東南大學

计算机组成原理

实验报告

学号：09021230 姓名：孙彦林

学号：_____ 姓名：_____

东南大学计算机科学与工程学院

二〇二二年九月

实验一　寄存器组的设计

一、实验内容

（1）测试 D 触发器的功能。

（2）设计并实现具有 1 个读端口、1 个写端口的 4×8 位寄存器组，验证其正确性。

二、电路设计与实现

（1）　测试 D 触发器的功能。

8 位 D 触发器使用 Quartus II 自带的 lpm 模块实现，其输入端包括数据端 DIn，使能端 WriteEnable，异步清零端 aCLR，异步置数端 aSET，时钟信号 CLK，输出端为 DOut。

设计如图 Fig 1 所示：

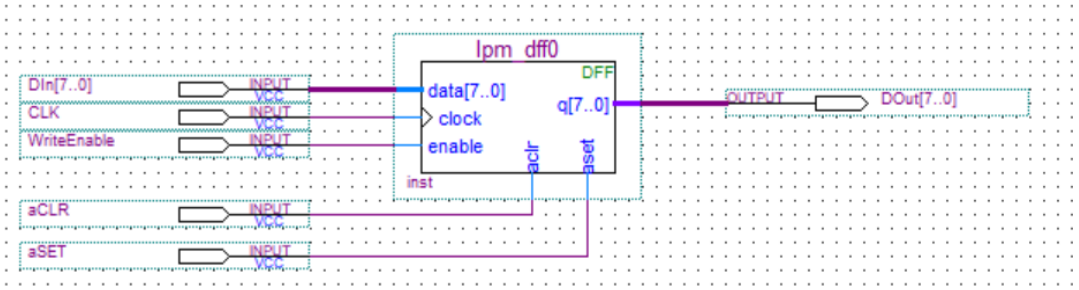


Fig 1 D 触发器的设计

（2）　设计并实现具有 1 个读端口、1 个写端口的 4×8 位寄存器组，验证其正确性。

将（1）中的元件进行打包，封装为 Dff_8Bit 八位 D 触发器元件。

使用 lmp_decode 对输入地址译码，通过其链接不同的 D 触发器的 WriteEnable 来选择所使用的 D 触发器。

D 触发器的其余端口均链接相应输入，aSET 因为不使用而接地。

使用 lpm_mux 来选择从哪个 D 触发器获得数据。

设计如图 Fig 2 所示：

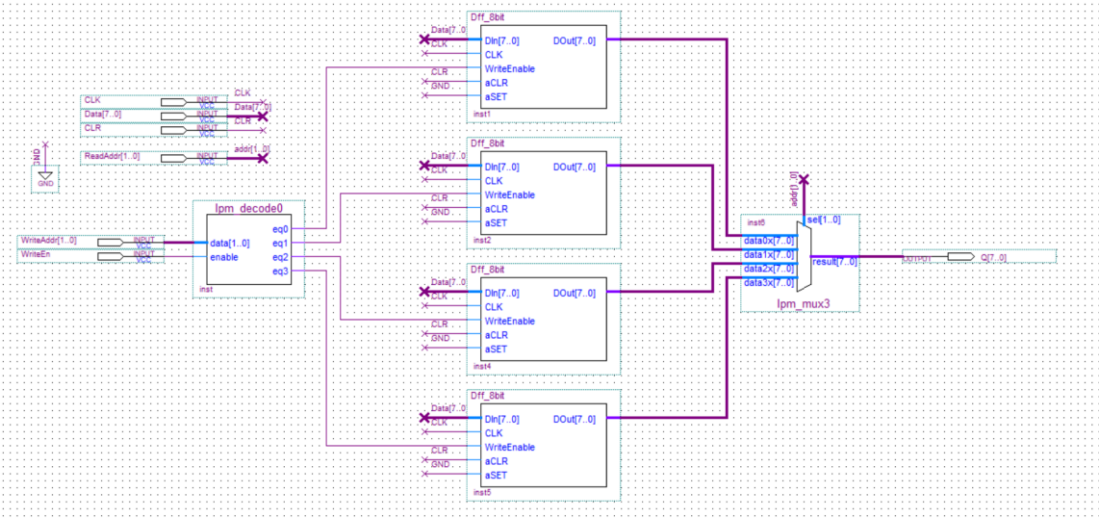


Fig 2 寄存器组设计

三、电路正确性验证

1、电路仿真

（1） D 触发器仿真

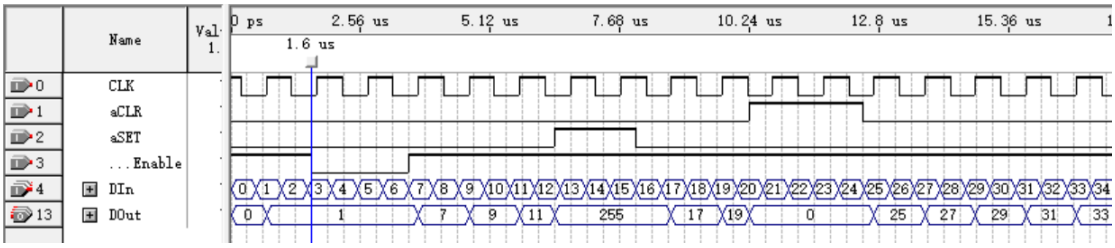


Fig 3 D 触发器仿真

（2） 寄存器组仿真

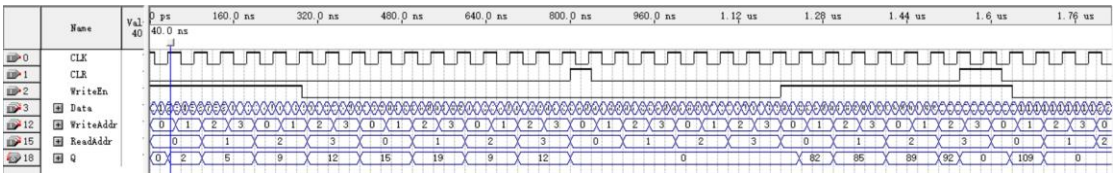


Fig 4 REGs 仿真

2、结果分析

（1） D 触发器

aSET 为异步置数端，它为 1 时所有的数据为 1，因此输出 255；aCLR 为异步清零端，为 1 时数据清零，因此输出为 0。WriteEnable 为使能端，为 1 时 D 触发器可以写入数据，为 0 时锁住数据不变。

