



浙江大学计算机学院

2023年9月

Outline



- Experiment Purpose
- Experiment Task
- Basic Principle
- Operating Procedures
- Checkpoints



Experiment Purpose



- Understand the principle of CPU exception & interrupt and its processing procedure.
- Master the design methods of pipelined CPU supporting exception & interrupt.
- master methods of program verification of Pipelined CPU supporting exception & interrupt.



Experiment Task



- Design of Pipelined CPU supporting exception & interrupt.
 - Design datapath
 - Design Co-processor & Controller

Verify the Pipelined CPU with program and observe the execution of program





Level	Encoding	Name	Abbreviation
0	00	User/Application	U
1	01	Supervisor	S
2	10	Reserved	
3	11	Machine	M

Number of levels	Supported Modes	Intended Usage
1	M	Simple embedded systems
2	M, U	Secure embedded systems
3	M, S, U	Systems running Unix-like operating systems







31	25 24	20 19	15 14	12 11	7.6)_
	csr	rs1	003	1 rd	1110011	I csrrw
	csr	rs1	010	0 rd	1110011	I csrrs
	csr	rs1	011	1 rd	1110011	I csrrc
	csr	zimm	10	1 rd	1110011	I csrrwi
	CST	zimm	110	0 rd	1110011	I cssrrsi
	csr	zimm	111	1 rd	1110011	I csrrci

INSTR. rd, csr, rs1/zimm

x[rd]=CSR[csr]

寄存器 x[rs1]/zimm的值直接/Set(或操作)/Clear(与操作)后,写入CSR[csr]寄存器





Number	Privilege	Name	Description							
		Mach	ine Information Registers							
0xF11	MRO	mvendorid	Vendor ID.							
0xF12	MRO	marchid	Architecture ID.							
0xF13	MRO	mimpid	Implementation ID.							
0xF14	MRO	mhartid	Hardware thread ID.							
	Machine Trap Setup									
0x300										
0x301	MRW	misa	ISA and extensions							
0x302	MRW	medeleg	Machine exception delegation register.							
0x303	MRW	mideleg	Machine interrupt delegation register.							
0x304	MRW	mie	Machine interrupt-enable register.							
0x305	MRW	mtvec	Machine trap-handler base address.							
0x306	MRW	mcounteren	Machine counter enable.							
		M	achine Trap Handling							
0x340	MRW	mscratch	Scratch register for machine trap handlers.							
0x341	MRW	mepc	Machine exception program counter.							
0x342	MRW	mcause	Machine trap cause.							
0x343	MRW	mtval	Machine bad address or instruction.							
0x344	MRW	mip	Machine interrupt pending.							





Environment Call and Breakpoint

31	20 19	15 14 12	11	7 6	0
funct12	rs1	funct3	rd	opcode	
12	5	3	5	7	
ECALL	0	PRIV	0	SYSTEM	
EBREAK	0	PRIV	0	SYSTEM	

Trap-Return Instructions

31	20 19	15 14 12	11	7 6	
funct12	rsl	1 funct3	rd	opcode	
12	5	3	5	7	5
MRET/SRET/URI	ET 0	PRIV	0	SYSTEM	





在 M 模式运行期间可能发生的同步例外有五种:

- 访问错误异常 当物理内存的地址不支持访问类型时发生 (例如尝试写入 ROM)。
- 断点异常 在执行 ebreak 指令,或者地址或数据与调试触发器匹配时发生。
- 环境调用异常 在执行 ecall 指令时发生。
- 非法指令异常在译码阶段发现无效操作码时发生。
- 非对齐地址异常 在有效地址不能被访问大小整除时发生。

Machine-mode status register (mstatus) for RV32.

