THỰC HÀNH 5

3)

Lệnh: add $t1,$t2,$t3			
Tên khối		Ngõ	Giá trị
Bộ Cộng	Input		**0x00400000**
	Output		**0x00400004**
Instruction Memory	Input	Read address	**0x00400000**
	Output	Instruction[31-0]	**0x014b4820**
Registers	Input	Read register 1	**01010**
		Read register 2	**01011**
		Write register	**01001**
		Write data	**0x00000000**
	Output	Read data 1	**0x00000000**
		Read data 2	**0x00000000**
ALU	Input	Input thứ nhất của ALU	**0x00000000**
		Input thứ hai của ALU	**0x00000000**
	Output	ALU result	**0x00000000**
		Zero	**1**
Data Memory	Input	Address	**X**
		Write data	**X**
	Output	Read data	**X**
Control	Input	Instruction [31-26]	**000000**
	Output	RegDst	**1**
		Branch	**0**
		MemRead	**0**
		MemtoReg	**0**
		ALUOp	**10**
		MemWrite	**0**
		ALUSrc	**0**
		RegWrite	**1**
Sign-extend	Input		**0x0020**
	Output		**0x00000020**
ALU Control	Input	Instruction [5-0]	**100000**
		ALUOp	**10**
	Output	Operation	**0010 (add)**
Shift-left-2 (dùng cho lệnh beq)	Input		**0x00000020**
	Output		**0x00000080**

Lệnh: addi $t1,$t1,5			
Tên khối		Ngõ	Giá trị
Bộ Cộng	Input		**0x00400000**
	Output		**0x00400004**
	Input	Read address	**0x00400000**

Instruction Memory	Output	Instruction[31-0]	**0x21290005**
Registers	Input	Read register 1	**01001**
		Read register 2	**01001**
		Write register	**01001**
		Write data	**0x00000005**
	Output	Read data 1	**0x00000000**
		Read data 2	**0x00000000**
ALU	Input	Input thứ nhất của ALU	**0x00000000**
		Input thứ hai của ALU	**0x00000005**
	Output	ALU result	**0x00000005**
		Zero	**0**
Data Memory	Input	Address	**X**
		Write data	**X**
	Output	Read data	**X**
Control	Input	Instruction [31-26]	**001000**
	Output	RegDst	**0**
		Branch	**0**
		MemRead	**0**
		MemtoReg	**0**
		ALUOp	**10**
		MemWrite	**0**
		ALUSrc	**1**
		RegWrite	**1**
Sign-extend	Input		**0x0005**
	Output		**0x00000005**
ALU Control	Input	Instruction [5-0]	**000101**
		ALUOp	**10**
	Output	Operation	**0010 (add)**
Shift-left-2 (dùng cho lệnh beq)	Input		**0x00000005**
	Output		**0x00000014**

Lệnh: sub $t1,$t2,$3			
Tên khối		Ngõ	Giá trị
Bộ Cộng	Input		**0x00400000**
	Output		**0x00400004**
Instruction Memory	Input	Read address	**0x00400000**
	Output	Instruction[31-0]	**0x01434822**
Registers	Input	Read register 1	**01010**
		Read register 2	**01010**
		Write register	**01001**
		Write data	**0x00000000**
	Output	Read data 1	**0x00000000**
		Read data 2	**0x00000000**

ALU	Input	Input thứ nhất của ALU	**0x00000000**
		Input thứ hai của ALU	**0x00000000**
	Output	ALU result	**0x00000000**
		Zero	**1**
Data Memory	Input	Address	**X**
		Write data	**X**
	Output	Read data	**X**
Control	Input	Instruction [31-26]	**000000**
	Output	RegDst	**1**
		Branch	**0**
		MemRead	**0**
		MemtoReg	**0**
		ALUOp	**10**
		MemWrite	**0**
		ALUSrc	**0**
		RegWrite	**1**
Sign-extend	Input		**0x0022**
	Output		**0x00000022**
ALU Control	Input	Instruction [5-0]	**100010**
		ALUOp	**10**
	Output	Operation	**0110 (sub)**
Shift-left-2 (dùng cho lệnh beq)	Input		**0x00000022**
	Output		**0x00000088**

Lệnh: lw $t1,4($t2)			
Tên khối		Ngõ	Giá trị
Bộ Cộng	Input		**0x00400000**
	Output		**0x00400004**
Instruction Memory	Input	Read address	**0x00400000**
	Output	Instruction[31-0]	**0x01434822**
Registers	Input	Read register 1	**01010**
		Read register 2	**01001**
		Write register	**01001**
		Write data	**0x00000000**
	Output	Read data 1	**0x10010000**
		Read data 2	**0x00000000**
ALU	Input	Input thứ nhất của ALU	**0x10010000**
		Input thứ hai của ALU	**0x00000004**
	Output	ALU result	**0x10010004**
		Zero	**0**
Data Memory	Input	Address	**0x10010004**
		Write data	**X**
	Output	Read data	**0x00000000**
Control	Input	Instruction [31-26]	**100011**

	Output	RegDst	**0**
		Branch	**0**
		MemRead	**1**
		MemtoReg	**1**
		ALUOp	**00**
		MemWrite	**0**
		ALUSrc	**1**
		RegWrite	**1**
Sign-extend	Input		**0x0004**
	Output		**0x00000004**
ALU Control	Input	Instruction [5-0]	**000100**
		ALUOp	**00**
	Output	Operation	**0010 (add)**
Shift-left-2 (dùng cho lệnh beq)	Input		**0x00000004**
	Output		**0x00000010**

Lệnh: sw $t1,8($t2)			
Tên khối		Ngõ	Giá trị
Bộ Cộng	Input		**0x00400000**
	Output		**0x00400004**
Instruction Memory	Input	Read address	**0x00400000**
	Output	Instruction[31-0]	**0xad490008**
Registers	Input	Read register 1	**01010**
		Read register 2	**01001**
		Write register	**X**
		Write data	**X**
	Output	Read data 1	**0x10010020**
		Read data 2	**0x00000000**
ALU	Input	Input thứ nhất của ALU	**0x10010020**
		Input thứ hai của ALU	**0x00000008**
	Output	ALU result	**0x10010008**
		Zero	**0**
Data Memory	Input	Address	**0x10010008**
		Write data	**0x00000000**
	Output	Read data	**X**
Control	Input	Instruction [31-26]	**001000**
	Output	RegDst	**X**
		Branch	**0**
		MemRead	**0**
		MemtoReg	**X**
		ALUOp	**00**
		MemWrite	**1**
		ALUSrc	**1**
		RegWrite	**0**

Sign-extend	Input		**0x0008**
	Output		**0x00000008**
ALU Control	Input	Instruction [5-0]	**000100**
		ALUOp	**00**
	Output	Operation	**0010 (add)**
Shift-left-2 (dùng cho lệnh beq)	Input		**0x00000008**
	Output		**0x00000032**

Lệnh: j label label: exit			
Tên khối		Ngõ	Giá trị
Bộ Cộng	Input		**0x00400000**
	Output		**0x00400004**
Instruction Memory	Input	Read address	**0x00400000**
	Output	Instruction[31-0]	**0x08100001**
Registers	Input	Read register 1	**00000**
		Read register 2	**10000**
		Write register	**X**
		Write data	**X**
	Output	Read data 1	**0x00000000**
		Read data 2	**0x00000000**
ALU	Input	Input thứ nhất của ALU	**0x00000000**
		Input thứ hai của ALU	**X**
	Output	ALU result	**X**
		Zero	**1**
Data Memory	Input	Address	**X**
		Write data	**X**
	Output	Read data	**X**
Control	Input	Instruction [31-26]	**000010**
	Output	RegDst	**X**
		Branch	**0**
		MemRead	**0**
		MemtoReg	**X**
		ALUOp	**00**
		MemWrite	**0**
		ALUSrc	**X**
		RegWrite	**X**
Sign-extend	Input		**0x0001**
	Output		**0x00000001**
ALU Control	Input	Instruction [5-0]	**000001**
		ALUOp	**00**
	Output	Operation	**0110**

