CDA3101

Single Cycle Datapath (Part 1)

Welcome!

Datapath Design

ISA determines many aspects of implementation.

Implementation strategies affect clock rate and CPI.

Pedagogic ARM Language
(subset to illustrate implementation)

Memory-Reference Instructions

Load register (LDUR)
Store register (STUR)

Integer Arithmetic and Logical Instructions

ADD, SUB, AND, ORR

Branch Instructions

Compare and branch on zero (CBZ)

Logic Design Review

Combinational Elements

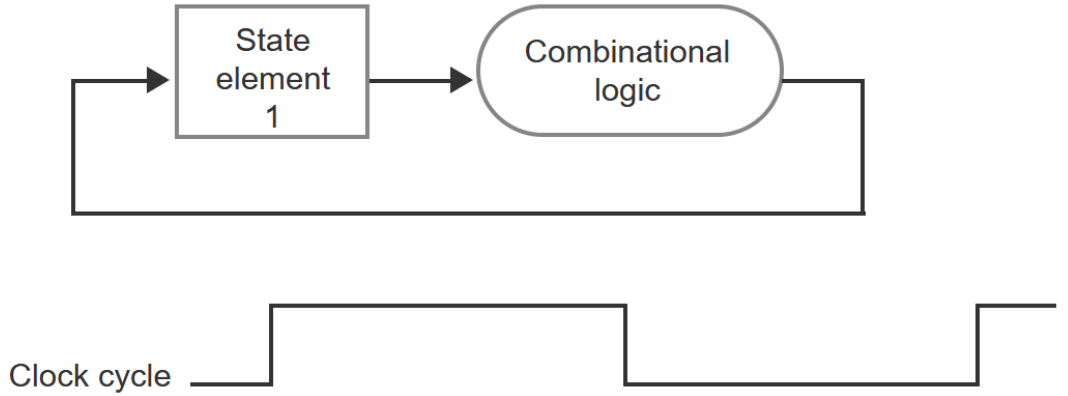
- Gates, combinations of gates (e.g., ALU)
- Outputs depend only on inputs

State Elements

- Memory / storage
- Data read from register is state value at previous clock cycle
- Data written to register will be the new state at the next clock cycle

Datapath

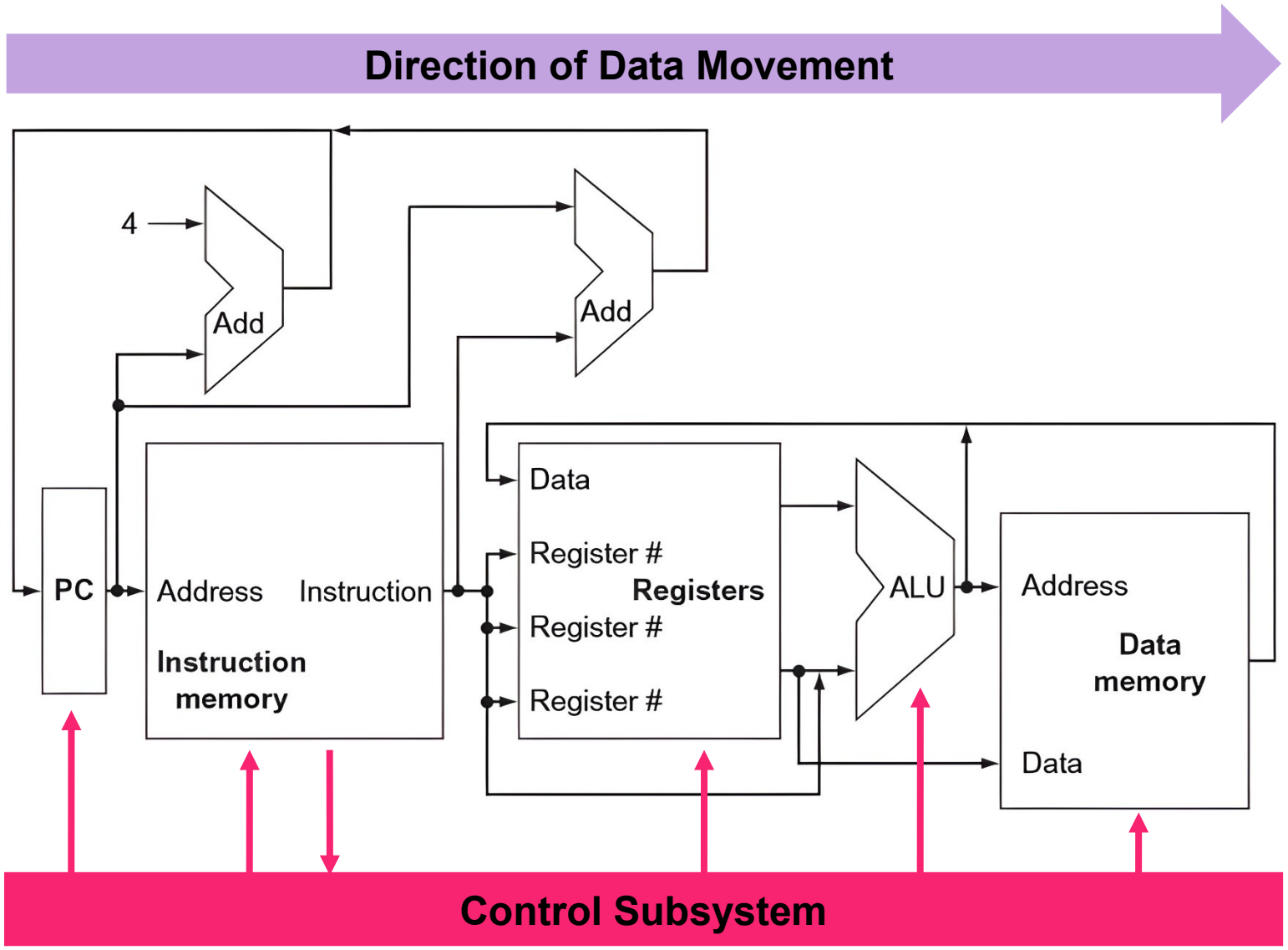
- Reads an instruction
- Executes instruction
- Writes registers



