

# 1、Data Path 和 Control Path 结构图

MIPS 指令集中每条指令都可以在五个时钟周期内实现。

## IF: Instruction fetch cycle

将程序计数器(PC)发送到存储器,从存储器提取当前指令。向程序计数器加4(因为每条指令的长度为4个字节),将程序计数器更新到下一个连续程序计数器。

## ID: Instruction decode/register fetch cycle

对指令进行译码,并从寄存器堆中读取与寄存器源说明符相对应的寄存器。在读取寄存器时对其进行相等测试,以确定是否为分支。必要时,对指令的偏移量字段进行符号扩展。符号扩展后的偏移量添加到所实现的程序计数器上,计算出可能的分支目标地址。由于一个指令的立即数部分也位于同一位置,所以在需要符号扩展立即数时,也是在这一周期计算的。

## EX: Execution/effective address cycle

ALU 对上一周期准备的操作数进行操作,根据指令类型执行三条指令之一。存储器引用—ALU 将基址寄存器和偏移量加到一起,形成有效地址。寄存器寄存器—ALU 指令—ALU 对读自寄存器堆的值执行由 ALU 操作码指定的操作。

寄存器—立即数 ALU 指令—ALU 对读自寄存器堆的第一个值和符号扩展立即数执行由 ALU 操作码指定的操作。

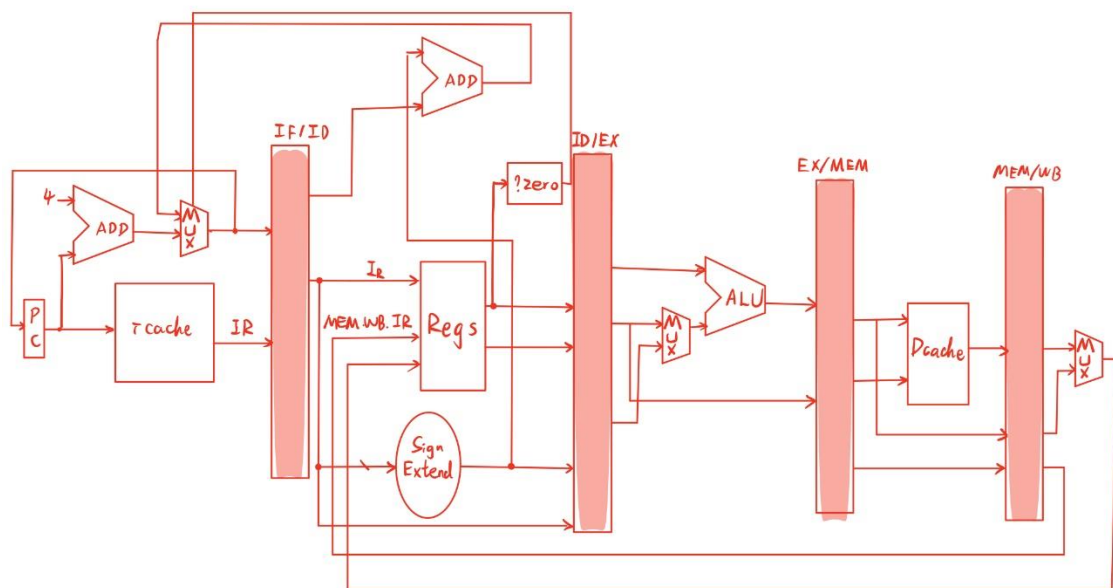
## MEM: Memory access

如果该指令是一条载入指令,则使用上一周期计算的有效地址从存储器中读取数据。如果是一条存储指令,则使用有效地址将从寄存器堆的第二个寄存器读取的数据写入存储器。

## WB: Write-back cycle

将结果写入寄存器堆,无论是来自寄存器系统(对于载入指令),还是来自 ALU(对于 ALU 指令)。

分支冒险在 ID 阶段进行判断,因此可能会造成数据冒险。



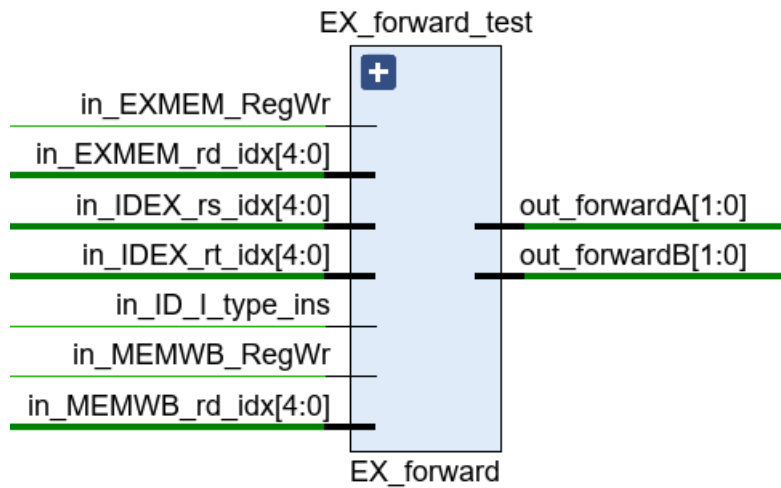
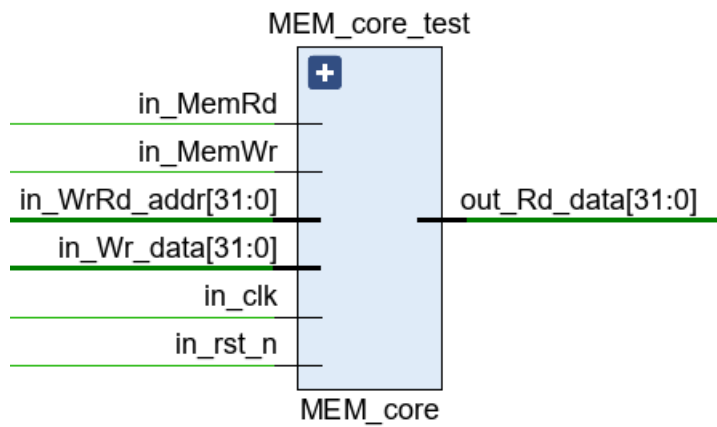
## 各阶段数据通路