Shift-left-2 (dùng cho lệnh beq)	Input		**0x00000001**
	Output		**0x00000004**

Lệnh: slt $t1, $t2, $t3			
Tên khối		Ngõ	Giá trị
Bộ Cộng	Input		**0x00400000**
	Output		**0x00400004**
Instruction Memory	Input	Read address	**0x00400000**
	Output	Instruction[31-0]	**0x014b482a**
Registers	Input	Read register 1	**01010**
		Read register 2	**01011**
		Write register	**01001**
		Write data	**0x00000000**
	Output	Read data 1	**0x00000000**
		Read data 2	**0x00000000**
ALU	Input	Input thứ nhất của ALU	**0x00000000**
		Input thứ hai của ALU	**0x00000000**
	Output	ALU result	**0x00000000**
		Zero	**1**
Data Memory	Input	Address	**X**
		Write data	**X**
	Output	Read data	**X**
Control	Input	Instruction [31-26]	**000010**
	Output	RegDst	**1**
		Branch	**0**
		MemRead	**0**
		MemtoReg	**0**
		ALUOp	**10**
		MemWrite	**0**
		ALUSrc	**0**
		RegWrite	**1**
Sign-extend	Input		**0x0000**
	Output		**0x00000000**
ALU Control	Input	Instruction [5-0]	**101010**
		ALUOp	**10**
	Output	Operation	**0111 (slt)**
Shift-left-2 (dùng cho lệnh beq)	Input		**0x0000002a**
	Output		**0x004000a8**

4)

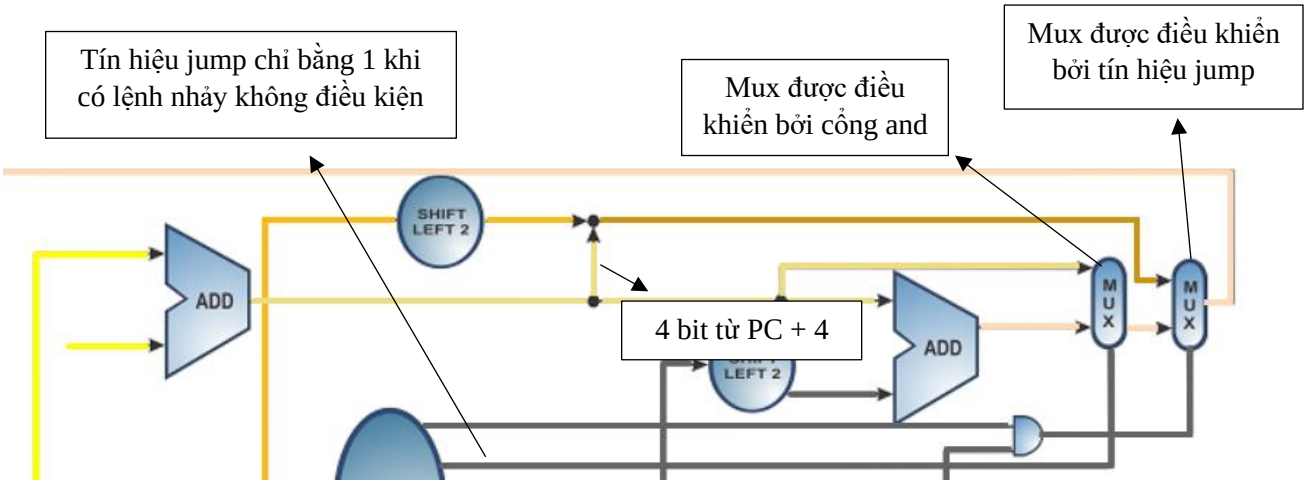
1) 2 lệnh sau là lệnh I-type do đối số có 2 thanh ghi và một số immediate

```
1   addi $s0, $s0, 6
2   addi $s1, $s1, 5
```

2 lệnh sau là lệnh R-type do có 3 thanh ghi và opcode = 0

```
3   sub $s3, $s0, $s1
4   add $s4, $s0, $s1
```

Do mô phỏng mips bị sai nên nhóm xin phép chỉnh lại như này ạ

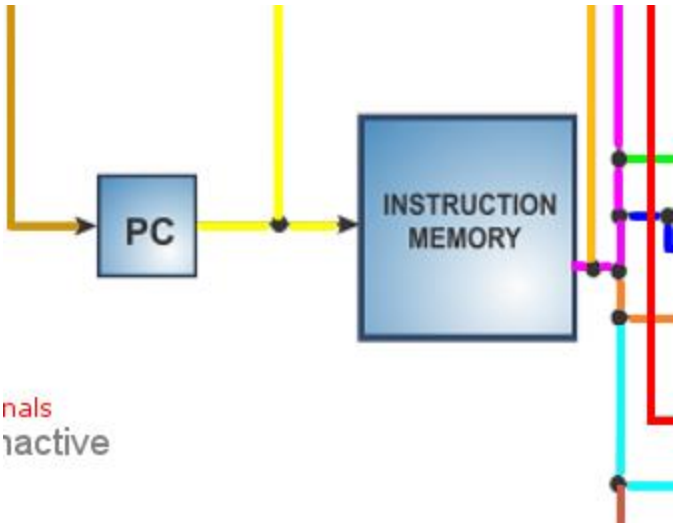


Lệnh addi $s0, $s0, 6

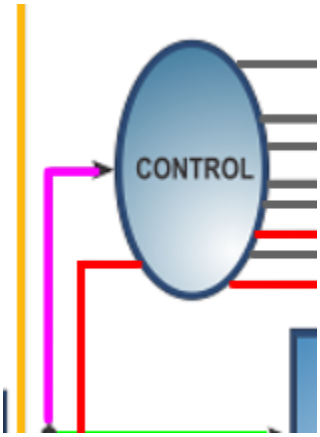
Đầu tiên lệnh sẽ được đọc từ địa chỉ ở PC bởi I – mem, I – mem sẽ cho ra output gồm lần lượt là opcode, rs, rt và immediate.

Mã nhị phân:

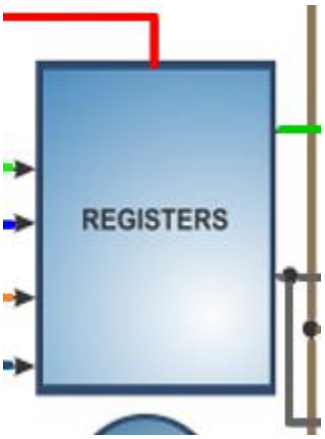
opcode	rs	rt	immediate
001000	10000	10000	0000000000000110



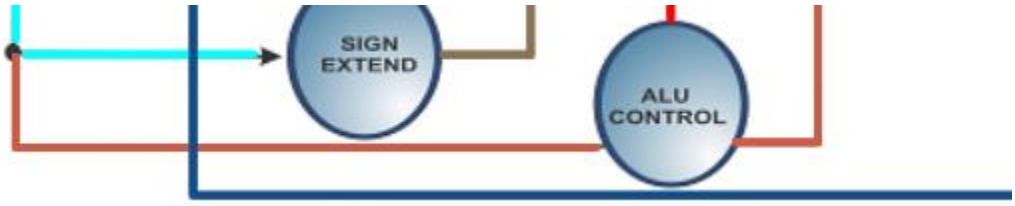
Output opcode sẽ đi đến khối control để cho ra tin hiệu điều khiển đường đi dữ liệu của datapath.



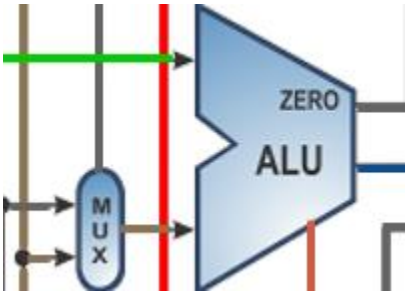
Tiếp đến là khối registers thì input Read Register 1 sẽ là rs và Read Register 2 là rt. Do lệnh này sẽ lưu giá trị xuống rt nên Register sẽ có RegWrite là 1 và RegDst sẽ là 0 dẫn đến mux truyền vào Write Register địa chỉ rt. Khối Registers sẽ cho ra input lần lượt là giá trị của đọc được từ địa chỉ rs, rt.



