

# MIPS 大作业

范元棋 2019211380

## 一. 完成度和代码说明

### 功能和结构

5 级流水线完整结构

数据相关：

- Idex/exmem 数据相关检测和处理。
- Idex/memwb 数据相关检测和处理。
- Writeback/readreg 数据相关检测和处理。
- Load-use 数据相关检测和处理，触发后流水线的空拍插入。

分支跳转，寄存器冲刷：

- Branch 提前分支预测
- Branch 和 j jal jr 指令的跳转，冲刷流水线处理

控制逻辑的布尔逻辑提取（非 case 语句）

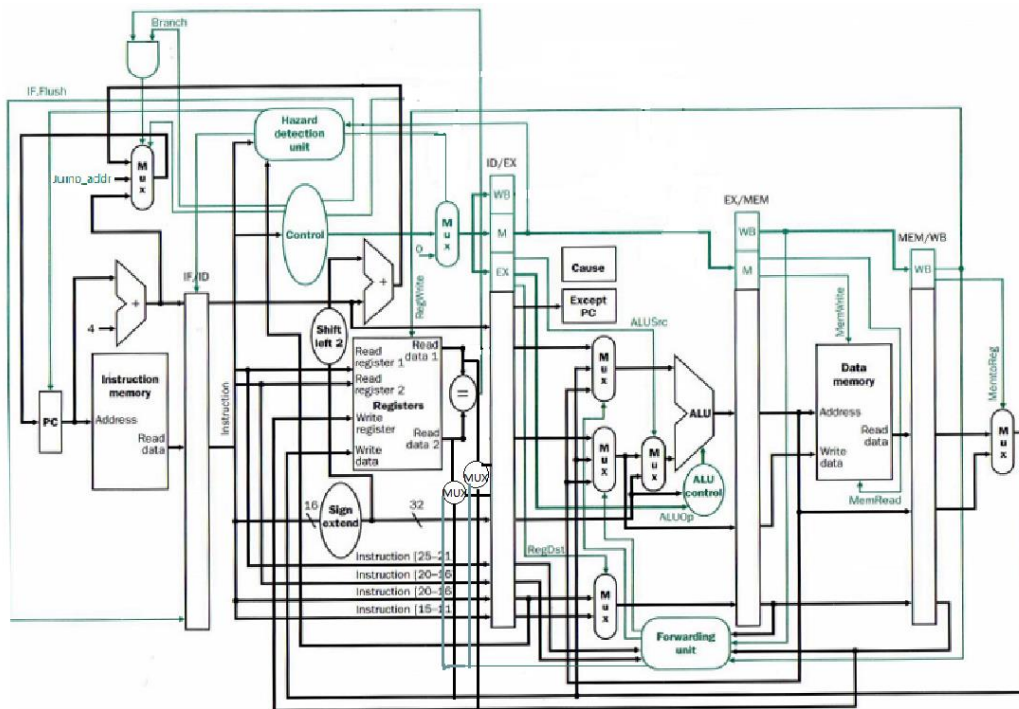
### 支持的指令

- lw sw
- add sub and xor or slt
- addi addiu andi ori xori lui sll srl
- beq bne
- j jal
- jr

### 代码说明

tb.v	Test 文件
mipscore.v	顶层设计
pc_next_mux.v	pc 更新逻辑，多路选择 pc+4/branch 地址/jump 地址/jr 地址
Registers.v	寄存器堆
load_use_detect.v	Load-use 相关检测以及
ctrl.v	控制码和 aluop 的产生逻辑
cache.v	Cache，不参与综合，包含了汇编代码。
branch_detect.v	分支提前预测逻辑，进行 branch 指令是否跳转的判断
ALU_all.v	Alu 各个功能的实现
adder.v	加法器

## 二. 控制图和控制逻辑



IF.Flush	0	0	0	0	0	0
PcSrc	0	0	0	0	0	0
AlusrcB(	1	1	0	0	0	0
AlusrcA	0	0	0	0	0	0
Memrea	1	0	0	0	0	0
Memwri	0	1	0	0	0	0
Mento	1	x	0	0	0	0
RegDst	0	x	1	1	1	1
Regwrit	1	0	1	1	1	1
overflow	0	0	1	1	0	0
alu 操作	加法	加法	加法	减法	与	或
imm_ho	1	1	x	x	x	x
指令/控制码	lw	sw	add	sub	and	or

