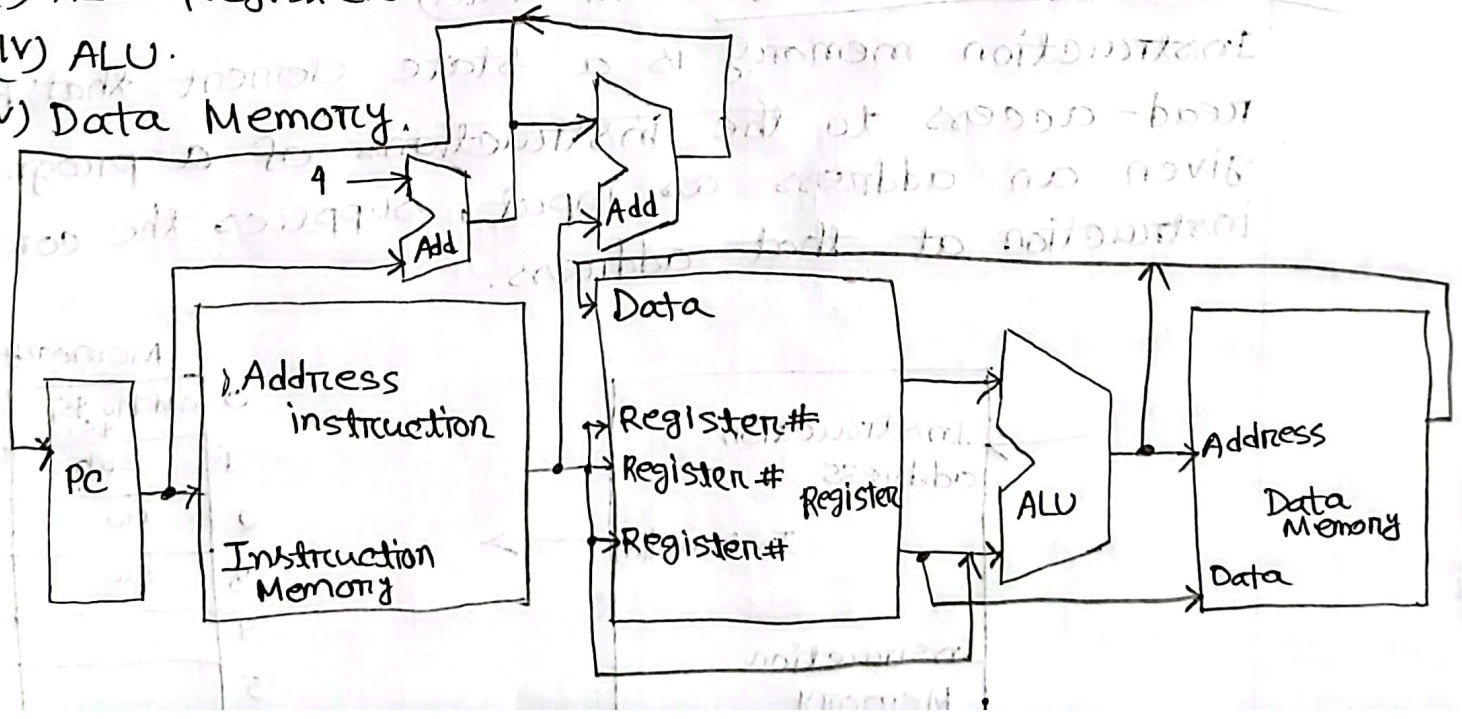
Datapath

- Datapath consists of the functional units of the processor.
- Two types of ~~data~~ element:
  1. Elements that hold data - Program counter, register file, instruction memory etc.
  2. Elements that operate on data - ALU, address etc.
- There are two paths for every instruction.
  (I) Fetch cycle → Same for all instruction (I, R, J).
  (II) Execution cycle → Depends on the type of the instruction.

5 components:
(I) PC (Program Counter)
(II) Instruction Memory
(III) ~~ALU~~ Register.
(IV) ALU.
(V) Data Memory.



Explain:

1. Program counter এর কাজ হলো আমরা যে instruction execute করবো তার address save রাখা।
So, যে instruction execute করবো তার address থাকবে program counter এর কাছে।

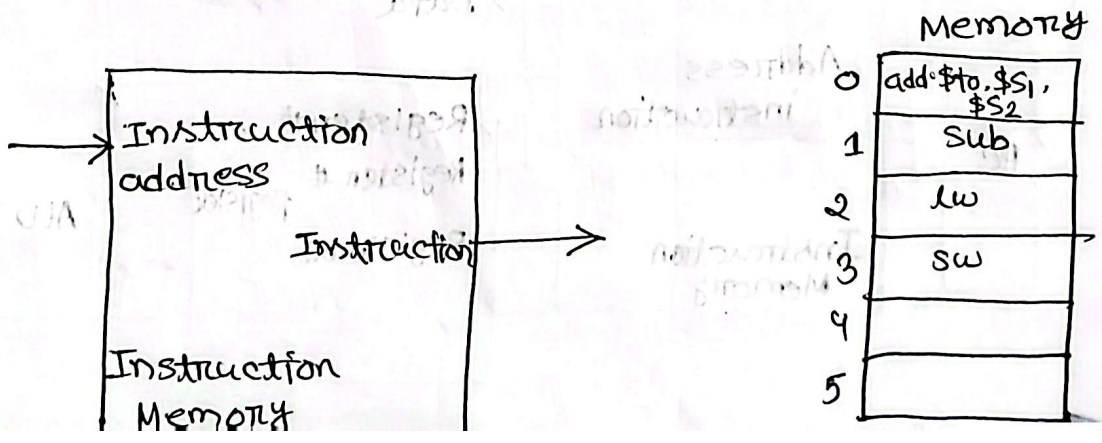
2. Instruction memory এর কাজ হলো- ~~instruction~~ ~~কে~~ address এ গিয়ে যে instruction থাকবে সেটি load করবে।

