Chapter 05: Basic Processing Units … Control Unit Design Organization

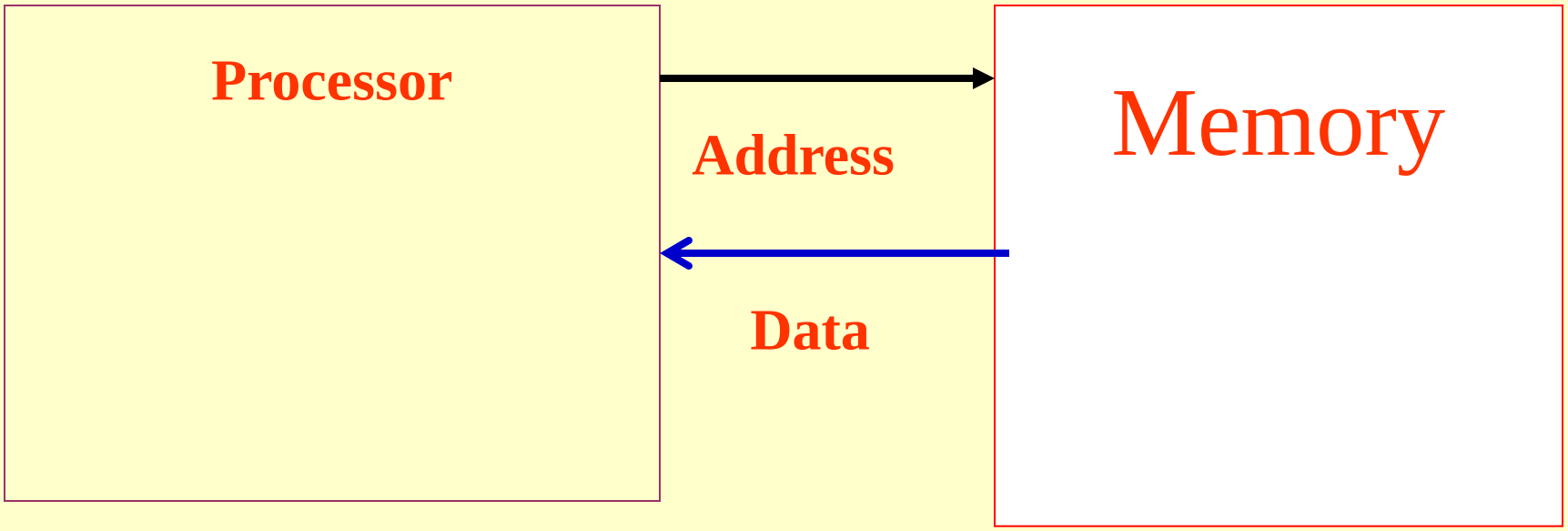
Lesson 09:
Execution of complete instruction

