Chapter 05: Basic Processing Units … Control Unit Design Organization

Lesson 08:

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

Objective

- Learn microoperations for fetch of a Word from Memory and transfer to IR or GPR or other Word Storing Unit by sequences of microoperations
- Learn how a Word from processor transfer and store into Memory by sequences of microoperations

Fetch operation

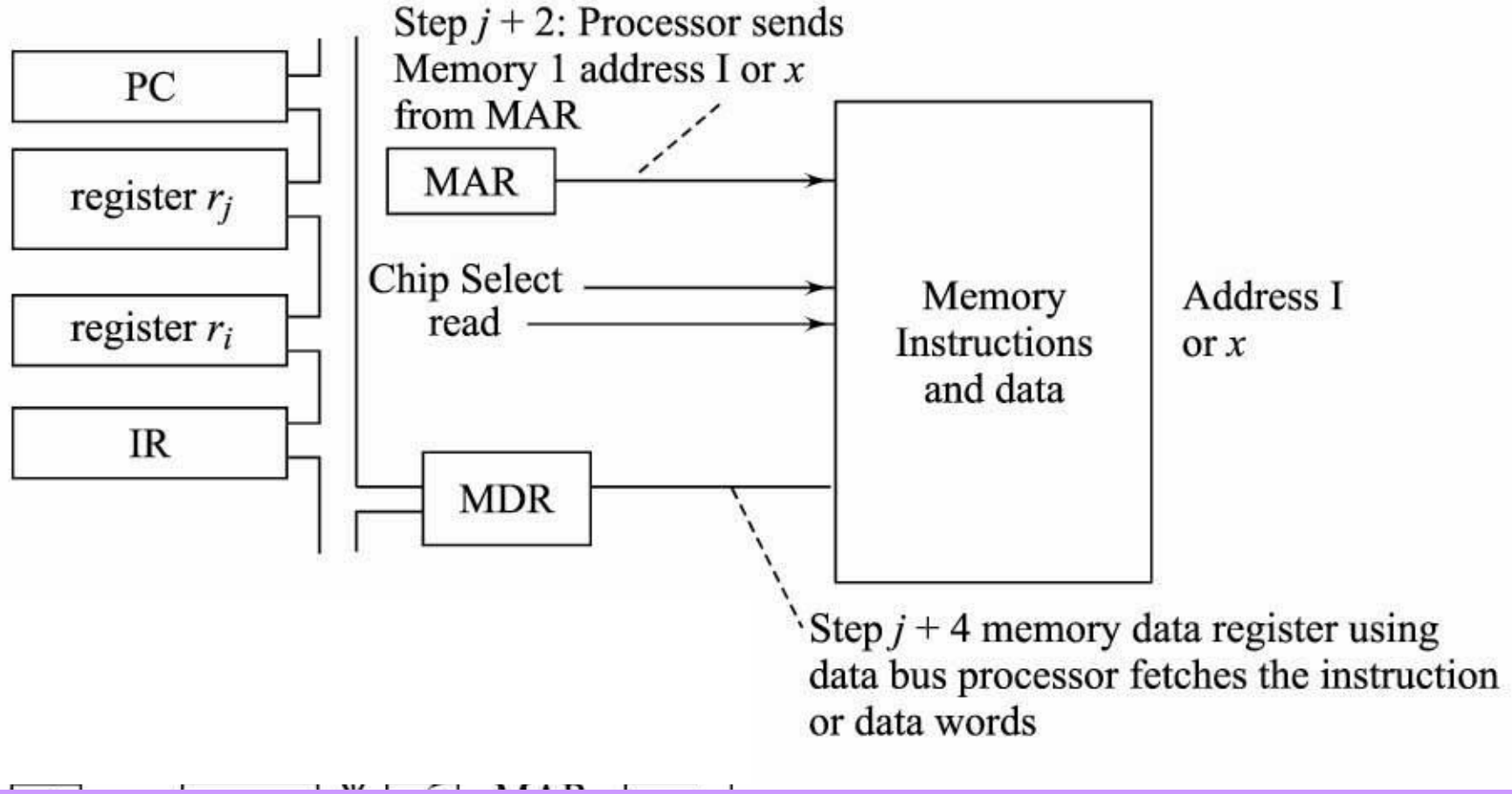
Execution of a fetch or store instruction by Data Path Implementation

