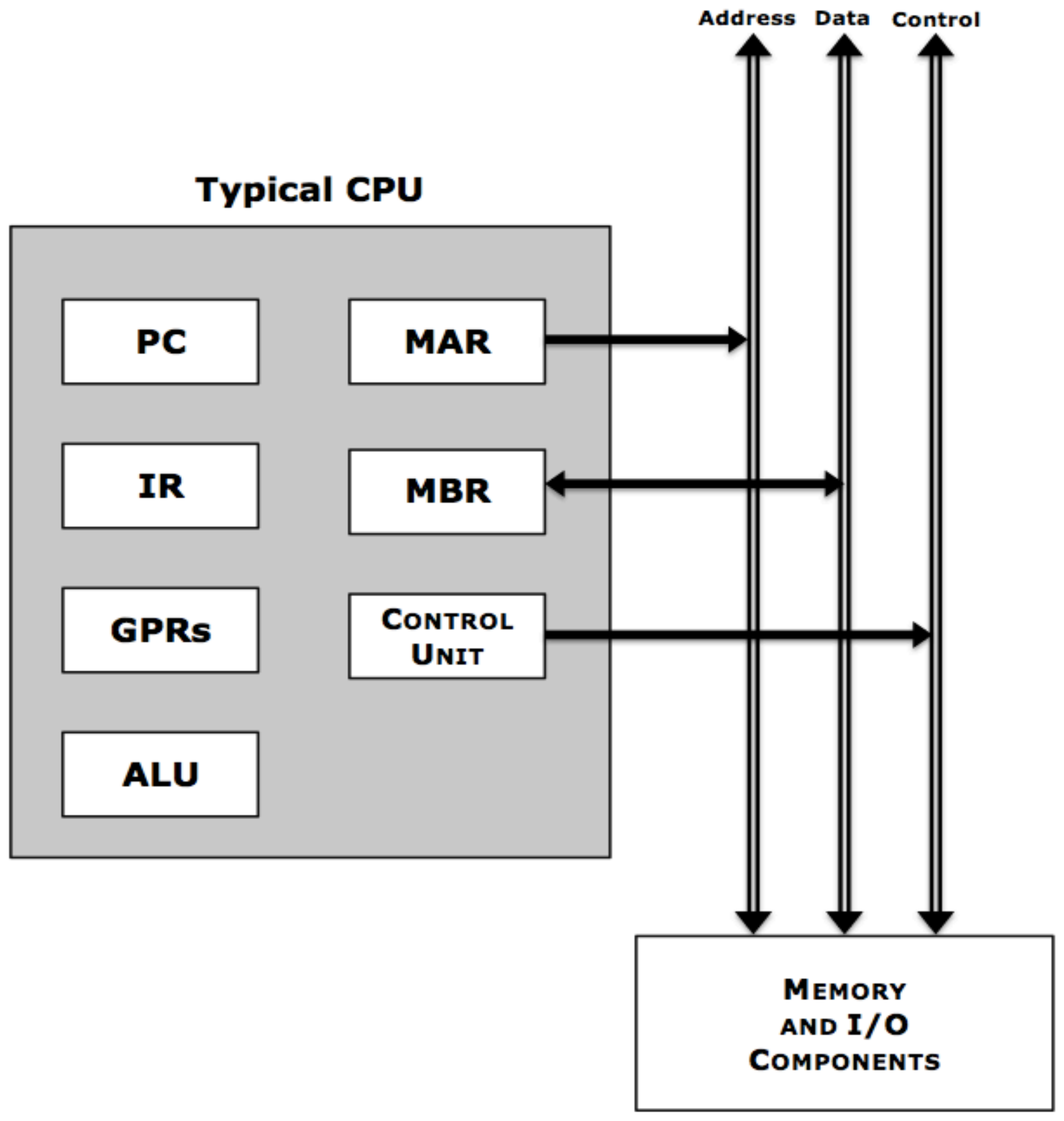
MICRO-OPERATIONS & CONTROL SIGNALS

- A **Program** is a **set of Instructions**.
- An **Instruction**, requires a **set of small** operations called **Micro-Operations**.
- A **Micro-Operation** is a **finite activity performed** by the processor **in one clock cycle**. One clock cycle is also called one **T-state** (Transition state).
- **One Micro-Operation** requires **One T-state**.
- Several **Independent** Micro-Operations can be performed in the same T-state.
- **Control unit generates control signals** to perform these very Micro-Operations.
- To understand Control Units, we must first clearly understand **Micro-Operations**.



MICRO-OPERATIONS FOR INSTRUCTION FETCHING

STATE	MICRO-OPERATION	EXPLANATION
T1:	MAR 🡐 PC	*PC puts address of the next instruction into MAR*
T2:	MBR 🡐 Memory (Instr)	*MBR gets instruction from memory through data bus*
T3:	IR 🡐 MBR	*IR gets instruction from MBR*
	PC 🡐 PC + 1	*PC gets incremented*

As we can see in the above table, two Micro-Operations can take place in the same T-state i.e.

IR🡐 MBR and PC🡐 PC + 1

This is because they are completely independent of each other.
In fact, PC becoming PC + 1 can also be performed in the 2nd T-state while the instruction is being fetched from the memory through the data bus into MBR.
This is shown below.

