

# 计算机设计与实践

## 单周期CPU设计-2



HITSZ 实验与创新实践教育中心  
Education Center of Experiments and Innovations, HITSZ

# 实验目的

---

- ◆ 加深对CPU**结构**和**工作原理**的理解
- ◆ 掌握根据数据通路表和控制信号取值表来实现**执行**、**访存**和**写回**单元的方法
- ◆ 熟练掌握使用Verilog HDL实现CPU的功能部件



# 实验内容

- ◆ 使用Verilog HDL实现单周期CPU的执行单元:
  - ◆ 根据数据通路表和控制信号取值表，确定执行单元的接口
  - ◆ 实现所需的功能部件（如ALU, etc.）



- ◆ 根据数据通路表，连接各个部件，形成执行单元

# 实验内容

- ◆ 使用Verilog HDL实现单周期CPU的访存单元：
  - ◆ 根据数据通路表和控制信号取值表，确定访存单元的接口
  - ◆ 实现所需的功能部件（如DRAM，etc.）



- ◆ 根据数据通路表，连接各个部件，形成访存单元
- ◆ 使用Verilog HDL实现单周期CPU的写回逻辑

# 构建功能部件 —— ALU

## ◆ ALU的功能:

### ◆ 算术运算 (add、sub)

✓ 加法器、补码逻辑

### ◆ 逻辑运算 (and、or, etc)

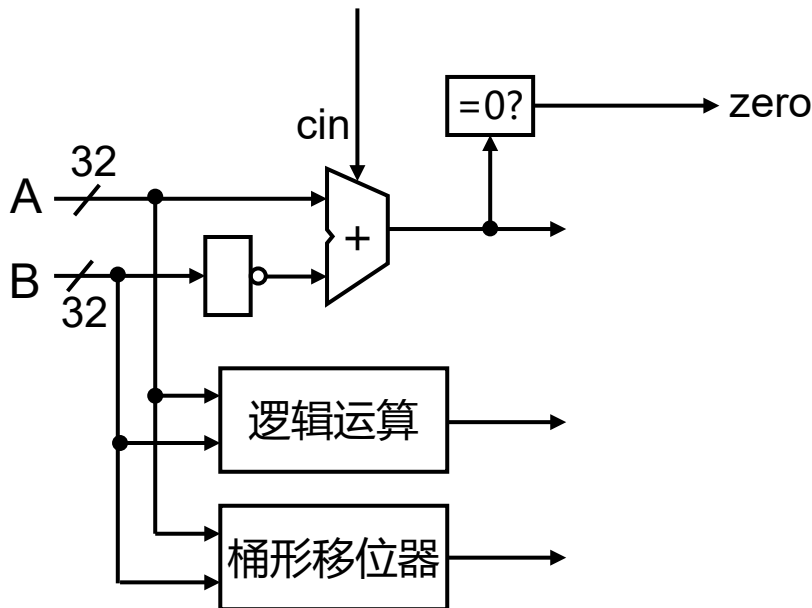
✓ 按位操作

### ◆ 移位运算 (sll、srl, etc)

✓ 桶形移位器

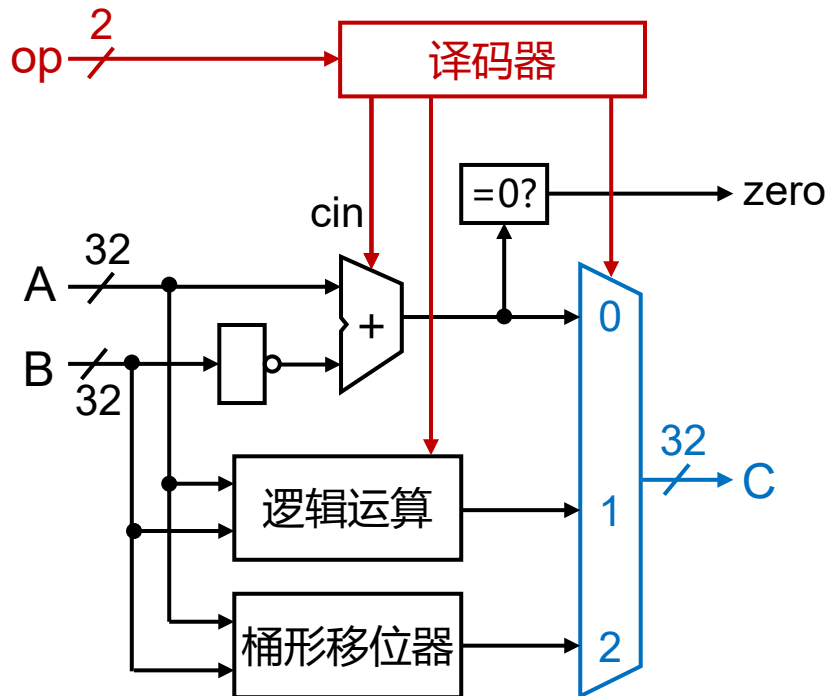
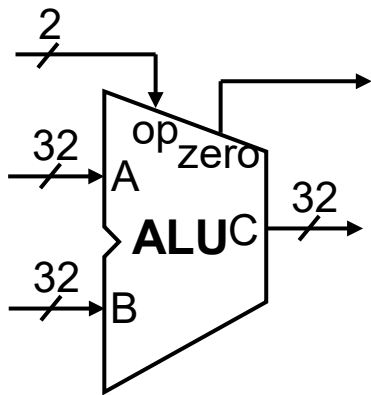
### ◆ 比较运算 (slt、sltu, etc)

