# Lab5: Pipelined CPU

Powered By TA-Pepcy TA-Ziyue

# 实验目标

- RV32I Pipelined CPU
- 对于Hazard，只需要Stall
- 支持以下指令
  - lui, auipc, jal, jalr, lw, sw
  - beq, bne, blt, bge, bltu, bgeu
  - addi, slti, sltiu, xori, ori, andi, slli, srli, srai
  - add, sub, sll, slt, sltu, xor, srl, sra, or, and

# PART 1: 基本的Pipelined CPU

# 先实现基本的指令

- lw, sw
- beq, bne, blt, bge, bltu, bgeu
- addi, slti, sltiu, xori, ori, andi, slli, srli, srai
- add, sub, sll, slt, sltu, xor, srl, sra, or, and

# 可复用的部件

- ALU
- Comperator
- IMem
- DMem

# Stall逻辑

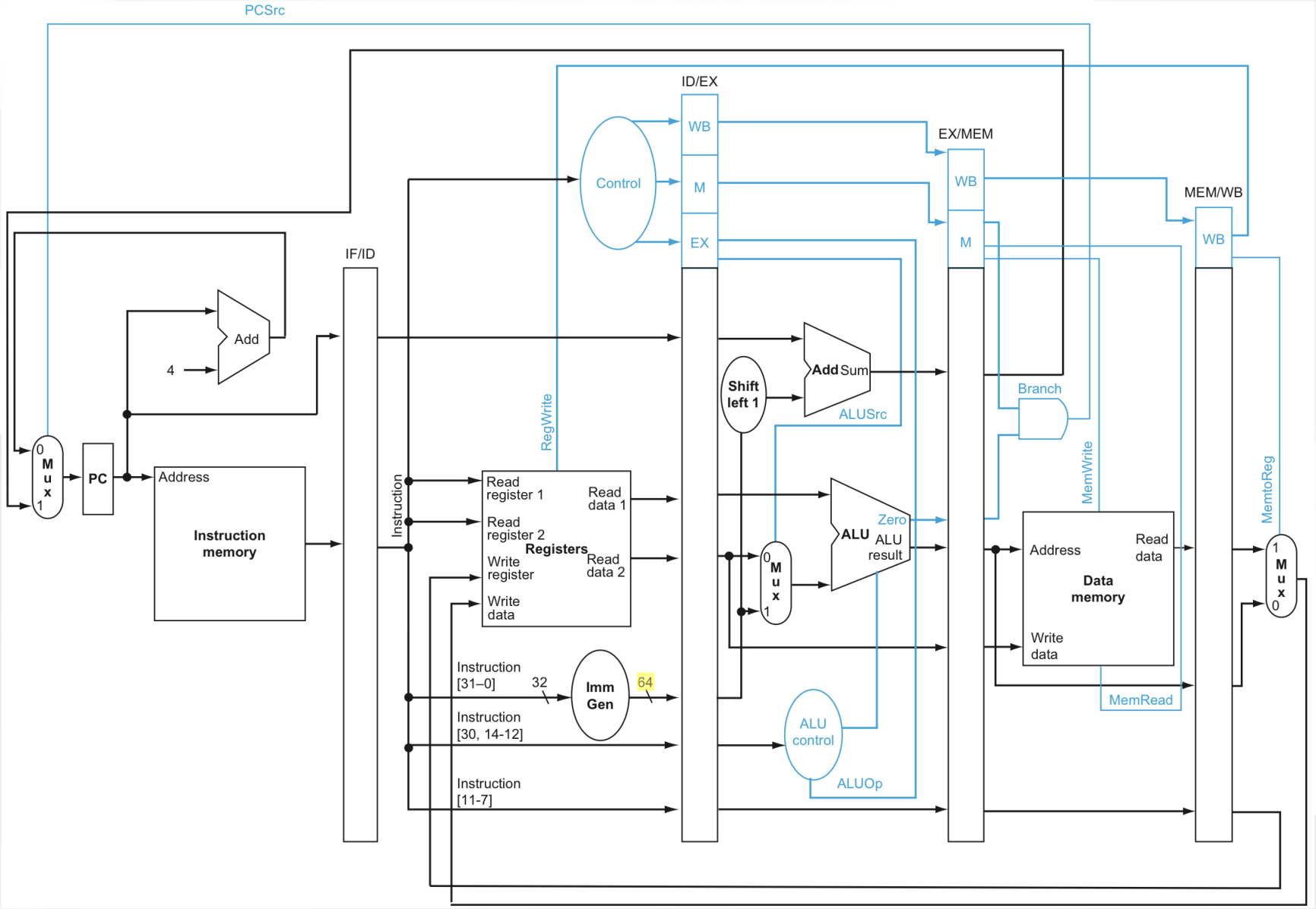
在这个实验中，不需要考虑Forwarding和Branch Prediction，只需要用Stall来处理所有Hazard和Branch即可