- Execution of a fetch or store 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

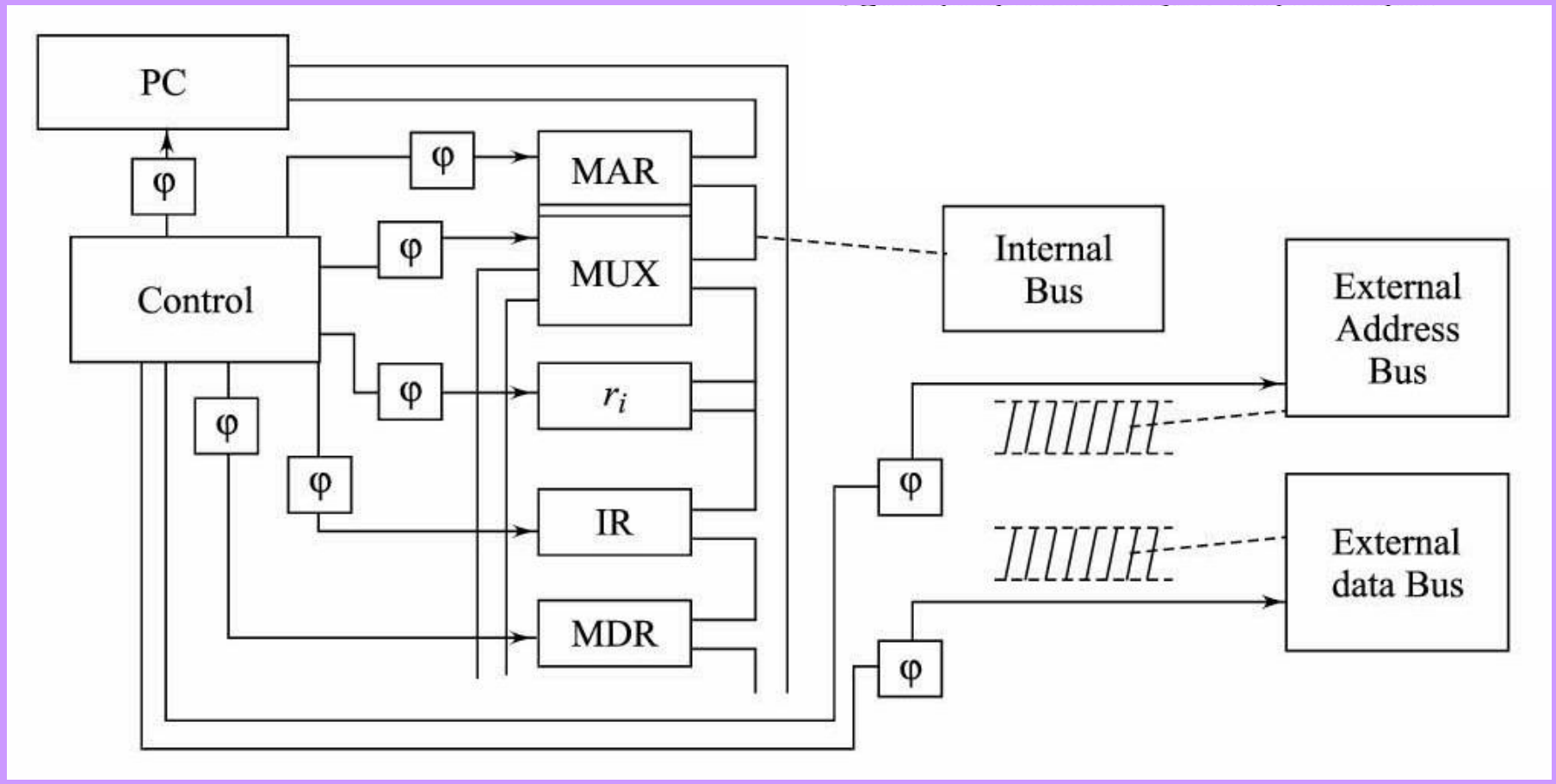
Sequence of actions to define the controlled transfers of data between processing subunits

- Processing subunits─ registers, PC, IR, MAR, MDR along a required data path
- Bus selected as per one data path among several paths