Output Immediate của I – mem sẽ đi tới Sign – Extented để từ tín hiệu 16bit thành tín hiện 32bit. Và 6bit trọng số thấp của Immediate sẽ đi tới ALU Control

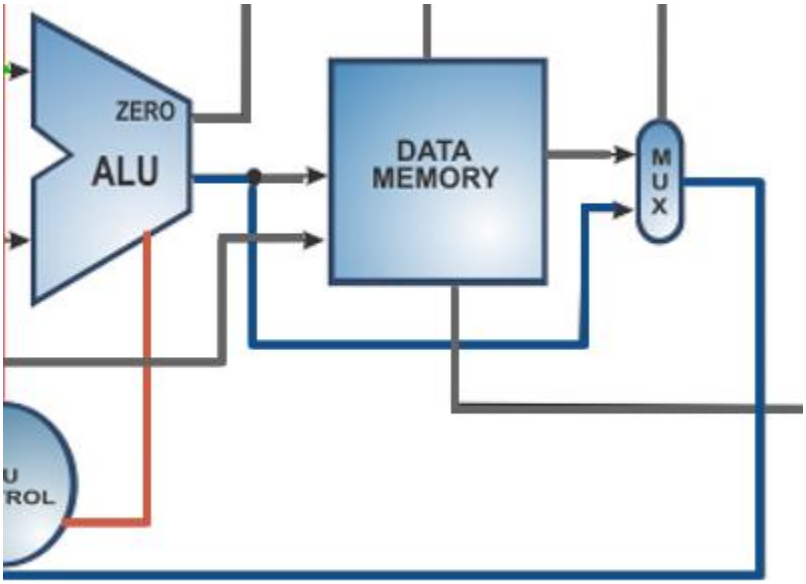


ALU sẽ nhận input đầu tiên là rs. Ở khối control sẽ có 1 tín hiệu ALUSrc để điều khiển mux cho input thứ hai của ALU. Do lệnh là lệnh addi tức là tổng của một thanh ghi với một số thì tín hiệu ALUScr sẽ mang giá trị 1 dẫn đến input thứ hai của ALU là Immediate 32bit.



Ở khối ALU control sau khi nhận tín hiệu từ ALUop và function thì nó sẽ cho ra tín hiệu 4 bit 0010 để điều khiển ALU thực hiện phép tính

Do lệnh này là lệnh addi nên datapath sẽ không sử dụng output zero và khối D – mem. Kết quả từ result của ALU sẽ đi tới mux được điều khiển bởi tín hiệu MemtoReg. Do chúng ta muốn lưu giá trị xuống rt nên MemtoReg sẽ bằng 0 và tín hiệu đầu ra của mux sẽ là result. Result sẽ chạy tới input Write Data của Registers để Registers có thể lưu kết quả xuống rt



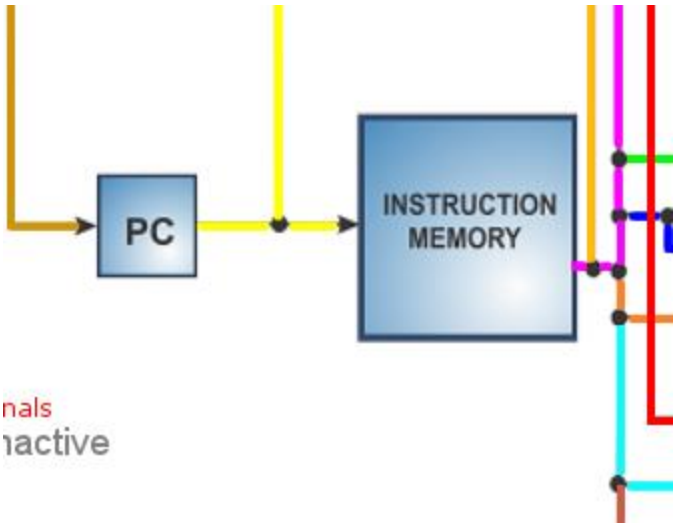
Sau cùng PC sẽ cập nhật thành PC + 4

Lệnh addi $s1, $s1, 5

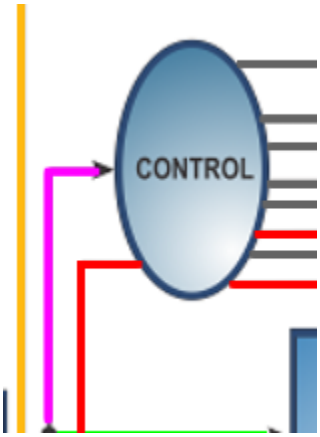
Đầu tiên lệnh sẽ được đọc từ địa chỉ ở PC bởi I – mem, I – mem sẽ cho ra output gồm lần lượt là opcode, rs, rt và immediate.

Mã nhị phân:

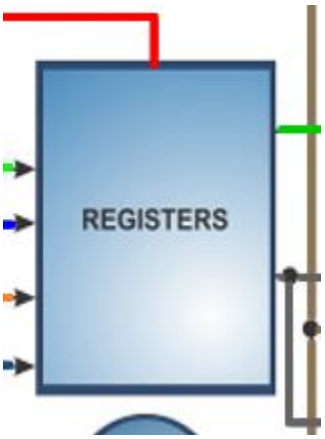
opcode	rs	rt	immediate
001000	10001	10001	0000000000000101



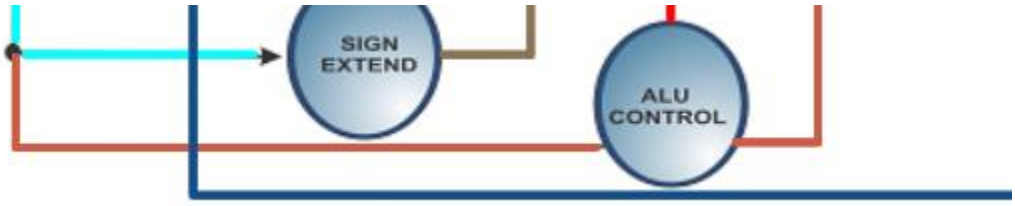
Output opcode sẽ đi đến khối control để cho ra tin hiệu điều khiển đường đi dữ liệu của datapath.



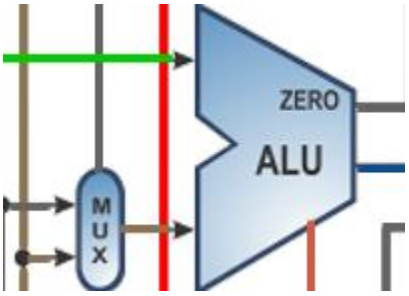
Tiếp đến là khối registers thì input Read Register 1 sẽ là rs và Read Register 2 là rt. Do lệnh này sẽ lưu giá trị xuống rt nên Register sẽ có RegWrite là 1 và RegDst sẽ là 0 dẫn đến mux truyền vào Write Register địa chỉ rt. Khối Registers sẽ cho ra input lần lượt là giá trị của đọc được từ địa chỉ rs, rt.



Output Immediate của I – mem sẽ đi tới Sign – Extented để từ tín hiệu 16bit thành tín hiện 32bit. Và 6bit trọng số thấp của Immediate sẽ đi tới ALU Control

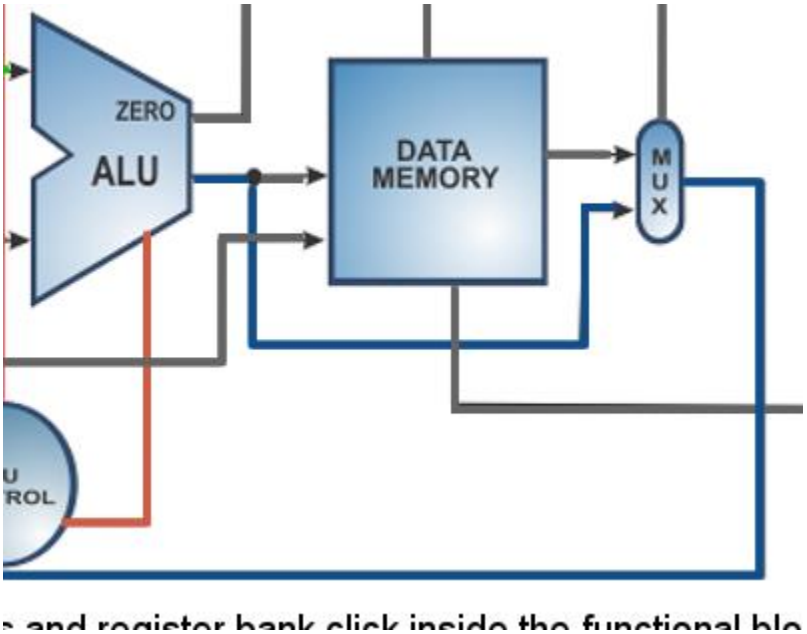


ALU sẽ nhận input đầu tiên là rs. Ở khối control sẽ có 1 tín hiệu ALUSrc để điều khiển mux cho input thứ hai của ALU. Do lệnh là lệnh addi tức là tổng của một thanh ghi với một số thì tín hiệu ALUScr sẽ mang giá trị 1 dẫn đến input thứ hai của ALU là Immediate 32bit.



