NWEN 242

6. Processor: single cycle datapath

Agenda

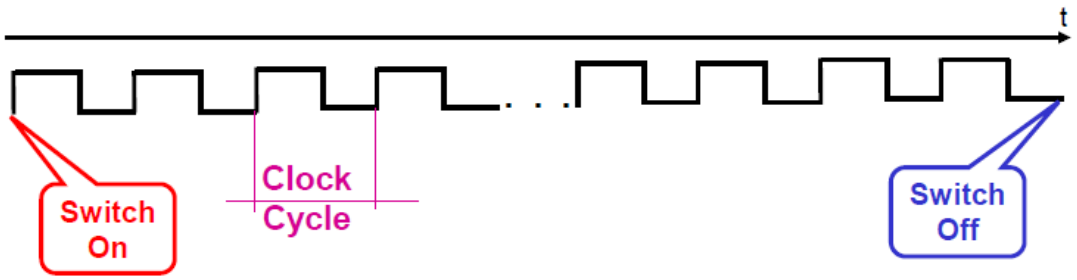
- What is a processor datapath?
- Design instruction-fetch logic
- Design 3 mini-datapaths
  - R-type (for e.g. add)
  - Memory (for e.g. lw and sw)
  - Branch (for e.g. beq)
- Combine the instruction-fetch logic with the 3 mini-datapaths

What is a processor datapath?

- Datapath
  - Literally, the path along which data flows
  - Hardware that connects:
    - Instruction memory
    - Register file
    - ALU
    - Data memory

Quick exercise

- If a processor operates on 1 GHz, the duration of 1 clock cycle is
  - A. 1 nanosecond
  - B. 1 second
  - C. 1 minute
  - D. None of the above



Quick exercise

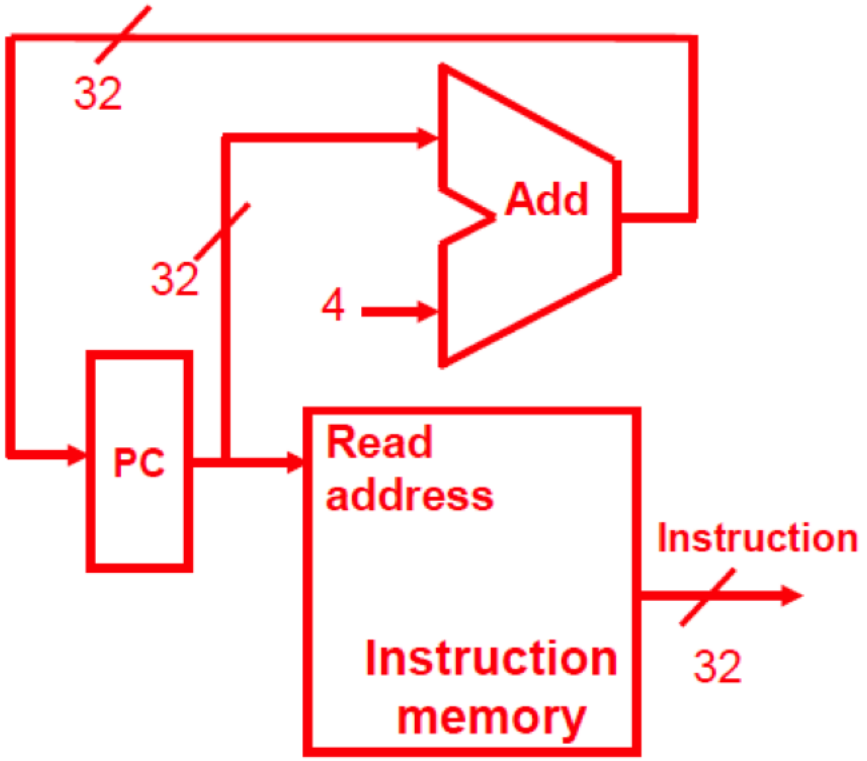
- The relationship between 1 nanosecond and 1 second is approximately the same as between:
  - A. 1 second and 1 day
  - B. 1 second and 16 weeks
  - C. 1 second and 12 months
  - D. 1 second and 32 years

Answer

- 1 second = 10^9 nanoseconds, therefore 1,000,000,000 ns
- 1 day = 24x3600 = 86,400 seconds
- 16 weeks = 16x7x86,400 = 9,676,800 seconds
- 12 months = 365x86,400 = 31,536,000 seconds
- 32 years = 32X31,536,000 = 1,009,152,000 seconds

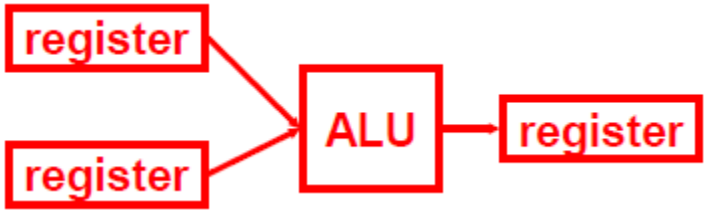
How instructions are fetched

- Instruction memory holds program instructions
- PC (program counter) holds the address of the instruction to be executed next
- Every clock cycle a new instruction is fetched from the memory and PC is incremented by a 4 adder



