

1、CPU 内部结构如图所示，假如该机有一个寄存器硬堆栈是\$30。指令系统由 push 、 pop 和 mov 三条指令组成，功能如下：

```

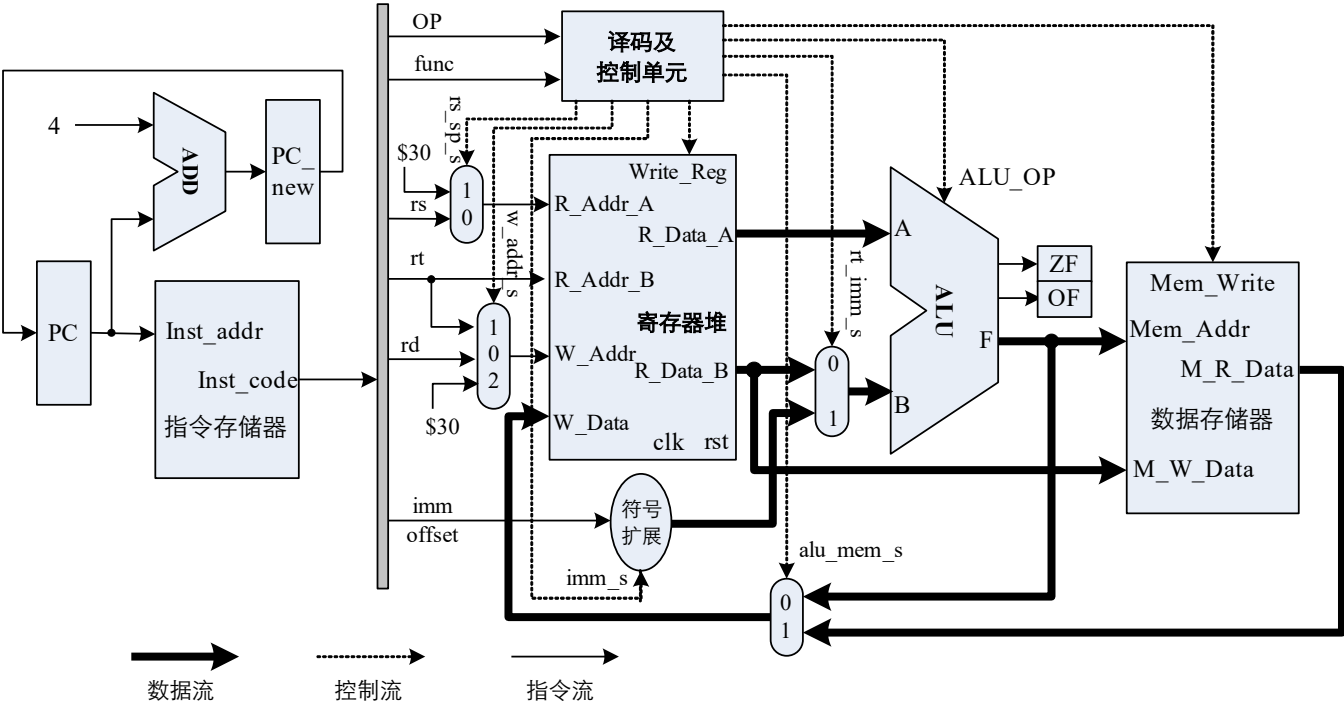
push rs      #将 rs 寄存器内容入栈，堆栈地址是$30
pop rd       #将堆栈内容读出，写入 rd 寄存器，堆栈地址是$30
mov rd,rs    # (rs)→rd
    
```

（1）如果 push 、 pop 和 mov 指令都采用 R 指令格式，它们的 func 字段分别是 111110、111111 和 100000，请将下表补充完整

助记符	指令格式（位）					
	OP(31..26)	rs(25..21)	rt(20..16)	rd(15..11)	shamt(10..6)	func(5..0)
PUSH						
POP						
MOV						

（2）对照 CPU 结构图，分析每条指令的数据通路，填表写出指令对应的控制信号

指令	rs_sp_s	w_addr_s	imm_s	rt_imm_s	alu_mem_s	ALU_OP P (3 位)	Write_Reg	Mem_Write e
PUSH								
POP								
MOV								



