Computer Architecture

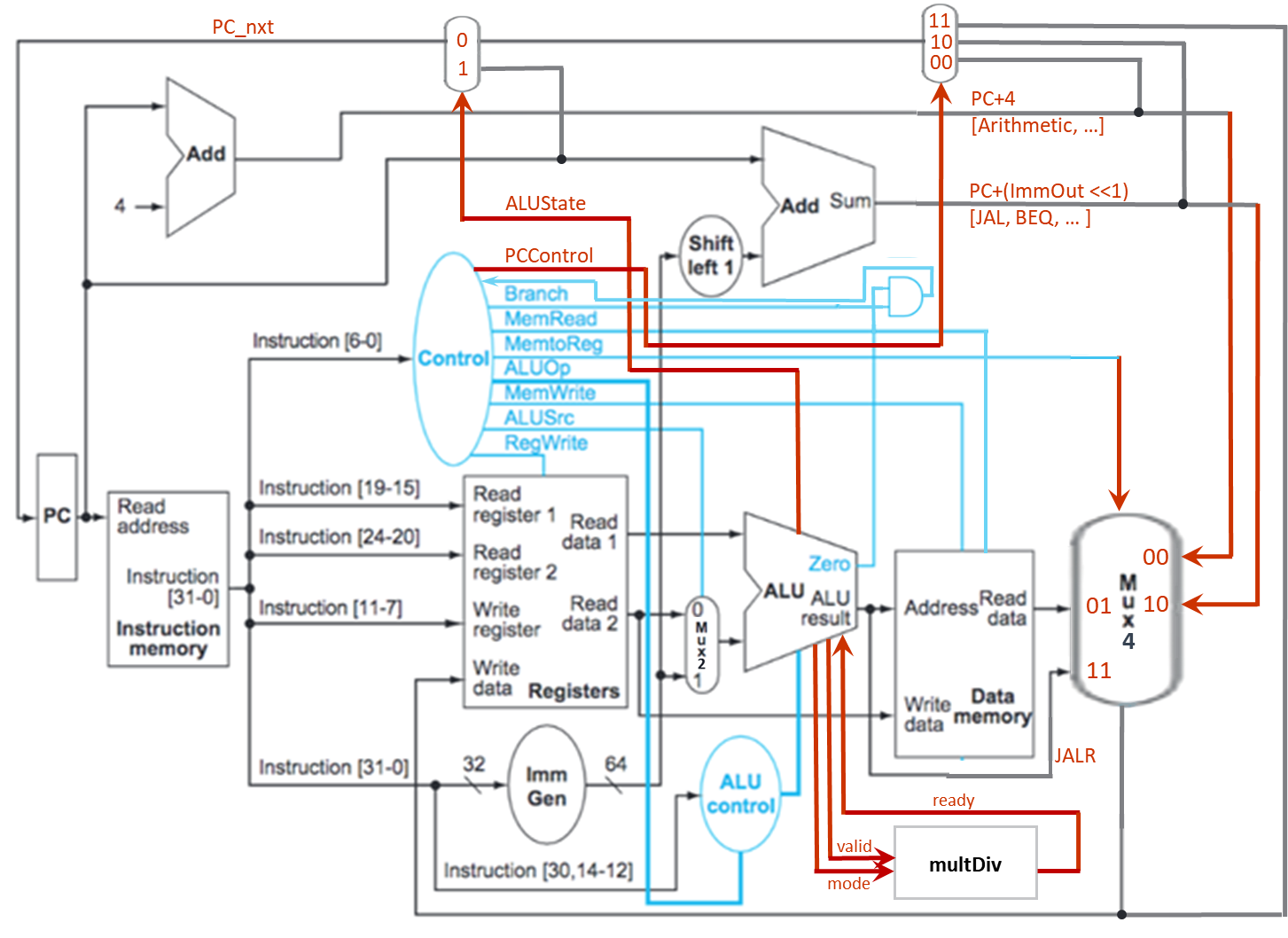
Final Project Report

2020/7/3

B05901168 陳冠豪 D07921016 王世全

1. CPU Architecture

The controls ALUState and PCControl are added in the CPU. ALUState is added to identify the state of ALU whether it’s operating under single cycle instructions (SCYCLE) or muti-cycle instructions (MCYCLE). The PCControl is added to control PC operation among PC+4, PC+ImmOut, and ALU’s ourput, ALUresult. A multiplexer Mux4 is re-designed for integrating the mux needs of PC states and that between memory data and ALUresult.