ARM Datapath

Datapath is based on register transfers required to execute instructions.

Control causes the correct actions to happen at the correct time.

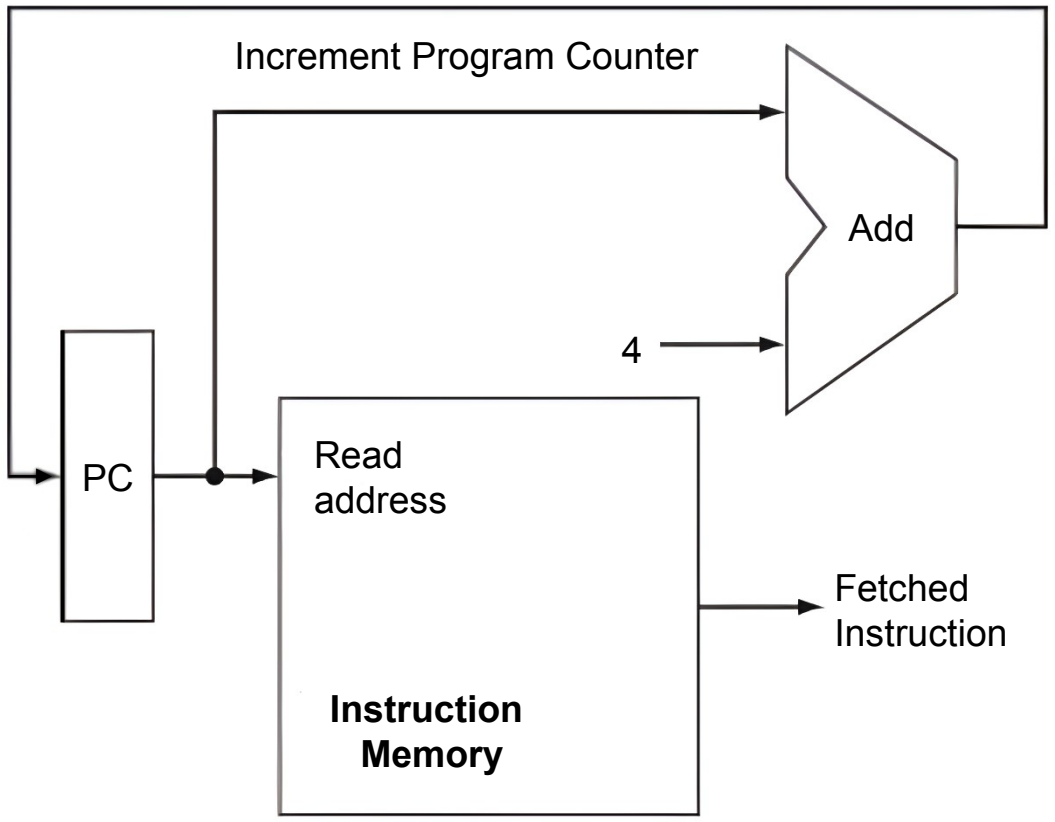


Instruction Fetch

PC stores address of current instruction.

Instruction Memory stores the instruction.

ALU / ADDER calculates the next instruction's address.