1.6us 之前，元件在 CLK↑时写入 Data，之后 WriteEnable 为 0，数据在一段时间内不写入，当 WriteEnable 为 1 时重新开始写入；当 aSET 为 1 时，无论 WriteEnable 的值与 CLK 变化触发器直接置 1，显示为 255，aSET 为 0 后正常写入；当 aCLR 为 1 时强制置 0。

（2） 寄存器组

当 WriteEn 为 1 时，元件根据 WriteAddr 提供的地址向相应寄存器中存储 Data。CLR 为 1 时，所有寄存器强制归零。且 CLR 优先级高于 WriteEn。ReadAddr 来回往复四个地址便于观察各个寄存器中的值。

最开始 WriteEn 为 1，元件在 CLK↑时向相应寄存器中写入 Data，因此 40ns 输出 Q 产生变化。待所有寄存器写入数据后终止 WriteEn，以便检测写入数据是否正确。ReadAddr 往复一周展示所有寄存器的数据后，经过 CLR 清零，重新开始 WriteEn 写入，并在写入的过程中施加 CLR 以验证两者的优先级关系。经验证，CLR 和 WriteEn 同时为 1 时置 0，满足要求。

四、实验小结

- 工作分工：
  本实验由孙彦林独立完成，
- 设计总结：
  学习使用 lmp 部件可以更加快捷的构建部件
- 待改进之处：

Timing 仿真会出现大量毛刺，尚未找到良好解决方法

- 实验体会：

通过实验可以更好的理解数电中基本的一些模块是如何构成计算机的一部分的，能对计算机结构有更好的认识。

五、思考题

思考 1：若要求触发器的写操作在时钟周期结束时完成，Wr 与 Clk 的时序关系应如何设置？

应该在 CLK 的上升沿部分将 Wr 置为 1。

思考 2：设置信号取值时，状态变化时机与标尺对齐很容易实现（拖动鼠标），若要求寄存器的写操作在时钟周期结束时完成，哪些信号的状态变化应与标尺对齐？为什么？

CLK 的上升沿和相应寄存器的数值变化对齐。

输出变化和输出地址变化对齐

六、教师评语

教师签字：

日期：

实验二　ALU 的设计

一、实验内容

（1）测试加减法器的功能。

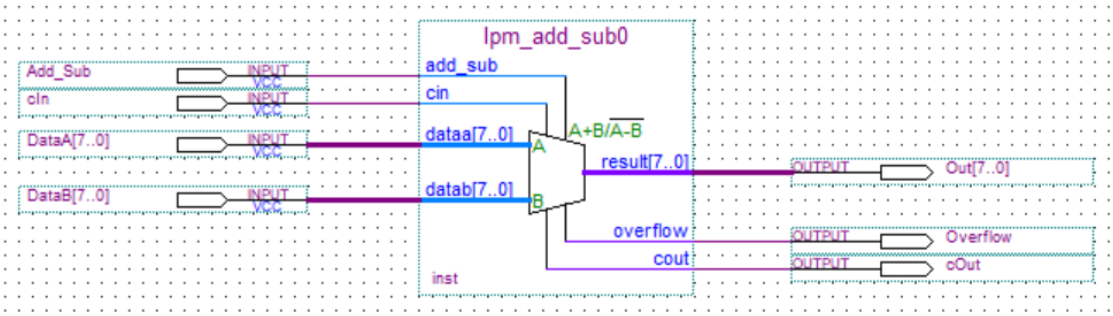
（2）设计并实现具有加法、减法、逻辑与、逻辑非功能的 8 位 ALU，算术运算产生标志 ZF、CF、OF、SF，逻辑运算仅产生 ZF 标志，验证其正确性。