1. Data path of instructions
2. R-TYPE

The R-TYPE instructions include ADD, SUB, SLL, SLT, SLTU, XOR, SRL, SRA, OR, and AND. Their control signal are listed in Table 2.1. The ALUOp is determined by the instruction’s “func” field and the state “000” controls ALU doing arithmetic operations. The ALUSignal controls the type of arithmetic operation in ALU as shown in Table 2.1. The instruction path are shown in the Figure 2.1 below.

Table 2.1 R-TYPE instructions’ control signals

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Operation** | **PCControl** | **Branch** | **MemRead** | **MemroReg** | **ALUOp** | **MemWrite** | **ALUSrc** | **RegWrite** | **ALUSignal** | **ALUState** | **valid** | **mode** |
| ADD | 00 | 0 | 0 | 00 | 000 | 0 | 0 | 1 | 0000 | 0 | 0 | x |
| SUB | 00 | 0 | 0 | 00 | 000 | 0 | 0 | 1 | 0001 | 0 | 0 | x |
| SLL | 00 | 0 | 0 | 00 | 000 | 0 | 0 | 1 | 0010 | 0 | 0 | x |
| SLT | 00 | 0 | 0 | 00 | 000 | 0 | 0 | 1 | 0011 | 0 | 0 | x |
| SLTU | 00 | 0 | 0 | 00 | 000 | 0 | 0 | 1 | 0100 | 0 | 0 | x |
| XOR | 00 | 0 | 0 | 00 | 000 | 0 | 0 | 1 | 0101 | 0 | 0 | x |
| SRL | 00 | 0 | 0 | 00 | 000 | 0 | 0 | 1 | 0110 | 0 | 0 | x |
| SRA | 00 | 0 | 0 | 00 | 000 | 0 | 0 | 1 | 0111 | 0 | 0 | x |
| OR | 00 | 0 | 0 | 00 | 000 | 0 | 0 | 1 | 1000 | 0 | 0 | x |
| AND | 00 | 0 | 0 | 00 | 000 | 0 | 0 | 1 | 1001 | 0 | 0 | x |