	31 30						23	22	21	20	19	18	17	
	SD		V	VPRI				TSR	TW	TVM	MXR	SUM	MPR	V
	1			8				1	1	1	1	1	1	
	16 15	14 13	12 11	10 9	8	7	6	3	5	4	3	2	1	0
	XS[1:0]	FS[1:0]	MPP[1:0]	WPRI	SPP	MPIE	WF	PRI	SPIE	UPIE	MIE	WPRI	SIE	UIE
31	2	2	2	2	1	1	1		1	1	1	1	1	1

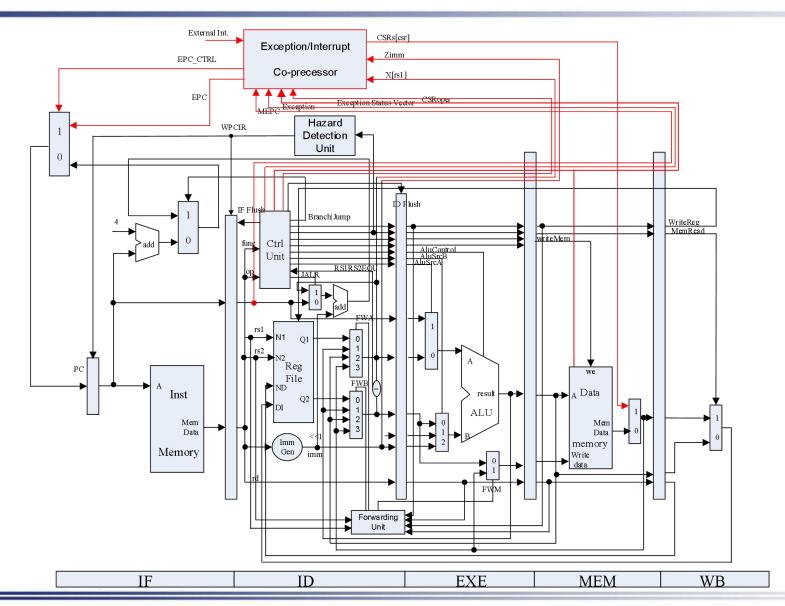


发生异常/中断时,硬件自动经历如下的状态转换:

- 异常指令的PC被保存在mepc中,PC被设置为mtvec。mepc指向导致异常的指令;对于中断,它指向中断处理后应该恢复执行的位置。
- 根据异常来源设置mcause,并将mtval设置为出错的地址或者其它适用于特定异常的信息字。
- 把控制状态寄存器mstatus中的MIE位置零以禁用中断,并把先前的MIE值保留到MPIE中。
- 发生异常之前的权限模式保留在mstatus的MPP域中,再把权限模式 更改为M。









Instr. Mem.(1)

NO.	Instruction	Addr.	Label	ASM	Comment
0	0000013	0	start:	addi x0, x0, 0	
1	00402103	4		lw x2, 4(x0)	
2	00802203	8		lw x4, 8(x0)	
3	00c02283	С		lw x5, 12(x0)	
4	01002303	10		lw x6, 16(x0)	
5	01402383	14		lw x7, 20(x0)	
6	306850f3	18		csrrwi x1, 0x306, 16	
7	306020f3	1C		csrr x1, 0x306	
8	306310f3	20		csrrw x1, 0x306, x6	
9	306020f3	24		csrr x1, 0x306	
10	0000013	28		addi x0, x0, 0	
11	07800093	2C		addi x1, x0, 120	
12	30509073	30		csrw 0x305, x1	
13	0000013	34		addi x0, x0, 0	
14	00000073	38		ecall	





Instr. Mem.(2)

NO.	Instruction	Addr.	Label	ASM	Comment
15	0000013	3C		addi x0, x0, 0	
16	00000012	40		addi x0, x0, 0	# change to illegal
17	00000013	44		addi x0, x0, 0	
18	07f02083	48		lw x1, 127(x0)	
19	08002083	4C		lw x1, 128(x0)	# I access fault
20	00000013	50		addi x0, x0, 0	
21	08102023	54		sw x1, 128(x0)	# s access fault
22	00000013	58		addi x0, x0, 0	
23	00000013	5C		addi x0, x0, 0	
24	00000013	60		addi x0, x0, 0	
25	00000013	64		addi x0, x0, 0	
26	00000013	68		addi x0, x0, 0	
27	00000013	6C		addi x0, x0, 0	
28	00000013	70		addi x0, x0, 0	
 29	00000067	74		jr x0	



Instr. Mem.(3)

NO	Э.	Instruction	Addr.	Label	ASM	Comment
3	0	34102cf3	78	trap:	csrr x25, 0x341	# mepc
3	1	34202df3	7C		csrr x27, 0x342	# mcause
3	2	30002e73	80		csrr x28, 0x300	# mstatus
3	3	30402ef3	84		csrr x29, 0x304	# mie
3	4	34402f73	88		csrr x30, 0x344	# mip
3	5	004c8113	8C		addi x2, x25, 4	
3	6	34111073	90		csrw 0x341, x2	
3	7	30200073	94		mret	# 30200073 mret
3	8	00000013	98		addi x0, x0, 0	
3	9	00000013	9C		addi x0, x0, 0	
4	0	00000013	Α0		addi x0, x0, 0	
4	1	00000013	A4		addi x0, x0, 0	
4						



Data Mem.

