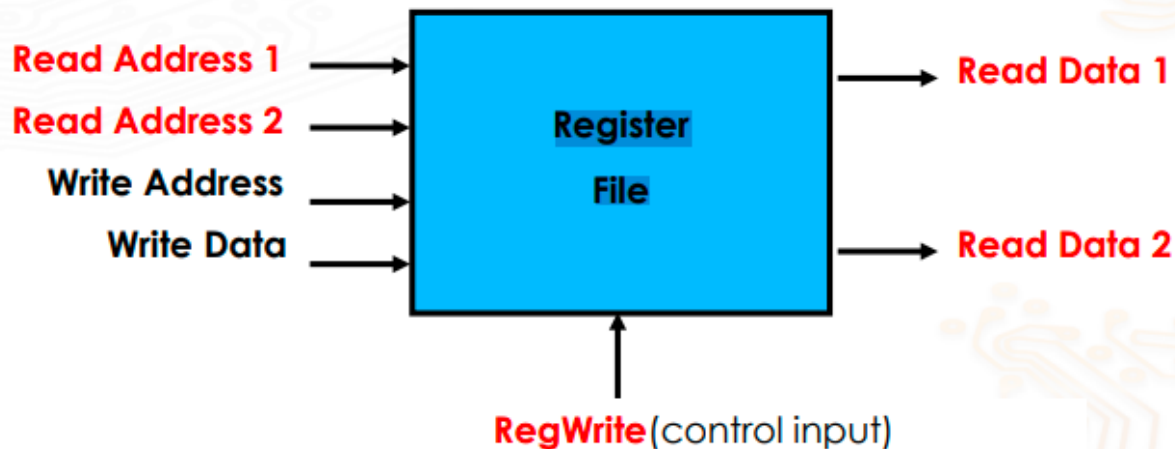


Module 3: Data path and Control Design

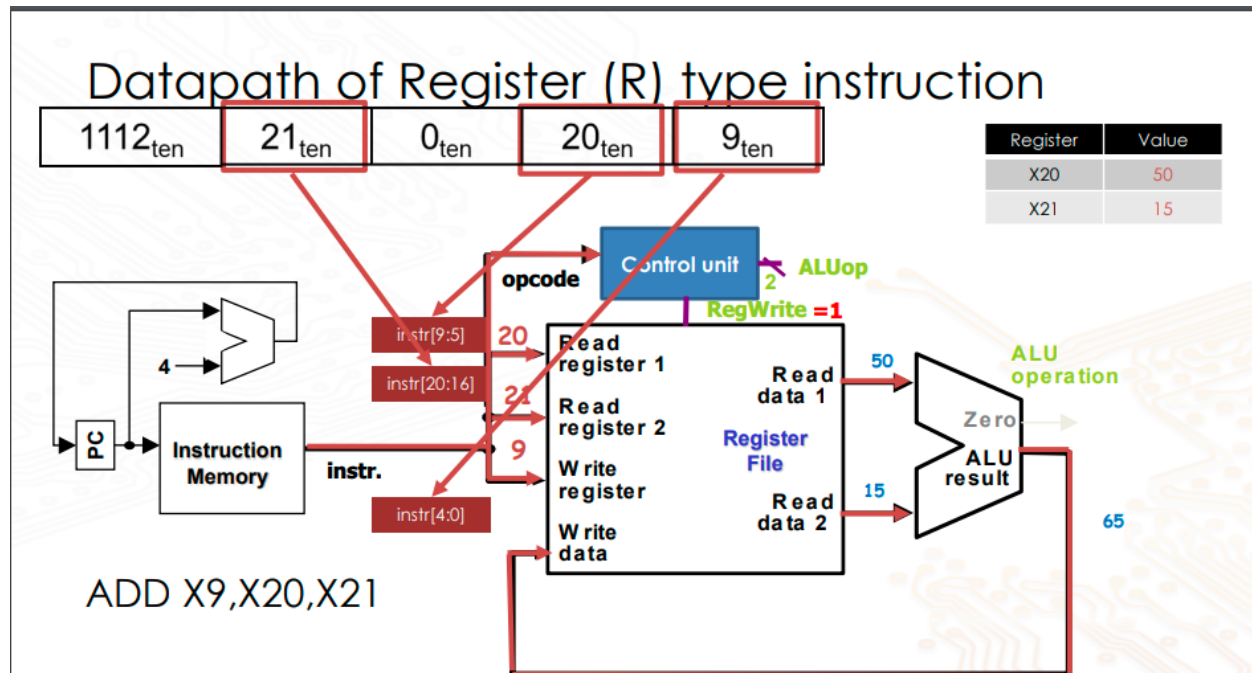
Review of basic logic devices and register file

