单周期与流水线MIPS-CPU

MIPS指令格式：

日程表

描述已自动生成 表格

描述已自动生成图片包含 门, 游戏机, 建筑, 钟表

描述已自动生成

表格

描述已自动生成

表格

描述已自动生成

表格

描述已自动生成

支持的指令

算术运算：

add/sub/ addu/subu/andi

逻辑计算：

and/or/ nor/lui / xor / ori

移位指令

slt/sltu/ sll/ slti/sra/srav/ srl/sllv/srlv

访存指令

lb/lh/lbu/lhu/sb/sh/lw/sw

分支指令

Beq/ bne

跳转指令

j/jal /jr/jalr

* **Add : R[rd] = R[rs] + R[rt]**
* **Subtract : R[rd] = R[rs] - R[rt]**
* **And: R[rd] = R[rs] & R[rt]**
* **Or : R[rd] = R[rs] | R[rt]**
* **SLT: R[rd] = 1 if R[rs] <  R[rt] else 0**
* **Jr: PC=R[rs]**
* **Lw: R[rt] = M[R[rs]+SignExtImm]**
* **Sw : M[R[rs]+SignExtImm] = R[rt]**
* **Beq : if(R[rs]==R[rt]) PC=PC+1+BranchAddr**
* **Addi: R[rt] = R[rs] + SignExtImm**
* **J : PC=JumpAddr**
* **Jal : R[7]=PC+2;PC=JumpAddr**
* **SLTI: R[rt] = 1 if R[rs] < imm else 0**

**SignExtImm = { 9{immediate[6]}, imm**

**JumpAddr =    { (PC+1)[15:13], address}**

**BranchAddr = { 7{immediate[6]}, immediate, 1’b0 }**

整体框图：

1.单周期MIPS-CPU

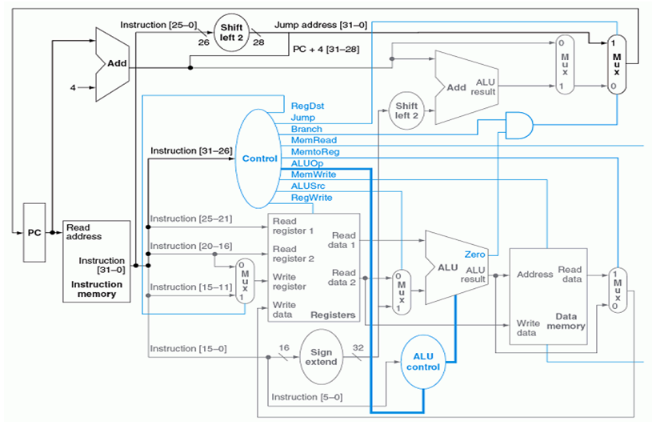
单周期 CPU 总体结构：MIPS 单周期 CPU 可以简单地划分为五个阶段，对应着上面已经提到的五个关键的模块：IF 对应于 PC，ID 阶段对应于 IM（指令寄存器）和 RegisterFile（寄存器组）， EXE 对应于 ALU，MEM 阶段对应于 DataMemory，WB 阶段对应于 RegisterFile。在数据通路中，除了这些模块之外，还需要完成具体指令的控制信号，控制信号生成集成在 CTRLUNIT（控制单元）， 此外，不同控制信号的选择控制具体实现还需要依赖于数据选择器（MUX）。当然，立即数的拓展不能离开数据拓展器（EXT）。以上所提到的所有模块，包括数据通路和两个存储器（IM 和 DM），最终构成一个较为完善的 MIPS 单周期 CPU 通路。