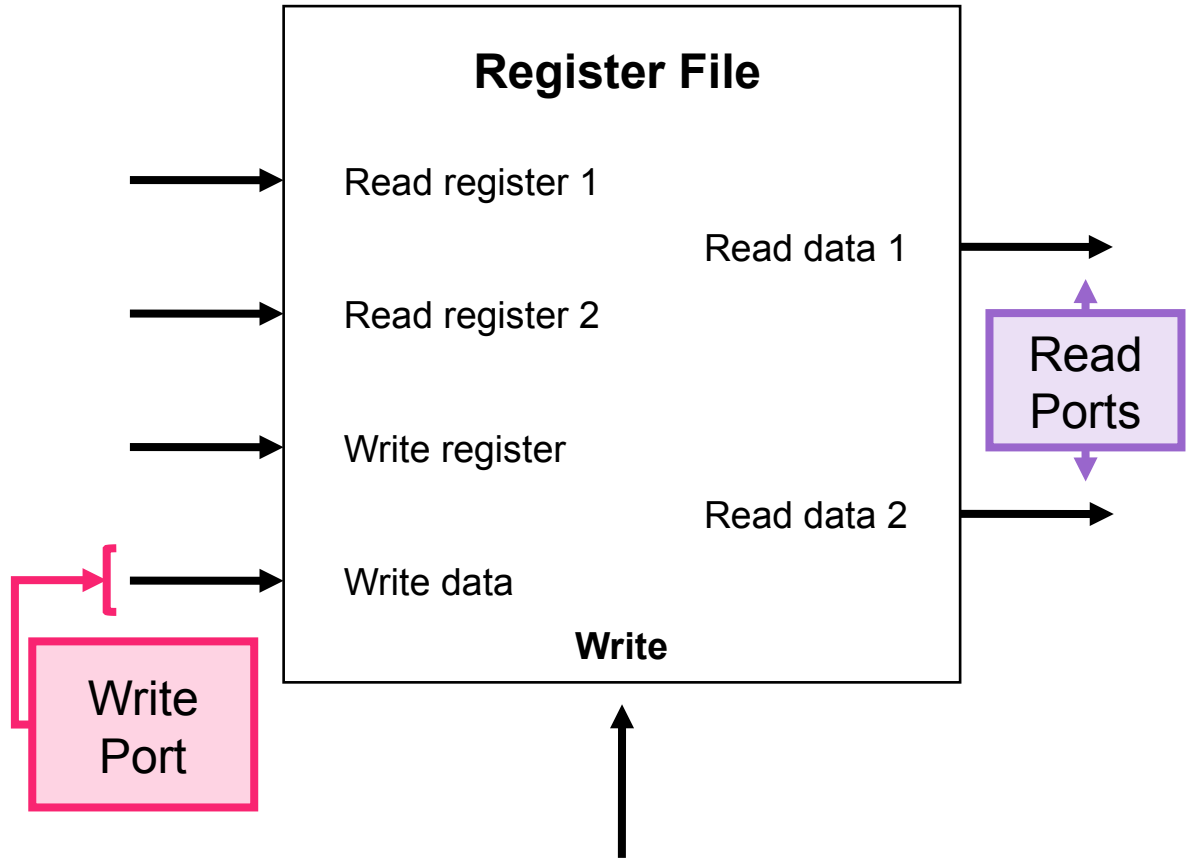


Register File

The register numbers are taken from the instruction and sent to the register file.

The data to be provided is sent through the read ports.

There is a signal with the register number to be written, and a write port through which the data will be sent.



