

به نام خدا

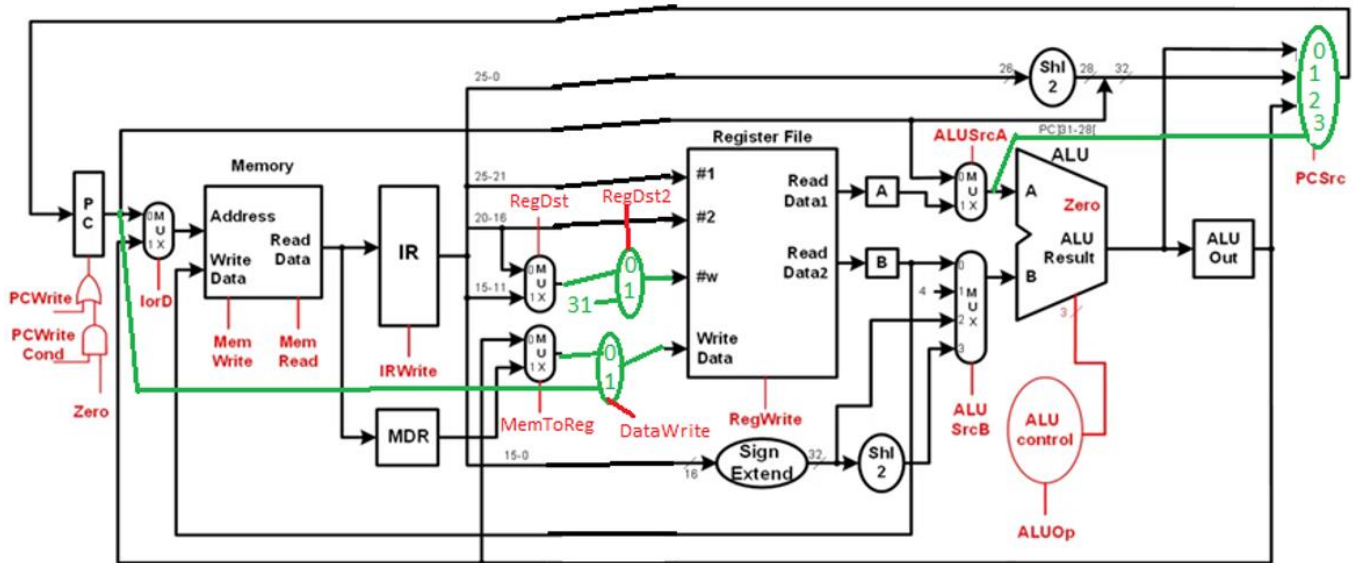
معماری کامپیوتر

تمرین کامپیوتری 3

احمد رضا شهسواری 810199445

حسین شهیدی 810899057

مسیر داده :



غالب دو دستور jal, jalr مشابه یکدیگر است (6 بیت برای آپکود و 26 بیت بعدی برای آدرس پرش)

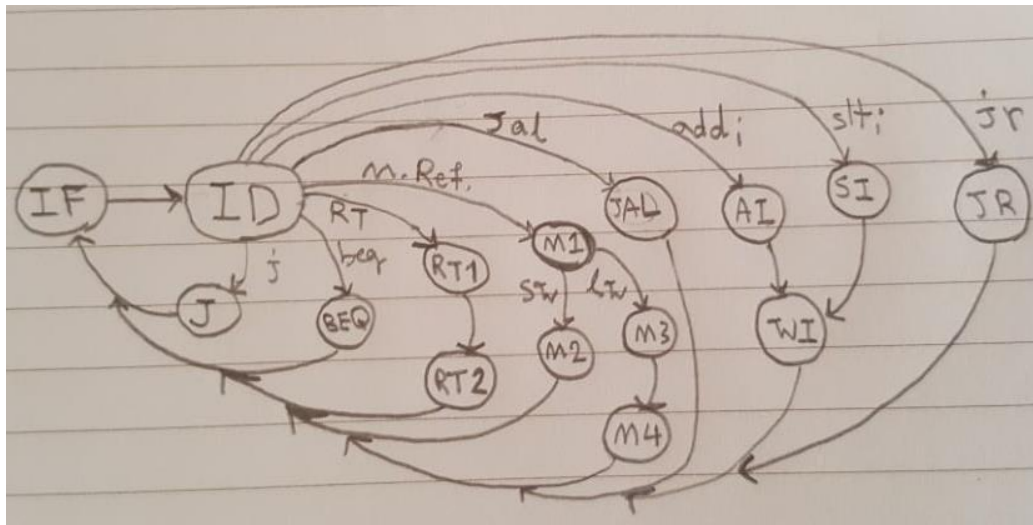
غالب دستور jr (مثلا اگر دستور jr Rs باشد)