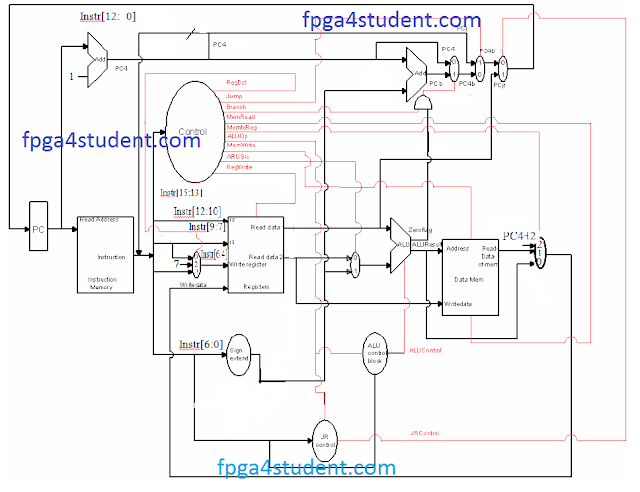
图示, 示意图

描述已自动生成

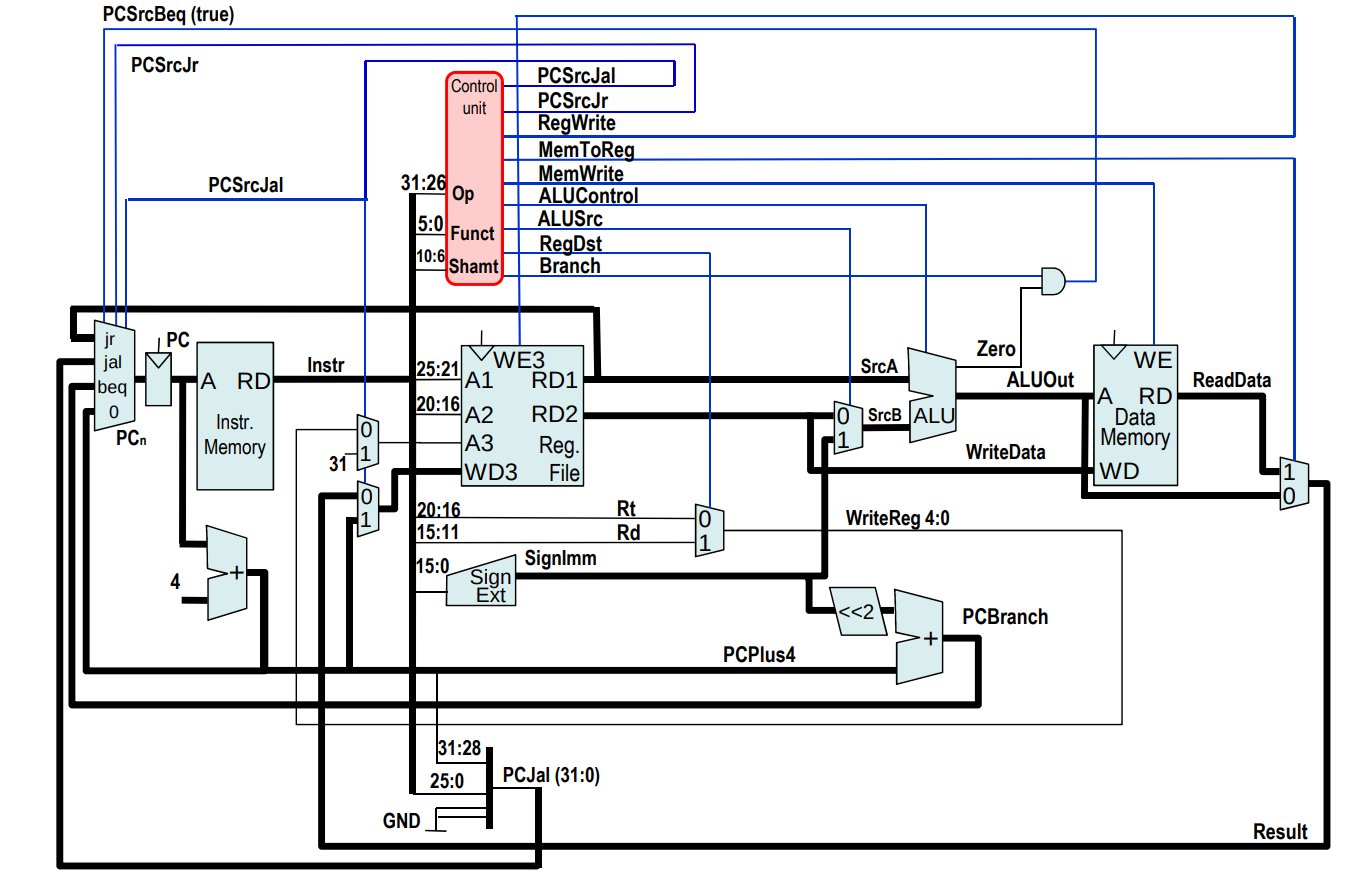
图示, 示意图

描述已自动生成



 图示, 示意图

描述已自动生成



图示, 示意图

描述已自动生成

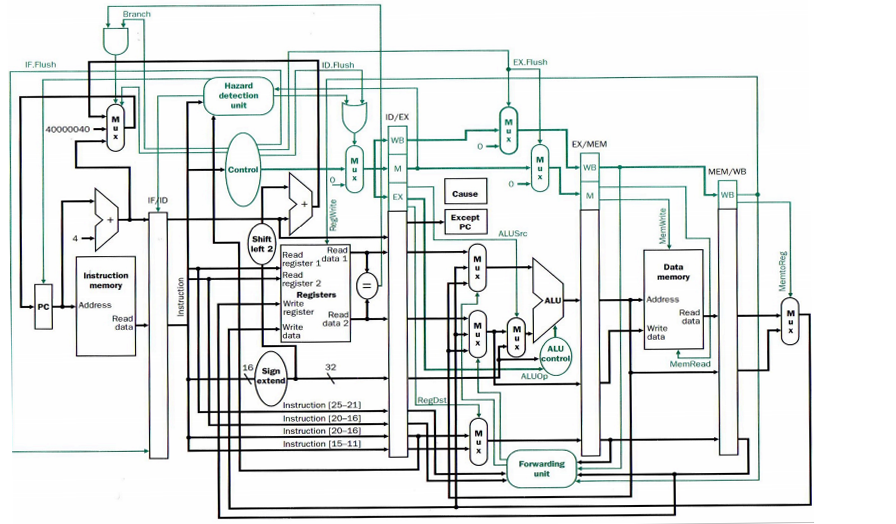
PC:

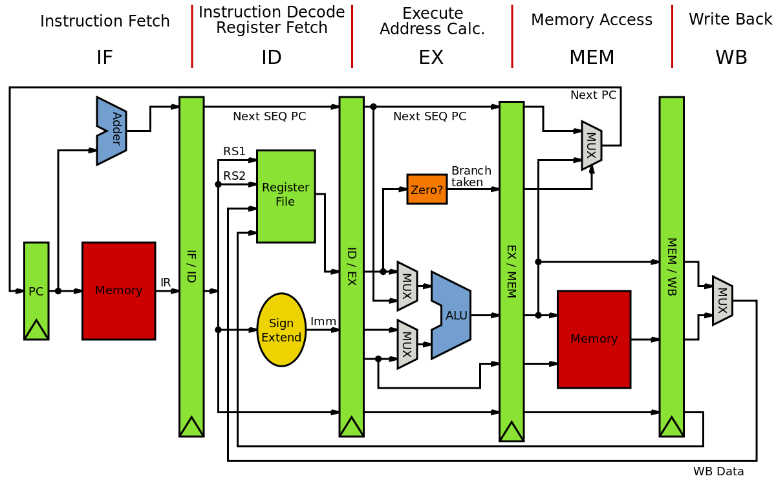
图示

描述已自动生成

文本

描述已自动生成





图示, 示意图

描述已自动生成

表格

描述已自动生成

output reg[1:0] reg\_dst,mem\_to\_reg,alu\_op,

output reg jump,branch,mem\_read,mem\_write,alu\_src,reg\_write,sign\_or\_zero

ALU:

|  |  |  |
| --- | --- | --- |
| ALUOp | 操作 | 描述 |
| 0000 | ALU\_NOP | 无操作，C = A |
| 0001 | ALU\_ADD | 加法操作 |
| 0010 | ALU\_SUB | 减法操作 |
| 0011 | ALU\_AND | 逻辑与操作 |
| 0100 | ALU\_OR | 逻辑或操作 |
| 0101 | ALU\_SLT | 小于则置位 |
| 0110 | ALU\_SLTU | 小于则置位，无符号数 |
| 0111 | ALU\_NOR | 逻辑或非操作 |
| 1000 | ALU\_SLL | 向左移位操作 |
| 1001 | ALU\_SRL | 向右移位操作 |
| 1010 | ALU\_SRA | 向右算术移位操作 |
| 1011 | ALU\_SLLV | 逻辑可变左移操作 |
| 1100 | ALU\_SRLV | 逻辑可变左移操作 |
| 1101 | ALU\_SLL16 | 向做移位（16 位）操作，使用与指令 LUI |
| 1110 | ALU\_XOR | 逻辑异或操作 |
| 1111 | ALU\_SRAV | 算数可变右移操作 |

