

# 2023 “龙芯杯” 个人赛设计报告

学校：安徽新华学院  
姓名：戴伟龙

## 目录

一、设计简介 .....	1
二、设计方案 .....	1
(一) 总体设计思路 .....	1
(二) SIF 取值模块设计 (state_if) .....	2
(三) SID 译码模块设计 (state_id) .....	3
(四) SEX 执行模块设计 (state_ex) .....	4
(五) SMEM 访存模块设计 (state_mem) .....	5
(六) D_BUFF5 缓存模块设计 .....	6
(七) regfile 通用寄存器模块设计 .....	6
(八) Pipe_ctrl 流水线控制模块 .....	7
(九) bus_ctrl 总线控制模块 .....	7
三、设计结果 .....	9
(一) 设计交付物说明 .....	9
(二) 设计演示结果 .....	10
四、参考设计说明 .....	12
五、参考文献 .....	12

## 表目录

支持指令表 1	支持指令表 .....	1
模块信号表 1	DCPU 模块端口信号 .....	2
模块信号表 2	取值模块端口信号.....	3
模块信号表 3	译码模块端口信号 .....	3
模块信号表 6	执行模块端口信号.....	4
模块信号表 4	访存模块端口信号.....	5
模块信号表 5	缓存模块端口信号.....	6
模块信号表 7	通用寄存器模块端口信号.....	6
模块信号表 8	流水线控制端口信号.....	7
模块信号表 9	总线控制模块端口信号.....	8
测试得分表 1	功能测试得分表 .....	10
测试得分表 2	性能测试运行时间表 .....	11

## 图目录

模块结构图 1	DCPU 模块结构图 .....	2
性能测试图 1	STREAM .....	11
性能测试图 2	MATRIX .....	12
性能测试图 3	CRYPTONIGHT .....	12

# 一、设计简介

该设计是针对于 2023 年“龙芯杯”计算机系统能力培养大赛个人赛所设计的一个 32 位 MIPS CPU，所设计的 CPU 采用了顺序单发射机制和经典的五级流水结构，可以运行实现 8 条算术运算指令、7 条逻辑运算指令、6 条移位指令、10 条分支跳转指令、4 条访存指令，共计 35 条指令（具体指令见支持指令表 1）。

本次设计的 CPU 在所有指令功能测试和性能测试满分通过的基础上，最高可以支持 65MHz 的时钟频率，且 WNS、TNS 都为正值，满足时序要求。在数据相关问题上采用数据前推的方式进行解决，在 load 相关问题和指令数据访存冲突时通过暂停流水线进行解决。同时设计了一个 32 位的超前进位加法器用于加减法的运算。

支持指令表 1 支持指令表

指令类型	指令内容
算术运算指令	ADD 、ADDU、ADDI 、ADDIU、SUB 、SLT 、SLTU 、MUL
逻辑运算	AND、ANDI、OR、ORI、XOR、XORI、LUI
移位指令	SLL、SLLV、SRL、SRLV、SRA、SRAV
分支跳转指令	BEQ、BNE、BGTZ、BLTZ、BGEZ、BLEZ、J、JAL、JR、JALR
访存指令	SB、SW、LB、LW