二、电路设计与实现

（1） 加减法器

加减法器直接使用 lpm_add_sub 模块实现 8 位有符号加法。

电路如图所示：



其中 cIn 端口需要在减法状态下手动输入 1。

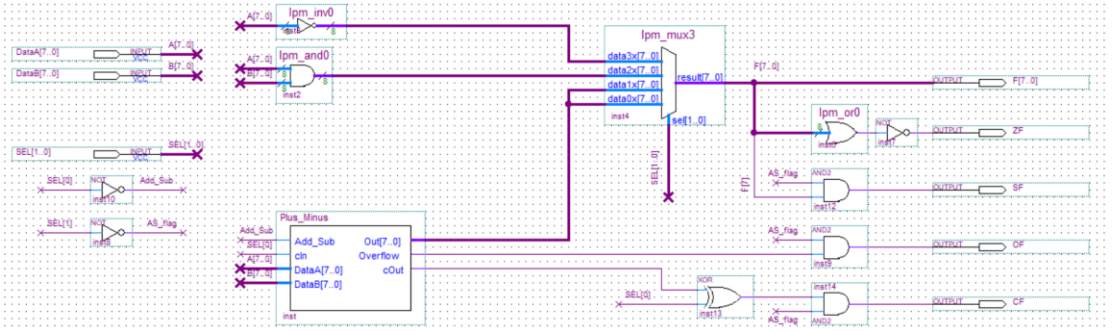
（2） ALU

ALU 中使用 lpm_add_sub 模块实现加减功能，lpm_and 实现逻辑与功能，lpm_inv 实现逻辑非功能。使用 lpm_mux 模块作为数据选择器。