Control unit

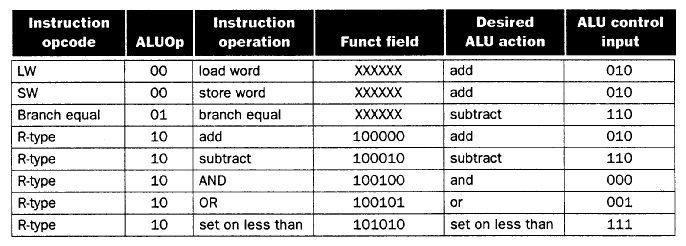
文本

描述已自动生成

表格

描述已自动生成

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Instruction** | **[31:26]** | **[25:21]** | **[20:16]** | **[15:11]** | **[10:6]** | **[5:0]** | **NPCSel** | **ALUASrc** | **ALUBSrc** | **ALUOp** | **RegDst** | **MemtoReg** | **RegWrite** | **MemWrite** | **ExtSel** | **Load** | **Store** | **Instruction operation** |
| **R-type** | **op** | **rs** | **rt** | **rd** | **Shamt(sa)** | **func** | **PC更行值选择**  **00：NPC=PC＋4**  **01：NPC=PC+4+偏移\*4( bne、beq )**  **10：NPC=(PC+4)[31:28]+IMM[25:0]+2’b00 (jr, jal)**  **11：NPC=RsData，(JR，JLR)** | **ALU端口A的源操作数**  **0: reg\_busA(Rs)**  **1: 立即数sa** | **ALU的端口B源操作数**  **0: reg\_busB(Rt)**  **1: sign-extend imm16** | **ALU运算操作** | **写寄存器地址**  **0: rd**  **1: rt** | **存储器写回寄存器的值**  **0: alu\_result**  **1: DataMem** | **寄存器写操作**  **0: unable**  **1: enable** | **存储器写操作**  **0: unable**  **1: enable** | **拓展控制**  **0: zero-extend**  **1: sign-extend** | **000：lw，加载全字**  **001：lb，加载字节**  **010：lbu，加载字节（无符号）**  **011：lh，加载半字**  **100：lhu，加载半字（无符号）**  **101-111：暂未设置** | **00：sw，存储全字**  **01：sb，存储字节**  **10：sh，存储半字**  **11：暂未设置** | **指令描述** |
| **add** | **000000** | **rs** | **rt** | **rd** | **00000** | **100000** | **00** | **0** | **0** | **0001** | **0** | **0** | **1** | **0** | **0** |  |  | **(rd)←(rs)+(rt);** |
| **addu** | **rs** | **rt** | **rd** | **00000** | **100001** | **00** | **0** | **0** | **0001** | **0** |  |  |  | **0** |  |  |  |
| **sub** | **rs** | **rt** | **rd** | **00000** | **100010** | **00** | **0** | **0** | **0010** | **0** |  |  |  | **0** |  |  |  |
| **subu** | **rs** | **rt** | **rd** | **00000** | **100011** | **00** | **0** | **0** | **0010** | **0** |  |  |  | **0** |  |  |  |
| **and** | **rs** | **rt** | **rd** | **00000** | **100100** | **00** | **0** | **0** | **0011** | **0** |  |  |  | **0** |  |  |  |
| **or** | **rs** | **rt** | **rd** | **00000** | **100101** | **00** | **0** | **0** | **0100** | **0** |  |  |  | **0** |  |  |  |
| **xor** | **rs** | **rt** | **rd** | **00000** | **100110** | **00** | **0** | **0** | **1110** | **0** |  |  |  | **0** |  |  |  |
| **nor** | **rs** | **rt** | **rd** | **00000** | **100111** | **00** | **0** | **0** | **0111** | **0** |  |  |  | **0** |  |  |  |
| **slt** | **rs** | **rt** | **rd** | **00000** | **101010** | **00** | **0** | **0** | **0101** | **0** |  |  |  | **0** |  |  |  |
| **sltu** | **rs** | **rt** | **rd** | **00000** | **101011** | **00** | **0** | **0** | **0110** | **0** |  |  |  | **0** |  |  |  |
| **sll** | **00000** | **rt** | **rd** | **sa** | **000000** | **00** | **1** | **0** | **1000** | **0** |  |  |  | **0** |  |  |  |
| **srl** | **00000** | **rt** | **rd** | **sa** | **000000** | **00** | **1** | **0** | **1001** | **0** |  |  |  | **0** |  |  |  |
| **sra** | **00000** | **rt** | **rd** | **sa** | **000000** | **00** | **1** | **0** | **1010** | **0** |  |  |  | **0** |  |  |  |
| **sllv** | **rs** | **rt** | **rd** | **00000** | **000100** | **00** | **0** | **0** | **1000** | **0** |  |  |  | **0** |  |  |  |
| **srlv** | **rs** | **rt** | **rd** | **00000** | **000110** | **00** | **0** | **0** | **1100** | **0** |  |  |  | **0** |  |  |  |
| **srav** | **rs** | **rt** | **rd** | **00000** | **000111** | **00** | **0** | **0** | **1111** | **0** |  |  |  | **0** |  |  |  |
| **jr** | **rs** | **00000** | **00000** | **00000** | **001000** | **11** | **0** | **0** | **xxxx** | **x** |  |  |  |  |  |  |  |
| **jalr** | **rs** | **00000** | **rd** | **00000** | **001001** | **11** | **0** | **0** | **xxxx** | **x** |  |  |  |  |  |  |  |