opc				Rs				Don't care											
31	26	25	21	20	21	20	21	20	21	20	21	20	21	20	21	20	21	20	0

غالب دستور slti مشابه addi است (مثلا اگر دستور slti Rs, Rt, num باشد):

opc				Rt				Rs				num							
31	26	25	21	20	21	20	21	20	21	20	21	20	16	15	16	15	16	15	0

کنترلر :



مقادیر سیگنال های کنترلی برای هرکدام از مراحل تصویر بالا :

	IorD	Mem Read	Mem Write	IRWrite	RegDst	MemToReg	RegWrite	ALUSrcA	ALUSrcB	ALUOp	PCWrite	PCWrite Cond	RegDst 2	Data Write	PCSrc
IF	0	1	0	1	x	x	0	0	01	00(+)	1	x	x	x	00
ID	x	0	0	0	x	x	0	0	11	00(+)	0	0	x	x	x
J	x	0	0	0	x	x	0	x	x	x	1	x	x	x	01
BEQ	x	0	0	0	x	x	0	1	00	01(-)	0	1	x	x	10
RT1	x	0	0	0	x	x	0	1	00	10	0	0	x	x	x
RT2	x	0	0	0	1	0	1	x	x	x	0	0	0	0	x
M1	x	0	0	0	x	x	0	1	10	00(+)	0	0	x	x	x
M2	1	0	1	0	x	x	0	x	x	x	0	0	x	x	x
M3	1	1	0	0	x	x	0	x	x	x	0	0	x	x	x
M4	x	0	0	0	0	1	1	x	x	x	0	0	0	0	x
JAL	x	0	0	0	x	x	1	x	x	x	1	x	1	1	01
AL	x	0	0	0	x	x	0	1	10	00(+)	0	0	x	x	x
WI	x	0	0	0	0	0	1	x	x	x	0	0	0	0	x
SI	x	0	0	0	x	x	0	1	10	11(<)	0	0	x	x	x
JR	x	0	0	0	x	x	0	1	x	x	1	x	x	x	11

کدهای اسمبلی برای تست پردازنده و کدهای 32 بیتی معادل آنها:

کوچکترین عنصر در R3 و اندیس آن در R4 نوشته می شوند.

pc	کد اسمبلی	معادل 32 بیتی
0	addi R1,R0,1000	{6'd9,5'd0,5'd1,16'd1000}
4	add R2,R0,R0	{6'd0,5'd0,5'd0,5'd2,5'd0,6'h20}
8	lw R3,0(R1)	{6'h23,5'd1,5'd3,16'd0}
12	add R4,R0,R0	{6'd0,5'd0,5'd0,5'd4,5'd0,6'h20}
16	Loop: addi R1,R1,4	{6'd9,5'd1,5'd1,16'd4}
20	addi R2,R2,1	{6'd9,5'd2,5'd2,16'd1}
24	slti R5,R2,20	{6'h0A,5'd2,5'd5,16'd20}
28	beq R5,R0,End_Loop pc=(pc+4)+6*4	{6'h04,5'd5,5'd0,16'd6}
32	lw R6,0(R1)	{6'h23,5'd1,5'd6,16'd0}
36	slt R7,R6,R3	{6'd0,5'd6,5'd3,5'd7,5'd0,6'h2A}
40	beq R7,R0,Loop pc=(pc+4)-7*4	{6'h04,5'd7,5'd0,-16'd7}
44	add R3,R6,R0	{6'd0,5'd6,5'd0,5'd3,5'd0,6'h20}
48	add R4,R2,R0	{6'd0,5'd2,5'd0,5'd4,5'd0,6'h20}
52	j Loop pc=4*4	{6'h02,26'd4}
56	End_Loop: sw R3,2000(R0)	{6'h2B,5'd0,5'd3,16'd2000}
60	sw R4,2004(R0)	{6'h2B,5'd0,5'd4,16'd2004}

