**Họ và tên: Đoàn Vũ Phú Minh.**

**MSSV: 22520859.**

# LAB 5’S REPORT

## Section 3

**Task name :**

Chạy và quan sát quá trình xử lý các lệnh sau thông qua datapath trên MARS.

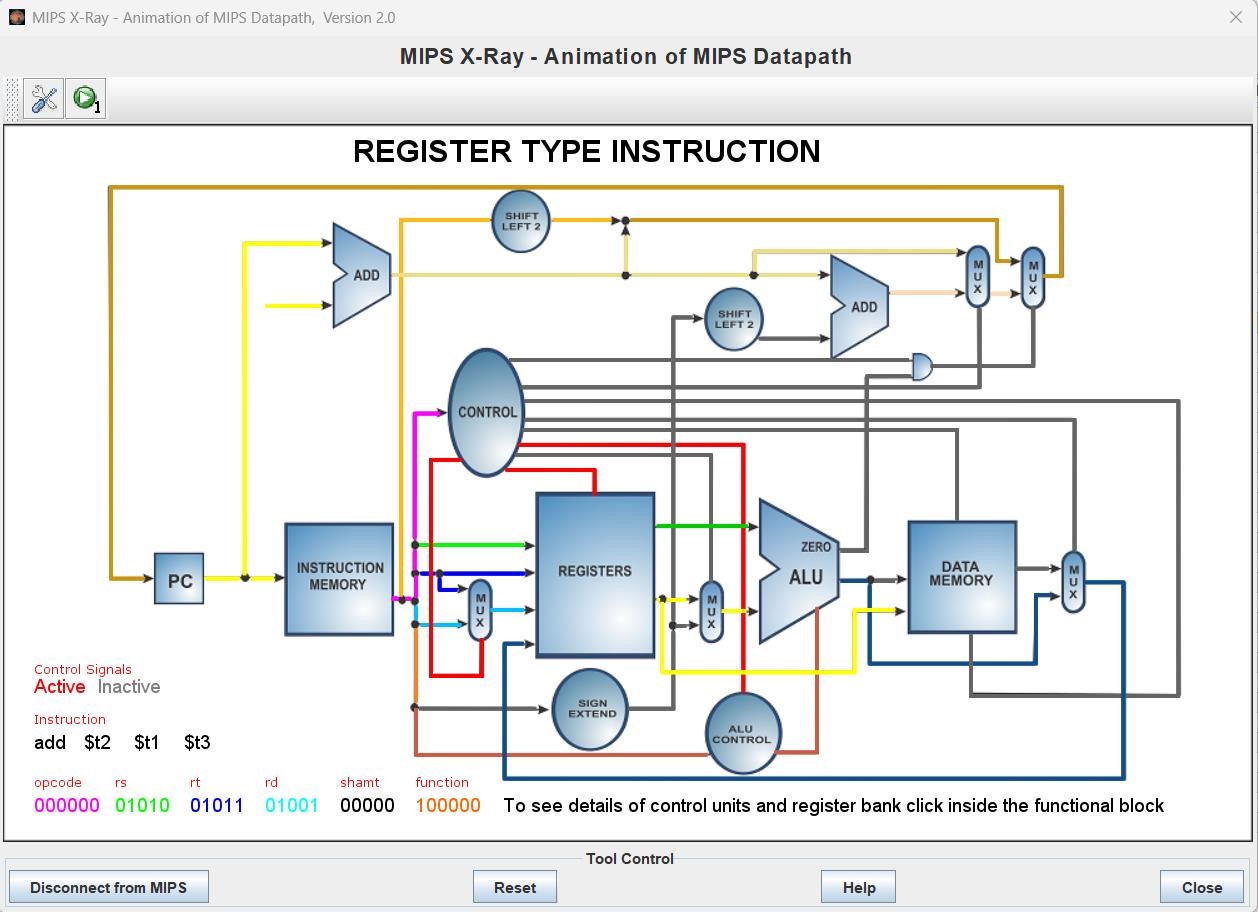
* add $t1,$t2,$t3.
* addi $t1,$t1,5
* sub $t1,$t2,$3
* lw $t1,4($t2) ; # $t2 = 0x10010000
* sw $t1,8($t2); #$t2 = 0x10010020
* J label

Label: exit

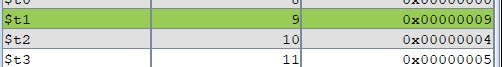
* slt $t1,$t2,$t3

\*Lệnh add $t1 ,$t2 ,$t3:

 Lệnh R – Type, thực hiện phép tính và lưu trên thanh ghi, do đó không có thao tác lưu kết quả vào memory.



*Hình 1: Đường đi Datapath của lệnh add*

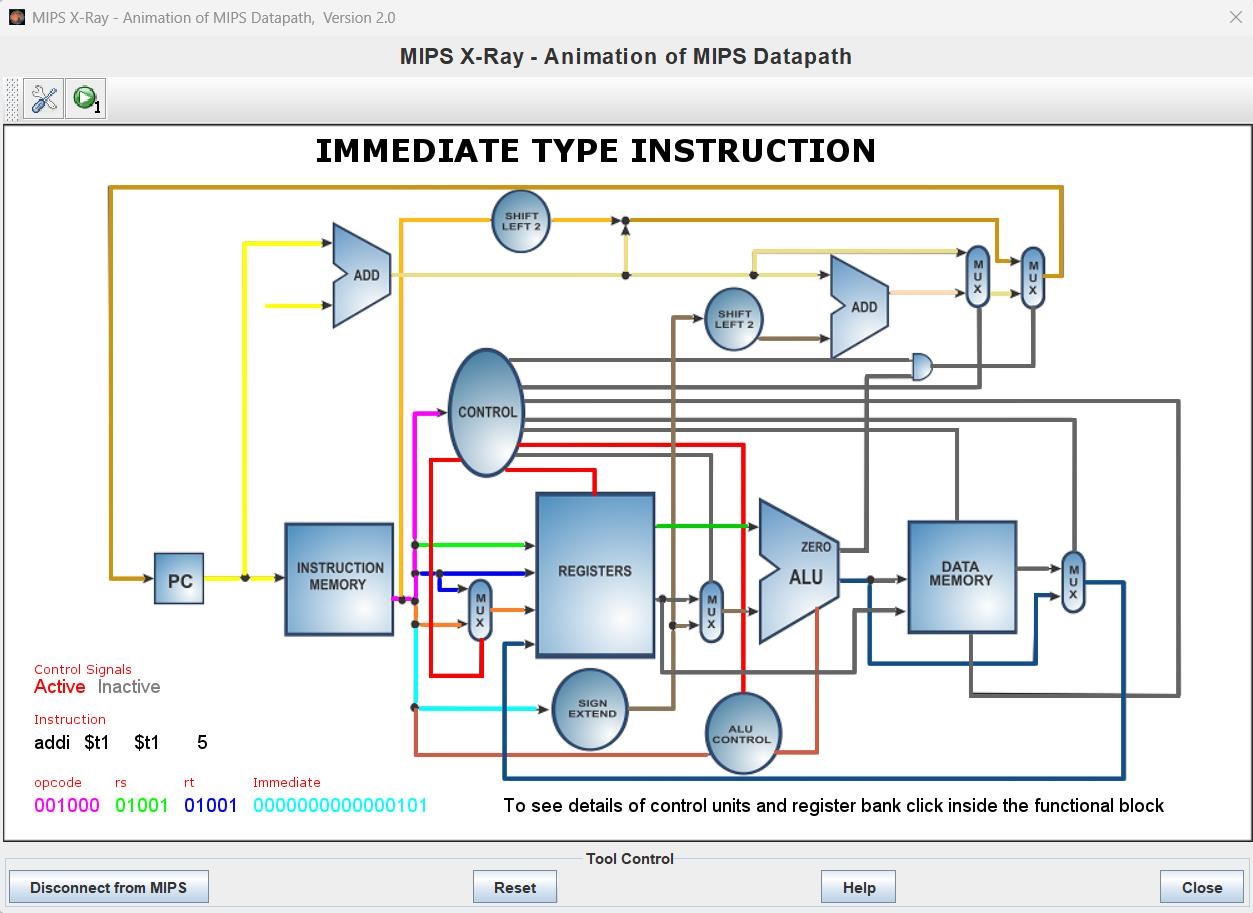


### Hình 2: Giá trị sau khi thực hiện lệnh

|  |  |  |  |
| --- | --- | --- | --- |
| *add $t1 ,$t2 ,$t3* |  |  | |
| Tên khối |  | Ngõ | Giá trị |
| **Bộ Cộng** | Input |  | **0x00400000** |
| Output |  | **0x00400004 Lệnh addi $t1, $t1, 5** |
| **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 | **0x00000009** |
| Output | Read data 1 | **0x00000004** |
| Read data 2 | **0x00000005** |
| **ALU** | Input | Input thứ nhất của ALU | **0x00000004** |
| Input thứ hai của ALU | **0x00000005** |
| Output | ALU result | **0x00000009** |
| Zero | **0** |
| **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 |  | **X** |
| Output |  | **X** |
| **ALU Control** | Input | Instruction [5-0] | **100000** |
| ALUOp | **10** |
| Output | Operation | **0010 (add)** |
| **Shift-left-2 (dùng cho lệnh beq)** | Input |  | **X** |
| Output |  | **X** |

\*Lệnh addi $t1 ,$t1 ,5:

 Lệnh I – Type nên sẽ đi qua SIGN EXTEND, thực hiện phép tính với số immediate và lưu trên thanh ghi, do đó không có thao tác lưu kết quả vào memory



### Hình 3: Đường đi Datapath của lệnh addi

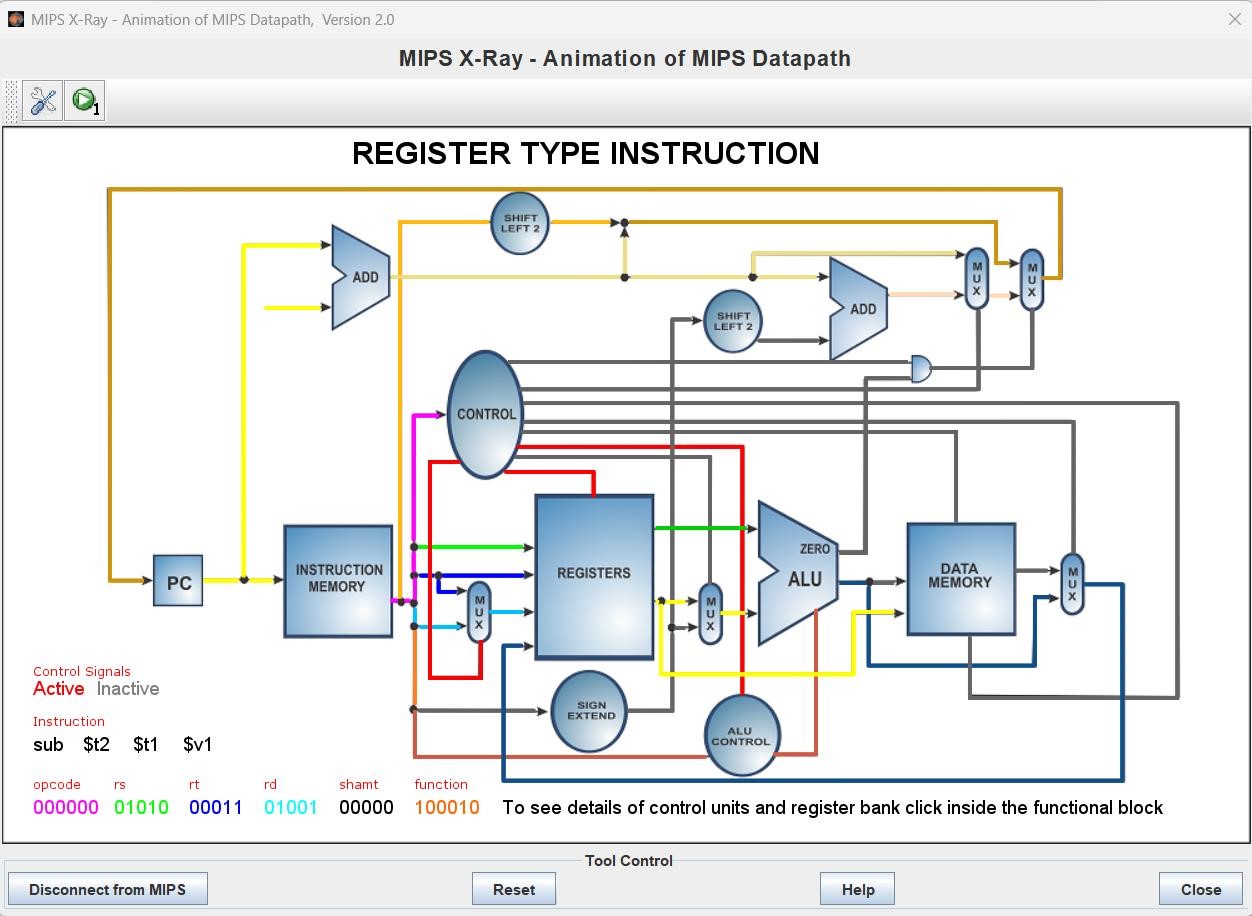


### Hình 4: Kết quả sau khi thực hiện lệnh

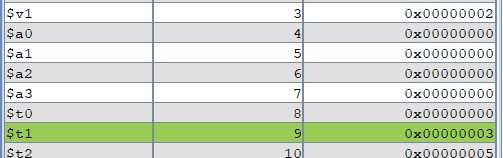
|  |  |  |  |
| --- | --- | --- | --- |
| *addi $t1, $t1, 5* |  |  |  |
| Tên khối |  | Ngõ | Giá trị |
| **Bộ Cộng** | Input |  | **0x00400004** |
| Output |  | **0x0040008 lệnh sub $t1 , $t2 , $3** |
| **Instruction**  **Memory** | Input | Read address | **0x00400004** |
| Output | Instruction[31-0] | **0x21290005** |
| **Registers** | Input | Read register 1 | **01001** |
| Read register 2 | **X** |
| Write register | **01001** |
| Write data | **0x00000005** |
| Output | Read data 1 | **0x00000000** |
| Read data 2 | **X** |
| **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 | **1** |
| 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 |  | **X** |
| Output |  | **X** |

\*Lệnh sub $t1 ,$t2 ,$3:

 Lệnh R – Type, thực hiện phép tính và lưu trên thanh ghi, do đó không có thao tác lưu kết quả vào memory.