Fetch of a word from memory into IR or GPR or other word unit by data path control



Data path control using bus, gates and control circuit



Microoperations for fetch opcode and operands (word) from memory to the required subunit

- ID decoded by decoding logic
- Then the logic results at register instruction decoder (ID) initiate control actions
- Each control signal selects an action through a gate input ϕ at each step

Microoperations for executing the instruction

1. Step j: PC transfers to MAR through internal bus. (MAR means a memory address register for sending address bits to the address bus)

   PC → MAR

Microoperations for executing the instruction

2. *Step j* + 1: If this is also an instruction for the last byte fetch operation, then increment PC to make it ready for the next instruction, which will execute after this instruction completes
   For instruction last byte fetch case, PC → PC + 4 for 32-bit memory-word alignments

Microoperations for executing the instruction

3. *Step j* + 2: Activate signal address latch enable for one cycle ─ ALE: MAR → (Address_Bus)

- A chip select then activates the memory and ALE deactivates after the cycle
- ALE needed when address and data bus bits multiplex
- After the cycle, the same signals will carry data signals
- The required chip select from an external decoder that selects the targeted memory chip among several ones

Microoperations for executing the instruction

4. *Step j* + 3: Activate signal for memory for a memory read (MEMRD) operation

Microoperations for executing the instruction

5. *Step j* + 4: Memory transfers opcode or operand to MDR through data bus

- MDR─ memory data register for input-output to data bus
- M[I] or M(x)→ MDR depending on whether I or x was put in MDR in a step

Microoperations for executing the instruction

6. *Step j* + 5: Deactivate signal MEMRD

Microoperations for executing the instruction

7. *Step j* + 6: MDR transfers through internal bus

- MDR → IR, GPR, or another word storing-unit through bus or a MUX
- Repeat the steps j to j + 6 if instruction operands fetch not complete

An opcode or operands read operation for an instruction

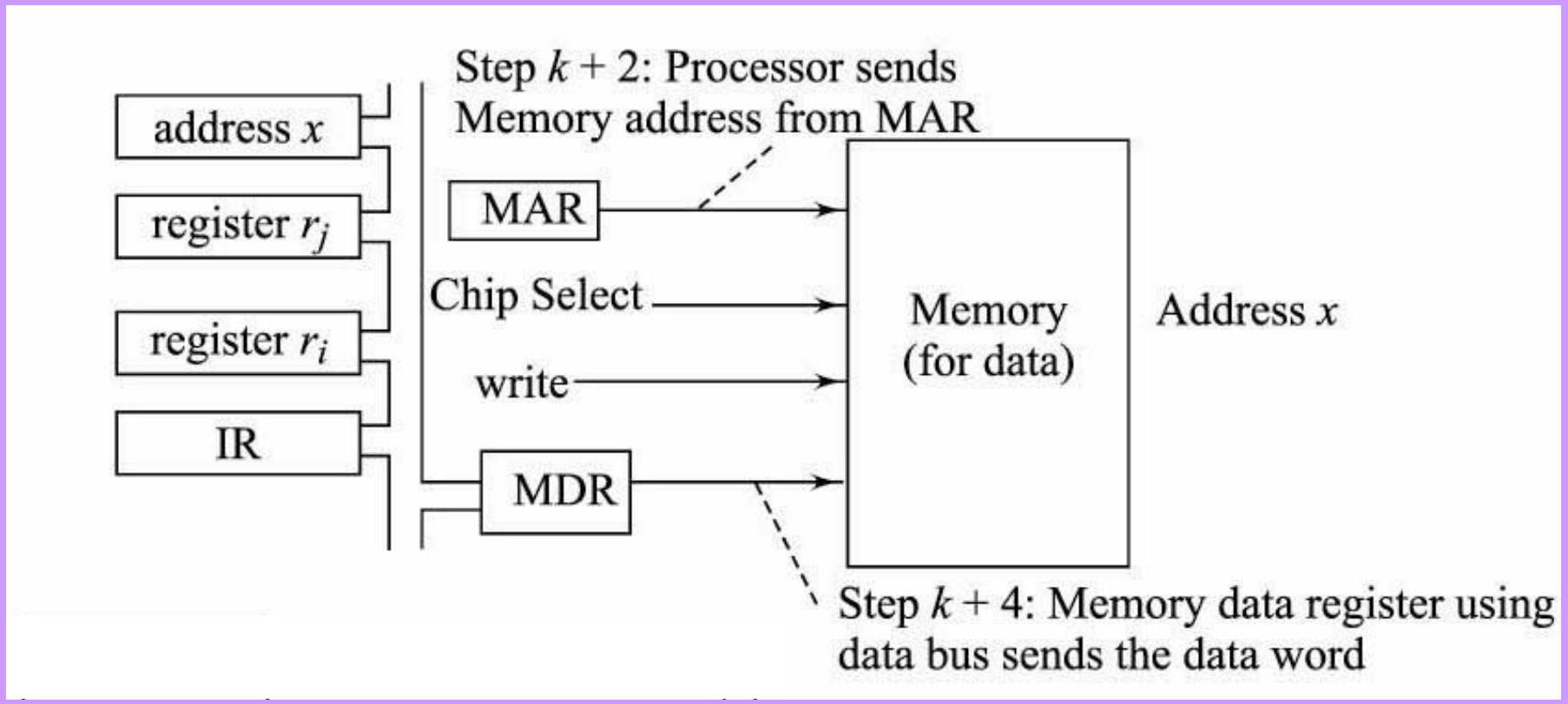
- Completes in seven or more steps
- The targeted word transfers to IR or GPR *ri* or another word storing-unit. (for example one for an immediate operand)
- The PC gets ready to fetch the next instruction in case that a cycle happens to be for the fetch instruction