NO.	Data	Addr.	Comment	NO.	Instruction	Addr.	Comment
0	000080BF	0		16	00000000	40	
1	8000000	4		17	00000000	44	
2	0000010	8		18	00000000	48	
3	0000014	С		19	00000000	4C	
4	FFFF0000	10		20	A3000000	50	
5	0FFF0000	14		21	27000000	54	
6	FF000F0F	18		22	79000000	58	
7	F0F0F0F0	1C		23	15100000	5C	
8	0000000	20		24	00000000	60	
9	0000000	24		25	00000000	64	
10	0000000	28		26	00000000	68	
11	0000000	2C		27	00000000	6C	
12	0000000	30		28	00000000	70	
13	0000000	34		29	00000000	74	
14	0000000	38		30	00000000	78	
2 , 15	0000000	3C		31	00000000	7C	

Test Bench

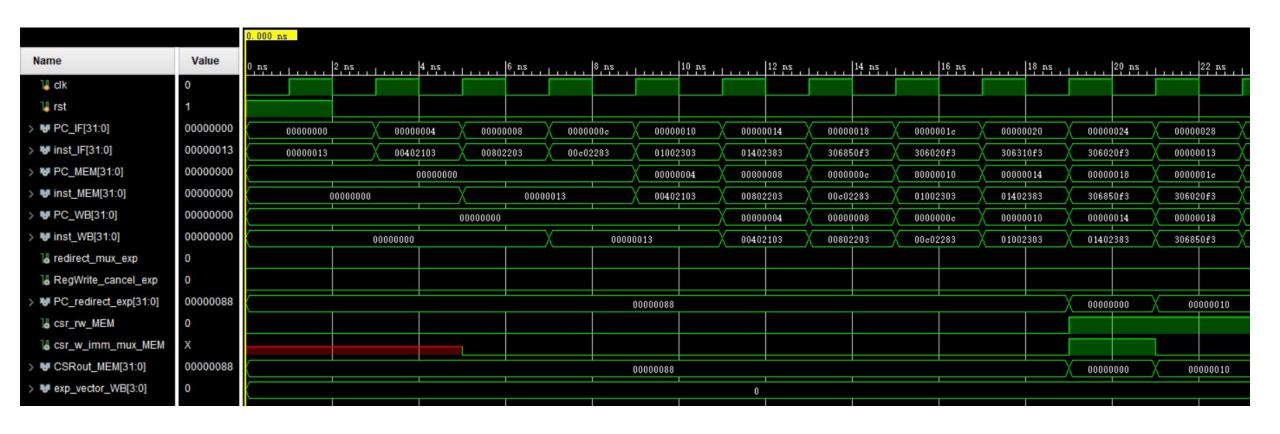


```
RV32core core(
    .debug_en(1'b0),
    .debug_step(1'b0),
    .debug_addr(7'b0),
    .debug_data(),
    .clk(clk),
    .rst(rst),
    .interrupter(1'b0)
 initial begin
   clk = 0;
    rst = 1;
   #2 rst = 0;
 end
 always #1 clk = ^{\sim}clk;
```



Simulation (1)







Simulation (2)