3.4. ALU তে যে arithmatic operation করবো সেটি ALU এর through তে হবে। ALU এর তে ২টি variable যাবে এবং ১টি flag থাকবে যেখানে বলা হবে কোন operation করা হবে (addition/sub/mul).
২টি variable যাবে এবং operation calculate করবে।

5. Data memory কে ১টি address দিবে সেটি read/write করবে।

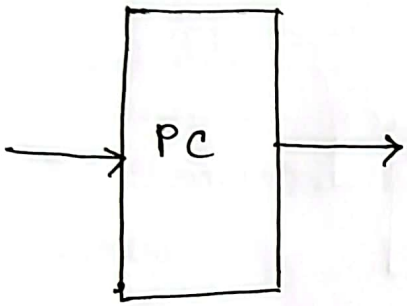
1. Instruction Memory: (Hold data)

Instruction memory is a state element that provides read-access to the instructions of a program and given an address as input, supplies the corresponding instruction at that address.



2. Program Counter (PC): (Hold data)

The PC is a state element that holds the address of the ~~counter~~ current instruction.



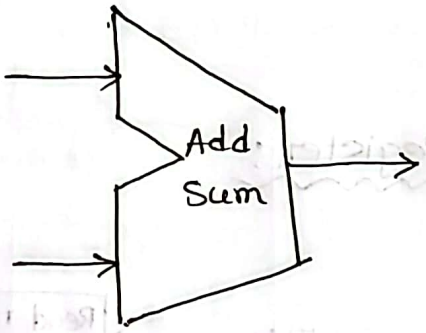
* The arrows on either side indicate that the PC state element is both readable and ~~writable~~ writeable.

~~3. ALU: (Adde~~

3. Adder (ALU):

The adder is responsible for incrementing the PC to hold the address of the next instruction.

It takes two input values, adds them together and outputs the result.

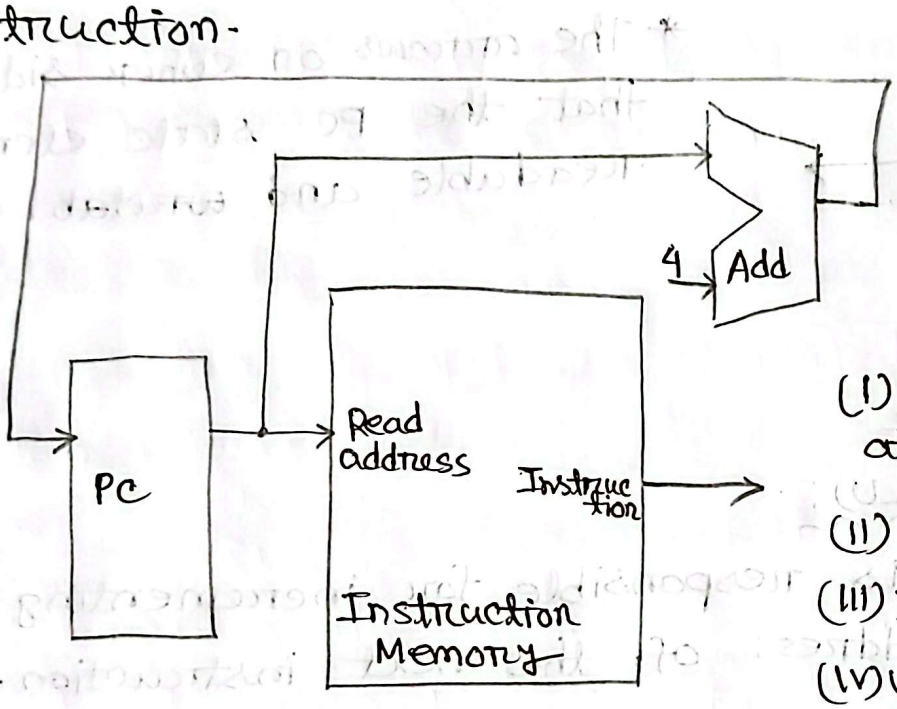


Now we have instruction memory, PC and adder datapath elements. Now, we can talk about the general steps taken to execute a program.

- Instruction fetching: Use the address in the PC to fetch the current instruction from instruction memory
- Instruction decoding: Determine the fields within the instruction.