(5) 如果采用微程序控制器, 而且是直接控制法, 无判别测试字段, 下址字段 10 位, 请设计微指令格式和控存容量。

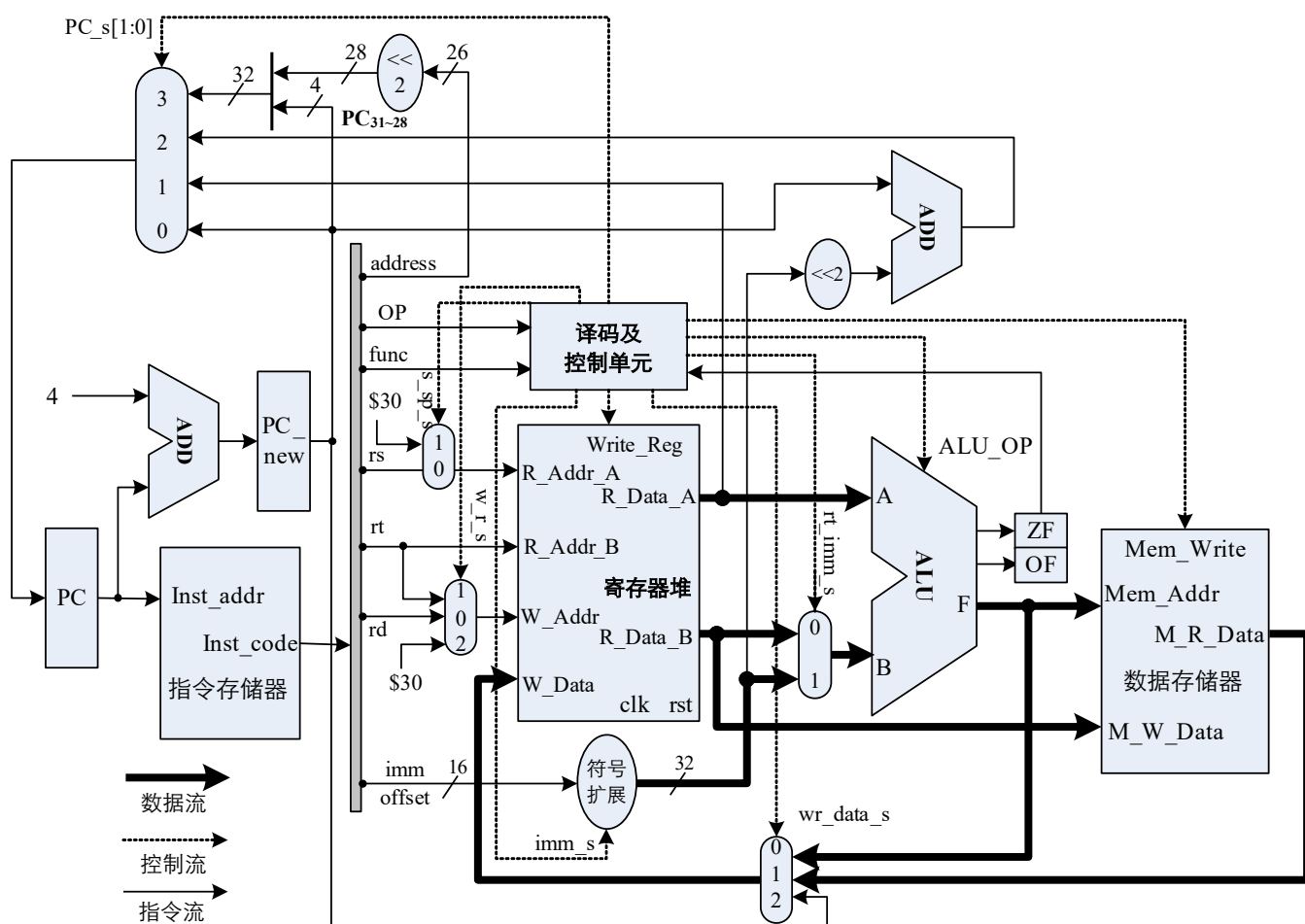
push rs	#将 rs 寄存器内容入栈，堆栈地址是\$30
pop rd	#将堆栈内容读出，写入 rd 寄存器，堆栈地址是\$30
mov rd,rs	# (rs)→rd
add rd,rs,rt	#(rs)+(rt)→rd
jr rs	#无条件跳转 rs→PC
beq rs,rt,label	#相等转移 if(rs=rt) PC+4+offset×4→PC
j label	#无条件跳转{(PC+4)高 4 位, address,0,0}→PC

助记符	指令格式（位）					
	OP(31..26)	rs(25..21)	rt(20..16)	rd(15..11)	shamt(10..6)	func(5..0)
push						
pop						
mov						
add						
jr						

[illegible]

add							0	0	01
jr									
beq				0		101	0	0	00/10
j							0	0	11

(3) 在 MIPS 指令集中增加一条 swap 指令有两种做法，（swap rs,rt #指令的功能是将 rs 和 rt 所指的两个寄存器的内容互换）。第一种做法是修改硬件来实现 swap 指令，第二种做法是采用伪指令方式（软件方式），即用指令系统中已有的指令构成指令序列来代替实现。请基于本题的指令系统，写出用伪指令方式实现 swap\$t0,\$t1 的指令序列。（提示：伪指令对应的指令序列中不能使用其他额外的寄存器，以免破坏这些寄存器原有的内容）



指令	s_sp_s	w_r_s	imm_s	rt_imm_s	wr_data_s	ALU_OP	Write_Reg	Mem_Write	PC_s
push									
pop									
mov									
add							0	0	01
jr									
beq				0		101	0	0	00/10
j							0	0	11

(4)