Objective

- Learn microoperations for complete instruction
-

Steps for completing instruction operations

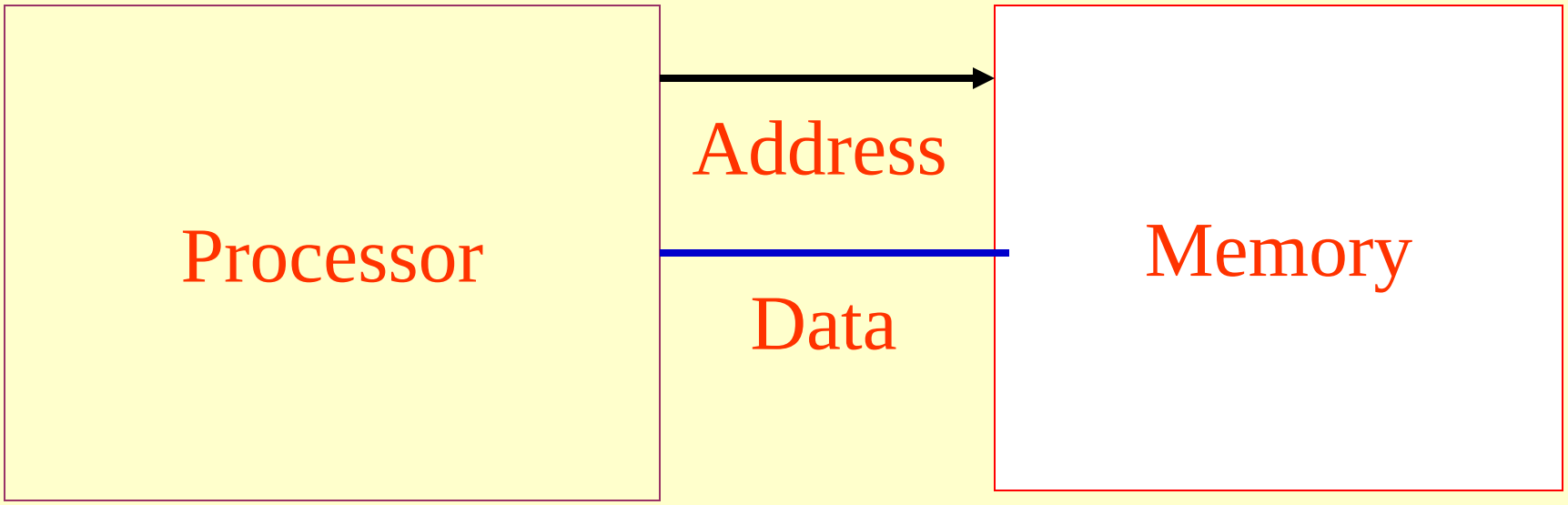
Instruction Fetch



Step 1) Processor requests instruction from memory using address in Program Counter PC register

Step 2) Memory using data bus returns the instruction to instruction register IR register

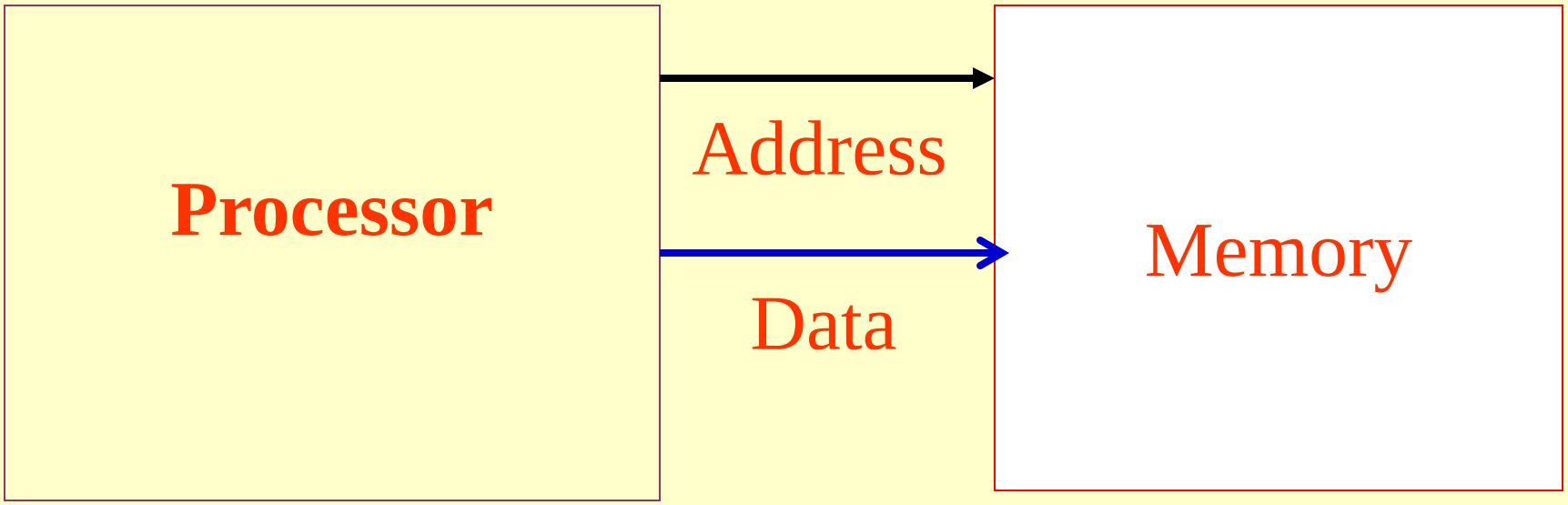
Instruction decoding and execution



Step 3) Processor decodes instruction and places at instruction decoder ID register