0	0	0	0	0	0	0	0	0	1/0	1/0
0	0	0	0	0	0	0	0	0	1/0	1/0
0	0	1	1	1	1	1	0	0	0	0
0	0	0	0	0	0	0	1	1	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	x	x
1	1	0	0	0	0	0	1	1	x	x
1	1	1	1	1	1	1	1	1	0	0
0	0	1	0	0	0	0	0	0	0	0
xor	比较	加法	加法	与	或	xor	移位<fix	移位<<	移位>>	x
x	x	1	1	0	0	0	1	x	x	x
xor	slt	addi	addiu	andi	ori	xori	lui	sll	srl	beq
										bne

j	x	x	0	0	x	x	0	0	x	x	2	1
jai	x	x	0	1	2 (31)	2 (pc in	0	0	x	x	2	1
jr	x	x	0	0	x	x	0	0	x	x	3	1

● 布尔逻辑:

Overflowck (是否进行溢出检测)	$(\text{inst}[29] \& (\sim \text{inst}[28]) \& (\sim \text{inst}[27]) \& (\sim \text{inst}[26])) \mid ((\sim \text{inst}[29]) \& (\sim \text{inst}[28]) \& (\sim \text{inst}[27]) \& (\sim \text{inst}[26])) \& ((\text{inst}[5] \& (\sim \text{inst}[3]) \& (\sim \text{inst}[2])))$ );
regwrite	$\sim((\text{inst}[31] \& \text{inst}[29]) \mid ((\sim \text{inst}[31]) \& (\sim \text{inst}[30]) \& (\sim \text{inst}[29]) \& \text{inst}[28]) \mid ((\sim \text{inst}[28]) \& \text{inst}[27] \& (\sim \text{inst}[26])) \mid ((\sim \text{inst}[29]) \& (\sim \text{inst}[28]) \& (\sim \text{inst}[27]) \& (\sim \text{inst}[26])) \& ((\sim \text{inst}[5]) \& (\text{inst}[3]) \& (\sim \text{inst}[2])))$ );
regdst[0]	$(\sim \text{inst}[29]) \& (\sim \text{inst}[28]) \& (\sim \text{inst}[27]) \& (\sim \text{inst}[26]);$
regdst[1]	$(\sim \text{inst}[31]) \& (\sim \text{inst}[29]) \& (\text{inst}[27]) \& (\text{inst}[26]);$
memtoreg[0]	$(\text{inst}[31]) \& (\sim \text{inst}[29]);$
memtoreg[1]	$(\sim \text{inst}[31]) \& (\sim \text{inst}[29]) \& (\sim \text{inst}[28]) \& (\text{inst}[26]);$
memwrite	$(\text{inst}[31]) \& (\sim \text{inst}[30]) \& (\text{inst}[29]);$
memread	$(\text{inst}[31]) \& (\sim \text{inst}[29]);$
alusrcA	$((\sim \text{inst}[29]) \& (\sim \text{inst}[28]) \& (\sim \text{inst}[27]) \& (\sim \text{inst}[26])) \& ((\sim \text{inst}[5]) \& (\sim \text{inst}[3]))$
alusrcB	$((\text{inst}[31])) \mid ((\sim \text{inst}[31]) \& (\sim \text{inst}[30]) \& (\text{inst}[29]))$ );
pcsrc[0]	$((\sim \text{inst}[29]) \& (\text{inst}[28]) \& (\sim \text{inst}[27])) \mid ((\sim \text{inst}[29]) \& (\sim \text{inst}[28]) \& (\sim \text{inst}[27]) \& (\sim \text{inst}[26])) \& ((\sim \text{inst}[5]) \& (\text{inst}[3]))$
pcsrc[1]	$((\sim \text{inst}[31]) \& (\sim \text{inst}[28]) \& (\text{inst}[27])) \mid ((\sim \text{inst}[29]) \& (\sim \text{inst}[28]) \& (\sim \text{inst}[27]) \& (\sim \text{inst}[26])) \& ((\sim \text{inst}[5]) \& (\text{inst}[3]));$
aluop[0]	$((\sim \text{inst}[29]) \& (\sim \text{inst}[28]) \& (\sim \text{inst}[27]) \& (\sim \text{inst}[26])) \& ((\sim \text{inst}[5]) \& (\sim \text{inst}[3]) \& (\sim \text{inst}[2]) \& (\sim \text{inst}[1])) \mid ((\sim \text{inst}[29]) \& (\sim \text{inst}[28]) \& (\sim \text{inst}[27]) \& (\sim \text{inst}[26])) \& ((\text{inst}[2]) \& (\sim \text{inst}[1]) \& (\text{inst}[0])) \mid ((\sim \text{inst}[29]) \& (\sim \text{inst}[28]) \& (\sim \text{inst}[27]) \& (\sim \text{inst}[26])) \& ((\text{inst}[5]) \& (\sim \text{inst}[3]) \& (\sim \text{inst}[2]) \& (\text{inst}[1])) \mid (((\sim \text{inst}[29]) \& (\sim \text{inst}[28]) \& (\sim \text{inst}[27]) \& (\sim \text{inst}[26])) \& ((\text{inst}[5]) \& (\text{inst}[3]))) \mid ((\text{inst}[29]) \& (\text{inst}[28]) \& (\sim \text{inst}[27]) \& (\text{inst}[26]));$
aluop[1]	$((\sim \text{inst}[29]) \& (\sim \text{inst}[28]) \& (\sim \text{inst}[27]) \& (\sim \text{inst}[26])) \& ((\text{inst}[2]) \& (\sim \text{inst}[1])) \mid ((\sim \text{inst}[29]) \& (\sim \text{inst}[28]) \& (\sim \text{inst}[27]) \& (\sim \text{inst}[26])) \& ((\sim \text{inst}[5]) \& (\sim \text{inst}[3]) \& (\sim \text{inst}[2]) \& (\sim \text{inst}[1])) \mid ((\sim \text{inst}[31]) \& (\text{inst}[29]) \& (\text{inst}[28]) \& (\sim \text{inst}[27])) \mid ((\text{inst}[29]) \& (\text{inst}[28]) \& (\text{inst}[27]) \& (\text{inst}[26]));$