- **ALU:** arithmetic ALU
- **Register file:** Consists of a set of registers that can be read and written by supplying a register to be accessed,



Single-cycle datapath design

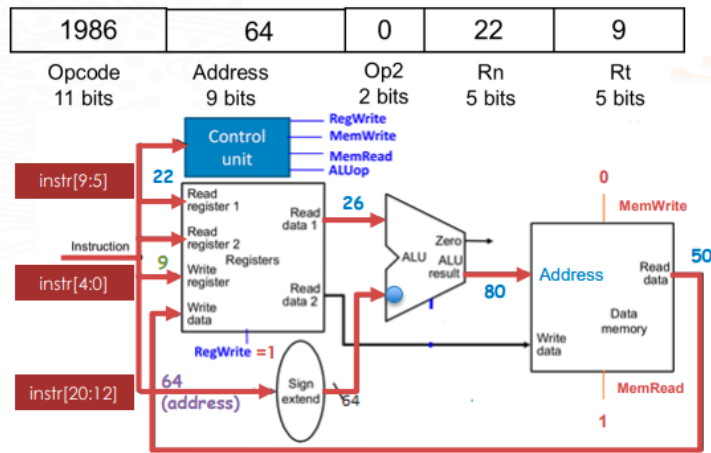
- Datapath for R-type instructions



- Datapath for D-type instructions

Data transfer (D)type – Base addressing Part 3/4)

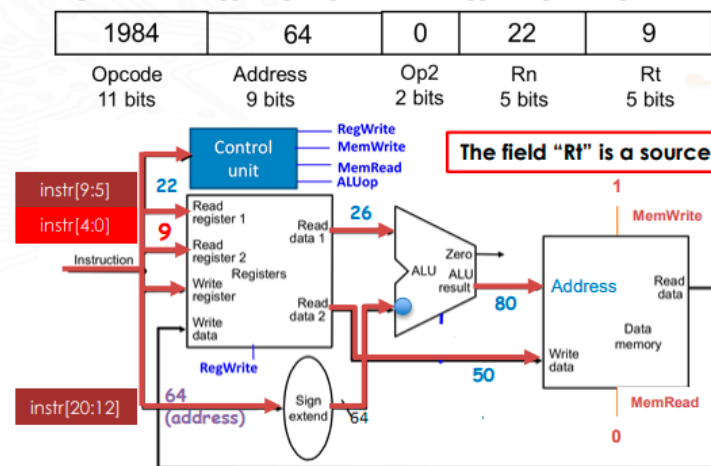
- Operation **LDUR Rt, [Rn, #address]**
- LDUR X9, [X22, #64] $[[X9] \leftarrow \text{mem}[[X22] + 64]$



Register	Value
X22	26
Memory[80]	
	50

Data transfer (D)type – Base addressing Part 4/4)

- Operation **STUR Rt, [Rn, #address]**
- STUR X9, [X22, #64] $[[X9] \rightarrow \text{mem}[[X22] + 64]$



Register	Value
X22	26
X9	50
Memory[80]	
	50

- Combined datapath for R and D-type instruction

Datapath for R and D-type Instructions

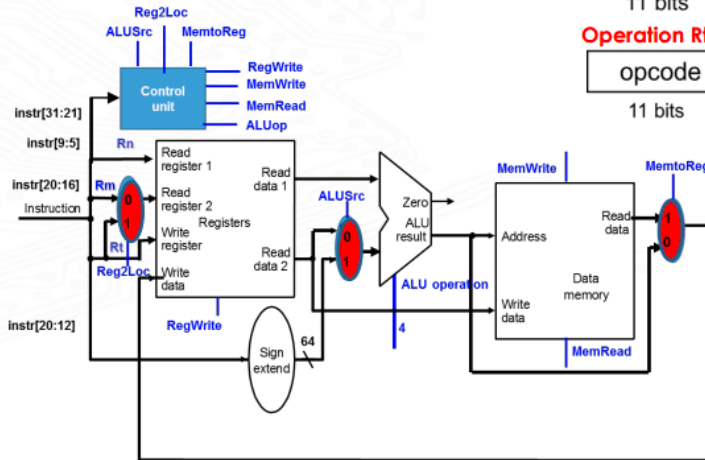
How can we combine the data path for R and D type instructions?