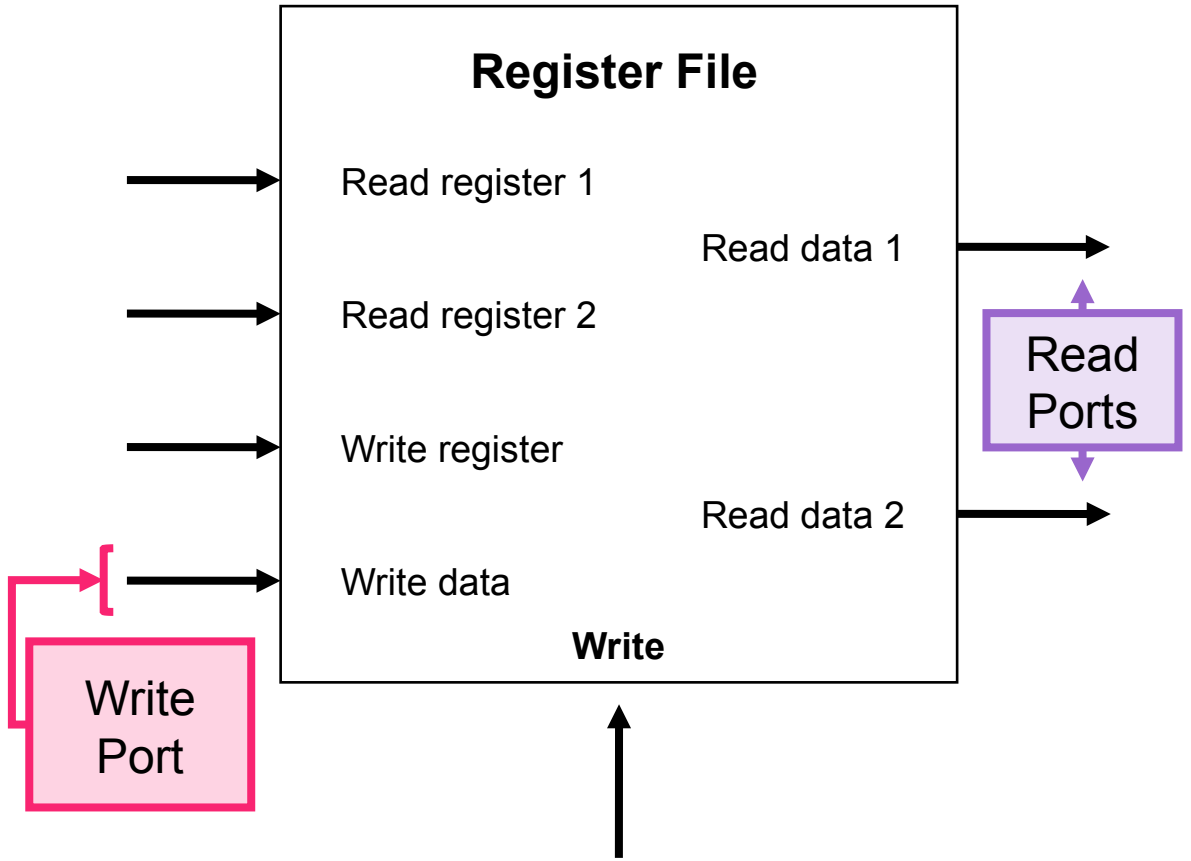
Register File

Read Data

State of register in the previous clock cycle

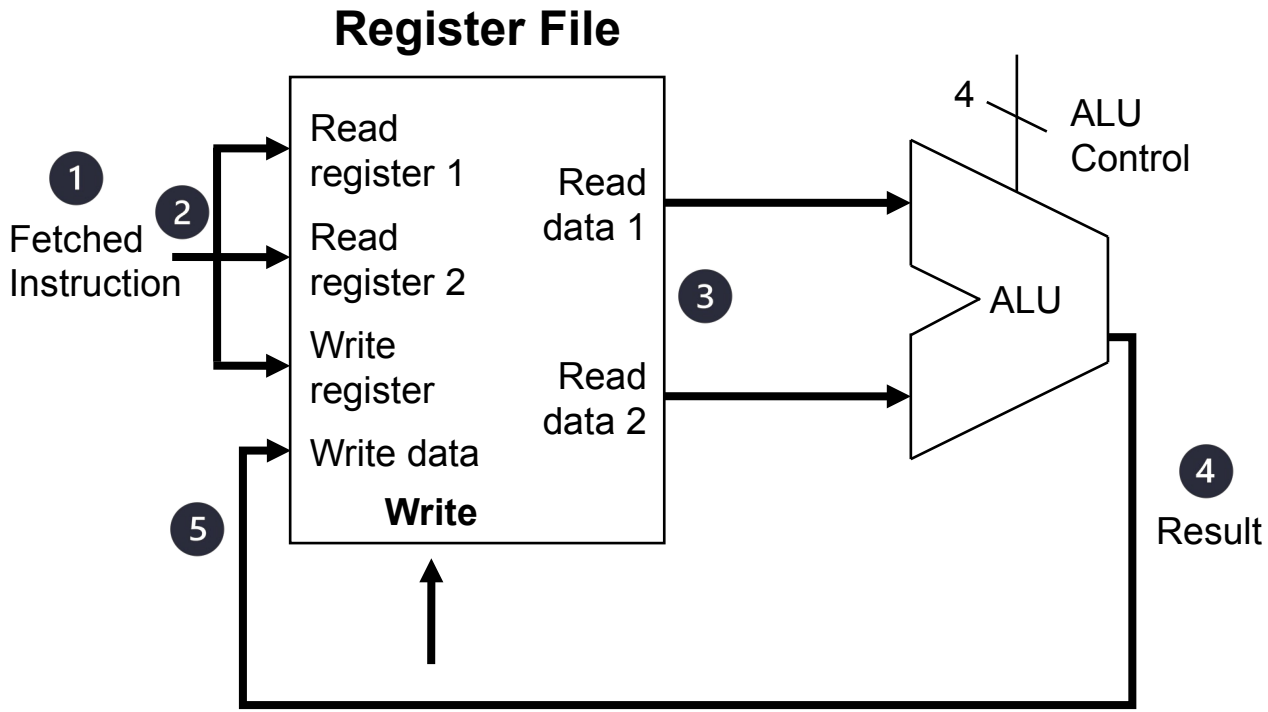
Write Data

Changed for the next clock cycle



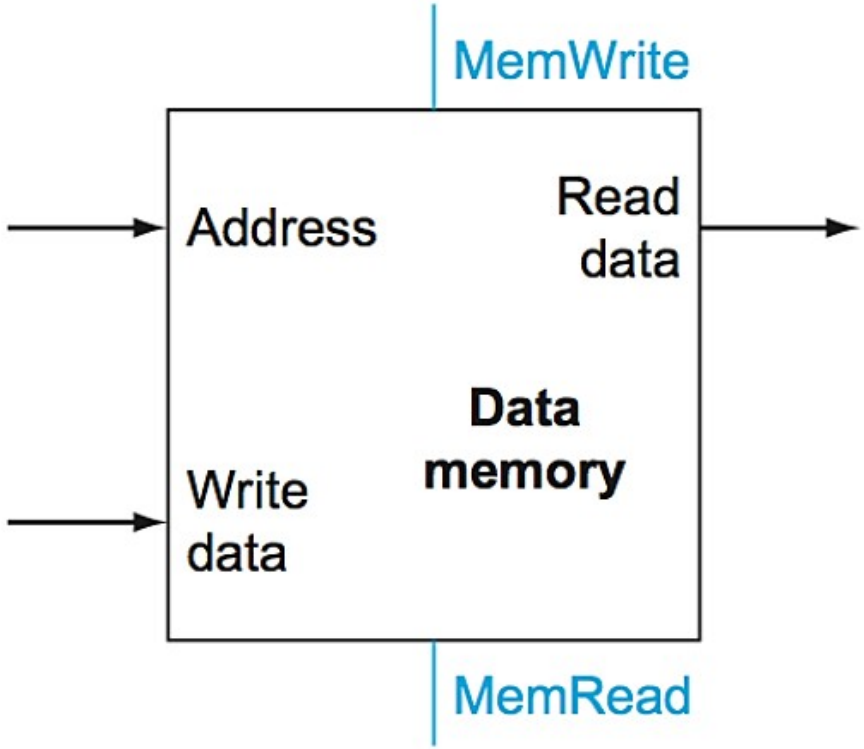
Datapath Elements for R-Format

1. Instruction is read from Instruction Memory.