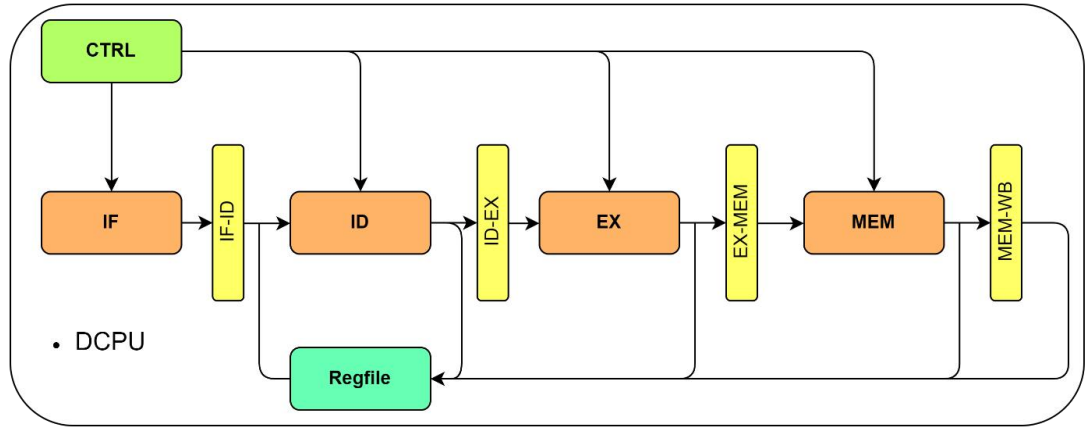
# 二、设计方案

## （一）总体设计思路

**总体设计：**所设计的 CPU 采用了顺序单发射机制和经典的五级流水结构，主要有两个模块构成：

- 1.DCPU 模块，用于实现一个 32 位的 MIPS CPU；
- 2.bus\_ctrl，用于实现 SRAM 内存控制和串口控制。

**DCPU 模块设计：**该模块主要由 SIF 取值模块、SID 译码模块、SEX 执行模块、SMEM 访存模块、regfile 寄存器模块和 pipe\_ctrl 流水线控制模块构成，还有用于 SIF-SID、SID-SEX、SEX-SMEM、SMEM-SWB 暂存的 D\_BUFF5。



DCPU 模块结构图

模块信号表 1 DCPU 模块端口信号

端口	位宽	含义
clk	1	输入：时钟信号
rst	1	输入：复位信号
rom_addr	32	输出：指令访存地址
rom_ce_n	1	输出：rom 使能信号
rom_data	32	输入：来自程序 rom 的数据
rom_stall	1	输入：是否应为空指令
ram_addr	32	输出：数据读写地址
ram_wdata	32	输出：写数据
ram_rdata	32	输入：sram 读出数据
ram_be_n	4	输出：数据存储区的字节使能
ram_oe_n	1	输出：读使能
ram_we_n	1	输出：写使能

## (二) SIF 取值模块设计(state\_if)

主要功能：

1. 从 ROM 中获取指令信息
2. 对 PC 值进行修改 (+4, 各种跳转指令)

模块信号表 2 取值模块端口信号

端口	位宽	含义
clk	1	输入：时钟信号
rst	1	输入：复位信号
jump_pc	32	输入：来自 ID 跳转结果
jump_signal	1	输入：来自 ID 跳转信号
o_pc	32	输出：pc 的最终结果
o_inst	32	输出：获取的指令
o_nop	1	输出：是否为空指令
rom_data_i	32	输入：来自程序 rom 的数据
rom_stall	1	输入：是否为空指令
rom_addr_o	32	输出：rom 的地址
rom_ce_n_o	1	输出：rom 使能信号
stall_pipe	1	输入：暂停流水线(pipe_ctrl)
stop_signal	1	输出：暂停流水线信号

### (三) SID 译码模块设计(state\_id)

主要功能：

1. 对指令进行解码：
  - a. 产生每条指令的控制信号
  - b. 产生执行阶段运算数和运算类型
2. 访问寄存器组

模块信号表 3 译码模块端口信号

端口	位宽	含义
i_pc	32	输入：来自 IF 阶段的 pc 值
i_inst	32	输入：来自 IF 阶段的指令
i_nop	1	输入：来自 IF 阶段的空信号

alu_data1	32	输出：操作数 1
alu_data2	32	输出：操作数 2
vir_addr	32	输出：虚地址
waddr	5	输出：写地址
rs_addr	5	输出：读寄存器的地址 1
rt_addr	5	输出：读寄存器的地址 2
rs_data	32	输入：读出寄存器的数据 1
rt_data	32	输入：读出寄存器的数据 2
load	1	输入：LOAD 相关信号
jump_pc,	32	输出：跳转结果
jump_signal	1	输出：跳转信号
stop_signal	1	输出：暂停流水线信号
opset_o	38	输出：存有 35 条指令+3 位用于执行阶段对运算指令进行区分的选择端