aluop[2]	( ( (~inst[29]) & (~inst[28]) & (~inst[27]) & (~inst[26])) & (( inst[5])&( inst[2])&( inst[1])&(~ inst[0]))  (( (~inst[29]) & (~inst[28]) & (~inst[27]) & (~inst[26])) & (( inst[5])&( inst[3])&( inst[1])&(~ inst[0]))  (( (~inst[29]) & (~inst[28]) & (~inst[27]) & (~inst[26])) & (( ~inst[5])&(~ inst[3])&(~ inst[1]))  (( inst[29])&( inst[28])&( inst[27])));
aluop[3]	((~inst[29]) & (~inst[28]) & (~inst[27]) & (~inst[26])) & ((~ inst[5])&(~ inst[2])&( inst[1])&(~ inst[0]));
how_imm (立即数的扩展方式: 0 扩展或者符号位扩展)	~((~(inst[27] & inst[26])) & inst[29] & inst[28]);

### 三. AES 汇编

汇编代码实现了密钥扩展和 aes 算法, 详细见附件。

Data\_cache 中, 数据存储如下

0-255 : s 盒空间

258 : 数据初始地址

259-274 : 数据空间

275-450 : 密钥空间 其中 275-290 为第一组, 由密钥扩展产生的密钥依次向后存储

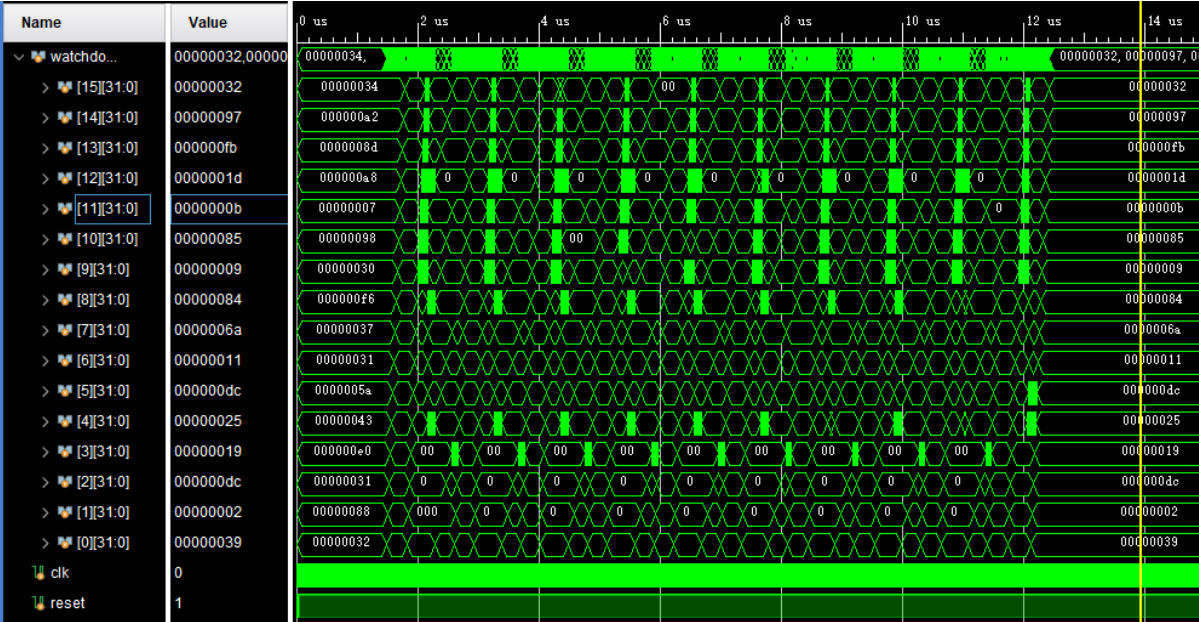
451 : 密钥初始地址

452 -462 : Rcon

### 四. 仿真结果

密钥扩展消耗 450 周期

加密 aes 消耗 5463 周期



## 五. 综合结果

**面积:** Total cell area: ..... 242710.779196

**延时:**

data required time: ..... 2.29

data arrival time ..... -2.29

slack (MET) ..... 0.00

**功耗:** Total Dynamic Power = 68.3781 mW

**面积延时乘积:** 242710.779196\*2.29=555807.684

## 附:

### 时间报告

Point	Incr	Path		
clock clk (rise edge)	0.00	0.00		
clock network delay (ideal)	0.00	0.00		
exmem_rt_reg[0]/CK (DFFRHQX4)		0.00 #	0.00 r	
exmem_rt_reg[0]/Q (DFFRHQX4)		0.20	0.20 r	
U6585/Y (NAND2BX4)	0.11	0.31 r		
U6642/Y (NOR3X4)	0.05	0.36 f		
U6644/Y (NOR2X4)	0.09	0.45 r		
U6662/Y (NAND2X4)	0.04	0.49 f		
U6567/Y (INVX3)	0.05	0.54 r		
U6615/Y (NAND2X4)	0.04	0.58 f		
U6641/Y (NAND2X2)	0.08	0.67 r		
U6521/Y (CLKINX8)	0.06	0.73 f		
U6572/Y (INVX8)	0.07	0.80 r		
U6594/Y (NOR2X4)	0.04	0.84 f		
U6557/Y (NOR3X4)	0.11	0.95 r		