### Hình 5: Đường đi Datapath của lệnh sub

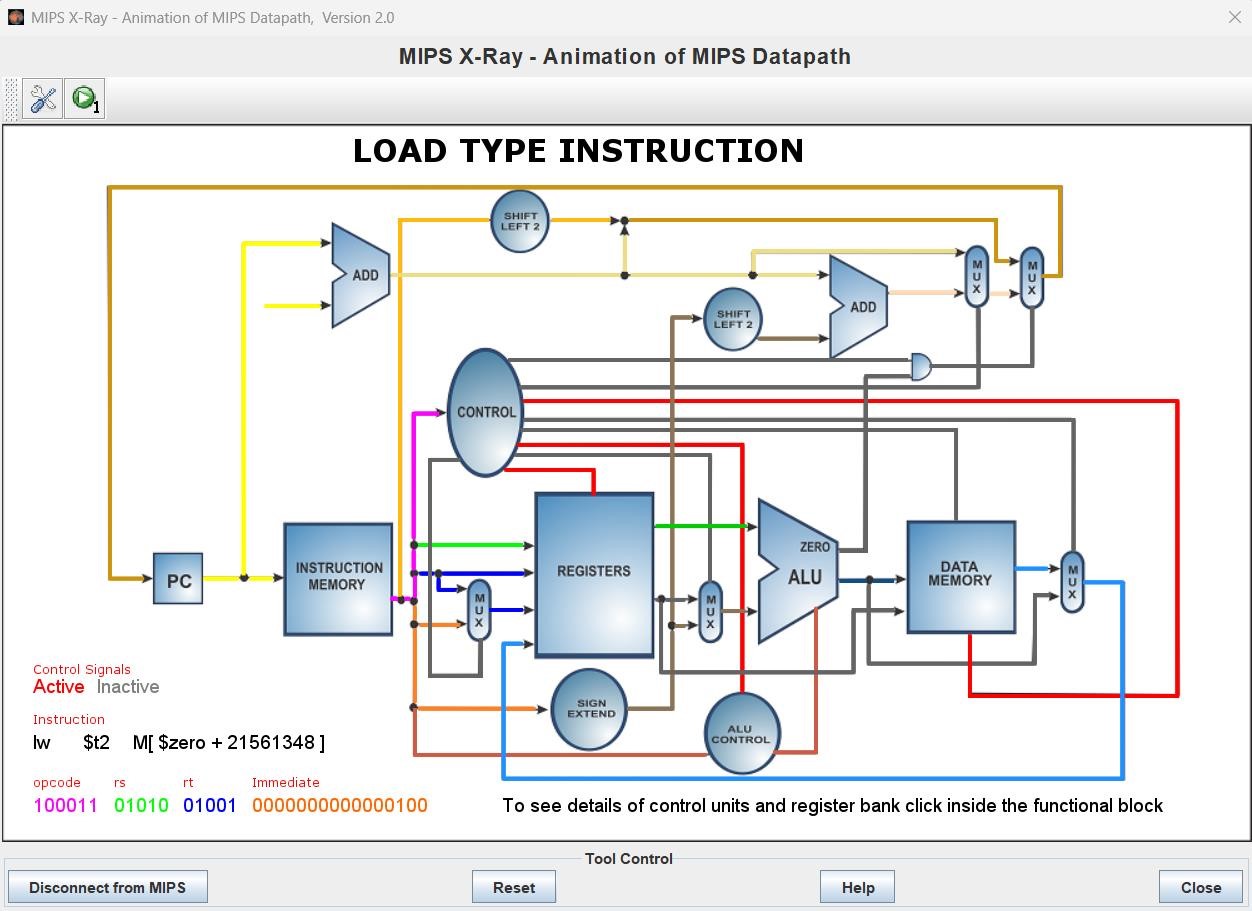


### Hình 6: Kết quả quá trình thực hiện

|  |  |  |  |
| --- | --- | --- | --- |
| *sub $t1, $t2, $3* | | | |
| Tên khối |  | Ngõ | Giá trị |
| **Bộ Cộng** | Input |  | **0x00400008** |
| Output |  | **0x0040000c lệnh lw $t1 ,4($t2)** |
| **Instruction**  **Memory** | Input | Read address | **0x00400008** |
| Output | Instruction[31-0] | **0x01434822** |
| **Registers** | Input | Read register 1 | **01010** |
| Read register 2 | **00011** |
| Write register | **01001** |
| Write data | **0x00000003** |
| Output | Read data 1 | **0x00000005** |
| Read data 2 | **0x00000002** |
| **ALU** | Input | Input thứ nhất của ALU | **0x00000005** |
| Input thứ hai của ALU | **0x00000002** |
| Output | ALU result | **0x00000003** |
| Zero | **0** |
| **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 |  | **X** |
| Output |  | **X** |
| **ALU Control** | Input | Instruction [5-0] | **100010** |
| ALUOp | **10** |
| Output | Operation | **0110 (sub)** |
| **Shift-left-2 (dùng cho lệnh beq)** | Input |  | **X** |
| Output |  | **X** |

\*Lệnh lw $t1 ,4($t2): # $t2 = 0x10010000

 Lệnh I – Type nên sẽ đi qua SIGN EXTEND, dùng để lấy dữ liệu từ Memory nên CONTROL sẽ đưa tín hiệu ReadMem đến Memory



*Hình 7: Đường đi Datapath của lệnh lw*

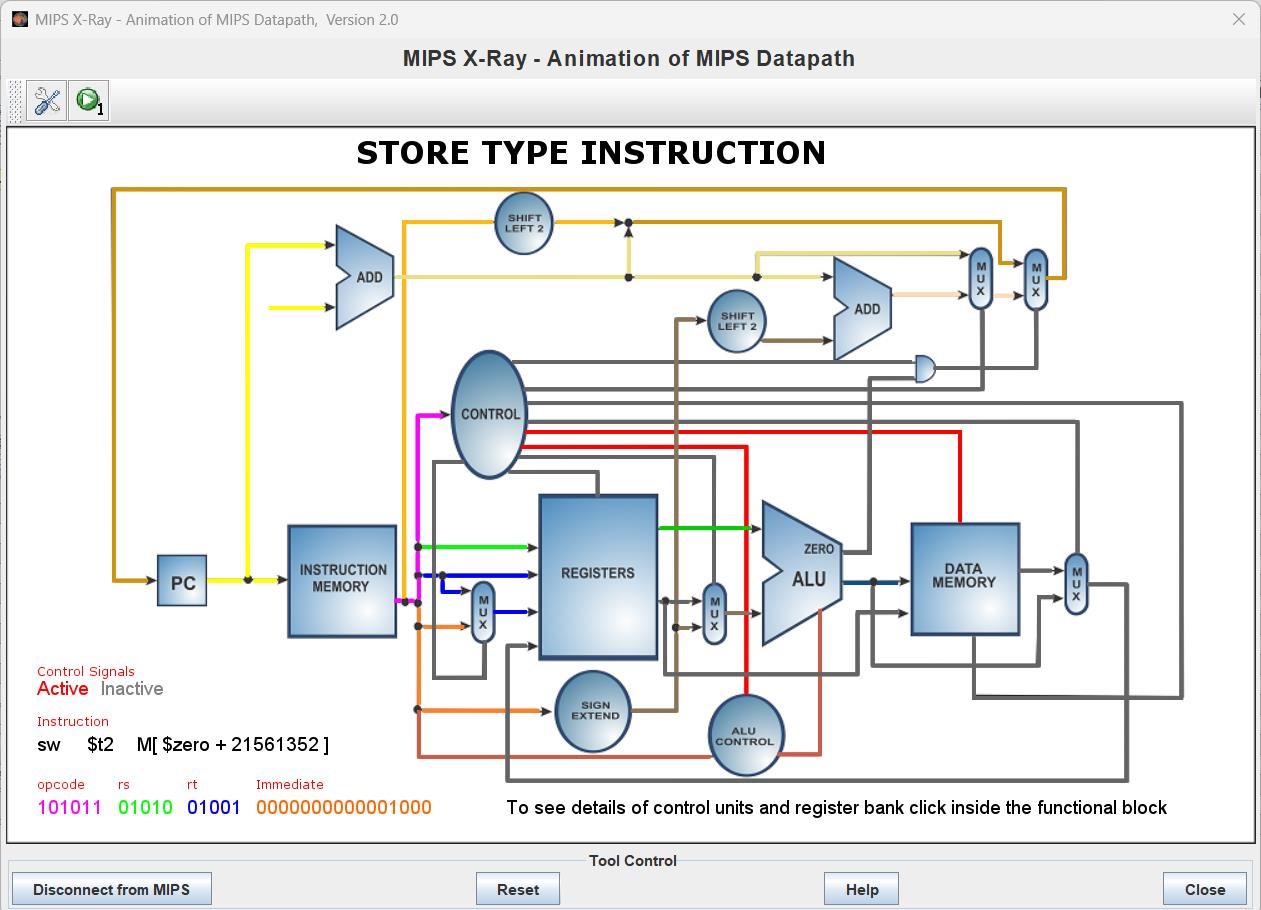


### Hình 8: Kết quả sau khi thực thi lệnh

|  |  |  |  |
| --- | --- | --- | --- |
| *lw $t1 , 4($t2)* | | | |
| Tên khối |  | Ngõ | Giá trị |
| **Bộ Cộng** | Input |  | **0x0040000c** |
| Output |  | **0x00400010 lệnh sw $t1 ,8($t2)** |
| **Instruction**  **Memory** | Input | Read address | **0x0040000c** |
| Output | Instruction[31-0] | **0x8D490004** |
| **Registers** | Input | Read register 1 | **01010** |
| Read register 2 | **X** |
| Write register | **01001** |
| Write data | **0x00000008** |
| Output | Read data 1 | **0x10010000** |
| Read data 2 | **X** |
| **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 | **0x00000008** |
| **Control** | Input | Instruction [31-26] | **101011** |
| 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 |  | **X** |
| Output |  | **X** |

\*Lệnh sw $t1 ,8($t2): #$t2 = 0x10010020

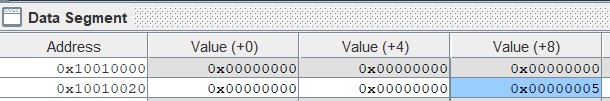
* Lệnh I – Type nên sẽ đi qua SIGN EXTEND, dùng để lấy dữ liệu từ Memory nên CONTROL sẽ đưa tín hiệu WriteMem đến Memory



