

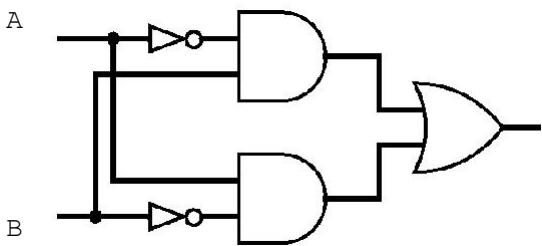
ICS 第四次小班课习题答案

【体系结构基础】

1. 下列描述更符合（早期）RISC 还是 CISC？

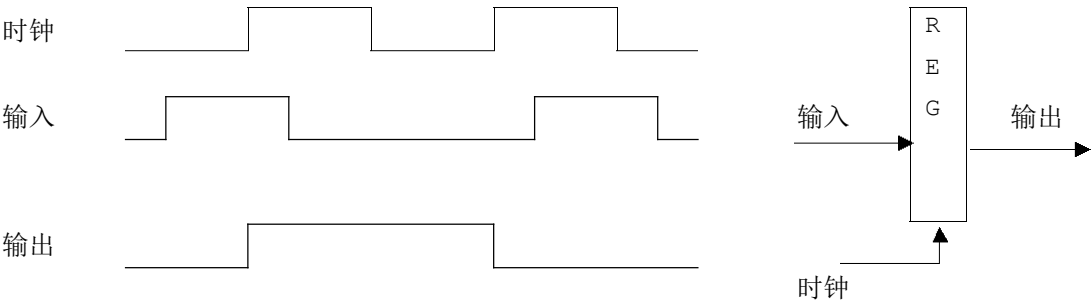
	描述	RISC	CISC
(1)	指令机器码长度固定	√	
(2)	指令类型多、功能丰富		√
(3)	不采用条件码	√	
(4)	实现同一功能，需要的汇编代码较多	√	
(5)	译码电路复杂		√
(6)	访存模式多样		√
(7)	参数、返回地址都使用寄存器进行保存	√	
(8)	x86-64		√
(9)	MIPS	√	
(10)	广泛用于嵌入式系统	√	
(11)	已知某个体系结构使用 <code>add R1,R2,R3</code> 来完成加法运算。当要将数据从寄存器 <code>S</code> 移动至寄存器 <code>D</code> 时，使用 <code>add S,#ZR,D</code> 进行操作（ <code>#ZR</code> 是一个恒为 0 的寄存器），而没有类似于 <code>mov</code> 的指令。	√	
(12)	已知某个体系结构提供了 <code>xlat</code> 指令，它以一个固定的寄存器 <code>A</code> 为基地址，以另一个固定的寄存器 <code>B</code> 为偏移量，在 <code>A</code> 对应的数组中取出下标为 <code>B</code> 的项的内容，放回寄存器 <code>A</code> 中。		√

2. 写出下列电路的表达式



【答】 $(\neg A \wedge B) \vee (A \wedge \neg B)$

3. 下列寄存器在时钟上升沿锁存数据，画出输出的电平（忽略建立/保持时间）



【顺序处理器】

4. 根据 32 位 Y86-64 模型完成下表

		call	jXX
Fetch	icode, ifun	icode:ifun <- M1[PC]	icode:ifun <- M1[PC]
	rA, rB	\	
	valC	valC <- M8[PC+1]	valC <- M8[PC+1]
	valP	valP <- PC+9	valP <- PC+9
Decode	valA, srcA	\	\
	valB, srcB	valB <- R[%rsp]	\
Execute	valE	valE <- valB + -8	\
	Cond Code	\	Cnd <- Cond(CC, ifun)
Memory	valM	M8[valE] <- valP	\
Write back	dstE	R[%rsp] <- valE	\
	dstM	\	\
PC	PC	PC <- valC	PC <- Cnd?valC:valP

5. 已知 valC 为指令中的常数值, valM 为访存得到的数据, valP 为 PC 自增得到的值,

完成以下的 PC 更新逻辑:

```
int new_pc = [
    icode == ICALL : valC;
    icode == IJXX && Cnd: valC;
    icode == IRET : valM;
    1: valP;
]
```