关于测试多周期 CPU 的简单方法

(特别说明:本表每个同学都必须建立,检查实验时,必须提供!)。

1、测试程序段

		1/10/10/11/1	17.							
				指令代						
		地址	汇编程序	op(6)	rs(5)	rt(5)	rd(5)/immediate (16)	16	进制数代	
rs	rt							码		
\$0	\$1=8	0x00000000	addiu \$1,\$0,8	001001	00000	00001	0000 0000 0000 1000	=	24010008	\$1=0+8=8
\$0	\$2=2	0x0000004	ori \$2,\$0,2	001101	00000	00010	0000 0000 0000 0010	II	34020002	\$2=0 2=2
\$2=2	\$3=10	0x00000008	xori \$3, \$2 ,8	001110	00010	00011	0000 0000 0000 1000	II	38430008	\$3=\$2^8=10
\$3	\$1=8	0x000000C	sub \$4, \$3 ,\$1	000000	00011	00001	00100 00000 100010	=	00642022	\$4=\$3-\$1=2
\$4=2	\$2=2	0x0000010	and \$5, \$4 ,\$2	000000	00100	00010	00101 00000 100100	=	00852024	\$5=2&2=2
无	\$5=2	0x0000014	sll \$5, <mark>\$5</mark> ,2	000000	00000	00101	00101 00010 000000	=	00052880	\$5=2<<2=8
\$5=8	\$1=8	0x0000018	beq \$5,\$1,-2(=,转 14)	000100	00101	00001	1111 1111 1111 1110	II	10A1FFFE	\$5=\$1(跳转
无	无	0x000001C	jal 0x0000050	000011	00000	00000	0000 0000 0001 0100	=	0C000014	\$31=PC+4
\$13=2	2 \$1=8	0x00000020	slt \$8,\$13,\$1	000000	01101	00001	01000 00000 101010	=	01A8402A	S13<\$1 \$8=1
\$0	\$14=-2	0x00000024	addiu \$14,\$0,-2	001001	00000	01110	1111 1111 1111 1110	=	2C000FFE	614=0-2=-2
\$8=1	\$14=-2	0x00000028	slt \$9,\$8,\$14	000000	01000	01110	01001 00000 101010	=	0109502A	88>\$14 \$9=0
\$9=0	\$10=1	0x0000002C	slti \$10,\$9,2	001010	01001	01010	0000 0000 0000 0010	=	252A0002	\$9<2 \$10=1
\$10	\$11=0	0x00000030	slti \$11, \$10 ,0	001010	01010	01011	0000 0000 0000 0000	=	255B0000	\$10>0 \$11=0
\$11	\$10=1	0x00000034	add \$11, \$11 ,\$10	000000	01011	01010	01011 00000 100000	=	01AB5820	\$11=0+1=1
\$11=	1 \$2=2	0x0000038	bne \$11 ,\$2,-2 (≠,转 34)	000101	01011	00010	1111 1111 1111 1110	=	1562FFFE	\$11!=\$2(跳
\$0	\$12	0x000003C	addiu \$12,\$0,-2	001001	00000	01100	1111 1111 1111 1110	=	240CFFFE	\$12=0-2=-2
\$12	\$12	0x00000040	addiu \$12,\$12,1	001001	01100	01100	0000 0000 0000 0001	=	258C0001	\$12+1=-1
\$12=-	1 无	0x00000044	bltz \$12,-2 (<0,转 40)	000001	01100	00000	1111 1111 1111 1110	=	0580FFFE	\$12<0 (跳转
\$2=2	\$12=2	0x00000048	andi \$12,\$2,2	001100	00010	01100	0000 0000 0000 0010	=	304C0002	\$12=\$2&2=2
无	无	0x0000004C	j 0x000005C	000010	00000	00000	0000 0000 0001 0111	=	08000017	PC=5C
\$1=8	\$2=2	0x0000050	sw \$2,4(\$1)	101011	00001	00010	0000 0000 0000 0100	=	AC220004	Л[(\$1+4)<<2]
\$1=8	\$13=2	0x0000054	lw \$13,4(\$1)	100011	00001	01101	0000 0000 0000 0100	=		613=M[48]
\$31=2		0x0000058	jr \$31	000000	11111	00000	00000 00000 001000	=	03E00008	PC=\$31
		0x000005C	halt	111111	00000	00000	0000 0000 0000 0000	=	FC000000	

- 2、将指令代码初始化到指令存储器中,直接写入。
- 3、初始化 PC 的值,也就是以上程序段首地址 PC=**0x00000000**,以上程序段从 **0x00000000** 地址开始存放。
- 4、运行 Xilinx Vivado 进行仿真,看波形。

Name	OP	func	PCWre	ALUSrcA	ALUSrcB	DBDataSrc	RegWre	InsMemRW	mRD	mWR	RegDst	ExtSel	PCSrc	ALUOp
add	000000	100000	1	0	0	0	1		X	0		x	00	000
sub	000000	100010	1	0	0	0	1	1	X	0	1	X	00	001
addiu	001001		1	0	1	0	1	1	×	0	0		00	000
andi	001100		1	0	1	0	1	1	×	0	0	0	00	100
and	000000	100100	1	0	0	0	1	1	×	0	1	X	00	100
ori	001101		1	0	1	0	1	1	X	0	0	0	00	011
or	000000		1	0	0	0	1	1	X	0	1	X	00	011
sll	000000	000000	1	1	0	0	1	1	X	0	1	×	00	010
slti	001010		1	0	1	0	1	1	X	0	0	1	00	110
SW	101011		1	0	1	×	0	1	×	1	X		00	000
lw	100011		1	0	1	1	1	1	1	0	0	1	00	000
beq	000100		1	0	0	×	0	1	X	0	X	1	zero01 or 00	001
bne	000101		1	0	0	×	0	1	X	0	X	1	!zero01 or 0	001
bltz	000001		1	0	0	×	0	1	X	0	X	1	sign or 00	001
j	000010		1	X	X	X	0	1	X	0	X	X	10	X
halt	111111		0	X	X	×	Х	x	Х	X	X	X	×	Х
xori	001110													
slt	000000	101010												
jr	000000	001000												
jal	000011													