ZF 的或非门采用 lpm_or 搭配非门使用。

SF，OF，CF 的输出有 AS_flag 控制，仅在加减模式下会输出此信号。

电路如图所示：



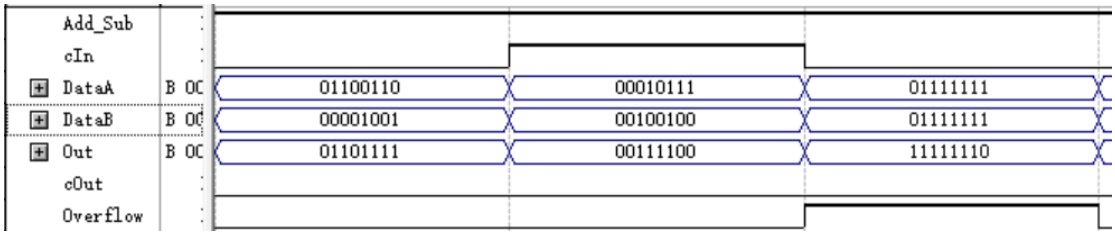
三、电路正确性验证

1、电路仿真

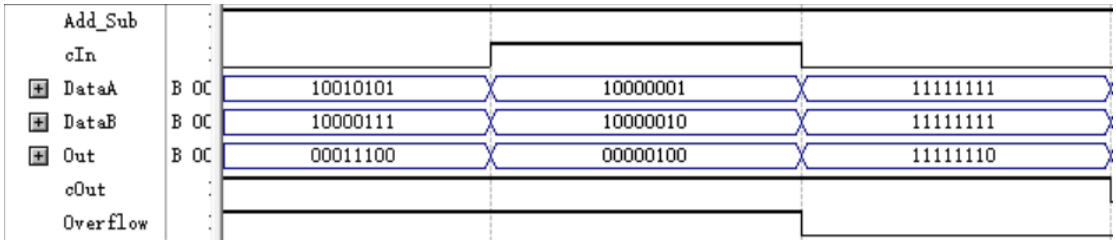
加减法器仿真：

（1） 加法

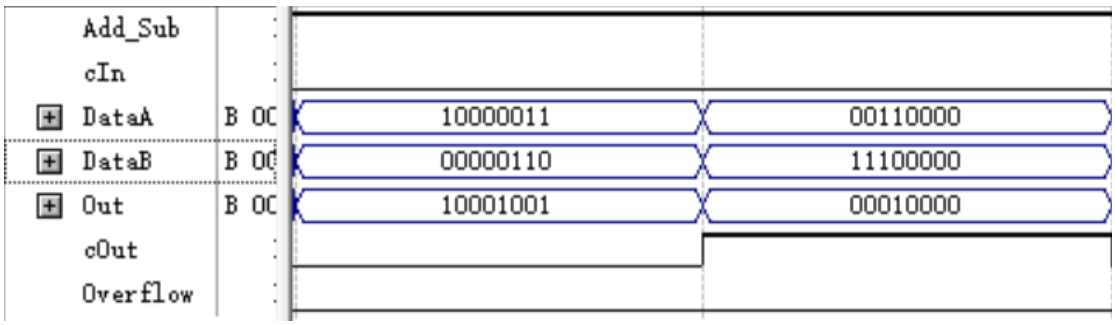
正数相加：



负数相加：

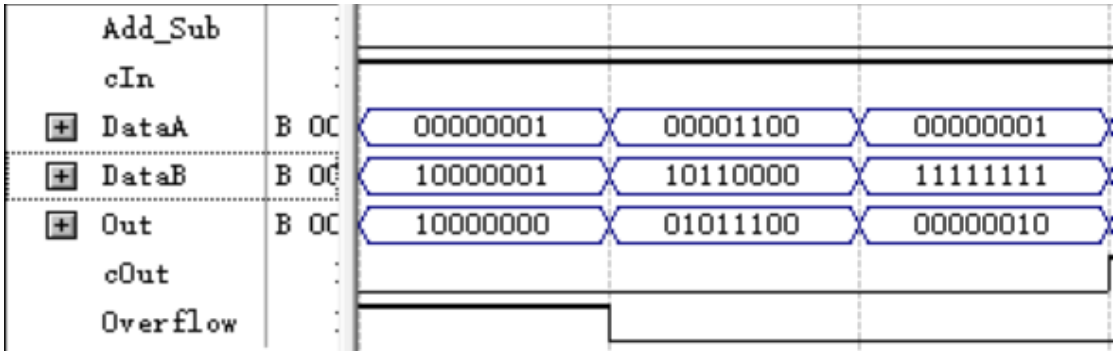


正负相加：

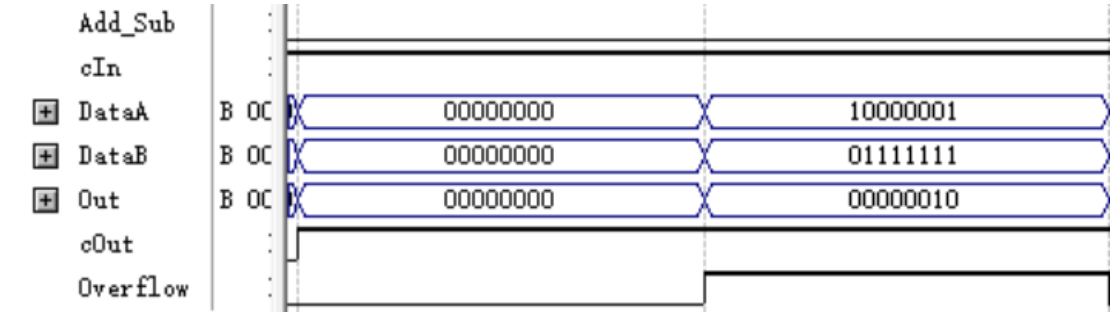


（2） 减法

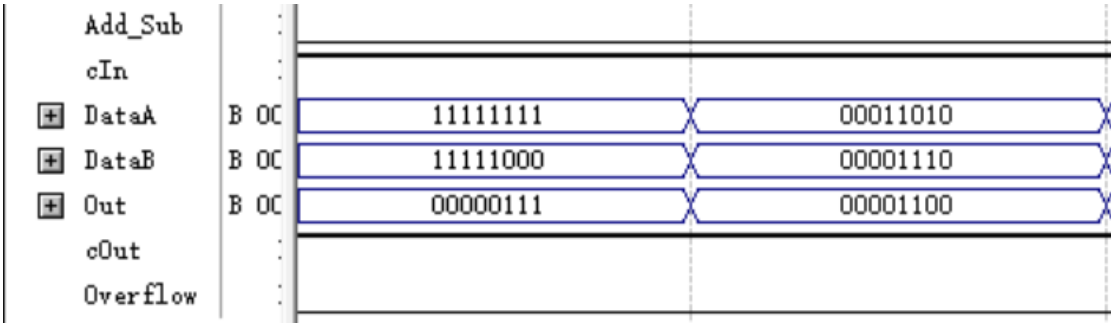
正数减负数：



负数减正数：

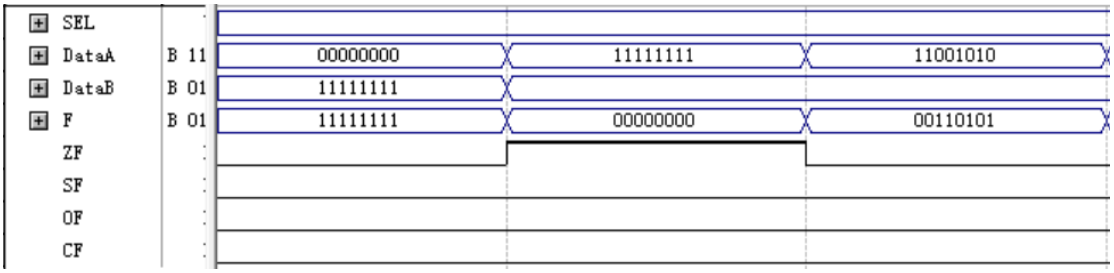


同号相减：

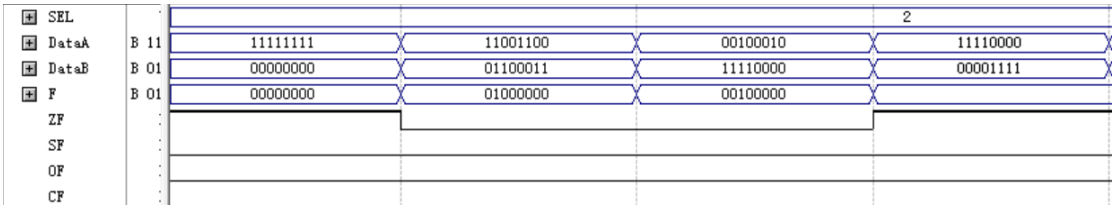


ALU 仿真：

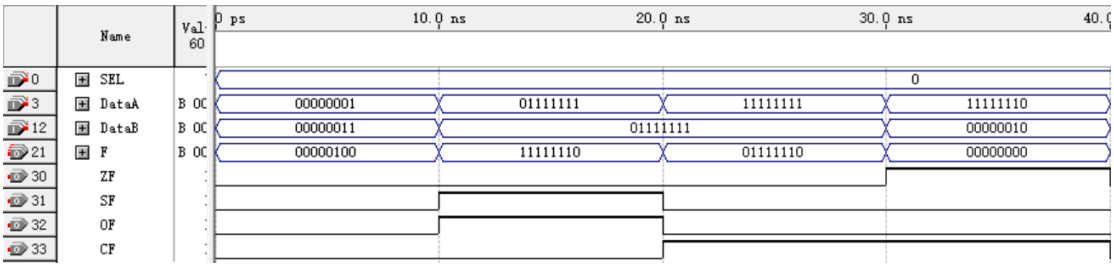
（1） 逻辑非功能



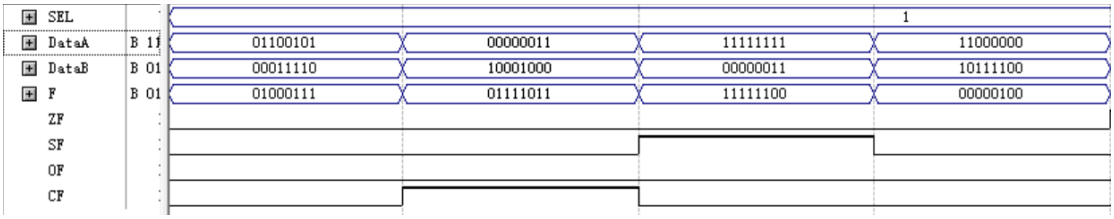
（2） 逻辑与功能



（3）加法功能



（4）减法功能



2、结果分析

（1）加减法器仿真

加法：

正数相加：

1）01100110+00001001=01101111，无进位，无溢出，结果吻合。

2）00010111+00100100+1=00111100，无进位，无溢出，结果吻合。

3）01111111+01111111=11111110，无进位，有溢出，结果吻合。

负数相加：

1）10010101+10000111=100011100，舍去超模的 1，有进位，有溢出，结果吻合。

2）10000001+10000010+1=100000100，舍去超模的 1，有进位，有溢出，结果吻合。

3）11111111+11111111=111111110，舍去超模的 1，有进位，无溢出，结果吻合。

正负相加：