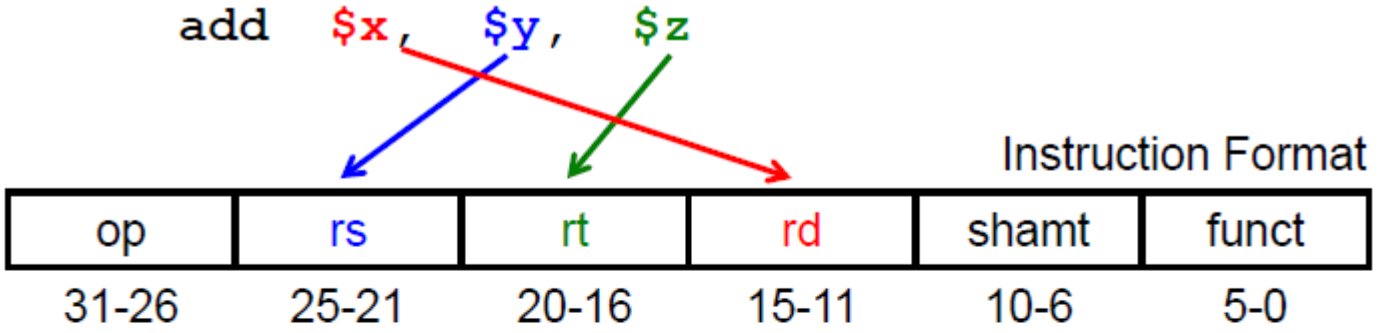
R-type mini datapath: what's required

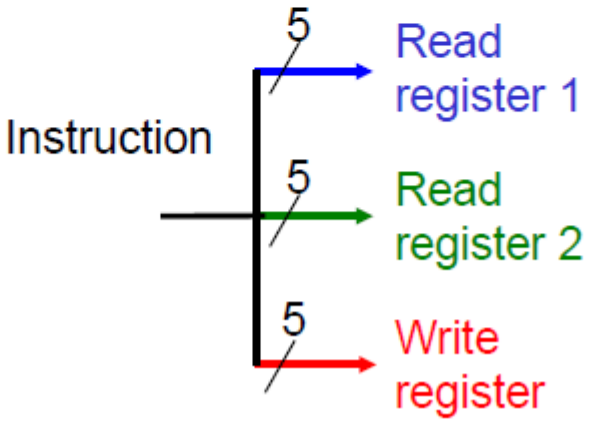
- Example instructions
  - `add $1, $2, $3` ⇒ add $1, $1, $3 ?
  - `sub $20, $2, $30`
  - `slt $1, $16, $17`
- General requirement
  - Two source registers
  - An ALU operation
  - One destination register



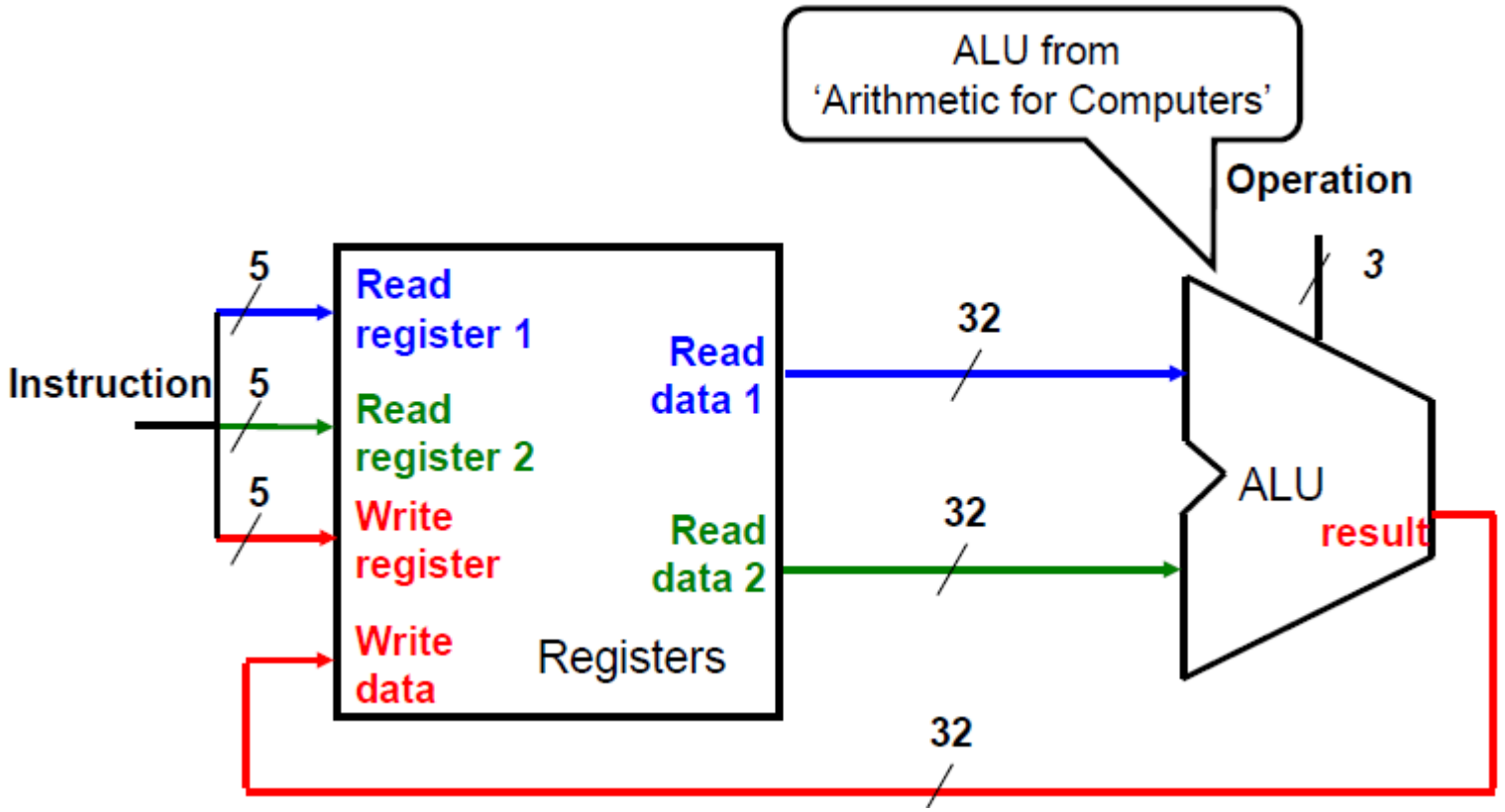
Where the instruction bits come from?