MICRO-OPERATIONS FOR INSTRUCTION FETCHING (ALTERNATE METHOD)

STATE	MICRO-OPERATION	EXPLANATION
T1:	MAR 🡐 PC	*PC puts address of the next instruction into MAR*
T2:	MBR 🡐 Memory (Instr)	*MBR gets instruction from memory through data bus*
	PC 🡐 PC + 1	*PC gets incremented*
T3:	IR 🡐 MBR	*IR gets instruction from MBR*

Please Note

PC 🡐 PC + 1 cannot take place in the 1st T-state.
That's because, in the 1st T-state, PC is providing the address on the address bus, through MAR. If at the same time PC gets incremented, then the incremented address will be put on the address bus.

Now that we know how an instruction is fetched, we can proceed further and learn Micro-Operations for various instructions, of different Addressing Modes.

MICRO-OPERATIONS FOR IMMEDIATE ADDRESSING MODE

E.g.: MOV R1, 25H; R1 register gets the immediate value 25H

STATE	MICRO-OPERATION	EXPLANATION
T1:	MAR ← PC	*PC puts address of the next instruction into MAR*
T2:	MBR ← Memory (Instr)	*MBR gets instruction from memory through data bus*
T3:	IR ← MBR	*IR gets instruction "MOV R1, 25H" from MBR*
	PC ← PC + 1	*PC gets incremented*
T4:	R1 ← 25H (IR)	*R1 register gets the value 25H from IR*

MICRO-OPERATIONS FOR REGISTER ADDRESSING MODE

E.g.: MOV R1, R2; R1 register gets the data from Register R2

STATE	MICRO-OPERATION	EXPLANATION
T1:	MAR ← PC	*PC puts address of the next instruction into MAR*
T2:	MBR ← Memory (Instr)	*MBR gets instruction from memory through data bus*
T3:	IR ← MBR	*IR gets instruction "MOV R1, R2" from MBR*
	PC ← PC + 1	*PC gets incremented*
T4:	R1 ← R2	*R1 register gets the value from R2 Register*

MICRO-OPERATIONS FOR DIRECT ADDRESSING MODE

E.g.: MOV R1, [2000H]; R1 register gets the data from memory location 2000H

STATE	MICRO-OPERATION	EXPLANATION
T1:	MAR ← PC	*PC puts address of the next instruction into MAR*
T2:	MBR ← Memory (Instr)	*MBR gets instruction from memory through data bus*
T3:	IR ← MBR	*IR gets instruction "MOV R1, [2000H]" from MBR*
	PC ← PC + 1	*PC gets incremented*
T4:	MAR ← IR (2000H)	*MAR gets the address 2000H from IR*
T5:	MBR ← Memory ([2000H])	*MBR gets contents of location 2000H from Memory.*
T6:	R1 ← MBR ([2000H])	*Register R1 gets contents of memory location 2000H from MBR*

MICRO-OPERATIONS FOR INDIRECT ADDRESSING MODE

E.g.: MOV R1, [R2]; R1 register gets the data from memory location pointed by R2

STATE	MICRO-OPERATION	EXPLANATION
T1:	MAR 🡐 PC	*PC puts address of the next instruction into MAR*
T2:	MBR 🡐 Memory (Instr)	*MBR gets instruction from memory through data bus*
T3:	IR 🡐 MBR	*IR gets instruction "MOV R1, [R2]" from MBR*
	PC 🡐 PC + 1	*PC gets incremented*
T4:	MAR 🡐 R2	*MAR gets the address R2 Register*
T5:	MBR 🡐 Memory ([R2])	*MBR gets contents of location pointed by R2 from Memory.*
T6:	R1 🡐 MBR ([R2])	*Register R1 gets contents of memory location pointed by R2 from MBR*
		In the exam, once, Add R1, [R2] was asked. *Everything else will be same. Only change: T6: R1 🡐 R1 + MBR*

MICRO-OPERATIONS FOR INDIRECT ADDRESSING MODE

E.g.: MOV [R2], R1; R1 register stores data into memory location pointed by R2

STATE	MICRO-OPERATION	EXPLANATION
T1:	MAR 🡐 PC	*PC puts address of the next instruction into MAR*
T2:	MBR 🡐 Memory (Instr)	*MBR gets instruction from memory through data bus*
T3:	IR 🡐 MBR	*IR gets instruction "MOV [R2], R1" from MBR*
	PC 🡐 PC + 1	*PC gets incremented*
T4:	MAR 🡐 R2	*MAR gets the address R2 Register*
T5:	MBR 🡐 R1	*R1 puts data into MBR to store it in the memory location pointed by R2.*

MICRO-OPERATIONS FOR IMPLIED ADDRESSING MODE

E.g.: STC; Set the Carry Flag; (CF 🡐 1).

STATE	MICRO-OPERATION	EXPLANATION
T1:	MAR 🡐 PC	*PC puts address of the next instruction into MAR*
T2:	MBR 🡐 Memory (Instr)	*MBR gets instruction from memory through data bus*
T3:	IR 🡐 MBR	*IR gets instruction "STC" from MBR*
	PC 🡐 PC + 1	*PC gets incremented*
T4:	CF 🡐 1	*Carry Flag in the Flag Register becomes 1*