RISC-V Simulator

PPCA 2020

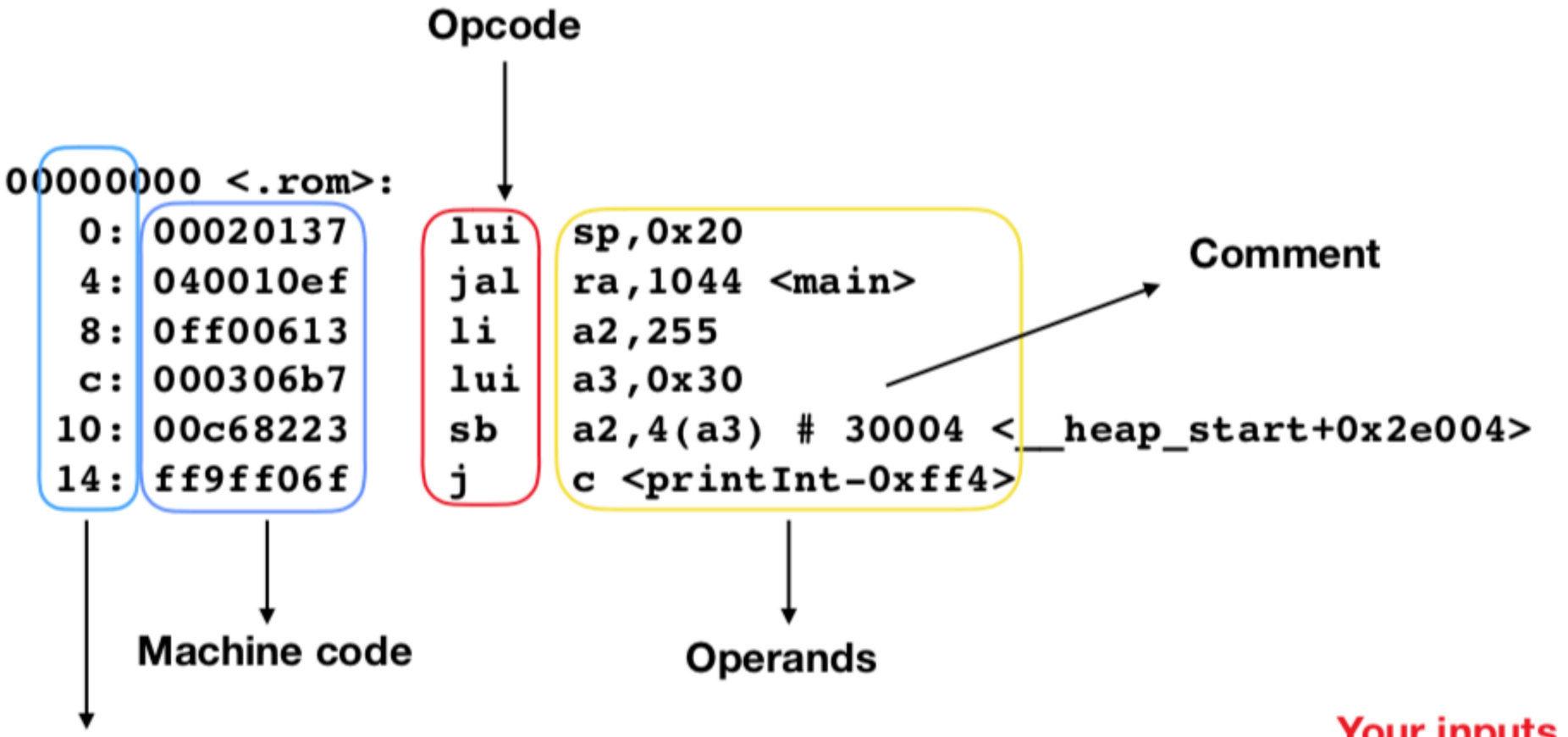
RISC-V Assembly

- 汇编语言 (assembly language) 的一种
- 是基于计算机基础结构的基础语言
- 每条指令语句都会是简单的，仅涉及三个元素的操作
- Reference: Chapter 2, The RISC-V Instruction Set Manual, VOL1（参考2.1-2.7即可）
- 使用RV32I基础整数指令集 (RV32I Base Integer ISA)