U6262/Y (MXI2X4)			0.08	1.02 f
alu_full/data2[8] (ALU_all)			0.00	1.02 f
alu_full/alu_full_inadder/b[8] (ALU_32bit_adder)			0.00	1.02 f
alu_full/alu_full_inadder/U49/Y (XOR2X4)			0.15	1.17 r
alu_full/alu_full_inadder/fst/b[8] (unsigned_adder_32bit_1)			0.00	1.17 r
alu_full/alu_full_inadder/fst/ad2/b[0] (adder4bits_6)		0.00	1.17 r	
alu_full/alu_full_inadder/fst/ad2/U8/Y (NAND2X4)	0.03	1.21 f		
alu_full/alu_full_inadder/fst/ad2/U24/Y (NAND2BX4)	0.10	1.31 f		
alu_full/alu_full_inadder/fst/ad2/U35/Y (CLKINX4)	0.04	1.35 r		
alu_full/alu_full_inadder/fst/ad2/U36/Y (NOR2X4)	0.03	1.38 f		
alu_full/alu_full_inadder/fst/ad2/U37/Y (NOR2X4)	0.06	1.44 r		
alu_full/alu_full_inadder/fst/ad2/U16/Y (NOR2X4)	0.04	1.48 f		
alu_full/alu_full_inadder/fst/ad2/U38/Y (OAI2BB1X4)	0.05	1.52 r		
alu_full/alu_full_inadder/fst/ad2/cio (adder4bits_6)	0.00	1.52 r		
alu_full/alu_full_inadder/fst/ad3/ci (adder4bits_5)	0.00	1.52 r		
alu_full/alu_full_inadder/fst/ad3/U31/Y (OAI2BB1X4)	0.10	1.62 r		
alu_full/alu_full_inadder/fst/ad3/cio (adder4bits_5)	0.00	1.62 r		
alu_full/alu_full_inadder/fst/ad4/ci (adder4bits_4)	0.00	1.62 r		
alu_full/alu_full_inadder/fst/ad4/U33/Y (OAI2BB1X4)	0.10	1.72 r		
alu_full/alu_full_inadder/fst/ad4/cio (adder4bits_4)	0.00	1.72 r		
alu_full/alu_full_inadder/fst/ad5/ci (adder4bits_3)	0.00	1.72 r		
alu_full/alu_full_inadder/fst/ad5/U30/Y (OAI2BB1X4)	0.10	1.82 r		
alu_full/alu_full_inadder/fst/ad5/cio (adder4bits_3)	0.00	1.82 r		
alu_full/alu_full_inadder/fst/ad6/ci (adder4bits_2)	0.00	1.82 r		
alu_full/alu_full_inadder/fst/ad6/U33/Y (OAI2BB1X4)	0.11	1.92 r		
alu_full/alu_full_inadder/fst/ad6/cio (adder4bits_2)	0.00	1.92 r		
