MIPS 处理器设计

微处理器结构与设计课程报告

2022-6-11

方佳豪
2021211066

一.课程任务

1. MIPS 处理器设计

基于 verilog 语言设计一款基于 5 段流水线的 MIPS 处理器，并用该处理器和对应的指令集运行自己编写的 AES 加密算法。按照实验必选和选做要求完成以下指令和功能。

1.1 支持的指令

A. 访存指令：lw，sw;

B. 算数逻辑指令：add，addi, addiu, sub，and，or，xor，andi，ori, xori, lui, slt，sll，srl 指令等;

C. 转移指令：beq，bne，j，jal, jr 指令等。

1.2 支持的功能

A. 支持数据相关检测处理（forwarding or bypass）;

B. 支持转移冒险处理（流水线冲刷）。

2. AES 加密算法

通过编写汇编代码完成 128 比特密钥长度下的 AES 加密功能。根据课程提供的 S 盒（aes_sbox.txt）和输入明文和密钥计算出正确的密文。

测试向量如下:

明文：32 43 f6 a8 88 5a 30 8d 31 31 98 a2 e0 37 07 34

密钥：2b 7e 15 16 28 ae d2 a6 ab f7 15 88 09 cf 4f 3c

密文：39 25 84 1d 02 dc 09 fb dc 11 85 97 19 6a 0b 32

3. 验证和综合

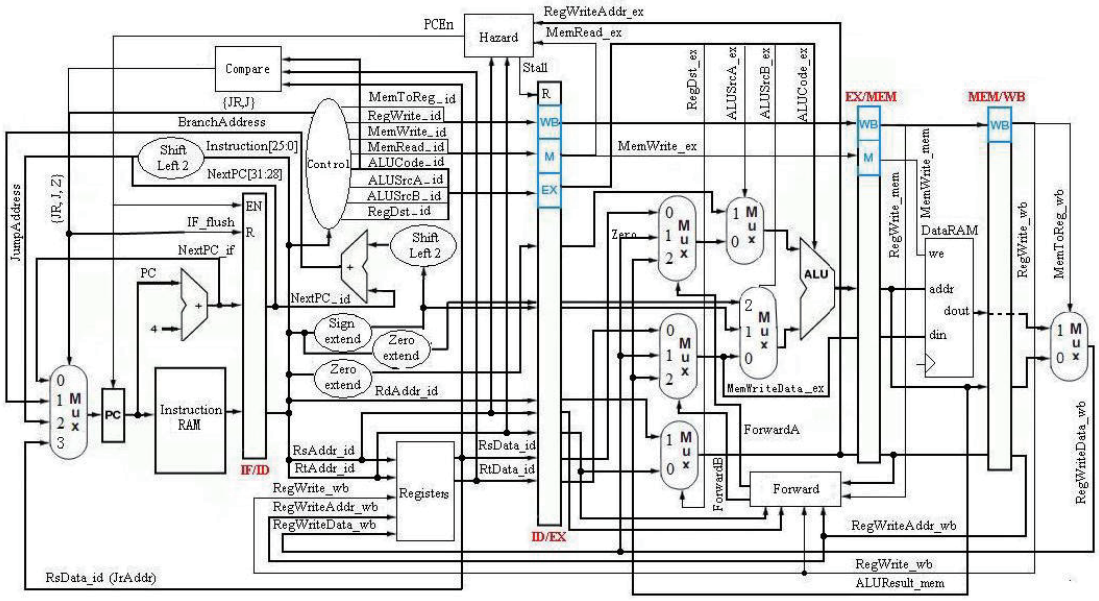
将 AES 加密汇编算法转化为机器码，写入 icache 和 dcache 中;

基于 modelsim 进行处理器的行为级仿真，得到正确的密文;

进行 DC 综合，给出时序报告，面积报告和功耗报告;