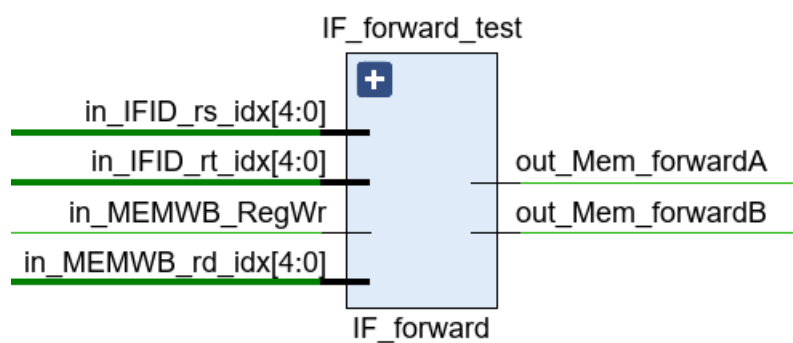
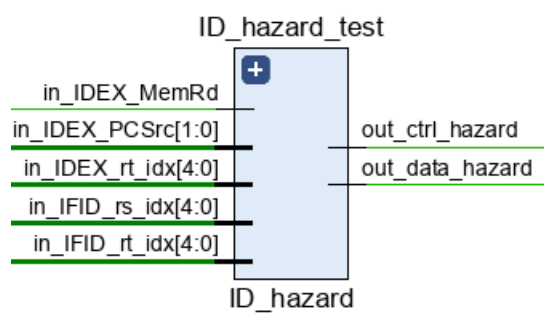
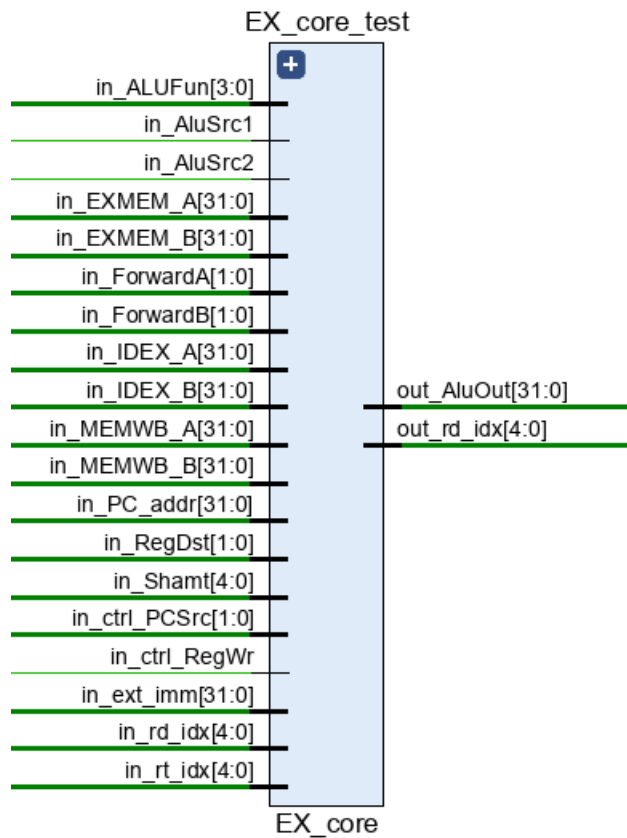
stage	Instruction		
IF	$IF/IF.IR \leftarrow Mem[PC];$ $IF/ID.NPC.PC \leftarrow (if ((EX/MEM.opcode == branch) \& EX/MEM.cond)\{EX/MEM.ALUOutput\} else \{PC+4\} );$		
ID	$ID/EX.A \leftarrow Regs[IF/ID.IR[rs]]; 10/EX.B \leftarrow Regs[IF/ID.IR[rt]];$ $ID/EX.NPC \leftarrow IF/ID.NPC; ID/EX.IR \leftarrow 1F/ID.IR;$ $1D/EX.Imm \leftarrow sign-extend(IF/ID.IR[immediate field]);$		
	<b>ALU instruction</b>	<b>Load or store instruction</b>	<b>Branch instruction</b>
EX	$EX/MEM.IR \leftarrow ID/EX.IR;$ $Ex/MEM.ALUOutput \leftarrow 1D/EX.A func 10/EX.B;$ or $Ex/MEM.ALUOutput \leftarrow ID/EX.A op 1D/EX.Im;$	$EX/MEM.IR \leftarrow ID/EX.IR;$ $EX/MEM.ALUOutput \leftarrow ID/EX.A + 1D/EX.Im;$	$EX/MEM.ALUOutput \leftarrow 1D/EX.NPC + (ID/EX.Imm < 2);$ $EX/MEM.cond \leftarrow (ID/EX.A == 0);$
MEM	$MEM/NB.IR \leftarrow EX/MEM.IR;$ $MEM/WB.ALUOutput \leftarrow EX/MEM.ALUOutput;$	$MEM/NB.IR \leftarrow EX/MEM.IR;$ $MEM/HB.LMD t - Mem[EX/MEM.ALUOutput];$ or $Mem[EX/MEM.ALUOutput] \leftarrow Ex/MEM.B;$	
WB	$Regs[MEM/WB.ALUOutput] \leftarrow Ex/MEM.ALUOutput;$	For load only:	