#### （四）SEX 执行模块设计(state\_ex)

主要功能：

1. 完成对各类运算的计算
2. 传输 mem 阶段的信号和总线控制的信号

其中包含一个 ALU 运算子模块实现对输入的操作数据根据运算类型进行运算，获取结果，并设计了一个 32 位的超前进位加法器。

模块信号表 4 执行模块端口信号

端口	位宽	含义
alu_din1	32	输入：来自译码阶段的源操作数 1
alu_din2	32	输入：来自译码阶段的源操作数 2
id_vir_addr	32	输入：来自译码阶段的虚地址
id_waddr	5	输入：来自译码阶段的写地址
alu_dout	32	输出：运算结果

is_lop	1	输出：是否是 LB 或 LW 指令
ex_waddr	5	输出：执行阶段的写地址
vir_addr	32	输出：虚地址
toram_wdata	32	输出：输向 ram 的写数据
toram_be_n	1	输出：输向 ram 的字节使能
toram_oe_n	1	输出：输向 ram 的读使能
toram_we_n	1	输出：输向 ram 的写使能
stop_signal	1	输出：暂停流水线信号
opset_i	38	输入：来自译码模块的指令信号
opset_o	38	输出：输向访存模块的指令信号

### (五) SMEM 访存模块设计(state\_mem)

主要功能：完成对数据存储器的读写操作

模块信号表 5 访存模块端口信号

端口	位宽	含义
ex_aluresult	32	输入：来自执行阶段的运算结果
vir_addr,	32	输入：来自执行阶段的虚地址
ex_waddr,	5	输入：来自执行阶段的写地址
wb_wdata,	32	输出：输向 regfile 的写数据
wb_waddr,	5	输出：输向 regfile 的写地址
mem_is_op,	1	输出：是否是 LB 或 LW 指令
mem_wdata,	32	输出：写数据
toram_rdata,	32	输入：来自 ram 的读数据
stop_signal,	1	输出：暂停流水线信号
opset_i	38	输入：来自执行模块的指令信号

## (六) D\_BUFF5 缓存模块设计

主要功能：

1. 当每一时钟周期的上升沿到来时，将上一阶段的结果传递到下一阶段
2. 执行是否暂停流水线

模块信号表 6 缓存模块端口信号

端口	位宽	含义
clk	1	输入：时钟信号
rst	1	输入：复位信号
stall_pipe	2	输入：来自 ctrl 模块的流水线暂停信号
din*	x	输入：来自上一级段的信号、结果
dout*	x	输出：输向下一阶段的信号、结果
opset_i	38	输入：来自上一级段的指令信号
opset_o	38	输出：输向下一阶段的指令信号

此处 din\*, dout\*是指不同位宽的输入和输出，作用相同，只有位宽不同。该模块分别例化被为 sif\_sid、sid\_sex、sex\_smem、smem\_swb，在每个例化模块中调用的输入输出信号并不相同。

## (七) regfile 通用寄存器模块设计

主要功能：实现的是 2 个读端口，3 个写端口的 32 个通用寄存器的寄存器组，对 0 地址始终输出 0，同时在此会解决 load 相关，与及当写入地址和读地址相同时，读出的数据将会是此时写入的数据