*Hình 9: Đường đi Datapath của lệnh sw*



*Hình 10: Gán giá trị vào thanh ghi*



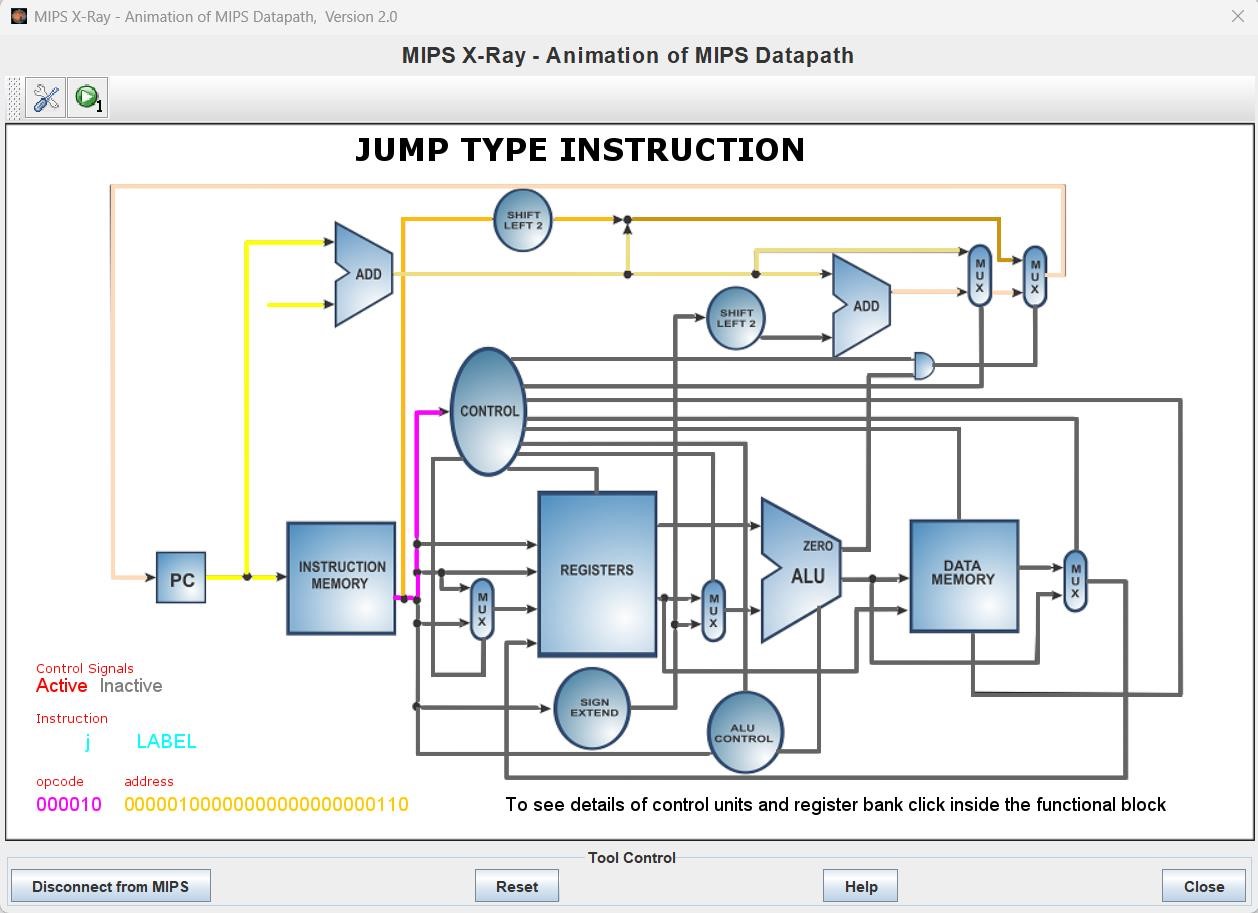
*Hình 11: Kết quả sau khi thực hiện lệnh*

|  |  |  |  |
| --- | --- | --- | --- |
| *sw $t1 , 8($t2)* | | | |
| Tên khối |  | Ngõ | Giá trị |
| **Bộ Cộng** | Input |  | **0x00400010** |
| Output |  | **0x00400014 lệnh j exit** |
| **Instruction**  **Memory** | Input | Read address | **0x00400010** |
| 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 | **0x00000005** |
| **ALU** | Input | Input thứ nhất của ALU | **0x10010020** |
| Input thứ hai của ALU | **0x00000008** |
| Output | ALU result | **0x10010028** |
| Zero | **0** |
| **Data Memory** | Input | Address | **0x10010028** |
| Write data | **0x00000005** |
| Output | Read data | **X** |
| **Control** | Input | Instruction [31-26] | **101011** |
| 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] | **001000** |
| ALUOp | **00** |
| Output | Operation | **0010 (add)** |
| **Shift-left-2 (dùng cho lệnh beq)** | Input |  | **X** |
| Output |  | **X** |

\*Lệnh J label:

Label: exit

* Lệnh J-Type, dùng để nhảy đến lệnh nằm sau label, do đó không có tín hiệu nào từ CONTROL đến ALU, register,…

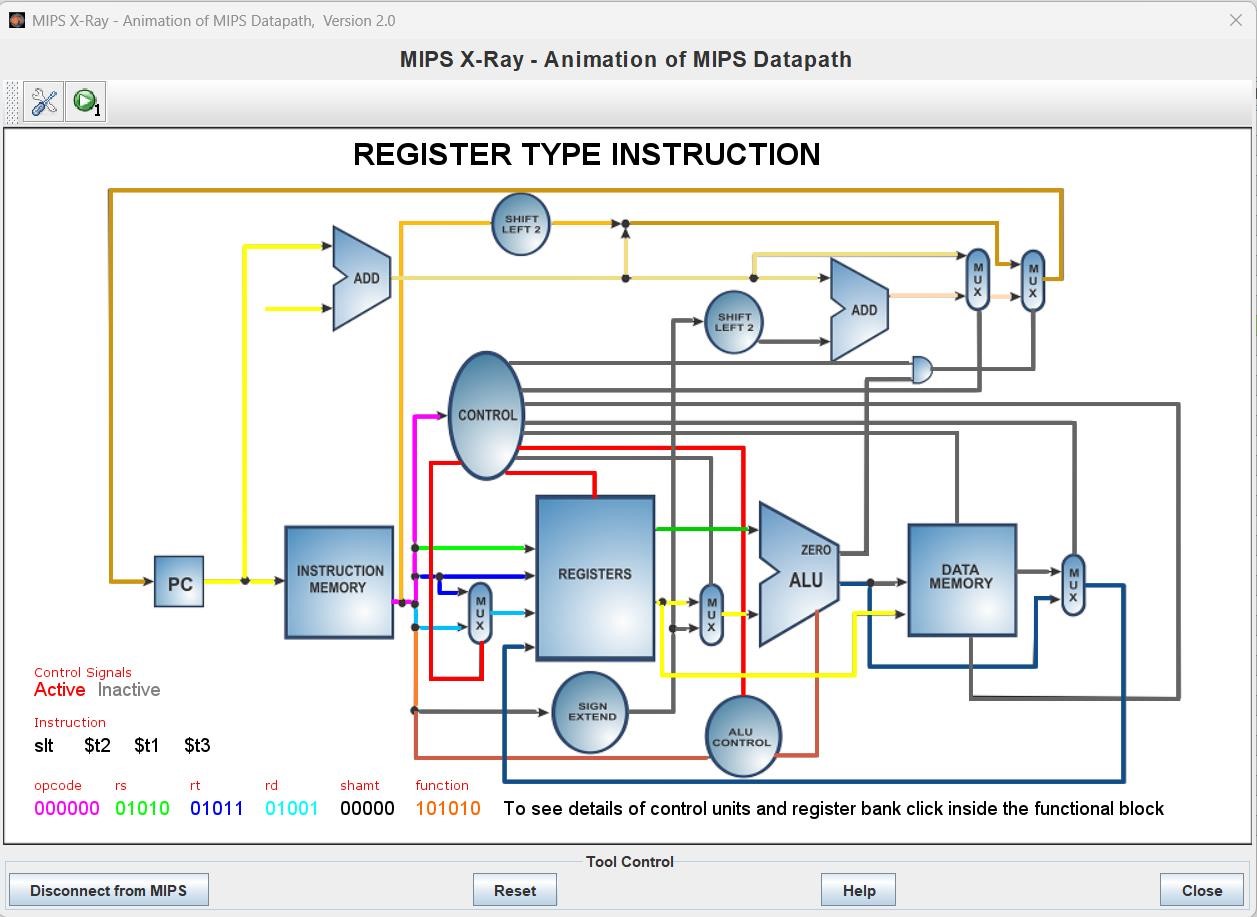


*Hình 12: Đường đi Datapath của lệnh j*

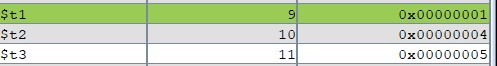
|  |  |  |  |
| --- | --- | --- | --- |
| *j exit* | | | |
| Tên khối |  | Ngõ | Giá trị |
| **Bộ Cộng** | Input |  | **0x00400014** |
| Output |  | **0x00400018 Lệnh slt $t1,$t2,$t3** |
| **Instruction**  **Memory** | Input | Read address | **0x00400014** |
| Output | Instruction[31-0] | **0x08100006** |
| **Registers** | Input | Read register 1 | **X** |
| Read register 2 | **X** |
| Write register | **X** |
| Write data | **X** |
| Output | Read data 1 | **X** |
| Read data 2 | **X** |
| **ALU** | Input | Input thứ nhất của ALU | **X** |
| Input thứ hai của ALU | **X** |
| Output | ALU result | **X** |
| Zero | **0** |
| **Data Memory** | Input | Address | **X** |
| Write data | **X** |
| Output | Read data | **X** |
| **Control** | Input | Instruction [31-26] | **000010** |
| Output | RegDst | **X** |
| Branch | **X** |
| MemRead | **0** |
| MemtoReg | **X** |
| ALUOp | **X** |
| MemWrite | **0** |
| ALUSrc | **X** |
| RegWrite | **0** |
| **Sign-extend** | Input |  | **X** |
| Output |  | **X** |
| **ALU Control** | Input | Instruction [5-0] | **X** |
| ALUOp | **X** |
| Output | Operation | **X** |
| **Shift-left-2 (dùng cho lệnh j)** | Input |  | **0x00100006** |
| Output |  | **0x00400018** |

\*Lệnh slt $t1, $t2, $t3:

* Lệnh R-Type, dùng để so sánh giá trị của 2 thanh ghi $t2 và $t3, sau đó lưu vào thanh $t1 giá trị 0 ($t2 >= $t3) hoặc 1 ($t2 < $t3)

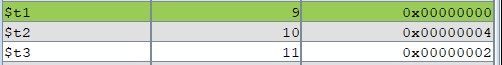


### Hình 13: Đường đi Datapath của lệnh slt



*Hình 14: Giá trị sau khi thực hiện lệnh ( 4 < 5 => Đúng thì kết quả =1)*

|  |  |  |  |
| --- | --- | --- | --- |
| *slt $t1, $t2, $t3* Khi kết quả đúng | | | |
| Tên khối |  | Ngõ | Giá trị |
| **Bộ Cộng** | Input |  | **0x00400018** |
| Output |  | **0x0040001C** |
| **Instruction**  **Memory** | Input | Read address | **0x00400018** |
| Output | Instruction[31-0] | **0x014B482A** |
| **Registers** | Input | Read register 1 | **01010** |
| Read register 2 | **01011** |
| Write register | **01001** |
| Write data | **0x00000001** |
| Output | Read data 1 | **0x00000004** |
| Read data 2 | **0x00000005** |
| **ALU** | Input | Input thứ nhất của ALU | **0x00000004** |
| Input thứ hai của ALU | **0x00000005** |
| Output | ALU result | **0xFFFFFFFF** |
| Zero | **0** |
| **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 |  | **X** |
| Output |  | **X** |
| **ALU Control** | Input | Instruction [5-0] | **101010** |
| ALUOp | **10** |
| Output | Operation | **0111 (set on less than)** |
| **Shift-left-2 (dùng cho lệnh beq)** | Input |  | **X** |
| Output |  | **X** |

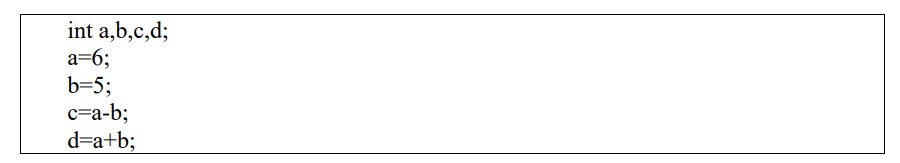


### Hình 15: Giá trị sau khi thực hiện lệnh ( 4 < 2 => Sai thì kết quả =0)

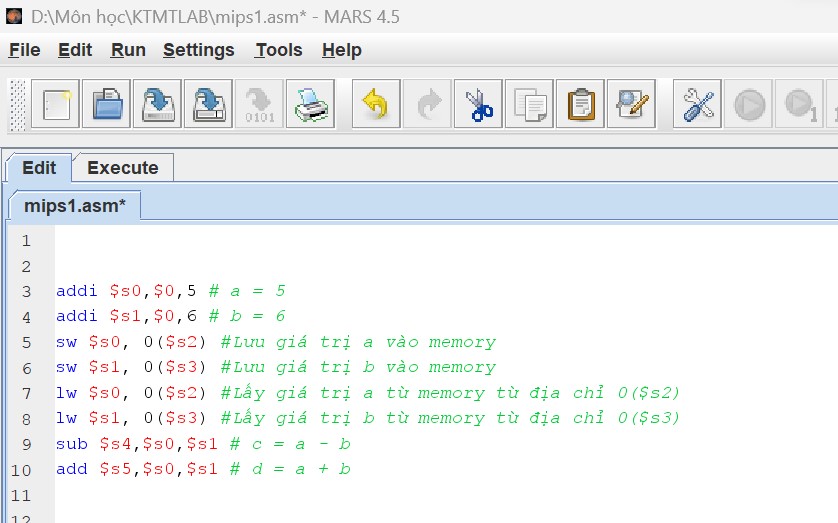
|  |  |  |  |
| --- | --- | --- | --- |
| *slt $t1, $t2, $t3* Khi kết quả sai | | | |
| Tên khối |  | Ngõ | Giá trị |
| **Bộ Cộng** | Input |  | **0x00400018** |
| Output |  | **0x0040001C** |
| **Instruction**  **Memory** | Input | Read address | **0x00400018** |
| Output | Instruction[31-0] | **0x014B482A** |
| **Registers** | Input | Read register 1 | **01010** |
| Read register 2 | **01011** |
| Write register | **01001** |
| Write data | **0x00000000** |
| Output | Read data 1 | **0x00000004** |
| Read data 2 | **0x00000002** |
| **ALU** | Input | Input thứ nhất của ALU | **0x00000004** |
| Input thứ hai của ALU | **0x00000002** |
| Output | ALU result | **0x00000002** |
| Zero | **0** |
| **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 |  | **X** |
| Output |  | **X** |
| **ALU Control** | Input | Instruction [5-0] | **101010** |
| ALUOp | **10** |
| Output | Operation | **0111 (set on less than)** |
| **Shift-left-2 (dùng cho lệnh beq)** | Input |  | **X** |
| Output |  | **X** |