1）10000011+00000110=10001001，无进位，无溢出，结果吻合。

2）00110000+11100000=100010000，舍去超模的 1，有进位，无溢出，结果吻合。

减法：

正数减负数：

1）00000001-10000001=1000000，无进位，有溢出，结果吻合。

2）10000001-10110000=01011100，有进位，无溢出，结果吻合。

3）00000001-11111111=00000010，有进位，无溢出，结果吻合。

负数减正数：

1）10000001-01111111=00000010，无进位，有溢出，结果吻合。

2）11111110-00000001=11111101，无进位，无溢出，结果吻合。

同号相减：

1）11111111-11111000=00000111，无进位，无溢出，结果吻合。

2）00011010-00001110=00001100，无进位，无溢出，结果吻合。

（2）ALU 仿真

逻辑非功能：逻辑非功能将输入 A 的各位取非后输出，与输入 B 的值无关。此功能下不会输出除 ZF 外的信号。

逻辑与功能：逻辑与功能使得输入 A 和输入 B 的各位进行对应的与运算。如 11001100 & 01100011 = 01000000。此功能下不会输出除 ZF 外的信号。

对四个标志位的分析：

CF: 当运算发生进位时，CF 为 1。如 11111111+01111110=101111110，发生进位，CF=1；

SF: 当运算结果为负时，SF 为 1。如 11111111-00000011=11111100，结果为负，SF=1；

OF: 当运算溢出时，OF 为 1。如 01111111+11111110=101111101，发生溢出，OF=1；

ZF: 当结果为 0 时，ZF 为 1。如 1111110+00000010=00000000，ZF=1；

四、实验小结

- 工作分工：
  本实验由孙彦林独立完成，
- 设计总结：
  将加减与非功能分别表示，使用数据选择器选择对应输出。使用 lmp 部件快捷构建部件。
- 待改进之处：
  在使用线直接连接和使用命名线链接之间没有做好平衡。
  测试数据不足，仅覆盖主要范围。
- 实验体会：
  选取合适的数据测试很重要，要注重测试。测试数据应当覆盖元件的所有功能。