模块信号表 7 通用寄存器模块端口信号

端口	位宽	含义
clk	1	输入：时钟信号
rst	1	输入：复位信号
rs_addr	5	输入：来自译码阶段的读寄存器 1
rt_addr	5	输入：来自译码阶段的读寄存器 2
rs_data	32	输出：输往译码阶段的读数据 1



rt_data	32	输出：输往译码阶段的读数据 2
stop_pipe	1	输出：暂停流水线信号，解决 LOAD 相关
waddr	5	输入：写回阶段的写地址
wdata	32	输入：写回阶段的写数据
mem_waddr	5	输入：访存阶段的写地址
mem_wdata	32	输入：访存阶段的写数据
mem_is_op	1	输入：是否是 LW，LB 指令
ex_waddr	5	输入：执行阶段的写地址
ex_wdata	32	输入：执行阶段的写数据
ex_is_lop	1	输入：是否是 LW，LB 指令

## （八）Pipe\_ctrl 流水线控制模块

主要功能：处理来自取值、译码、执行、访存模块的暂停信号，判断是否需要使流水线暂停并发出流水线暂停信号。

模块信号表 8 流水线控制模块端口信号

端口	位宽	含义
stop_signal	4	输入：每一位分别来自取值、译码、执行、访存模块
stall_pipe	5	输出：输往各个 D_BUFF5 模块，使流水线暂停

## （九）bus\_ctrl 总线控制模块

主要功能：

1. SRAM 内存控制
2. 串口控制

模块信号表 9 总线控制模块端口信号

端口	位宽	含义
clk	1	输入：时钟信号
rst	1	输出：复位信号
if_rom_addr	32	输入：来自取值模块的数据
if_rom_ce_n	1	输入：来自取值模块的使能信号
if_rom_data	32	输出：读出的数据
rom_stall	1	输出：输往取值模块的空指令信息
ex_ram_addr	32	输入：来自执行阶段的虚地址
ex_ram_wdata	32	输入：来自执行阶段的写入数据
mem_ram_rdata	32	输出：输往访存阶段的读数据
ex_ram_be_n	1	输入：来自执行阶段的字节使能
ex_ram_oe_n	1	输入：来自执行阶段的读使能
ex_ram_we_n	1	输入：来自执行阶段的写使能
base_ram_data	32	BaseRAM 数据
base_ram_addr	20	BaseRAM 地址
base_ram_be_n	1	BaseRAM 字节使能，低有效
base_ram_ce_n	1	BaseRAM 片选，低有效
base_ram_oe_n	1	BaseRAM 读使能，低有效
base_ram_we_n	1	ExtRAM 写使能，低有效
ext_ram_data	32	ExtRAM 数据
ext_ram_addr	20	ExtRAM 地址
ext_ram_be_n	1	ExtRAM 字节使能，低有效
ext_ram_ce_n	1	ExtRAM 片选，低有效
ext_ram_oe_n	1	ExtRAM 读使能，低有效
ext_ram_we_n	1	ExtRAM 写使能，低有效
txd	1	ExtRAM 数据
rxid	1	ExtRAM 地址

### 三、设计结果

#### (一) 设计交付物说明

本设计交付源代码的 thinpad\_top.srcs 目录层次如下：

```
thinpad_top.srcs
├── constrs_1 //约束文件，添加了部分输入的时序延时，
│   └── new
│       └── thinpad_top.xdc //管教约束文件，来源于发布包
├── sim_1 //使用发布包的仿真文件，未列出
│   └── new
└── sources_1
    ├── ip //所调用的 ip 核(通过调参，vivado 自动生成)
    │   └── pll_example //时钟分频模块, 分频为 65MHz
    └── new //设计源文件
        ├── ADD_SUB.v //超前进位加法器
        ├── ALUModule.v //ALU 运算模块
        ├── async.v //串口实例化模块
        ├── bus_ctrl.v //总线控制模块
        ├── DEFINES.v //宏定义
        ├── D_BUFF5.v //缓存暂存模块(IF-ID、ID-EX、EX-MEM、MEM-WB)
        ├── D_CPU.v //所设计的 DCPU 顶层模块
        ├── pipe_ctrl.v //流水线控制模块
        ├── regfile.v //寄存器模块
        ├── SEG7_LUT.v //数码管模块（验证板卡是否启动，与 CPU 无关）
        ├── state_ex.v //执行模块
        ├── state_id.v //译码模块
        ├── state_if.v //取值模块
        ├── state_mem.v //访存模块
        ├── thinpad_top.v //顶层文件
        └── vga.v //VGA 显示模块（验证板卡是否启动，与 CPU 无关）
```