## Section 4

**Task name 1: Chuyển chương trình sau sang MIPS:**

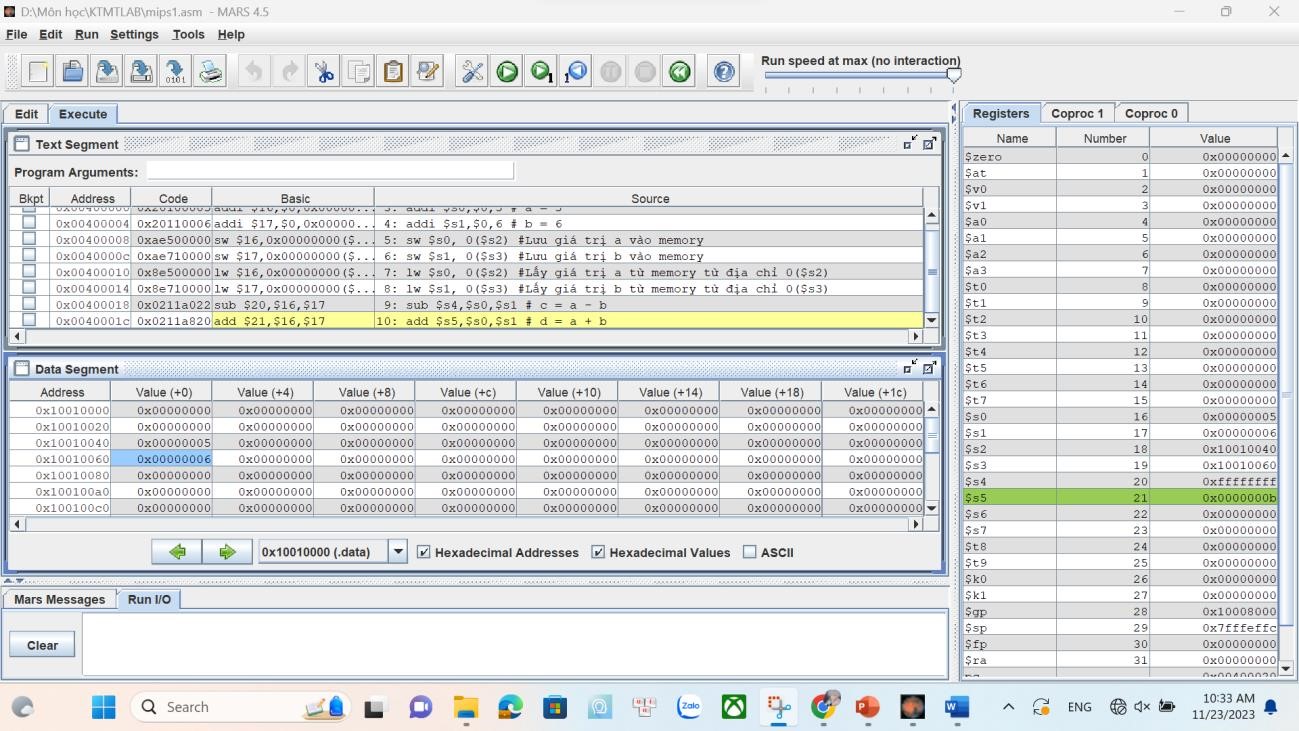


* Các biến được lưu trong memory.
* Xác định các lệnh tương ứng là loại lệnh nào (R-type, I-Type, J-Type)? Giải thích?
* Kết nối chương trình với MIPS X-Ray trong MARS. Chạy từng bước các lệnh và ở mỗi lệnh giải thích quá trình thực thi lệnh đó trên datapath trong MARS.



### Hình 1: Chuyển chương trình sang MIPS

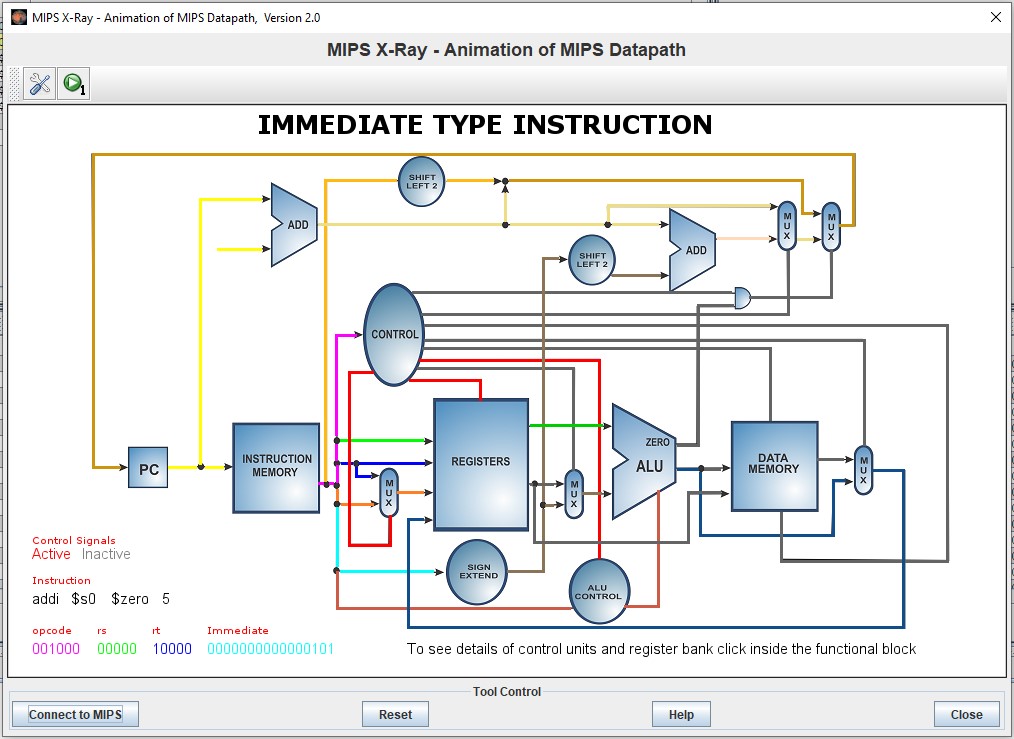
- Gán giá trị ban đầu: $s2 =0x10010040, $s3 = 0x10010060 để tạo các địa chỉ lưu các giá trị của các biến



### Hình 2: Thực thi chương trình

* Lệnh: addi $s0,$0,5 :

+ Loại lệnh I do opcode = 001000 =0x8

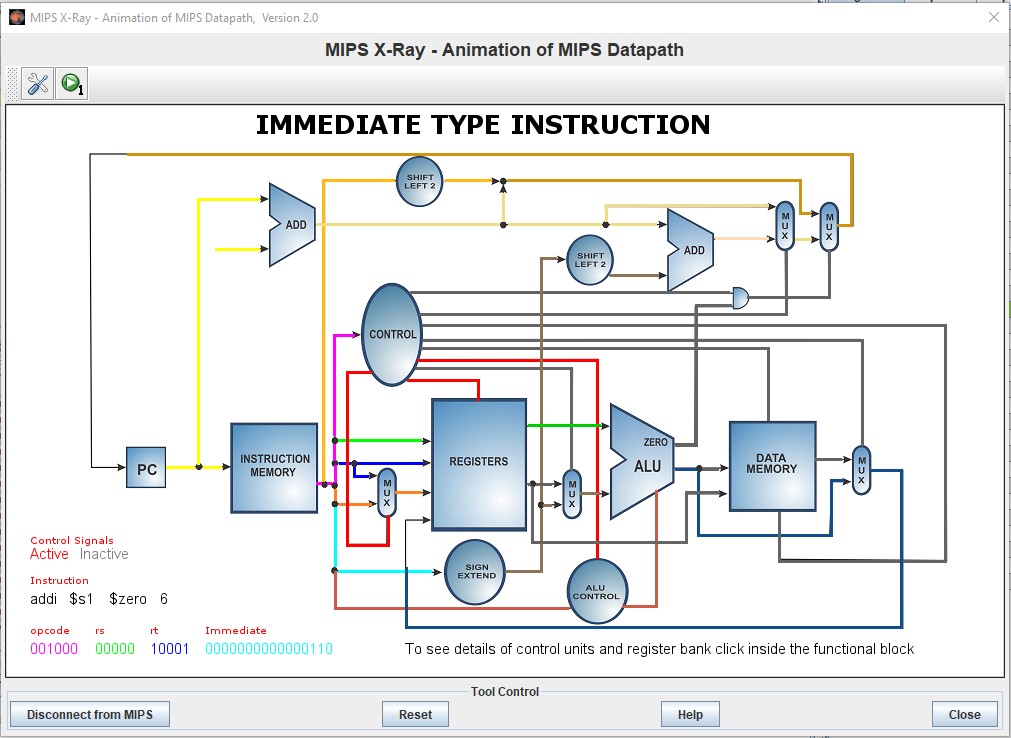


*Hình 3: Datapth của lệnh addi $s0,$0,5*

|  |  |  |  |
| --- | --- | --- | --- |
| *addi $s0,$0,5* |  |  |  |
| Tên khối |  | Ngõ | Giá trị |
| **Bộ Cộng** | Input |  | **0x00400000** |
| Output |  | **0x00400004 lệnh addi $s1,$0,6** |
| **Instruction**  **Memory** | Input | Read address | **0x00400000** |
| Output | Instruction[31-0] | **0x20100005** |
| **Registers** | Input | Read register 1 | **00000** |
| Read register 2 | **X** |
| Write register | **10000** |
| Write data | **00005** |
| 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 | **0x00000000** |
| Write data | **0x00000000** |
|  | Output | Read data | **X** |
| **Control** | Input | Instruction [31-26] | **001000** |
| Output | RegDst | **0** |
| Branch | **0** |
| MemRead | **0** |
| MemtoReg | **0** |
| ALUOp | **11** |
| MemWrite | **0** |
| ALUSrc | **1** |
| RegWrite | **1** |
| **Sign-extend** | Input |  | **0x0005** |
| Output |  | **0x00000005** |
| **ALU Control** | Input | Instruction [5-0] | **000101** |
| ALUOp | **11** |
| Output | Operation | **0010 (add)** |
| **Shift-left-2 (dùng cho lệnh beq)** | Input |  | **0x00000000** |
| Output |  | **0x00000000** |
|  |  |  |  |

* Lệnh addi $s1,$0,6

+ Loại lệnh I do opcode = 001000 =0x8

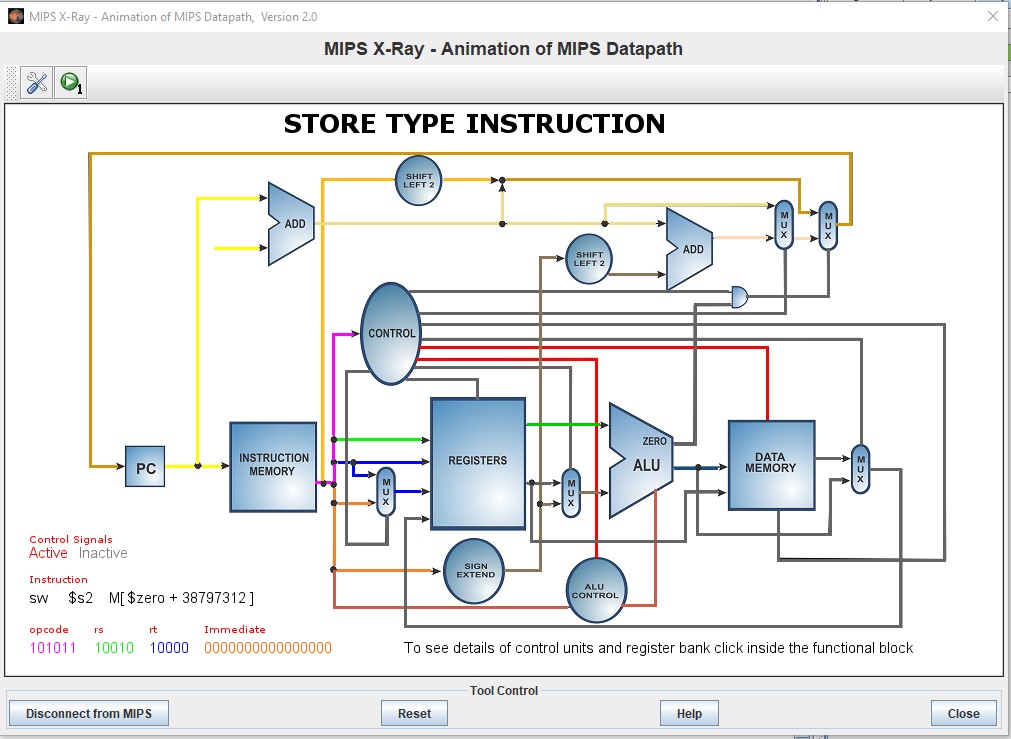


*Hình 4: Datapath của lệnh addi $s1,0,6*

|  |  |  |  |
| --- | --- | --- | --- |
| *addi $s1,$0,6* |  |  |  |
| Tên khối |  | Ngõ | Giá trị |
| **Bộ Cộng** | Input |  | **0x00400004** |
| Output |  | **0x00400008 lệnh sw $s0, 0($s2)** |
| **Instruction**  **Memory** | Input | Read address | **0x00400004** |
| Output | Instruction[31-0] | **0x20110006** |
| **Registers** | Input | Read register 1 | **00000** |
| Read register 2 | **X** |
| Write register | **10001** |
| Write data | **00006** |
| Output | Read data 1 | **0x00000000** |
| Read data 2 | **0x00000000** |
| **ALU** | Input | Input thứ nhất của ALU | **0x00000000** |
| Input thứ hai của ALU | **0x00000006** |
| Output | ALU result | **0x00000006** |
| Zero | **0** |
| **Data Memory** | Input | Address | **0x00000000** |
| Write data | **0x00000000** |
|  | Output | Read data | **X** |
| **Control** | Input | Instruction [31-26] | **001000** |
| Output | RegDst | **0** |
| Branch | **0** |
| MemRead | **0** |
| MemtoReg | **0** |
| ALUOp | **11** |
| MemWrite | **0** |
| ALUSrc | **1** |
| RegWrite | **1** |
| **Sign-extend** | Input |  | **0x0006** |
| Output |  | **0x00000006** |
| **ALU Control** | Input | Instruction [5-0] | **000110** |
| ALUOp | **11** |
| Output | Operation | **0010 (add)** |
| **Shift-left-2 (dùng cho lệnh beq)** | Input |  | **0x00000000** |
| Output |  | **0x00000000** |
|  |  |  |  |