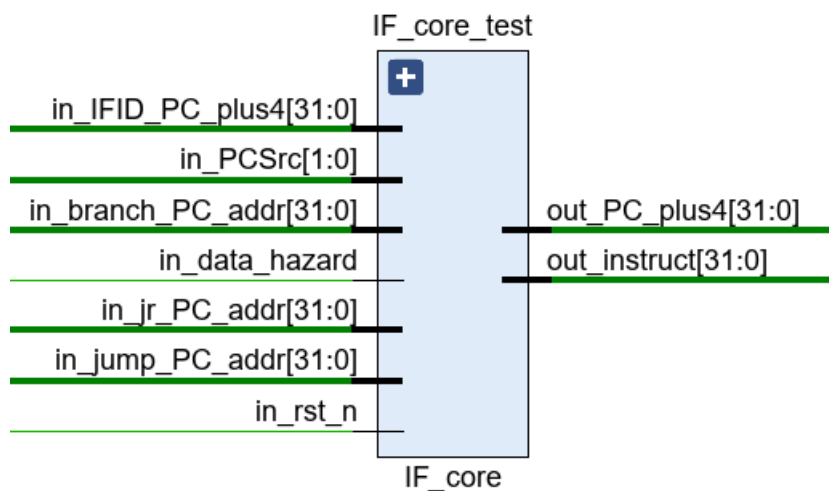
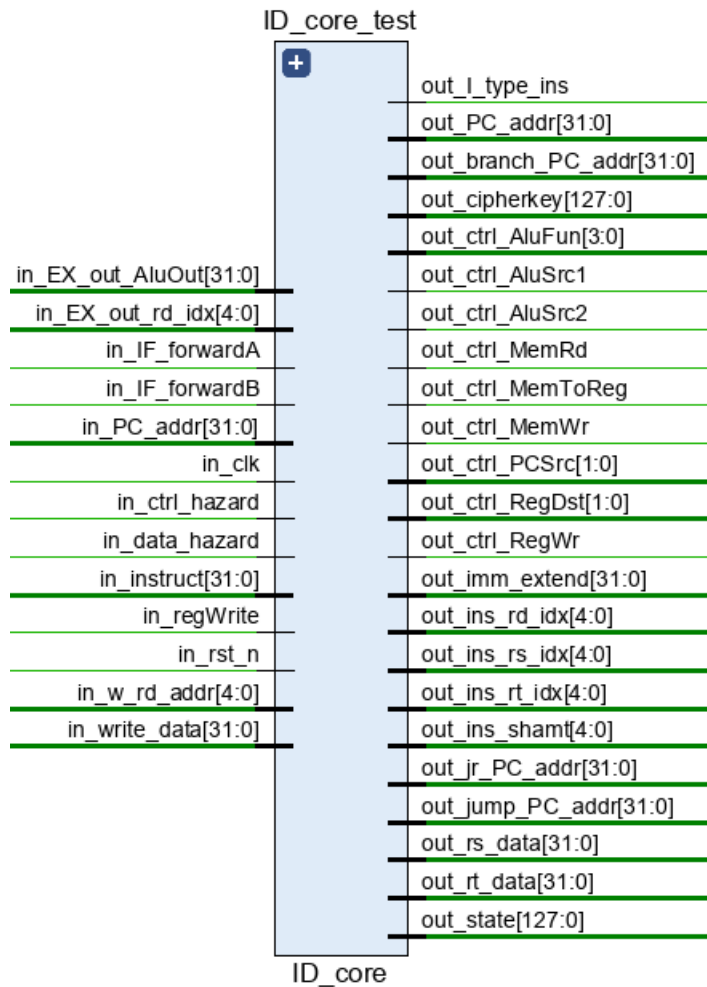
	IR[rd]← MEM/WB. ALUOutput; or Regs [MEM/WB.IR[rt]]← MEM/WB. ALUOutput;	Regs [MEM/WB. IR[rt]]← MEM/NB. LMD;	
--	---	--	--

