

CO - process1 - Datapath Design

涂珂 2011011273
傅左右 2011011264
计 14 - 402 组

November 18, 2013

指令集任务

设计方案

方案细节

指令集任务

THCO MIPS 基本指令集

序号	指令	序号	指令
1	ADDIU	14	LW_SP
2	ADDIU3	15	MFIH
3	ADDSP	16	MFPC
4	ADDU	17	MTIH
5	AND	18	MTSP
6	B	19	NOP
7	BEQZ	20	OR
8	BNEZ	21	SLL
9	BTEQZ	22	SRA
10	CMP	23	SUBU
11	JR	24	SW
12	LI	25	SW_SP
13	LW		

扩展指令集 (402)

- ▶ JRRA
- ▶ SLTI
- ▶ ADDSP3
- ▶ NOT
- ▶ SLT

设计方案

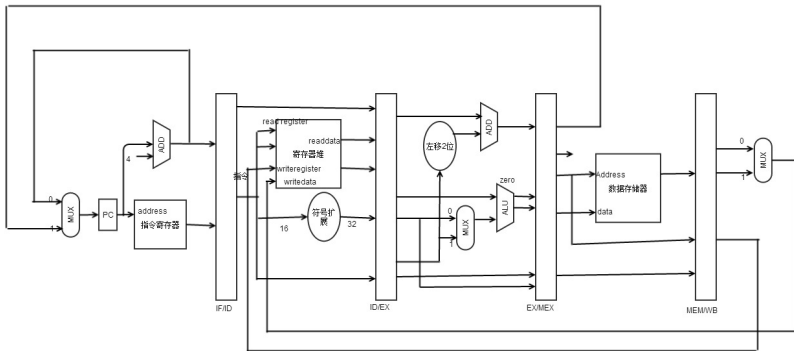


Figure: 一个简单、初步的数据通路设计图

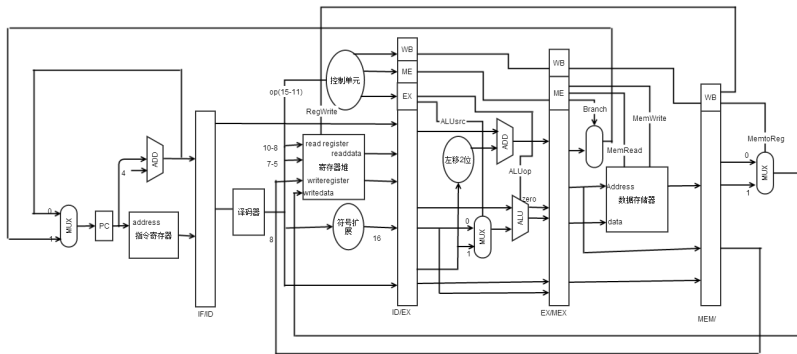


Figure: 今天半夜写的一个稍稍更复杂的数据通路设计图 > <

方案选择

由指令的功能和计算机硬件结构可将指令的执行分为 5 个步骤。(IF、ID、EXE、MEM、WB)

虽然流水线结构和设计略复杂，但是流水线效率更高，也已经是当前 CPU 中被普遍使用的方案。所以我们选择用流水线的方案来实现 CPU。(在上图中暂时还没有加入对冲突的处理)

根据 4 个流水线寄存器（图中最大的矩形）将流水线分为 5 个部分：

1. IF: 取指令、形成下条指令的地址
2. ID: 指令译码，读寄存器堆（准备数据）
3. EXE: 指令执行，地址计算
4. MEM: 数据内存访问（存储器读写）
5. WB: 写回

指令和数据随执行过程从左到右依次从五个过程走过。每个时钟周期 5 个部分并行处理不同指令的对应部分。

每条指令对数据通路的操作

1. IF 取指令，PC 从内存中取指令，并将数据放入 IF/ID 寄存器。将 PC 中地址 +4 取下一条指令，但也可能会有跳转指令，所以有一个数据选择器选择。
2. ID 指令译码，其将数据存入 2 个读寄存器（可能只用一个，由指令决定），有的指令有立即数的输入，这里提供了一个符号扩展单元。将扩展后的数以及 2 个寄存器中数据，还有增加后的 PC 值写入 ID/EX 寄存器。
3. EX 将立即数左移两位进行加法运算，根据指令类型进行不同的 ALU 运算，结果放入 EX/MEM 寄存器。
4. MEM 根据得到的结果读取数据内存。写入 MEM/WB 寄存器。
5. WB 将得到的写入寄存器堆中。