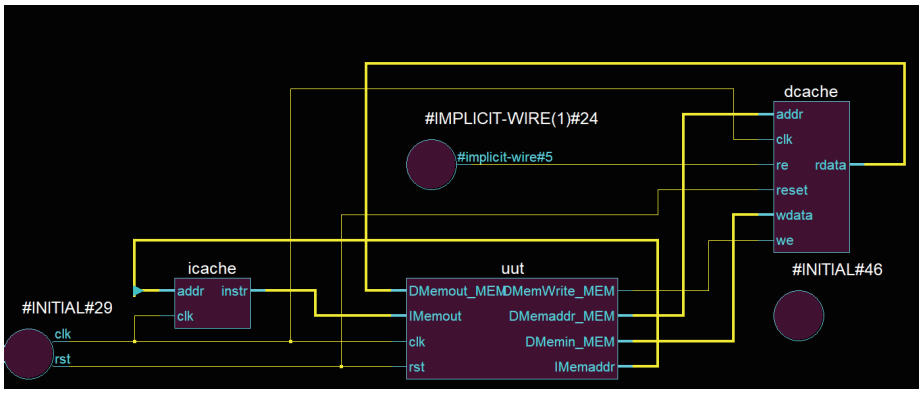
二.MIPS 处理器架构

4. 结构与数据通路



图中关于 branch 的 ID 段 forward 和 hazard 没有画出，branch 指令在 ID 段访问寄存器得到数值，同时与 MEM 和 WB 段要写回寄存器的地址进行对比，判断是否相关，这是由 forward 通路所决定的；根据相关结果来决定是否停滞（stall）流水线一拍；

5. 仿真 Schematic 图



该图显示了处理器功能模块 uut 和 icache 以及 dcache 模块的连接关系。仿真时，只需要将数据存放到 cache 中即可开始运行。Uut 内部信号 schematic 见附录文件 uut.pdf

三.设计细节描述

6. 流水线与冒险处理

- Forward

非分支指令 forward 在 EX stage 进行。判断逻辑如下：

```
RegRdout1Sel_Forward_EX[0] = RegWrite_WB && (RegWtaddr_WB != 0) && (RegWtaddr_MEM != Rs_EX) && (RegWtaddr_WB == Rs_EX);

RegRdout1Sel_Forward_EX[1] = RegWrite_MEM && (RegWtaddr_MEM != 0) && (RegWtaddr_MEM == Rs_EX);

RegRdout2Sel_Forward_EX[0] = RegWrite_WB && (RegWtaddr_WB != 0) && (RegWtaddr_MEM != Rt_EX) && (RegWtaddr_WB == Rt_EX);

RegRdout2Sel_Forward_EX[1] = RegWrite_MEM && (RegWtaddr_MEM != 0) && (RegWtaddr_MEM == Rt_EX);
```

RegRdout1Sel_Forward_EX，RegRdout2Sel_Forward_EX 分别是判断 Rs，Rt 是否与 MEM 和 WB 段要写入的地址是否相关。

由于分支指令的 forward 提前到 ID 段进行（为了减少 FLUSH 的时间代价），因此要重新编写 forward。判断逻辑如下：

```
RegRdout1Sel_Forward_ID[0] =isBranch? RegWrite_WB && (RegWtaddr_WB != 0) && (RegWtaddr_MEM != RegRdaddr1_ID) && (RegWtaddr_WB == RegRdaddr1_ID):0;

RegRdout1Sel_Forward_ID[1] =isBranch? RegWrite_MEM && (RegWtaddr_MEM != 0) && (RegWtaddr_MEM == RegRdaddr1_ID):0;

RegRdout2Sel_Forward_ID[0] =isBranch? RegWrite_WB && (RegWtaddr_WB != 0) && (RegWtaddr_MEM != RegRdaddr2_ID) && (RegWtaddr_WB == RegRdaddr2_ID):0;

RegRdout2Sel_Forward_ID[1] = isBranch? RegWrite_MEM && (RegWtaddr_MEM != 0) && (RegWtaddr_MEM == RegRdaddr2_ID):0;
```

- Hazard

Hazard 承担起将 IF 和 ID 段的流水线停滞的功能。与之相关的指令包括分支指令和访存指令。

```
ID_EX_Flush_isBranch = ((RegWtaddr_EX == Rs_ID && Rs_ID!=0) || (RegWtaddr_EX == Rt_ID && Rt_ID!=0 ))
||(DMemRead_MEM &&((RegWtaddr_MEM == Rs_ID && Rs_ID!=0) || (RegWtaddr_MEM == Rt_ID && Rt_ID!=0)))
ID_EX_Flush_LWSW=((RegWtaddr_EX == Rs_ID) || (RegWtaddr_EX == Rt_ID)) && DMemRead_EX;
IF_ID_En = ~ID_EX_Flush;
PCEn = ~ID_EX_Flush;
```