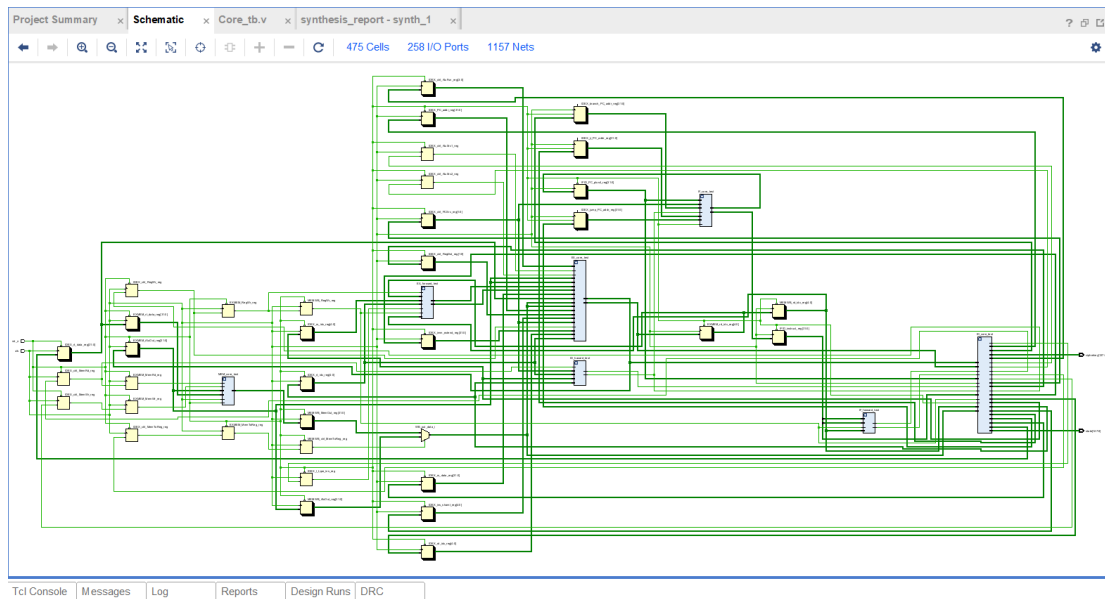
## 2、verilog 源码

见附件









### 3、AES 汇编程序仿真结果

System CLK Period = 10ns , Freq = 100MHz

由产生最终密文 39 25 84 1d 02 dc 09 fb dc 11 85 97 19 6a 0b 32 的用时推知，AES 加密运算时钟周期数 = 4851