需要解决的问题

- ▶ 数据通路
- ▶ 控制信号
- ▶ 执行时序

方案细节

CPU 设计

主要部件

- ▶ ALU

功能

- ▶ 数据运算：算术、逻辑、移位
- ▶ 数据移动
- ▶ 流程控制：转移、调用/返回、中断

分工

- ▶ Datapath：数据移动和计算
- ▶ Controller：控制整个过程

指令设计

- ▶ 用前 5 位表示 op。共 30 条。
- ▶ 加 * 为扩展指令。
- ▶ XXX, YYY, ZZZ 为寄存器标号。
- ▶ III 为立即数。
- ▶ 把类型相近的 op 连续起来，这样写代码就可以用大于小于判断了。

R 型指令

R	指令结构
MFIH	00001XXX00000000
MFPC	00010XXX00000000
MTIH	00011XXX00000000
MTSP	00100XXX00000000
AND	00101XXXYYY00000
OR	00110XXXYYY00000
*NOT	00111XXXYYY00000
*SLT	01000XXXYYY00000
CMP	01001XXXYYY00000
SLL	01010XXXYYYIII00
SRA	01011XXXYYYIII00
ADDU	01100XXXYYYZZZ00
SUBU	01101XXXYYYZZZ00

I 型指令

I	指令结构
ADDSP	01110IIIIIIII000
LW_SP	01111XXX00000000
ADDIU	10000XXXIIIIIIII
*SLTI	10001XXXIIIIIIII
*ADDSP3	10010XXXIIIIIIII
LI	10011XXXIIIIIIII
ADDIU3	10100XXXYYY0IIII
LW	10101XXXYYYIIIIII
SW	10110XXXYYYIIIIII
SW_SP	10111XXXYYYIIIIII

B 型指令

B	指令结构
---	------

B	11000IIIIIIIIII
---	-----------------

BTEQZ	11001IIIIIIII000
-------	------------------

BEQZ	11010XXXIIIIIIII
------	------------------

BNEZ	11011XXXIIIIIIII
------	------------------

J, R 型指令

J 型指令

J	指令结构
*JRRA	1110000000000000
JR	11101XXX00000000

NOP 指令

NOP	0000000000000000
-----	------------------

控制信号设计

信号名称	设 0 事件	设 1 事件	作用阶段
Reg-Write	无	将数据写入到寄存器输出对应的寄存器 ID	ID
ALU-Src	第二个 ALU 操作数来自寄存器堆的第二个输出	第二个 ALU 操作数符号扩展后的立即数	EXE
PC-Src	$PC += 4$	PC 为分支计算器的输出	IF
Mem-Read	无	输入地址对应数据的内存放到数据输出	MEM
Mem-Write	无	输入地址对应数据内存放到数据的输入	MEM
Mem-Reg	ALU 提供寄存器写的输入值	数据内存提供寄存器写数据的输入值	WB

最后回过头来再看看...

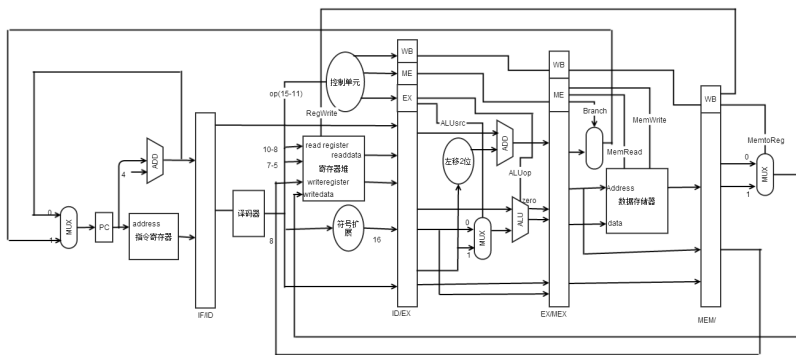


Figure: 一个稍稍更复杂的数据通路设计图 > <

The End. Thank you :P!