- FLUSH

FLUSH 需要将从 IF stage 进入 ID stage 的寄存器全置零，使得该条指令不执行。主要用于无条件跳转指令和分支指令。

```
IF_ID_Flush = (PCSrc_ID != 2'b00 &&!Stall);
```

- 分支策略

如前所述，总是假设分支不成功，即分支后的指令照常进入流水线。当分支指令进入 ID stage 时，先检测是否数据相关，若相关，则 stall 分支指令。否则，根据分支指令的判断结果给出控制信号，使得 PCSrc_ID 来源为 branch，将分支指令之后的那条指令冲刷掉，并载入跳转到的 PC。

7. AES 转化为机器码

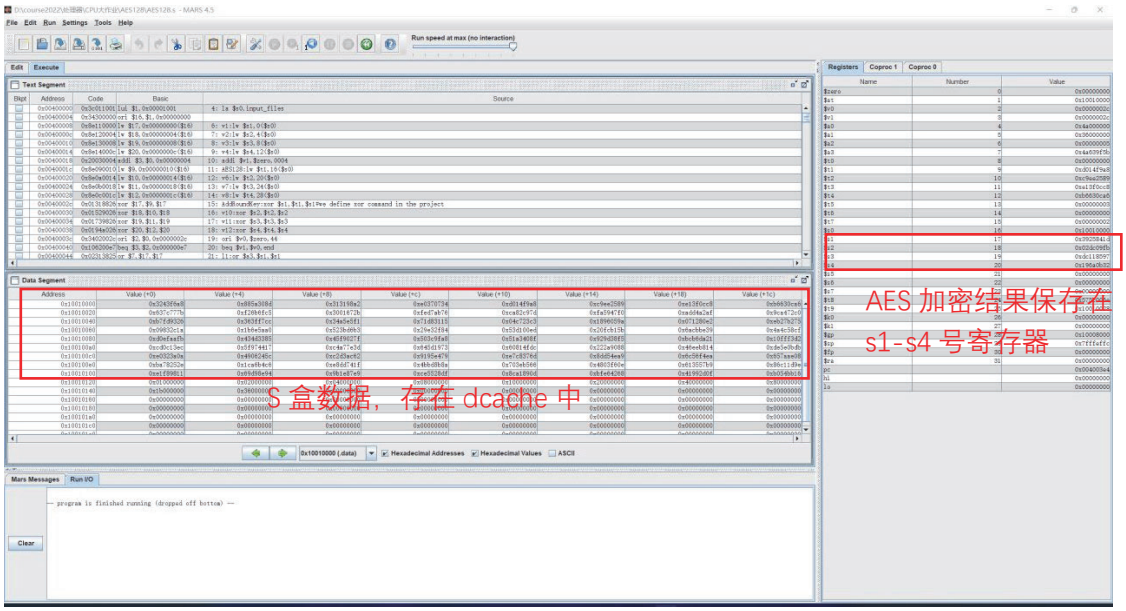
- 代码的格式

AES 算法实现见附录 AES128.s 文件。下面是节选代码。首先需要将 s 盒中的数据载入，利用.data 和 .word 实现。.text 后面代码即 AES 算法的加密部分。寄存器的命名需要按照 MIPS 的规则进行。

```
.data
input_files: .word 0x3243f6a8,0x885a308d,0x313198a2,0xe0370734,0x2b7e1516,0x28aed2a6,0xabf71588,0x09cf4f3c,0x637c777b,0xf26b6fc5,
.text
la $s0,input_files
Main:
v1:lw $s1,0($s0)
v2:lw $s2,4($s0)
v3:lw $s3,8($s0)
v4:lw $s4,12($s0)
addi $v1,$zero,0004
AES128:lw $t1,16($s0)
v6:lw $t2,20($s0)
v7:lw $t3,24($s0)
v8:lw $t4,28($s0)
AddRoundKey:xor $s1,$t1,$s1#we define xor command in the project
v10:xor $s2,$t2,$s2
v11:xor $s3,$t3,$s3
v12:xor $s4,$t4,$s4
ori $v0,$zero,44
```