Ở khối ALU control sau khi nhận tín hiệu từ ALUop và function thì nó sẽ cho ra tín hiệu 4 bit 0010 để điều khiển ALU thực hiện phép tính

Do lệnh này là lệnh addi nên datapath sẽ không sử dụng output zero và khối D – mem. Kết quả từ result của ALU sẽ đi tới mux được điều khiển bởi tín hiệu MemtoReg. Do chúng ta muốn lưu giá trị xuống rt nên MemtoReg sẽ bằng 0 và tín hiệu đầu ra của mux sẽ là result. Result sẽ chạy tới input Write Data của Registers để Registers có thể lưu kết quả xuống rt



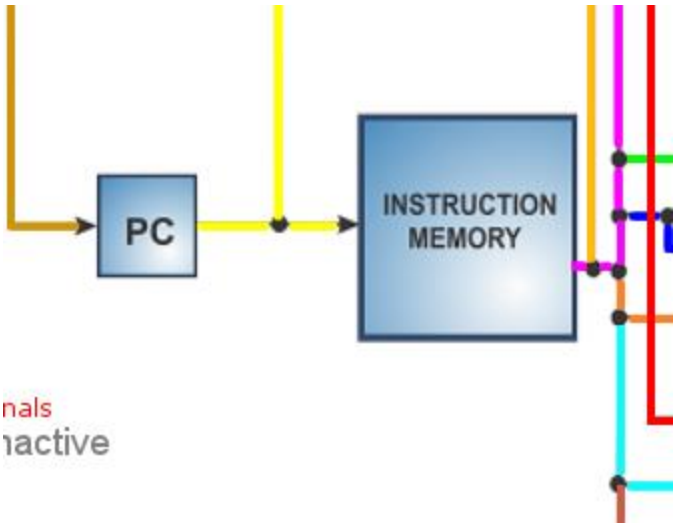
Sau cùng PC sẽ cập nhật thành PC + 4

Lệnh sub $s3, $s0, $s1

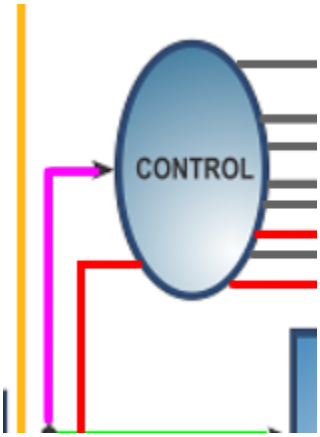
Đầu tiên lệnh sẽ được đọc từ địa chỉ ở PC bởi I – mem, I – mem sẽ cho ra output gồm lần lượt là opcode, rs, rt, rd, shamt và function

Mã nhị phân:

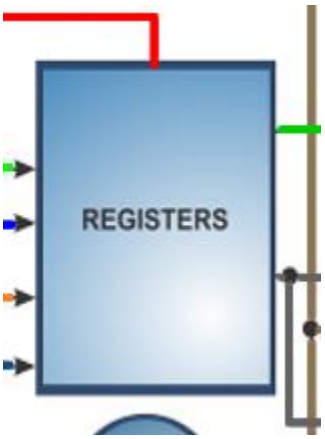
opcode	rs	rt	rd	shamt	function
000000	10000	10001	10011	00000	100010



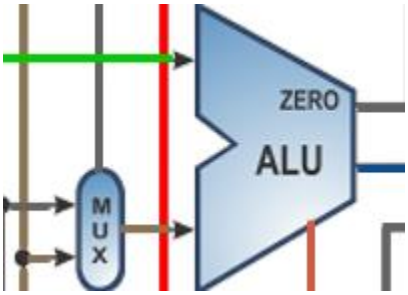
Output opcode sẽ đi đến khối control để cho ra tin hiệu điều khiển đường đi dữ liệu của datapath.



Tiếp đến là khối registers thì input Read Register 1 sẽ là rs và Read Register 2 là rt. Do lệnh này sẽ lưu giá trị xuống rd nên Register sẽ có RegWrite là 1 và RegDst sẽ là 1 dẫn đến mux truyền vào Write Register địa chỉ rd. Khối Registers sẽ cho ra input lần lượt là giá trị của đọc được từ địa chỉ rs, rt.

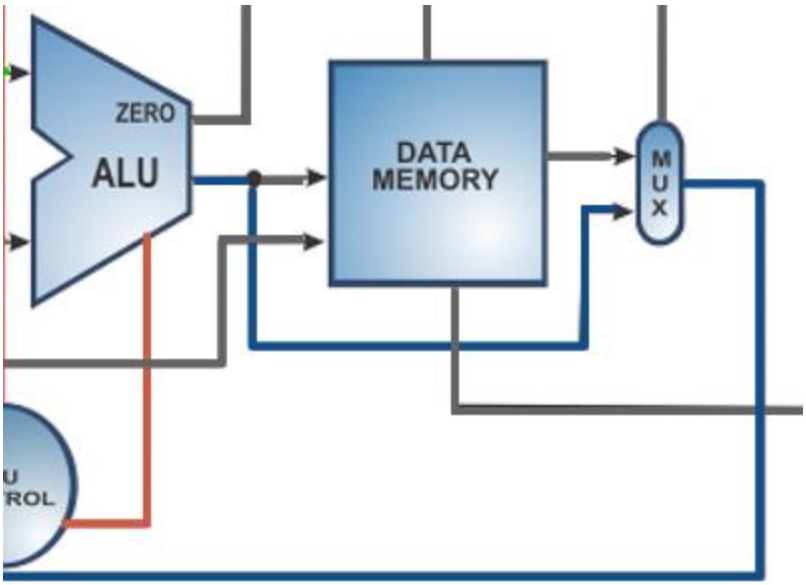


ALU sẽ nhận input đầu tiên là rs. Ở khối control sẽ có 1 tín hiệu ALUSrc để điều khiển mux cho input thứ hai của ALU. Do lệnh là lệnh sub tức là phép trừ một thanh ghi với một thanh ghi thì tín hiệu ALUScr sẽ mang giá trị 0 dẫn đến input thứ hai của ALU là rt.