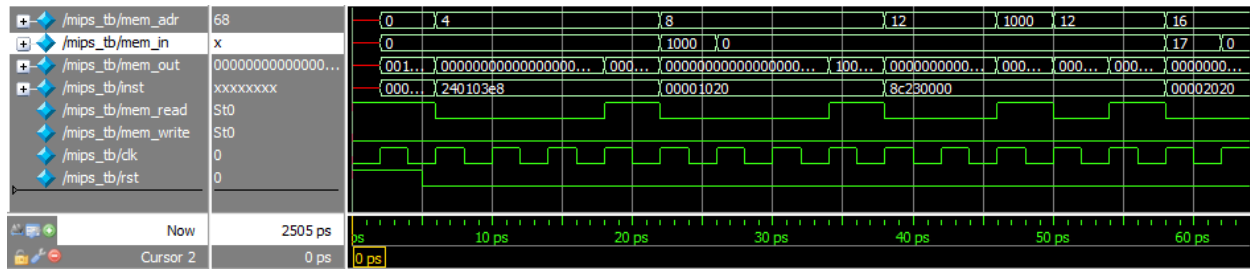
باتوجه به گنجایش حافظه، این کدهای 32 بیتی در فایل mem\_data.mem نوشته شده است(از آدرس صفر شروع شده اند) که آن را به حافظه می دهیم.

مقادیر آرایه 20 عنصری با آدرس شروع 1000 را بعنوان مثال اعداد زیر می گیریم:

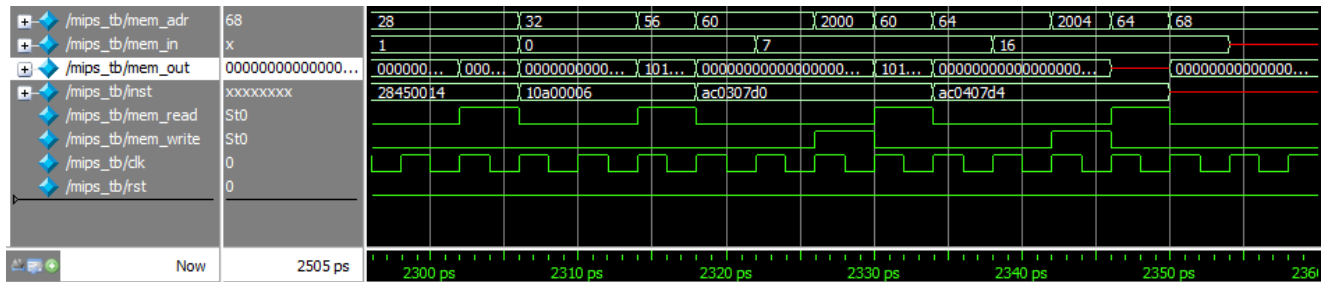
{17,20,14,19,41,15,16,71,11,34,27,24,25,12,30,31,7,10,45,9}

باتوجه به گنجایش حافظه کدهای 32 بیتی معادل این اعداد در همان فایل mem\_data.mem نوشته شده است که آن را به حافظه می دهیم(باتوجه به اینکه آدرس شروع این داده ها از 1000 است با دستورات ذخیره شده دچار تداخل نمیشوند ) ، باتوجه به این آرایه انتظار داریم در آدرس 2000 حافظه عدد 7 نوشته شود و در آدرس 2004 حافظه اندیس آن که 16 است نوشته شود،باتوجه به تصاویر خروجی در صفحه بعد می توان صحت این موضوع را بررسی کرد.

لحظه شروع به کار :



لحظه به انتها رسیدن محاسبات مربوطه :



```
# The content of mem[2000] = 7
#
# The content of mem[2004] = 16
```

بنابراین همانطور که انتظار داشتیم در خانه 2000 مقدار کوچکترین عنصر (7) و در خانه 2004 اندیس آن (16) نوشته شده است که نشانگر عملکرد درست پردازنده است.