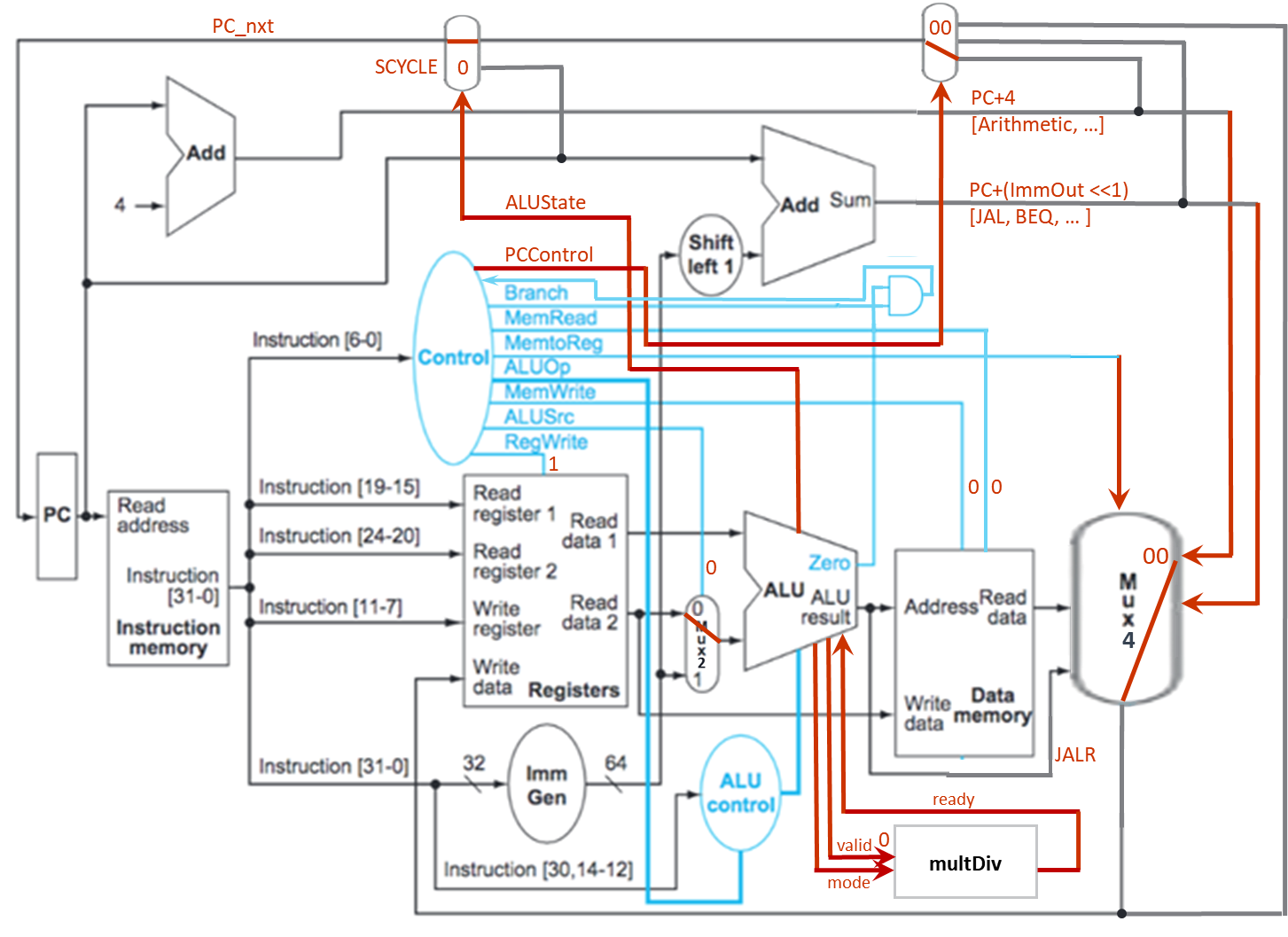


Figure 2.1 R-TYPE instructions’ data path

1. I-TYPE

The I-TYPE instructions include ADDI, SLTI, SRLI, SRAI, etc. The ALUSignal controls the type of operation in ALU as shown in Table 2.2. The instruction path are shown in the Figure 2.2 below.

Table 2.2 I-TYPE instructions’ control signals

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Operation** | **PCControl** | **Branch** | **MemRead** | **MemroReg** | **ALUOp** | **MemWrite** | **ALUSrc** | **RegWrite** | **ALUSignal** | **ALUState** | **valid** | **mode** |
| ADDI | 00 | 0 | 0 | 00 | 001 | 0 | 1 | 1 | 0000 | 0 | 0 | x |
| SLLI | 00 | 0 | 0 | 00 | 001 | 0 | 1 | 1 | 0010 | 0 | 0 | x |
| SLTI | 00 | 0 | 0 | 00 | 001 | 0 | 1 | 1 | 0011 | 0 | 0 | x |
| SLTIU | 00 | 0 | 0 | 00 | 001 | 0 | 1 | 1 | 0100 | 0 | 0 | x |
| XORI | 00 | 0 | 0 | 00 | 001 | 0 | 1 | 1 | 0101 | 0 | 0 | x |
| SRLI | 00 | 0 | 0 | 00 | 001 | 0 | 1 | 1 | 0110 | 0 | 0 | x |
| SRAI | 00 | 0 | 0 | 00 | 001 | 0 | 1 | 1 | 0111 | 0 | 0 | x |
| ORI | 00 | 0 | 0 | 00 | 001 | 0 | 1 | 1 | 1000 | 0 | 0 | x |
| ANDI | 00 | 0 | 0 | 00 | 001 | 0 | 1 | 1 | 1001 | 0 | 0 | x |