Ở khối ALU control sau khi nhận tín hiệu từ ALUop và function thì nó sẽ cho ra tín hiệu 4 bit 0110 để điều khiển ALU thực hiện phép tính

Do lệnh này là lệnh sub nên datapath sẽ không sử dụng output zero và khối D – mem. Kết quả từ result của ALU sẽ đi tới mux được điều khiển bởi tín hiệu MemtoReg. Do chúng ta muốn lưu giá trị xuống rd nên MemtoReg sẽ bằng 0 và tín hiệu đầu ra của mux sẽ là result. Result sẽ chạy tới input Write Data của Registers để Registers có thể lưu kết quả xuống rd



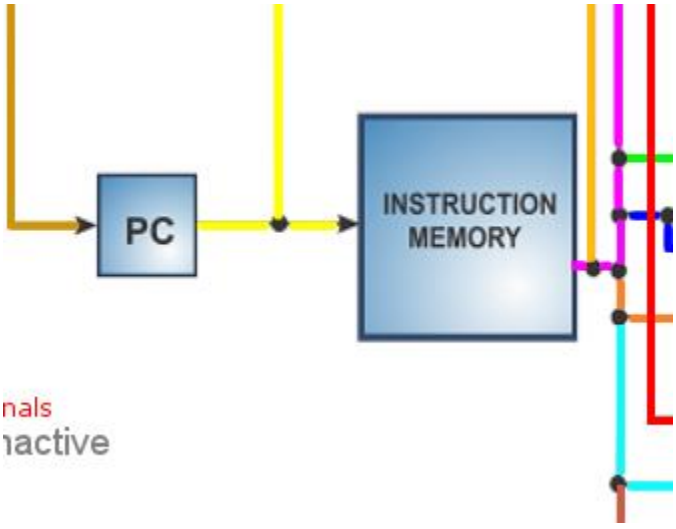
Sau cùng PC sẽ cập nhật thành PC + 4

Lệnh add $s4, $s0, $s1

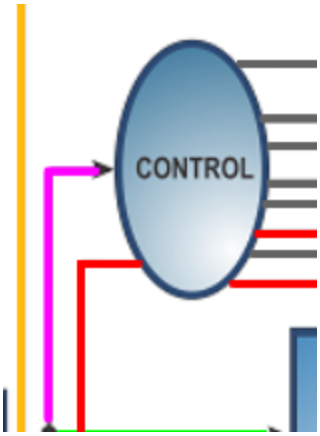
Đầu tiên lệnh sẽ được đọc từ địa chỉ ở PC bởi I – mem, I – mem sẽ cho ra output gồm lần lượt là opcode, rs, rt ,rd, shamt và function

Mã nhị phân:

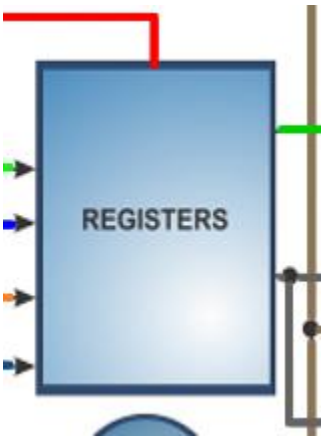
opcode	rs	rt	rd	shamt	function
000000	10000	10001	10100	00000	100000



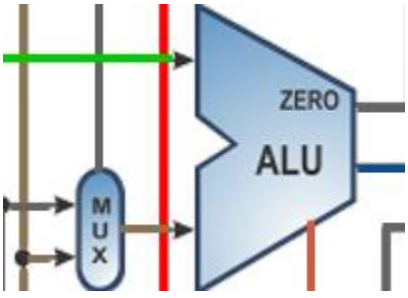
Output opcode sẽ đi đến khối control để cho ra tin hiệu điều khiển đường đi dữ liệu của datapath.



Tiếp đến là khối registers thì input Read Register 1 sẽ là rs và Read Register 2 là rt. Do lệnh này sẽ lưu giá trị xuống rd nên Register sẽ có RegWrite là 1 và RegDst sẽ là 1 dẫn đến mux truyền vào Write Register địa chỉ rd. Khối Registers sẽ cho ra input lần lượt là giá trị của đọc được từ địa chỉ rs, rt.

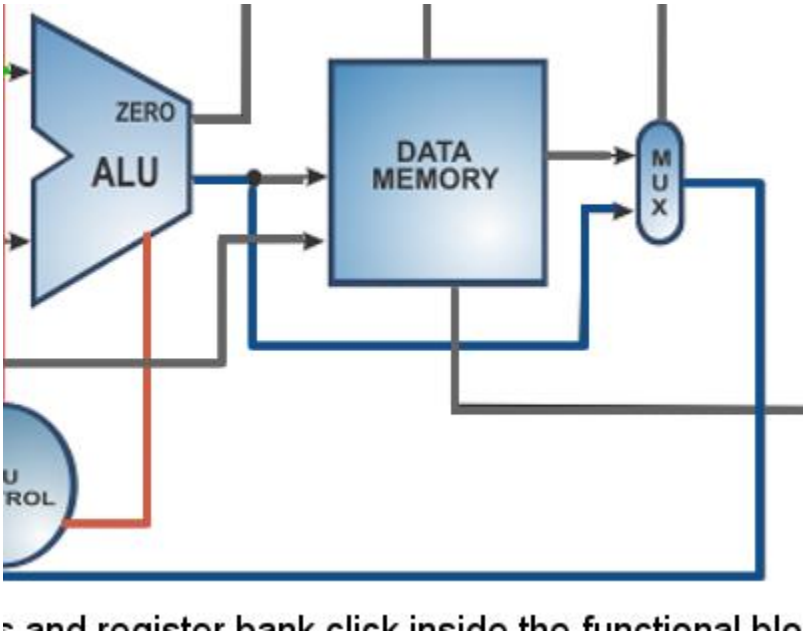


ALU sẽ nhận input đầu tiên là rs. Ở khối control sẽ có 1 tín hiệu ALUSrc để điều khiển mux cho input thứ hai của ALU. Do lệnh là lệnh add tức là phép toán một thanh ghi với một thanh ghi thì tín hiệu ALUScr sẽ mang giá trị 0 dẫn đến input thứ hai của ALU là rt.



Ở khối ALU control sau khi nhận tín hiệu từ ALUop và function thì nó sẽ cho ra tín hiệu 4 bit 0010 để điều khiển ALU thực hiện phép tính

Do lệnh này là lệnh add nên datapath sẽ không sử dụng output zero và khối D – mem. Kết quả từ result của ALU sẽ đi tới mux được điều khiển bởi tín hiệu MemtoReg. Do chúng ta muốn lưu giá trị xuống rd nên MemtoReg sẽ bằng 0 và tín hiệu đầu ra của mux sẽ là result. Result sẽ chạy tới input Write Data của Registers để Registers có thể lưu kết quả xuống rd



Sau cùng PC sẽ cập nhật thành PC + 4

2) Chương trình sau khi chuyển sang MIPS

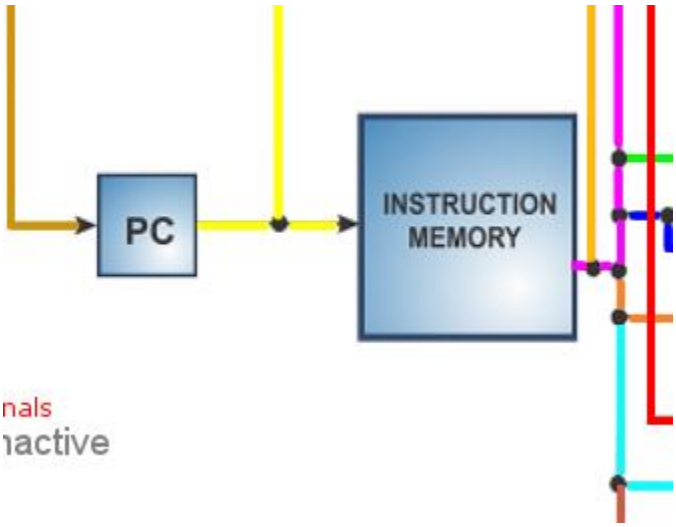
```
1   bne $s3, $s4, else
2   add $s0, $s1, $s2
3   j exit
4   else: sub $s0, $s1, $s2
5   exit:
```

bne $s3, $s4, else

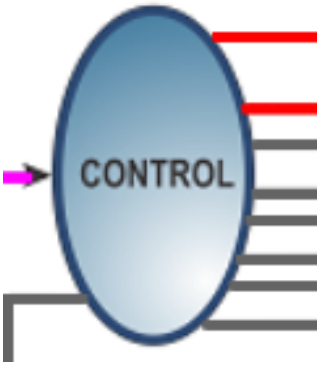
Đầu tiên lệnh sẽ được đọc từ địa chỉ ở PC bởi I – mem, I – mem sẽ cho ra output gồm lần lượt là opcode, rs, rt và immediate.