Step 4) Processor executes instruction at execution unit

Result Write back and PC update for the next



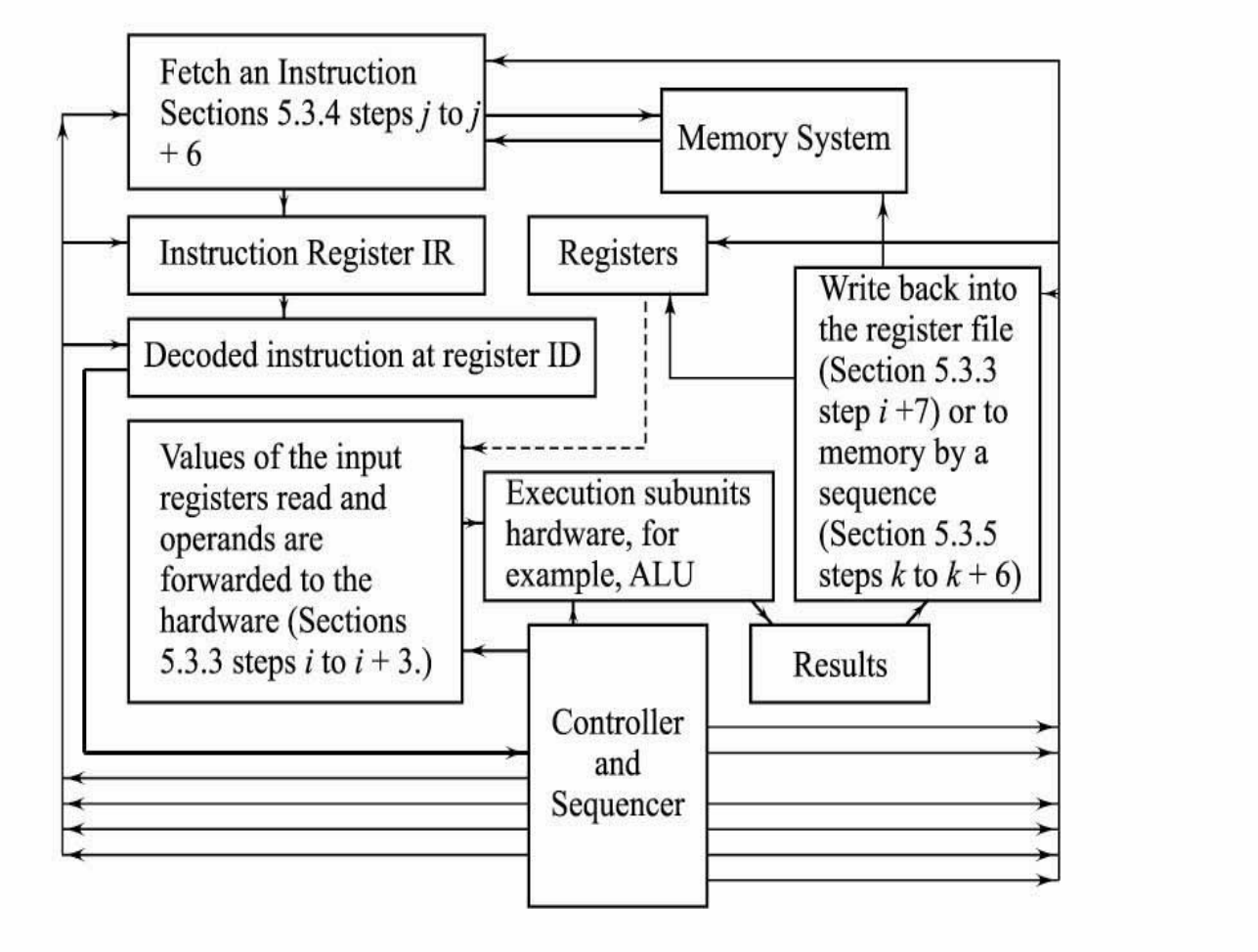
Step 5) Result of instruction written back for register or memory