Stall逻辑放在哪一个流水级取决于你的设计。（一个经典的做法是放在ID级）

# DataPath

可以参考课本第294页的DataPath或者课件中的DataPath设计。

*但要注意课件上的DataPath是64-bit的，我们需要微调一下数据宽度。*


PCSrc
ID/EX
EX/MEM
MEM/WB
WB
M
EX
Control
IF/ID
Add
4
M u x
0
1
PC
Address
Instruction memory
Instruction
RegWrite
Read register 1
Read register 2
Registers
Write register
Write data
Read data 1
Read data 2
Shift left 1
Add Sum
ALUSrc
Branch
MemWrite
MemtoReg
ALU
Zero
ALU result
Address
Read data
Data memory
Write data
MemRead
Instruction [31–0]
32
Imm Gen
64
Instruction [30, 14-12]
Instruction [11-7]
ALU control
ALUOp

# PART 2: 扩展更多指令

- lui
- auipc
- jal
- jalr

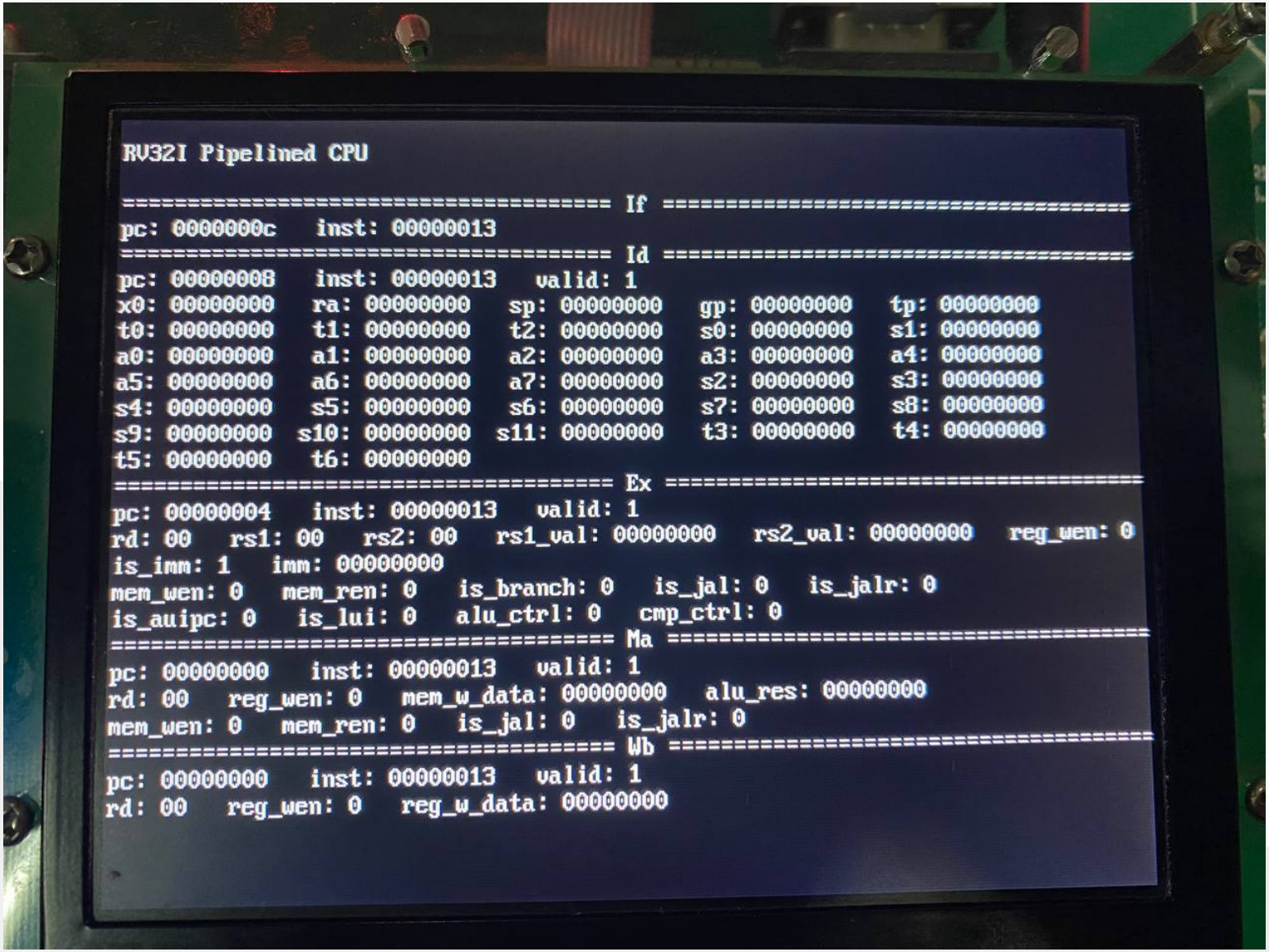
# 扩展指令的思路

与单周期CPU中扩展指令的方式一样，我们需要分析目前的数据通路是否能满足这条指令的运行。

如果不行，就要考虑增减部件、控制信号。

# 上板

这一次我们也有对应的VGA部件，请见`Material.zip`中的EDIF核。


RV32I Pipelined CPU
========== If ==========
pc: 0000000c inst: 00000013
========== Id ==========
pc: 00000008 inst: 00000013 valid: 1
x0: 00000000 ra: 00000000 sp: 00000000 gp: 00000000 tp: 00000000
t0: 00000000 t1: 00000000 t2: 00000000 s0: 00000000 s1: 00000000
a0: 00000000 a1: 00000000 a2: 00000000 a3: 00000000 a4: 00000000