alu_full/alu_full_inadder/fst/ad7/ci (adder4bits_1)	0.00	1.92 r		
alu_full/alu_full_inadder/fst/ad7/U19/Y (OAI2BB1X4)	0.10	2.03 r		

alu_full/alu_full_inadder/fst/ad7/co[3] (adder4bits_1)	0.00	2.03 r
alu_full/alu_full_inadder/fst/co[31] (unsigned_adder_32bit_1)	0.00	2.03 r
alu_full/alu_full_inadder/U23/Y (INVX1)	0.03	2.06 f
alu_full/alu_full_inadder/U27/Y (NAND2X1)	0.05	2.11 r
alu_full/alu_full_inadder/U26/Y (NAND2X1)	0.04	2.15 f
alu_full/alu_full_inadder/U3/Y (MXI2X1)	0.07	2.22 r
alu_full/alu_full_inadder/overf (ALU_32bit_adder)	0.00	2.22 r
alu_full/overflow (ALU_all)	0.00	2.22 r
U6579/Y (AOI21XL)	0.06	2.29 f
exmem_regwrite_reg/D (DFFRX4)	0.00	2.29 f
data arrival time	2.29	
clock clk (rise edge)	2.44	2.44
clock network delay (ideal)	0.00	2.44
exmem_regwrite_reg/CK (DFFRX4)	0.00	2.44 r
library setup time	-0.15	2.29
data required time	2.29	
-----		
data required time	2.29	
data arrival time	-2.29	
-----		
slack (MET)	0.00	



### **功耗报告:**

Global Operating Voltage = 1.8

Power-specific unit information :

Voltage Units = 1V

Capacitance Units = 1.000000pf

Time Units = 1ns

Dynamic Power Units = 1mW (derived from V,C,T units)

Leakage Power Units = 1pW

Cell Internal Power = 65.9454 mW (96%)

Net Switching Power = 2.4326 mW (4%)

-----

Total Dynamic Power = 68.3781 mW (100%)

Cell Leakage Power = 647.8627 nW

### **面积报告**

Number of ports: 164

Number of nets: 1952

Number of cells: 1675

Number of references: 100

Combinational area: 137859.323851

Noncombinational area: 104851.455345

Net Interconnect area: undefined (No wire load specified)

Total cell area: 242710.779196

Total area: undefined