

- Y86-64 指令学习

- popq 与 pushq

阶段	pushq rA	popq rA
取指	$\text{icode: ifun} \leftarrow M_1[\text{PC}]$ $\text{rA:rB} \leftarrow M_1[\text{PC}+1]$ $\text{valP} \leftarrow \text{PC}+2$	$\text{icode: ifun} \leftarrow M_1[\text{PC}]$ $\text{rA:rB} \leftarrow M_1[\text{PC}+1]$ $\text{valP} \leftarrow \text{PC}+2$
译码	$\text{valA} \leftarrow R[\text{rA}]$ $\text{valB} \leftarrow R[\%rsp]$	$\text{valA} \leftarrow R[\%rsp]$ $\text{valB} \leftarrow R[\%rsp]$
执行	$\text{valE} \leftarrow \text{valB} + (-8)$	$\text{valE} \leftarrow \text{valB} + 8$
访存	$M_8[\text{valE}] \leftarrow \text{valA}$	$\text{valE} \leftarrow M_8[\text{valA}]$
写回	$R[\%rsp] \leftarrow \text{valE}$	$R[\%rsp] \leftarrow \text{valE}$ $R[\text{rA}] \leftarrow \text{valM}$
更新 PC	$\text{PC} \leftarrow \text{valP}$	$\text{PC} \leftarrow \text{valP}$

Y86-64 指令 pushq 和 popq 在顺序实现中的计算。这些指令将值压入或弹出

pushq 执行阶段需要用 ALU 将指针减 8，也就是内存写的地址。

- 控制转移指令：jxx、call 和 ret

阶段	jXX Dest	call Dest	ret
取指	$\text{icode: ifun} \leftarrow M_1[\text{PC}]$ $\text{valC} \leftarrow M_8[\text{PC}+1]$ $\text{valP} \leftarrow \text{PC}+9$	$\text{icode: ifun} \leftarrow M_1[\text{PC}]$ $\text{valC} \leftarrow M_8[\text{PC}+1]$ $\text{valP} \leftarrow \text{PC}+9$	$\text{icode: ifun} \leftarrow M_1[\text{PC}]$ $\text{valP} \leftarrow \text{PC}+1$
译码		$\text{valB} \leftarrow R[\%rsp]$	$\text{valA} \leftarrow R[\%rsp]$ $\text{valB} \leftarrow R[\%rsp]$
执行	$\text{Cnd} \leftarrow \text{Cond}(\text{CC}, \text{ifun})$	$\text{valE} \leftarrow \text{valB} + (-8)$	$\text{valE} \leftarrow \text{valB} + 8$
访存		$M_8[\text{valE}] \leftarrow \text{valP}$	$\text{valM} \leftarrow M_8[\text{valA}]$
写回		$R[\%rsp] \leftarrow \text{valE}$	$R[\%rsp] \leftarrow \text{valE}$
更新 PC	$\text{PC} \leftarrow \text{Cnd?valC:valP}$	$\text{PC} \leftarrow \text{valC}$	$\text{PC} \leftarrow \text{valM}$

图 4-21 Y86-64 指令 jXX、call 和 ret 在顺序实现中的计算。这些指令导致控制转移

- SEQ 硬件结构

- 硬件单元与各个处理阶段相关联：取指、译码、执行、访存、写回、更新 PC

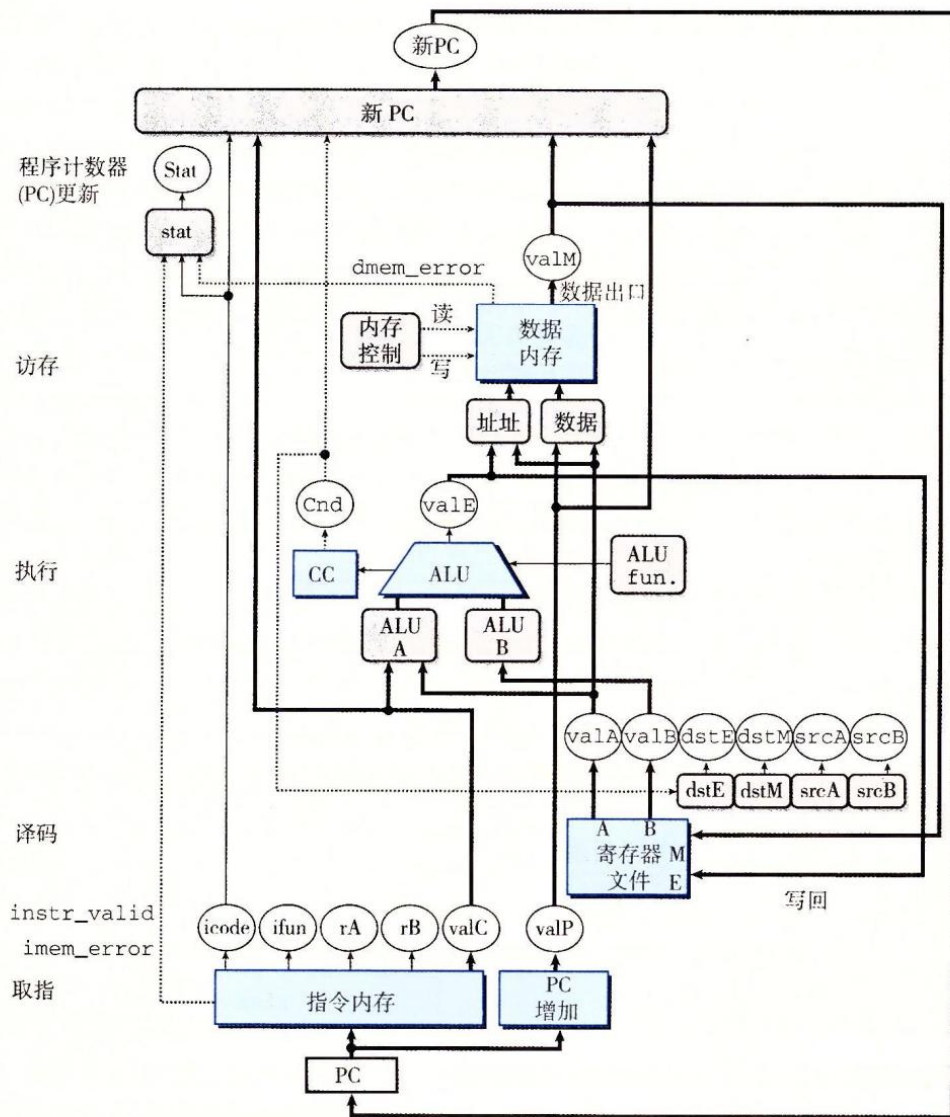


图 4-23 SEQ 的硬件结构，一种顺序实现。有些控制信号以及寄存器和控制字连接没有画出来