- 代码的转换

利用 Mars 软件，将汇编代码转化为二进制代码。软件界面如图所示：



$s1	17	0x3925841d
$s2	18	0x02dc09fb
$s3	19	0xdc118597
$s4	20	0x196a0b32

将转换后的二进制代码分别写入 icache 和 dcache 中。

8. 其他子模块功能描述

alu.v			
I/O	Width	Name	
input	[31:0]	alu_a	操作数 a，如果有负数，是以补码存储
input	[31:0]	alu_b	操作数 b，如果有负数，是以补码存储
input	[4:0]	alu_op	运算类型
output	[31:0]	alu_out	运算结果，如果有负数，是以补码存储

registers.v			
I/O	Width	Name	说明
input	[0:0]	clk	时钟沿
input	[0:0]	rst_n	复位信号，低电平有效
input	[4:0]	rAddr1	读地址 1
output	[31:0]	rDout1	读数据 1
input	[4:0]	rAddr2	读地址 2
output	[31:0]	rDout2	读数据 2
input	[4:0]	wAddr	写地址
input	[31:0]	wDin	写数据
input	[0:0]	wEna	写使能，高电平有效

mux.v			
I/O	Width	Name	说明
input	[0:0]	sel	选择信号
input	[WIDTH-1:0]	d0	选择数据1
input	[WIDTH-1:0]	d1	选择数据2
output	[WIDTH-1:0]	out	输出

dff.v寄存器组 用于流水线			
I/O	Width	Name	说明
input	[0:0]	clk	时钟沿
input	[0:0]	en	使能信号高电平有效
input	[0:0]	rst	复位信号高电平有效
input	[WIDTH-1:0]	datain	输入数据
output	[WIDTH-1:0]	dataout	输出数据

compare.v 用于branch比较			
输入/输出	宽度	信号名	说明
input	[31:0]	a	有符号数 a
input	[31:0]	b	有符号数b
output	[1:0]	res	比较结果

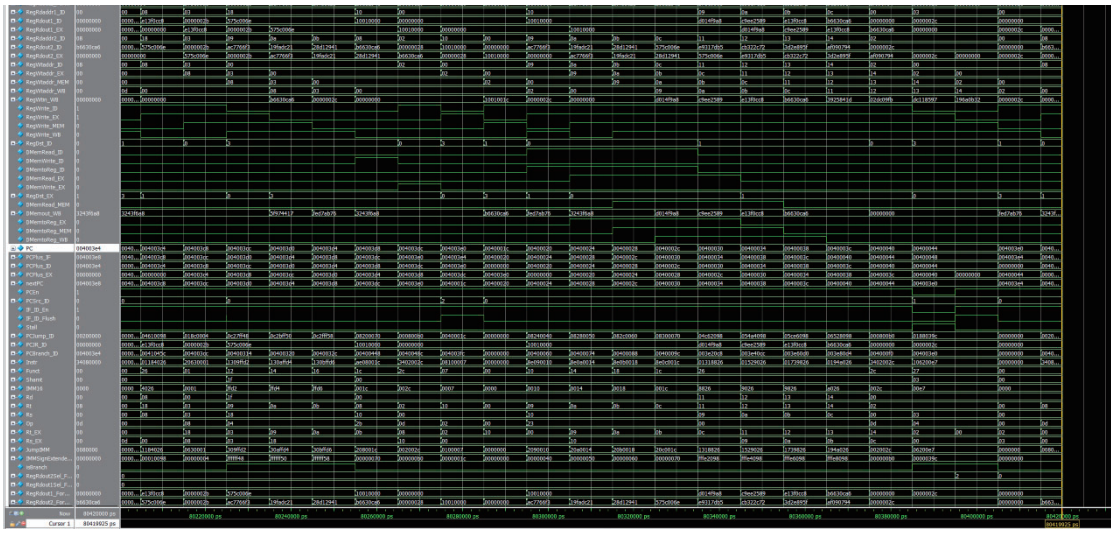
四.功能与性能

9. 加密执行结果

- 利用 modelsim 的 Memorylist 功能可以直接查看寄存器组 Registers 的值。S1-S4 号寄存器分别对应的十进制编号为 17-20。读下面图中对应位置的寄存器值即可知道数据运算正确。