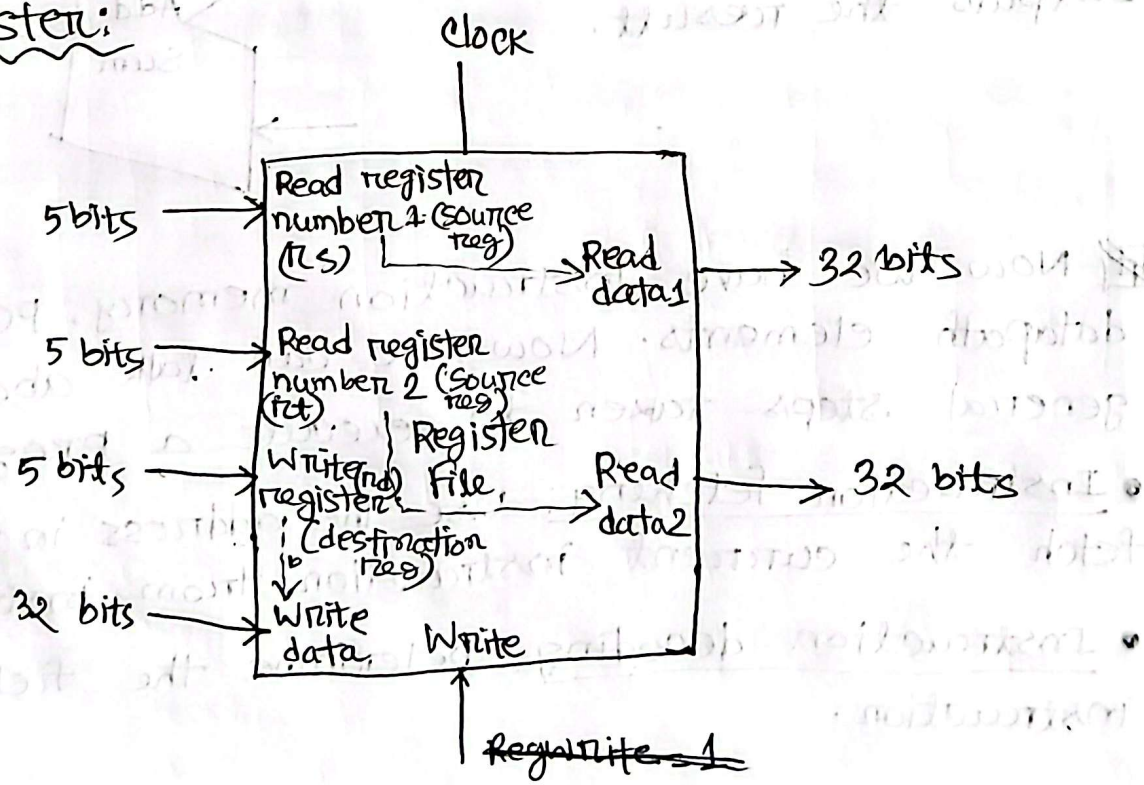
- Instruction execution: Perform the operation indicated by the instruction.

Update the PC to hold the address of the next instruction.



(I) Fetch the instruction at the address in PC.
(II) Decode the instruction.
(III) Execute the instruction.
(IV) Update the PC to hold the address to of the next instruction.

4. Register:



R-type Instruction:

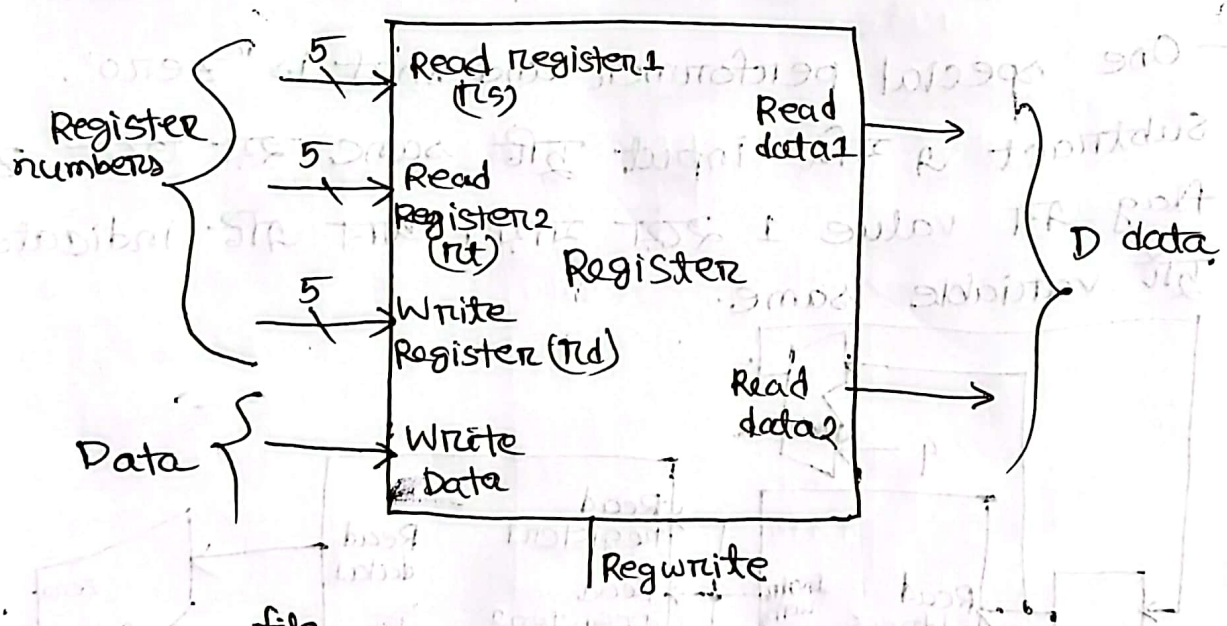
Name	Fields					
Field size	6 bits	5 bits	5 bits	5 bits	5 bits	6 bits
R format	Op	rs	rt	rd	shamt	~~fun~~ funct

Two elements used to implement R-type instructions.

Register.

ALU.