Mã nhị phân:

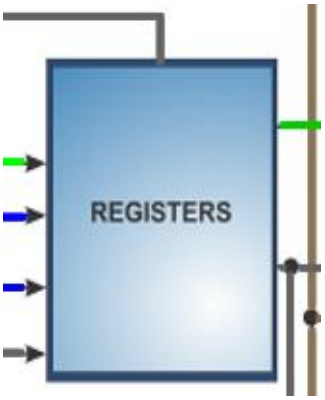
opcode	rs	rt	immediate
000101	10011	10100	0000000000000010



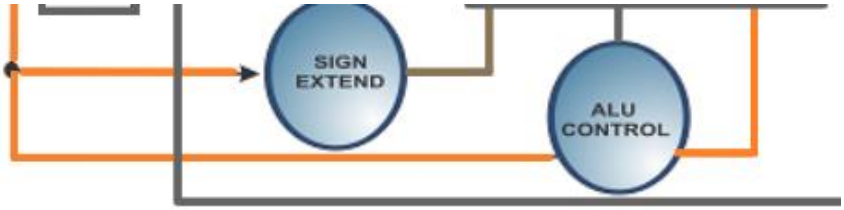
Output opcode sẽ đi đến khối control để cho ra tin hiệu điều khiển đường đi dữ liệu của datapath.



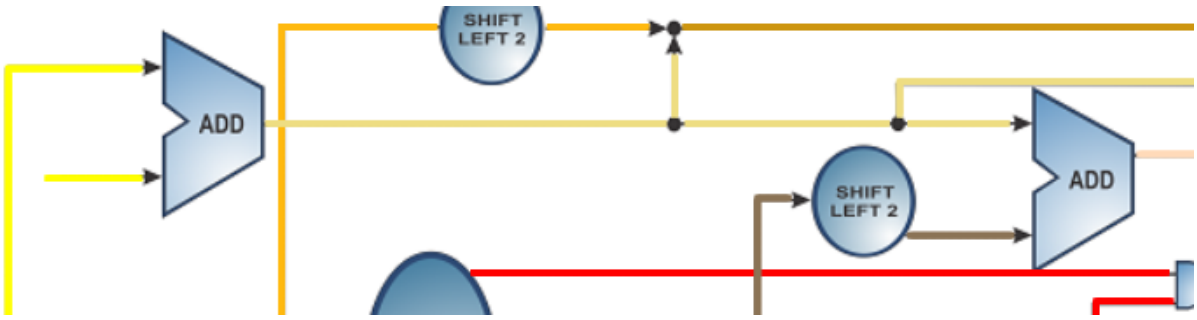
Tiếp đến là khối registers thì input Read Register 1 sẽ là rs và Read Register 2 là rt. Do lệnh này sẽ không lưu giá trị nên Register sẽ có RegWrite là 0 và RegDst sẽ là X. Khối Registers sẽ cho ra input lần lượt là giá trị của đọc được từ địa chỉ rs, rt.



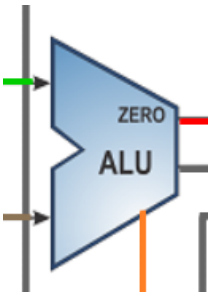
Output Immediate của I – mem sẽ đi tới Sign – Extented để từ tín hiệu 16bit thành tín hiện 32bit. Và 6bit trọng số thấp của Immediate sẽ đi tới ALU Control



Tín hiệu Immediate sau khi thành tín hiệu 32bit sẽ đi tới khối shift-left 2 để nhân 4. Tín hiệu sau khi nhân 4 thì sẽ đi vào bộ cộng, cộng với PC + 4



ALU sẽ nhận input đầu tiên là rs. Ở khối control sẽ có 1 tín hiệu ALUSrc để điều khiển mux cho input thứ hai của ALU. Do lệnh là lệnh bne nên ALUScr sẽ mang giá trị 0 dẫn đến input thứ hai của ALU là rt.



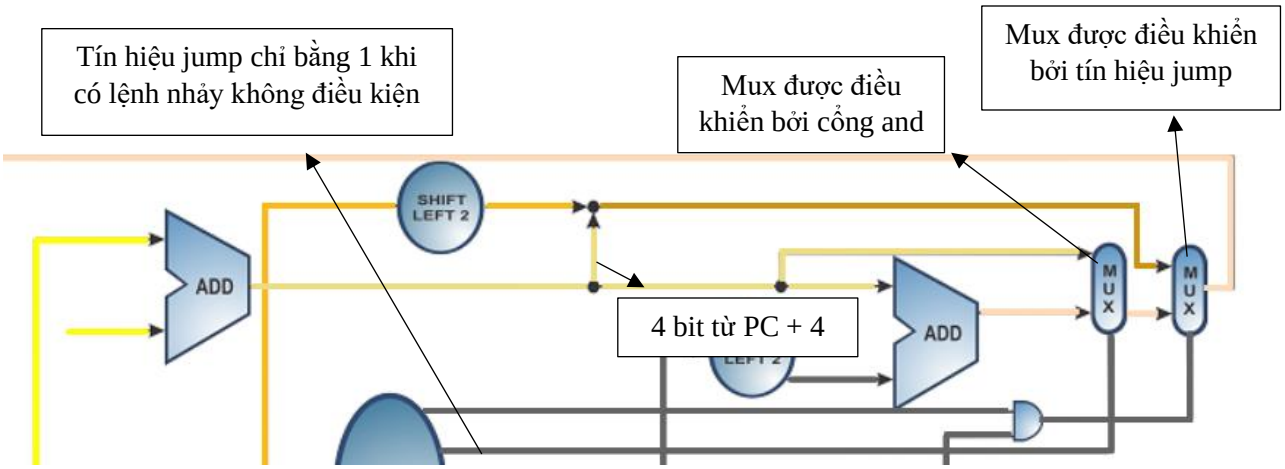
Ở khối ALU control sau khi nhận tín hiệu từ ALUop và function thì nó sẽ cho ra tín hiệu 4 bit 0110 để điều khiển ALU thực hiện phép tính

Do lệnh này là lệnh bne nên datapath sẽ không sử dụng output result và khối D – mem. Tín hiệu từ zero của ALU sẽ đi tới cổng and. Cổng and sẽ nhận thêm giá branch = 1 nữa nên sẽ kết quả phụ thuộc vào zero

Tùy theo output cổng and thì input thứ hai của mux tiếp theo sẽ có 2 trường hợp:

+ trường hợp tín hiệu bằng 0 thì sẽ lấy giá trị PC + 4

+ trường hợp tín hiệu bằng 1 thì sẽ lấy giá trị Immediate 32bit * 4 + PC + 4



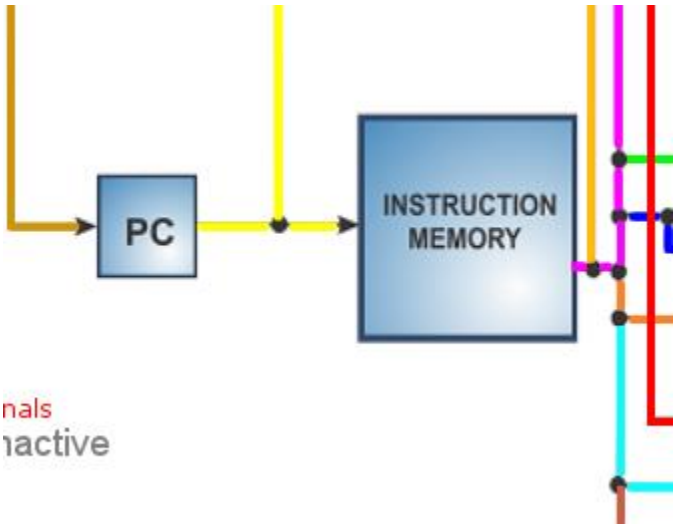
Sau cùng PC sẽ cập nhật thành giá trị từ output cổng mux trước (do jump = 0)

add $s0, $s1, $s2

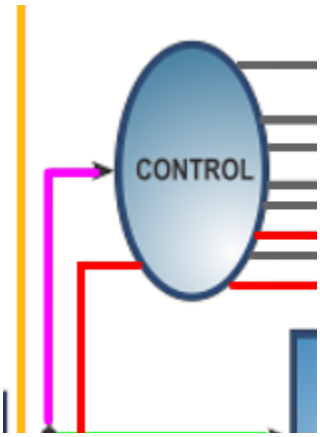
Đầu tiên lệnh sẽ được đọc từ địa chỉ ở PC bởi I – mem, I – mem sẽ cho ra output gồm lần lượt là opcode, rs, rt, rd, shamt và function

Mã nhị phân:

opcode	rs	rt	rd	shamt	function
000000	10001	10010	10000	00000	100000

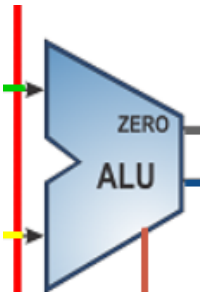


Output opcode sẽ đi đến khối control để cho ra tin hiệu điều khiển đường đi dữ liệu của datapath.