五、教师评语

教师签字：
日期：

实验三　存储器设计及总线互连

一、实验内容

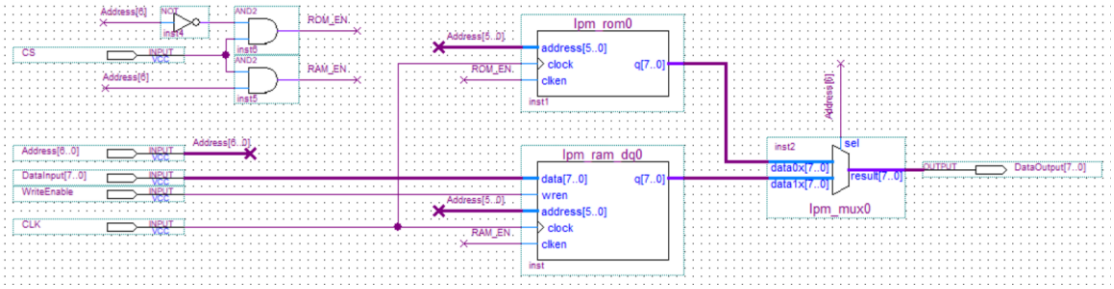
（1）测试 RAM 的功能。

（2）设计并实现单向数据引脚的 128×8 位存储模块，存储模块的前 64B 为只读空间，验证其正确性。

（3）将上述存储模块连接到地址线/数据线复用的 8 位总线上，通过总线对该存储模块进行操作。

二、电路设计与实现

1. 存储模块的实现及验证

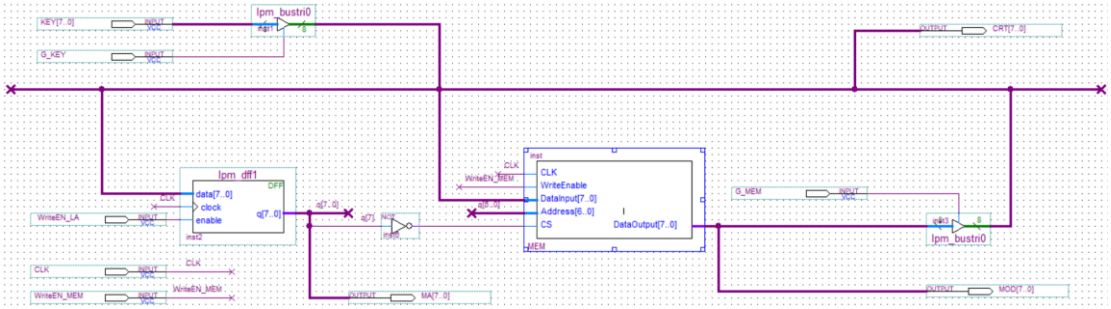


使用 CS 控制两个储存器的 clken，CS=0 时锁定两个储存器。

使用地址线最高位区分 RAM 和 ROM。

使用地址线最高位作为数据选择器的地址，对输出的数据进行选择。

2. 存储器与总线连接的实现与验证



使用 bustri 三态门控制器件对总线的输入输出。

加入 MA，MOD 两个辅助引脚进行输出确认。

使用地址锁存器暂存地址。

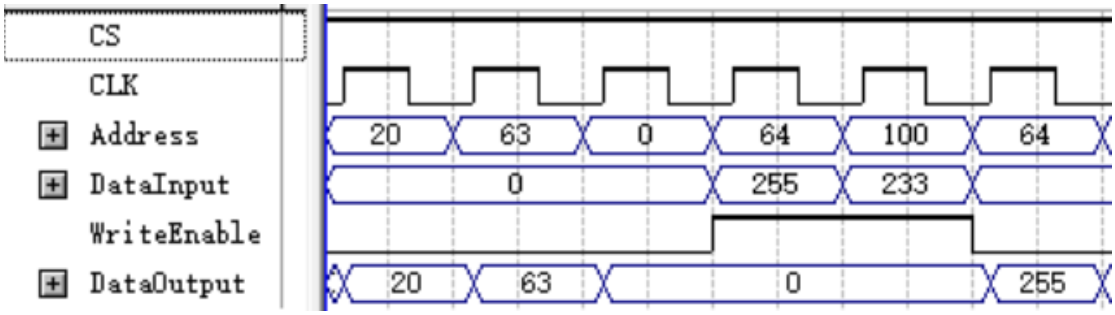
三、电路正确性验证

1、电路仿真

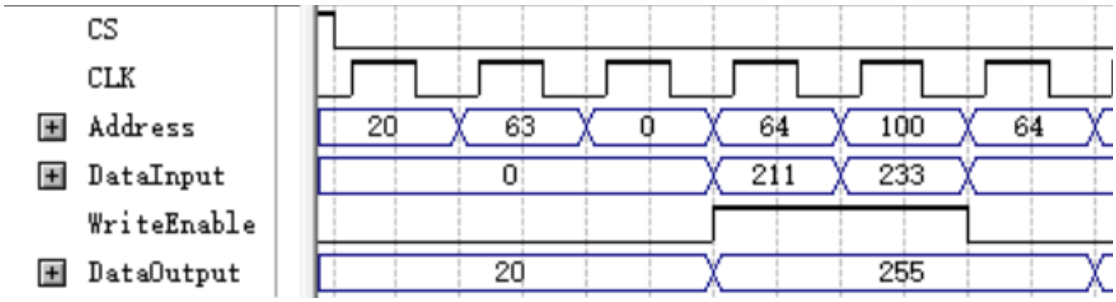
（1） 存储模块的实现及验证

操作为：对 ROM（20）读取；对 ROM（63）读取；向 RAM（64）写入 255；向 RAM（100）写入 233；对 RAM（64）读取。

片选线 CS=1 时进行的操作有效。



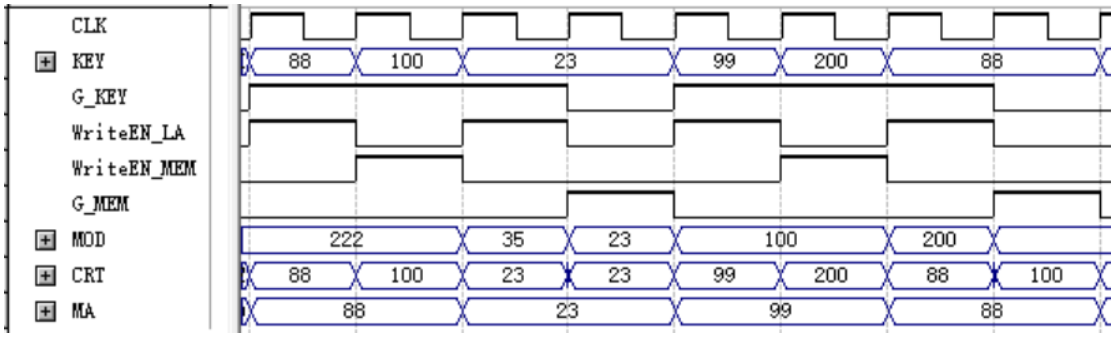
片选线 CS=0 时，操作无效，输出保持。



（2） 总线

操作为：向 RAM（88）写入 100；读取 ROM（23）；向 RAM（99）写入 200；对 RAM（88）读取。

两个时钟周期完成一次访问。



CLK 时钟略微延后（0.1ns）以保证在时钟上升沿到来时其余已完成变化。

2、结果分析

（1） 存储模块的实现和验证

对于存储模块，略微调整时钟周期使得地址和读写控制在时钟周期之前稳定。