ST M(*x*), *rj* instruction for Storing a Word in Memory

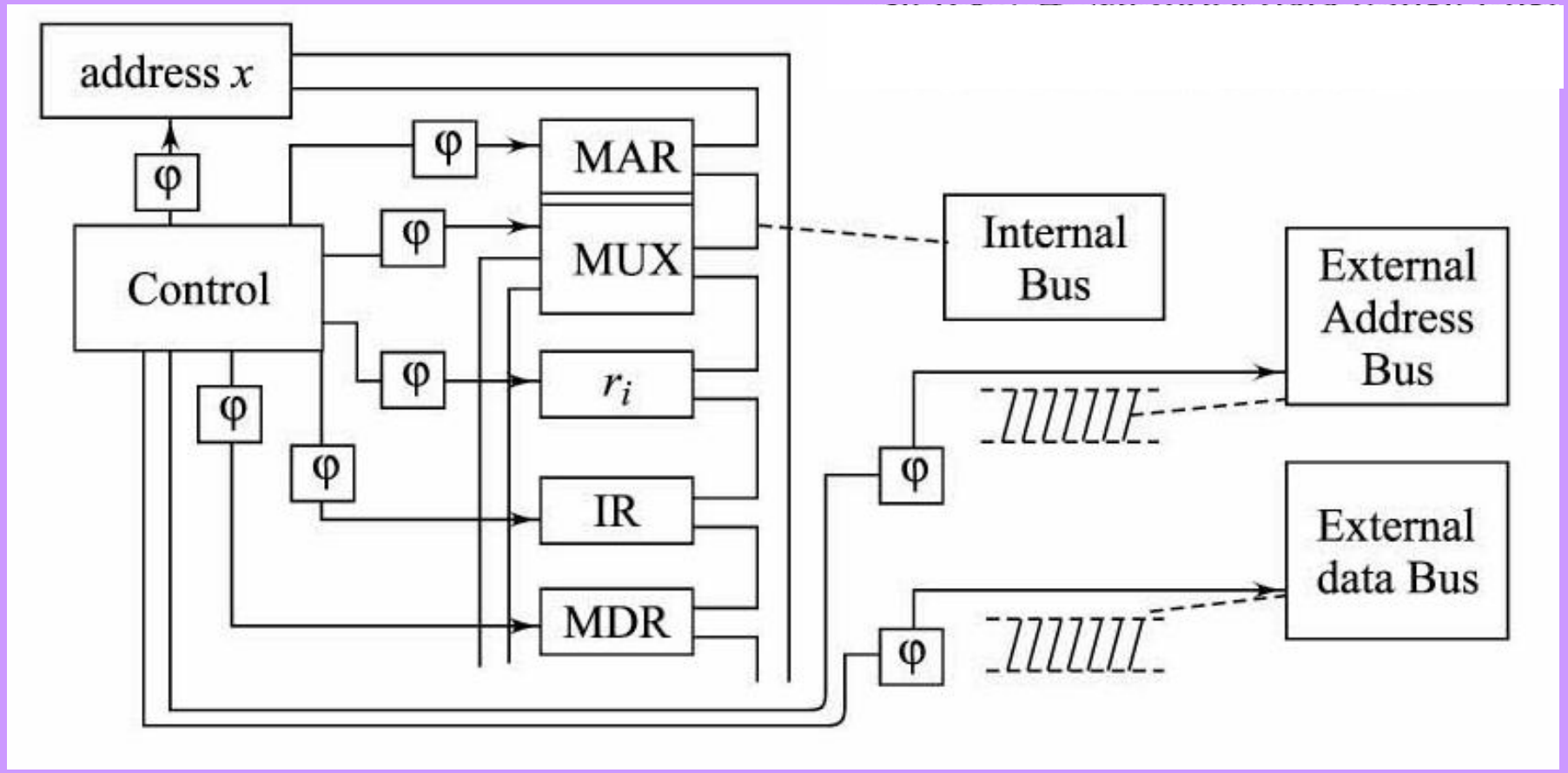
Microoperations to control the storing of data from a processing subunit to memory