| **I-type** | **op** | **rs** | **rt** | **immediate/offset** | | |  |  |  |  |  |  |  |  |  |  |  |  |
| **addi** | **001000** | **rs** | **rt** | **immediate** | | | **00** | **0** | **1** | **0001** | **1** | **11** | **1** | **0** |  |  |  |  |
| **addiu** | **001001** | **rs** | **rt** | **immediate** | | | **00** | **0** | **1** | **0001** | **1** |  | **1** |  |  |  |  |  |
| **andi** | **001100** | **rs** | **rt** | **immediate** | | | **00** | **0** | **1** | **0011** | **1** |  | **1** |  |  |  |  |  |
| **ori** | **001101** | **rs** | **rt** | **immediate** | | | **00** | **0** | **1** | **0100** | **1** |  | **1** |  |  |  |  |  |
| **xori** | **001110** | **rs** | **rt** | **immediate** | | | **00** | **0** | **1** | **1110** | **1** |  | **1** |  |  |  |  |  |
| **slti** | **001010** | **rs** | **rt** | **immediate** | | | **00** | **0** | **1** | **0101** | **1** |  | **1** |  |  |  |  |  |
| **sltiu** | **001011** | **rs** | **rt** | **immediate** | | | **00** | **0** | **1** | **0101** | **1** |  | **1** |  |  |  |  |  |
| **lui** | **001111** | **00000** | **rt** | **immediate** | | | **00** | **0** | **1** | **1101** | **1** |  | **1** |  |  |  |  |  |
| **lb** | **100000** | **base** | **rt** | **offset** | | | **00** | **0** | **1** | **0001** | **1** |  | **1** |  |  |  |  |  |
| **lh** | **100001** | **base** | **rt** | **offset** | | | **00** | **0** | **1** | **0001** | **1** | **11** | **1** | **0** |  |  |  |  |
| **lw** | **100011** | **base** | **rt** | **offset** | | | **00** | **0** | **1** | **0001** | **1** |  | **1** |  |  |  |  |  |
| **lbu** | **100100** | **base** | **rt** | **offset** | | | **00** | **0** | **1** | **0001** | **1** |  | **1** |  |  |  |  |  |
| **lhu** | **100101** | **base** | **rt** | **offset** | | | **00** | **0** | **1** | **0001** | **1** | **11** | **1** | **1** |  |  |  |  |
| **sb** | **101000** | **base** | **rt** | **offset** | | | **00** | **0** | **1** | **0001** | **x** |  | **1** |  |  |  |  |  |
| **sh** | **101001** | **base** | **rt** | **offset** | | | **00** | **0** | **1** | **0001** | **x** |  | **0** |  |  |  |  |  |
| **sw** | **101011** | **base** | **rt** | **offset** | | | **00** | **0** | **1** | **0001** | **x** |  | **xx** |  |  |  |  |  |
| **beq** | **000100** | **rs** | **rt** | **offset** | | | **00/01** | **0** | **0** | **0010** | **x** | **01** | **xx** | **0** |  |  |  |  |
| **bne** | **000101** | **rs** | **rt** | **offset** | | | **01/00** | **0** | **0** | **0010** | **x** | **10** | **0** | **0** |  |  |  |  |
| **J-type** | **op** | **address** | | | | |  |  |  |  |  |  |  |  |  |  |  |  |
| **j** | **000010** | **instr\_index** | | | | | **10** | **x** | **x** | **xxxx** | **x** | **0** | **0** | **00** |  |  |  |  |
| **jal** | **000100** | **instr\_index** | | | | | **10** | **x** | **x** | **xxxx** | **x** | **PC+4** | **1** | **0** |  |  |  |  |



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ALU Control** | | | | |
| **ALU op** | **Function** | **ALUcnt** | **ALU Operation** | **Instruction** |
| **11** | **xxxx** | **000** | **ADD** | **Addi,lw,sw** |
| **01** | **xxxx** | **001** | **SUB** | **BEQ** |
| **00** | **00** | **000** | **ADD** | **R-type: ADD** |
| **00** | **01** | **001** | **SUB** | **R-type: sub** |
| **00** | **02** | **010** | **AND** | **R-type: AND** |
| **00** | **03** | **011** | **OR** | **R-type: OR** |
| **00** | **04** | **100** | **slt** | **R-type: slt** |
| **10** | **xxxxxx** | **100** | **slt** | **i-type: slti** |