2. Instruction is split by Decoder-Logic into opcode, rs, rt, and rd fields.
3. Register contents for rs and rt are read from Register File.
4. ALU computes using rs & rt registers and produces Result.
5. Result is written to Register File at WriteData port.



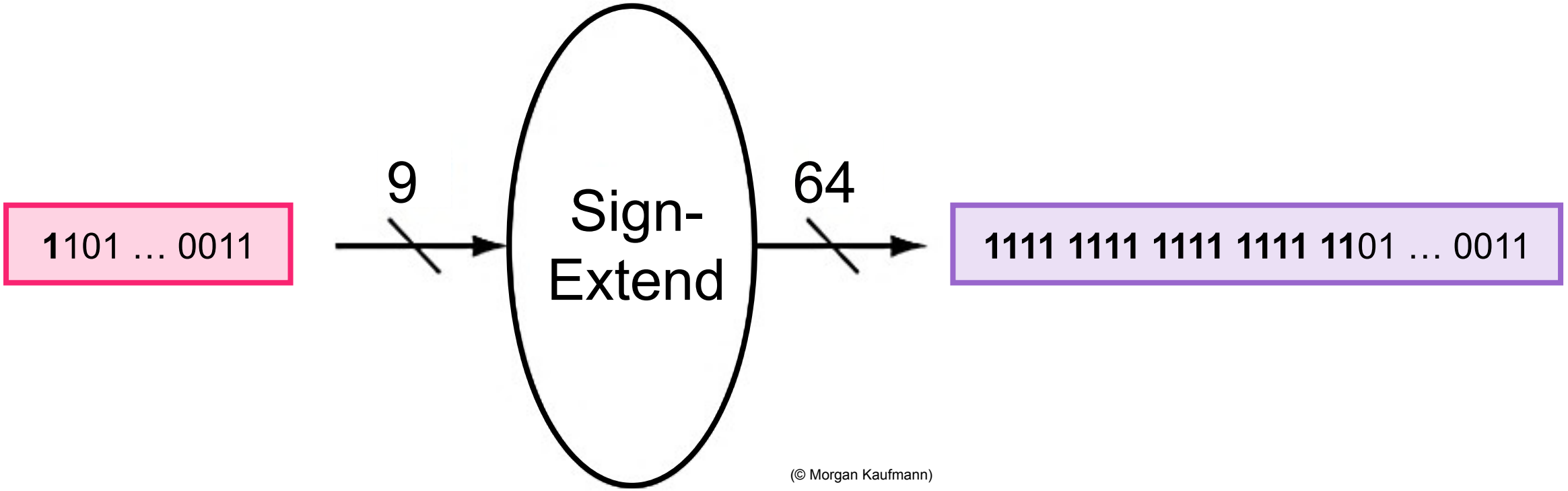
Datapath Elements for D-Format

Data Memory is part of Main Memory or Cache Memory.