* Lệnh sw $s0,0($s2):

+ Loại lệnh I do opcode = 101011 = 0x2b

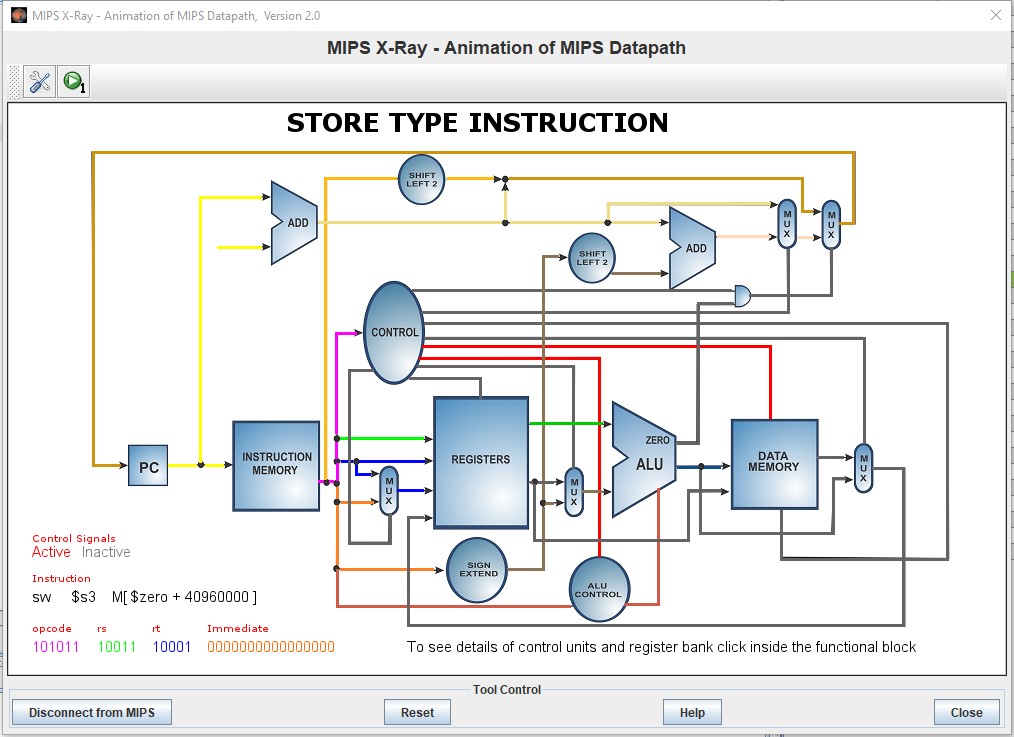


*Hình 5: Datapath của lệnh sw $s0, 0($s2)*

|  |  |  |  |
| --- | --- | --- | --- |
| *sw $s0, 0($s2)* |  |  |  |
| Tên khối |  | Ngõ | Giá trị |
| **Bộ Cộng** | Input |  | **0x00400008** |
| Output |  | **0x0040000c lệnh sw $s1, 0($s3)** |
| **Instruction**  **Memory** | Input | Read address | **0x00400008** |
| Output | Instruction[31-0] | **0xAE50000** |
| **Registers** | Input | Read register 1 | **10010** |
| Read register 2 | **10000** |
| Write register | **X** |
| Write data | **X** |
| Output | Read data 1 | **0x10010040** |
| Read data 2 | **0x00000005** |
| **ALU** | Input | Input thứ nhất của ALU | **0x10010040** |
| Input thứ hai của ALU | **0x00000000** |
| Output | ALU result | **0x10010040** |
| Zero | **0** |
| **Data Memory** | Input | Address | **0x10010040** |
| Write data | **0x00000005** |
|  | Output | Read data | **X** |
| **Control** | Input | Instruction [31-26] | **101011** |
| Output | RegDst | **X** |
| Branch | **0** |
| MemRead | **0** |
| MemtoReg | **X** |
| ALUOp | **00** |
| MemWrite | **1** |
| ALUSrc | **1** |
| RegWrite | **0** |
| **Sign-extend** | Input |  | **0x0000** |
| Output |  | **0x00000000** |
| **ALU Control** | Input | Instruction [5-0] | **000000** |
| ALUOp | **00** |
| Output | Operation | **0010 (add)** |
| **Shift-left-2 (dùng cho lệnh beq)** | Input |  | **0x00000000** |
| Output |  | **0x00000000** |
|  |  |  |  |

* Lệnh sw $s1, 0($s3)

+ Loại lệnh I do opcode = 101011 = 0x2b

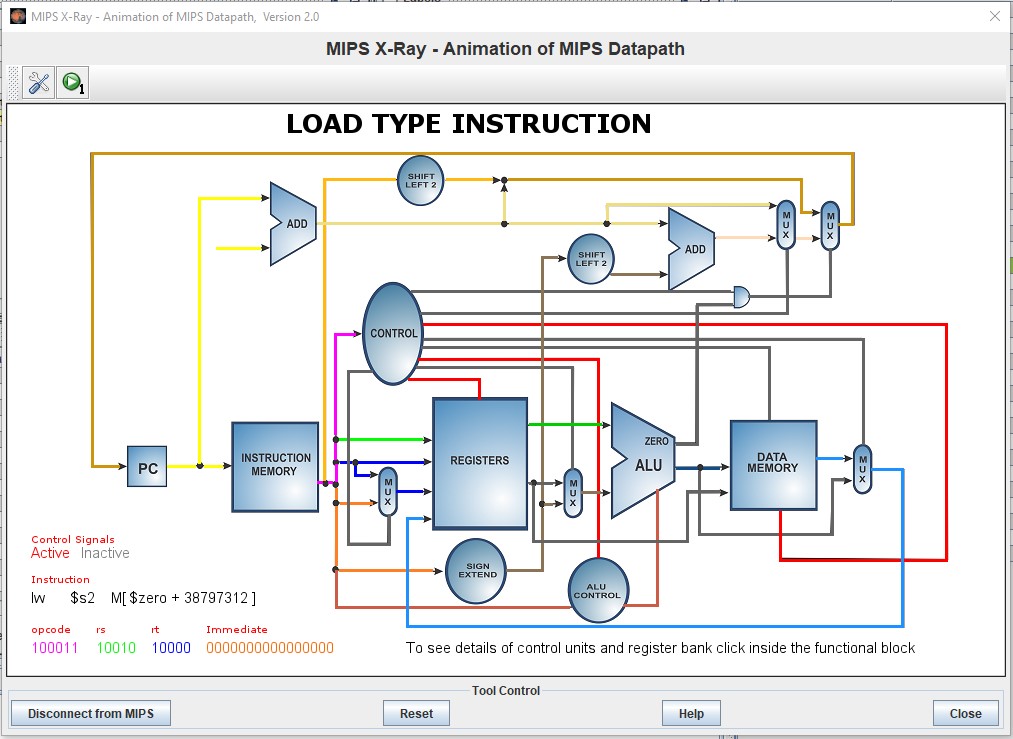


*Hình 6: Datapath của lệnh sw $s1, 0($s3)*

|  |  |  |  |
| --- | --- | --- | --- |
| *sw $s1, 0($s3)* |  | | |
| Tên khối |  | Ngõ | Giá trị |
| **Bộ Cộng** | Input |  | **0x0040000c** |
| Output |  | **0x00400010 lệnh lw $s0, 0($s2)** |
| **Instruction**  **Memory** | Input | Read address | **0x0040000c** |
| Output | Instruction[31-0] | **0xAE710000** |
| **Registers** | Input | Read register 1 | **10011** |
| Read register 2 | **10001** |
| Write register | **X** |
| Write data | **X** |
| Output | Read data 1 | **0x10010060** |
| Read data 2 | **0x00000006** |
| **ALU** | Input | Input thứ nhất của ALU | **0x10010060** |
| Input thứ hai của ALU | **0x00000000** |
| Output | ALU result | **0x10010060** |
| Zero | **0** |
| **Data Memory** | Input | Address | **0x10010060** |
| Write data | **0x00000006** |
| Output | Read data | **X** |
| **Control** | Input | Instruction [31-26] | **101011** |
| Output | RegDst | **X** |
| Branch | **0** |
| MemRead | **0** |
| MemtoReg | **X** |
| ALUOp | **00** |
| MemWrite | **1** |
| ALUSrc | **1** |
| RegWrite | **0** |
| **Sign-extend** | Input |  | **0x0000** |
| Output |  | **0x00000000** |
| **ALU Control** | Input | Instruction [5-0] | **000000** |
| ALUOp | **00** |
| Output | Operation | **0010 (add)** |
| **Shift-left-2 (dùng cho lệnh beq)** | Input |  | **0x00000000** |
| Output |  | **0x00000000** |
|  |  |  |  |

* Lệnh lw $s0, 0($s2):

+ Loại lệnh I do opcode = 100011 = 0x23

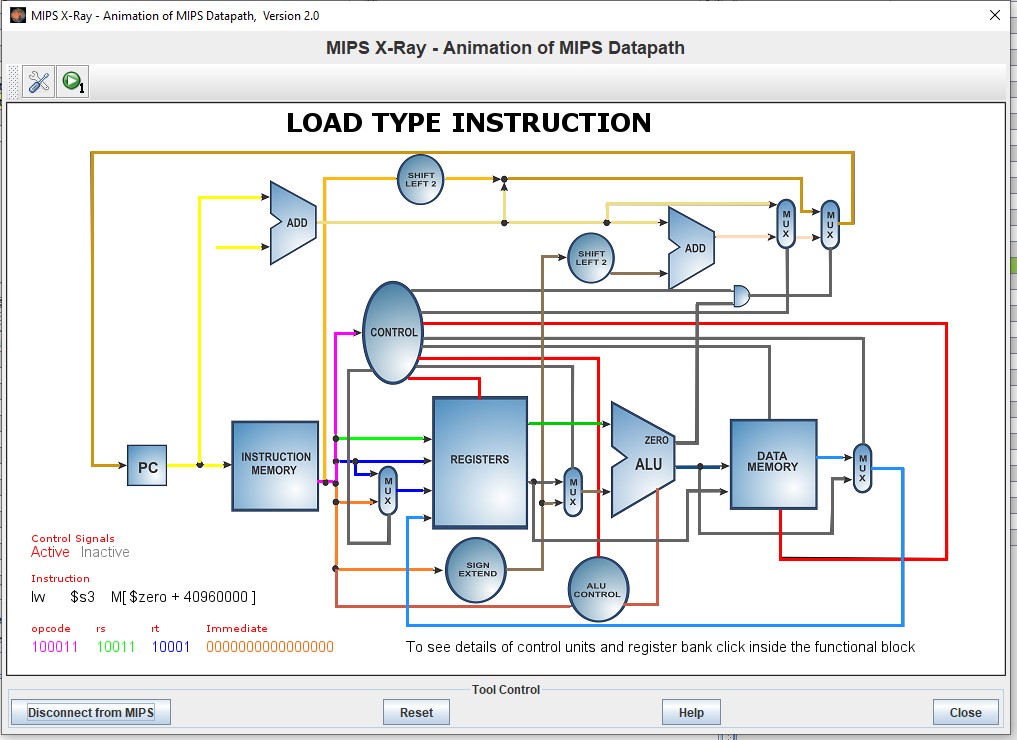


*Hình 7 : Data path của lệnh lw $s0, 0($s2)*

|  |  |  |  |
| --- | --- | --- | --- |
| *lw $s0, 0($s2)* |  | | |
| Tên khối |  | Ngõ | Giá trị |
| **Bộ Cộng** | Input |  | **0x00400010** |
| Output |  | **0x00400014 lệnh lw $s1, 0($s3)** |
| **Instruction**  **Memory** | Input | Read address | **0x00400010** |
| Output | Instruction[31-0] | **0x8E500000** |
| **Registers** | Input | Read register 1 | **10010** |
| Read register 2 | **X** |
| Write register | **10000** |
| Write data | **00101** |
| Output | Read data 1 | **0x10010040** |
| Read data 2 | **0x00000000** |
| **ALU** | Input | Input thứ nhất của ALU | **0x10010040** |
| Input thứ hai của ALU | **0x00000000** |
| Output | ALU result | **0x10010040** |
| Zero | **0** |
| **Data Memory** | Input | Address | **0x10010040** |
| Write data | **X** |
| Output | Read data | **0x00000005** |
| **Control** | Input | Instruction [31-26] | **100011** |
| Output | RegDst | **X** |
| Branch | **0** |
| MemRead | **1** |
| MemtoReg | **1** |
| ALUOp | **00** |
| MemWrite | **0** |
| ALUSrc | **1** |
| RegWrite | **1** |
| **Sign-extend** | Input |  | **0x0000** |
| Output |  | **0x00000000** |
| **ALU Control** | Input | Instruction [5-0] | **000000** |
| ALUOp | **00** |
| Output | Operation | **0010 (add)** |
| **Shift-left-2 (dùng cho lệnh beq)** | Input |  | **0x00000000** |
| Output |  | **0x00000000** |
|  |  |  |  |

