1

- 延时和面积代价:流水线设计会在电路层面带来额外代价,随着K的不断增加,延时不断增加,电路面积不断增大,电路的关键路径将由 t_{cq} 和 t_{su} 主导,而非原本的 T_l ,实际电路的加速比是小于K的。
- 流水级不平衡:实际设计中,很难做到流水线的每一级都有完全一样的延时,一个K级流水线设计中,通常会存在一些流水级的延时大于原本延时的1/K,从而导致流水线加速比小于K。
- 流水线的预热: K级流水线实现K倍加速的重要基础之一是在执行足够多且连续的任务的条件下。流水线刚开始工作时,靠后的流水级处于闲置状态,没有满负荷工作,此时流水线处于预热状态,那么流水线的收益将会受到影响,即流水线加速比小于K。
- 流水线的冒险:对于K级流水线,前后相邻的K个任务会同时在流水线中执行,如果这些任务包含前后依赖关系,同时执行这些任务会导致计算出错,此时就需要增加电路或者阻塞流水线,这就会导致流水线加速比小于K。

2. a)

将该寄存器放入组合逻辑A中时,显然将其插入组合逻辑A的正中央,可以使关键路径最短,即关键路径的最小值为50+1=51ns。

此时延时也为最小,为 $(50+1) \times 2 = 102ns$ 。

2. b)

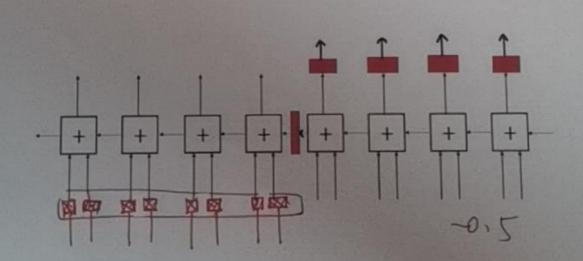
将寄存器均匀插入组合逻辑A,可以得到关键路径的最小值为10+1=11ns。 延时最小值为 $(10+1)\times 10=110ns$ 。

2. c)

由上面的计算可以看出,流水线设计不可以无限提高电路频率。继续对上面的问题进行分析,可以得到,当设计为N级流水线时,延时变为 $(100/N+1)\times N=(100+N)ns$,故而随着流水级的增加,电路延时不断增大,受到建立时间的影响越来越大。

3.

需要增加5个1-bit寄存器,修改如下图: (红色块为增加的1-bit寄存器,或者为了延长同时输出的时间的话,可以再在所有输出箭头上再加一层1-bit寄存器,这样就需要总共增加14个1-bit寄存器)



最高频率为1/20ns = 50MHz.

4. b)

最高频率不变,仍为1/20ns = 50MHz。

4. c)

PCI变为了原来的 $1/0.7\approx 1.43$,而时钟周期变为了原来的15ns/20ns=0.75,故而性能变为了原来的 $1/(1.43\times 0.75)\approx 0.933$,也就是性能大约降低了6.7%。

补充题

控制信号	解释	sw	addi	beq	J	jr	使用阶段
RegReadNum1	寄存器堆读取寄存器号1 (rs/rt/rd)	rs	rs	rs	×	rs	ID
RegReadNum2	寄存器堆读取寄存器号2 (rs/rt/rd)	rt	XX	rt	×	×	ID
RegWr	寄存器堆写入有效信号(0/1)	0	1	0	0	0	WB
ExtOp	立即数拓展单元配置Signed(1), Zero(0)	1	1	1	X	c×	ID
ALUSrc	从寄存器堆(0)或立即数(1)中选择第 二个ALU操作数	1	1	0	×	×	EX
ALUConf	ALU的功能 (AND/OR/COMP/ADD/SUB/)	ADD	ADD	COMP	×	×	EX
MemWr	数据内存写入有效(0/1)	1	0	0	0	0	MEM
MemRead	数据内存读取有效(0/1)	0	0	0	0	0	MEM
MemToReg	选择内存读取结果(1)或ALU结果(0) 写回寄存器堆	×	0	×	×	×	WB
Branch	选择PC+4(0)或者跳转地址(1)	0	0	1	1	1	X