Register:



A register file is a collection of readable/writeable register.

- Read Register 1: First source register. 5 bits wide.
- Read Register 2: Second source register. 5 bits wide.
- Write Register: Destination register. 5 bits wide
- Write data: Data to be written to a register. 32 ~~bits~~ bits wide.

At the bottom, we have the RegWrite input. A writing operation only occurs when this bit is set.

The two outputs:

- Read data 1: Contents of source register 1.
- Read data2: Contents of source register 2.

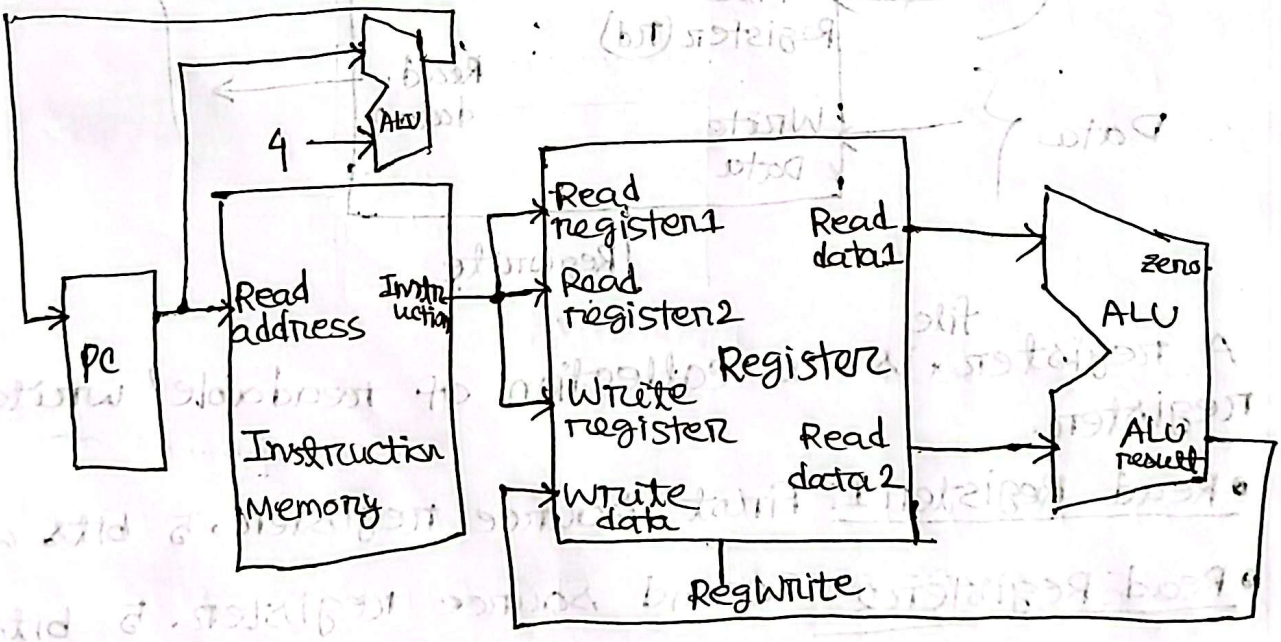
ALU:

ALU perform the operation indicated by the instruction.

- Two inputs and one output.

- One special performer and that is "zero".

Subtract ও যদি input দুটি same হয় তবে zero flag এর value 1 হয়ে যায়, আর এটি indicate করে দুটি variable same.



1. Grab instruction address from PC
2. Fetch instruction from instruction memory.
3. Decode instruction
4. Pass rs, rt & rd into read register & write register arguments.

5. Retrieve data from read register 1 and read register 2 (rs & rt).

6. pass contents of rs & rt into the ALU as operands of the operation to be performed.

7. Retrieve result of operation performed by ALU and pass back as the write data argument of the register file (with the RegWrite bit set)

8. Add 4 bytes to the PC value to obtain the word-aligned address of the next instruction.

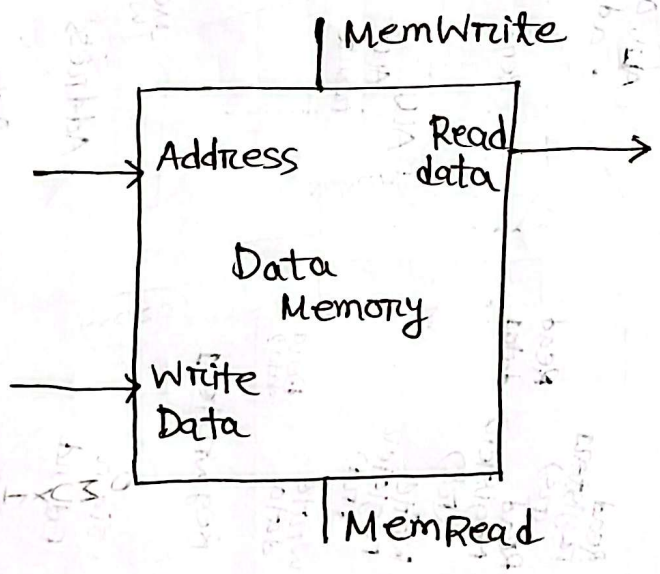
I-Format Instructions:

Two component:
1. Data memory unit
2. Sign extension bit

Name	Fields					
Field Size	6bits	65 bits	5bits	5 bits	5 bits	6 bits
I Format	OP	rs	rt	immed		

Data memory unit

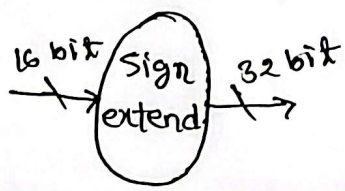
- ২টি address কে input দেওয়া হয় data memory unit এ। data memory unit এ সেই address কে memory তে plot করে। Address এ যে value থাকে সেটিকে output করে দেয়। data read & write এর জন্য MemRead এবং MemWrite.



