Pipelined RISC-V Processor

1 功能介绍

该流水线处理器基于 32-bit RISC-V(RV32I)指令集架构，目前仅支持 R 型指令 add, sub, and, or, slt；I 型指令 lw, addi；S 型指令 sw；B 型指令 beq；J 型指令 jal。

2 设计结构框图

该 5 级流水线 RISC-V 处理器的顶层框图如 Figure 1 所示。

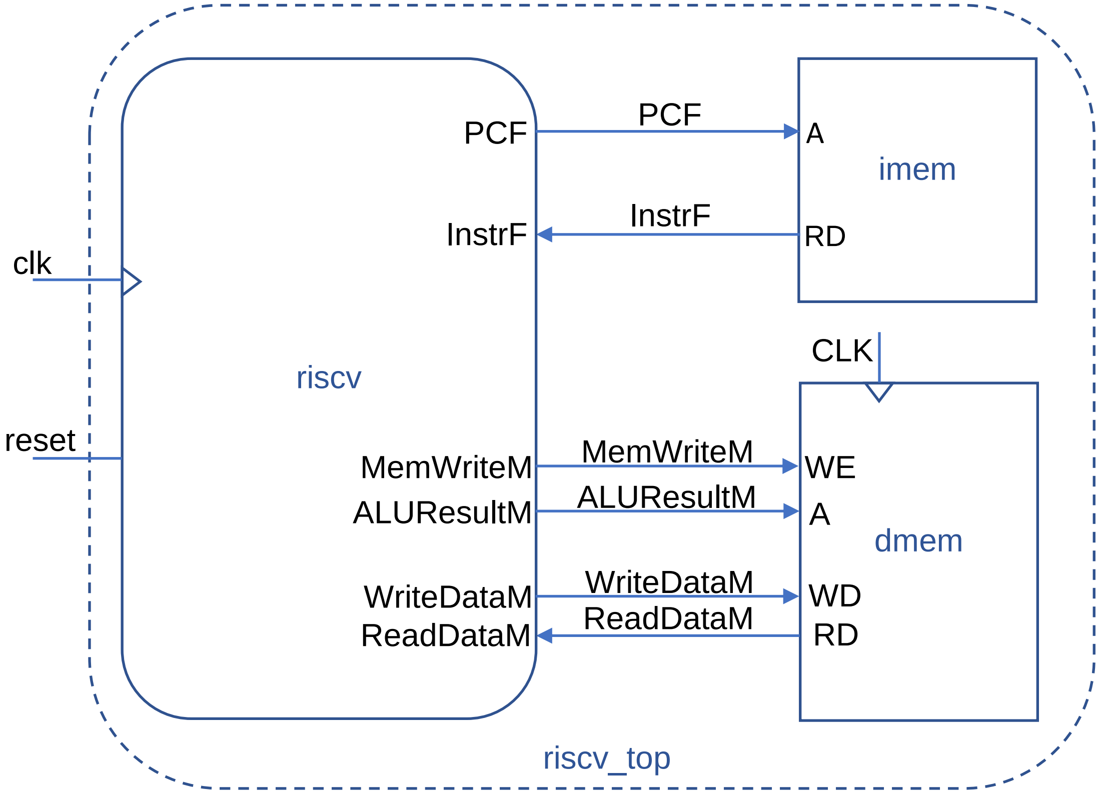


Figure 1

具体的内部模块结构如 Figure 2 所示。

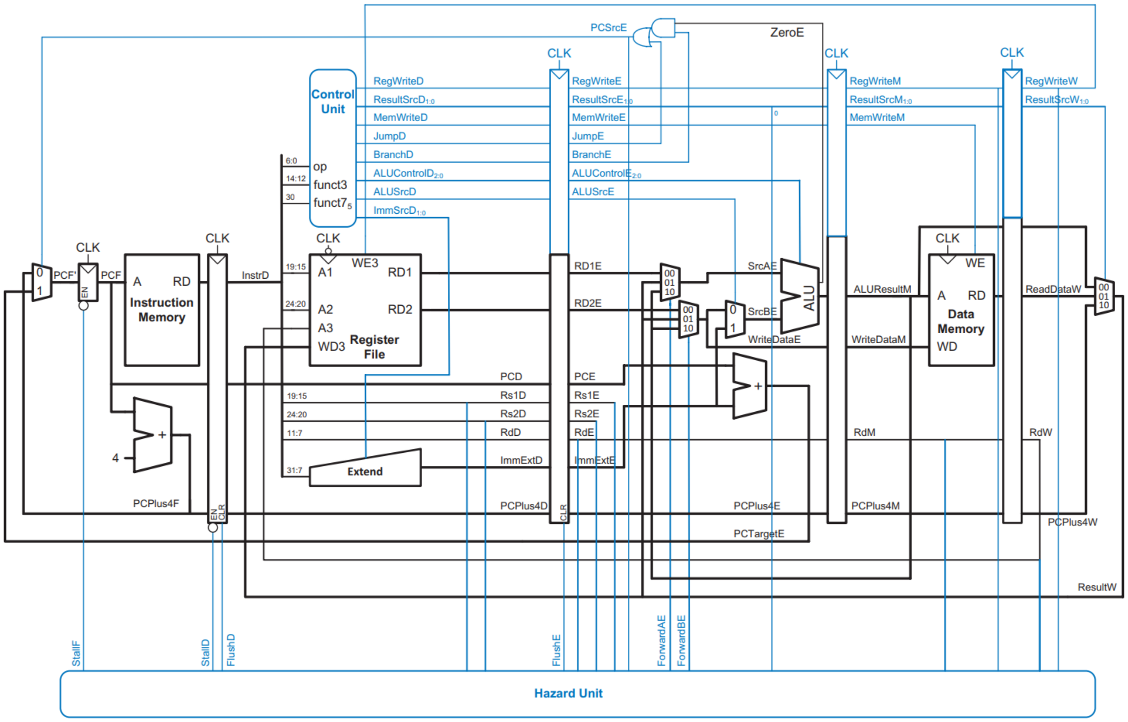