Name	Value	20 ns	22 ns	24 ns	26 ns	28 ns	30 ns	32 ns	34 ns	36 ns	38 ns	40 ns	42 ns
¹. dk	0												
¼ rst	1												
₩ PC_IF[31:0]	00000000	00000024	00000028	0000002c	00000030	00000034	00000038	0000003c	00000040	00000044	00000048	0000004c	00000078
₩ inst_IF[31:0]	00000013	306020f3	00000013	07800093	30509073	00000013	00000073	00000013	00000012	00000013	07f02083	08002083	34102cf3
₩ PC_MEM[31:0]	00000000	00000018	0000001c	00000020	00000024	00000028	00000020	00000030	00000034	00000038	00000030	00000040	00000044
Inst_MEM[31:0]	00000000	306850f3	306020f3	306310f3	306020f3	00000013	07800093	30509073	00000013	00000073	00000013	000	00000
₩ PC_WB[31:0]	00000000	00000014	00000018	00000010	00000020	00000024	00000028	0000002c	00000030	00000034	00000038	00000030	00000040
₩ inst_WB[31:0]	00000000	01402383	306850f3	306020f3	306310f3	306020f3	00000013	07800093	30509073	00000013	00000073	000	100000
laredirect_mux_exp	0												
RegWrite_cancel_exp	0												
PC_redirect_exp[31:0]	8800000	00000000	00000	0010		ffff0000		00000000	X		00000078		
csr_rw_MEM	0												
la csr_w_imm_mux_MEM	X												
♥ CSRout_MEM[31:0]	8800000	00000000	00000	0010		ffff0000		00000000	Χ		00000078		
■ exp_vector_WB[3:0]	0					0					X 4	χ	0



Simulation (3)



Name	Value	40 ns 42 ns 44 ns 46 ns	48 ns	50 ns	52 ns .	54 ns	56 ns	58 ns	60 ns	62 ns
¼ clk	0									
¼ rst	1									
> W PC_IF[31:0]	00000000	0000004c	00000084	00000088	0000008c	00000090	00000094	00000098	0000009c	00000000
> W inst_IF[31:0]	00000013	08002083 34102cf3 34202df3 30002e73	30402ef3	34402f73	004c8113	34111073	30200073	X	00000013	
> ₩ PC_MEM[31:0]	00000000	00000040 00000044	00000078	00000070	00000080	00000084	00000088	00000080	00000090	00000094
> W inst_MEM[31:0]	00000000	0000000	34102cf3	34202df3	30002e73	30402ef3	34402f73	004c8113	34111073	30200073
> W PC_WB[31:0]	00000000	0000003c X 00000040 X 00000044		00000078	0000007c	00000080	00000084	00000088	0000008c	00000000
> W inst_WB[31:0]	00000000	0000000		34102cf3	34202df3	30002e73	30402ef3	34402f73	004c8113	34111073
redirect_mux_exp	0			100						
RegWrite_cancel_exp	0									
> W PC_redirect_exp[31:0]	00000088	00000078	00000038	(0000000Ъ	00000080	00000fff	0000	0000	00000038	00000036
Gr_rw_MEM	0									
dia csr_w_imm_mux_MEM	X									
> W CSRout_MEM[31:0]	88000000	00000078	00000038	(0000000Ъ	00000080	00000fff	0000	0000	00000038	00000036
> W exp_vector_WB[3:0]	0				0					



Simulation (4)



Name	Value	60 ns	62 ns	64 ns	66 ns	68 ns	70 ns	72 ns	74 ns	76 ns		80 ns	82 ns
¼ dk	0												
¼ rst	1												
> W PC_IF[31:0]	00000000	0000009c	0000000	0000003c	00000040	00000044	00000048	0000004c	00000050	00000054	00000078	0000007c	00000080
> W inst_IF[31:0]	00000013		00000013		00000012	00000013	07f02083	08002083	00000013	08102023	34102cf3	34202df3	30002e73
> ₩ PC_MEM[31:0]	00000000	00000090	00000094	00000098	0000	009c	00000036	00000040	00000044	00000048	X	0000004c	
> W inst_MEM[31:0]	00000000	34111073	30200073	Χ	00000000		00000013	00000012	00000013	X	0000	1000	
> W PC_WB[31:0]	00000000	0000008c	00000090	00000094	00000098	0000	1009c	0000003c	00000040	00000044	00000048	000	0004c
> W inst_WB[31:0]	00000000	004c8113	34111073	30200073	Χ	00000000		00000013	00000012	X	000	00000	
<pre>redirect_mux_exp</pre>	0									1			
RegWrite_cancel_exp	0								1				
> VP PC_redirect_exp[31:0]	8800000	00000038	Х			0000003c				X	0000	078	
csr_rw_MEM	0												
discrete control of the con	Х												
> W CSRout_MEM[31:0]	88000000	00000038	X			0000003c				X	0000	0078	
> W exp_vector_WB[3:0]	0				0				8	X		0	