- Two inputs. Address and Write data.
- One output.
- Reads & writes are signaled by MemRead and MemWrite respectively.

Sign Extend:

16 bit কে 32 bit এ convert করে দিবে। যদি 16 bit এর number positive হয় অর্থাৎ ~~no MS~~ MSB = 0 হয় তবে ~~0 16টি 0 দিয়ে~~ MSB এ 16টি 0 দিয়ে 32 bit করে দিব। same ভাবে negative এর ক্ষেত্রে 1 দিব।

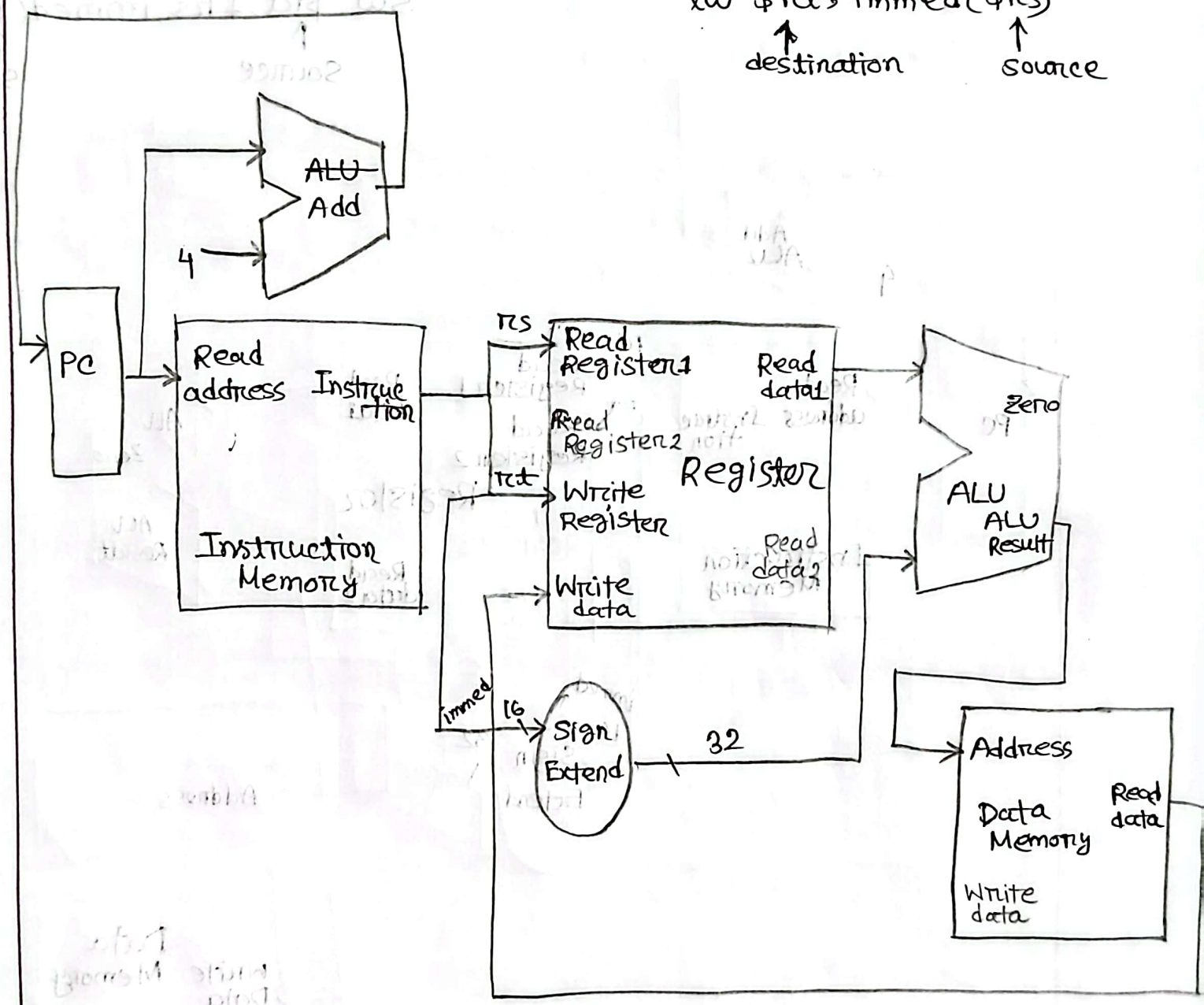


The sign extension element takes as input a 16-bit wide value to be extended to 32-bits.