提交文件在 master 分支下（设计报告也在该分支下）

master

2023295 / +

Find file

Web IDE

↓

Clone

README

CI/CD configuration

Add LICENSE

Add CHANGELOG

Add CONTRIBUTING

Auto DevOps enabled

Add Wiki

Name	Last commit	Last update
.ci-scripts	FPGA template	3 months ago
asm	final0	3 hours ago
thinpad_top.srscs	Final0	4 hours ago
.gitignore	FPGA template	3 months ago
.gitlab-ci.yml	FPGA template	3 months ago
.gitmodules	FPGA template	3 months ago
README.md	FPGA template	3 months ago
thinpad_top.xpr	test1	3 days ago
大赛设计报告.pdf	Final	3 hours ago

CPU 各个模块所在路径

master

2023295 / thinpad\_top.srscs / sources\_1 / new / +

History

Find file

Web IDE

↓

Clone

Name	Last commit	Last update
..		
ADD_SUB.v	test0	3 days ago
ALUModule.v	test1	3 days ago
DEFINES.v	test0	3 days ago
D_BUFF5.v	test2	1 day ago
D_CPU.v	test3	22 hours ago
SEG7_LUT.v	test0	3 days ago
async.v	test0	3 days ago
bus_ctrl.v	Final0	4 hours ago
pipe_ctrl.v	test2	1 day ago
regfile.v	test2	1 day ago
state_ex.v	test2	1 day ago
state_id.v	test2	1 day ago

(二) 设计演示结果

测试得分表 1 功能测试得分表

	得分
一级测试	100

二级测试	100
三级测试	100
性能测试	100

测试得分表 2 性能测试运行时间表

	运行时间
STREAM	0.097 s
MATRIX	0.151 s
CRYPTONIGHT	0.387 s
总用时	0.635 s

100

perf 在 FPGA 板

3006 3003 3012

上的结果

```

=== Test STREAM ===
Boot message: 'MONITOR for MIPS32 - initialized.'
User program written
Program Readback:
1080043c4080053c3000063c21308600050086100400a5240000828cfcffa2acfl
Program memory content verified
Data memory content verified
Test STREAM run for 0.097s

```

性能测试图 1 STREAM

100

### perf 在 FPGA 板 3006 3003 3012 上的结果

```
=== Test MATRIX ===  
Boot message: 'MONITOR for MIPS32 - initialized.'  
User program written  
Program Readback:  
4080043c4180053c4280063c60000724251800001a00671080400300405203002:  
Program memory content verified  
Data memory content verified  
Test MATRIX run for 0.151s
```

性能测试图 2 MATRIX

100

### perf 在 FPGA 板 3006 3003 3012 上的结果

```
=== Test CRYPTONIGHT ===  
Boot message: 'MONITOR for MIPS32 - initialized.'  
User program written  
Program Readback:  
4080043cadde053cefbea534cefa063c0cb0c6341000073c25180400251000000:  
Program memory content verified  
Data memory content verified  
Test CRYPTONIGHT run for 0.387s
```

性能测试图 3 CRYPTONIGHT

## 四、参考设计说明

1. DCPU 的流水线框借鉴了《自己动手写 CPU》.(雷思磊)的五级流水框架
2. SRAM 内存控制参考了 2022 “龙芯杯” 个人赛的开源代码  
<https://github.com/ghj1222/cpu-for-nscsc2022.git>

## 五、参考文献

- [1] 雷思磊. 自己动手写 CPU. 电子工业出版社
- [2] 杨全胜. CPU 设计实践教程: 从数字电路到计算机组成. 清华大学出版社