Figure 2

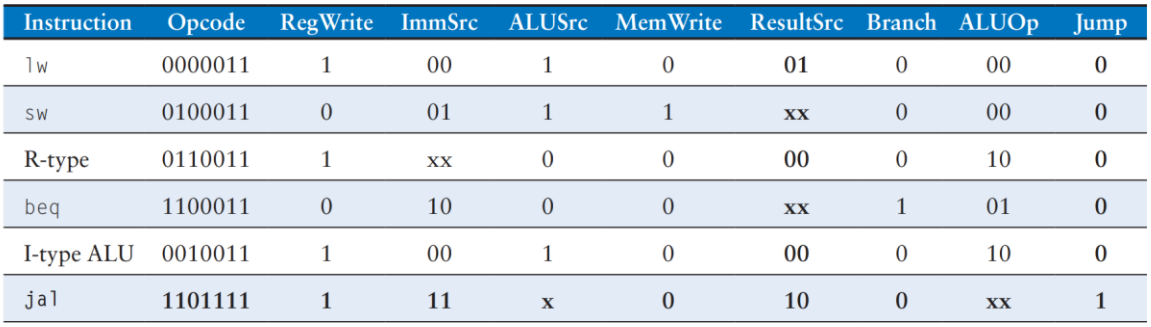
模块的端口及部分内部信号描述如 Table 1 所示。

Table 1

Name	Direction	Bits	Description
clk	I	1	处理器的时钟域
reset	I	1	Active high, 异步复位输入
PCF	B	32	程序计数器，用于指令存储器的地址
InstrF	B	32	机器指令
ALUResultM	B	32	ALU 运算结果
WriteDataM	B	32	数据存储器写数据
ReadDataM	B	32	数据存储器读数据
MemWriteM	B	1	数据存储器写使能控制信号