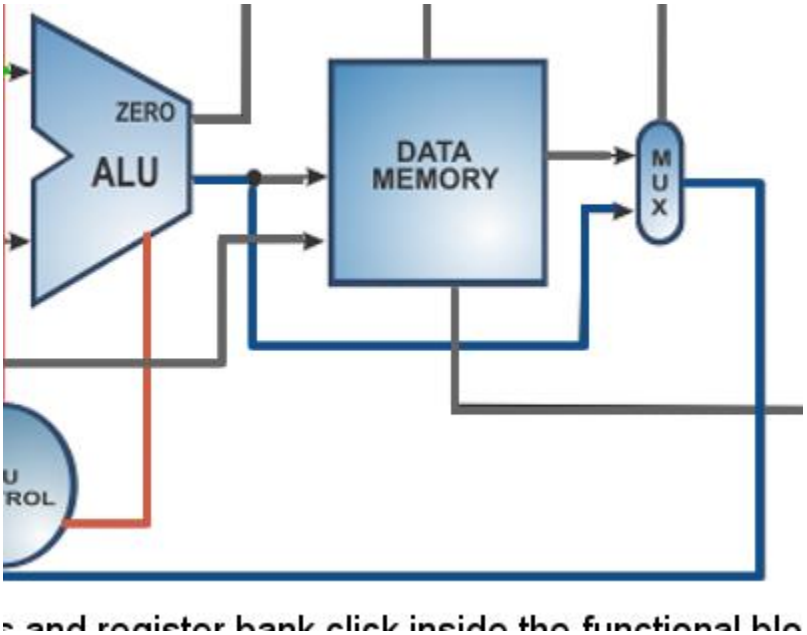
Tiếp đến là khối registers thì input Read Register 1 sẽ là rs và Read Register 2 là rt. Do lệnh này sẽ lưu giá trị xuống rd nên Register sẽ có RegWrite là 1 và RegDst sẽ là 1 dẫn đến mux truyền vào Write Register địa chỉ rd. Khối Registers sẽ cho ra input lần lượt là giá trị của đọc được từ địa chỉ rs, rt.

ALU sẽ nhận input đầu tiên là rs. Ở khối control sẽ có 1 tín hiệu ALUSrc để điều khiển mux cho input thứ hai của ALU. Do lệnh là lệnh add tức là phép toán một thanh ghi với một thanh ghi thì tín hiệu ALUScr sẽ mang giá trị 0 dẫn đến input thứ hai của ALU là rt.



Ở khối ALU control sau khi nhận tín hiệu từ ALUop và function thì nó sẽ cho ra tín hiệu 4 bit 0010 để điều khiển ALU thực hiện phép tính

Do lệnh này là lệnh add nên datapath sẽ không sử dụng output zero và khối D – mem. Kết quả từ result của ALU sẽ đi tới mux được điều khiển bởi tín hiệu MemtoReg. Do chúng ta muốn lưu giá trị xuống rd nên MemtoReg sẽ bằng 0 và tín hiệu đầu ra của mux sẽ là result. Result sẽ chạy tới input Write Data của Registers để Registers có thể lưu kết quả xuống rd



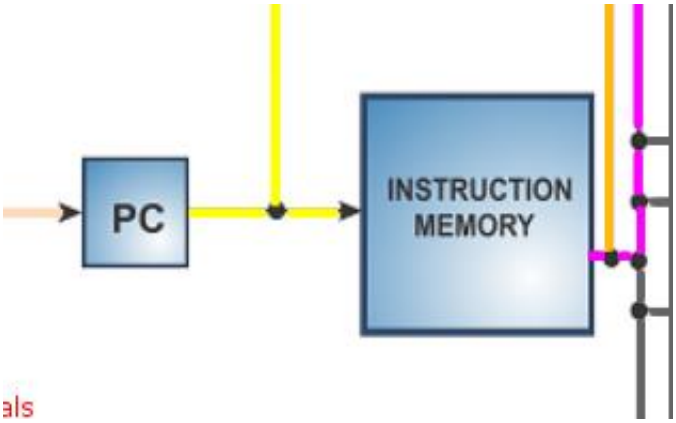
Sau cùng PC sẽ cập nhật thành PC + 4

j exit

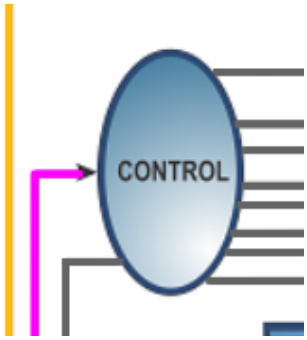
Đầu tiên lệnh sẽ được đọc từ địa chỉ ở PC bởi I – mem, I – mem sẽ cho ra output gồm lần lượt là opcode và immediate

Mã nhị phân:

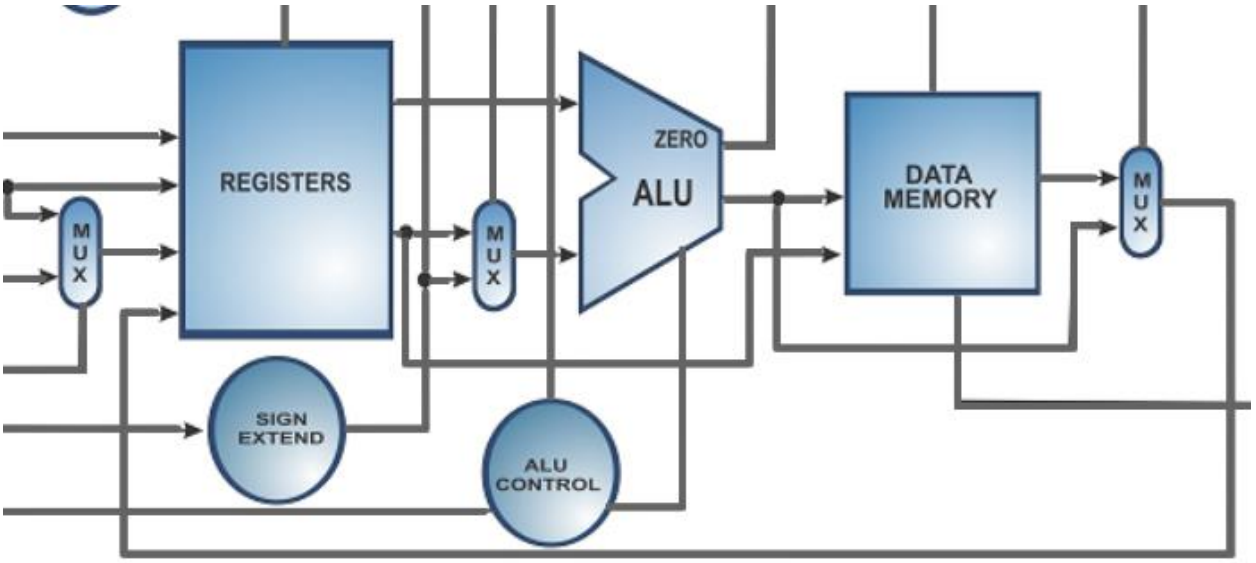
opcode	immediate
000010	00000100000000000000000100



Output opcode sẽ đi đến khối control để cho ra tin hiệu điều khiển đường đi dữ liệu của datapath.