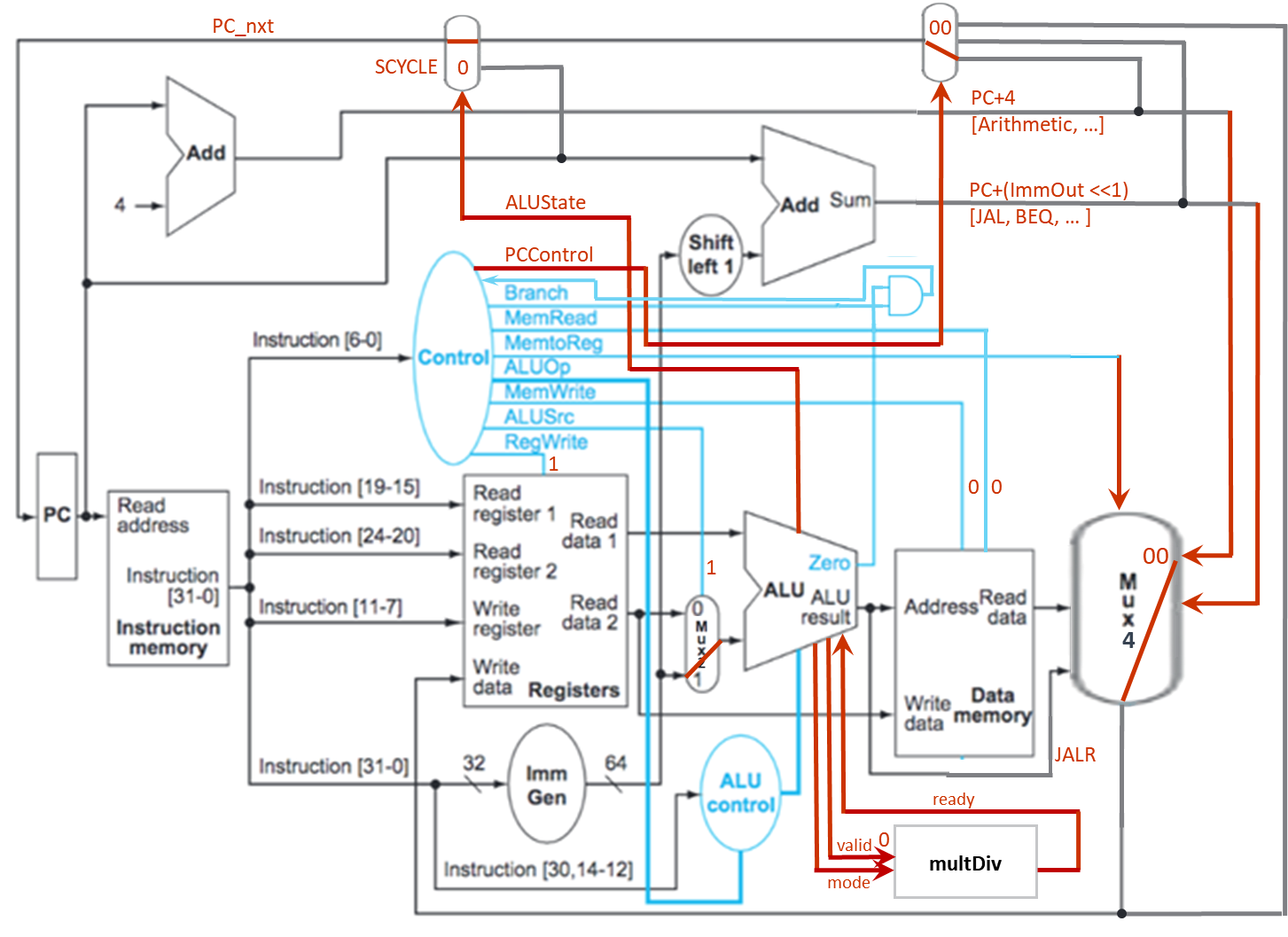


Figure 2.2 I-TYPE instructions’ data path

1. B-TYPE

The B-TYPE instructions include BEQ and BNE, etc. The ALUSignal controls the type of operation in ALU as shown in Table 2.3. The instruction path are shown in the Figure 2.3 below.