Datapath for Load Word

lw $rt, immed($rs)

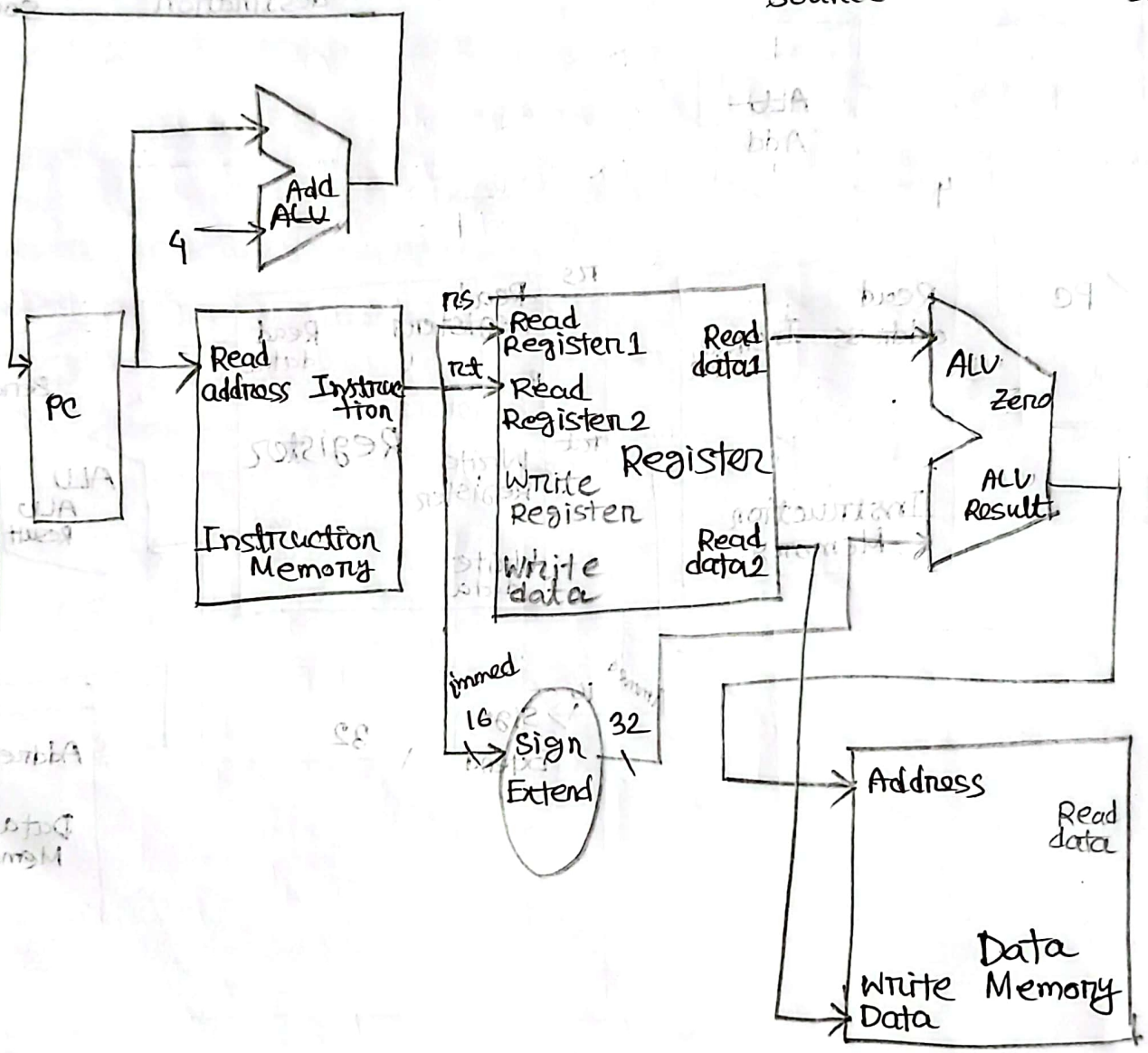
destination ↑ source ↑



Datapath for Store Word

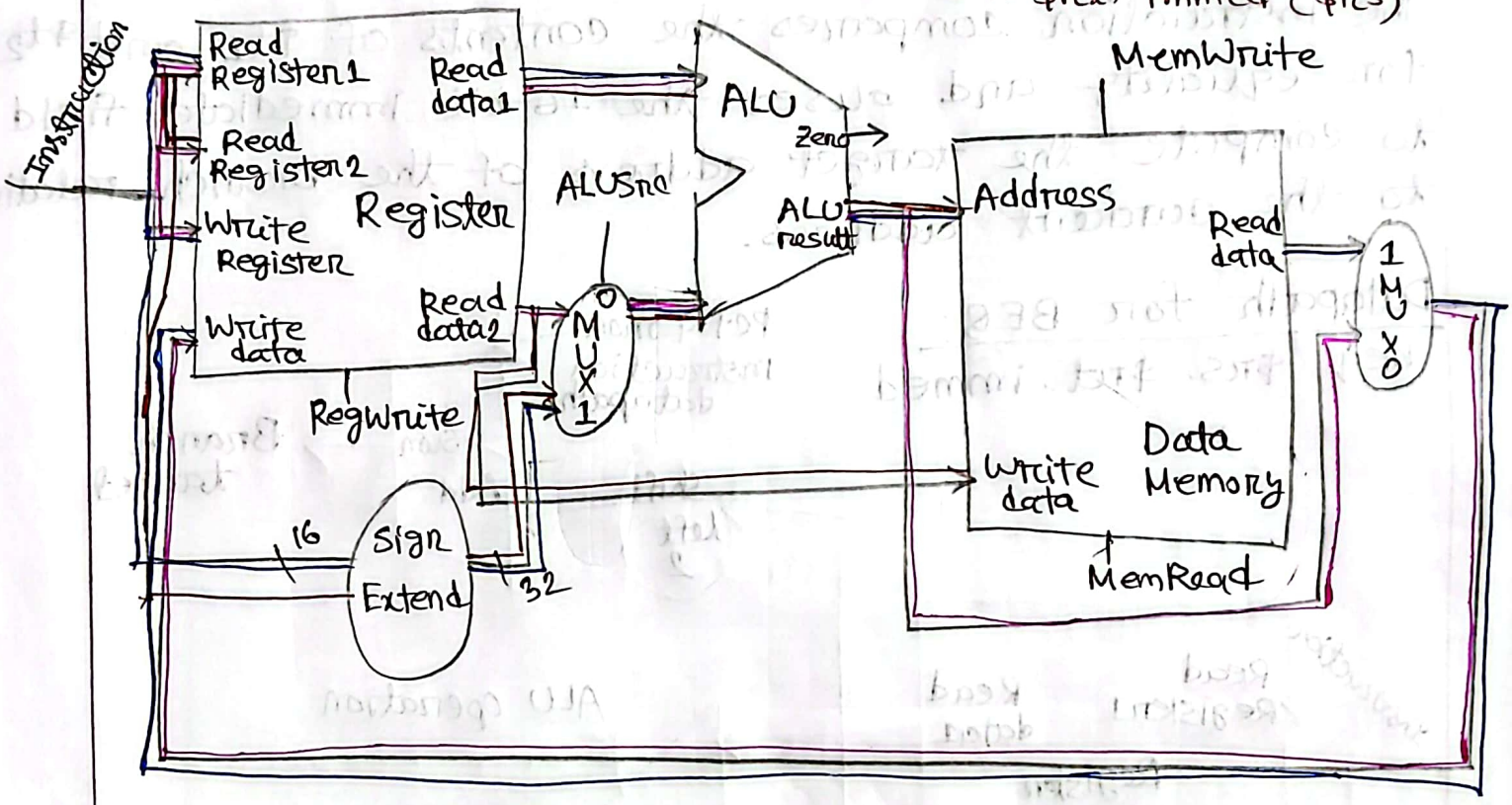
sw, $rt, $rs, immed($rs)