图 modelsim

# 4、时延、资源情况

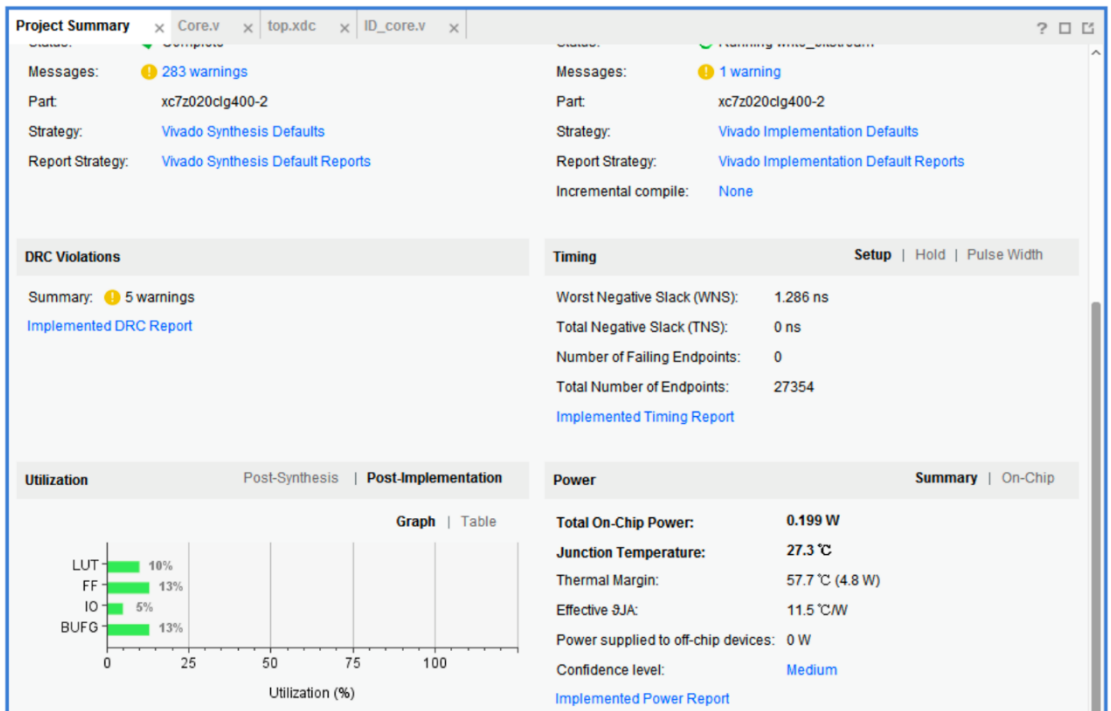


图 project summary

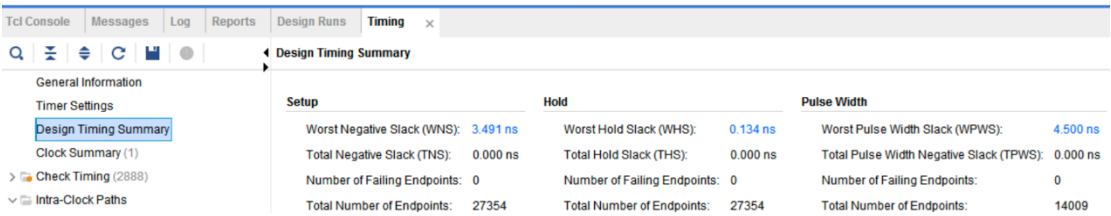


图 timing

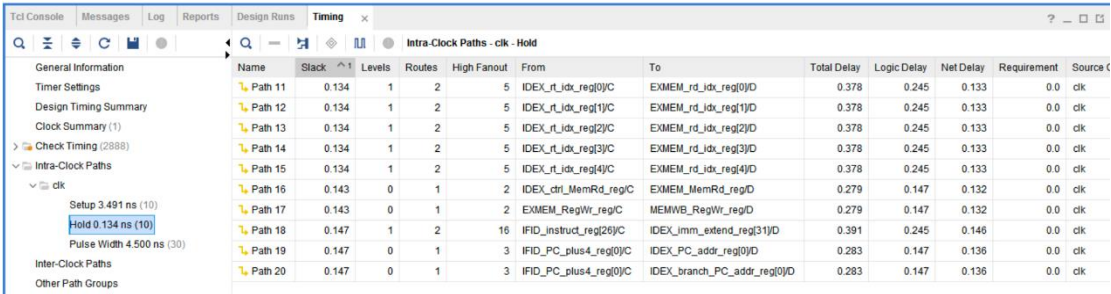


图 hold

Tcl Console	Messages	Log	Reports	Design Runs	Timing												
Intra-Clock Paths - clk - Setup																	
General Information					Name	Slack	Levels	Routes	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement	Source	
Timer Settings					Path 1	3.491	12	13	32	IDEX_ctrl_RegDst_reg[1]C	IDEX_ctrl_PCsrc_reg[0]D	6.372	2.371	4.001	10.0	clk	
Design Timing Summary					Path 2	4.807	6	7	114	IFID_instruct_reg[25]C	IFID_instruct_reg[16]D	4.828	1.523	3.305	10.0	clk	
Clock Summary (1)					Path 3	4.807	6	7	114	IFID_instruct_reg[25]C	IFID_instruct_reg[16]repID	4.828	1.523	3.305	10.0	clk	
>	Check Timing (2888)					Path 4	4.807	6	7	114	IFID_instruct_reg[25]C	IFID_instruct_reg[16]rep_0D	4.828	1.523	3.305	10.0	clk