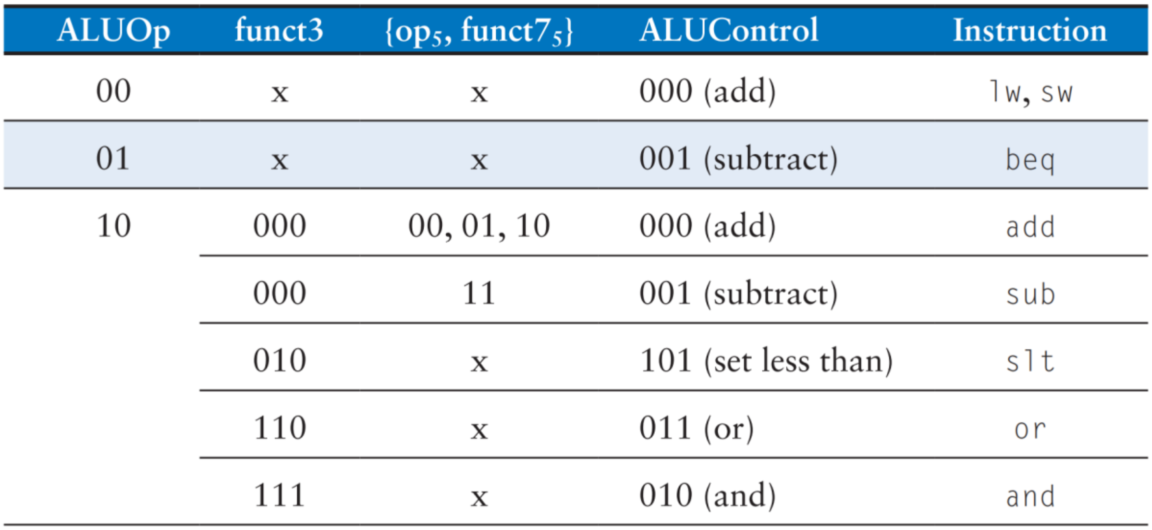
3 详细逻辑实现

Main Decoder:



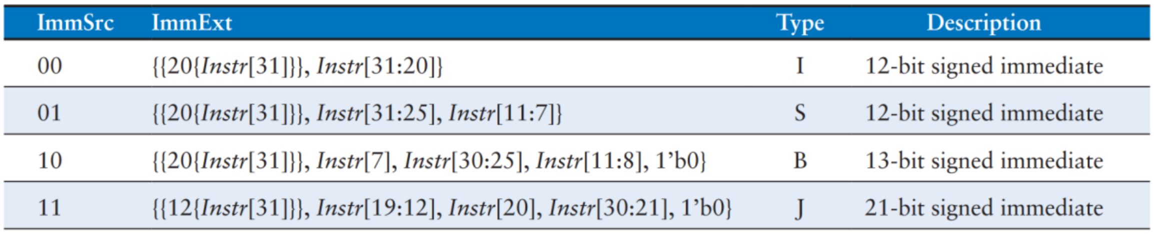
Instruction	Opcode	RegWrite	ImmSrc	ALUSrc	MemWrite	ResultSrc	Branch	ALUOp	Jump
`lw`	0000011	1	00	1	0	01	0	00	0
`sw`	0100011	0	01	1	1	xx	0	00	0
R-type	0110011	1	xx	0	0	00	0	10	0
`beq`	1100011	0	10	0	0	xx	1	01	0
I-type ALU	0010011	1	00	1	0	00	0	10	0
`jal`	**1101111**	**1**	**11**	**x**	**0**	**10**	**0**	**xx**	**1**

ALU Decoder



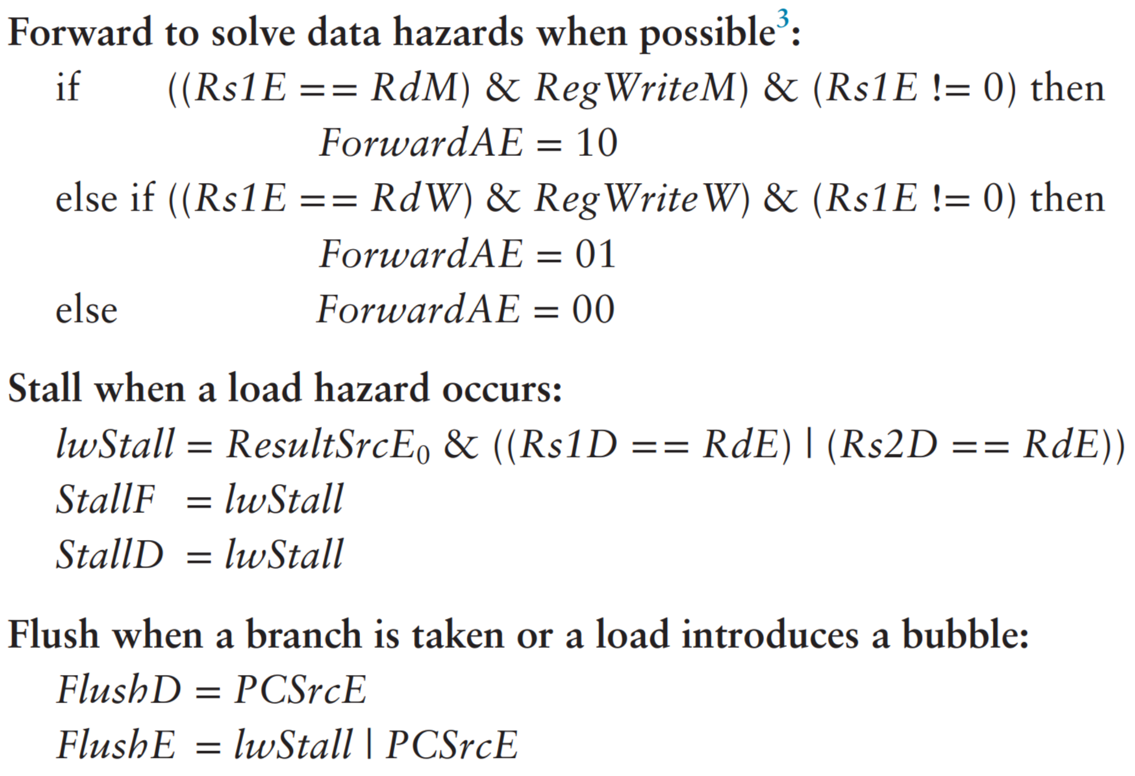
ALUOp	funct3	{op_5, $funct7_5$}	ALUControl	Instruction
00	x	x	000 (add)	`lw`, `sw`
01	x	x	001 (subtract)	`beq`
10	000	00, 01, 10	000 (add)	`add`
	000	11	001 (subtract)	`sub`
	010	x	101 (set less than)	`slt`
	110	x	011 (or)	`or`
	111	x	010 (and)	`and`