具体指令

- 摘自manual的P104
- 指令呈现方式为二进制，即使用一个32位的01串来描述
- 每条指令的详细含义参考Chapter2
- 右图为本项目需要实现的全部指令

imm[31:12]				rd	0110111	LUI
imm[31:12]				rd	0010111	AUIPC
imm[20|10:1|11|19:12]				rd	1101111	JAL
imm[11:0]		rs1	000	rd	1100111	JALR
imm[12|10:5]	rs2	rs1	000	imm[4:1|11]	1100011	BEQ
imm[12|10:5]	rs2	rs1	001	imm[4:1|11]	1100011	BNE
imm[12|10:5]	rs2	rs1	100	imm[4:1|11]	1100011	BLT
imm[12|10:5]	rs2	rs1	101	imm[4:1|11]	1100011	BGE
imm[12|10:5]	rs2	rs1	110	imm[4:1|11]	1100011	BLTU
imm[12|10:5]	rs2	rs1	111	imm[4:1|11]	1100011	BGEU
imm[11:0]		rs1	000	rd	0000011	LB
imm[11:0]		rs1	001	rd	0000011	LH
imm[11:0]		rs1	010	rd	0000011	LW
imm[11:0]		rs1	100	rd	0000011	LBU
imm[11:0]		rs1	101	rd	0000011	LHU
imm[11:5]	rs2	rs1	000	imm[4:0]	0100011	SB
imm[11:5]	rs2	rs1	001	imm[4:0]	0100011	SH
imm[11:5]	rs2	rs1	010	imm[4:0]	0100011	SW
imm[11:0]		rs1	000	rd	0010011	ADDI
imm[11:0]		rs1	010	rd	0010011	SLTI
imm[11:0]		rs1	011	rd	0010011	SLTIU
imm[11:0]		rs1	100	rd	0010011	XORI
imm[11:0]		rs1	110	rd	0010011	ORI
imm[11:0]		rs1	111	rd	0010011	ANDI
0000000	shamt	rs1	001	rd	0010011	SLLI
0000000	shamt	rs1	101	rd	0010011	SRLI
0100000	shamt	rs1	101	rd	0010011	SRAI
0000000	rs2	rs1	000	rd	0110011	ADD
0100000	rs2	rs1	000	rd	0110011	SUB
0000000	rs2	rs1	001	rd	0110011	SLL
0000000	rs2	rs1	010	rd	0110011	SLT
0000000	rs2	rs1	011	rd	0110011	SLTU
0000000	rs2	rs1	100	rd	0110011	XOR
0000000	rs2	rs1	101	rd	0110011	SRL
0100000	rs2	rs1	101	rd	0110011	SRA
0000000	rs2	rs1	110	rd	0110011	OR
0000000	rs2	rs1	111	rd	0110011	AND



.dump file

Your inputs

```
@00000000
37 01 02 00 EF 10 00 04 13 06 F0 0F B7 06 03 00
23 82 C6 00 6F F0 9F FF
```