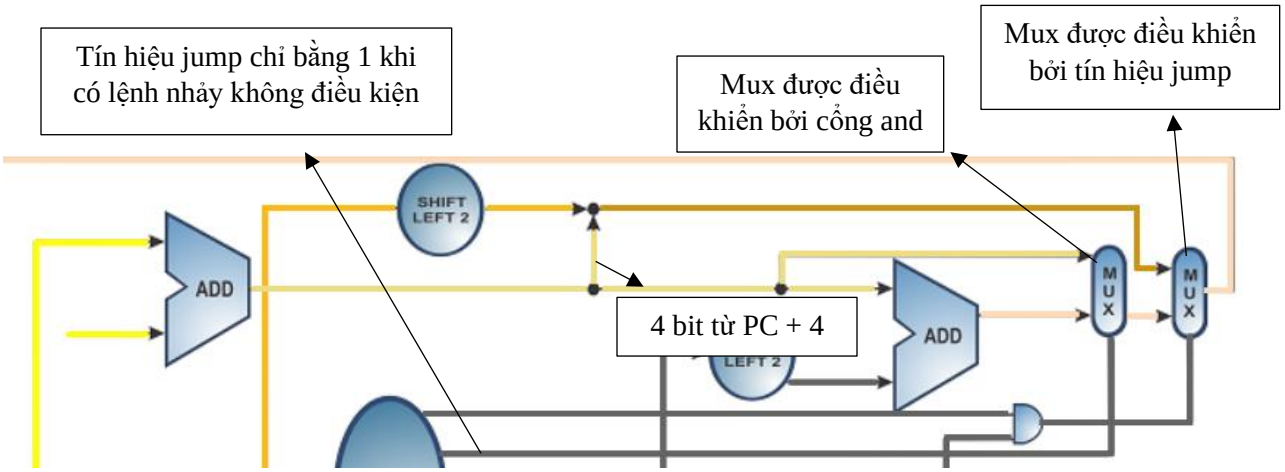


Do là lệnh nhảy không điều kiện nên khối Regs, ALU, D – mem, Sign – extend, ALU control không cần sử dụng



Output từ I – mem là chuỗi bit immediate sẽ đi thẳng lên shift – left 2 để dịch trái 2 bit. Do mux bị điều khiển bởi tín hiệu jump luôn bằng 1 nên mips

sẽ luôn lấy tín hiệu shift – left 2 và cập nhật cho PC. Mà chuỗi bit shift – left 2 vẫn chưa đủ (thiếu 4 bit) nên chúng ta sẽ cần thêm 4 bit trọng số cao từ tín hiệu PC + 4

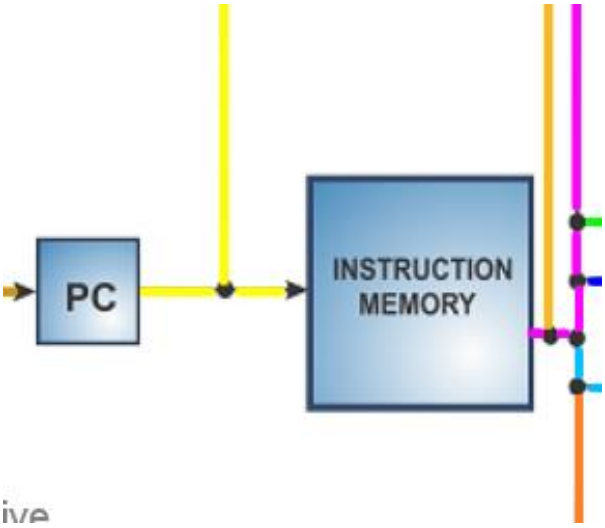


sub $s0, $s1, $s2

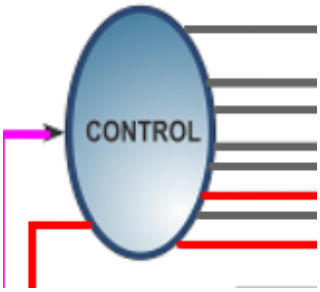
Đầu tiên lệnh sẽ được đọc từ địa chỉ ở PC bởi I – mem, I – mem sẽ cho ra output gồm lần lượt là opcode, rs, rt, rd, shamt và function

Mã nhị phân:

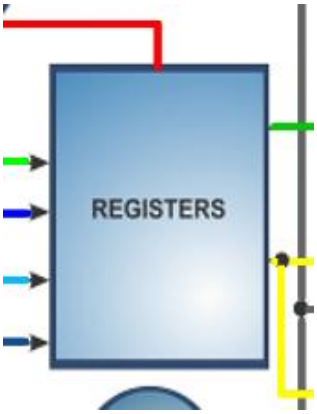
opcode	rs	rt	rd	shamt	function
000000	10001	10010	10000	00000	100010



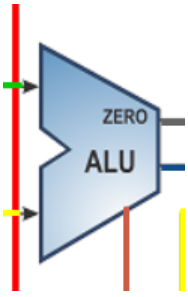
Output opcode sẽ đi đến khối control để cho ra tin hiệu điều khiển đường đi dữ liệu của datapath.



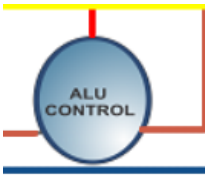
Tiếp đến là khối registers thì input Read Register 1 sẽ là rs và Read Register 2 là rt. Do lệnh này sẽ lưu giá trị xuống rd nên Register sẽ có RegWrite là 1 và RegDst sẽ là 1 dẫn đến mux truyền vào Write Register địa chỉ rd. Khối Registers sẽ cho ra input lần lượt là giá trị của đọc được từ địa chỉ rs, rt.



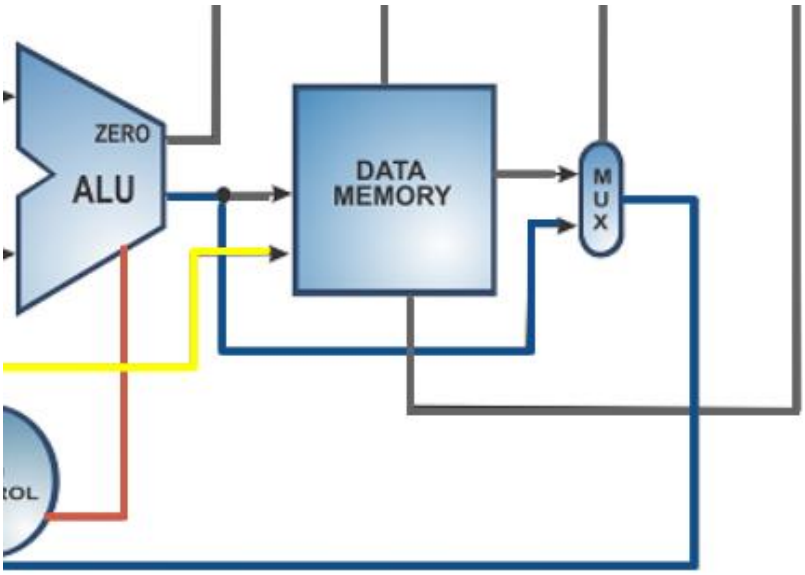
ALU sẽ nhận input đầu tiên là rs. Ở khối control sẽ có 1 tín hiệu ALUSrc để điều khiển mux cho input thứ hai của ALU. Do lệnh là lệnh sub tức là phép toán một thanh ghi với một thanh ghi thì tín hiệu ALUScr sẽ mang giá trị 0 dẫn đến input thứ hai của ALU là rt.



Ở khối ALU control sau khi nhận tín hiệu từ ALUop và function thì nó sẽ cho ra tín hiệu 4 bit 0110 để điều khiển ALU thực hiện phép tính



Do lệnh này là lệnh sub nên datapath sẽ không sử dụng output zero và khối D – mem. Kết quả từ result của ALU sẽ đi tới mux được điều khiển bởi tín hiệu MemtoReg. Do chúng ta muốn lưu giá trị xuống rd nên MemtoReg sẽ bằng 0 và tín hiệu đầu ra của mux sẽ là result. Result sẽ chạy tới input Write Data của Registers để Registers có thể lưu kết quả xuống rd



Sau cùng PC sẽ cập nhật thành PC + 4