Sign Extension:



ImmSrc	ImmExt	Type	Description
00	{{20{*Instr*[31]}}, *Instr*[31:20]}	I	12-bit signed immediate
01	{{20{*Instr*[31]}}, *Instr*[31:25], *Instr*[11:7]}	S	12-bit signed immediate
10	{{20{*Instr*[31]}}, *Instr*[7], *Instr*[30:25], *Instr*[11:8], 1'b0}	B	13-bit signed immediate
11	{{12{*Instr*[31]}}, *Instr*[19:12], *Instr*[20], *Instr*[30:21], 1'b0}	J	21-bit signed immediate

Hazard Unit:



Forward to solve data hazards when possible[3]**:**

```
if        $((Rs1E == RdM)\ \&\ RegWriteM)\ \&\ (Rs1E\ != 0)$ then
                    $ForwardAE = 10$
else if   $((Rs1E == RdW)\ \&\ RegWriteW)\ \&\ (Rs1E\ != 0)$ then
                    $ForwardAE = 01$
else                $ForwardAE = 00$
```

Stall when a load hazard occurs:

$$lwStall = ResultSrcE_0\ \&\ ((Rs1D == RdE)\ |\ (Rs2D == RdE))$$

$$StallF = lwStall$$

$$StallD = lwStall$$

Flush when a branch is taken or a load introduces a bubble:

$$FlushD = PCSrcE$$

$$FlushE = lwStall\ |\ PCSrcE$$

4 配置寄存器

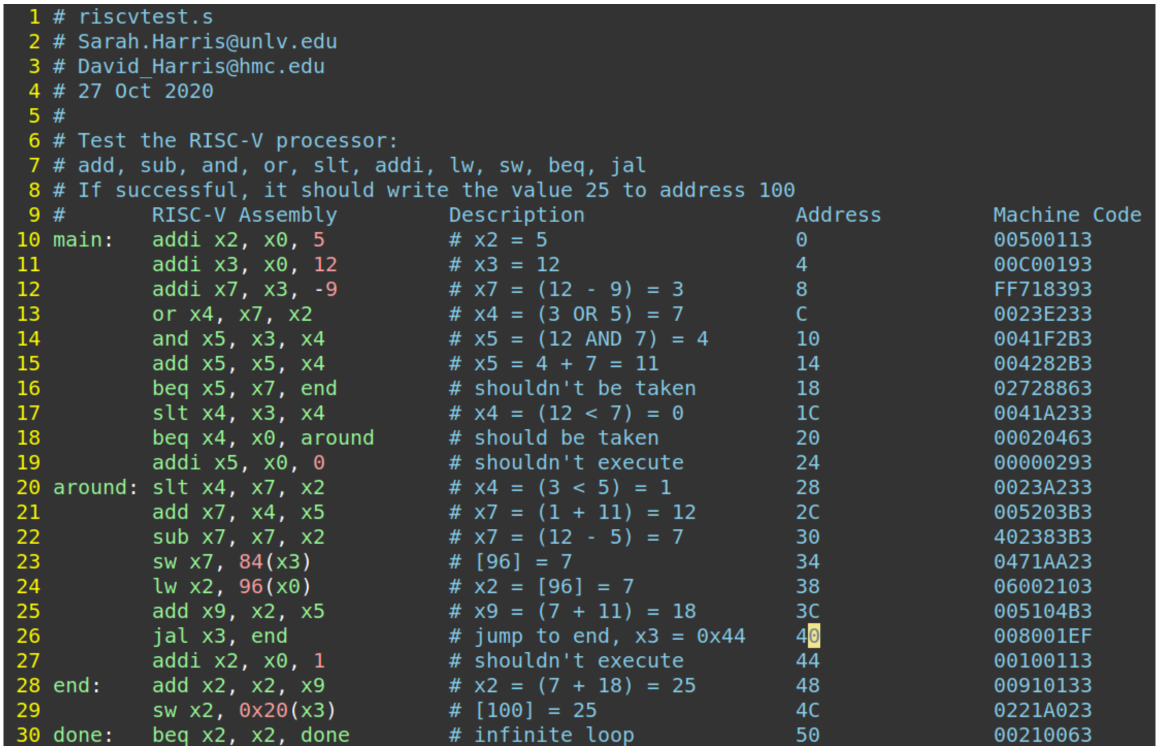
寄存器堆的寄存器号及用途见 Table 2。

Table 2

Name	Register Number	Use
zero	x0	Constant value 0
ra	x1	Return Address
sp	x2	Stack pointer
gp	x3	Global pointer
tp	x4	Thread pointer
t0~2	x5~7	Temporory registers
s0/fp	x8	Saved register / Frame pointer
s1	x9	Saved registers
a0~1	x10~11	Function argument / Return values
a2~7	x12~17	Function arguments
s2~11	x18~27	Saved registers
t3~6	x28~31	Temporary registers

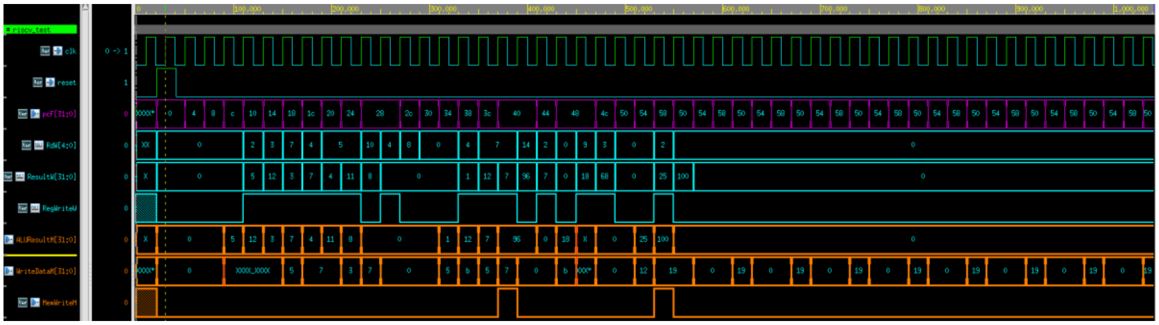
5 测试结果

测试指令序列如下：



```
# riscvtest.s
# Sarah.Harris@unlv.edu
# David_Harris@hmc.edu
# 27 Oct 2020
#
# Test the RISC-V processor:
# add, sub, and, or, slt, addi, lw, sw, beq, jal
# If successful, it should write the value 25 to address 100
#       RISC-V Assembly         Description                 Address         Machine Code
main:   addi x2, x0, 5          # x2 = 5                    0               00500113
        addi x3, x0, 12         # x3 = 12                   4               00C00193
        addi x7, x3, -9         # x7 = (12 - 9) = 3         8               FF718393
        or x4, x7, x2           # x4 = (3 OR 5) = 7         C               0023E233
        and x5, x3, x4          # x5 = (12 AND 7) = 4       10              0041F2B3
        add x5, x5, x4          # x5 = 4 + 7 = 11           14              004282B3
        beq x5, x7, end         # shouldn't be taken        18              02728863
        slt x4, x3, x4          # x4 = (12 < 7) = 0         1C              0041A233
        beq x4, x0, around      # should be taken           20              00020463
        addi x5, x0, 0          # shouldn't execute         24              00000293
around: slt x4, x7, x2          # x4 = (3 < 5) = 1          28              0023A233
        add x7, x4, x5          # x7 = (1 + 11) = 12        2C              005203B3
        sub x7, x7, x2          # x7 = (12 - 5) = 7         30              402383B3
        sw x7, 84(x3)           # [96] = 7                  34              0471AA23
        lw x2, 96(x0)           # x2 = [96] = 7             38              06002103
        add x9, x2, x5          # x9 = (7 + 11) = 18        3C              005104B3
        jal x3, end             # jump to end, x3 = 0x44    40              008001EF
        addi x2, x0, 1          # shouldn't execute         44              00100113
end:    add x2, x2, x9          # x2 = (7 + 18) = 25        48              00910133
        sw x2, 0x20(x3)         # [100] = 25                4C              0221A023
done:   beq x2, x2, done        # infinite loop             50              00210063
```

仿真波形如下：



基本与参考结果一致。