- Example



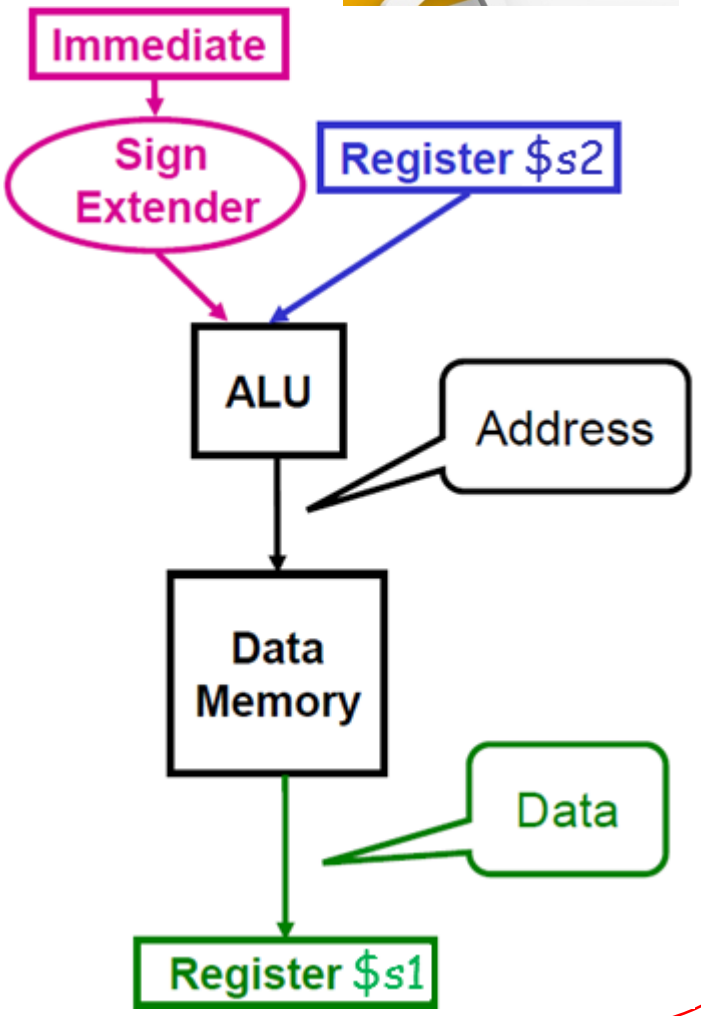


Implement R-type mini datapath



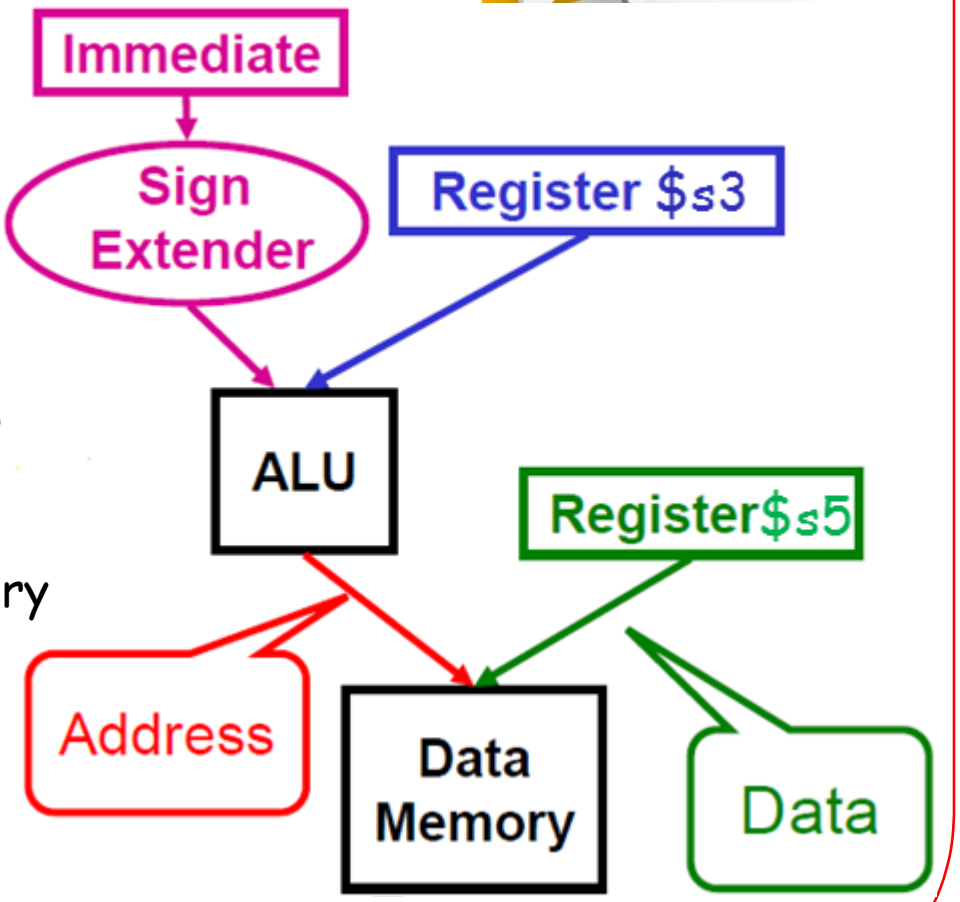
Read from memory: I-format

- Typical instruction:
  - lw $s1, offset($s2)
- Required activities
  - Read of the base address register
    - rs=$s2
  - 16-to-32 bit sign extension for the immediate field (i.e. offset)
  - ALU add to calculate memory address
  - Access data memory to read a word at the calculated address
  - Write data to the destination register
    - rt=$s1