Source Source



Datapath for R-format and ~~I-format~~ Memory Access:

add $rd, $rs, $rt
lw, $rt, immed ($rs)
sw, $rt, immed ($rs)



~~Datapath~~

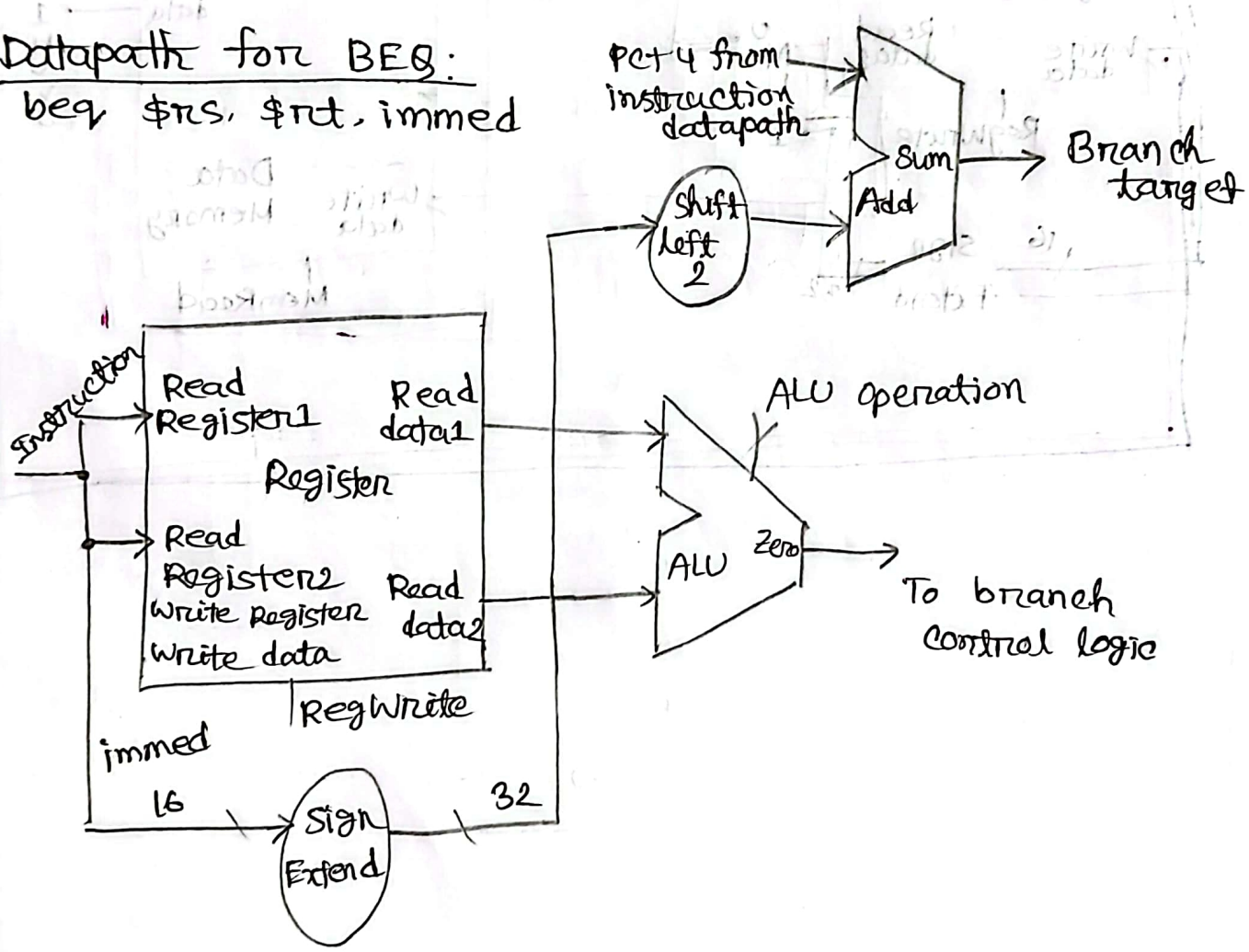
Branching Instructions:

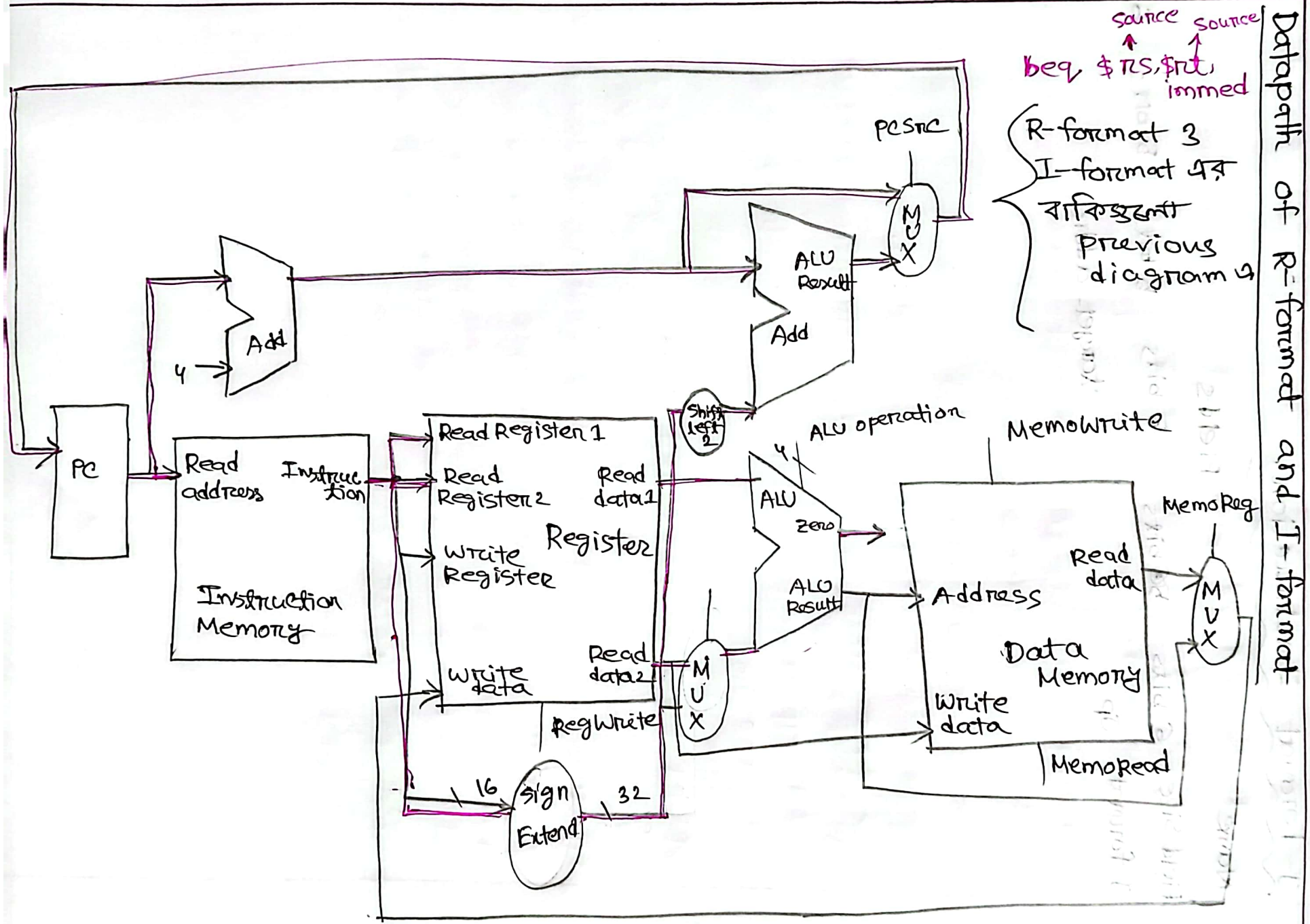
beq $t1, $t2, target

The instruction compares the contents of $t1 and $t2 for equality and ouses the 16-bit immediate field to compute the target address of the branch relative to the current address.

Datapath for BEQ:

beq $rs, $rt, immed




Source
Source
beq $rs, $rt, immed
R-format 3
I-format এর
বাকিগুলো
Previous
diagram ও
PCSrc
M
U
X
ALU
Result
Add
Add
4
Shift
left
2
PC
Read
address
Instruc
tion
Instruction
Memory
Read Register 1
Read
Register 2
Register
Write
Register
Write
data
Read
data 1
Read
data 2
RegWrite
16
Sign
Extend
32
ALU operation
4
ALU
zero
ALU
Result
MemoWrite
MemoReg
Address
Read
data
Data
Memory
Write
data
MemoRead
Datapath of R-format and I-format