>	Intra-Clock Paths					Path 5	4.807	6	7	114	IFID_instruct_reg[25]C	IFID_instruct_reg[21]D	4.828	1.523	3.305	10.0	clk
>	clk					Path 6	4.807	6	7	114	IFID_instruct_reg[25]C	IFID_instruct_reg[21]repID	4.828	1.523	3.305	10.0	clk
	Setup 3.491 ns (10)					Path 7	4.807	6	7	114	IFID_instruct_reg[25]C	IFID_instruct_reg[21]rep_0D	4.828	1.523	3.305	10.0	clk
	Hold 0.134 ns (10)					Path 8	4.807	6	7	114	IFID_instruct_reg[25]C	IFID_instruct_reg[22]D	4.828	1.523	3.305	10.0	clk
	Pulse Width 4.500 ns (30)					Path 9	4.807	6	7	114	IFID_instruct_reg[25]C	IFID_instruct_reg[22]repID	4.828	1.523	3.305	10.0	clk
	Inter-Clock Paths					Path 10	4.807	6	7	114	IFID_instruct_reg[25]C	IFID_instruct_reg[22]rep_0D	4.828	1.523	3.305	10.0	clk

图 setup

Tcl Console

Messages

Log

Reports

Design Runs

Timing

×

🔍

⏮

⏪

🔄

💾

●

⏪

⏮

⏪

Clock Summary

General Information

Timer Settings

Design Timing Summary

Clock Summary (1)

> Check Timing (2888)

Name	Waveform	Period (ns)	Frequency (MHz)
clk	{0.000 5.000}	10.000	100.000

图 CLK