Step 6) PC updates for next instruction

Execution of a Complete Instruction by Data Path Implementation

- Execution of a complete instruction can be considered as the implementation of a specific data path flow, as per the specific instruction
- The processing unit composition is as controlled data-path unit and control unit (controlling and sequencing unit)
- Control unit generates control signals to implement each step using signals ϕs

Steps involved in executing an instruction and the different modules of the processor interactions during instruction execution



Execution of Complete Instruction

(1) First, the processor fetches the instruction from the memory

- Steps j to $j + 6$— the sequence of actions involved in the *fetch* operation

Execution of Complete Instruction

(2) Second, the instruction is then *decoded* to determine what instruction it is and what its input and output registers are

- The decoded instruction is represented as a set of bit patterns that tell the hardware how to execute the instruction
- These bit patterns are sent on to the next section of the execution unit, which reads the instruction's inputs from the register file

Actions take place at the processing units

1. The decoded instruction and the values of the input registers are forwarded to the ALU hardware (Steps i to $i + 3$)
2. Hardware computes (steps $i + 4$ to $i + 6$) and gives the result and
3. The result is written back into the register file (step $i + 7$) or to memory by the sequences (steps k to $k + 6$)

Sequence of control operations required to execute the SUB (*ri*), *rj* instruction

- The instruction means subtract *rj* from memory address operand pointed by *ri* and write back the result at memory address operand pointed by *ri*
- The operation involved are the actions on the data path and control sequences

a. Fetch the instruction from memory address I pointed by PC

1. *Step j:* PC → MAR.
2. *Step j + 1:* PC ← PC + 4 for 32-bits memory word alignment.
3. *Step j + 2:* Activate signal ALE for one cycle
4. *Step j + 3:* Activate signal MEMRD
5. *Step j + 4:* M[I] → MDR
6. *Step j + 5:* Deactivate signal MEMRD.
7. *Step j + 6:* MDR → IR

a. Fetch the instruction from memory address I pointed by PC

5. *Step j + 4:* M[I] → MDR
6. *Step j + 5:* Deactivate signal MEMRD.
7. *Step j + 6:* MDR → IR