Simulation (5)



Name	Value	80 ns 82	ns 84 ns		2n 88	90 ns .	92 ns	94 ns	96 ns .	98 ns .	100 ns	102 ns
¹ dk	0											
¼ rst	1											
> ₩ PC_IF[31:0]	00000000	0000007c 00000080	0 00000084	00000088	0000008c	00000090	00000094	8000000	0000009c	00000040	00000044	00000048
> W inst_IF[31:0]	00000013	34202df3 30002e73	3 30402ef3	34402f73	004c8113	34111073	30200073	X	0000	0013		07f02083
> ₩ PC_MEM[31:0]	00000000	0000004c	00000078	0000007c	0800000	00000084	00000088	00000080	00000090	00000094	00000098	00000090
> ₩ inst_MEM[31:0]	00000000	00000000	34102cf3	34202df3	30002e73	30402ef3	34402f73	004c8113	34111073	30200073	000	100000
▶ PC_WB[31:0]	00000000	00000	04c	00000078	X 0000007c	00000080	00000084	00000088	0000008c	00000090	00000094	00000098
> ₩ inst_WB[31:0]	00000000	00000	000	34102cf3	X 34202df3	30002e73	30402ef3	34402f73	004c8113	34111073	30200073	00000000
<pre>redirect_mux_exp</pre>	0											
RegWrite_cancel_exp	0											
▶ PC_redirect_exp[31:0]	00000088	00000078	00000040	00000002	0800000	00000fff	0000	0000	00000040	X	00000044	
d csr_rw_MEM	0											
d csr_w_imm_mux_MEM	X											
♥ CSRout_MEM[31:0]	88000000	00000078	00000040	00000002	00000080	00000fff	0000	10000	00000040	X	00000044	
w exp_vector_WB[3:0]	0						0					



Simulation (6)



	100 ns	102 ns	10	4 ns	106 ns	1	108 ns	110	ns	112 ns		14 ns	116 ns	118 n	s 120 ns	122 ns
	النفنا									ننننا						
000000	00000044	00000048	0000004	c 00	000050	0000	0054	00000058	00	00005c	000000	060	00000078	0000007c	00000080	00000084
000013	00000013	07f02083	0800208	3 00	000013	0810	2023		00	000013		X	34102cf3	34202df3	30002e73	30402ef3
000000	00000098	00000	009c	X 00	000044	0000	0048	0000004c	00	000050	000000	054		00000058		00000078
000000		00000000		00	000013	07f0	2083	08002083	00	000013	X .		00000	0000		34102cf3
000000	00000094	00000098	X	0000009c		0000	0044	00000048	00	00004c	000000)50	00000054		00000058	X
000000	30200073		0000000	10		0000	0013	07f02083	08	002083	Х			00000000		X
					\bot											
					\bot											
000088					00000044						Х		00000	0078		0000004c
880000					00000044						X :		00000	0078		0000004c
				0					X	2	X			0		
000	00013 = 000000 = 000000 = 000000 = 000000 = 000000	000013 00000013 00000098 000000 00000094 000000 00000094 0000088	000013 00000013 07f02083 000000 00000000 0000000 00000000 000000	000013	00013	000013	00000	000013	00000	00013	000013	00013	00013	00013	00013	00013



Simulation (7)



Name	Value	120 ns	122 ns	124 ns	126 ns	128 ns	130 ns	132 ns	134 ns	136 ns	138 ns	140 ns	142 ns
¼ clk	0												
™ rst	1												
₩ PC_IF[31:0]	00000000	00000080	00000084	00000088	0000008c	00000090	00000094	00000098	0000009c	000000a0	00000050	00000054 000000	058
₩ inst_IF[31:0]	00000013	30002e73	30402ef3	34402f73	004c8113	34111073	30200073		00000	013		08102023 00000	0013
♥ PC_MEM[31:0]	00000000	00000058	00000078	00000070	00000080	00000084	00000088	0000008c	00000090	00000094	00000098	00000090	
Inst_MEM[31:0]	00000000	00000000	34102cf3	34202df3	30002e73	30402ef3	34402f73	004c8113	34111073	30200073	Х	00000000	
₩ PC_WB[31:0]	00000000	000	00058	00000078	0000007c	00000080	00000084	00000088	0000008c	00000090	00000094	00000 8 00000	009c
₩ inst_WB[31:0]	00000000	000	00000	34102cf3	34202df3	30002e73	30402ef3	34402f73	004c8113	34111073	30200073	00000000	
la redirect_mux_exp	0												
RegWrite_cancel_exp	0												
♥ PC_redirect_exp[31:0]	88000000	00000078	0000004c	00000005	00000080	00000fff	00000	000	0000004c		0000	10050	
csr_rw_MEM	0												
csr_w_imm_mux_MEM	X												
♥ CSRout_MEM[31:0]	00000088	00000078	0000004c	00000005	00000080	00000fff	00000	000	0000004c		0000	0050	
■ exp_vector_WB[3:0]	0							0					