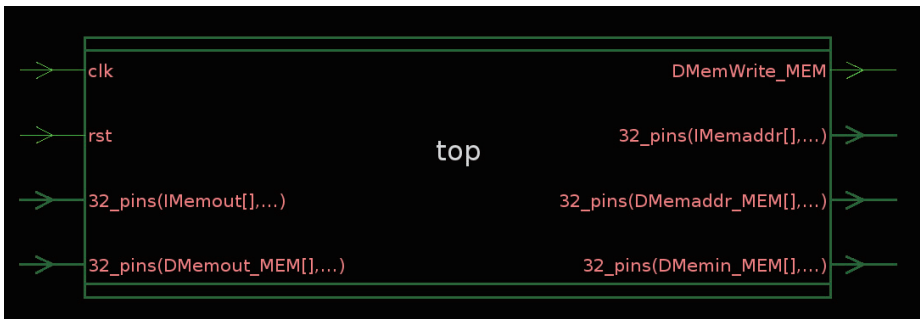
```
 0 00000000 10010000 0000002c 0000002c 4a000000 36000000 00000005 4a639f5b b6630ca6 d014f9a8 c9ee2589
11 e13f0cc8 b6630ca6 00000003 00000000 00000002 10010000 3925841d 02dc09fb dc118597 196a0b32 00000000
22 00000000 00000000 575c006e 10010028 00000000 00000000 00000000 00000000 00000000 00000000
```

- 运算的周期数=80420000ps/ 10000ps =8042 个周期，仿真时每个周期为 10ns，如下图所示。这里本人用$stop 语句停止程序仿真/AES 加密结束，其标志是读取第一条未定义的指令地址 249，本人最后一条指令地址为 248。



10. DC 综合结果

只对处理器的功能模块进行了综合，icache 和 dcache 没有综合。主要原因是 DC 综合默认 cache 里面的赋值为常量，会被不规则优化，导致面积不够和功耗不准确。下图是 DC 综合的 top 模块。



性能参数如下表所示：

项目	指标
时钟周期/ns	4
面积/um^2	199593.981445
功耗 mW	42.2706

时序信息节选：

```
clock clk (rise edge)                                  0.00       0.00
clock network delay (ideal)                            0.00       0.00
registers_uut/data_reg[29][20]/CK (DFFRHQX1)           0.00 #     0.00 r
registers_uut/data_reg[29][20]/Q (DFFRHQX1)            0.18       0.18 f
registers_uut/U2943/Y (AOI22XL)                        0.13       0.32 r
registers_uut/U2944/Y (NAND2XL)                        0.05       0.36 f
registers_uut/U2948/Y (NAND4BBX2)                      0.12       0.48 f
registers_uut/U2949/Y (NOR2X2)                         0.08       0.56 r
registers_uut/U2956/Y (NAND2X2)                        0.04       0.60 f
registers_uut/rDout1[20] (Registers)                   0.00       0.60 f
U3833/Y (AOI21X2)                                      0.13       0.73 r
U2031/Y (AOI22X2)                                      0.09       0.82 f
U2030/Y (NAND4X2)                                      0.14       0.97 r
U2029/Y (NOR3X2)                                       0.06       1.03 f
U2028/Y (AOI2BB2X4)                                    0.13       1.16 f
U2055/Y (NAND4X4)                                      0.12       1.28 r
U4128/Y (MXI2X4)                                       0.11       1.39 f
U4203/Y (NOR2X4)                                       0.09       1.48 r
U4209/Y (NOR2X4)                                       0.05       1.52 f
U4210/Y (BUFX20)                                       0.11       1.63 f
U4371/Y (AOI22X1)                                      0.11       1.74 r
U1756/Y (NAND4XL)                                      0.06       1.80 f
DFFPC/dataout_reg[6]/D (DFFHQX4)                       0.00       1.80 f
data arrival time                                                 1.80

clock clk' (rise edge)                                 2.00       2.00
clock network delay (ideal)                            0.00       2.00
clock uncertainty                                      0.00       2.00
DFFPC/dataout_reg[6]/CK (DFFHQX4)                      0.00       2.00 r
library setup time                                    -0.20       1.80
data required time                                                1.80
-----------------------------------------------------------------------------
data required time                                                1.80
data arrival time                                                -1.80
-----------------------------------------------------------------------------
slack (MET)                                                       0.00
```

五.附录

./src_modelsim 文件夹包含仿真的代码，有不可综合语句
./dc/src 内为综合代码
./dc/run.tcl 综合脚本
./dc/reports/ 内为综合报告信息
./dc/output/ 内为进行 RTL 仿真的 map 以及 sdf 文件
./uut.pdf 为 MIPS 的 schematic 图

./AES128/ 内有 aes 算法的汇报文件和机器码