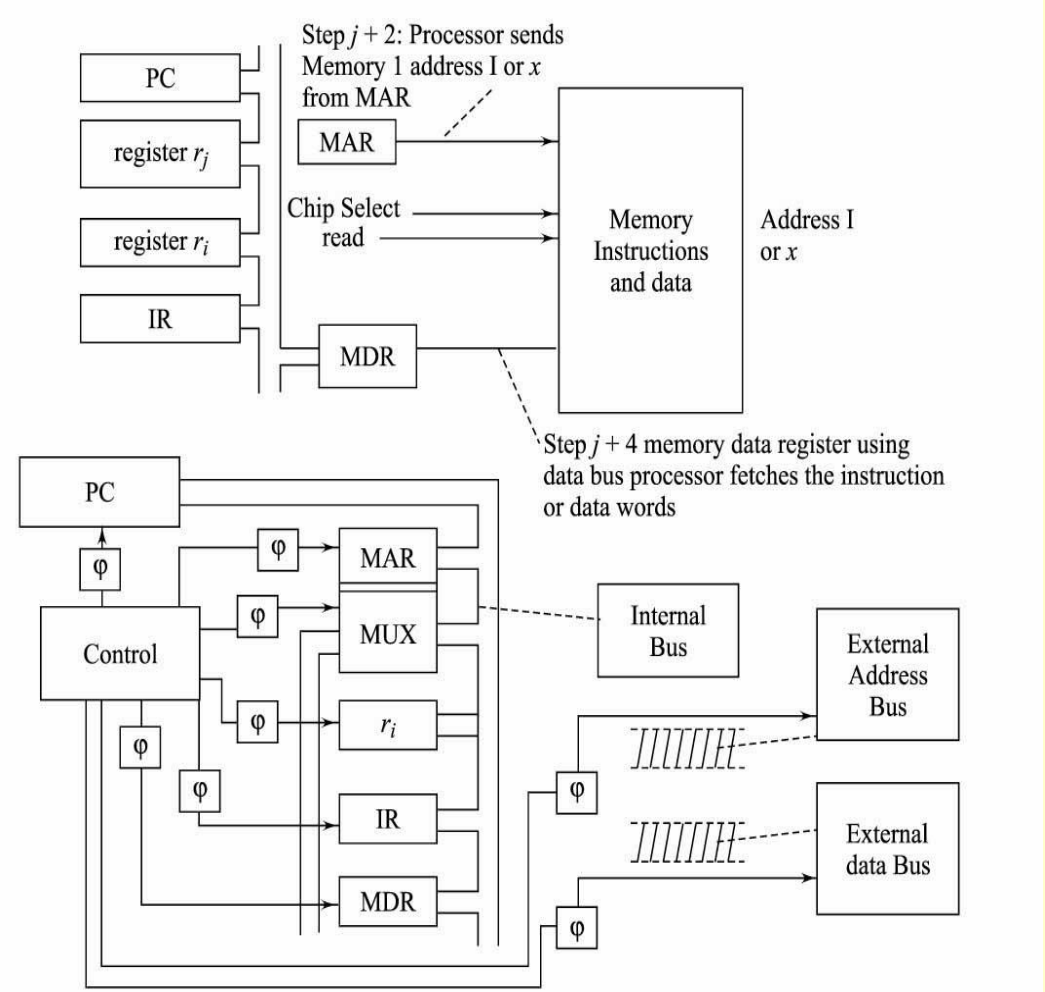
b. Read the operand from memory address *x* pointed by *ri*

1. *Step j1: ri* → MAR
2. Step *j*1 + 1: No action as PC already ready for the next instruction
3. *Step j1 + 2:* Activate signal ALE for one cycle.
4. Step *j*1 + 3: Activate signal MEMRD

b. **Read the operand from memory address *x* pointed by *ri***

5. *Step j1 + 4:* $M[x] \rightarrow$ MDR
6. *Step j1 + 5:* Deactivate signal MEMRD.
7. *Step j1 + 6:* MDR transfers through internal bus. MDR $\rightarrow$ TEMP (a temporary register for Operand *X*)

Fetch a Word from Memory and Transfer to IR or GPR or other Word Storing Unit



c. Perform Arithmetic operations on the operands

1. *Step i:* TEMP → X.
2. *Step i + 1: X → ALU*
3. *Step i + 2: rj* → Y
4. *Step i + 3:* Y → ALU
5. *Step i + 4: Selects through a gates* ϕ_i *an operation for SUB at the ID for instruction received at IR during* **a** *steps j to j + 6*

c. Perform Arithmetic operations on the operands

6. *Step i + 5:* (Z) ← ALU.
7. *Step i + 6:* Transfers status flags generated; borrow and overflow to status register. (Status Register) ← ALU
8. *Step i + 7:* TEMP ← Z

d. Write back the TEMP to the memory address pointed by r_i

1. *Step k:* address at *ri* →MAR.
2. *Step k + 1:* Activate ALE
3. *Step k + 2:* Activate signal MEMWR.
4. *Step k + 3:* TEMP → MDR
5. *Step k + 4:* MDR → data to address at *ri*.
6. *Step k + 5:* Deactivate signal MEMWR.

Summary

We Learnt

- Instruction steps in completing all microoperations
- Fetch instruction
- Read operands
- Execute instruction at ALU
- Store the result back in memory

We Learnt

- Execution of a complete instruction can be considered as the implementation of a specific data path flow, as per the specific instruction
- The processing unit composition is as controlled data-path unit and control unit (controlling and sequencing unit)
- Control unit generates control signals to implement each step using signals ϕs

End of Lesson 09 on

Execution of complete instruction