Write to memory: what's required

- Typical instruction
  - sw $s5, offset($s3)
- Required activities
  - Read of the base address register
    - rs=$s3
  - 16-to-32 bit sign extension of the immediate field (i.e. offset)
  - ALU add to calculate memory address
  - Read data from register
    - rt=$s5
  - Store a word from $s5 to data memory

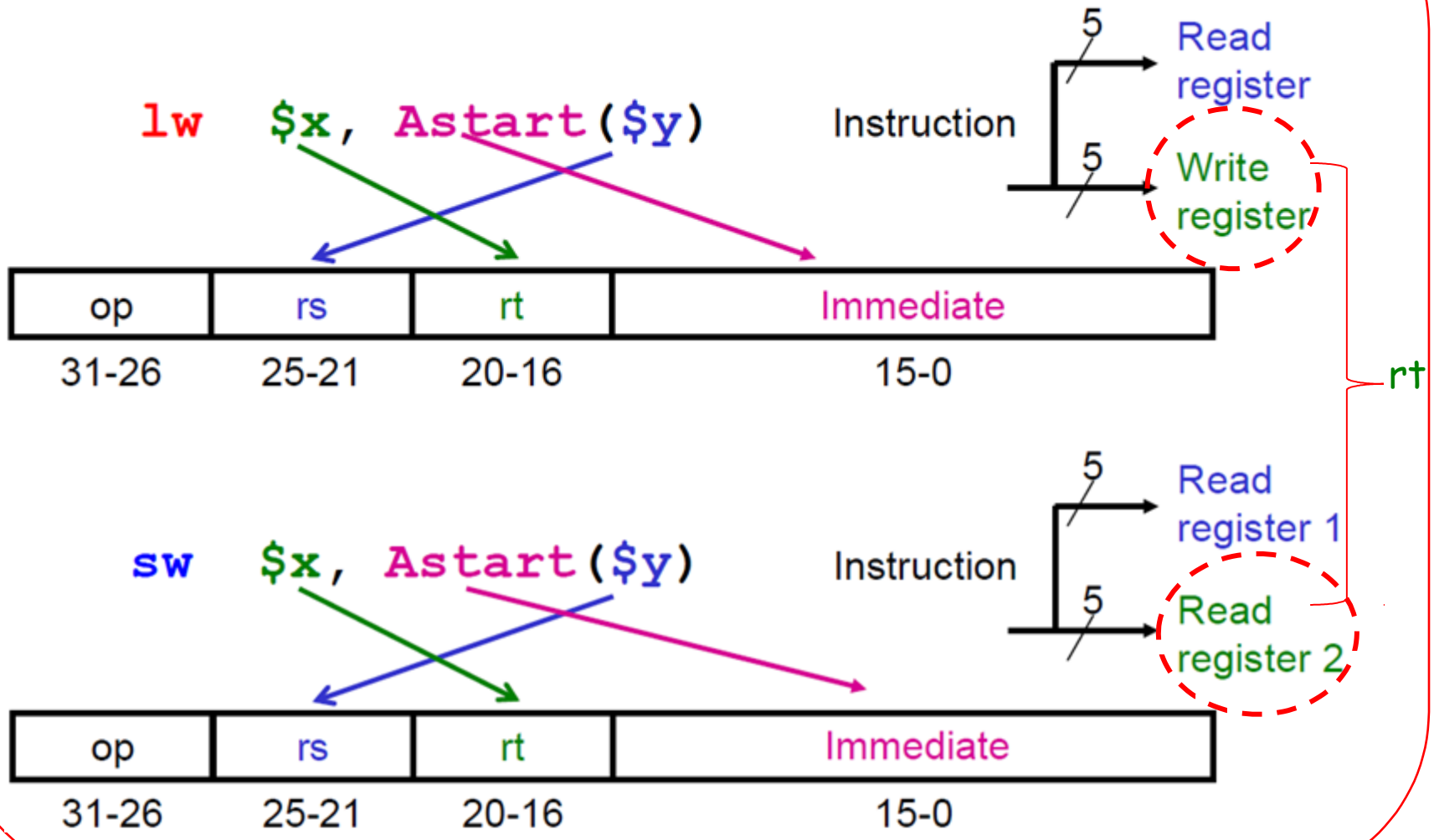


Quick exercise