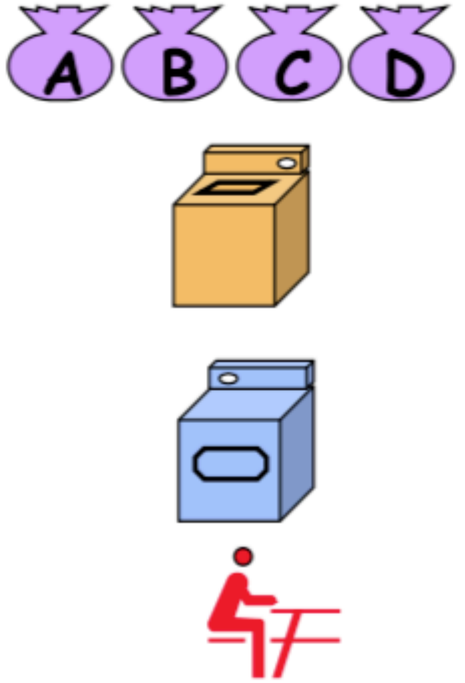
.data file

流水 Pipeling

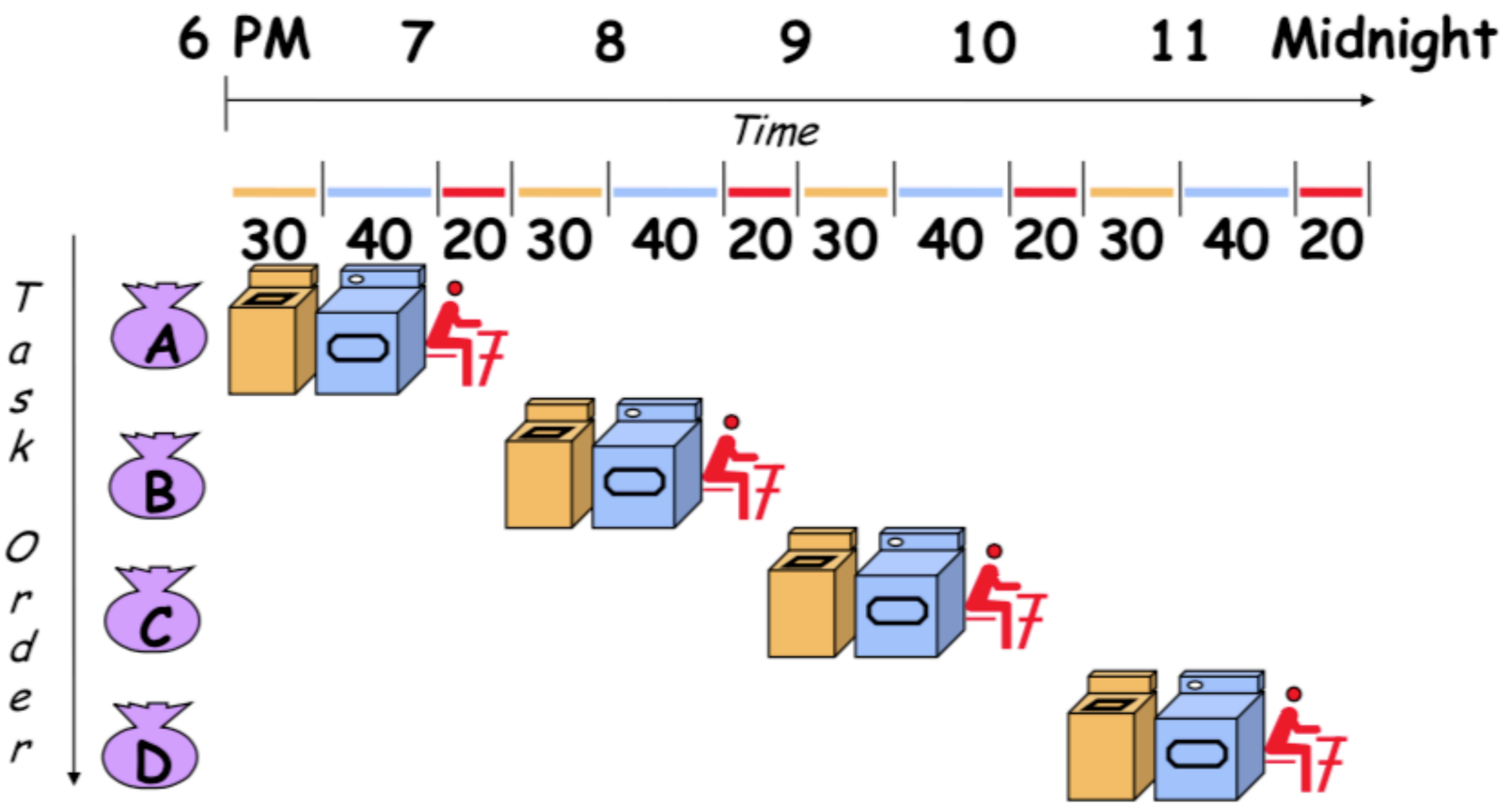
- Reference: CAAQA, John L. Hennessy & David A. Patterson
- 参考：《计算机体系结构：量化研究方法》

Pipelining: Its Natural!

- Laundry Example
- Ann, Brian, Cathy, Dave each have one load of clothes to wash, dry, and fold
- Washer takes 30 minutes
- Dryer takes 40 minutes
- "Folder" takes 20 minutes