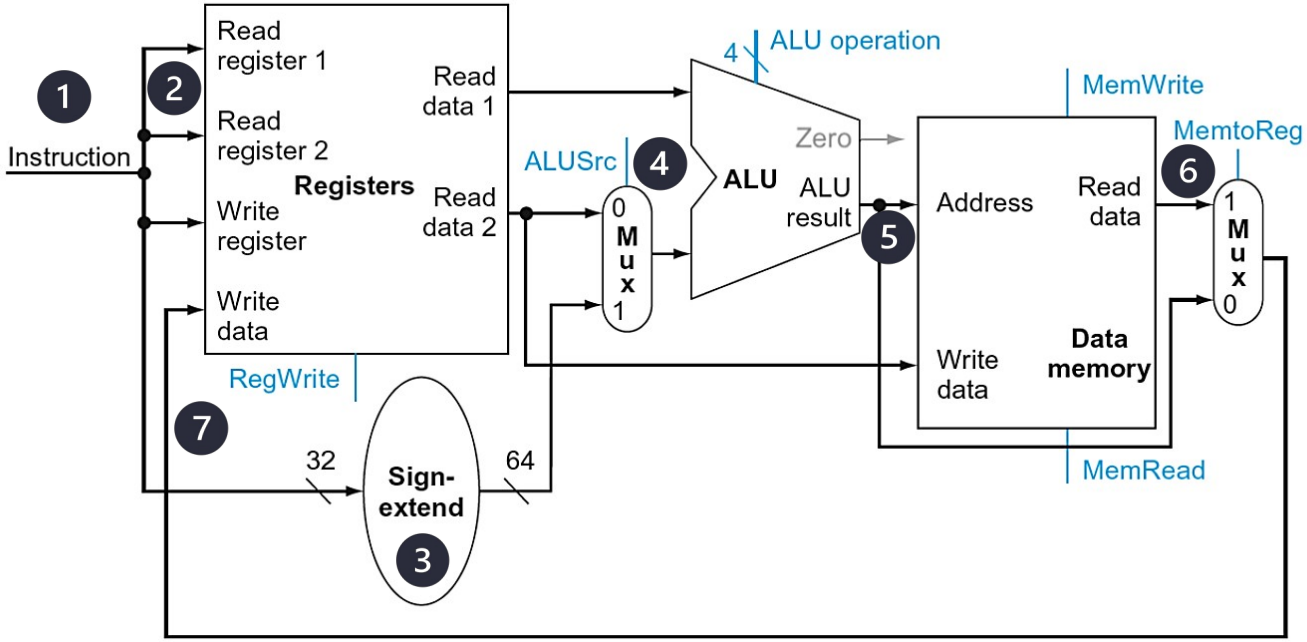
Datapath Elements for D-Format

Sign Extender replicates sign bit of 9-bit input to form a 64-bit result.



Datapath Elements for D-Format

1	Instruction is read from Instruction Memory.
2	Instruction is split by Decoder-Logic into opcode, rs, offset and base.
3	Offset value is sent to Sign Extension.
4	Register-Read output is sent to Read Data 1 and 2.
5	ALU computes memory address.
6	ALU output applied to data memory.
7	**LDUR only**: Data Memory output written to Register File.