Table 2.3 B-TYPE instructions’ control signals

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Operation** | **PCControl** | **Branch** | **MemRead** | **MemroReg** | **ALUOp** | **MemWrite** | **ALUSrc** | **RegWrite** | **ALUSignal** | **ALUState** | **valid** | **mode** |
| BEQ | 01 | 1 | 0 | 00 | 011 | 0 | 0 | 1 | 0001 | 0 | 0 | x |
| BNE | 01 | 1 | 0 | 00 | 011 | 0 | 0 | 1 | 0001 | 0 | 0 | x |

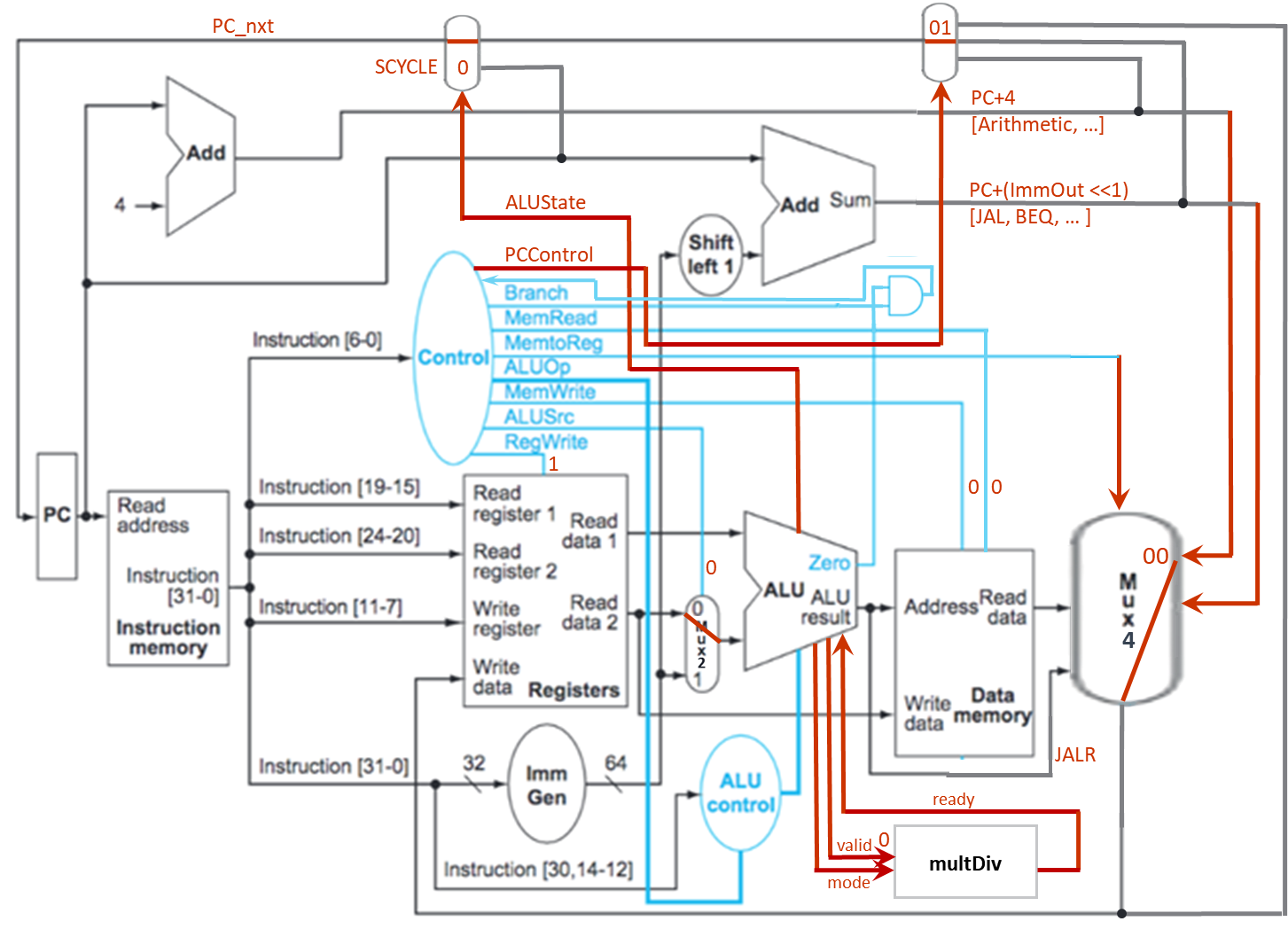


Figure 2.3 B-TYPE instructions’ data path

1. LI-TYPE

The LI-TYPE instructions are LW, etc. The ALUSignal controls the type of operation in ALU as shown in Table 2.4. The instruction path are shown in the Figure 2.4 below.

Table 2.4 LI-TYPE instructions’ control signals

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Operation** | **PCControl** | **Branch** | **MemRead** | **MemroReg** | **ALUOp** | **MemWrite** | **ALUSrc** | **RegWrite** | **ALUSignal** | **ALUState** | **valid** | **mode** |
| LW | 00 | 0 | 1 | 01 | 110 | 0 | 1 | 1 | 0000 | 0 | 0 | x |