WriteEnable 控制是否对存储器写入。高电平时在对应地址写入数据（对 ROM 不启用）

CS 片选线控制存储器是否启用。片选线 CS=0 时，操作无效，输出保持

（2） 总线

由于数据和地址公用总线，因此要分时传输，两个时钟周期完成一次访问。

写入时，KEY 三态门保持开启，MEM 三态门保持关闭。第一周期 LA 写入地址并锁存，第二周期 MEM 按照地址写入数据。

读取时，KEY 三态门在第一阶段开启，从 KEY 输入对应地址，写入 LA 并锁存。第二周期 MEM 三态门开启，输出数据。

共有七位地址线，八位数据。因此第八位作为片选线。

读取时 KEY 第二周期的输出无影响。

四、实验小结

- 工作分工：

  本实验由孙彦林独立完成，
- 设计总结：

  将 RAM 和 ROM 字扩展组合形成 MEM 存储器，并将 MEM 存储器连接到地址/数据复用总线上。
- 待改进之处：

  在测试开始数据选择不当导致精力浪费。

  对时钟进行偏移保障地址稳定后上升沿再到来。
- 实验体会：

  加深了对存储器的理解与对时钟信号的控制。

五、教师评语

教师签字：

日期：

实验四　数据通路的组织

一、实验内容

（1）设计并实现单总线结构的数据通路，支持教材中 Demo_IS 指令系统的取数(LD)、减法(SUB)、双字长分支(JNZ)指令。

（2）编写测试程序并存入存储器，根据所组织的 μOPCmd 序列控制程序执行过程，验证数据通路的正确性。

二、电路设计与实现

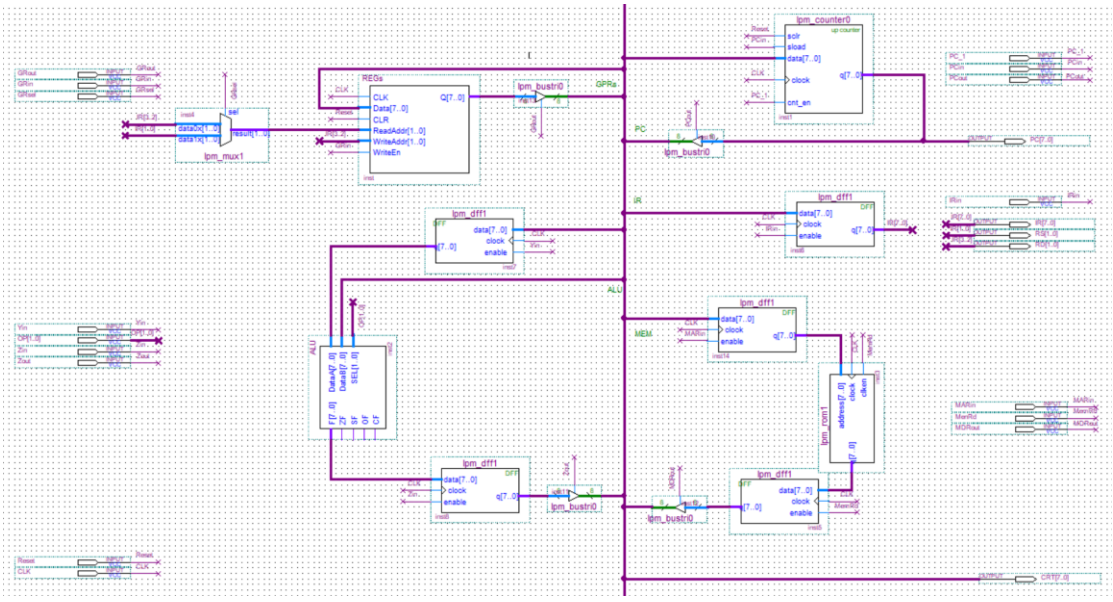
实现取数(LD)、减法(SUB)、双字长分支(JNZ)指令需要的元件有：ALU, MEM, GPRs, PC, IR。其中，ALU、GPRs 可以取用实验一、实验二的封装结果；PC 使用 lpm_counter 模块；三态门使用 lpm_bustri；IR 以及各锁存器使用 D 触发器 lpm_dff 代替；由于不需要向内存写入，MEM 采用 lpm_rom 即可。