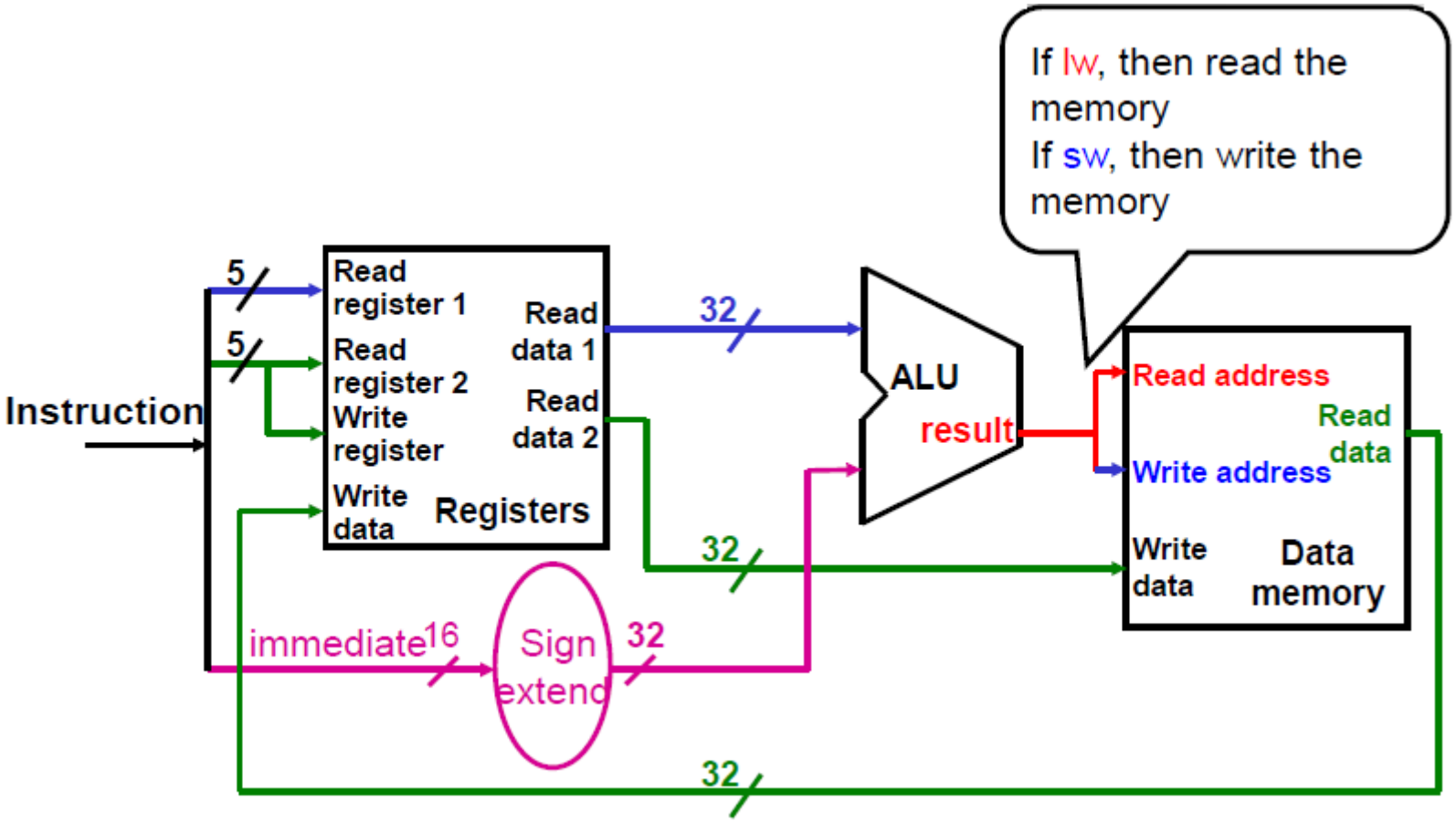
- If sign extension and register reading can be performed in 1ns
- If ALU takes 1ns
- If accessing the memory needs to be performed in 2ns
- If writing a register needs to be performed in 1ns

- What is the total time required for lw and sw?
  - A. lw=5ns, sw=5ns
  - B. lw=5ns, sw=4ns
  - C. lw=4ns, sw=5ns
  - D. None of the above

Where do the instruction bits come from?