* Lệnh lw $s1, 0($s3):

+ Loại lệnh I do opcode = 100011 = 0x23

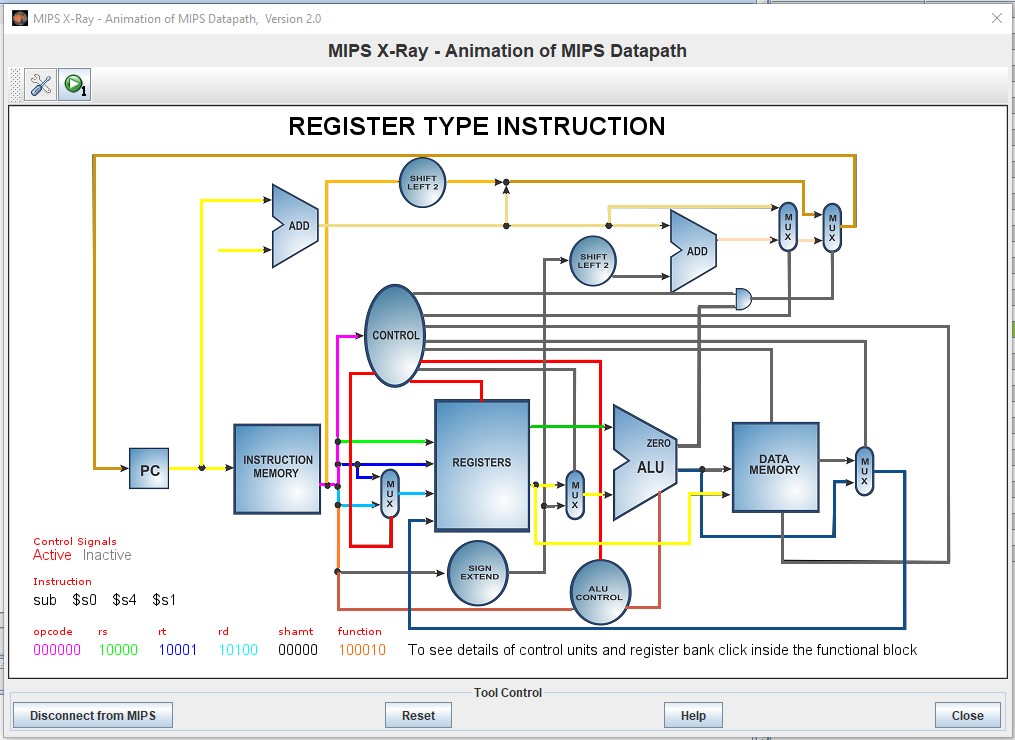


*Hình 8 : Data path của lệnh lw $s1, 0($s3)*

|  |  |  |  |
| --- | --- | --- | --- |
| *lw $s1, 0($s3)* |  | | |
| Tên khối |  | Ngõ | Giá trị |
| **Bộ Cộng** | Input |  | **0x00400014** |
| Output |  | **0x00400018 lệnh sub $s4, $s0,$s1** |
| **Instruction**  **Memory** | Input | Read address | **0x00400014** |
| Output | Instruction[31-0] | **0x8E710000** |
| **Registers** | Input | Read register 1 | **10011** |
| Read register 2 | **X** |
| Write register | **10001** |
| Write data | **00110** |
| Output | Read data 1 | **0x10010060** |
| Read data 2 | **0x00000000** |
| **ALU** | Input | Input thứ nhất của ALU | **0x10010060** |
| Input thứ hai của ALU | **0x00000000** |
| Output | ALU result | **0x10010060** |
| Zero | **0** |
| **Data Memory** | Input | Address | **0x10010060** |
| Write data | **X** |
| Output | Read data | **0x00000006** |
| **Control** | Input | Instruction [31-26] | **100011** |
| Output | RegDst | **X** |
| Branch | **0** |
| MemRead | **1** |
| MemtoReg | **1** |
| ALUOp | **00** |
| MemWrite | **0** |
| ALUSrc | **1** |
| RegWrite | **1** |
| **Sign-extend** | Input |  | **0x0000** |
| Output |  | **0x00000000** |
| **ALU Control** | Input | Instruction [5-0] | **000000** |
| ALUOp | **00** |
| Output | Operation | **0010 (add)** |
| **Shift-left-2 (dùng cho lệnh beq)** | Input |  | **0x00000000** |
| Output |  | **0x00000000** |
|  |  |  |  |

* Lệnh sub $s4,$s0,$s1:

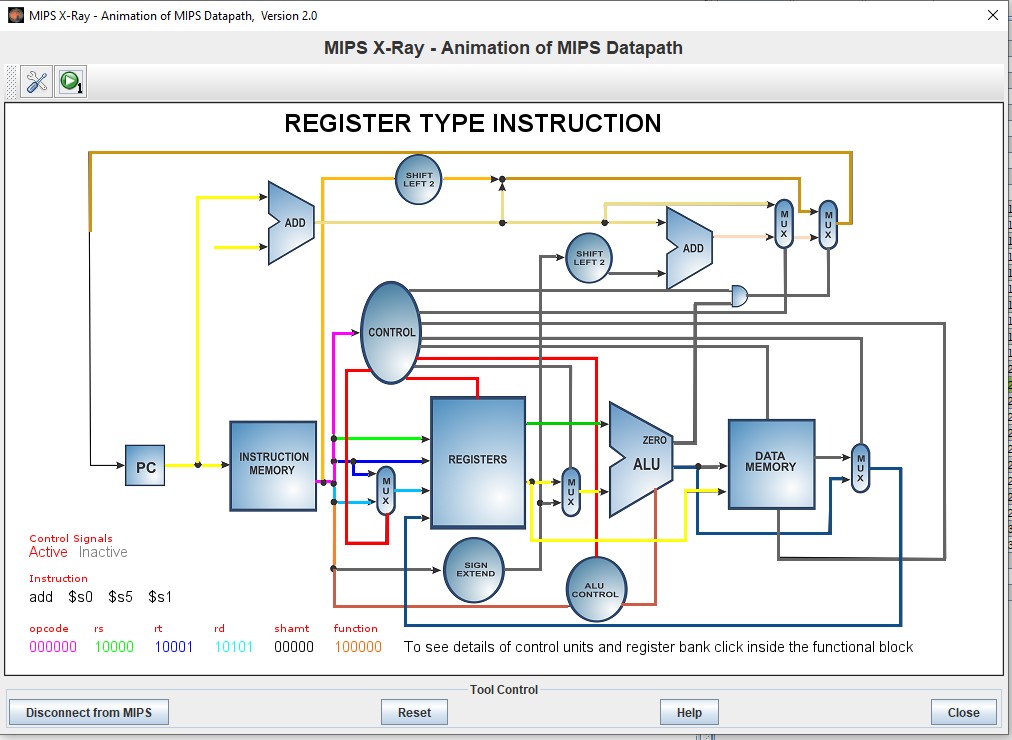
+ Loại lệnh R do opcode = 000000 = 0x0 và funct = 100010 = 0x22



### Hình 9: Datapath của lệnh sub $s4, $s0,$s1

|  |  |  |  |
| --- | --- | --- | --- |
| *sub $s4,$s0,$s1* |  |  |  |
| Tên khối |  | Ngõ | Giá trị |
| **Bộ Cộng** | Input |  | **0x00400018** |
| Output |  | **0x0040001c lệnh add $s5, $s0,$s1** |
| **Instruction**  **Memory** | Input | Read address | **0x00400018** |
| Output | Instruction[31-0] | **0x0211A022** |
| **Registers** | Input | Read register 1 | **10000** |
| Read register 2 | **10001** |
| Write register | **10100** |
| Write data | **0xffffffff** |
| Output | Read data 1 | **0x00000005** |
| Read data 2 | **0x00000006** |
| **ALU** | Input | Input thứ nhất của ALU | **0x00000005** |
| Input thứ hai của ALU | **0x00000006** |
| Output | ALU result | **0xffffffff** |
| Zero | **0** |
| **Data Memory** | Input | Address | **0x00000000** |
| Write data | **0x00000000** |
| 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 |  | **0x0000** |
| Output |  | **0x00000000** |
| **ALU Control** | Input | Instruction [5-0] | **100010** |
| ALUOp | **10** |
| Output | Operation | **0110 (subtract)** |
| **Shift-left-2 (dùng cho lệnh beq)** | Input |  | **0x00000000** |
| Output |  | **0x00000000** |
|  |  |  |  |

* Lệnh add $s5,$s0,$s1:

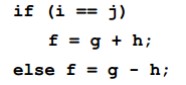


*Hình 10: Datapath của lệnh add $s5, $s0,$s1*

+ Loại lệnh R do opcode = 000000 = 0x0 và funct = 100000 = 0x20

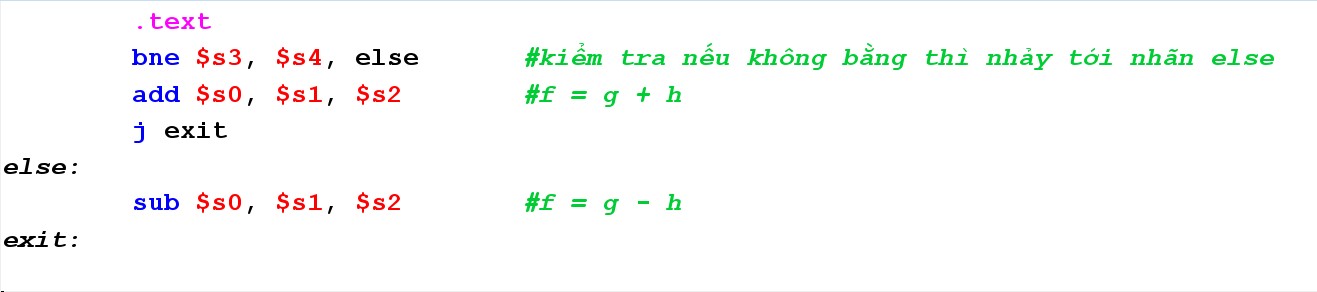
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *add $s5,$s0,$s1* | | | |  |
| Tên khối |  | Ngõ |  | Giá trị |
| **Bộ Cộng** | Input |  | **0x0040001c** |  |
| Output |  | **0x00400020** |  |
| **Instruction**  **Memory** | Input | Read address | **0x0040001c** |  |
| Output | Instruction[31-0] | **0x0211A820** |  |
| **Registers** | Input | Read register 1 | **10000** |  |
| Read register 2 | **10001** |  |
| Write register | **10101** |  |
| Write data | **0x0000000b** |  |
| Output | Read data 1 | **0x00000005** |  |
| Read data 2 | **0x00000006** |  |
| **ALU** | Input | Input thứ nhất của ALU | **0x00000005** |  |
| Input thứ hai của ALU | **0x00000006** |  |
| Output | ALU result | **0x0000000b** |  |
| Zero | **0** |  |
| **Data Memory** | Input | Address | **0x00000000** | |
| Write data | **0x00000000** | |
| 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 |  | **0x0000** | |
| Output |  | **0x00000000** | |
| **ALU Control** | Input | Instruction [5-0] | **100000** | |
| ALUOp | **10** | |
| Output | Operation | **0010 (add)** | |
| **Shift-left-2 (dùng cho lệnh beq)** | Input |  | **0x00000000** | |
| Output |  | **0x00000000** | |
|  |  |  |  | |

**Task name 2: Chuyển chương trình sau sang MIPS:**



* + I được lưu trong $s3, j trong $s4, f trong $s0, g trong $s1, h trong $s2.
  + Phải sử dụng lệnh bne và j trong chương trình.
  + Kết nối chương trình với MIPS X-Ray trong MARS. Chạy từng bước các lệnh và ở mỗi lệnh giải thích quá trình thực thi lệnh đó trên datapath trong MARS.