Operation Rd, Rn, Rm

opcode	Rm	shamt	Rn	Rd
11 bits	5 bits	6 bits	5 bits	5 bits

Operation $Rt, [Rn, address]$

opcode	address	op2	Rn	Rt
11 bits	9 bits	2 bits	5 bits	5 bits



"Rn" -Source for R and D type
 "Rd" -Destination for R-type
 "Rt" -destination/source for D-type

Extra mux needed

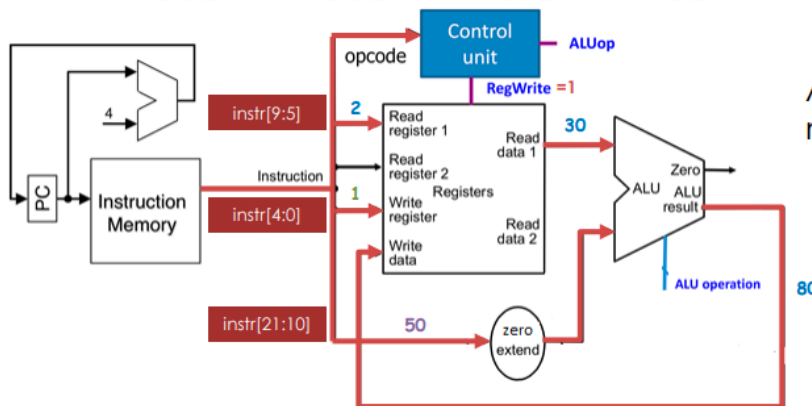
"Reg2Loc" -Selects between "Rt" and "Rm" as the source register address
 "ALUSrc" -Selects between "read data2" and "address" as the source to ALU
 "MemtoReg" -select the result from memory or from ALU

- Datapath for I-type instructions

Datapath for I type – Immediate addressing

580	50	2	1
Opcode	immediate	Rn	Rd
10 bits	12 bits	5 bits	5 bits

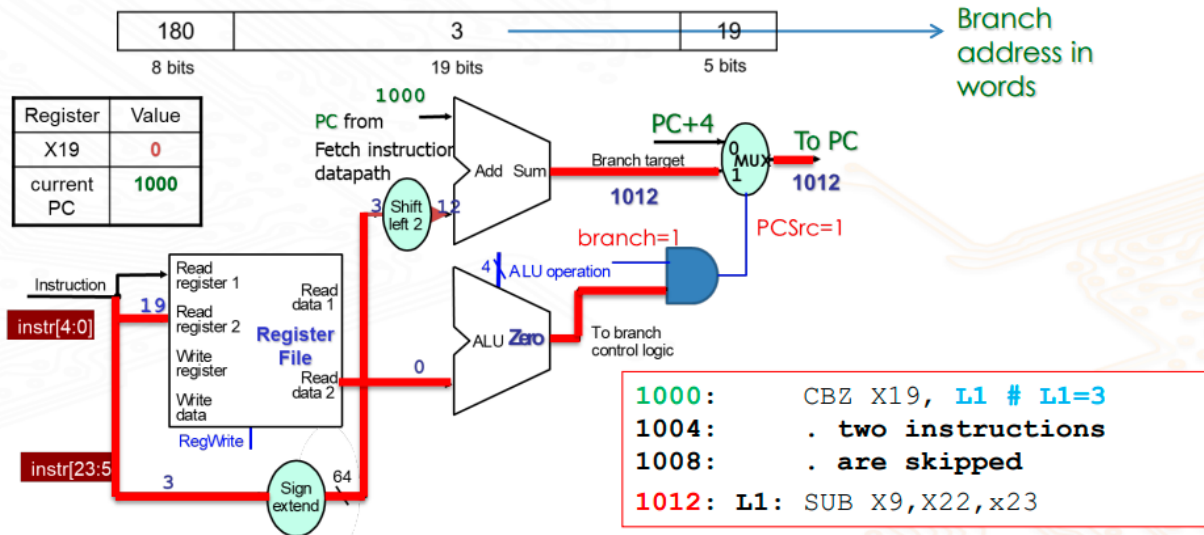
Register	Value
X2	30



ADDI X1, X2, #50
 meaning $[X1] \leftarrow [X2] + 50$

- Datapath for CB-type instructions

Datapath for CB type – Branch instruction



- Combined datapath for R,D, I, CB, and B-type instructions

Complete Datapath including B Instructions