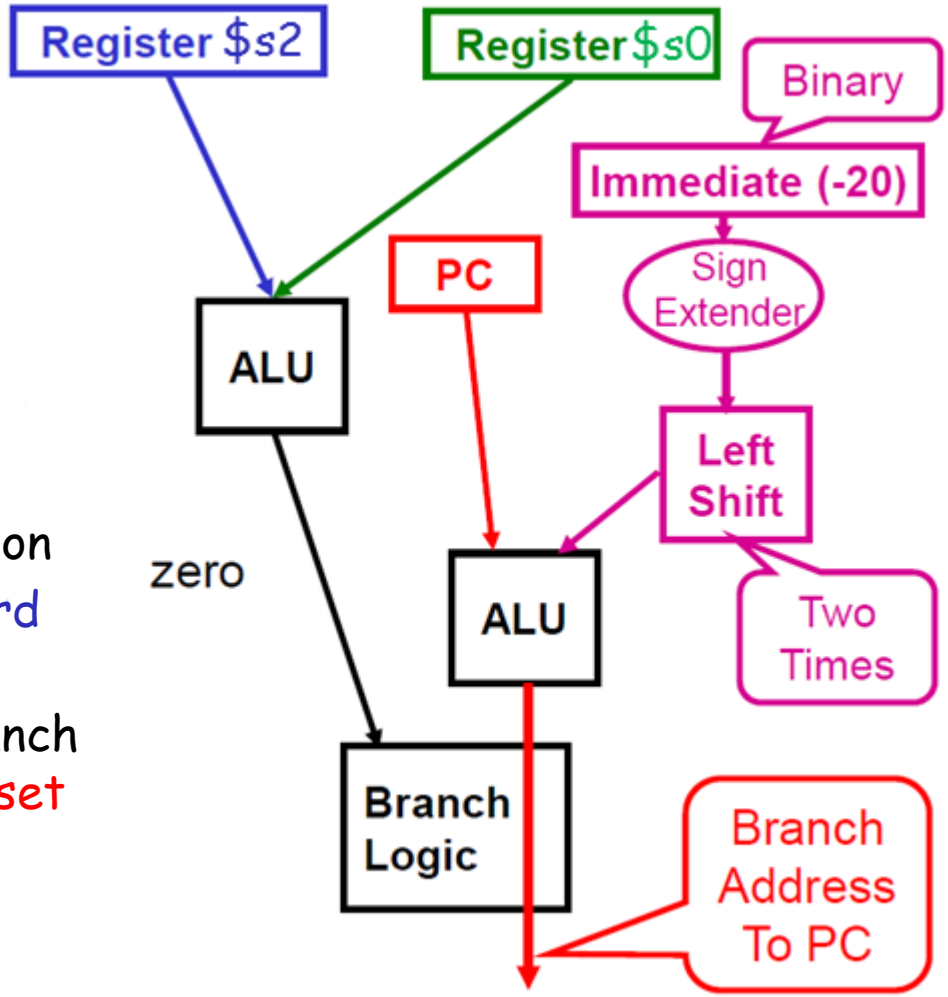


Implement memory mini datapath

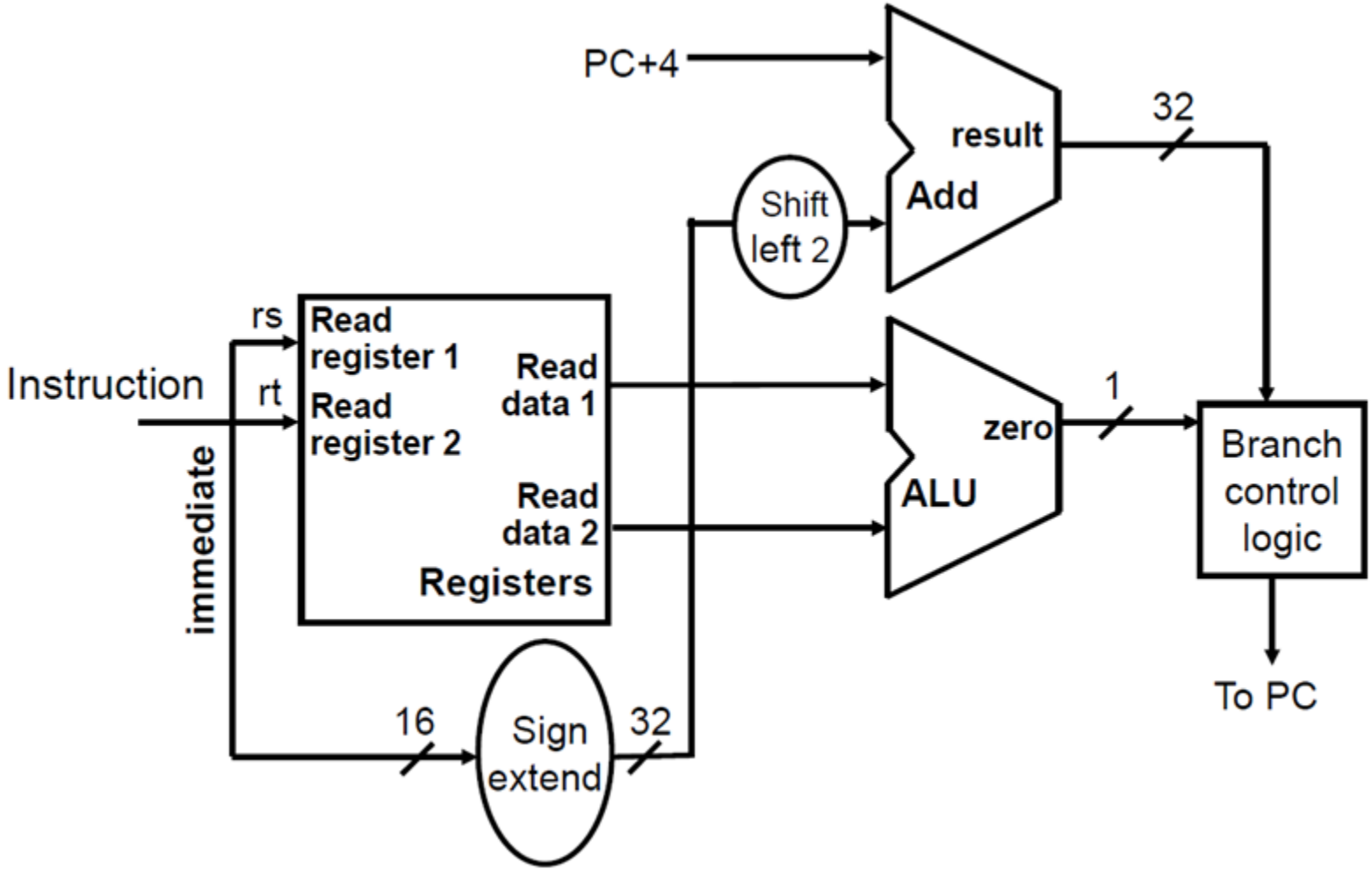


Branch mini datapath: what's required

- Typical instruction
  - beq $s2, $s0, -20
- Required activities
  - Read two register
    - $s2, $s0
  - ALU compare operation
  - 16-to-32 bit sign extension
  - Left shift to convert word into byte offset
  - ALU add to calculate branch address from PC and offset
  - Branch logic