✓ 复用sub, 判断标志位



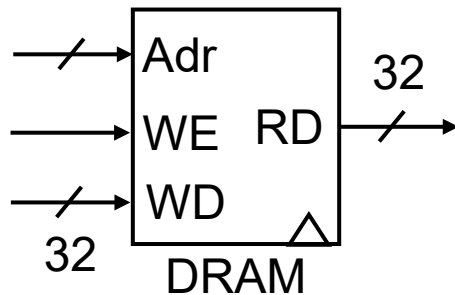
# 构建功能部件 —— ALU

- ◆ 集成各项功能：
  - ◆ 多路选择器选择ALU输出
  - ◆ 译码器选择具体运算



# 构建功能部件 —— DRAM

- ◆ DRAM是一个按地址访问的存储器，可读可写
  - ◆ 读操作：输入地址，输出数据
  - ◆ 写操作：输入地址、数据，受写使能信号WE控制
- ◆ DRAM的读写逻辑：
  - ◆  $WE=0$ :  $RD \leftarrow DRAM[Adr]$
  - ◆  $WE=1$ :  $DRAM[Adr] \leftarrow WD$



# 执行模块实现

## ◆ 选择运算数据

### ◆ 根据数据通路表，选择源操作数：

input [31:0] rd\_dat1\_i; // (rs1)

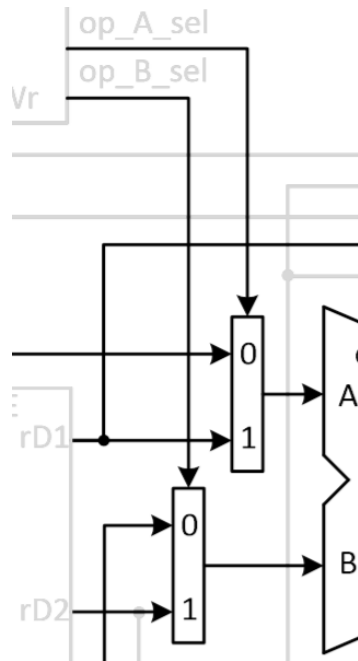
input [31:0] rd\_dat2\_i; // (rs2)

input [31:0] npc\_i; // Next PC

input [31:0] imm\_i; // 扩展后的立即数

assign op\_A = op\_A\_sel ? npc\_i : rd\_dat1\_i;

assign op\_B = op\_B\_sel ? imm\_i : rd\_dat2\_i;



# 执行模块实现

---

- ◆ 选择运算数据
- ◆ 完成算术、逻辑、移位运算
  - ◆ **要点：**调用设计好的ALU模块；用控制器的op信号选择运算功能
- ◆ 完成比较运算的PC值计算
  - ◆ **要点：**B型、J型指令的立即数在符号扩展后，需要加上PC的值
- ◆ 完成运算结果输出



# 实验步骤

---

- ① 根据数据通路表，确定执行、访存单元**包含哪些部件**
- ② 根据数据通路表，确定各功能部件的**接口**和需要实现的**功能**
- ③ 使用Verilog HDL实现各功能部件
- ④ 根据数据通路表，将各功能**部件连接**起来
- ⑤ **封装**成模块，得到执行单元和访存单元
- ⑥ 根据数据通路表，连接相应的信号，实现写回逻辑







# 开始实验



HITSZ 实验与创新实践教育中心  
Education Center of Experiments and Innovations, HITSZ