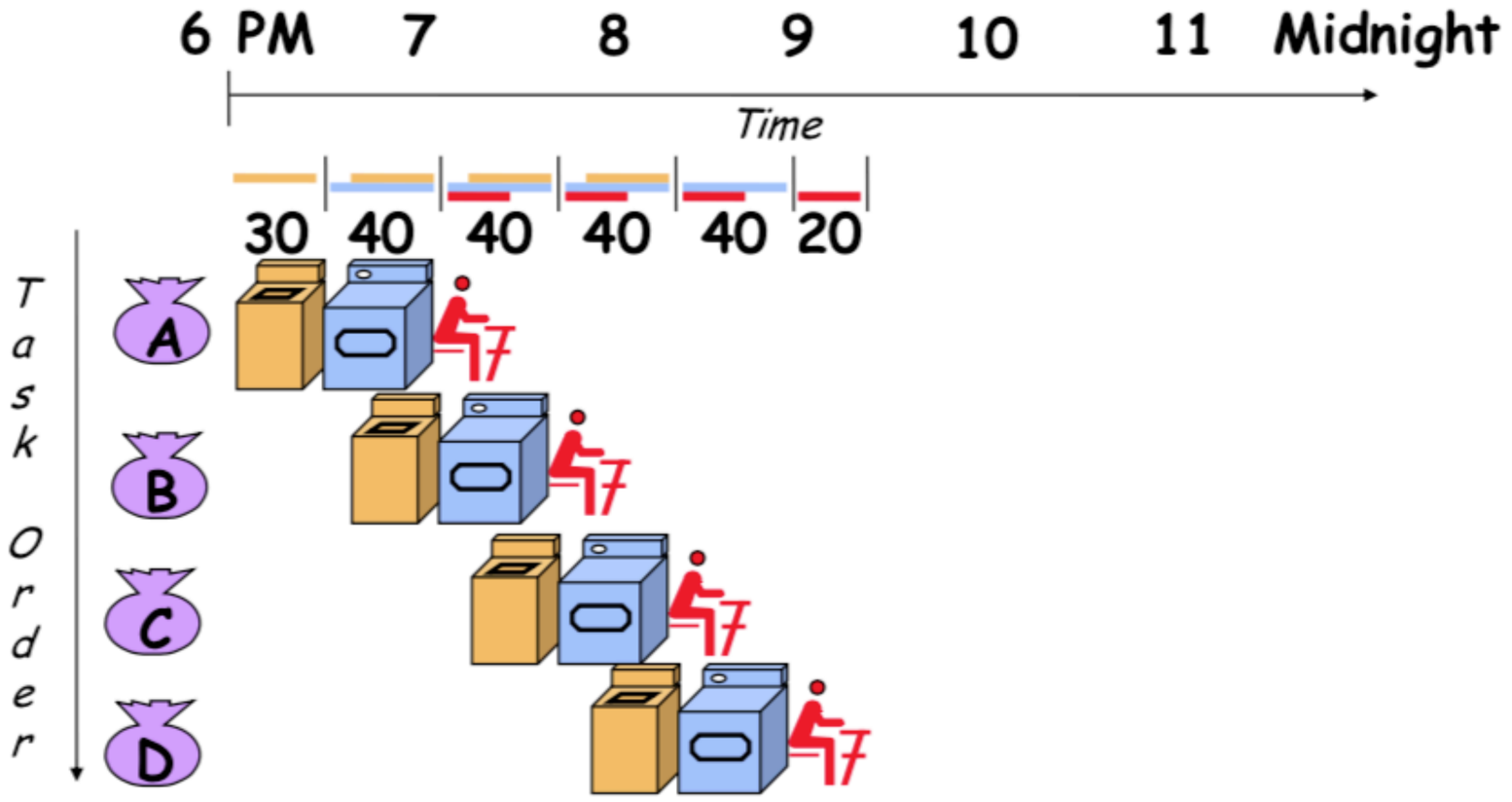


Sequential Laundry



- Sequential laundry takes 6 hours for 4 loads
- If they learned pipelining, how long would laundry take?