* Thực hiện chuyển đoạn mã C sang MIPS:

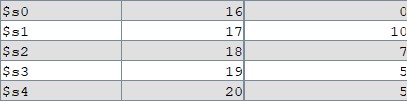


### Hình 1: Đoạn mã MIPS chuyển từ mã C theo đề bài

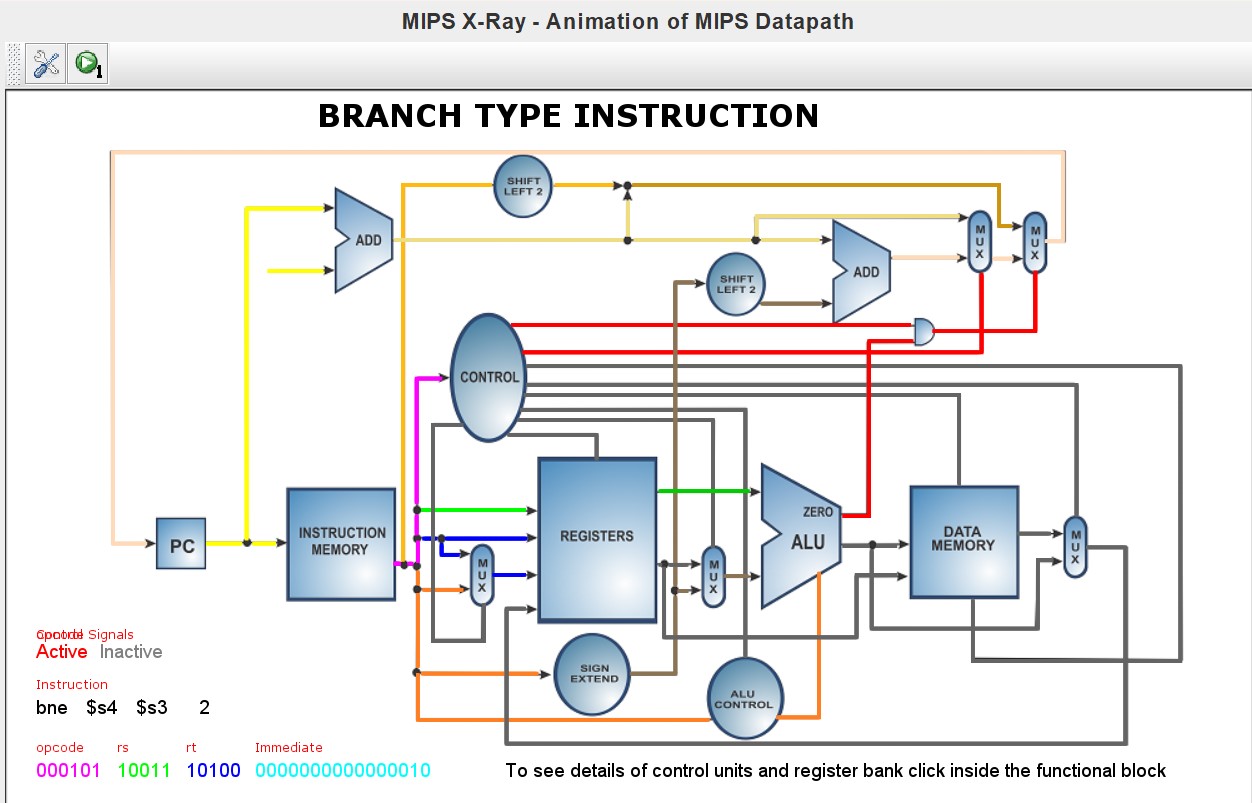
 Kiểm tra xem *$s3* có bằng *$s4* không. Nếu bằng, thực hiện cộng giá trị ở hai thanh ghi *$s1* và *$s2.* Ngược lại, thực hiện trừ giá trị ở thanh ghi *$s1* cho giá trị ở thanh ghi *$s2*. Kết quả được lưu vào thanh ghi *$s0*.

 Trạng thái của các thành phần trong bộ xử lý khi thực hiện lần lượt các lệnh trên:

\*Trường hợp 1: Giá trị ở hai thanh ghi *$s3* và *$s4* bằng nhau



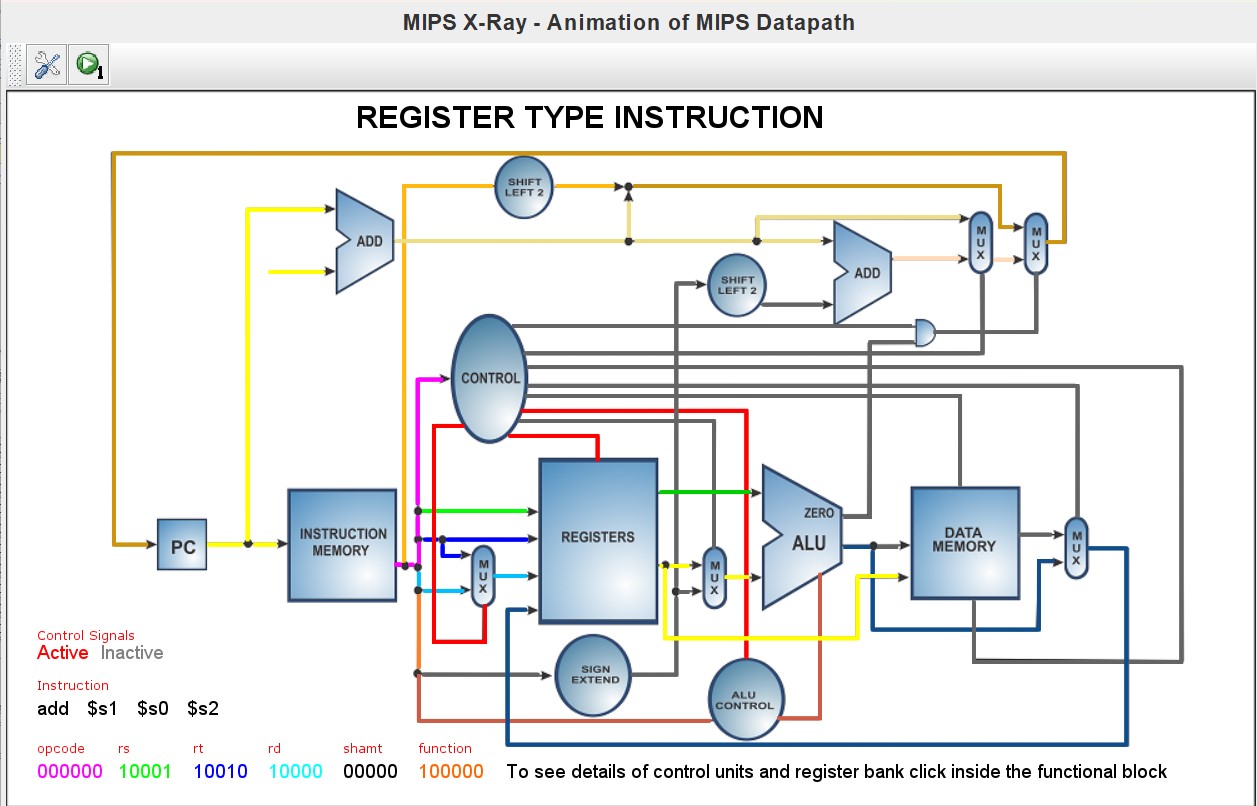
*Hình 2: Giá trị của các thanh ghi trong trường hợp 1* - Lệnh *bne $s3, $s4, else*:



### Hình 3: Datapath của lệnh bne $s3, $s4, else

|  |  |  |  |
| --- | --- | --- | --- |
| *bne $s3, $s4, else* | | | |
| Tên khối |  | Ngõ | Giá trị |
| **Bộ Cộng** | Input |  | **0x00400000** |
| Output |  | **0x00400004**  **(lệnh add $s0, $s1, $s2)** |
| **Instruction**  **Memory** | Input | Read address | **0x00400000** |
| Output | Instruction[31-0] | **0x16740002** |
| **Registers** | Input | Read register 1 | **10011** |
| Read register 2 | **10100** |
| Write register | **X** |
| Write data | **X** |
| Output | Read data 1 | **0x00000005** |
| Read data 2 | **0x00000005** |
| **ALU** | Input | Input thứ nhất của ALU | **0x00000005** |
| Input thứ hai của ALU | **0x00000005** |
| Output | ALU result | **X** |
| Zero | **0** |
| **Data**  **Memory** | Input | Address | **X** |
| Write data | **X** |
| Output | Read data | **X** |
| **Control** | Input | Instruction [31-26] | **000101** |
| Output | Jump | **1** |
| RegDst | **X** |
| Branch | **1** |
| MemRead | **0** |
| MemtoReg | **X** |
| ALUOp | **01** |
| MemWrite | **0** |
| ALUSrc | **0** |
| RegWrite | **0** |
| **Sign-extend** | Input |  | **0x0002** |
| Output |  | **0x00000002** |
| **ALU**  **Control** | Input | Instruction [5-0] | **X** |
| ALUOp | **01** |
| Output | Operation | **0110 (sub)** |
| **Shift-left-2**  **(dùng cho lệnh beq hoặc bne)** | Input |  | **0x00000002** |
| Output |  | **0x00000008** |

- Lệnh *add $s0, $s1, $s2*:

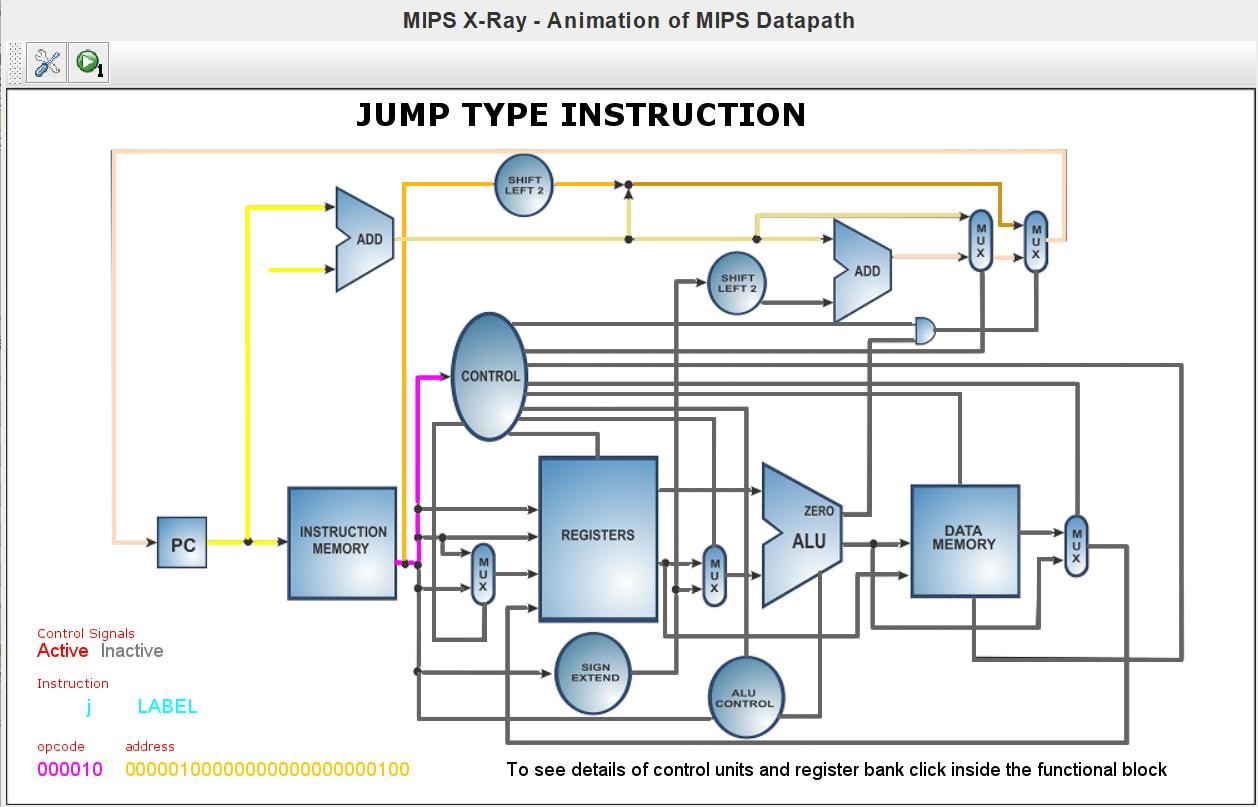


### Hình 4: Datapath của lệnh add $s0, $s1, $s2

|  |  |  |  |
| --- | --- | --- | --- |
| *add $s0, $s1, $s2* | | | |
| Tên khối |  | Ngõ | Giá trị |
| **Bộ Cộng** | Input |  | **0x00400004** |
| Output |  | **0x00400008**  **(lệnh j exit)** |
| **Instruction**  **Memory** | Input | Read address | **0x00400004** |
| Output | Instruction[31-0] | **0x02328020** |
| **Registers** | Input | Read register 1 | **10001** |
| Read register 2 | **10010** |
| Write register | **10000** |
| Write data | **0x00000011** |
| Output | Read data 1 | **0x0000000A** |
| Read data 2 | **0x00000007** |
| **ALU** | Input | Input thứ nhất của ALU | **0x0000000A** |
| Input thứ hai của ALU | **0x00000007** |
| Output | ALU result | **0x00000011** |
| Zero | **0** |
| **Data**  **Memory** | Input | Address | **X** |
| Write data | **X** |
| Output | Read data | **X** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Control** | Input | Instruction [31-26] | **000000** |
| Output | Jump | **0** |
| RegDst | **1** |
| Branch | **0** |
| MemRead | **0** |
| MemtoReg | **0** |
| ALUOp | **10** |
| MemWrite | **0** |
| ALUSrc | **0** |
| RegWrite | **1** |
| **Sign-extend** | Input |  | **X** |
| Output |  | **X** |
| **ALU**  **Control** | Input | Instruction [5-0] | **100000** |
| ALUOp | **10** |
| Output | Operation | **0010 (add)** |
| **Shift-left-2**  **(dùng cho lệnh beq hoặc bne)** | Input |  | **X** |
| Output |  | **X** |