采用总线结构，通过寄存器和三态门防止数据干扰。

状态寄存器省略，产生的状态手动识别。

指令译码器省略，产生的时钟由手动输入实现。

通过修改 ROM 初始化文件来实现编程。以黄色表示取数，蓝色表示运算，绿色表示跳转。结果如图：



内存初始化为：

Addr	+0	+1	+2	+3	+4	+5	+6	+7
00	24	29	69	C0	22	69	C0	22
08	00	00	00	00	00	00	00	00
10	00	00	00	00	00	00	00	00
18	00	00	00	00	00	00	00	00
20	00	00	FF	00	48	00	00	00

三、电路正确性验证

1、电路仿真

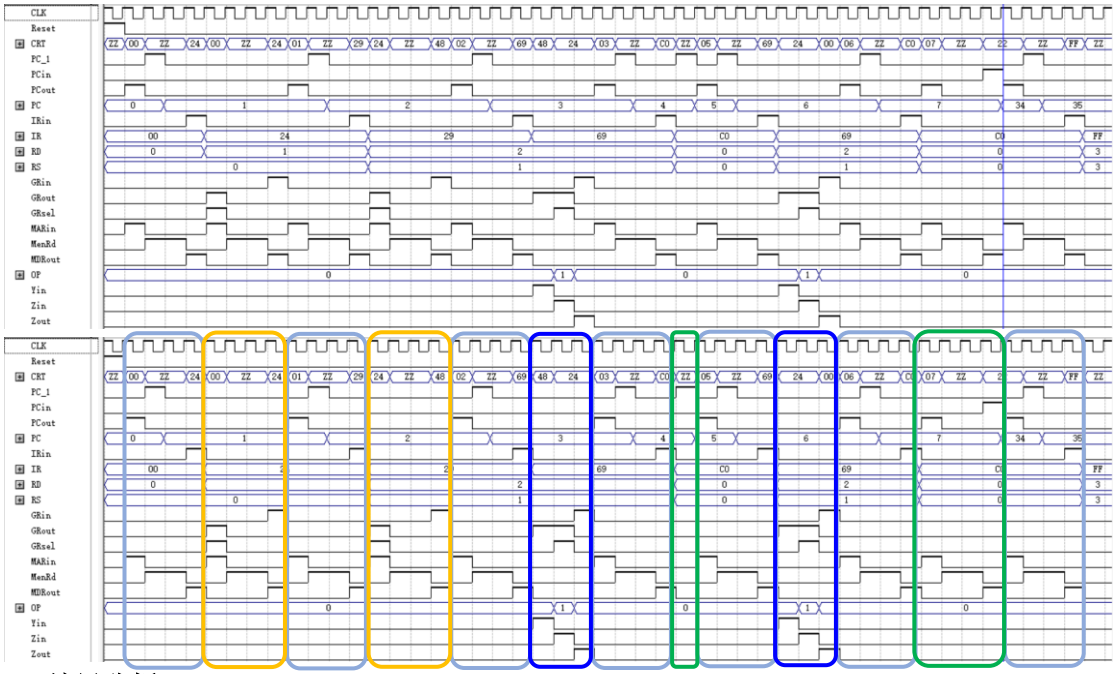
第一个节拍进行初始化，后面开始执行指令。

指令，【】内为本例中的说明：

①R1←M[(R0)]【24】　②R2←M[(R1)]【48】　③R2←(R2)－(R1)【48-24】

④JNZ 22H【跳转失败】　⑤ R2←(R2)－(R1)【24-24】　⑥JNZ 22H【跳转成功】

蓝色直线之后又进行了一次取址，取到 22H 中的内容 FFH，表明运行结果正确。



2、结果分析

a) 取址(灰色标记)

取指令阶段的 μOP 序列与 μOPCmd 序列为：

t1:MAR←(PC)	t1:PCout, MARin
t2:MDR←M[(MAR)], PC←(PC)+1	t2:MenRd, PC+1
t3:IR←(MDR)	t3:MDRout, IRin

b) 执行

i. 取数(LD)（黄色标记）

t4:MAR←RS	t4:MARin, Grout, Rsel
t5:MDR←M[(MAR)]	t5:MenRd
t6:RD←(MDR)	t6:MDRout, GRin

ii. 减法(SUB)（蓝色标记）

t4:Y←(RD)	t4:GRout, Yin
t5:Z←(Y)-(RS)	t5:GRout, Rsel, op=01, Zin
t6:RD←(Z)	t6:Zout, GRin

iii. 双字长分支(JNZ)（绿色标记）

1) Z=0 时跳转

t4:MAR←(PC)	t4:PCout, MARin
t5:MDR←M[(MAR)]	t5: MenRd
t6:PC←(MDR)	t6:MDRout, PCin

2) Z!=0 时继续向下

t4: PC←(PC)+1	t4: PC+1

四、实验小结

- 工作分工：
  本实验由孙彦林独立完成，
- 设计总结：
  利用一二次的实验结果与第三次实验的经验完成数据通路。
  编写微操作实现部分功能
- 待改进之处：
  明确线路命名，防止报错。
  明确指令内容，是 ZF=0 跳转还是 ZF!=0 跳转。
  创建宏模块时对界面不熟悉浪费过多时间。
- 实验体会：
  加深了对微指令的理解。
  学会了将复杂问题模块化解决的思考方式

五、教师评语

教师签字：
日期：