Datapath Elements for CB-Format

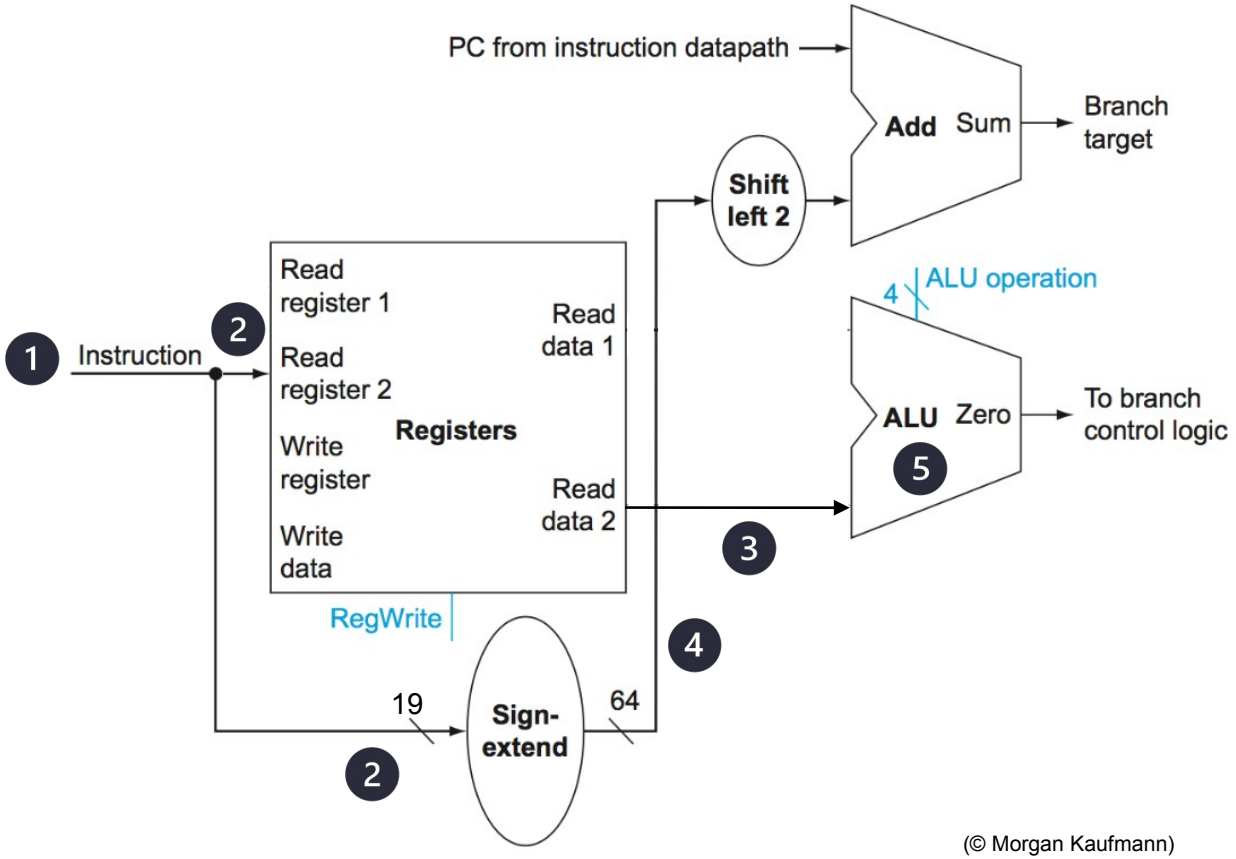
1 Instruction is read from Instruction Memory.