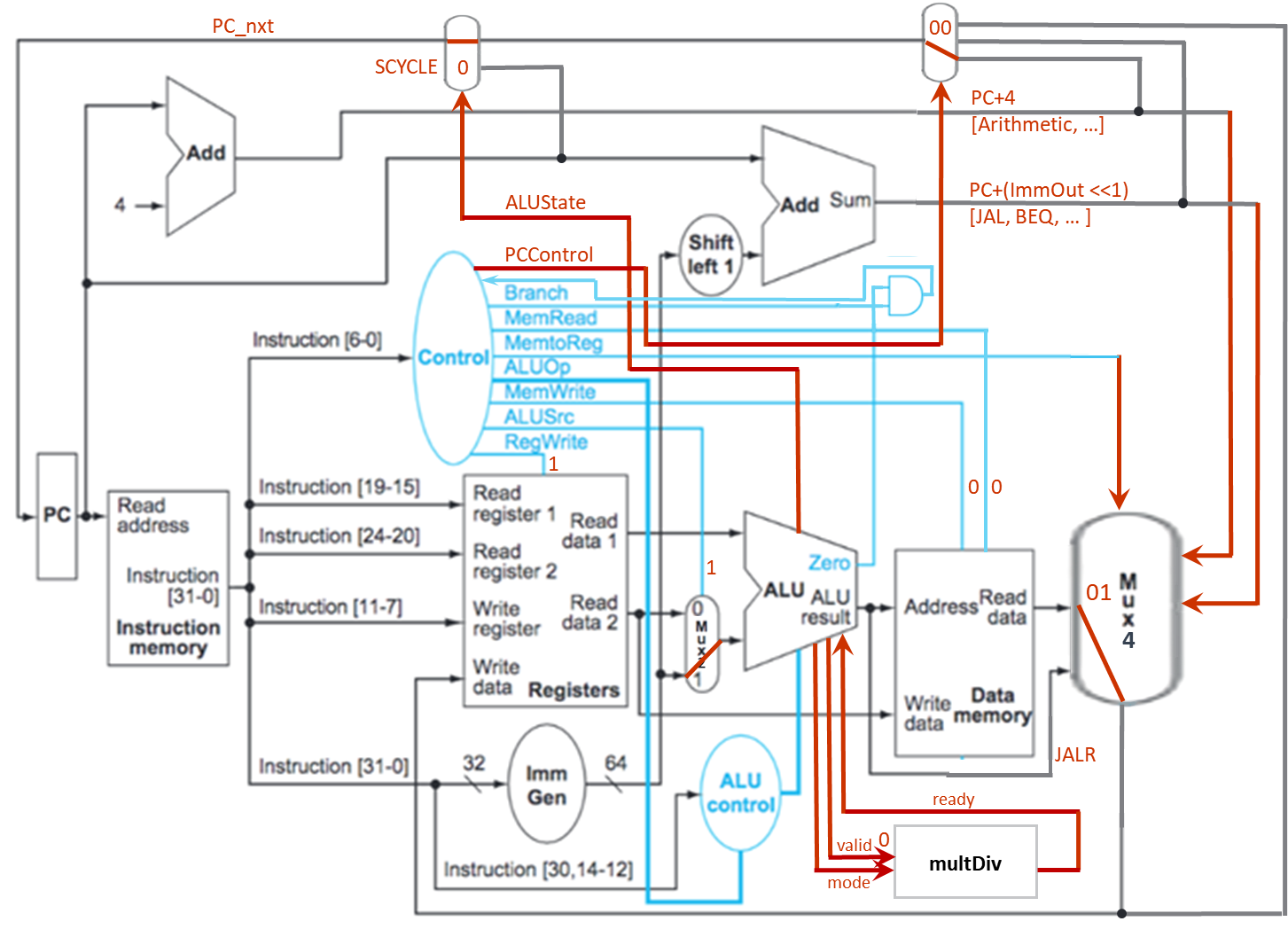


Figure 2.4 LI-TYPE instructions’ data path

1. S-TYPE

The S-TYPE instructions are such as SW, etc.

1. U-TYPE

The U-TYPE instructions are such as AUIPC, etc.

1. J-TYPE

The J-TYPE instructions are such as JAL, etc.

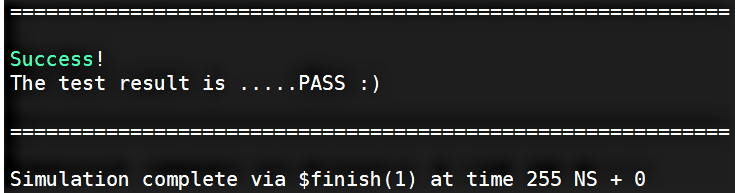
1. JI-TYPE

The JI-TYPE instructions are such as JALR, etc

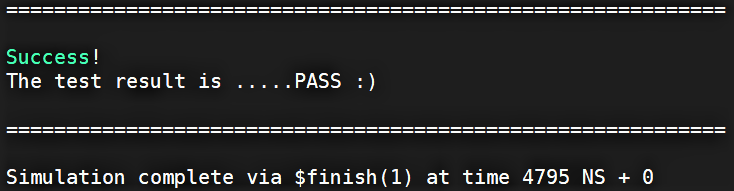
1. Multi-cycle instructions

//todo

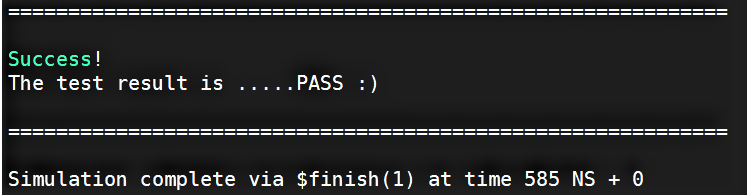
1. Total simulation time
   1. Leaf: a = 1, b = 9, c = 2, d = 2



* 1. Fact: n = 10



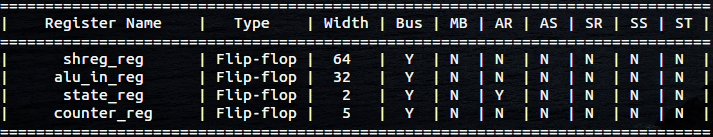
* 1. (Bonus) HW1: n = 10



1. Observation

//todo

1. Snapshot of the “Register Table”



1. Work distribution

|  |  |  |
| --- | --- | --- |
|  | 陳冠豪 | 王世全 |
| CHIP.v | V | V |
| Control.v | V | V |
| Alu\_Control.v | V | V |
| Alu.v | V | V |
| Imm\_Gen.v | V |  |
| Mux.v | V |  |
| leaf | V | V |
| fact | V | V |
| hw1 | V | V |
| Report |  | V |