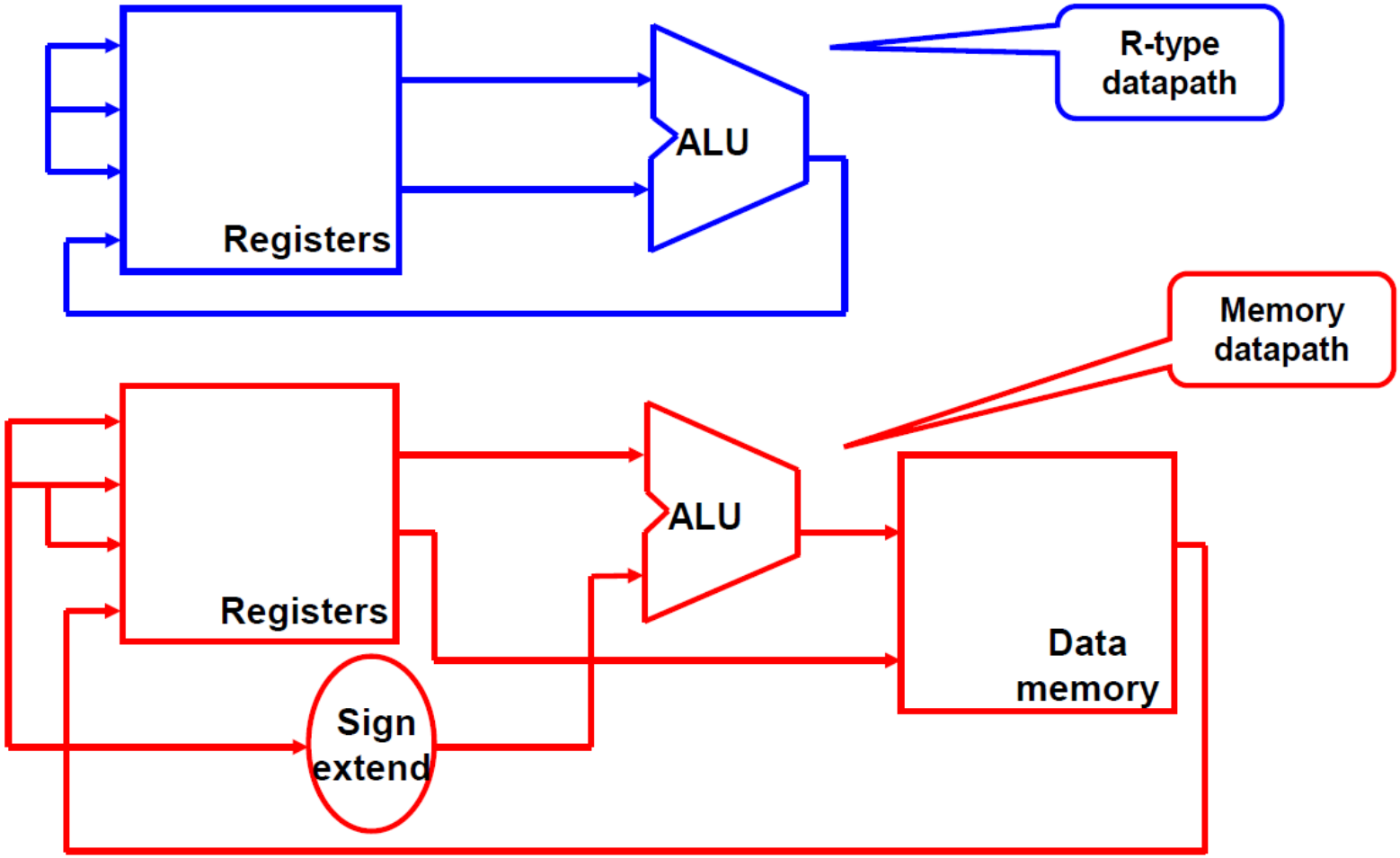
Implement branch mini datapath



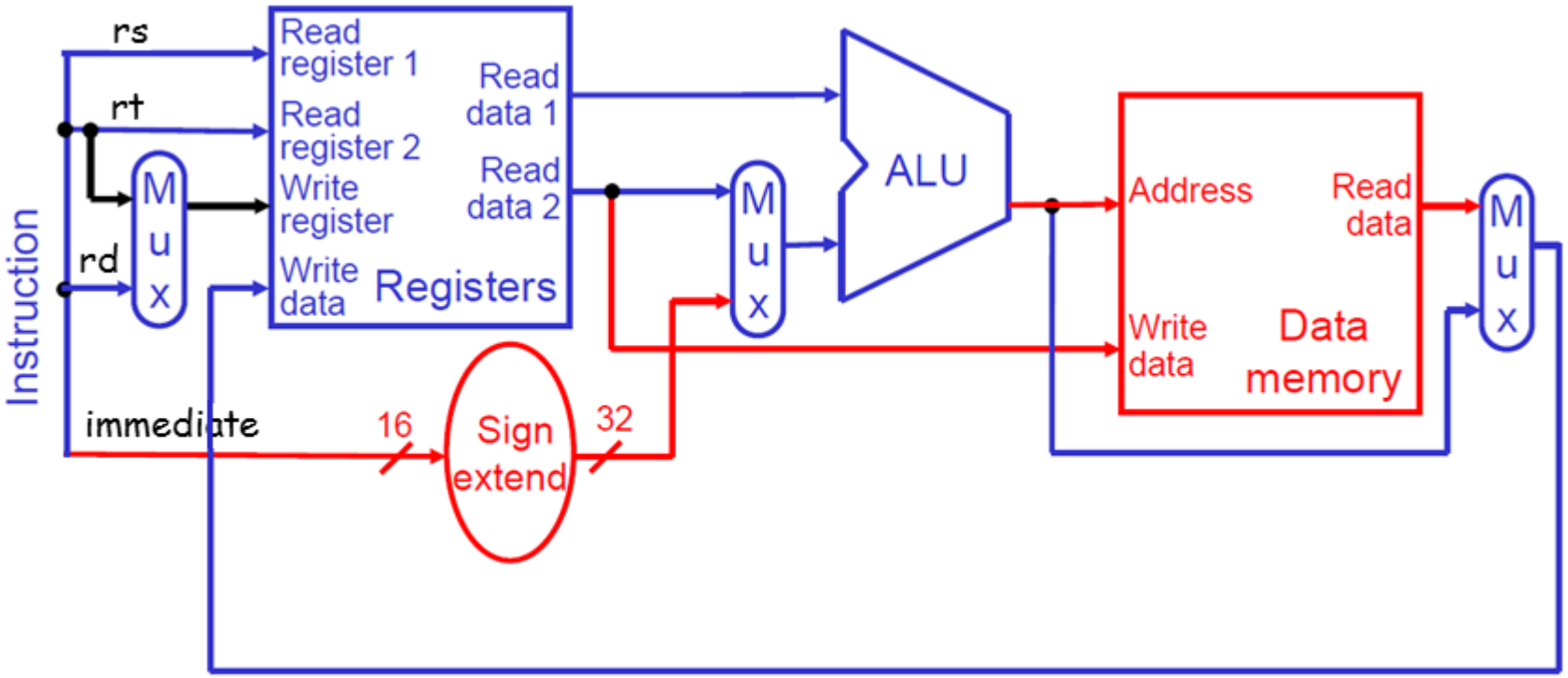
Combine separate mini-datapaths

- Sometime sharing is possible
  - e.g. sharing of register file
- Sometime sharing is not possible
  - e.g. the branch datapath needs two ALUs
    - One for comparison operation
    - One for calculating the branch instruction address

R-type & memory mini datapaths



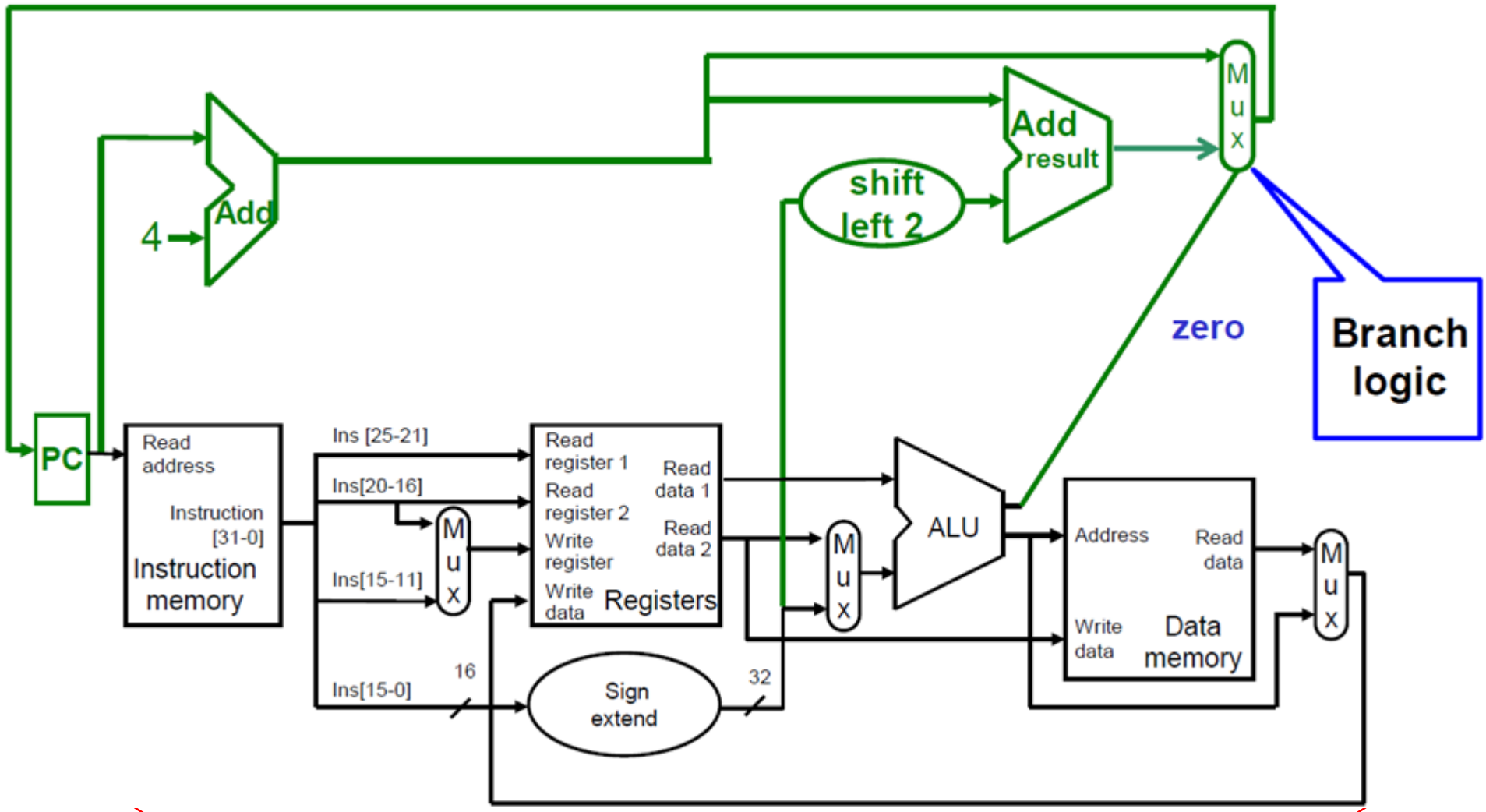
R-type & memory datapaths combined



Blue – R-type Datapath

Red – what Memory Datapath adds to R-type Datapath

All 3 mini datapaths & instruction fetch



Quick exercise

- Which hardware component in the combined datapath is always used for executing every instruction?
  - A. Instruction Memory
  - B. ALU
  - C. Sign extender
  - D. None of the above

Summary

- Instruction fetch logic
  - Fetches each 32-bit instruction
  - Updates PC
- Datapath
  - Literally, the path along which data flows
  - Connects
    - Instruction fields
    - Registers
    - ALU
    - Memory