a5: 00000000 a6: 00000000 a7: 00000000 s2: 00000000 s3: 00000000
s4: 00000000 s5: 00000000 s6: 00000000 s7: 00000000 s8: 00000000
s9: 00000000 s10: 00000000 s11: 00000000 t3: 00000000 t4: 00000000
t5: 00000000 t6: 00000000
========== Ex ==========
pc: 00000004 inst: 00000013 valid: 1
rd: 00 rs1: 00 rs2: 00 rs1_val: 00000000 rs2_val: 00000000 reg_wen: 0
is_imm: 1 imm: 00000000
mem_wen: 0 mem_ren: 0 is_branch: 0 is_jal: 0 is_jalr: 0
is_auipc: 0 is_lui: 0 alu_ctrl: 0 cmp_ctrl: 0
========== Ma ==========
pc: 00000000 inst: 00000013 valid: 1
rd: 00 reg_wen: 0 mem_w_data: 00000000 alu_res: 00000000
mem_wen: 0 mem_ren: 0 is_jal: 0 is_jalr: 0
========== Wb ==========
pc: 00000000 inst: 00000013 valid: 1
rd: 00 reg_wen: 0 reg_w_data: 00000000

# 验收

## 截止日期: 2021-06-28 23:59:59

```
Lab5_319010xxxx.zip
    ├── Lab5.bit      // 生成的bitstream二进制文件
    ├── Source        // 源代码
    │     └── ...
    ├── Project       // 工程目录
    │     └── ...
    └── README        // 需要补充的说明（如果没有需要特殊说明的则不需要此文件）
```

# 报告

## 截止日期: 2021-06-23 23:59:59

```
Lab5_319010xxxx.pdf // 单个pdf文件
```

# Q & A