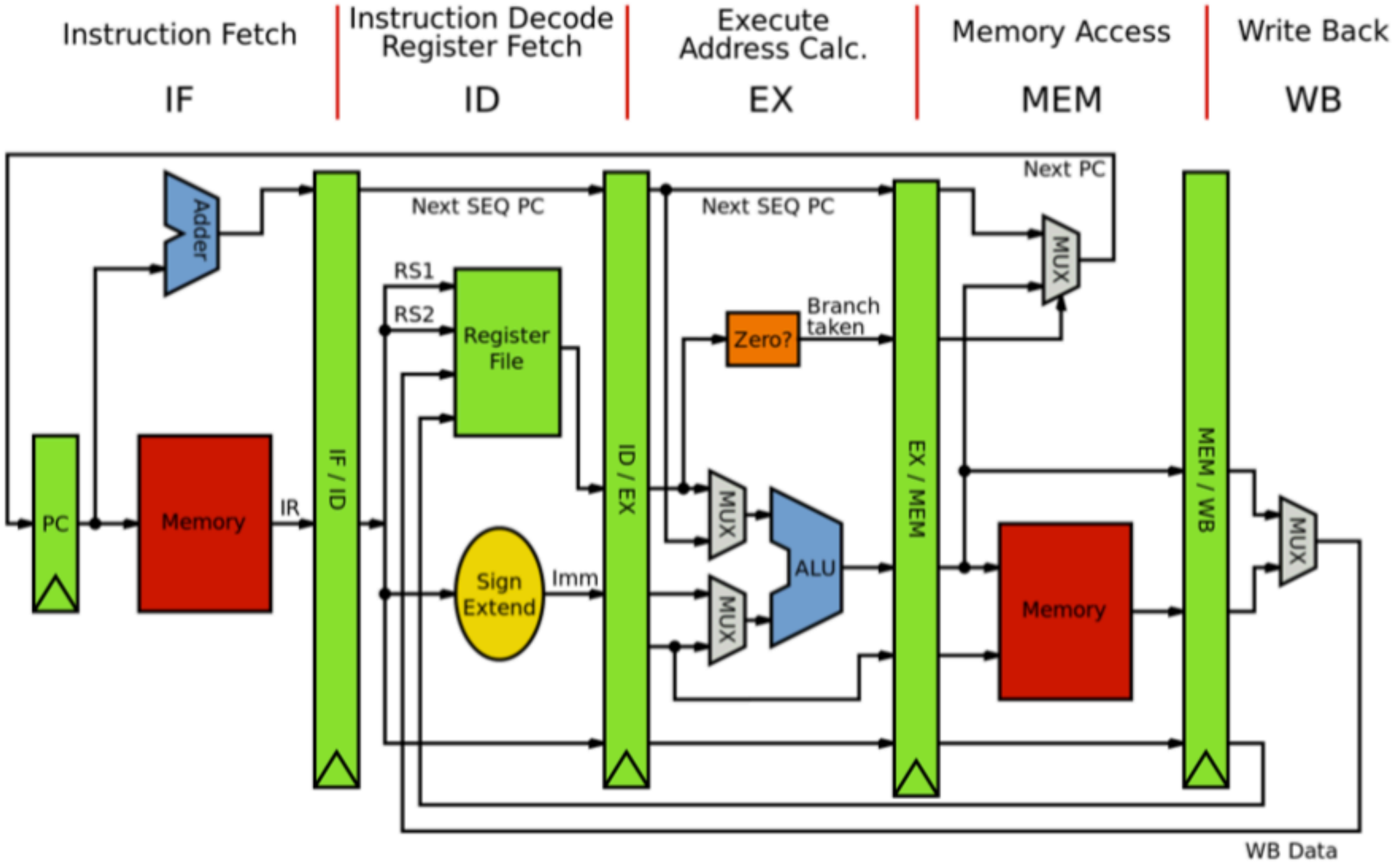
Pipelined Laundry
Start work ASAP



- Pipelined laundry takes 3.5 hours for 4 loads

五级流水 5-Stage Pipeline

- Instruction Fetch (IF)：根据PC寄存器访问内存得到指令
- Instruction Decode (ID)：根据指令类型找到作为右值的寄存器并读值、并且解析立即数的值
- Execute (EX)：对解析好的值按指令要求进行计算
- Memory Access (MEM)：根据计算出的地址从内存读出数据值，或将已准备好的数据值写入内存
- Write Back (WB)：完成对于左值寄存器的赋值，即写回寄存器


Instruction Fetch
IF
Instruction Decode
Register Fetch
ID
Execute
Address Calc.
EX
Memory Access
MEM
Write Back
WB
Adder
Next SEQ PC
Next SEQ PC
Next PC
RS1
RS2
Register
File
Zero?
Branch
taken
MUX
PC
Memory
IR
IF / ID
ID / EX
EX / MEM
MEM / WB
Sign
Extend
Imm
MUX
ALU
MUX
Memory
MUX
WB Data

Hazards

- Structural hazard（模拟时不会出现这种hazard）
- Data hazard
- Control hazard
- 最简单的解决方案即等待闲置，暂停流水

分支预测

- 遇到跳转分支指令时的预测，最简单的预测方案为：总是跳转，或总是不跳转（但这种预测方案不符合本次作业分支预测的最低要求）
- 最低要求：实现一个2位饱和计数器分支预测，当然也可以实现更高级的分支预测
- 支持留出接口统计分支预测成功率并展示之

Bonus

- Tomasulo 乱序执行（含 Speculation）
- 精准中断
- 其他特别的地方

Simulation

- 给出的数据中含有.c, .dump和.data文件
- Input (stdin)：.data 文件
- Output (stdout)：模拟器运行结果
- .c文件和.dump文件可供参考