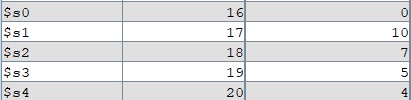
- Lệnh *j exit*:



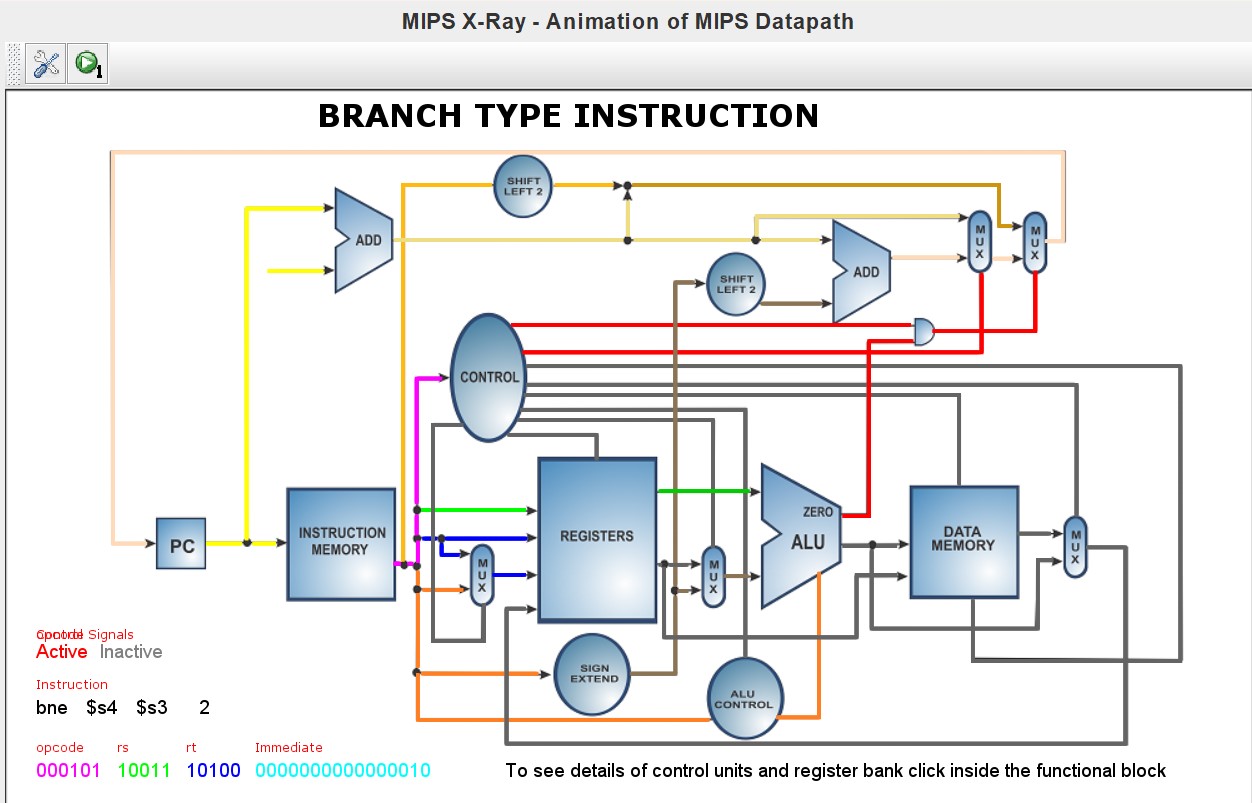
### Hình 5: Datapath của lệnh j exit

|  |  |  |  |
| --- | --- | --- | --- |
| *j exit* | | | |
| Tên khối |  | Ngõ | Giá trị |
| **Bộ Cộng** | Input |  | **0x00400008** |
| Output |  | **0x0040000C**  **(lệnh sub $s0, $s1, $s2)** |
| **Instruction**  **Memory** | Input | Read address | **0x00400008** |
| Output | Instruction[31-0] | **0x08100004** |
| **Registers** | Input | Read register 1 | **X** |
| Read register 2 | **X** |
| Write register | **X** |
| Write data | **X** |
| Output | Read data 1 | **X** |
| Read data 2 | **X** |
| **ALU** | Input | Input thứ nhất của ALU | **X** |
| Input thứ hai của ALU | **X** |
| Output | ALU result | **X** |
| Zero | **0** |
| **Data**  **Memory** | Input | Address | **X** |
| Write data | **X** |
| Output | Read data | **X** |
| **Control** | Input | Instruction [31-26] | **000010** |
| Output | Jump | **1** |
| RegDst | **0** |
| Branch | **0** |
| MemRead | **0** |
| MemtoReg | **0** |
| ALUOp | **X** |
| MemWrite | **0** |
| ALUSrc | **0** |
| RegWrite | **0** |
| **Sign-extend** | Input |  | **X** |
| Output |  | **X** |
| **ALU**  **Control** | Input | Instruction [5-0] | **X** |
| ALUOp | **X** |
| Output | Operation | **X** |
| **Shift-left-2**  **(dùng cho lệnh j)** | Input |  | **0x00100004** |
| Output |  | **0x00400010** |

\*Trường hợp 2: Giá trị ở hai thanh ghi *$s3* và *$s4* khác nhau



*Hình 6: Giá trị của các thanh ghi trong trường hợp 2* - Lệnh *bne $s3, $s4, else*:

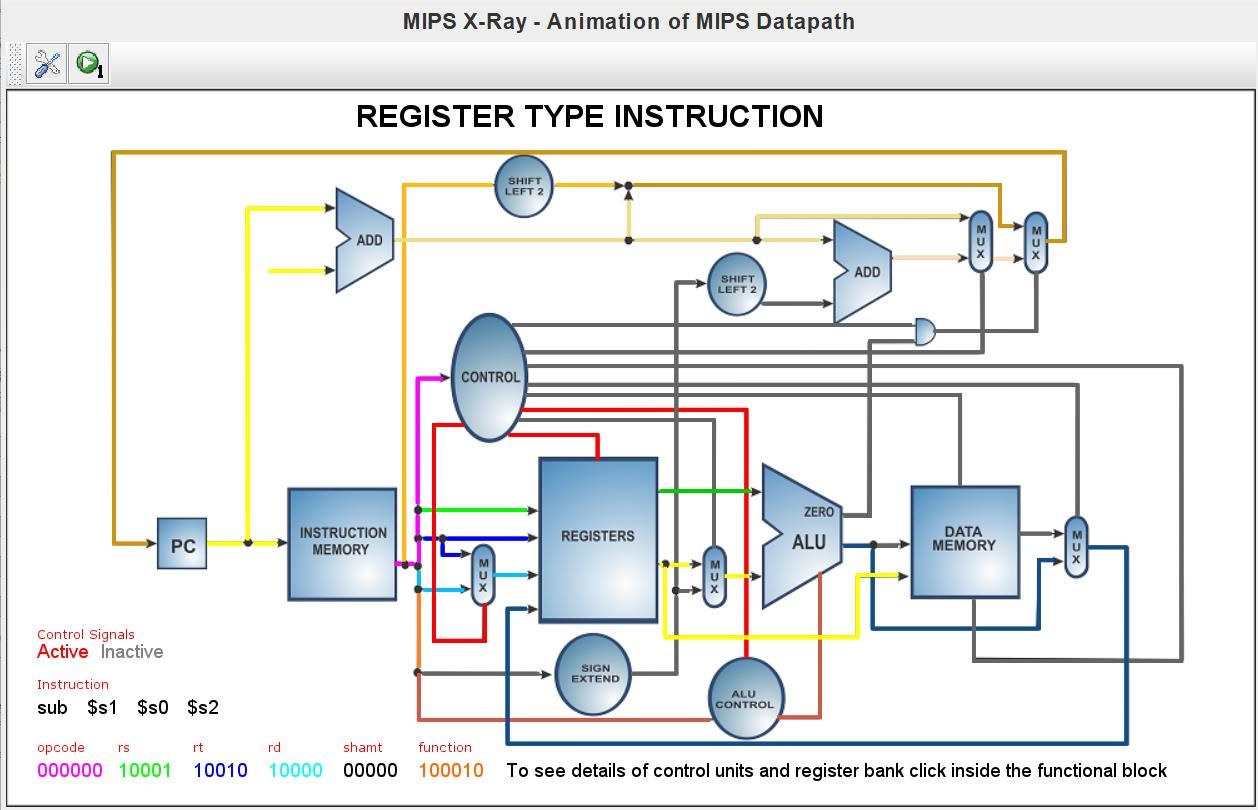


### Hình 7: Datapath của lệnh bne $s3, $s4, else

|  |  |  |  |
| --- | --- | --- | --- |
| *bne $s3, $s4, else* | |  |  |
| Tên khối |  | Ngõ | Giá trị |
| **Bộ Cộng** | Input | Input 1 | **0x00400004** |
| Input 2 | **0x00000008** |
| Output |  | **0x0040000C**  **(lệnh sub $s0, $s1, $s2)** |
| **Instruction**  **Memory** | Input | Read address | **0x00400000** |
| Output | Instruction[31-0] | **0x16740002** |
| **Registers** | Input | Read register 1 | **10011** |
| Read register 2 | **10100** |
| Write register | **X** |
| Write data | **X** |
| Output | Read data 1 | **0x00000005** |
| Read data 2 | **0x00000004** |

|  |  |  |  |
| --- | --- | --- | --- |
| **ALU** | Input | Input thứ nhất của ALU | **0x00000005** |
| Input thứ hai của ALU | **0x00000004** |
| Output | ALU result | **X** |
| Zero | **1** |
| **Data**  **Memory** | Input | Address | **X** |
| Write data | **X** |
| Output | Read data | **X** |
| **Control** | Input | Instruction [31-26] | **000101** |
| Output | Jump | **1** |
| RegDst | **X** |
| Branch | **1** |
| MemRead | **0** |
| MemtoReg | **X** |
| ALUOp | **01** |
| MemWrite | **0** |
| ALUSrc | **0** |
| RegWrite | **0** |
| **Sign-extend** | Input |  | **0x0002** |
| Output |  | **0x00000002** |
| **ALU**  **Control** | Input | Instruction [5-0] | **X** |
| ALUOp | **01** |
| Output | Operation | **0110 (sub)** |
| **Shift-left-2**  **(dùng cho lệnh beq hoặc bne)** | Input |  | **0x00000002** |
| Output |  | **0x00000008** |

- Lệnh *sub $s0, $s1, $s2*:



### Hình 8: Datapath của lệnh sub $s0, $s1, $s2

|  |  |  |  |
| --- | --- | --- | --- |
| *sub $s0, $s1, $s2* | | | |
| Tên khối |  | Ngõ | Giá trị |
| **Bộ Cộng** | Input |  | **0x0040000C** |
| Output |  | **0x00400010**  **(nhãn exit)** |
| **Instruction**  **Memory** | Input | Read address | **0x0040000C** |
| Output | Instruction[31-0] | **0x02328022** |
| **Registers** | Input | Read register 1 | **10001** |
| Read register 2 | **10010** |
| Write register | **10000** |
| Write data | **0x00000003** |
| Output | Read data 1 | **0x0000000A** |
| Read data 2 | **0x00000007** |
| **ALU** | Input | Input thứ nhất của ALU | **0x0000000A** |
| Input thứ hai của ALU | **0x00000007** |
| Output | ALU result | **0x00000003** |
| Zero | **0** |
| **Data**  **Memory** | Input | Address | **X** |
| Write data | **X** |
| Output | Read data | **X** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Control** | Input | Instruction [31-26] | **000000** |
| Output | Jump | **0** |
| RegDst | **1** |
| Branch | **0** |
| MemRead | **0** |
| MemtoReg | **0** |
| ALUOp | **10** |
| MemWrite | **0** |
| ALUSrc | **0** |
| RegWrite | **1** |
| **Sign-extend** | Input |  | **X** |
| Output |  | **X** |
| **ALU**  **Control** | Input | Instruction [5-0] | **100010** |
| ALUOp | **10** |
| Output | Operation | **0110 (sub)** |
| **Shift-left-2**  **(dùng cho lệnh beq hoặc bne)** | Input |  | **X** |
| Output |  | **X** |