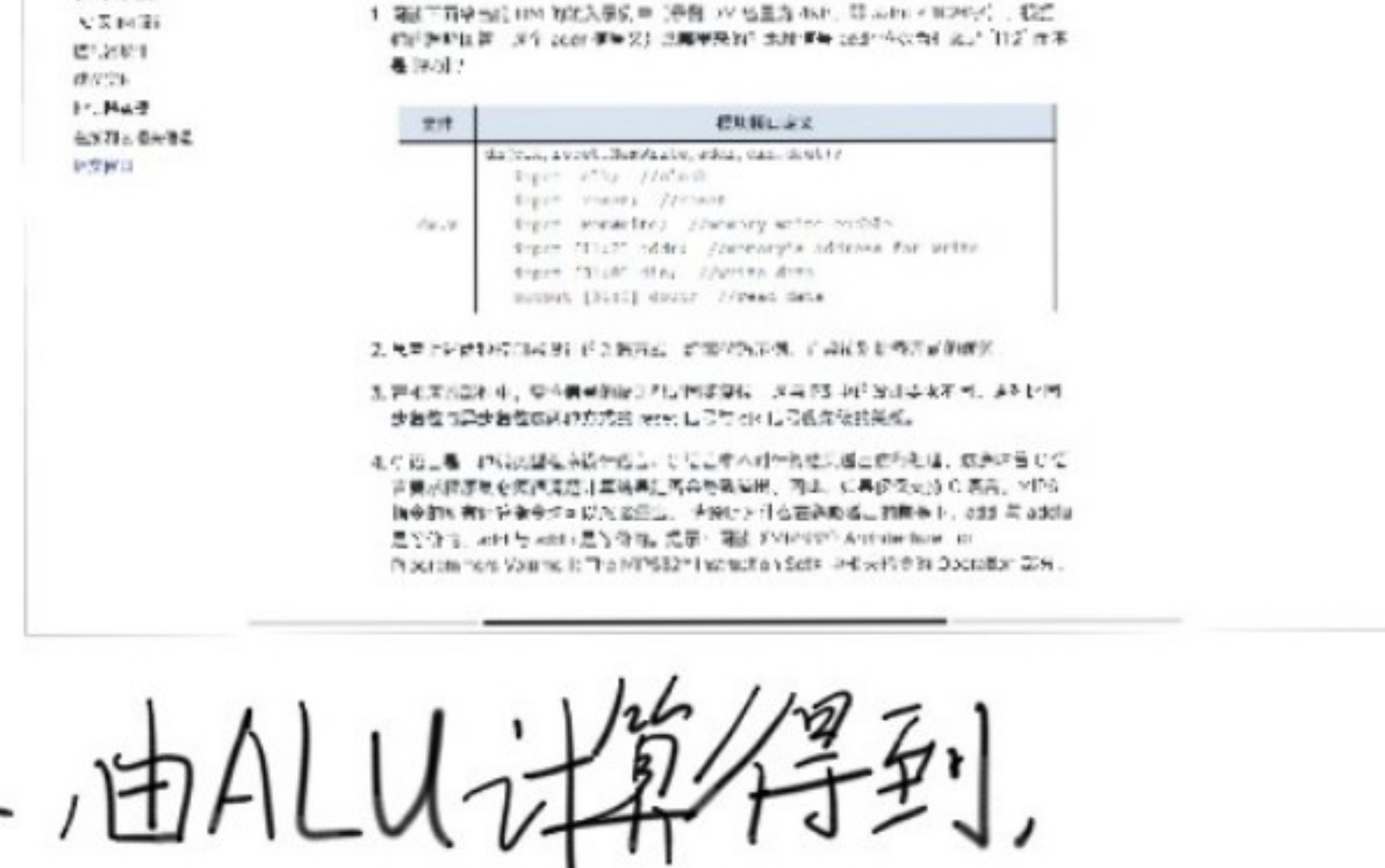
Regdst = { 0 rt, 1 rd, 2 31 }

nPCsel = 10

其余 0.

MUX: 3个

P4 设计文档:



1. addr为寄存器的地址, 由ALU计算得到, 32bit x 1024字, 由于CPU是32位, 一次处理4个字节, 地址应为4的整数倍, ... 0/00代表寄存器堆[1]位置, 共1024 = 2¹⁰个寄存器, 需10位地址, 即[11:2]即可

2. 指令对应的控制信号如何取值: 本人所用此方法
if (RegDst = 1, RegWrite = 1, RegWriteSel = 0, MemWrite = 0, nPCSel = 0, ExtOp = 0, ALUControl = 0, ALUSrc = 0), Command = ADD
if (RD = 0, RW = 1, RWS = 0, MW = 0, NPCS = 0, ExtOp = 0, ALUC = 2, ALUSrc = 1), Command = ORI

控制信号取值对应的指令: 转置即可 ADD RD RW ...

ADD ORI ... ORI 0

RD 1 0

RW 1 1

3. 同步: clk优先级高

异步: reset优先级高

4.

31	25	21	17	13	9	5	1	0
OP	OP	OP	OP	OP	OP	OP	OP	OP
OP	OP	OP	OP	OP	OP	OP	OP	OP

31	25	21	17	13	9	5	1	0
OP	OP	OP	OP	OP	OP	OP	OP	OP
OP	OP	OP	OP	OP	OP	OP	OP	OP

二者计算上无差别, 带符号其实与符号无关, 仅判断是否溢出

设计草稿: 前面有一部分译码

数据通路同p3,