Simulation (8)



Name	Value	140 ns 142 n	ns 144 r	ns 146 ns	148 n	150 ns	152 ns	154 ns	156 ns	158 ns	160 ns	162 n
¼ dk	0											
[™] rst	1					10						
₩ PC_IF[31:0]	00000000	00000054 00000058	0000005c	00000060	00000064	8900000	00000078	0000007c \ 00	000080	00000084	00000088	0000008c
₩ inst_IF[31:0]	00000013	08102023		00000013			34102cf3	34202df3 30	002e73	30402ef3	34402f73	004c8113
₩ PC_MEM[31:0]	00000000	0000009c	00000050	00000054	00000058	00000050		00000060	X	00000078	00000070	00000080
■ inst_MEM[31:0]	00000000	00000000	00000013	08102023	00000013	X	00000	000	X	34102cf3	34202df3	30002e73
₩ PC_WB[31:0]	00000000	00000098 0	000009c	00000050	00000054	00000058	0000005c	00	000060		00000078	0000007c
Inst_WB[31:0]	00000000	0000000	0	00000013	08102023	X		00000000			34102cf3	34202df3
redirect_mux_exp	0					N. Company						
RegWrite_cancel_exp	0											
♥ PC_redirect_exp[31:0]	88000000		0000005	0		X	00000	078	X	00000054	00000007	00000080
d csr_rw_MEM	0											
d csr_w_imm_mux_MEM	Х											
♥ CSRout_MEM[31:0]	00000088		0000005	0		X	00000	078	X	00000054	00000007	00000080
₩ exp_vector_WB[3:0]	0		0		1	X			0			



Simulation (9)



Name	Value	160 ns	162 ns	164 ns	166 ns	168 ns	170 ns	172 ns	174 ns	176 ns	178 ns 180 ns	182
[™] clk	0											
¹₄ rst	1											
> W PC_IF[31:0]	00000000	00000088	0000008c	00000090	00000094	00000098	0000009c	00000040	00000058	X 0000005c X 000000	060 00000064	00000068
> W inst_IF[31:0]	00000013	34402f73	004c8113	34111073	30200073	X			000	00013		
> ₩ PC_MEM[31:0]	00000000	00000070	00000080	00000084	00000088	00000080	00000090	00000094	00000098	00000090	00000058	00000050
> W inst_MEM[31:0]	00000000	34202df3	30002e73	30402ef3	34402f73	004c8113	34111073	30200073	X	00000000	X 01	0000013
> W PC_WB[31:0]	00000000	00000078	0000007c	00000080	00000084	00000088	00000080	00000090	00000094	00000098	0000009c	00000058
> W inst_WB[31:0]	00000000	34102cf3	34202df3	30002e73	30402ef3	34402f73	004c8113	34111073	30200073	000000	000	00000013
la redirect_mux_exp	0											
RegWrite_cancel_exp	0											
> W PC_redirect_exp[31:0]	00000088	00000007	00000080	00000fff	0000	0000	00000054	(00000058		
CSr_rw_MEM	0						1 5					
csr_w_imm_mux_MEM	Х											
> W CSRout_MEM[31:0]	00000088	00000007	00000080	00000fff	0000	0000	00000054			00000058		
> W exp_vector_WB[3:0]	0							0				



Checkpoints



• CP 1:

Waveform Simulation of the Pipelined CPU with the verification program

• CP 2:

FPGA Implementation of the Pipelined CPU with the verification program





Thanks