- Processing subunits─ registers, PC, IR, MAR, MDR along a required data path
- Bus selected as per one data path among several paths

Storing a word into memory from GPR or other word unit by data path control



Data path control using bus, gates and control circuit



Microoperations for store operand (word) into memory from the required subunit

- ID decoded by decoding logic
- Then the logic results at register instruction decoder (ID) initiate control actions
- Each control signal selects an action through a gate input ϕ at each step

Microoperations for executing the instruction

1. *Step k:* An address x generated for example, by indirect register address, transfers to MAR through the internal bus.

- MAR─ a memory address register for sending address bits to the address bus
- The address $x \rightarrow$ MAR

Microoperations for executing the instruction

2. *Step k + 1:* Activate signal for address latch enable (ALE) for a clock cycle and to enable MAR to bus so that MAR bits carry the address bus signals to memory through the latch

- The chip select then activates at memory and ALE deactivates after one cycle
- The ALE needed when address and data bus bits multiplex
- The required chip select is from an external decoder that selects the targeted data memory chip among several ones

Microoperations for executing the instruction

3. *Step k + 2:* Activate signal for memory for a memory write (MEMWR) operation

Microoperations for executing the instruction

4. *Step k + 3:* Transfer output operand data for memory to MDR through internal bus

- MDR memory data register for input-output to data bus
- Output operand → MDR

Microoperations for executing the instruction

5. *Step k + 4:* MDR transfers. MDR $\rightarrow$ M[*x*]

Microoperations for executing the instruction

6. *Step k + 5:* Deactivate signal MEMWR

An opcode or operands read operation for an instruction

- Completes in six or more steps
- The targeted word transfers from GPR *ri* or another word storing-unit
- The PC gets ready to fetch the next instruction in case that a cycle happens to be for the fetch instruction

Control Signal during an interval for a register transfer operation during selecting a fetch or store operation

Control input during an interval for a register transfer operation

- One active C_{alu} among ten ϕs for eight operations

1. ϕ_{rs0} select an ri at the bus
2. ϕ_{rs1} select an ri at the bus
3. ϕ_{rs2} select an ri at the bus
4. ϕ_{MDR} select MDR at the bus

Control input during an interval for a register transfer operation

5. ϕ_{PC} select PC at the bus
6. ϕ_{MAR} select MAR at the bus
7. ϕ_{IN} select input action from the bus
8. ϕ_{Out} select output action from the bus
9. ϕ_{IR} select IR at the bus
10. ϕ_{ID} select IR at the bus

Summary

We learnt

- A fetch or store operation performed by sequences of microoperations
- 8 control signals to select an ri among PC, MDR, MAR, IR, ID and 8 GPRs
- 2 control signals to select input or output from internal bus at the processing units

End of Lesson 08 on

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