2 Instruction is split by Decoder-Logic into opcode, rs, rt, and dest fields.

3 Register contents read from Register File (output at Read Data 1 and 2).

4 ALU adds PC to sign-extended shifted offset to produce BTA.

5 ALU compares rs and rt registers and calculates Zero output (valued 1 or 0)

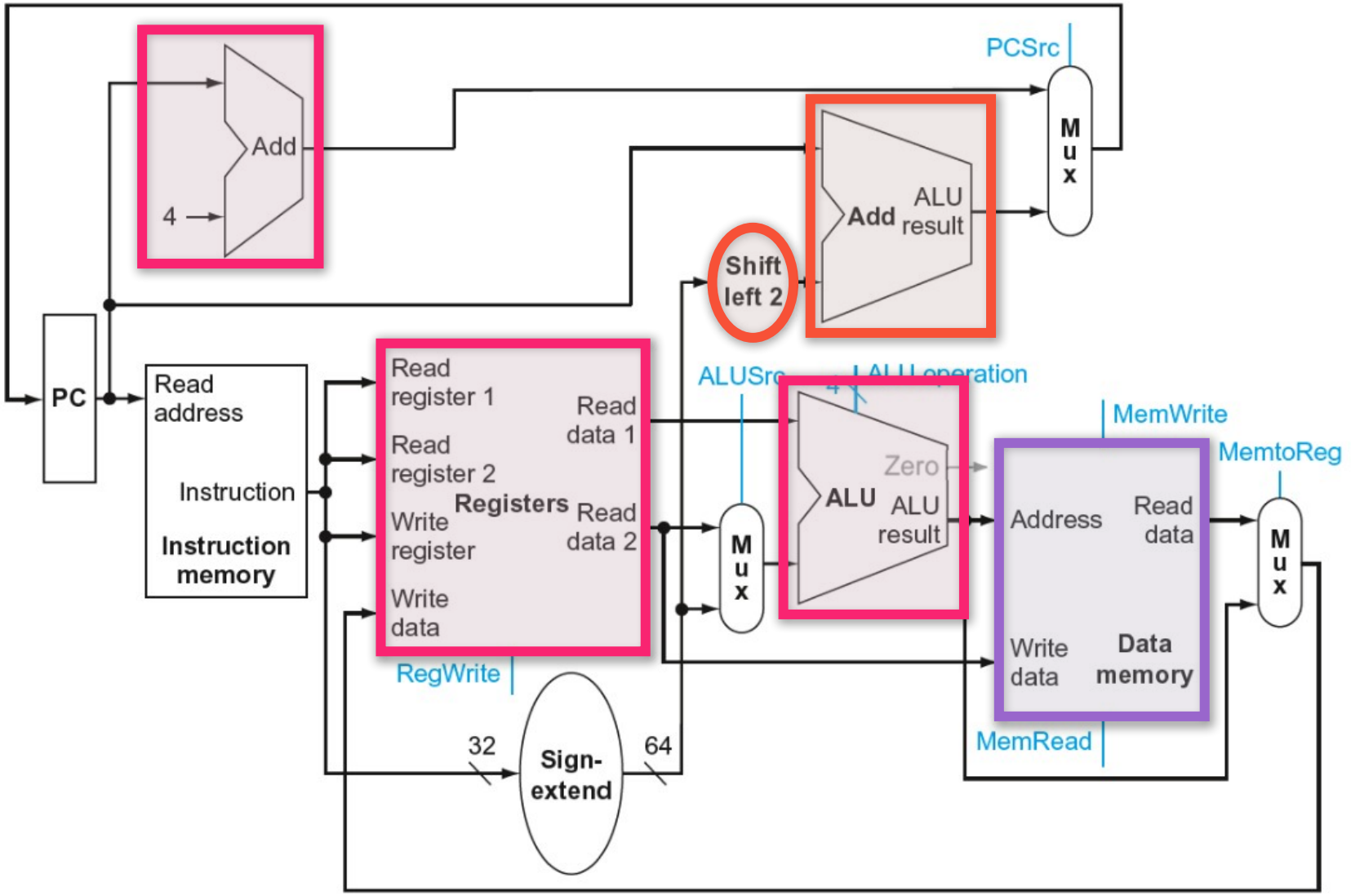


Putting It All Together

All Instructions

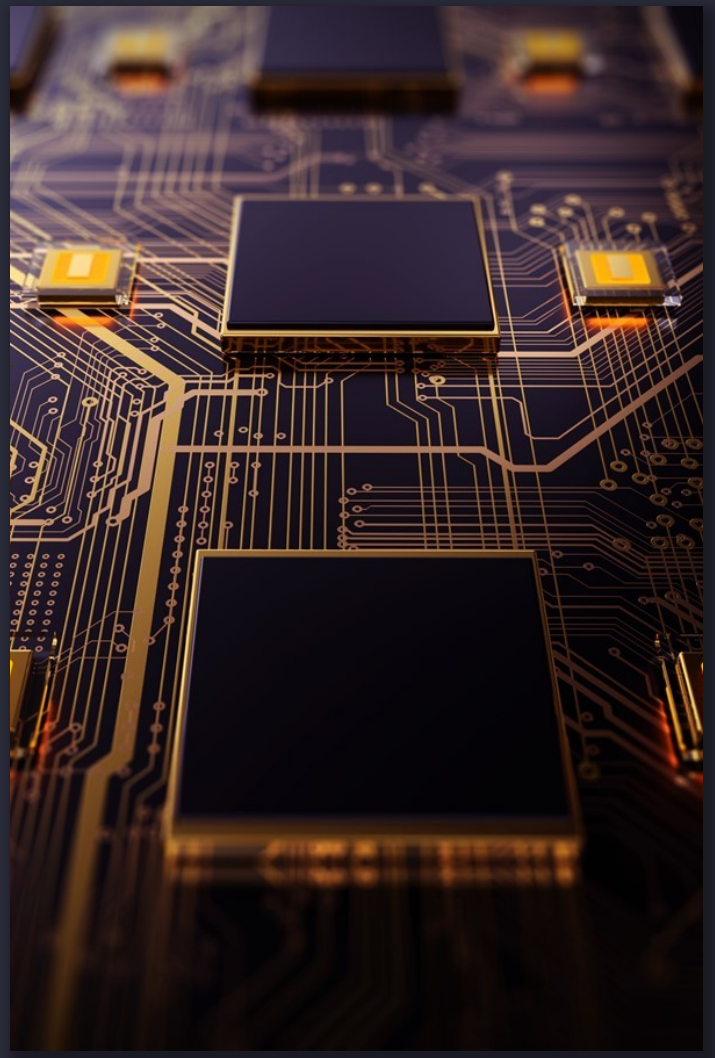
D-Format

CB-Format



(© Morgan Kaufmann)

Wrap Up



Thank you for watching.

UF | UNIVERSITY of FLORIDA